

**MERANCANG ANIMASI MENGGUNAKAN LED CUBE  
8X8X8 BERBASIS ARDUINO UNO**

**SKRIPSI**

**Oleh :  
KHUMAIROH SILVIANI  
NPM : 202310011**



**PROGRAM STUDI TEKNOLOGI INFORMASI  
FAKULTAS INFORMATIKA DAN PARIWISATA  
INSTITUT BISNIS DAN INFORMATIKA  
KESATUAN BOGOR**

**2024**

**MERANCANG ANIMASI MENGGUNAKAN LED CUBE  
8X8X8 BERBASIS ARDUINO UNO**

**SKRIPSI**

Sebagai salah Satu Syarat  
Untuk Memperoleh Gelar Sarjana Teknologi Informasi  
Pada Program Studi Sarjana Teknologi Informasi

**Oleh :**

**KHUMAIROH SILVIANI**

NPM : 202310011



**PROGRAM STUDI TEKNOLOGI INFORMASI  
FAKULTAS INFORMATIKA DAN PARIWISATA  
INSTITUT BISNIS DAN INFORMATIKA  
KESATUAN BOGOR**

**2024**

# **MERANCANG ANIMASI MENGGUNAKAN LED CUBE 8X8X8 BERBASIS ARDUINO UNO**

## **SKRIPSI**

Telah diujikan dan disetujui pada sidang sarjana  
Institut Bisnis dan Informatika Kesatuan, pada:

Hari :

Tanggal :

Mengetahui,

Dekan Fakultas  
Informatika dan Pariwisata

Ketua Program Studi S1  
Teknologi Informasi

Dr. Ir. Jan Horas V. Purba,  
M.Si., CBPA<sup>®</sup>, CPSP<sup>®</sup>

Edi Nurachmad, S.Kom., M.Kom

**MERANCANG ANIMASI MENGGUNAKAN LED CUBE  
8X8X8 BERBASIS ARDUINO UNO**

**SKRIPSI**

Telah disetujui oleh,

Pembimbing

Anton Sukamto, S.Kom., M.TI.

Telah diujikan pada Sidang Sarjana dan dinyatakan LULUS

Pada tanggal seperti tertera dibawah ini

Bogor, ..... 2024

Dosen Penguji

Penguji I

Penguji II

Isnan Mulia, S.Komp., M.Kom

Dr. Nusa Muktiadji, Ir., M.M

## SURAT PERNYATAAN

Yang bertanda tangan dibawah ini :

Nama : Khumairoh Silviani

NPM : 202310011

Program Studi : S-1 Teknologi Informasi

Judul Skripsi : *Merancang Animasi Menggunakan Led Cube 8x8x8 Berbasis Arduino Uno*

Menyatakan dengan sebenar-benarnya bahwa segala pernyataan dalam Skripsi ini adalah hasil pekerjaan saya sendiri, dengan bimbingan Pembimbing, kecuali jika ada penunjukan rujukan yang jelas. Skripsi ini belum pernah diajukan untuk memperoleh gelar pada program sejenis di Perguruan Tinggi lain. Semua sumber data dan informasi yang digunakan telah dinyatakan secara jelas dan dapat diperiksa kebenarannya. Jika ternyata terbukti pernyataan ini tidak benar, sepenuhnya menjadi tanggung jawab saya.

Bogor, 04 Juni 2024

Yang Menyatakan,

Khumairoh Silviani

## ABSTRAK

KHUMAIROH SILVIANI, NPM: 2023100011 (2024). Merancang Animasi Menggunakan Led Cube 8x8x8 Berbasis Arduino Uno. Dibawah Bimbingan ANTON SUKAMTO.

---

Penelitian ini bertujuan merancang dan mengimplementasikan animasi menggunakan LED *Cube* 8x8x8 yang dikendalikan oleh mikrokontroler Arduino Uno. LED *Cube* adalah matriks tiga dimensi terdiri dari 512 LED yang dapat menampilkan pola cahaya dan animasi kompleks. Arduino Uno digunakan sebagai pusat kendali untuk mengatur nyala dan padamnya LED secara presisi sehingga membentuk animasi yang diinginkan. Metode penelitian meliputi perancangan *hardware* dan *software*. Pada tahap *hardware*, LED disusun dalam formasi kubus 8x8x8 dan dihubungkan dengan Arduino Uno melalui rangkaian *driver transistor* untuk memastikan aliran arus memadai. Pada tahap *software*, Arduino diprogram menggunakan bahasa C++ dalam Arduino IDE, dengan algoritma yang dirancang untuk menampilkan berbagai animasi. Hasil penelitian menunjukkan bahwa LED *Cube* 8x8x8 berhasil menampilkan animasi seperti pola bergerak, teks berjalan, dan efek visual tiga dimensi. Arduino Uno terbukti efektif dalam mengelola dan mengontrol LED secara simultan. Sistem ini dapat dikembangkan lebih lanjut untuk aplikasi lebih kompleks, seperti tampilan visual interaktif atau media edukasi.

Kata kunci: LED Cube, Arduino Uno, animasi, mikrokontroler, tiga dimensi.

## ***ABSTRACT***

KHUMAIROH SILVIANI, NPM: 2023100011 (2024). *Designing Animations Using an 8x8x8 LED Cube Based on Arduino Uno. Supervised by ANTON SUKAMTO.*

---

*This research aims to design and implement animations using an 8x8x8 LED Cube controlled by an Arduino Uno microcontroller. The LED Cube is a three-dimensional matrix consisting of 512 LEDs that can display complex light patterns and animations. The Arduino Uno is used as the control center to precisely manage the turning on and off of the LEDs to form the desired animations. The research method includes hardware and software design. In the hardware stage, the LEDs are arranged in an 8x8x8 cube formation and connected to the Arduino Uno through a transistor driver circuit to ensure an adequate current flow. In the software stage, the Arduino is programmed using C++ in the Arduino IDE, with algorithms designed to display various animations. The research results show that the 8x8x8 LED Cube successfully displays animations such as moving patterns, scrolling text, and three-dimensional visual effects. The Arduino Uno has proven effective in managing and controlling the LEDs simultaneously. This system can be further developed for more complex applications, such as interactive visual displays or educational media.*

*Keywords: LED Cube, Arduino Uno, animation, microcontroller, three-dimensional.*

## KATA PENGANTAR

Dengan penuh rasa syukur kepada Allah SWT, penulis ingin mengirimkan Sholawat kepada Nabi Muhammad SAW, serta kepada keluarga dan para sahabatnya. Ini merupakan ungkapan terima kasih atas berhasilnya penyelesaian Proposal Skripsi, yang menjadi salah satu syarat untuk meraih gelar sarjana Teknologi Informasi di Institut Bisnis dan Informatika Kesatuan. Dengan tulus hati dan penuh hormat, penulis ingin mengungkapkan penghargaan mendalam kepada semua yang telah memberikan dukungan, baik secara moril maupun materil. khususnya kepada :

1. Allah SWT, yang melimpahkan berkah dan rahmat-Nya sehingga penulis dapat menyelesaikan laporan kuliah kerja praktek ini.
2. Prof. Dr. Bambang Pamungkas, Ak., MBA, CA, CPA, CPA (Aust), ASEAN CPA, CIMBA, CSFA, CFrA, CGAE, Rektor Institut Bisnis Dan Informatika Kesatuan.
3. Bapak Edi Nurachmad, S.Kom., M.Kom, Ketua Program Studi Teknologi Informasi.
4. Bapak Anton Sukamto, S.Kom., M. TI, selaku dosen pembimbing yang telah dengan kesabaran memberikan bimbingan dan arahan dalam tahapan penyusunan Proposal Skripsi ini.
5. Orang tua dan keluarga yang selalu memberikan semangat, dukungan, dan doa yang sangat memotivasi penulis.
6. Seluruh dosen di IBI Kesatuan yang telah memberikan ilmu pengetahuan selama perkuliahan.

Penulis sadar bahwa Proposal Skripsi ini masih memiliki kekurangan. Oleh karena itu, penulis sangat mengharapkan masukan, kritik, dan saran yang membangun untuk perbaikan penyusunan Proposal Skripsi ini.

Bogor, Mei 2024

Penulis



## DAFTAR ISI

ABSTRAK .....	i
<i>ABSTRACT</i> .....	ii
KATA PENGANTAR .....	iii
DAFTAR ISI.....	iv
DAFTAR TABEL.....	vii
DAFTAR LAMPIRAN.....	viii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang Penelitian .....	1
1.2 Rumusan Masalah .....	2
1.3 Tujuan Penelitian .....	2
1.4 Manfaat Penelitian .....	2
1.5 Ruang Lingkup.....	2
BAB II TINJAUAN PUSTAKA.....	3
2.1 Dasar Teori.....	3
2.1.1 Animasi .....	3
2.1.2 Software .....	3
2.1.3 LED .....	4
2.1.4 Arduino Uno .....	4
2.1.5 Push Button .....	4
2.1.6 Brass Wire .....	5
2.1.7 Transistors .....	5
2.1.8 Resistor.....	6
2.1.9 74HC595 Shift Registers .....	6
2.1.10 PCB .....	7
2.1.11 Cable AWG24.....	7
2.1.12 Soldering .....	8
2.1.13 Papan PVC .....	8
2.2 Flowchart .....	9
2.3 Penelitian Terdahulu .....	11
BAB III METODOLOGI PENELITIAN.....	14
3.1 Waktu dan Tempat Penelitian .....	14

3.2 Alat dan Bahan.....	14
3.2.1 Alat.....	14
3.2.2 Bahan .....	14
3.3 Prosedur Kerja.....	13
3.3.1 Analisis Kebutuhan .....	13
3.3.2 Membuat Desain .....	14
3.3.3 Bangun Prototipe.....	16
3.3.4 Evaluasi Prototipe .....	16
3.3.5 Perbaikan Prototipe .....	16
3.3.6 Implementasi .....	16
3.4 Analisis Data .....	16
3.5 Jadwal Penelitian.....	17
BAB IV HASIL DAN PEMBAHASAN .....	18
4.1 Analisis Kebutuhan .....	18
4.1.1 Perangkat Animasi LED .....	18
4.1.2 Instalasi Perangkat Lunak (Software) .....	19
4.2 Perancangan Rangkaian LED <i>Cube</i> 8x8x8.....	20
4.2.1 Diagram Blok.....	21
4.3 Pembangunan Prototipe .....	22
4.3.1 Penginputan Program .....	22
4.3.2 Implementasi Rancangan Prototipe LED Cube .....	22
4.3.3 Implementasi Rangkaian Elektronika Dengan Arduino Uno.....	23
4.4 Tahap Pengujian Alat.....	24
4.4.1 Pengujian Tegangan Animasi LED Cube .....	24
4.4.2 Durasi Animasi LED Cube .....	25
BAB V KESIMPULAN DAN SARAN.....	26
KESIMPULAN DAN SARAN.....	26
5.1 Kesimpulan .....	26
5.2 Saran.....	26
DAFTAR PUSTAKA .....	28
LAMPIRAN - LAMPIRAN.....	29

## DAFTAR GAMBAR

Gambar 2.1 LED .....	4
Gambar 2.2 Arduino Uno.....	4
Gambar 2.3 Push Button .....	5
Gambar 2.4 Brass Wire .....	5
Gambar 2.5 Transistors .....	6
Gambar 2.6 Resistors .....	6
Gambar 2.7 74HC595 Shift Registers.....	7
Gambar 2.8 PCB .....	7
Gambar 2.9 Cable AWG .....	8
Gambar 2.10 Soldering .....	8
Gambar 2.11 Papan PVC .....	9
Gambar 3.1 Tahapan Penelitian .....	13
Gambar 3.2 Flowchart Proses Aplikasi LED Cube .....	15
Gambar 4.1 Tampilan Awal Software IDE.....	20
Gambar 4.2 Skema Rangkaian LED Cube 8x8x8.....	20
Gambar 4.3 Diagram Blok .....	21
Gambar 4.4 Tampilan Arduino IDE Setelah Terupload .....	22
Gambar 4.5 Rancangan Prototipe LED Cube 8x8x8 .....	23
Gambar 4.6 Rancangan Elektronika Dengan Arduino Uno.....	23

## DAFTAR TABEL

Tabel 2.1 Flowchart .....	9
Tabel 2.2 Penelitian Terdahulu .....	11
Tabel 3.1 Kebutuhan Alat .....	14
Tabel 3.2 Kebutuhan Bahan .....	14
Tabel 3.3 Jadwal Penelitian.....	17
Tabel 4.1 Bahan Yang Digunakan .....	18
Tabel 4.2 Pengujian Tegangan Animasi .....	24
Tabel 4.3 Durasi Animasi LED Cube .....	25

## **DAFTAR LAMPIRAN**

Lampiran 1 Gambar Beberapa Animasi.....	30
Lampiran 2 Sorce Code.....	35

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Penelitian**

Pemanfaatan teknologi dalam beragam aspek kehidupan manusia terus mengalami perkembangan pesat sejalan dengan kemajuan teknologi. Salah satu aspek menarik yang berkaitan dengan penerapan teknologi adalah animasi visual. Animasi memegang peranan yang sangat penting dalam berbagai bidang, termasuk hiburan, pendidikan, seni, dan komunikasi. Animasi memungkinkan kita untuk menyampaikan pesan, menghadirkan ide-ide, serta menciptakan pengalaman visual yang memukau. Untuk mencapai efek visual yang mengagumkan, khususnya dalam konteks animasi tiga dimensi (3D), diperlukan berbagai perangkat keras dan perangkat lunak canggih.

Pertumbuhan teknologi *mikrokontroler* dan mikroprosesor juga telah membuka peluang tidak terbatas dalam pengembangan berbagai proyek elektronika. Salah satu aplikasi menarik dari teknologi ini adalah dalam pembuatan animasi LED. Animasi LED adalah seni mengatur lampu LED secara berurutan untuk menciptakan tampilan visual yang menarik. Animasi ini bisa digunakan dalam berbagai konteks, mulai dari tampilan iklan hingga seni instalasi.

Animasi 8x8x8 dengan LED merupakan jenis animasi yang menggabungkan susunan matriks tiga dimensi dari 8x8x8 lampu LED, menghasilkan tampilan tiga dimensi yang mengesankan. Keistimewaan dari proyek ini terletak pada kemampuan untuk menciptakan efek visual yang menakjubkan melalui berbagai kombinasi warna dan pola yang dapat diprogram. Prinsip dasar operasi LED adalah menampilkan setiap segmen dari kubus dengan mengaktifkan LED sesuai dengan pola yang telah diatur melalui kode pemrograman yang ditulis dalam bahasa C++. Berdasarkan penjelasan ini, penulis tertarik untuk melakukan penelitian yang berjudul **"Merancang Animasi Menggunakan LED Cube 8x8x8 Berbasis Arduino Uno"**.

## **1.2 Rumusan Masalah**

Berdasarkan penjelasan yang tertera di latar belakang tersebut, penulis memfokuskan penelitian berkaitan dengan "Bagaimana cara menyusun rangkaian LED *cube* dan memberikan program pada Arduino uno untuk menciptakan animasi yang berbeda pada LED 8x8x8?"

## **1.3 Tujuan Penelitian**

Berdasarkan rumusan masalah pada uraian di atas, maka penulis membuat tujuan penelitian untuk menciptakan rangkaian LED 8x8x8 dan pola animasi yang bermacam-macam, hal ini melibatkan perancangan dan implementasi algoritma atau logika pemrograman yang memungkinkan perubahan pola, pergerakan, transisi dan efek visual pada LED 8x8x8.

## **1.4 Manfaat Penelitian**

Harapan penulis penelitian ini berguna untuk :

### **1. Akademik**

Hasil penelitian ini di harapkan bisa memberikan pengetahuan pada bidang perkembangan ilmu pengetahuan di aspek akademik dalam merancang animasi LED 8x8x8 menggunakan Arduino Uno dan sebagai referensi untuk peneliti selanjutnya yang khususnya bagi mahasiswa dan mahasiswi yang akan menyusun proposal skripsi.

### **2. Secara Praktek**

Dengan harapan penelitian yang di buat saat ini bisa menambah ilmu pengetahuan sekaligus memperoleh pengalaman dalam merancang animasi LED 8x8x8 menggunakan Arduino Uno.

## **1.5 Ruang Lingkup**

Penelitian ini akan berfokus pada pengembangan animasi LED dengan menggunakan matriks tiga dimensi berukuran 8x8x8 yang artinya terdapat 512 lampu yang dapat dikendalikan secara individu.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Dasar Teori**

##### **2.1.1 Animasi**

Animasi adalah proses yang menciptakan kesan gerakan atau perubahan dalam objek atau gambar yang awalnya statis dengan cara menggabungkan serangkaian gambar atau frame secara berurutan. Tujuan utama dari animasi adalah menciptakan tampilan visual yang menyerupai pergerakan atau perubahan, sehingga menciptakan kesan adanya aktivitas atau perubahan yang terjadi dalam objek atau gambar tersebut.

Dalam merancang animasi led 8x8x8 menggunakan arduino, animasi dapat merujuk pada berbagai pola cahaya atau gambar yang ditampilkan pada tampilan led matriks. Beberapa contoh animasi yang dapat dirancang, yaitu :

1. Animasi bergerak : dalam animasi ini, pola cahaya bergerak di sepanjang tampilan led 8x8x8. Contohnya, animasi dapat menampilkan bola bergerak dari satu sisi ke sisi lainnya atau objek bergerak secara diagonal.
2. Animasi Transisi : animasi transisi melibatkan perubahan mulus antara satu pola cahaya atau gambar ke pola cahaya. Contohnya, animasi dapat menggunakan efek *fading* untuk secara perlahan menghilangkan satu pola cahaya dan menggantinya dengan pola cahaya yang baru.
3. Animasi Efek Visual : Animasi ini melibatkan pengguna efek visual yang menarik, seperti perubahan warna, memancarkan sinar atau modulasi intensitas cahaya. Contohnya, animasi dapat membuat pola cahaya berkedip dengan kecepatan tertentu atau mengatur pola cahaya untuk membentuk gambar tertentu.

##### **2.1.2 Software**

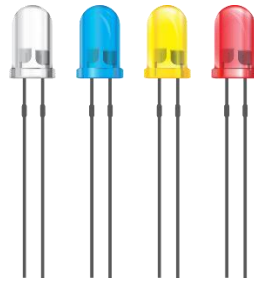
Untuk mengembangkan proyek Arduino, perangkat lunak utama yang digunakan adalah Arduino IDE. Meskipun ada berbagai perangkat lunak lain yang juga bermanfaat, IDE (Integrated Development Environment) adalah program



yang khusus digunakan di komputer untuk merancang dan membuat sketsa program untuk papan Arduino.

### 2.1.3 LED

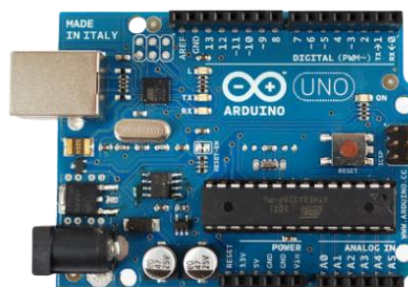
LED (*Light Emitting Diode*) adalah suatu komponen elektronika yang memiliki kemampuan untuk menghasilkan cahaya *monokromatik* dengan mengalirkan tegangan maju



Gambar 2.1 LED

### 2.1.4 Arduino Uno

Arduino Uno merupakan sebuah papan mikrokontroler yang menggunakan mikrokontroler ATmega328 sebagai basisnya. Papan ini memiliki 14 pin input/output digital, di mana 6 pin dapat difungsikan sebagai output PWM, serta 6 pin input analog. Selain itu, papan ini dilengkapi dengan osilator kristal 16 MHz, koneksi USB, dan tombol reset.



Gambar 2.2 Arduino Uno

### 2.1.5 Push Button

*Push Button* merupakan perangkat sederhana yang berfungsi untuk menghubungkan atau memutuskan aliran arus listrik dengan cara menekan tombolnya dalam sistem kerja tekan *unlock* (tidak mengunci).



Gambar 2.3 Push Button

#### 2.1.6 Brass Wire

*Brass wire* atau kawat kuningan, merupakan bahan yang sangat bermanfaat dan serbaguna yang sering digunakan dalam berbagai industri karena memiliki sifat yang unik. Terbuat dari campuran tembaga dan seng, kawat kuningan memiliki banyak karakteristik yang bermanfaat yang diwarisi dari logam-logam komponennya.

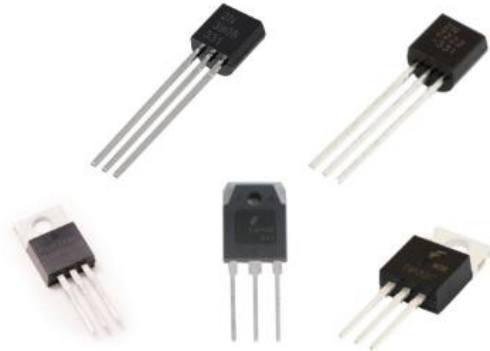


Gambar 2.4 Brass Wire

#### 2.1.7 Transistors

*Transistor* adalah sebuah komponen semikonduktor yang digunakan untuk memperkuat sinyal, mengatur aliran arus, menghubungkan dan memutuskan sirkuit listrik, menjaga stabilitas tegangan serta mengatur modulasi sinyal. Mirip

dengan fungsi kran listrik, *transistors* memungkinkan pengendalian aliran listrik yang sangat presisi tergantung pada arus inputnya (BJT) atau tegangan inputnya (FET).



Gambar 2.5 Transistors

### 2.1.8 Resistor

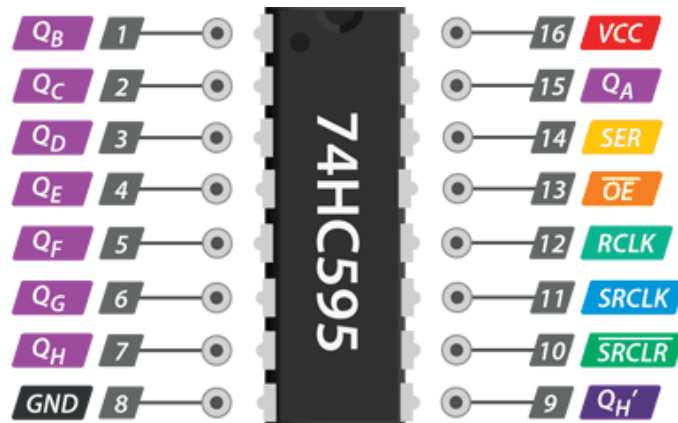
*Resistor* merupakan sebuah komponen dalam dunia elektronika yang berfungsi untuk mengontrol aliran arus listrik. Nilai tahanan dari *resistor* diukur dalam satuan ohm dan biasanya direpresentasikan dengan simbol  $\Omega$ . Informasi mengenai nilai tahanan *resistor* seringkali disampaikan melalui kode warna yang terdapat pada fisik *resistor* tersebut.



Gambar 2.6 Resistors

### 2.1.9 74HC595 Shift Registers

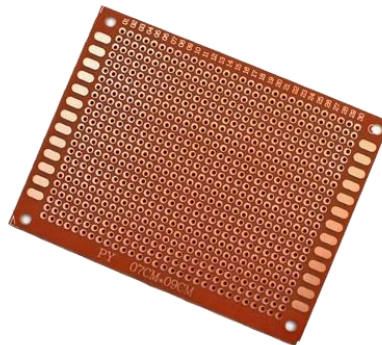
*Shift register* adalah komponen yang digunakan untuk menggeser data serial masuk (*serial input*) menjadi data paralel keluar (*parallel output*) dalam rangkaian elektronik.



Gambar 2.7 74HC595 Shift Registers

#### 2.1.10 PCB

PCB (*Printed Circuit Board*) adalah plat atau *substrat* datar yang biasanya terbuat dari bahan isolator seperti *fiberglass*, yang memiliki jalur tembaga yang dicetak pada permukaannya. PCB digunakan dalam berbagai perangkat elektronik untuk menghubungkan dan mendukung komponen-komponen elektronik seperti *transistor*, *resistor*, mikrokontroler, dan lainnya.



Gambar 2.8 PCB

#### 2.1.11 Cable AWG24

*Cable AWG* adalah sistem pengukuran yang digunakan untuk mengklasifikasikan ukuran kabel listrik dan logam. Semakin kecil nomornya, semakin besar diameter kabelnya. Kabel dengan nomor AWG yang lebih kecil

memiliki kapasitas arus yang lebih besar daripada kabel dengan nomor AWG yang lebih besar.



Gambar 2.9 Cable AWG24

#### **2.1.12 Soldering**

*Soldering* merupakan alat yang digunakan dalam proses penyolderan atau *soldering*. Alat ini menghasilkan suhu tinggi pada ujungnya, yang berguna untuk mencairkan solder sehingga dapat digunakan untuk menggabungkan atau memperbaiki.



Gambar 2.10 Soldering

#### **2.1.13 Papan PVC**

Papan PVC merupakan materi yang terdiri dari polivinil klorida (PVC), sejenis polimer termoplastik. Papan ini dikenal atas kekuatannya, daya tahan terhadap cuaca, dan sifat tahan korosi serta serangan kimia. PVC umumnya

dimanfaatkan dalam beragam penggunaan, mencakup bidang konstruksi, industri, dan produksi bahan tahan air.

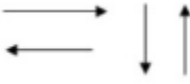




Gambar 2.11 Papan PVC

## 2.2 Flowchart

*Flowchart* atau bagan alir merupakan diagram yang menggambarkan langkah-langkah dan keputusan yang diperlukan untuk menjalankan sebuah proses dalam suatu program. Setiap langkah direpresentasikan dalam bentuk diagram dan dihubungkan oleh garis atau panah penunjuk arah.

Tabel 2.1 Flowchart

Simbol	Keterangan
	<i>Flow Direction</i> : Simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini di sebut juga <i>connecting line</i> .
	Terminal : Simbol untuk pemulaan atau akhir dari suatu program.
	Proses : Simbol yang digunakan untuk menggambarkan suatu aksi atau langkah dalam suatu <i>flowchart</i> .



*Input* atau *Output* : Simbol yang digunakan untuk merepresentasikan *input* atau *output* dalam suatu *flowchart*.

---



*Predefined Process* : Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan didalam *storage*.

---



*Decision* : Simbol untuk kondisi yang akan menghasilkan beberapa kemungkinan jawaban atau aksi.

---



*Off-page Connector* : Simbol untuk penghubung antar halaman berbeda.

---

## 2.3 Penelitian Terdahulu

Tabel 2.2 Penelitian Terdahulu

Judul Penelitian	Nama Penelitian	Aspek					
		Arduino Uno	Led Blue	Software IDE	Remote Control	Led RGB	PCB
Rancang Bangun Text dan Animasi 3 Dimensi Persistence of Vision Pada Led Cube Berbasis Arduino Uno	Alwi Fadli, Mardiana, ST, M.Kom dan Rismayanti, ST, M.Kom	✓	✓	✓	-	-	-
Rancang Bangun LED Cube 8x8x8	Jaka Romadon, Djiryanul Fadlie dan Hesa Prasandi Iqro	✓	✓	✓	✓	-	✓
Rancang Bangun LED Cube 2x2x2 Berbasis Android Uno Sebagai Bahan Ajar Kreativitas Anak Sekolah	Kristian A. Dame dan Victori S.G Polly	✓	✓	✓	-	-	-
Merancang Animasi LED 8x8x8 Menggunakan Arduino Uno	Khumairoh Silviani	✓	-	✓	-	✓	✓
Rancang Bangun Text dan Animasi 3 Dimensi Pada LED Cubie Berbasis Arduino Uno Atmega 328	Muhammad Iqbal, Reza Nandika dan Endang Susanti	✓	✓	✓	✓	-	-



Dalam membandingkan penelitian yang dilakukan oleh penulis dengan beberapa penelitian lain yang tercantum dalam tabel 2.2, kita dapat melihat berbagai aspek seperti tujuan, metodologi, teknologi yang digunakan, dan hasil yang dicapai. Mari kita menjelajahi perbandingan ini dengan lebih mendalam.

Namun, penelitian oleh Kristian A. Dame dan Victori S.G Polly mengembangkan LED cube 2x2x2 dengan fokus pada meningkatkan kreativitas anak-anak sekolah. Mereka menggunakan Arduino Uno dan *Software IDE* untuk pengembangan perangkat lunak, serta Led *blue* sebagai sumber pencahayaan utama. Meskipun tidak menggunakan Led RGB untuk variasi warna yang lebih kompleks seperti pada penelitian lain, penggunaan Led *blue* tetap memberikan nilai tambah dalam mendukung pengalaman interaktif dan pembelajaran anak-anak

Jaka Romadon, Djiryanul Fadlie, dan Hesa Prasandi Iqro juga melakukan penelitian dengan LED cube, namun dalam skala yang lebih besar yaitu 8x8x8. Mereka menggunakan Arduino Uno dan *Software IDE* seperti khumairoh silviani, tetapi mereka juga mengimplementasikan fitur *remote control* untuk pengendalian jarak jauh dan PCB untuk manajemen komponen yang lebih baik. Penelitian ini menunjukkan aplikasi yang lebih kompleks dari teknologi LED *cube* dalam konteks kontrol dan integrasi sistem. Tetapi, penelitian ini tidak menggunakan LED RGB yang digunakan oleh khumairoh silviani.

Muhammad Iqbal, Reza Nandika, dan Endang Susanti juga memfokuskan penelitian mereka pada animasi 3D dengan menggunakan Arduino Uno. Mereka mengimplementasikan *Software IDE* untuk pengembangan perangkat lunak dan Remote Control untuk mengontrol animasi dari jarak jauh. Namun, mereka tidak menggunakan LED RGB atau PCB seperti yang dilakukan oleh khumairoh silviani, menunjukkan pendekatan yang berbeda dalam pengembangan animasi dan efek visual.

Alwi Fadli, Mardiana, ST, M.Kom, dan Rismayanti, ST, M.Kom mengembangkan animasi 3D menggunakan konsep *Persistence of Vision* (POV) pada LED *cube* mereka, menggunakan Arduino Uno dan *Software IDE*. Penelitian ini fokus pada efek visual yang unik dan teknik animasi, dengan tidak menggunakan fitur tambahan seperti LED RGB, *Remote Control*, atau PCB seperti dalam penelitian khumairoh silviani.

Penulis dalam penelitiannya mengembangkan LED *cube* berukuran 8x8x8 menggunakan Arduino Uno sebagai *platform* utama. Penelitian ini memiliki fokus utama pada penggunaan teknologi LED RGB untuk menciptakan efek visual yang dinamis dan menarik. Selain itu, Silviani juga mengimplementasikan PCB (*Printed Circuit Board*) untuk mengintegrasikan komponen elektronik dengan lebih terstruktur. Tujuan dari penelitian ini adalah untuk menghasilkan animasi yang kreatif dan menarik melalui penggunaan animasi LED *cube* yang canggih.

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Waktu dan Tempat Penelitian**

Dalam perancangan Proposal Skripsi ini, waktu dan tempat penelitian yang sesuai telah dijadwalkan. Penelitian ini direncanakan akan dimulai pada bulan Oktober 2023 dan tempat pelaksanaannya adalah di Kota Bogor lebih spesifiknya di Laboratorium Workshop Institut Bisnis dan Informatika Kesatuan yang akan berlangsung selama 7 bulan. Selain itu, sebagian penelitian juga akan dilakukan di rumah penulis. Hal ini bertujuan untuk meningkatkan efisiensi dalam penggunaan tenaga dan waktu penelitian.

#### **3.2 Alat dan Bahan**

##### **3.2.1 Alat**

Tabel 3.1 Kebutuhan Alat

<b>Nama Barang</b>	<b>Kegunaan</b>
<b>Arduino IDE</b>	Mengkomposisi kode program ke dalam Arduino untuk mengembangkan animasi pada kubus LED 8x8x8.

##### **3.2.2 Bahan**

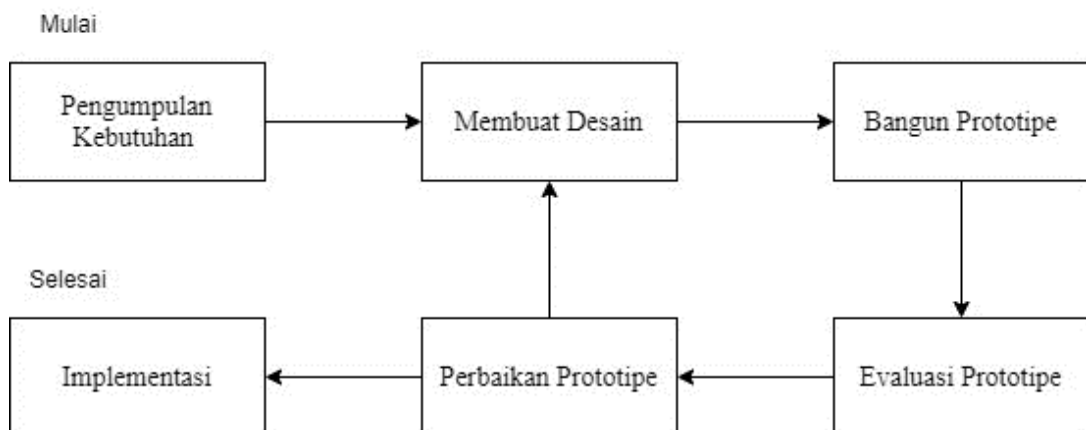
Tabel 3.2 Kebutuhan Bahan

<b>Nama Bahan</b>	<b>Jumlah</b>
<b>Arduino Uno</b>	1 pcs
<b>LED RGB</b>	512 pcs
<b>Push Button</b>	1 pcs
<b>Papan PVC</b>	1 lembar
<b>Transistors BD241</b>	8 pcs
<b>Resistor 220 ohm</b>	64 pcs
<b>Brass Wire</b>	83 meter
<b>Kabel AWG 24</b>	10 meter

<b>Shift Registers 74hc595</b>	9 pcs
<b>PCB</b>	1 pcs
<b>Soldering</b>	1 pcs

### 3.3 Prosedur Kerja

Prosedur kerja adalah serangkaian langkah atau tahap yang digunakan dalam pelaksanaan penelitian untuk memberikan struktur dan sistematisasi pada prosesnya. Dalam konteks pengembangan sistem, terdapat serangkaian metode yang umumnya dikenal sebagai *System Development Life Cycle* (SDLC). SDLC ini memiliki berbagai model yang dapat digunakan, dan dalam penelitian ini, model yang digunakan adalah model *Prototyping*. Berikut adalah langkah-langkah dan penjelasan lebih rinci dalam penerapan metode *Prototyping* :



Gambar 3.1 Tahapan Penelitian

#### 3.3.1 Analisis Kebutuhan

Langkah awal dalam prosedur penelitian mengenai animasi pada kubus LED 8x8x8 adalah melakukan pengumpulan kebutuhan dan menganalisis sistem.

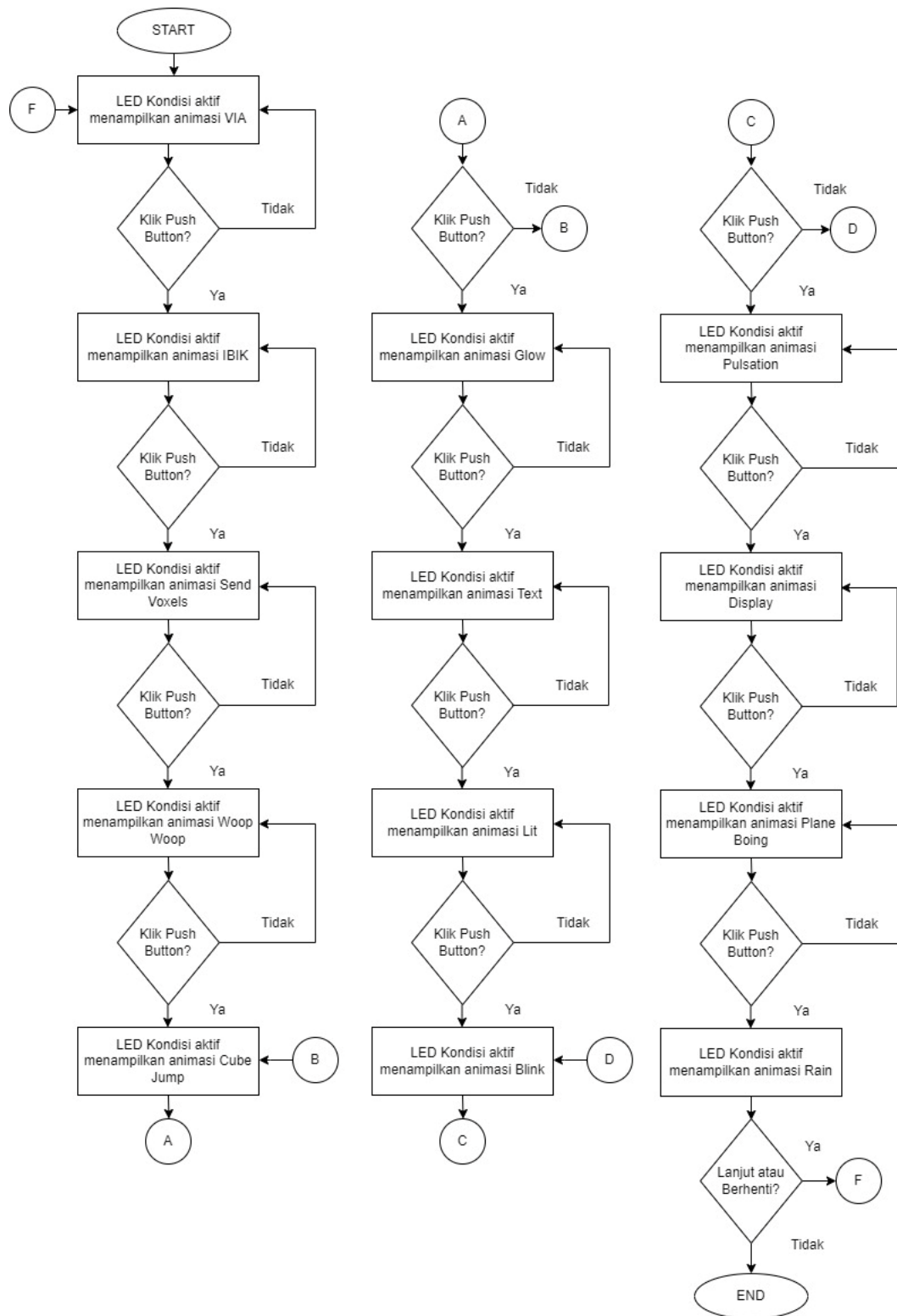
##### A. Pengumpulan Kebutuhan

1. Observasi : Mulailah dengan mengumpulkan informasi dan data tentang komponen yang akan digunakan, seperti LED 8x8x8, Arduino Uno, dan komponen lainnya. Amati dan pelajari contoh animasi LED serupa sebagai perbandingan.

2. Studi Pustaka : Kumpulkan informasi dari berbagai sumber seperti buku, jurnal, dan literatur yang relevan untuk digunakan sebagai panduan dalam mengembangkan animasi LED.

### **3.3.2 Membuat Desain**

Pada fase desain ini, penulis merencanakan susunan sistem dengan menggunakan *flowchart*. Tujuannya adalah untuk memberikan panduan yang jelas dalam proses pengembangan animasi LED 8x8x8 dengan Arduino Uno, dengan harapan mengurangi potensi kebingungan selama tahapan pengembangan. Selain itu, penulis juga merancang rangkaian digital yang mendukung animasi LED 8x8x8 ini untuk mempermudah proses perakitan dengan Arduino dan komponen yang dibutuhkan.



**Gambar 3.2 Flowchart Proses Aplikasi LED Cube**

### **3.3.3 Bangun Prototipe**

Berdasarkan sistem yang telah dirancang melalui tahapan desain, selanjutnya penulis melakukan perakitan prototipe Animasi LED 8x8x8 menggunakan Arduino Uno. Proses perakitan ini melibatkan Arduino Uno yang telah dikoneksikan dengan berbagai komponen pendukung untuk menghasilkan animasi yang diinginkan.

### **3.3.4 Evaluasi Prototipe**

Evaluasi prototipe animasi LED yang telah dibuat dilakukan pemeriksa lagi apakah animasi berjalan sesuai rencana atau tidak.

### **3.3.5 Perbaikan Prototipe**

Jika ada kendala yang masih ada, ulangi tahap perancangan dan perbaikan hingga prototipe sesuai dengan rencana.

### **3.3.6 Implementasi**

Jika prototipe telah berhasil, lanjutkan dengan tahap implementasi. Misalnya, letakkan animasi LED 8x8x8 dalam wadah atau proyek miniatur.

## **3.4 Analisis Data**

Analisis data yang digunakan dalam penelitian bertujuan untuk mengetahui kinerja animasi LED 8x8x8 menggunakan Arduino Uno terhadap waktu responnya, data analisis yang digunakan dalam penelitian ini meliputi visualisasi animasi, kestabilan sistem terhadap respon waktu LED *cube*, dimana penelitian ini sesuai dengan pengukuran kuantitatif yaitu ingin mengamati hasil pengaruh visual animasi, respon waktu terhadap waktu respon LED *cube*.

### 3.5 Jadwal Penelitian

Tabel 3.3 Jadwal Penelitian

No	Nama Kegiatan	Tahapan Penelitian						
		Okt 2023	Nov 2023	Des 2023	Jan 2024	Feb 2024	Maret 2024	April 2024
1	Pengumpulan Kebutuhan							
2	Pembuatan Desain							
3	Membangun Prototipe							
4	Evaluasi dan Perbaikan							
5	Implementasi							



## **BAB IV**

### **HASIL DAN PEMBAHASAN**

Dalam konteks yang telah dijelaskan pada bab sebelumnya, penelitian ini mengadopsi Metode *Prototyping*. Metode ini merupakan teknik pengembangan sistem yang menggunakan prototipe untuk mengilustrasikan sistem secara lebih konkret, memungkinkan pengguna atau pemilik sistem memahami dengan lebih baik sistem yang akan dibangun. Tahapan dalam model prototipe ini meliputi analisis kebutuhan, desain cepat, pembuatan prototipe, evaluasi, perbaikan prototipe, serta implementasi dan pemeliharaan.

#### **4.1 Analisis Kebutuhan**

##### **4.1.1 Perangkat Animasi LED**

Untuk merancang animasi LED diperlukan berbagai alat dan bahan, antara lain :

##### **A. Bahan yang Digunakan Dalam Merancang Animasi LED**

Tabel 4.1 Bahan yang Digunakan

<b>Nama Bahan</b>	<b>Jumlah</b>
<b>Arduino Uno</b>	1 pcs
<b>LED RGB</b>	512 pcs
<b>Push Button</b>	1 pcs
<b>Transistors BD241</b>	8 pcs
<b>Resistor 220ohm</b>	64 pcs
<b>Brass Wire</b>	83 meter
<b>Kabel AWG 24</b>	10 meter
<b>Shift Registers 74HC595</b>	9 pcs
<b>PCB</b>	1 pcs
<b>Soldering</b>	1 pcs
<b>Papan PVC</b>	1 lembar

## **B. Kebutuhan perangkat lunak (*software*) yang diperlukan**

Beberapa perangkat yang diperlukan termasuk:

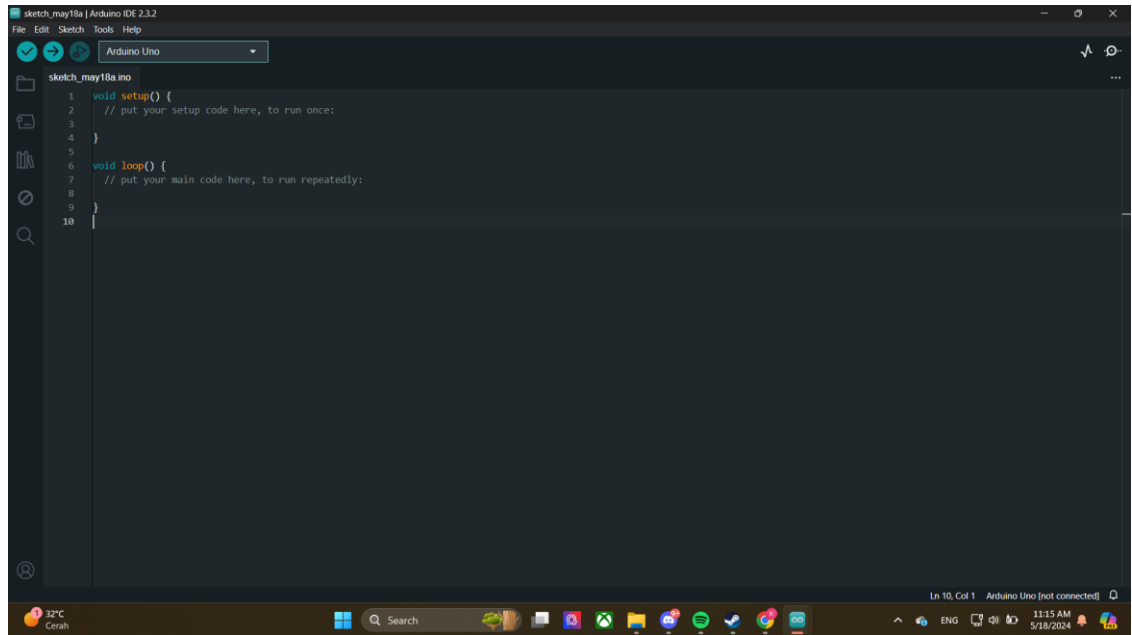
- a. *Software IDE (Integrate Development Environment)*
- b. Operating System Microsoft Windows 11

### **4.1.2 Instalasi Perangkat Lunak (Software)**

Untuk menginstruksikan program, Anda akan menggunakan perangkat lunak Arduino IDE. Berikut adalah langkah-langkah umum untuk menginstal perangkat lunak Arduino *Software* (IDE) yang diperlukan dalam pembuatan animasi LED 8x8x8 menggunakan Arduino Uno:

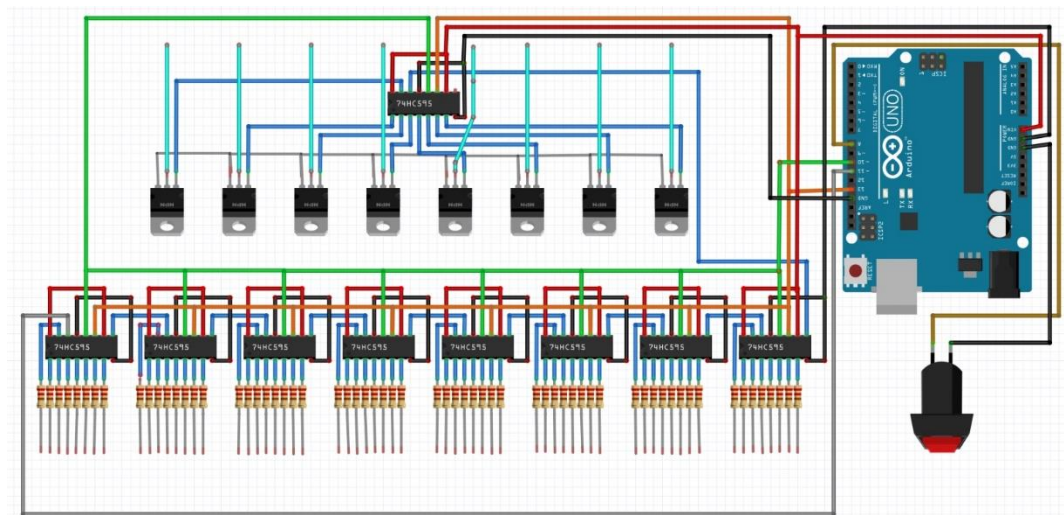
1. Unduh Arduino IDE dari situs resmi Arduino. Anda dapat mengunjungi <https://www.arduino.cc/en/software> untuk mengakses halaman unduhan.
2. Pilih versi yang sesuai dengan sistem operasi yang Anda gunakan (*Windows, macOS, atau Linux*).
3. Setelah selesai mengunduh, buka file instalasi Arduino IDE.
4. Ikuti petunjuk instalasi yang muncul di layar. Pilih lokasi instalasi dan opsi lain yang diinginkan. Anda juga dapat memilih untuk menginstal driver yang diperlukan untuk menghubungkan Arduino dengan komputer Anda.
5. Setelah proses instalasi selesai, buka Arduino IDE.
6. Selanjutnya, kita perlu memilih papan Arduino yang akan digunakan. Pilih "*Tools*" dari menu atas, kemudian pilih "*Board*" dan pilih papan Arduino yang sesuai.
7. Selanjutnya, kita perlu memilih port serial yang digunakan oleh Arduino. Kembali ke menu "*Tools*", pilih "*Port*", dan pilih *port* yang sesuai.

Setelah menyelesaikan langkah-langkah ini, Arduino IDE siap digunakan untuk membuat dan mengunggah program ke papan Arduino. Kita dapat mulai menulis kode, memverifikasi program, dan mengunggahnya ke papan menggunakan tombol yang tersedia di Arduino IDE.



Gambar 4.1 Tampilan Awal *Software IDE*

## 4.2 Perancangan Rangkaian LED *Cube* 8x8x8



Gambar 4.2 Skema Rangkaian LED *Cube* 8x8x8

Pada Gambar 4.2 terdapat skema rangkaian LED *Cube* 8x8x8 yang menunjukkan bagaimana Arduino Uno mengontrol animasi pada rangkaian tersebut. Arduino Uno bertindak sebagai pengontrol utama yang mengirimkan

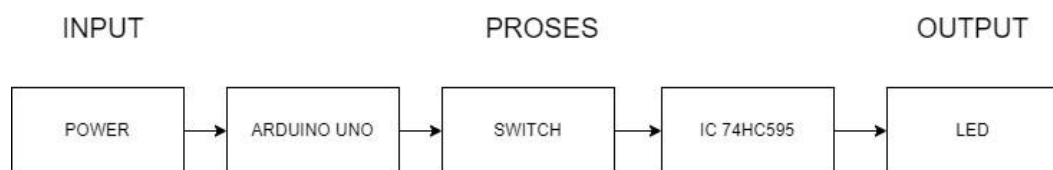
instruksi kepada IC 74HC595. IC 74HC595 pada rangkaian ini berperan sebagai register geser paralel-seri yang digunakan untuk mengatur sumber daya positif ke LED. Setiap LED dalam kubus ini diberi daya melalui resistor yang terhubung dengan jalur keluaran IC 74HC595.

Selain itu, untuk mengatur sumber daya negatif ke LED, digunakan transistor sebagai saklar elektronik yang di kendalikan oleh Arduino Uno. Transistor bertanggung jawab untuk mengatur arus yang mengalir ke kutub negatif LED, sesuai dengan instruksi yang diterima dari Arduino Uno melalui sinyal kontrol.

Dengan konfigurasi ini, Arduino Uno dapat menghasilkan berbagai animasi yang menarik dan kompleks pada LED *Cube* 8x8x8, dengan mengatur secara tepat waktu dan intensitasnya melalui IC 74HC595 dan transistor. Ini memungkinkan pengguna untuk menciptakan tampilan visual yang menarik dan kreatif menggunakan kubus LED ini.

#### 4.2.1 Diagram Blok

LED *cube* dapat dinyalakan dengan daya listrik melalui kabel USB. Kemudian, ada tombol *switch* yang berfungsi untuk memberikan perintah kepada arduino uno yang bertujuan untuk mengirimkan daya ke IC yang berfungsi untuk mengatur tegangan LED sehingga dapat menciptakan ritme atau irama dari program yang dibuat menggunakan bahasa C++.

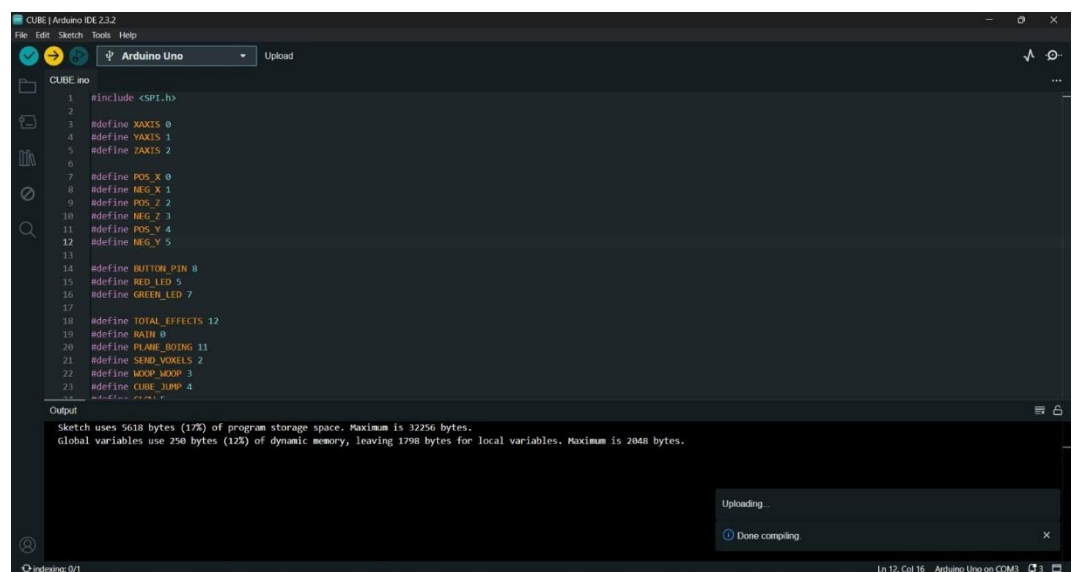


Gambar 4.3 Diagram Blok

## 4.3 Pembangunan Prototipe

### 4.3.1 Penginputan Program

Proses dalam penginputan program dari komputer ke mikrokontroler ini yaitu melalui *Software* Arduino IDE. Hal yang perlu diperhatikan dalam penginputan program kedalam mikrokontroler ini adalah pastikan komputer sudah terdapat *software* Arduino IDE dan muncul jenis board Arduino Uno, kemudian tuliskan program yang akan digunakan pada LED *cube* yang telah di rancang. Setelah program selesai, maka lakukan upload program ke Arduino Uno.

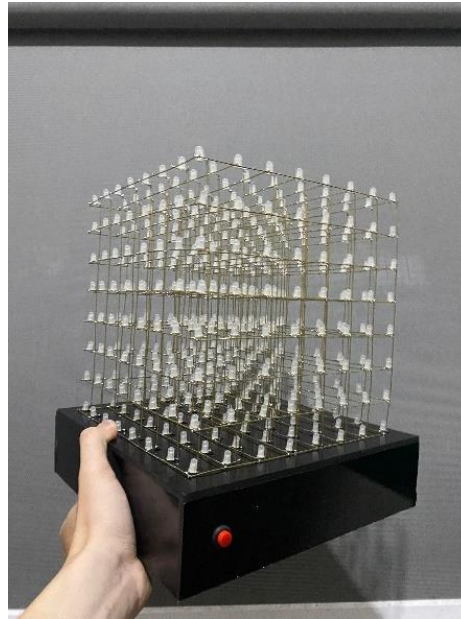


Gambar 4.4 Tampilan Arduino IDE Setelah Terupload

### 4.3.2 Implementasi Rancangan Prototipe LED Cube

Pada gambar 4.7 menampilkan hasil implementasi LED Cube 8x8x8 yang terhubung dengan komponen LED dan mikrokontroler. Struktur kubus LED dengan matriks 8x8 ini dikendalikan oleh mikrokontroler seperti Arduino, memungkinkan animasi yang beragam. Proses implementasi mencakup pemasangan LED, pengaturan resistor, dan integrasi dengan komponen elektronik lain seperti transistor atau IC 74HC595 untuk pengaturan daya. Gambar ini mencerminkan kemajuan dalam teknologi LED untuk aplikasi seni

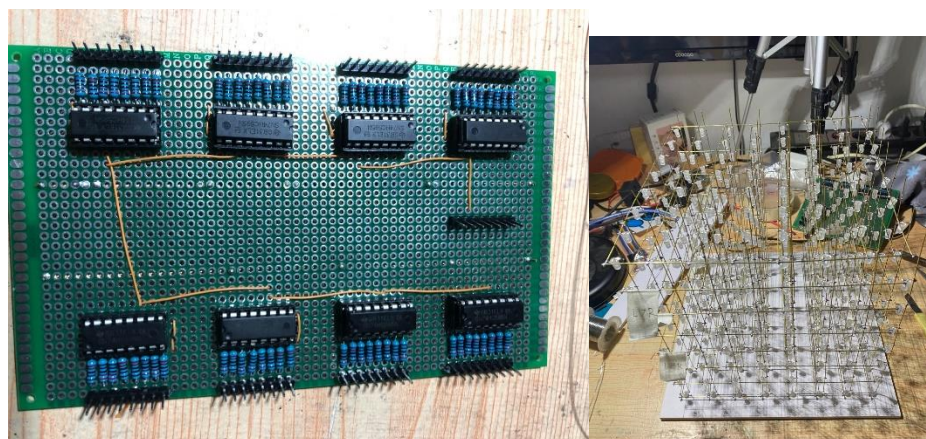
digital dan teknologi interaktif, memperlihatkan potensi penggunaan dalam berbagai konteks kreatif dan teknis.



Gambar 4.5 Rancangan Prototipe LED Cube

#### 4.3.3 Implementasi Rangkaian Elektronika Dengan Arduino Uno

Pada gambar 4.6 ditunjukkan gambar hasil implementasi penggabungan dari rancangan komponen LED dengan mikrokontroler Arduino Uno yang telah terisi kode dan program-program untuk *LED Cube* 8x8x8.



Rangkaian IC Pengatur Kolom

Rangkaian LED 8x8x8

Gambar 4.6 Rancangan Elektronika Dengan Arduino Uno

#### 4.4 Tahap Pengujian Alat

Setelah membuat *hardware* dan *software*, penulis perlu melakukan pengujian dan analisis terhadap alat yang telah dibuat untuk memastikan apakah alat tersebut dapat berfungsi dengan semestinya.

##### 4.4.1 Pengujian Tegangan Animasi LED Cube

Tabel 4.2 Pengujian Tegangan Animasi

No	Bentuk Animasi	Tegang Animasi
1	Via	0,9 V
2	Ibik	0,7 V
3	Send_Voxels	0,7 V
4	Woop_Woop	0,9 V
5	Cube_Jump	1 V
6	Glow	1,1 V
7	Text	0,4 V
8	Lit	1 V
9	Blink	0,4 V
10	Pulsation	1,1 V
11	Display	0,6 V
12	Plane_Boing	0,8 V
13	Rain	0,6 V

Tabel 4.2 di atas menunjukkan hasil pengujian tegangan pada animasi LED *Cube*. Dari tabel tersebut, kita dapat melihat bahwa tegangan terbesar terjadi pada animasi *Glow* dan animasi *Pulsation*, sedangkan tegangan terendah terjadi pada animasi *Text* dan animasi *Blink*.

#### 4.4.2 Durasi Animasi LED Cube

Tabel 4.3 Durasi Animasi LED Cube

No	Bentuk Animasi	Durasi LED	Berjalan		Keterangan
			Ya	Tidak	
1	Via	300 detik	✓		-
2	Ibik	300 detik	✓		-
3	Send_Voxels	140 detik	✓		-
4	Woop_Woop	350 detik	✓		-
5	Cube_Jump	200 detik	✓		-
6	Glow	8 detik	✓		-
7	Text	300 detik	✓		-
8	Lit	500 detik	✓		-
9	Blink	300 detik	✓		-
10	Pulsation	300 detik	✓		-
11	Display	300 detik	✓		-
12	Plane_Boing	220 detik	✓		-
13	Rain	260 detik	✓		-

Tabel 4.3 adalah hasil pengujian durasi animasi pada LED *Cube*, memberikan informasi tentang berapa lama setiap animasi dapat berjalan. Analisis dari tabel ini menunjukkan variasi signifikan dalam durasi animasi, memperlihatkan berbagai macam waktu yang dibutuhkan untuk menyelesaikan setiap animasi. Animasi *Lit* memiliki durasi terpanjang mencapai 500 detik, menunjukkan animasi ini berlangsung untuk waktu yang cukup lama. Sementara itu, *Woop\_Woop* memiliki durasi 350 detik, menegaskan animasi ini memerlukan waktu yang cukup lama untuk dipertimbangkan dalam pengaturan dan penggunaan daya. Di sisi lain spektrum, animasi *Glow* memiliki durasi paling singkat, hanya 8 detik, sementara *Send\_Voxel* durasinya sekitar 140 detik. Perbedaan durasi seperti ini memberikan pandangan mendalam tentang berbagai animasi dalam kubus LED, memungkinkan pengguna memilih sesuai preferensi visual atau kebutuhan proyek mereka.



## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Berdasarkan penelitian dan penerapan yang dilakukan dalam skripsi yang berjudul "Merancang Animasi Menggunakan LED Cube 8x8x8 Berbasis Arduino Uno", beberapa kesimpulan dapat disimpulkan sebagai berikut:

1. Penelitian mengidentifikasi berbagai komponen *hardware* seperti Arduino Uno, LED RGB, *push button*, dan komponen lainnya yang dibutuhkan untuk merancang animasi LED *Cube*. Selain itu, perangkat lunak yang diperlukan adalah Arduino IDE dan sistem operasi Microsoft Windows 11.
2. Pengujian terhadap tegangan dan durasi animasi LED *Cube* menunjukkan bahwa alat ini berfungsi dengan baik. Tegangan terbesar dicapai pada animasi "*Glow*" dan "*Pulsation*," sementara tegangan terendah pada animasi "*Text*" dan "*Blink*." Semua bentuk animasi berhasil berjalan sesuai dengan durasi yang telah direncanakan.
3. Ketika daya terhubung, alat akan segera beroperasi menjalankan animasi tertentu, dan perpindahan antar animasi dilakukan secara berurutan dengan satu tombol.
4. Animasi yang dirancang dapat dijalankan dengan lancar sesuai dengan perangkat lunak yang diimplementasikan menggunakan Arduino IDE.

#### 5.2 Saran

Untuk menyempurnakan lebih lanjut, perlu ditambahkan beberapa saran, seperti :

1. Mengembangkan LED *Cube* dengan ukuran yang lebih besar, seperti 16x16x16 untuk menciptakan animasi yang lebih kompleks dan menarik.
2. Menambahkan fitur interaktif seperti sensor gerak, sensor suara atau konektivitas *bluetooth* atau Wi-Fi untuk memungkinkan kontrol animasi secara *real-time* melalui perangkat lain seperti *smartphone* atau komputer..

3. Penambahan fitur untuk pergantian animasi secara otomatis.
4. Pertimbangkan untuk menggunakan komponen hardware dengan kualitas yang lebih tinggi untuk meningkatkan keandalan dan umur panjang sistem. Misalnya, menggunakan LED dengan intensitas cahaya yang lebih tinggi atau mikrokontroler dengan kemampuan pemrosesan yang lebih baik.

## DAFTAR PUSTAKA

- Abd Wahid, A. A., Abdussamad, S., & Nasibu, I. Z. (2020). Rancang bangun running text pada dot matrix 16x160 berbasis Arduino Uno dengan update data system menggunakan perangkat Android via Bluetooth. *Jambura Journal of Electrical and Electronics Engineering*, 2(1), 8-13.
- Dame, K., & Polly, V. (2018). Rancang bangun LED cube 2x2x2 berbasis Arduino Uno sebagai bahan ajar kreativitas anak sekolah. *Jurnal Ilmiah Realtech*, 14(2), 185-189.
- Djiryanul Fadlie, H., Prasandi Iqro, H., & Romadon, J. (2019). Rancang Bangun LED Cube 8x8x8. Perpustakaan Politeknik Harapan Bersama Tegal.
- Fadli, A., Mardiana, M., & Rismayanti, M. (2023). Rancang bangun text dan animasi 3 dimensi persistence of vision pada LED cube berbasis Arduino Uno. Medan: Program Studi Teknik Informatika Sekolah Tinggi Teknik Harapan Medan.
- Helma, W., Alam, H., & Syafrawali, J. W. (2020). Rancang bangun running text LED display jadwal waktu sholat berbasis Arduino Uno sebagai media informasi. *Journal of Electrical Technology*, 5(2), 2502-3624.
- Herlin, A. (2015, Mei 27). LED Cube 5x5x5. [Internet]. Diakses pada 28 Agustus 2023, dari <https://www.youtube.com/watch?v=oJF6agYrVMY>.
- Iqbal, M., Nandika, R., & Susanti, E. (2019). Rancang bangun text dan animasi 3 dimensi pada LED cube berbasis Arduino Uno Atmega 328. *Jurnal Sigma Teknika*, 2(2), 158-172.
- Sahidah, L. (2022). Unified Modelling Language (UML) untuk perancangan sistem informasi penerimaan siswa baru di SMK Marga Insan Kamil. *Jurnal Sistem Informasi, J-SIKA*, 4(1).

**LAMPIRAN - LAMPIRAN**

## Lampiran 1 Gambar Beberapa Animasi

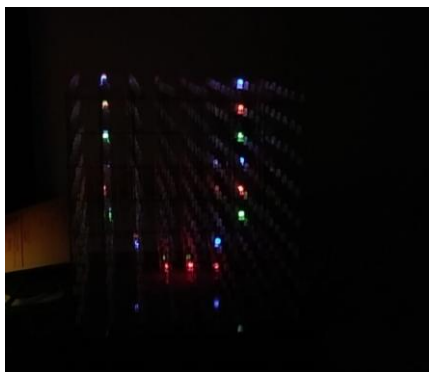
- Animasi Rain

Animasi ini berbentuk seperti air hujan.



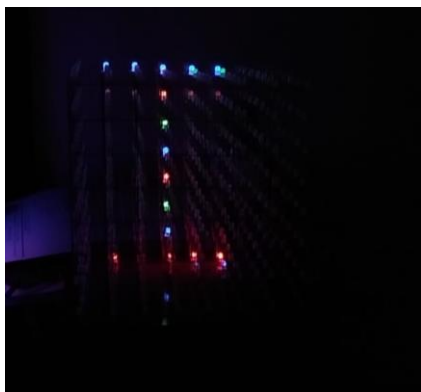
- Animasi VIA

Animasi ini berbentuk huruf yaitu V I A



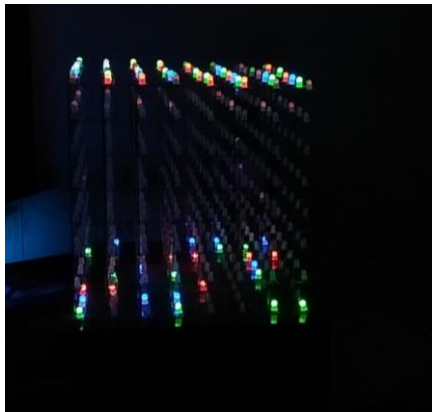
- Animasi IBIK

Animasi ini berbentuk huruf yaitu I B I K



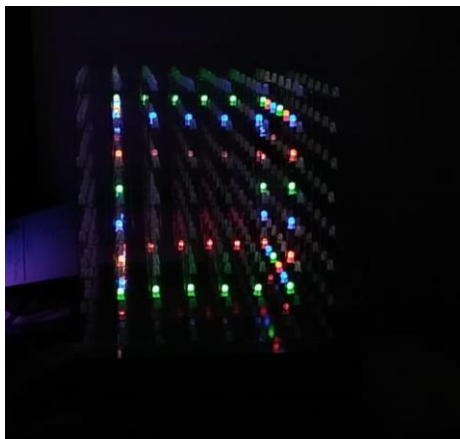
- Animasi Send\_Voxels

Animasi ini berbentuk seperti titik-titik yang bergerak dari atas ke bawah.



- Animasi Whoop\_Whoop

Animasi ini berbentuk seperti berbentuk kubus yang membesar dan mengecil.



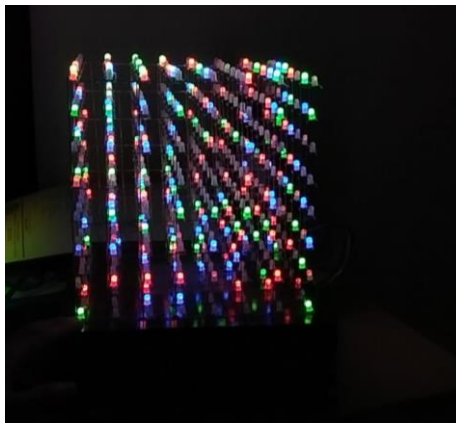
- Animasi Cube\_Jump

Animasi ini berbentuk kubus yang berpindah dari satu sudut ke sudut yang lain.



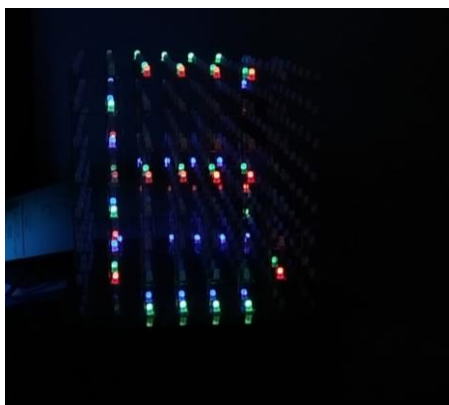
- Animasi Glow

Animasi ini menyalakan semua lampu LED yang ada secara berurutan dan mematikannya secara berurutan.



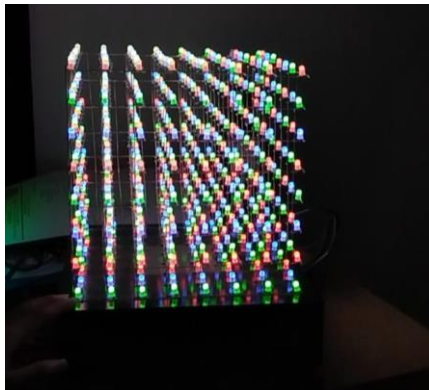
- Animasi Text atau Number

Animasi ini berbentuk angka yaitu 0 sampai 9.



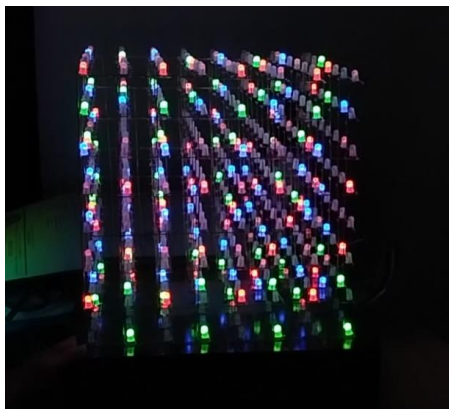
- Animasi Lit

Animasi ini menyalakan semua lampu LED secara bersamaan.



- Animasi Blink

Animasi ini menyalakan dan mematikan lampu LED secara acak.



- Animasi Pulsation

Animasi ini menyalakan semua lampu LED yang ada secara bersamaan dan mematikannya secara bersamaan.





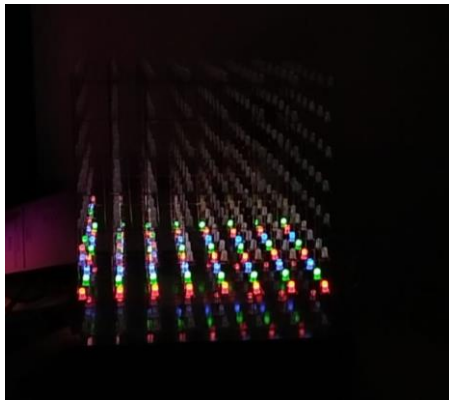
- Animasi Display

Animasi ini menyalakan lampu LED dengan pola tertentu.



- Animasi Plane\_Boing

Animasi ini seperti bidang datar yang bergerak secara vertikal dan horizontal.



## Lampiran 2 Source Code

```
#include <SPI.h>

#define XAXIS 0
#define YAXIS 1
#define ZAXIS 2

#define POS_X 0
#define NEG_X 1
#define POS_Z 2
#define NEG_Z 3
#define POS_Y 4
#define NEG_Y 5

#define BUTTON_PIN 8
#define RED_LED 5
#define GREEN_LED 7

#define TOTAL_EFFECTS 13
#define RAIN 12
#define PLANE_BOING 11
#define SEND_VOXELS 2
#define WOOP_WOOP 3
#define CUBE_JUMP 4

#define GLOW 5
#define TEXT 6
#define LIT 7
#define BLINK 8
#define PULSATION 9
#define DISPLAY 10
#define IBIK 1
#define VIA 0

#define RAIN_TIME 260
#define PLANE_BOING_TIME 220
#define SEND_VOXELS_TIME 140
#define WOOP_WOOP_TIME 350
#define CUBE_JUMP_TIME 200
#define GLOW_TIME 8
#define TEXT_TIME 300
#define CLOCK_TIME 500
#define BLINK_TIME 300
#define PULSATION_TIME 300
#define DISPLAY_TIME 300
#define IBIK_TIME 300
#define VIA_TIME 300
```

```

uint8_t characters[10][8] = {

    {0x3C, 0x42, 0x42, 0x42, 0x42,
0x42, 0x42, 0x3C}, //0

    {0x3C, 0x10, 0x10, 0x10, 0x10,
0x14, 0x18, 0x10}, //1

    {0x7E, 0x02, 0x02, 0x3C, 0x40,
0x40, 0x42, 0x3C}, //2

    {0x3C, 0x42, 0x40, 0x40, 0x3C,
0x40, 0x40, 0x3C}, //3

    {0x20, 0x20, 0x20, 0x7E, 0x22,
0x22, 0x22, 0x22}, //4

    {0x3C, 0x42, 0x40, 0x40, 0x3E,
0x02, 0x02, 0x7E}, //5

    {0x3C, 0x42, 0x42, 0x42, 0x3E,
0x02, 0x02, 0x3C}, //6

    {0x40, 0x40, 0x40, 0x40, 0x40,
0x40, 0x40, 0x3C}, //7

    {0x3C, 0x42, 0x42, 0x42, 0x3C,
0x42, 0x42, 0x3C}, //8

    {0x3C, 0x40, 0x40, 0x3C, 0x42,
0x42, 0x42, 0x3C}, //9};

uint8_t charibik[4][8] = {

    {0x7C, 0x10, 0x10, 0x10, 0x10,
0x10, 0x10, 0x7C}, //I

    {0x3C, 0x40, 0x44, 0x44, 0x3C,
0x44, 0x44, 0x3C}, //B

    {0x7C, 0x10, 0x10, 0x10, 0x10,
0x10, 0x10, 0x7C}, //I

    {0x44, 0x24, 0x14, 0xC, 0xC, 0x14,
0x24, 0x44}, //K};

uint8_t charVIA[4][8] = {

    {0x18, 0x24, 0x42, 0x42, 0x42,
0x42, 0x42, 0x42}, //V

    {0x7C, 0x10, 0x10, 0x10, 0x10,
0x10, 0x10, 0x7C}, //I

    {0x42, 0x42, 0x42, 0x3E, 0x42,
0x42, 0x24, 0x18}, //A};

uint8_t cube[8][8];

uint8_t currentEffect;

uint16_t timer;

bool autoRotate = true;

uint64_t lastEffectChange = 0;

uint32_t effectDuration = 7000;

uint64_t randomTimer;

```

```

bool loading;

void setup() {
    loading = true;

    randomTimer = 0;

    currentEffect = RAIN;

    SPI.begin();

    SPI.beginTransaction(SPISettings(80
00000, MSBFIRST, SPI_MODE0));

    pinMode(BUTTON_PIN,
INPUT_PULLUP);

    pinMode(RED_LED, OUTPUT);

    pinMode(GREEN_LED, OUTPUT);

    randomSeed(analogRead(0));

    digitalWrite(GREEN_LED,
HIGH);}

void loop() {

    uint64_t currentTime = millis();

    randomTimer++;

    if (digitalRead(BUTTON_PIN) ==
LOW || (currentTime -

lastEffectChange >= effectDuration
&& autoRotate)) {

        lastEffectChange = millis();

        clearCube();

        loading = true;

        timer = 0;

        currentEffect++;

        if (currentEffect ==
TOTAL_EFFECTS) {

            currentEffect = 0;}

        randomSeed(randomTimer);

        randomTimer = 0;

        digitalWrite(RED_LED, HIGH);

        digitalWrite(GREEN_LED, LOW);

        delay(500);

        digitalWrite(RED_LED, LOW);

        digitalWrite(GREEN_LED,
HIGH);}

    switch (currentEffect) {

        case RAIN: rain(); break;

        case PLANE_BOING:
planeBoing(); break;

        case SEND_VOXELS:
sendVoxels(); break;

```

```

    case WOOP_WOOP:
woopWoop(); break;

    case CUBE_JUMP: cubeJump();
break;

    case GLOW: glow(); break;

    case TEXT: text("0123456789",
10); break;

    case LIT: lit(); break;

    case BLINK: blink(); break;

    case PULSATION: pulsation();
break;

    case DISPLAY: display(); break;

    case IBIK: ibik("0123", 4); break;

    case VIA: via("012", 3); break;

default: rain();}

renderCube();}

```

```

void renderCube() {

    for (uint8_t i = 0; i < 8; i++) {

        digitalWrite(SS, LOW);

        SPI.transfer(0x01 << i);

        for (uint8_t j = 0; j < 8; j++) {

            SPI.transfer(cube[i][j]);}

        digitalWrite(SS, HIGH); }}

```

```

void rain() {

    if (loading) {

        clearCube();

        loading = false;

    }

    timer++;

    if (timer > RAIN_TIME) {

        timer = 0;

        shift(NEG_Y);

        uint8_t numDrops = random(0, 5);

        for (uint8_t i = 0; i < numDrops;
i++) {

            setVoxel(random(0, 8), 7,
random(0, 8));}}

    uint8_t planePosition = 0;

    uint8_t planeDirection = 0;

    bool looped = false;

    void planeBoing() {

        if (loading) {

            clearCube();

            uint8_t axis = random(0, 3);

            planePosition = random(0, 2) * 7;

```

```

setPlane(axis, planePosition);

if (axis == XAXIS) {
    if (planePosition == 0) {
        planeDirection = POS_X;
    } else {
        planeDirection = NEG_X;
    }
} else if (axis == YAXIS) {
    if (planePosition == 0) {
        planeDirection = POS_Y;
    } else {
        planeDirection = NEG_Y;}}
else if (axis == ZAXIS) {
    if (planePosition == 0) {
        planeDirection = POS_Z;
    } else {
        planeDirection = NEG_Z;
    }
}

timer = 0;

looped = false;

loading = false;
}

timer++;

if (timer > PLANE_BOING_TIME)
{
    timer = 0;

    shift(planeDirection);

    if (planeDirection % 2 == 0) {
        planePosition++;

        if (planePosition == 7) {
            if (looped) {
                loading = true;
            } else {
                planeDirection++;

                looped = true;
            }
        }
    } else {
        planePosition--;

        if (planePosition == 0) {
            if (looped) {
                loading = true;
            } else {
                planeDirection--;
            }
        }
    }
}

```

```

        looped = true;}}}}

uint8_t selX = 0;

uint8_t selY = 0;

uint8_t selZ = 0;

uint8_t sendDirection = 0;

bool sending = false;

void sendVoxels() {
    if (loading) {
        clearCube();

        for (uint8_t x = 0; x < 8; x++) {
            for (uint8_t z = 0; z < 8; z++) {
                setVoxel(x, random(0, 2) * 7, z);
            }
        }

        loading = false;
    }

    timer++;

    if (timer > SEND_VOXELS_TIME)
    {
        timer = 0;

        if (!sending) {
            selX = random(0, 8);

            selZ = random(0, 8);

            if (getVoxel(selX, 0, selZ)) {
                selY = 0;

                sendDirection = POS_Y;
            } else if (getVoxel(selX, 7, selZ))
            {
                selY = 7;

                sendDirection = NEG_Y;
            }

            sending = true;
        } else {
            if (sendDirection == POS_Y) {
                selY++;

                setVoxel(selX, selY, selZ);

                clearVoxel(selX, selY - 1, selZ);

                if (selY == 7) {
                    sending = false;
                }
            } else {
                selY--;

                setVoxel(selX, selY, selZ);

                clearVoxel(selX, selY + 1, selZ);

                if (selY == 0) {

```

```

        sending = false;}}}}}}

uint8_t cubeSize = 0;

bool cubeExpanding = true;

void woopWoop() {
    if (loading) {
        clearCube();

        cubeSize = 2;

        cubeExpanding = true;

        loading = false;
    }

    timer++;

    if (timer > WOOP_WOOP_TIME) {
        timer = 0;

        if (cubeExpanding) {
            cubeSize += 2;

            if (cubeSize == 8) {
                cubeExpanding = false;
            }
        } else {
            cubeSize -= 2;

            if (cubeSize == 2) {
                cubeExpanding = true;
            }
        }

        clearCube();

        drawCube(4 - cubeSize / 2, 4 -
        cubeSize / 2, 4 - cubeSize / 2,
        cubeSize);
    }

    uint8_t xPos;

    uint8_t yPos;

    uint8_t zPos;

    void cubeJump() {
        if (loading) {
            clearCube();

            xPos = random(0, 2) * 7;

            yPos = random(0, 2) * 7;

            zPos = random(0, 2) * 7;

            cubeSize = 8;

            cubeExpanding = false;

            loading = false;
        }
    }
}

```



```

    }

    timer++;

    if (timer > CUBE_JUMP_TIME) {

        timer = 0;

        clearCube();

        if (xPos == 0 && yPos == 0 &&
            zPos == 0) {

            drawCube(xPos, yPos, zPos,
                cubeSize);

            } else if (xPos == 7 && yPos == 7
                && zPos == 7) {

                drawCube(xPos + 1 - cubeSize,
                    yPos + 1 - cubeSize, zPos + 1 -
                    cubeSize, cubeSize);

            } else if (xPos == 7 && yPos == 0
                && zPos == 0) {

                drawCube(xPos + 1 - cubeSize,
                    yPos, zPos, cubeSize);

            } else if (xPos == 0 && yPos == 7
                && zPos == 0) {

                drawCube(xPos, yPos + 1 -
                    cubeSize, zPos, cubeSize);

            } else if (xPos == 0 && yPos == 0
                && zPos == 7) {

                drawCube(xPos, yPos, zPos + 1 -
                    cubeSize, cubeSize);

            }

        if (cubeExpanding) {

            cubeSize++;

            if (cubeSize == 8) {

                cubeExpanding = false;

                xPos = random(0, 2) * 7;

                yPos = random(0, 2) * 7;

                zPos = random(0, 2) * 7;

            }

        } else {

            cubeSize--;

```

```

    if (cubeSize == 1) {
        cubeExpanding = true;
    }
}

bool glowing;
uint16_t glowCount = 0;

void glow() {
    if (loading) {
        clearCube();
        glowCount = 0;
        glowing = true;
        loading = false;
    }

    timer++;
    if (timer > GLOW_TIME) {
        timer = 0;
        if (glowing) {
            if (glowCount < 448) {
                do {
                    selX = random(0, 8);
                    selY = random(0, 8);
                    selZ = random(0, 8);
                } while (getVoxel(selX, selY, selZ));

                setVoxel(selX, selY, selZ);
                glowCount++;
            } else if (glowCount < 512) {
                lightCube();
                glowCount++;
            } else {
                glowing = false;
                glowCount = 0;
            }
        } else {
            if (glowCount < 448) {
                do {
                    selX = random(0, 8);
                    selY = random(0, 8);
                    selZ = random(0, 8);
                } while (!getVoxel(selX, selY, selZ));

                clearVoxel(selX, selY, selZ);
            }
        }
    }
}

```

```

        glowCount++;

    } else {

        clearCube();

        glowing = true;

        glowCount = 0;}}}}

uint8_t charCounter = 0;

uint8_t charPosition = 0;

void text(char string[], uint8_t len) {

    if (loading) {

        clearCube();

        charPosition = -1;

        charCounter = 0;

        loading = false;

    }

    timer++;

    if (timer > TEXT_TIME) {

        timer = 0;

        shift(NEG_Z);

        charPosition++;

        if (charPosition == 7) {

            charCounter++;

            if (charCounter > len - 1) {

                charCounter = 0;

            }

            charPosition = 0;

        }

        if (charPosition == 0) {

            for (uint8_t i = 0; i < 8; i++) {

                cube[i][0] =

                characters[string[charCounter] -

                '0'][i];}}}}

            uint8_t charCounters = 0;

            uint8_t charPositions = 0;

        }

        void ibik(char string[], uint8_t len) {

            if (loading) {

                clearCube();

                charPositions = -1;

                charCounters = 0;

                loading = false;

            }

            timer++;

```

```

if (timer > IBIK_TIME) {

    timer = 0;

    shift(NEG_Z);

    charPositions++;

    if (charPositions == 7) {

        charCounters++;

        if (charCounters > len - 1) {

            charCounters = 0;

        }

        charPositions = 0;

    }

    if (charPositions == 0) {

        for (uint8_t i = 0; i < 8; i++) {

            cube[i][0] =
charibik[string[charCounters] -
'0'][i];}}}

uint8_t charCounterss = 0;

uint8_t charPositionss = 0;

void via(char string[], uint8_t len) {

    if (loading) {

        clearCube();

        charPositionss = -1;

        charCounterss = 0;

        loading = false;

    }

    timer++;

    if (timer > VIA_TIME) {

        timer = 0;

        shift(NEG_Z);

        charPositionss++;

        if (charPositionss == 7) {

            charCounterss++;

            if (charCounterss > len - 1) {

                charCounterss = 0;

            }

            charPositionss = 0;

        }

        if (charPositionss == 0) {

            for (uint8_t i = 0; i < 8; i++) {

```

```

        cube[i][0] =
charVIA[string[charCounterss] -
'0'][i];}}}}

void blink() {
    if (loading) {
        clearCube();

        loading = false;
        timer = 0;
    }

    timer++;

    if (timer > BLINK_TIME) {
        timer = 0;

        // Simulate fire effect
        for (int y = 0; y < 8; y++) {
            for (int x = 0; x < 8; x++) {
                cube[x][y] = random(0, 3) * 85;
            }
        }
    }

    void pulsation() {
        static uint32_t lastPulsationTime =
0;

```

```

static bool lightState = false;

    if (loading) {
        clearCube();

        lastPulsationTime = millis();

        lightState = false;
        loading = false;}

        if (millis() - lastPulsationTime >
PULSATION_TIME) {
            lastPulsationTime = millis();

            lightState = !lightState;
            if (lightState) {
                lightCube();
            } else {
                clearCube();}}}}

char *textToDisplay = "HEL";

void display() {
    static uint8_t charIndex = 0;

    static uint8_t charPosition = 0;

    if (loading) {
        clearCube();

```

```

charIndex = 0;

charPosition = 0;

loading = false;

timer = 0;}

timer++;

if (timer > DISPLAY_TIME) {

    timer = 0;

    shift(POS_Y);

    charPosition++;

    if (charPosition > 7) {

        charPosition = 0;

        charIndex++;

        if (charIndex >=
strlen(textToDisplay)) {

            charIndex = 0;

        }

    }

    // Display character at current
position

    for (uint8_t i = 0; i < 8; i++) {

        cube[i][charPosition] =
characters[textToDisplay[charIndex] -
'0'][i];}}

void lit() {

    if (loading) {

        clearCube();

        for(uint8_t i=0; i<8; i++) {

            for(uint8_t j=0; j<8; j++) {

                cube[i][j] = 0xFF;}}

        loading = false;}}

void setVoxel(uint8_t x, uint8_t y,
uint8_t z) {

    cube[7 - y][7 - z] |= (0x01 << x);}

void clearVoxel(uint8_t x, uint8_t y,
uint8_t z) {

    cube[7 - y][7 - z] ^= (0x01 << x);}

bool getVoxel(uint8_t x, uint8_t y,
uint8_t z) {

    return (cube[7 - y][7 - z] & (0x01 <<
x)) == (0x01 << x);}

void setPlane(uint8_t axis, uint8_t i) {

    for (uint8_t j = 0; j < 8; j++) {

        for (uint8_t k = 0; k < 8; k++) {

```

```

    if (axis == XAXIS) {
        setVoxel(i, j, k);
    } else if (axis == YAXIS) {
        setVoxel(j, i, k);
    } else if (axis == ZAXIS) {
        setVoxel(j, k, i);}}}

void shift(uint8_t dir) {
    if (dir == POS_X) {
        for (uint8_t y = 0; y < 8; y++) {
            for (uint8_t z = 0; z < 8; z++) {
                cube[y][z] = cube[y][z] << 1;
            }
        }
    } else if (dir == NEG_X) {
        for (uint8_t y = 0; y < 8; y++) {
            for (uint8_t z = 0; z < 8; z++) {
                cube[y][z] = cube[y][z] >> 1;}}
    } else if (dir == POS_Y) {
        for (uint8_t y = 1; y < 8; y++) {
            for (uint8_t z = 0; z < 8; z++) {
                cube[y - 1][z] = cube[y][z];}}
        for (uint8_t i = 0; i < 8; i++) {
            cube[7][i] = 0;
        }
    } else if (dir == NEG_Y) {
        for (uint8_t y = 7; y > 0; y--) {
            for (uint8_t z = 0; z < 8; z++) {
                cube[y][z] = cube[y - 1][z]; }}
        for (uint8_t i = 0; i < 8; i++) {
            cube[0][i] = 0;
        }
    } else if (dir == POS_Z) {
        for (uint8_t y = 0; y < 8; y++) {
            for (uint8_t z = 1; z < 8; z++) {
                cube[y][z - 1] = cube[y][z];}}
        for (uint8_t i = 0; i < 8; i++) {
            cube[i][7] = 0;
        }
    } else if (dir == NEG_Z) {
        for (uint8_t y = 0; y < 8; y++) {
            for (uint8_t z = 7; z > 0; z--) {
                cube[y][z] = cube[y][z - 1];}}
        for (uint8_t i = 0; i < 8; i++) {
            cube[i][0] = 0;}}}

```

```

void drawCube(uint8_t x, uint8_t y,          cube[i][j] = 0;}}
uint8_t z, uint8_t s) {

    for (uint8_t i = 0; i < s; i++) {

        setVoxel(x, y + i, z);

        setVoxel(x + i, y, z);

        setVoxel(x, y, z + i);

        setVoxel(x + s - 1, y + i, z + s - 1);

        setVoxel(x + i, y + s - 1, z + s - 1);

        setVoxel(x + s - 1, y + s - 1, z + i);

        setVoxel(x + s - 1, y + i, z);

        setVoxel(x, y + i, z + s - 1);

        setVoxel(x + i, y + s - 1, z);

        setVoxel(x + i, y, z + s - 1);

        setVoxel(x + s - 1, y, z + i);

        setVoxel(x, y + s - 1, z + i);}}

```

```

void lightCube() {

    for (uint8_t i = 0; i < 8; i++) {

        for (uint8_t j = 0; j < 8; j++) {

            cube[i][j] = 0xFF;}}}

```

```

void clearCube() {

    for (uint8_t i = 0; i < 8; i++) {

        for (uint8_t j = 0; j < 8; j++) {

```



