

**UNIVERSITAS GUNADARMA**  
**FAKULTAS TEKNOLOGI INDUSTRI**



**TULISAN ILMIAH**

**APLIKASI PENGENDALI LAMPU BERBASIS ANDROID  
DENGAN MENGGUNAKAN ARDUINO UNO, MODUL  
BLUETOOTH DAN MODUL DIMMER**

**Nama** : Hafidz Arupadatu  
**NPM** : 52417609  
**Jurusan** : Teknik Informatika  
**Pembimbing** : Dini Triasanti, ST., MMSI

**Diajukan Guna Melengkapi Sebagian Syarat Dalam Mencapai  
Gelar Setara Sarjana Muda**

**Jakarta**

**2021**

## **PERNYATAAN ORIGINALITAS DAN PUBLIKASI**

Saya yang bertanda tangan dibawah ini :

Nama : Hafidz Arupadatu  
NPM : 52417609  
Judul PI : Aplikasi Pengedali Lampu Berbasis Android  
dengan Menggunakan Arduino Uno, Modul  
Bluetooth dan Modul Dimmer  
Tanggal Sidang :  
Tanggal Lulus :

Dengan ini menyatakan bahwa hasil penelitian yang telah saya buat merupakan hasil karya saya sendiri dan benar-benar keasliannya serta dapat dipublikasikan sepenuhnya oleh Universitas Gunadarma. Segala dalam bentuk apapun telah saya mengikuti kaidah yang berlaku.

Apabila kemudian hari penulisan ilmiah ini merupakan hasil plagiat atau penjiplakan terhadap karya orang lain maka saya bersedia mempertanggung jawabkan sekaligus bersedia menerima sanksi berdasarkan aturan dan tata tertib Universitas Gunadarma.

Demikian pernyataan ini saya buat dengan seadanya dalam keadaan sadar dan tidak dipaksakan.

Jakarta, Februari 2021

(Hafidz Arupadatu)

## **LEMBAR PENGESAHAN**

Judul Penulisan Ilmiah : APLIKASI PENGENDALI LAMPU BERBASIS  
ANDROID DENGAN MENGGUNAKAN  
ARDUINO UNO, MODUL BLUETOOTH DAN  
MODUL DIMMER

Nama : HAFIDZ ARUPADATU

NPM : 52417609

Tanggal Sidang :

Tanggal Lulus :

Menyetujui

**Pembimbing**

**Koordinator PI**

(Dini Triasanti, ST., MMSI)

(Dr. Achmad Fahrurozi, S.SI., MSI)

**Ketua Jurusan Teknik Informatika**

(Dr. Lintang Yuniar Banowosari, S.Kom., M.Sc.)

## ABSTRAKSI

HAFIDZ ARUPADATU, 52417609

### **APLIKASI PENGENDALI LAMPU BERBASIS ANDROID DENGAN MENGGUNAKAN ARDUINO UNO, MODUL BLUETOOTH DAN MODUL DIMMER.**

PI. Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Gunadarma, 2021.

Kata Kunci : Aplikasi, Android, Bluetooth, Arduino

**(x + 58 + Lampiran)**

Pengguna saat ini menyalakan lampu dengan cara berjalan mendekati saklar. Hal ini membuat penulis menyadari dengan memanfaatkan perangkat *smartphone*, *Arduino Uno*, Modul *Bluetooth* dan Modul *dimmer* dapat menggantikan fungsi saklar. Pembuatan alat ini menggunakan *software Android Studio* untuk merancang aplikasi *smartphone* dan *Arduino IDE* sebagai mengkompilasi kode – kode pada *Arduino Uno*. Dalam perancangan alat ini, *Arduino Uno* digunakan sebagai alat pemrosesan data. Lalu, modul *Bluetooth* HC-05 sebagai media penghubung antara *smartphone* dan mikrokontroler. Kemudian, modul *dimmer* sebagai pengatur arus listrik yang akan diberikan oleh lampu. sedangkan aplikasi *smartphone* sebagai media input. Aplikasi ini dibuat menjadi beberapa tahap yaitu, merancang aplikasi di *Android Studio*, pengkodean mikrokontroler di *Arduino IDE* dan uji coba. Berdasarkan hasil uji coba alat ini dapat mematikan, menghidupkan, dan mengatur kecerahan lampu.

Daftar Pustaka (2004 - 2020)

## KATA PENGANTAR

Puji syukur yang sebesar-besarnya penulis panjatkan kehadiran Tuhan Yang Maha Esa, karena atas berkat rahmat dan karunia-Nya penulisan ilmiah ini yang berjudul “Aplikasi Pengendali Lampu Berbasis Android dengan Menggunakan Arduino Uno, Modul Bluetooth dan Modul Dimmer” dapat terselesaikan tepat pada waktunya.

Keberhasilan dalam menyelesaikan penulisan ilmiah ini adalah berkat bantuan dan dukungan dari berbagai pihak serta keteguhan hati penulis, meskipun banyak hambatan yang dihadapi oleh penulis, namun semua menjadi pelajaran dan pengalaman yang berkesan.

Dalam kesempatan ini perkenankan penulis menyampaikan ucapan terima kasih atas segala dukungan dan bantuan yang diterima oleh penulis, sehingga dapat menyelesaikan penulisan ini. Dan perkenankan penulis mengucapkan terima kasih kepada :

1. Ibu Prof. Dr. E. S. Margianti, S.E., MM., selaku Rektor Universitas Gunadarma.
2. Bapak Prof. Dr. -Ing. Adang Suhendra, S.SI., S.Kom., MSc., selaku Dekan Fakultas Teknologi Industri, Universitas Gunadarma.
3. Ibu Dr. Lintang Yuniar Banowosari, S.Kom., M,Sc., selaku Ketua Jurusan Teknik Informatika.
4. Bapak Dr. Achmad Fahrurozi, S.SI., MSI, selaku Kasubag Sidang Penulisan Ilmiah.
5. Ibu Dini Triasanti, ST., MMSI, selaku Dosen Pembimbing yang telah dengan sabar memberikan bimbingan serta masukan sehingga penulis dapat menyelesaikan penulisan ini dengan baik.
6. Orang tua penulis Irwan Tepianto dan Rodliyah yang telah banyak memberikan bantuan baik itu moril maupun materil serta dorongan semangat yang tak ternilai. Dan berkat doa serta restu mereka jugalah terlaksananya penulisan ini.

7. Ardhya Alga Lingga dan teman – teman polterghost yang selalu memberikan dukungan moril dan masukan untuk penulis agar dapat menyelesaikan penulisan ini pada waktunya.

Penulis ingin mengucapkan terima kasih kepada semua pihak yang tidak dapat penulis sebutkan satu persatu, atas bantuan, saran, dan masukan yang telah diberikan kepada penulis.

Oleh karena itu penulis sangat menghargai kritik maupun saran yang berguna bagi kesempurnaan penyusunan penulisan ilmiah ini. Akhir kata penulis berharap semoga penulisan ini dapat bermanfaat bagi para pembaca.

Jakarta, 22 Februari 2021

(Hafidz Arupadatu)

## DAFTAR ISI

|  |            |
|--|------------|
| <b>PERNYATAAN ORIGINALITAS DAN PUBLIKASI .....</b> | <b>ii</b>  |
| <b>LEMBAR PENGESAHAN .....</b>                     | <b>iii</b> |
| <b>ABSTRAKSI .....</b>                             | <b>iv</b>  |
| <b>KATA PENGANTAR.....</b>                         | <b>v</b>   |
| <b>DAFTAR ISI.....</b>                             | <b>vii</b> |
| <b>DAFTAR GAMBAR.....</b>                          | <b>ix</b>  |
| <b>DAFTAR TABEL .....</b>                          | <b>x</b>   |
| <b>BAB I.....</b>                                  | <b>1</b>   |
| <b>PENDAHULUAN .....</b>                           | <b>1</b>   |
| <b>1.1 Latar Belakang.....</b>                     | <b>1</b>   |
| <b>1.2 Batasan Masalah .....</b>                   | <b>2</b>   |
| <b>1.3 Tujuan Penelitian.....</b>                  | <b>2</b>   |
| <b>1.4 Metode Penulisan .....</b>                  | <b>3</b>   |
| <b>1.5 Sistematika Penulisan .....</b>             | <b>4</b>   |
| <b>BAB II .....</b>                                | <b>5</b>   |
| <b>LANDASAN TEORI .....</b>                        | <b>5</b>   |
| <b>2.1 Android .....</b>                           | <b>5</b>   |
| <b>2.2 Android Studio .....</b>                    | <b>6</b>   |
| <b>2.3 Bahasa Pemrograman Java.....</b>            | <b>7</b>   |
| <b>2.4 Struktur Navigasi.....</b>                  | <b>8</b>   |
| <b>2.5 Flowchart .....</b>                         | <b>8</b>   |
| <b>2.6 Arduino Uno .....</b>                       | <b>9</b>   |
| <b>2.7 Arduino IDE .....</b>                       | <b>12</b>  |
| <b>2.8 Bahasa Pemrograman C.....</b>               | <b>13</b>  |
| <b>2.9 Bluetooth Module HC-05.....</b>             | <b>14</b>  |
| <b>2.10 AC Light Dimmer Module .....</b>           | <b>16</b>  |
| <b>2.11 Lampu Bohlam .....</b>                     | <b>20</b>  |

|   |           |
|---|-----------|
| <b>BAB III.....</b>                                   | <b>22</b> |
| <b>PEMBAHASAN .....</b>                               | <b>22</b> |
| <b>3.1    Arduino .....</b>                           | <b>22</b> |
| <b>3.1.1    Blok Diagram .....</b>                    | <b>22</b> |
| <b>3.1.2    Skematik Rangkaian.....</b>               | <b>23</b> |
| <b>3.1.3    Implementasi Program Arduino Uno.....</b> | <b>24</b> |
| <b>3.2    Android .....</b>                           | <b>27</b> |
| <b>3.2.1    Perancangan Struktur Navigasi .....</b>   | <b>27</b> |
| <b>3.2.2    Perancangan Halaman .....</b>             | <b>27</b> |
| <b>3.3    Flowchart Program.....</b>                  | <b>50</b> |
| <b>3.4    Langkah pembuatan Aplikasi.....</b>         | <b>52</b> |
| <b>3.5    Uji Coba .....</b>                          | <b>53</b> |
| <b>BAB IV .....</b>                                   | <b>55</b> |
| <b>PENUTUP.....</b>                                   | <b>55</b> |
| <b>4.1 Kesimpulan.....</b>                            | <b>55</b> |
| <b>4.2 Saran .....</b>                                | <b>55</b> |
| <b>DAFTAR PUSTAKA .....</b>                           | <b>56</b> |
| <b>LAMPIRAN LISTING PROGRAM.....</b>                  | <b>A</b>  |
| <b>LAMPIRAN OUTPUT PROGRAM.....</b>                   | <b>HH</b> |



## DAFTAR GAMBAR

|  |    |
|--|----|
| <b>Gambar 2.1</b> Logo Android .....   | 5  |
| <b>Gambar 2.2</b> Logo Android Studio .....  | 6  |
| <b>Gambar 2.3</b> Stuktur Navigasi Linier .....  | 8  |
| <b>Gambar 2.4</b> Arduino Uno .....  | 10 |
| <b>Gambar 2.5</b> Perbandingan Versi Arduino Uno .....   | 10 |
| <b>Gambar 2.6</b> Arduino Uno Pinout.....  | 11 |
| <b>Gambar 2.7</b> Tampilan Arduino IDE.....  | 13 |
| <b>Gambar 2.8</b> Bluetooth Module HC-05 .....   | 14 |
| <b>Gambar 2.9</b> Library Bluetooth Module .....   | 15 |
| <b>Gambar 2.10</b> Deklarasi Variabel Bluetooth Module .....   | 15 |
| <b>Gambar 2.11</b> Void Setup Bluetooth Module .....   | 16 |
| <b>Gambar 2.12</b> Void Loop Bluetooth Module.....   | 16 |
| <b>Gambar 2.13</b> AC Light Dimmer Module .....  | 17 |
| <b>Gambar 2.14</b> Instalasi Library Dimmer Module .....   | 19 |
| <b>Gambar 2.15</b> Library Dimmer Module .....   | 19 |
| <b>Gambar 2.16</b> Deklarasi Variabel Dimmer Module .....  | 19 |
| <b>Gambar 2.17</b> Void Setup Dimmer Module .....  | 20 |
| <b>Gambar 2.18</b> Void Loop Dimmer Module .....   | 20 |
| <b>Gambar 2.19</b> Lampu Bohlam .....  | 21 |
| <br>   |    |
| <b>Gambar 3.1</b> Blok Diagram pada Perancangan Alat.....  | 22 |
| <b>Gambar 3.2</b> Skematik Aplikasi Pengendali Lampu Berbasis Android dengan<br>Menggunakan Arduino Uno, Modul Bluetooth dan Modul Dimmer.....                             | 23 |
| <b>Gambar 3.3</b> Perancangan Struktur Navigasi Linier Aplikasi Pengendali Lampu Berbasis<br>Android dengan Menggunakan Arduino Uno, Modul Bluetooth dan Modul Dimmer..... | 27 |
| <b>Gambar 3.4</b> Rancangan Halaman Splash Screen.....   | 28 |
| <b>Gambar 3.5</b> Tampilan Halaman Splash Screen .....   | 30 |
| <b>Gambar 3.6</b> Rancangan Halaman Mulai .....  | 31 |
| <b>Gambar 3.7</b> Tampilan Halaman Mulai dan Tombol GO Saat Ditekan .....  | 35 |
| <b>Gambar 3.8</b> Rancangan Halaman Search Device.....   | 35 |
| <b>Gambar 3.9</b> Tampilan Halaman Search Device dan Tombol Search Saat Ditekan.....   | 40 |
| <b>Gambar 3.10</b> Rancangan Halaman Main Activity.....  | 40 |
| <b>Gambar 3.11</b> Tampilan Halaman Main Activity .....  | 48 |
| <b>Gambar 3.12</b> Flowchart Program .....   | 50 |
| <b>Gambar 3.13</b> Menghubungkan Smartphone dengan Komputer .....  | 52 |
| <b>Gambar 3.14</b> Memilih Perangkat Smartphone .....  | 52 |
| <b>Gambar 3.15</b> Tampilan Build .....  | 52 |
| <b>Gambar 3.16</b> Menghubungkan Arduino Uno dengan Komputer .....   | 53 |
| <b>Gambar 3.17</b> Tombol Upload pada Arduino IDE .....  | 53 |

## DAFTAR TABEL

|   |    |
|---|----|
| <b>Tabel 2.1</b> Simbol Flowchart .....   | 9  |
| <b>Tabel 2.2</b> Spesifikasi Arduino Uno .....  | 12 |
| <b>Tabel 2.3</b> Pin Zero Cross Pada Setiap Mikrokontroler.....   | 17 |
| <b>Tabel 2.4</b> Spesifikasi AC Light Dimmer Module .....   | 18 |
| <b>Tabel 3.1</b> Hubungan Pin Arduino pada Aplikasi Pengendali Lampu Berbasis Android<br>dengan Menggunakan Arduino Uno, Modul Bluetooth dan Modul Dimmer ..... | 24 |
| <b>Tabel 3.2</b> Tabel Hasil Pengujian .....  | 54 |

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Perkembangan teknologi saat ini berkembang dengan pesat, hal ini dapat dirasakan dalam dunia industri maupun masyarakat. Salah satunya adalah pemanfaatan mikrokontroler dalam kehidupan sehari-hari, yaitu *Arduino Uno*. Alat ini telah banyak dimanfaatkan untuk keperluan penelitian, pendidikan, maupun komersial. Menurut data *sparkfun.com* (2015), pengguna *Arduino Uno* menempati urutan pertama daripada mikrokontroler lainnya karena alat ini banyak di pasaran, harganya terjangkau dan sering digunakan untuk penelitian dalam skala besar maupun kecil. Untuk menulis program pada *Arduino Uno* dibutuhkan software *Arduino IDE* yang menggunakan *C Language*.

Pada sisi lain perkembangan *smartphone* banyak tersedia di pusat perbelanjaan dengan harga terjangkau dan menggunakan sistem operasi *Android* bersifat *open source*. Dengan sifat *open source* pengguna dapat membuat atau memodifikasi aplikasi pada *Android* dan biaya pembuatan aplikasi menjadi murah. Menurut data *andi.link* (2019), total penduduk Indonesia per Januari 2019 adalah 268 juta jiwa. Pengguna *smartphone* di Indonesia mencapai 355 juta pengguna, lebih banyak dari pada jumlah penduduk Indonesia. Berdasarkan data *Statcounter.com* (2020) menampilkan bahwa sistem operasi *Android* menempati urutan pertama pada nilai pasar di Indonesia dengan persentase 92,02% pengguna.

Pada era globalisasi ini banyak tercipta inovasi-inovasi baru yang membuat pekerjaan manusia menjadi lebih praktis dan efisien. Salah satunya adalah penelitian yang terkait dengan aplikasi pengendali saklar lampu yang dilakukan oleh Chandan Kumar pada tahun 2019. Penelitian yang dilakukan oleh Chandan Kumar aplikasi pengendalian lampu hanya dapat menyalakan atau mematikan lampu. Hal tersebut membuat penulis tertarik untuk mengembangkan penelitian

tersebut. Pada penelitian ini pengguna tidak hanya menyalakan atau mematikan lampu melalui perangkat *smartphone*, tetapi dapat mengatur tingkat kecerahan lampu sesuai kebutuhan pengguna. Lampu yang digunakan pada penelitian ini adalah lampu yang bersifat *dimmerable* agar dapat diatur tingkat intensitas cahaya contohnya adalah lampu bohlam. Tingkat kecerahan lampu dapat diatur dengan menggunakan modul *AC Light Dimmer* sebagai pengatur arus listrik ke lampu. Selain itu, pengguna tidak perlu berjalan mendekati saklar konvensional untuk menyalakan atau mematikan lampu.

Berdasarkan latar belakang tersebut, penulis termotivasi membuat suatu penulisan ilmiah berjudul **“Aplikasi Pengendali Lampu Berbasis Android dengan Menggunakan Arduino Uno, Modul Bluetooth dan Modul Dimmer”**.

## **1.2 Batasan Masalah**

Batasan masalah bertujuan untuk membatasi pembahasan agar terfokus dan tidak keluar dari topik permasalahan. Adapun batasan masalah tersebut adalah :

1. Lampu yang digunakan pada penelitian ini adalah lampu bersifat *dimmerable* (dapat diatur tingkat kecerahan).
2. Pada penelitian ini penulis hanya menggunakan 1 buah lampu bohlam sebagai alat pengujian.

## **1.3 Tujuan Penelitian**

Adapun tujuan penelitian pada penulisan ini adalah membuat aplikasi lampu pengendali lampu berbasis *android*.

## 1.4 Metode Penulisan

Metode Penulisan Ilmiah yang digunakan adalah :

### 1. Perangkat keras yang dibutuhkan

Penelitian ini membutuhkan perangkat keras dan lunak. Perangkat keras yang dibutuhkan yaitu :

- 1 unit komputer pribadi dengan *AMD Ryzen 5 3600*.
- *NVIDIA GeForce GTX 750 Ti*.
- *Memory RAM 16 GB DDR4 3000 MHz*.
- *SSD 256 GB dan HardDisk 1 TB*.
- 1 unit *Arduino Uno*.
- 1 unit *Bluetooth Module*.
- 1 unit *Dimmer Module*.
- 1 unit lampu bohlam.
- Kabel *USB Android*.

Sedangkan perangkat lunak yang digunakan yaitu:

- *Microsoft Windows 10 Pro 64-bit*.
- *Android Studio*.
- *Arduino IDE*.

### 2. Prosedur pembuatan aplikasi

Pembuatan Aplikasi Pengendali Lampu Berbasis Android dengan Menggunakan Arduino Uno, Modul Bluetooth dan Modul Dimmer penulis telah melalui berbagai tahapan pembuatan aplikasi diantaranya :

- Tahap Pengumpulan Data dan Informasi  
Data Aplikasi ini dikumpulkan melalui internet. Format gambar yang dipakai adalah .png.
- Perancangan Struktur Navigasi  
Struktur navigasi yang dirancang dalam pembuatan aplikasi ini adalah struktur navigasi linier.
- Perancangan Halaman

Perancangan halaman pembuka atau *splash screen*, halaman *search device* dan halaman utama (*main activity*).

- Pengkodean Program  
Naskah program dibuat menggunakan bahasa pemrograman java dan XML.
- Uji Coba  
Penulis melakukan uji coba pada *Android Smartphone Xiaomi Redmi 3S*.

### 1.5 Sistematika Penulisan

Penulisan ilmiah ini diurutkan menjadi 4 bab dan diharapkan pembaca dapat memahami penulisan diantaranya :

#### 1. PENDAHULUAN

Bab ini menjelaskan tentang latar belakang, batasan masalah, tujuan penelitian, metode penelitian, dan sistematika penulisan.

#### 2. LANDASAN TEORI

Bab ini menjelaskan tentang teori dasar dari sistem operasi *Android*, *Arduino Uno*, dan komponen pendukung lainnya.

#### 3. PEMBAHASAN

Bab ini membahas tentang blok diagram, skematik alat, alur program, rancangan aplikasi, dan uji coba.

#### 4. PENUTUP

Bab ini menjelaskan tentang kesimpulan dan saran berdasarkan hasil dari alat dan aplikasi yang sudah dibuat.

## BAB II

### LANDASAN TEORI

#### 2.1 Android

*Android* adalah sebuah sistem operasi untuk perangkat mobile berbasis *Linux* yang mencakup sistem operasi, *middleware*, dan aplikasi. *Android* merupakan platform terbuka (*Open Source*) bagi para pengembang untuk menciptakan aplikasi mereka.

Dengan sifat *open source* pengguna dapat membuat atau mengembangkan aplikasi yang diinginkan, menekan biaya pembuatan aplikasi dan aman untuk digunakan. Namun kekurangan sifat *open source* terdapat banyak *bug*, perlu waktu untuk mempelajari *software* tersebut dan *interface* kurang *user friendly*.

Pada tanggal 05 November 2007, merupakan perilisan perdana *Android*. *Android* bersama *Open Handset Alliance* menyatakan mendukung pengembangan *open source* pada perangkat *mobile*. Di sisi lain, *Google* merilis kode-kode *Android* di bawah lisensi *Apache*, sebuah lisensi perangkat lunak dan *open source* perangkat seluler<sup>[3]</sup>.



**Gambar 2.1** Logo Android

## 2.2 Android Studio

*Android Studio* adalah pengembangan aplikasi *Android* yang bersifat *open source*. *Android Studio* diluncurkan oleh *Google* pada 16 Mei 2013 pada event *Google I/O Conference*. Sejak saat itu, *Android Studio* menggantikan *Eclipse* sebagai IDE (*Integrated Development Environment*) resmi untuk mengembangkan atau membuat aplikasi *Android*.

Setiap proyek di *Android Studio* berisi satu atau beberapa modul dengan file kode sumber dan file sumber daya. Jenis-jenis modul mencakup modul aplikasi *Android*, modul pustaka, dan modul *google app engine*. Secara *default*, *Android Studio* akan menampilkan file proyek dalam tampilan proyek *Android*. Tampilan disusun berdasarkan modul untuk memberikan akses cepat ke file sumber utama proyek. Semua file versi terlihat di bagian bawah *Gradle Scripts* dan masing-masing modul aplikasi berisi *folder* berikut:

1. Manifest : berisi file “AndroidManifest.xml”.
2. Java : Berisi file kode utama untuk aplikasi *Android*.
3. Res : Berisi *resource* untuk aplikasi *Android* seperti tata letak, gambar, dll<sup>[5]</sup>.



**Gambar 2.2** Logo Android Studio



### 2.3 Bahasa Pemrograman Java

*Java* merupakan bahasa pemrograman tingkat tinggi yang berorientasi pada *object* dan program *java* tersusun dari bagian yang disebut dengan *Class*. *Class* terdiri dari metode- metode yang melakukan pekerjaan dan mengembalikan informasi setelah melakukan tugasnya. Kumpulan class di pustaka class *Java* yang disebut dengan *Java Application Programming Interface (API)*. *Class- class* ini diorganisasikan menjadi sekelompok yang disebut dengan paket (*package*). *Java API* telah menciptakan *applet* dan aplikasi canggih dengan menyediakan fungsionalitas yang memadai .

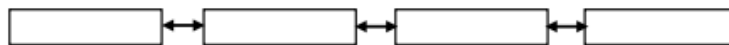
Jadi ada dua hal yang harus dipelajari dalam *Java*, yaitu bagaimana mempergunakan *class* pada *Java API* dan mempelajari bahasa *Java*. Tidak ada cara lain selain *class* yang merupakan satu-satunya cara menyatakan bagian eksekusi program. Pada *Java* program *javac* untuk mengkompilasi file kode sumber *Java* menjadi *class-class bytecode*. File kode sumber mempunyai ekstensi \*.java. Kompilator *javac* menghasilkan file *bytecode class* dengan ekstensi \*.class. Interpreter adalah modul utama pada sistem *Java* yang digunakan aplikasi *Java* dan menjalankan program *bytecode Java*.

Dengan kata lain *Java* adalah bahasa pemrograman yang dapat membuat seluruh bentuk aplikasi tidak hanya *desktop* dan *web* namun juga bisa membuat aplikasi *mobile* dan lainnya, sebagaimana dibuat dengan menggunakan bahasa pemrograman konvensional yang lain. *Java* merupakan bahasa pemrograman yang bersifat umum atau non-spesifik (*general purpose*). Bahasa Pemrograman *Java* berorientasi object (OOP-*Object Oriented Programming*), dan dapat dijalankan pada berbagai platform sistem operasi. Pada OOP, program komputer sebagai kelompok *object* yang saling berinteraksi<sup>[21]</sup>.

## 2.4 Struktur Navigasi

Struktur navigasi adalah alur yang digunakan dalam aplikasi yang dibuat. Sebelumnya menyusun aplikasi multimedia kedalam sebuah *software*, harus ditentukan terlebih dahulu alur apa yang akan digunakan dalam aplikasi yang dibuat. Pada penulisan ini penulis menggunakan struktur navigasi linier.

Struktur navigasi *linier* merupakan struktur yang mempunyai suatu rangkaian cerita berurutan. Struktur ini menampilkan satu demi satu tampilan *layer* secara berurutan menurut aturannya<sup>[6]</sup>.



**Gambar 2.3** Stuktur Navigasi Linier



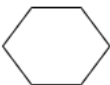
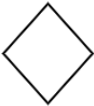

## 2.5 Flowchart

Pengertian Flowchart adalah urutan proses kegiatan yang digambarkan dalam bentuk simbol. Flowchart (bagan alir) juga didefinisikan sebagai diagram yang menyatakan aliran proses dengan menggunakan anotasi- anotasi semisal persegi, panah, oval, wajik dll.

Penggunaan flowchart sangat penting, Bahkan Dr. Kauro Ishikawa seorang ahli teori organisasi, menjadikannya sebagai 1 dari 7 alat kualitas dasar (7 basic quality tools) yang harus dikuasai para anggota Quality Control Circle atau gugus kendali kualitas.

Melalui flowchart, kita bisa melihat langkah-langkah proses secara mendetail, lengkap dengan aktivitas yang terjadi. Flowchart banyak dipergunakan sebagai alat komunikasi dan dokumentasi. Berikut ini adalah tabel simbol flowchart yang digunakan pada penelitian ini<sup>[20]</sup>.

**Tabel 2.1** Simbol Flowchart

| No | Gambar Simbol  | Nama Simbol    | Kegunaan   |
|----|--|----------------|--|
| 1. |   | Flow Direction | simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain (connecting line). |
| 2. |   | Terminal Point | menunjukkan permulaan (start) atau akhir (stop) dari suatu proses.   |
| 3. |   | Preparation    | merupakan simbol yang digunakan untuk mempersiapkan penyimpanan di dalam storage.                            |
| 4. |   | Decision       | merupakan simbol yang digunakan untuk memilih proses atau keputusan berdasarkan kondisi yang ada.            |
| 5. |  | Processing     | digunakan untuk menunjukkan kegiatan yang dilakukan oleh komputer.   |

## 2.6 Arduino Uno

*Arduino* adalah mikrokontroler *single-board* bersifat *open source*, dirancang untuk mengatur penggunaan elektronik dalam berbagai bidang. *Hardware* dalam *Arduino* sendiri memiliki *processor Atmel AVR* dan menggunakan *Arduino IDE* untuk mengkompilasi kode-kode program dan memasukkan kedalam *chip Arduino* tersebut. *Arduino* memiliki banyak macam-macam *board*. Namun, dalam penelitian ini penulis menggunakan *Arduino Uno* sebagai bahan penelitian.

*Arduino Uno* adalah papan mikrokontroler berbasis *ATmega328*. *Arduino Uno* memiliki 14 pin digital *input/output*, dimana 6 pin tersebut dapat digunakan sebagai *output PWM (Pulse Width Modulation)*, 6 pin digunakan sebagai *analog input*, 16 MHz *Crystal Oscillator*, koneksi USB (*Universal Serial Bus*), kabel *jack* sebagai catu daya external, *header ICSP*, dan tombol reset. Semua ini berisi hal-hal yang diperlukan untuk mendukung mikrokontroler. Agar mikrokontroller bisa

digunakan, cukup menghubungkan *Arduino Uno Board* ke komputer melalui kabel USB, sumber tegangan dengan adaptor AC-DC, dan baterai<sup>[1][7]</sup>.



**Gambar 2.4** Arduino Uno

Nama “*Uno*” berasal dari bahasa Italia yang artinya satu, untuk menandai peluncuran pertama *Arduino 1.0*. Pada versi 1.0 sebagai referensi untuk *Arduino* yang selanjutnya. *Arduino Uno R3* merupakan versi terbaru dan standar *Arduino Uno*<sup>[13]</sup>.



**Gambar 2.5** Perbandingan Versi Arduino Uno

Berikut ini adalah spesifikasi dari *Arduino Uno*:

1. 14 pin IO Digital (pin 0-13)

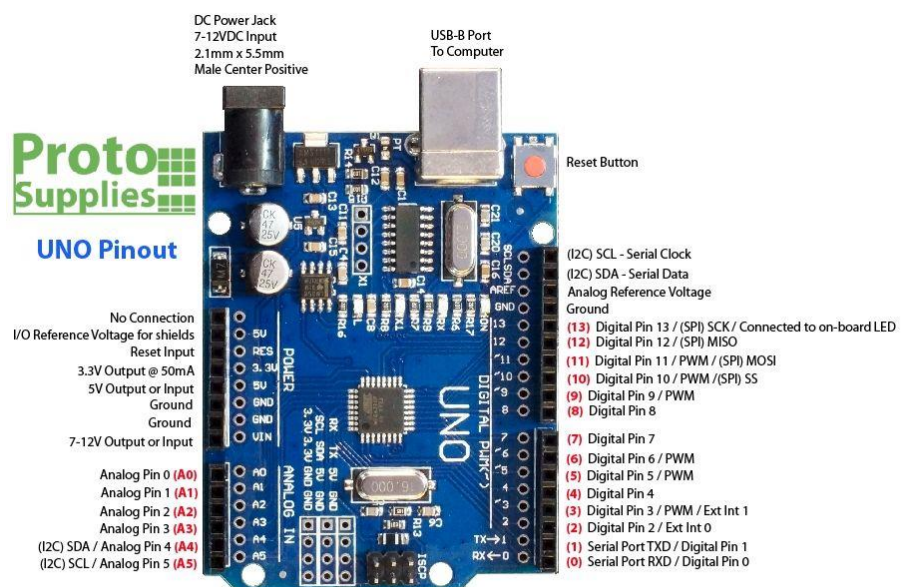
Sejumlah *digital pin* dengan nomor 0-13 yang dapat digunakan sebagai *input/output* yang dapat diatur dengan cara membuat program pada IDE.

2. 6 pin *Input Analog* (pin 0-5)

Sejumlah *analog pin* dengan nomor 0-5 yang dapat digunakan untuk membaca nilai *input* yang memiliki nilai *analog* dan mengubahnya ke dalam angka 0-1023.

3. 6 pin *Output Analog* (pin 3, 5, 6, 9, 10, dan 11)

Sejumlah pin yang sebenarnya merupakan *digital pin* tetapi sejumlah pin tersebut dapat diprogram kembali menjadi *output analog* dengan cara membuat program pada IDE<sup>[2]</sup>.



Red numbers in paranthesis are the name to use when referencing that pin.  
Analog pins are references as A0 thru A5 even when using as digital I/O

**Gambar 2.6** Arduino Uno Pinout

**Tabel 2.2** Spesifikasi Arduino Uno

| Microcontroller                          | ATmega328                               |
|--|---|
| Clock Speed                              | 16MHz                                   |
| Operating Voltage                        | 5V                                      |
| Maximum supply Voltage (not recommended) | 20V                                     |
| Supply Voltage (recommended)             | 7-12V                                   |
| Analog Input Pins                        | 6                                       |
| Digital Input/Output Pins                | 14                                      |
| DC Current per Input/Output Pin          | 40mA                                    |
| DC Current in 3.3V Pin                   | 50mA                                    |
| SRAM                                     | 2KB                                     |
| EEPROM                                   | 1KB                                     |
| Flash Memory                             | 32KB of which 0.5KB used by boot loader |

## 2.7 Arduino IDE

*Arduino* IDE merupakan perangkat lunak untuk mengkompilasi kode-kode program pada *Arduino*. Bahasa pemrograman yang digunakan adalah bahasa pemrograman C. Program yang sudah selesai ditulis di *Arduino* IDE bisa langsung *dicompile* dan *diupload* ke *Arduino Board*. Program dalam *Arduino* terbagi menjadi 3 bagian, yaitu:

### 1. Header

Pada bagian ini ditulis penggunaan *library* dan pendefinisian *variable* yang dibutuhkan. Blok ini dijalankan hanya sekali pada waktu *compiling*. Contoh perintah ini adalah penulisan kata *define* pada awal pengkodean.

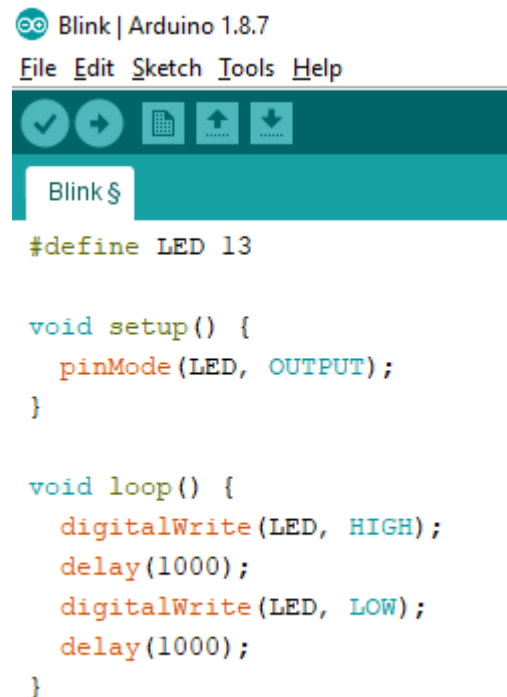
### 2. Setup

Pada bagian ini awal program *Arduino* berjalan. Contoh perintah ini adalah penulisan fungsi pada void setup. Lalu, blok ini diisi suatu penentuan apakah suatu pin digunakan sebagai *input* atau *output*. Inisialisasi *variable* bisa juga dilakukan di blok ini.

### 3. Loop

Pada bagian ini program akan dijalankan secara terus menerus. Contoh perintah ini adalah penulisan fungsi pada void loop. Pada bagian ini berisi

perintah utama pada *Arduino*. Apabila program sudah sampai di baris terakhir, maka akan dilanjutkan dengan mengulang eksekusi dari awal blok *loop*<sup>[2]</sup>.



**Gambar 2.7** Tampilan Arduino IDE

## 2.8 Bahasa Pemrograman C

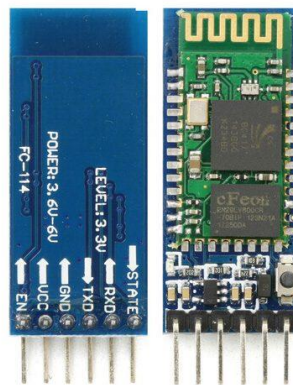
Bahasa Pemrograman C adalah sebuah bahasa pemrograman komputer yang bisa digunakan untuk membuat berbagai aplikasi (*general-purpose programming language*), mulai dari sistem operasi (seperti *Windows* atau *Linux*), *antivirus*, *software* pengolah gambar (*image processing*), hingga *compiler* untuk bahasa pemrograman, dimana C banyak digunakan untuk membuat bahasa pemrograman lain yang salah satunya adalah PHP.

Meskipun termasuk *general-purpose programming language*, yakni bahasa pemrograman yang bisa membuat berbagai aplikasi, bahasa pemrograman C paling cocok merancang aplikasi yang berhubungan langsung dengan Sistem Operasi dan *hardware*. Ini tidak terlepas dari tujuan awal bahasa C dikembangkan<sup>[22]</sup>.

## 2.9 Bluetooth Module HC-05

*Bluetooth Module HC-05* merupakan modul *Bluetooth SPP (Serial Port Protocol)* yang dirancang untuk komunikasi nirkabel pada frekuensi 2,4 GHz. Modul ini merupakan salah satu yang sering diproduksi dan digunakan. *Interface* yang digunakan adalah *serial RXD, TXD, VCC, dan GND*. Terdapat *build in LED* sebagai indikator koneksi *bluetooth*. Jarak efektif modul ini mencapai 10 meter. Berikut ini adalah spesifikasi dari HC-05:

1. Menggunakan versi *Bluetooth V2.0+EDR (Enhanced Data Range)* dengan *data rate* sebesar 3 Mbps.
2. Frekuensi 2,4 GHz ISM Band.
3. Sensitivitas mencapai -80 dBm.
4. Menggunakan gelombang radio kelas 2, dengan daya transmisi mencapai 4 dBm dan jarak mencapai 10 meter.
5. Interface UART dengan spesifikasi *baudrate* yang mendukung (9600, 19200, 38400, 57600, 230400 dan 460800 bps)<sup>[4]</sup>.

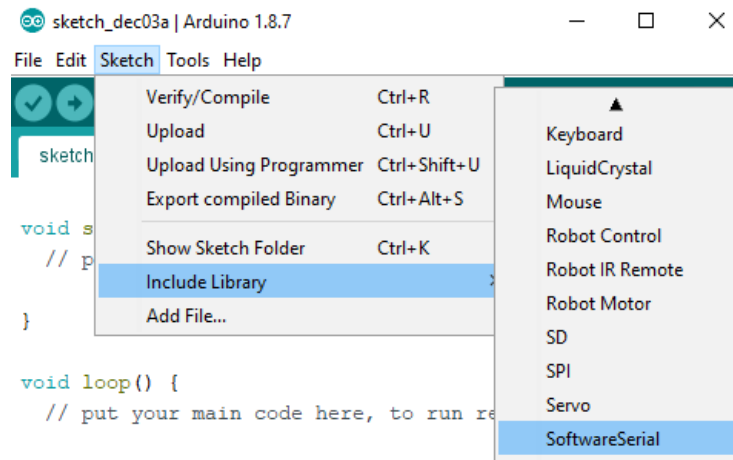


**Gambar 2.8** Bluetooth Module HC-05

Untuk menggunakan *Bluetooth Module* diperlukan perintah - perintah seperti berikut ini :

1. Komponen ini membutuhkan *header* dengan cara tekan menu *sketch, include library* dan pilih *SoftwareSerial*.





**Gambar 2.9** Library Bluetooth Module

2. Deklarasikan pin RXD dan TXD sesuai dengan gambar 2.9 dan 2.11. Lalu, beri nama variabel untuk *header bluetooth* dan variabel untuk menampung *input bluetooth*

```
sketch_dec03a $
#include <SoftwareSerial.h> //header bluetooth
#define rx_pin 0           //hubungkan dengan pin arduino
#define tx_pin 1           //hubungkan dengan pin arduino

SoftwareSerial BTSerial(rx_pin, tx_pin); // pendeklarasian variabel
String BTInput;              // pendeklarasian variabel
```

**Gambar 2.10** Deklarasi Variabel Bluetooth Module

3. Pada bagian *void setup*, perintah tersebut untuk memulai berkomunikasi dengan *bluetooth module* dengan rate 9600 (*default*).

```
sketch_dec03a $
#include <SoftwareSerial.h> //header bluetooth
#define rx_pin 0           //hubungkan dengan pin arduino
#define tx_pin 1           //hubungkan dengan pin arduino

SoftwareSerial BTSerial(rx_pin, tx_pin); // pendeklarasian variabel
String BTInput;              // pendeklarasian variabel

void setup() {
  BTSerial.begin(9600); // untuk memulai module bluetooth
}
```

**Gambar 2.11** Void Setup Bluetooth Module

4. Pada bagian *void loop*, diperlukan perintah *if* untuk mensortir koneksi bluetooth. Jika koneksi *bluetooth* terhubung dengan perangkat, maka akan membaca *input* dari user kedalam tipe data *string*. Kemudian, masukkan perintah utama kedalam blok ini<sup>[8]</sup>.

```
sketch_dec03a $
#include <SoftwareSerial.h> //header bluetooth
#define rx_pin 0           //hubungkan dengan pin arduino
#define tx_pin 1           //hubungkan dengan pin arduino

SoftwareSerial BTSerial(rx_pin, tx_pin); // pendeklarasian variabel
String BTInput;           // pendeklarasian variabel

void setup() {
  BTSerial.begin(9600); // untuk memulai module bluetooth
}

void loop() {
  if (BTSerial.available()) // mensortir koneksi bluetooth
  {
    BTInput = BTSerial.readString(); // membaca inputan user kedalam tipe data string
    // masukkan perintah utama disini
  }
}
```

**Gambar 2.12** Void Loop Bluetooth Module

## 2.10 AC Light Dimmer Module

Modul ini dapat mengontrol jumlah daya listrik yang dialirkan ke sirkuit AC (*Alternating Current*) 110 atau 220V dengan cara yang efisien. Modul ini biasanya digunakan untuk mengatur kecerahan lampu, mengatur kecepatan motor elektrik AC, mengatur kecepatan kipas angin, dan mengatur peralatan elektronik lainnya yang berarus listrik AC. *Interface* yang digunakan adalah *serial* VCC, GND, ZC (*Zero Cross*), dan PWM<sup>[9]</sup>. Modul ini berbasis TRIAC BTA16-600B sebagai pusat pengendali arus listrik AC<sup>[10][11]</sup>.



**Gambar 2.13** AC Light Dimmer Module

Modul ini berkomunikasi dengan mikrokontroler melalui 2 pin digital, yaitu:

1. ZC (*Zero Cross*) pin

Pada pin ini untuk menginformasikan mikrokontroler seberapa besar arus listrik yang dialirkan dan memungkinkan mikrokontroler untuk mengubah *power supply* yang terkoneksi dengan sirkuit AC menjadi *on/off*. Setiap mikrokontroller memiliki perbedaan pin untuk ZC (lihat tabel 2.3).

**Tabel 2.3** Pin Zero Cross Pada Setiap Mikrokontroler

| Board                                      | INPUT Pin<br>Zero Cross   |   | OUTPUT Pin  |   |
|--|---|---|---|---|
| Leonardo                                   | D7 (NOT CHANGABLE)  |   | D0-D6, D8-D13   |   |
| Mega                                       | D2 (NOT CHANGABLE)  |   | D0-D1, D3-D70   |   |
| UNO<br>NANO                                | D2 (NOT CHANGABLE)  |   | D0-D1, D3-D20   |   |
| ESP8266                                    | <ul style="list-style-type: none"> <li>D1(I/O5)</li> <li>D5(I/O14)</li> <li>D7(I/O13)</li> </ul>  | <ul style="list-style-type: none"> <li>D2(I/O4)</li> <li>D6(I/O12)</li> <li>D8(I/O15)</li> </ul>  | <ul style="list-style-type: none"> <li>D0(I/O16)</li> <li>D2(I/O4)</li> <li>D6(I/O12)</li> <li>D8(I/O15)</li> </ul>   | <ul style="list-style-type: none"> <li>D1(I/O5)</li> <li>D5(I/O14)</li> <li>D7(I/O13)</li> </ul>  |
| ESP32                                      | <ul style="list-style-type: none"> <li>4(GPI36)</li> <li>5(GPI39)</li> <li>8(GPO32)</li> <li>10(GPIO25)</li> <li>12(GPIO27)</li> <li>14(GPIO12)</li> <li>21(GPIO7)</li> <li>24(GPIO2)</li> <li>26(GPIO4)</li> <li>28(GPIO17)</li> <li>30(GPIO18)</li> <li>33(GPIO21)</li> <li>36(GPIO22)</li> </ul> | <ul style="list-style-type: none"> <li>6(GPI34)</li> <li>7(GPI35)</li> <li>9(GPO33)</li> <li>11(GPIO26)</li> <li>13(GPIO14)</li> <li>16(GPIO13)</li> <li>23(GPIO15)</li> <li>25(GPIO0)</li> <li>27(GPIO16)</li> <li>29(GPIO5)</li> <li>31(GPIO19)</li> <li>35(GPIO1)</li> <li>37(GPIO23)</li> </ul> | <ul style="list-style-type: none"> <li>8(GPO32)</li> <li>10(GPIO25)</li> <li>12(GPIO27)</li> <li>14(GPIO12)</li> <li>23(GPIO15)</li> <li>25(GPIO0)</li> <li>27(GPIO16)</li> <li>29(GPIO5)</li> <li>31(GPIO19)</li> <li>34(GPIO3)</li> <li>36(GPIO22)</li> </ul> | <ul style="list-style-type: none"> <li>9(GPO33)</li> <li>11(GPIO26)</li> <li>13(GPIO14)</li> <li>16(GPIO13)</li> <li>24(GPIO2)</li> <li>26(GPIO4)</li> <li>28(GPIO17)</li> <li>30(GPIO18)</li> <li>33(GPIO21)</li> <li>35(GPIO1)</li> <li>37(GPIO23)</li> </ul> |
| Arduino M0<br>Arduino Zero                 | D7 (NOT CHANGABLE)  |   | D0-D6, D8-D13   |   |
| Arduino Due                                | D0-D53  |   | D0-D53  |   |
| STM32<br>Black Pill<br>Blue Pill<br>Etc... | PA0-PA15, PB0-PB15<br>PC13-PC15   |   | PA0-PA15, PB0-PB15<br>PC13-PC15   |   |

## 2. PWM (*Pulse Width Modulation*) pin

Pin ini sebagai *output digital pin* dan dapat mengatur keluaran listrik yang dihasilkan. Pada *Arduino Uno* pin ini terletak pada 3, 5, 6, 9, 10, dan 11 (gambar 2.5)<sup>[9]</sup>.

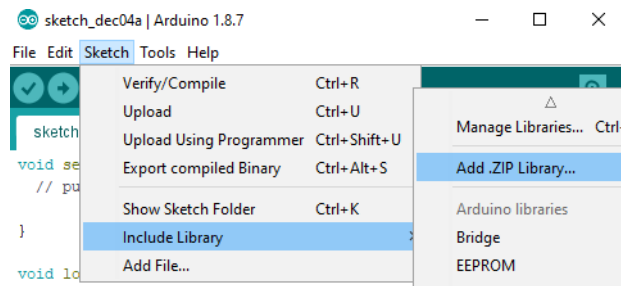
Berikut ini adalah tabel spesifikasi dari *AC Light Dimmer Module*:

**Tabel 2.4** Spesifikasi AC Light Dimmer Module

|                               |   |
|-------------------------------|---|
| Power                         | 600V - 16A  |
| AC frequency                  | 50/60 Hz  |
| TRIAC                         | BTA16 - 600B  |
| Isolation                     | Optocoupler   |
| Logic level                   | 3.3V/5V   |
| Zero point                    | Logic level   |
| Modulation (PWM with trigger) | logic level ON/OFF TRIAC  |
| Signal current                | > 10mA  |
| Environment:                  | <ul style="list-style-type: none"> <li>• For indoor and outdoor use</li> <li>• Operating temperatures: -20°C to 80°C</li> </ul> |
| Operating humidity            | Dry environment only  |
| ROHS3                         | Compliant   |

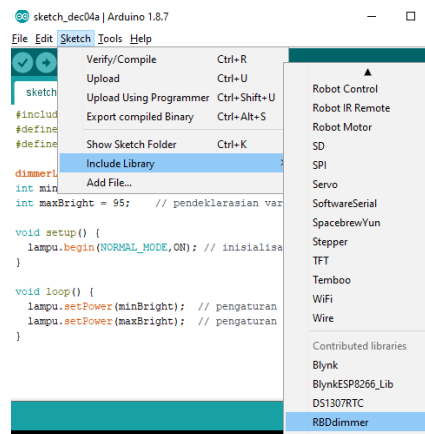
Untuk menggunakan *AC Light Dimmer Module* diperlukan perintah - perintah seperti berikut ini :

1. Mengunduh *library* komponen pada website github (link terdapat di daftar pustaka).
2. Melakukan instalasi *library* tersebut dengan cara menekan menu *sketch*, *include library*, *add .ZIP library* dan mencari unduhan *library*.



**Gambar 2.14** Instalasi Library Dimmer Module

3. Pilih *library* tersebut dengan cara pilih menu *sketch*, *include library* dan pilih *RBDDimmer*.



**Gambar 2.15** Library Dimmer Module

4. Deklarasikan variabel untuk *header dimmer* modul dan tingkat kecerahan lampu.

```
sketch_dec04a $
#include <RBDDimmer.h> //header modul dimmer
#define pwm 3 // pin arduino sebagai output lampu
#define zerocross 2 // hubungkan dengan pin arduino

dimmerLamp lampu(pwm); // pendeklarasian variabel
int minBright = 0; // pendeklarasian variabel
int maxBright = 95; // pendeklarasian variabel
```

**Gambar 2.16** Deklarasi Variabel Dimmer Module

5. Pada *void setup* berisikan inisialisasi lampu saat pertama kali mikrokontroller menyala.

```
#include <RBDDimmer.h> //header modul dimmer
#define pwm 3 // pin arduino sebagai output lampu
#define zerocross 2 // hubungkan dengan pin arduino

dimmerLamp lampu(pwm); // pendeklarasian variabel
int minBright = 0; // pendeklarasian variabel
int maxBright = 95; // pendeklarasian variabel

void setup() {
  lampu.begin(NORMAL_MODE,ON); // inisialisasi lampu
}
```

**Gambar 2.17** Void Setup Dimmer Module

6. *void loop* berisikan perintah untuk mengatur tingkat intensitas cahaya<sup>[18]</sup>

```
#include <RBDDimmer.h> //header modul dimmer
#define pwm 3 // pin arduino sebagai output lampu
#define zerocross 2 // hubungkan dengan pin arduino

dimmerLamp lampu(pwm); // pendeklarasian variabel
int minBright = 0; // pendeklarasian variabel
int maxBright = 95; // pendeklarasian variabel

void setup() {
  lampu.begin(NORMAL_MODE,ON); // inisialisasi lampu
}

void loop() {
  lampu.setPower(minBright); // pengaturan intensitas cahaya
  lampu.setPower(maxBright); // pengaturan intensitas cahaya
}
```

**Gambar 2.18** Void Loop Dimmer Module

## 2.11 Lampu Bohlam

Lampu pijar atau lampu bohlam pertama kali ditemukan oleh Thomas Alva Edison. Lini bekerja dengan cara menyalurkan arus listrik melalui filamen yang kemudian memanaskan dan menghasilkan cahaya. Kaca yang menyelubungi filamen panas tersebut menghalangi kontak langsung dengan udara sehingga filamen dapat bertahan lama karena tidak terkena oksidasi.

Kelebihan lampu ini adalah harga di pasaran lebih murah apabila dibandingkan dengan jenis lampu lainnya. Kekurangan dari lampu ini adalah Tidak efisien dalam penggunaan listrik dikarenakan energi yang digunakan cukup besar. Selain itu, panas yang dihasilkan cukup besar<sup>[19]</sup>.



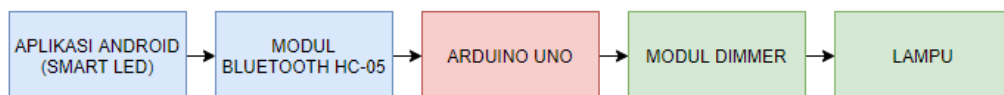
**Gambar 2.19** Lampu Bohlam

## BAB III

### PEMBAHASAN

#### 3.1 Arduino

##### 3.1.1 Blok Diagram



**Gambar 3.1** Blok Diagram pada Perancangan Alat

Pada perancangan ini, blok diagram terdiri 3 komponen utama, input sebagai media masukan (biru), proses sebagai unit utama dalam mengolah data (merah), dan output sebagai hasil dari pengolahan data (hijau).

Pada komponen input menjelaskan bahwa aplikasi android berfungsi untuk mengatur tingkat kecerahan lampu. Hal ini dikarenakan aplikasi *android* terhubung dengan modul *bluetooth*. Lalu modul *bluetooth* akan meneruskan perintah menuju komponen proses.

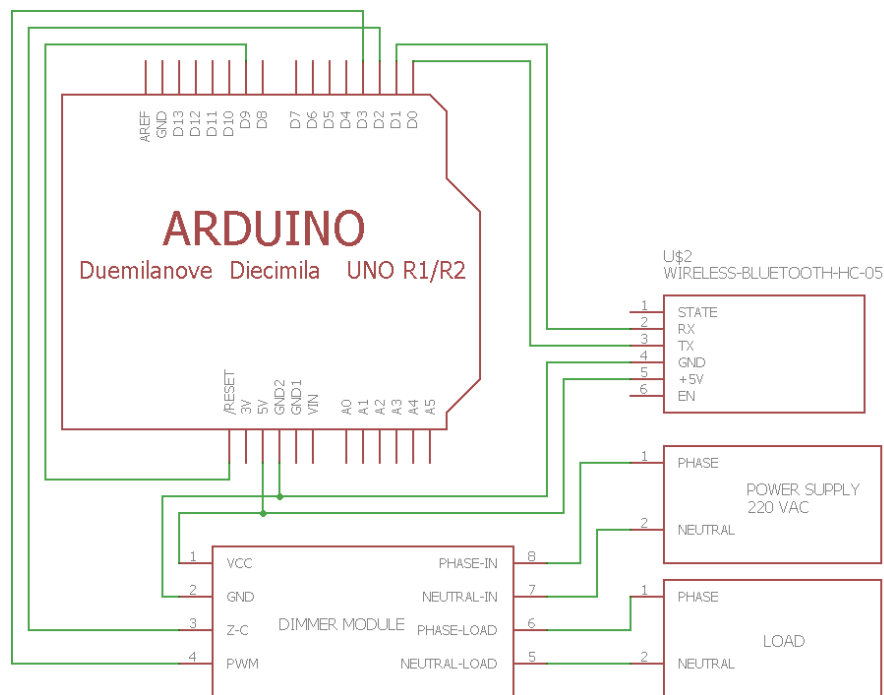
Pada komponen proses menjelaskan bahwa *arduino uno* akan memproses perintah yang dikirimkan dari komponen *input*. Perintah yang sudah diproses kemudian dikirimkan menuju komponen output.

Pada komponen *output* terdapat 2 komponen, yaitu modul *dimmer*, dan lampu. Modul *dimmer* berfungsi sebagai pengontrol tegangan listrik sesuai dengan perintah yang dikirimkan pada komponen proses dan lampu sebagai hasil dari perintah yang diberikan.



### 3.1.2 Skematik Rangkaian

Rancangan rangkaian pada Aplikasi Pengendali Lampu Berbasis Android dengan Menggunakan Arduino Uno, Modul Bluetooth dan Modul Dimmer dapat dilihat pada skematik berikut ini.



**Gambar 3.2** Skematik Aplikasi Pengendali Lampu Berbasis Android dengan Menggunakan Arduino Uno, Modul Bluetooth dan Modul Dimmer

Hubungan pin *Arduino Uno* dengan komponen catu daya, input dan output akan ditunjukkan pada tabel berikut.

**Tabel 3.1** Hubungan Pin Arduino pada Aplikasi Pengendali Lampu Berbasis Android dengan Menggunakan Arduino Uno, Modul Bluetooth dan Modul Dimmer

| Pin Arduino Uno | Pin Komponen | Keterangan                             |
|-----------------|--------------|--|
| 5V              | VCC          | Modul Bluetooth HC-05 dan Modul Dimmer |
| GND             | GND          | Modul Bluetooth HC-05 dan Modul Dimmer |
| 0               | TX           | Modul Bluetooth HC-05                  |
| 1               | RX           | Modul Bluetooth HC-05                  |
| 2               | Z-C          | Modul Dimmer                           |
| 3               | PWM          | Modul Dimmer                           |
| 9               | RESET        | Arduino Uno                            |

### 3.1.3 Implementasi Program Arduino Uno

Pada bagian ini menjelaskan kode program pada *Arduino Uno* sebagai pengontrol bagian komponen pemrosesan pada Aplikasi Pengendali Lampu Berbasis Android dengan Menggunakan Arduino Uno, Modul Bluetooth dan Modul Dimmer. Berikut ini adalah kode program dari *Arduino Uno*.

```
#include <RBDdimmer.h>
#include <SoftwareSerial.h>
#define rx_pin 0
#define tx_pin 1
#define pwm 3
#define zerocross 2
#define resetPin 9

SoftwareSerial BTSerial(rx_pin, tx_pin);
dimmerLamp lampu(pwm);

String BTInput;
int waiting = 1000;
int minBright = 0;
int maxBright = 95;
int value, counter;
int storeData[300];
boolean isPowered = false;
```

Pada penggalan kode diatas merupakan bagian header. Bagian ini biasanya diisi dengan library atau pendeklarasian variabel. Pada line 1 dan 2 merupakan contoh library pada dimmer dan bluetooth modul. Pada line 3 – 5 merupakan hubungan antara pin pada arduino uno dan komponen bluetooth dan dimmer modul. Pada line 9 dan 10 merupakan pendeklarasian variabel untuk masing – masing

header module. Pada line 12 – 18 merupakan pendeklarasian variabel yang dibutuhkan dalam pembuatan alat ini.

```
void setup()
{
  digitalWrite(resetPin, HIGH);
  BTSerial.begin(9600);
  pinMode(resetPin, OUTPUT);
  lampu.begin(NORMAL_MODE, ON);
}
```

Pada penggalan kode diatas merupakan bagian *setup*. Bagian ini berisi inisialisasi pin pada *Arduino Uno*. Pada line 3 digunakan untuk insialisasi reset pada arduino uno. Pada line 4 merupakan untuk inisialisasi bluetooth dengan baudrate 9600 (default). Pada line 5 perintah tersebut digunakan untuk pin reset sebagai output yang nantinya akan dibutuhkan. Pada line 6 digunakan untuk inisialisasi mode lampu saat mikrokontroller pertama kali menyala dengan mode normal.

```
void loop()
{
  arrayCheck();
  if (BTSerial.available())
  {
    BTInput = BTSerial.readString();
    powerCheck();
    if (value <= maxBright && isPowered == false)
    {
      delay(waiting);
      lampu.setPower(minBright);
    }

    else if (value > minBright && isPowered == true)
    {
      delay(waiting);
      lampu.setPower(value);
    }
  }
  else if (value == minBright)
  {
    delay(waiting);
    lampu.setPower(storeData[counter-1]);
  }
}
```

Pada penggalan kode diatas merupakan bagian loop. Bagian ini akan dijalankan secara terus menerus. Maksud dari perintah diatas adalah jika variabel `isPowered` bernilai `false` dan value lebih kecil daripada 95 maka lampu akan mati. Jika variabel `isPowered` bernilai `true` dan value lebih besar dari 0 maka lampu akan menyala sesuai dengan variabel `value`.

```
void powerCheck()
{
    if (BTInput.equals("ON"))
    {
        isPowered = true;
    }

    else if (BTInput.equals("OFF"))
    {
        isPowered = false;
    }

    else
    {
        value = BTInput.toInt();
        counter = counter + 1;
        storeData[counter] = value;
    }
}

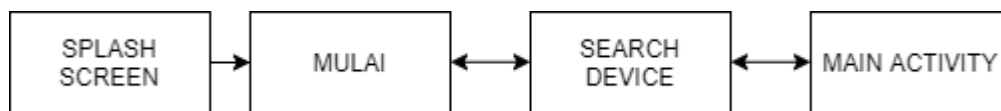
void arrayCheck()
{
    if (counter == 295)
    {
        digitalWrite(resetPin, LOW);
    }
}
```

Pada penggalan kode diatas merupakan pendeklarasian *function* pada program *Arduino*. *Function* akan dieksekusi jika dipanggil pada blok program lain. Pada *function* `powerCheck` merupakan pemilah data. Jika *Arduino* menerima perintah “ON” maka variabel `isPowered` menjadi `true` begitu juga sebaliknya. Jika *Arduino* menerima perintah berupa angka maka akan dirubah menjadi tipe data integer. Karena variabel `BTInput` merupakan tipe data string. Variabel `counter` untuk menambahkan elemen array pada variabel `storeData`. Variabel `storeData`

untuk menyimpan data berupa angka kecerahan lampu karena saat *bluetooth disconnected* akan memberikan angka 0 yang menyebabkan lampu mati saat koneksi terputus. Untuk mengatasi hal tersebut penulis menambahkan variabel array agar lampu tetap hidup walau koneksi terputus. Pada *function* `arrayCheck` merupakan perintah untuk mereset *Arduino*. Jika tidak direset *Arduino* akan mengalami *out of array*.

## 3.2 Android

### 3.2.1 Perancangan Struktur Navigasi



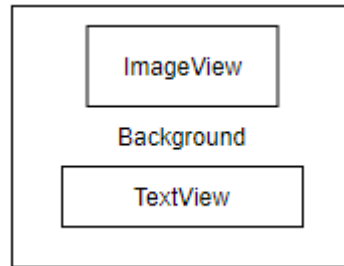
**Gambar 3.3** Perancangan Struktur Navigasi Linier Aplikasi Pengendali Lampu Berbasis Android dengan Menggunakan Arduino Uno, Modul Bluetooth dan Modul Dimmer

Berdasarkan sub bab 2.4 penulis menggunakan struktur navigasi linier karena halaman ditunjukkan secara berurut. Terdapat 3 halaman utama, yaitu *mulai*, *search device*, dan *main activity*. Pada *activity* *mulai* berisi perintah untuk menyalakan *bluetooth*. Pada *search device activity* berisi perintah untuk mencari dan menghubungkan *bluetooth*. Pada *main activity* berisi perintah untuk mengontrol penggunaan lampu.

### 3.2.2 Perancangan Halaman

#### 3.2.2.1 Perancangan Halaman Splash Screen

Perancangan halaman *Splash Screen* berisi *ImageView* dan *TextView*. Fungsi dari *Splash Screen* adalah memberikan kesan pertama kali saat pengguna membuka aplikasi. Berikut ini merupakan tampilan dari rancangan halaman *Splash Screen*.



**Gambar 3.4** Rancangan Halaman Splash Screen

### Implementasi Koding :

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/background"
    tools:context=".splash_screen">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="291dp"
        android:layout_height="296dp"
        android:src="@drawable/logo"
        app:layout_constraintBottom_toTopOf="@+id/textView"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.616" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="SMART LED"
        android:textColor="#FFFFFF"
        android:textSize="35sp"
        android:textStyle="bold"
        app:fontFamily="@font/cherry_cream_soda"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">
```

```

app:layout_constraintVertical_bias="0.628" />

<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Simple application to control your lamp"
    android:textColor="#FFFFFF"
    android:textSize="20sp"
    android:textStyle="italic"
    app:fontFamily="@font/assistant_extralight"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.494"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.714" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

Penggalan program diatas merupakan pengaturan tampilan pada splash screen. Terdapat background yang menjadi latar belakang dari splash screen. Untuk menggunakan background berikan perintah “@drawable”. Berikut ini adalah penulisan kode splash\_screen.java.

```

package com.example.bluetooth;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.view.WindowManager;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.ImageView;
import android.widget.TextView;

public class splash_screen extends AppCompatActivity
{
    private static int splash_time = 5000;

    Animation topAnim,botAnim;
    ImageView gambar;
    TextView first,second;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
    }
}

```

```

getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,WindowManager.Lay
outParams.FLAG_FULLSCREEN);

setContentView(R.layout.activity_splash_screen);

topAnim = AnimationUtils.loadAnimation(this,R.anim.top_animation);
botAnim = AnimationUtils.loadAnimation(this,R.anim.bottom_animation);

gambar = findViewById(R.id.imageView);
first = findViewById(R.id.textView);
second = findViewById(R.id.textView2);

gambar.setAnimation(topAnim);
first.setAnimation(botAnim);
second.setAnimation(botAnim);

new Handler().postDelayed(new Runnable()
{
    @Override
    public void run()
    {
        Intent intent = new Intent(splash_screen.this, MulaiActivity.class);
        startActivity(intent);
        finish();
    }
},splash_time);
}
}

```

Penggalan program diatas menjelaskan ketika aplikasi dijalankan maka halaman splash screen akan diam selama 5 detik dan program masuk ke halaman mulai. Berikut ini adalah tampilan dari splash screen.

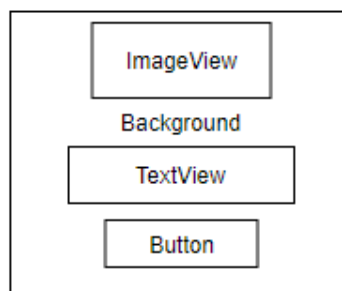


**Gambar 3.5** Tampilan Halaman Splash Screen



### 3.2.2.2 Perancangan Halaman Mulai

Perancangan halaman Mulai berisi *ImageView*, *TextView*, dan *Button*. Jika *Button* di tekan maka akan memberikan perintah untuk menyalakan *Bluetooth* lalu akan menuju ke halaman *search device*. Berikut ini merupakan rancangan halaman mulai.



**Gambar 3.6** Rancangan Halaman Mulai

Implementasi Koding :

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/background"
    tools:context=".MulaiActivity">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="291dp"
        android:layout_height="296dp"
        android:src="@drawable/logo"
        app:layout_constraintBottom_toTopOf="@+id/textView"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.616" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="SMART LED"
        android:textColor="#FFFFFF"

```

```

        android:textSize="35sp"
        android:textStyle="bold"
        app:fontFamily="@font/cherry_cream_soda"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.628" />

<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Simple application to control your lamp"
    android:textColor="#FFFFFF"
    android:textSize="20sp"
    android:textStyle="italic"
    app:fontFamily="@font/assistant_extralight"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.494"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.714" />

<Button
    android:id="@+id/Mulai"
    android:layout_width="70dp"
    android:layout_height="70dp"
    android:background="@drawable/go_button"
    android:text="GO"
    android:textColor="#FBFBFB"
    android:textSize="20dp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.857" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

Penggalan program diatas menjelaskan pengaturan tampilan pada halaman mulai. *ImageView* berfungsi sebagai menampilkan gambar pada suatu activity. *TextView* berfungsi sebagai menampilkan teks pada suatu activity. Berikut ini adalah penulisan kode MulaiActivity.java.

```

package com.example.bluetooth;

import androidx.appcompat.app.AppCompatActivity;

```

```

import android.bluetooth.BluetoothAdapter;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.os.Bundle;
import android.view.View;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.Toast;
import java.lang.String;

public class MulaiActivity extends AppCompatActivity
{

    Button xMulai;
    BluetoothAdapter mBlueAdapter;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);

        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,WindowManager.Lay
        outParams.FLAG_FULLSCREEN);
        setContentView(R.layout.activity_mulai);

        xMulai = findViewById(R.id.Mulai);

        mBlueAdapter = BluetoothAdapter.getDefaultAdapter();

        xMulai.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View view)
            {
                enableBT();
            }
        });
    }

    public void enableBT()
    {
        if (!mBlueAdapter.isEnabled())
        {
            Intent enableBTIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
            startActivity(enableBTIntent);

            IntentFilter BTIntent = new IntentFilter(BluetoothAdapter.ACTION_STATE_CHANGED);
            registerReceiver(mBroadcastReceiver1, BTIntent);
        }
    }
}

```

```

public BroadcastReceiver mBroadcastReceiver1 = new BroadcastReceiver()
{
    @Override
    public void onReceive(Context context, Intent intent)
    {
        String action = intent.getAction();
        if (action.equals(mBlueAdapter.ACTION_STATE_CHANGED))
        {
            final int state = intent.getIntExtra(BluetoothAdapter.EXTRA_STATE,
mBlueAdapter.ERROR);
            switch(state)
            {
                case BluetoothAdapter.STATE_TURNING_ON:
                    showMessage("Activating Bluetooth...");
                    break;

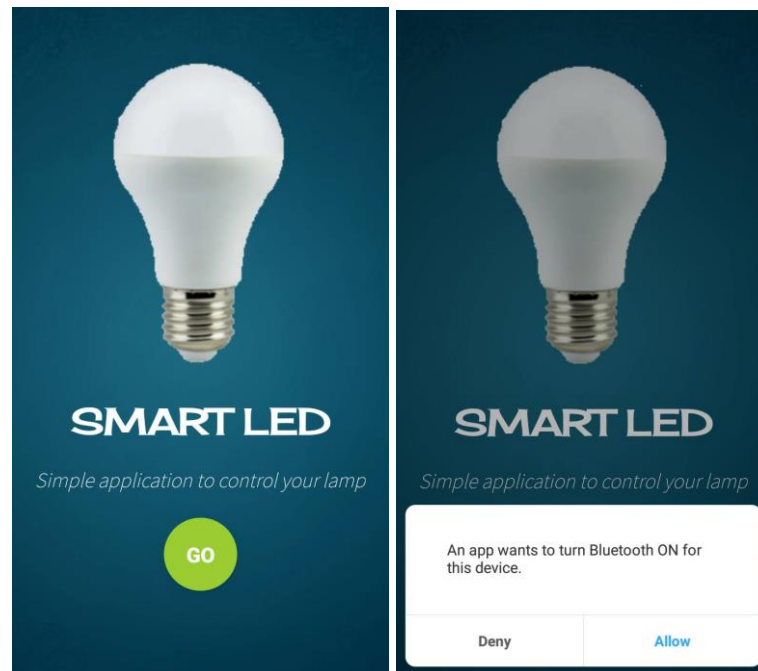
                case BluetoothAdapter.STATE_ON:
                    showMessage("Bluetooth Activated!");
                    nextActivity();
                    break;
            }
        }
    }
};

public void nextActivity()
{
    Intent aw = new Intent(this, SearchActivity.class);
    startActivity(aw);
    overridePendingTransition(R.anim.slide_in_right, R.anim.slide_out_left);
}

public void showMessage(String msg)
{
    Toast.makeText(this, msg, Toast.LENGTH_SHORT).show();
}
}

```

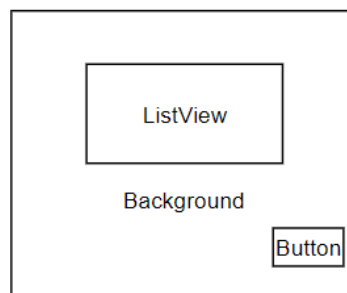
Penggalan program diatas menjelaskan ketika pengguna menekan tombol “GO” maka program akan menyalakan *Bluetooth* dan menuju ke halaman search device. Berikut ini tampilan halaman mulai.



**Gambar 3.7** Tampilan Halaman Mulai dan Tombol GO Saat Ditekan

### 3.2.2.3 Perancangan Halaman Search Device

Perancangan halaman *Search Device* berisi *ListView*, dan *Button*. Jika *Button* di tekan maka akan memberikan perintah untuk mencari alamat *Bluetooth* dan *ListView* akan menampilkan alamat *Bluetooth*. Pengguna memilih alamat *Bluetooth* yang ingin dihubungkan dan akan memberikan perintah *pairing* dan akan menuju ke halaman *main activity*. Berikut ini merupakan rancangan halaman mulai.



**Gambar 3.8** Rancangan Halaman Search Device

Implementasi Koding :

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">
```

```

xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@drawable/background"
tools:context=".SearchActivity">

<Button
    android:id="@+id/search"
    android:layout_width="80dp"
    android:layout_height="80dp"
    android:layout_marginEnd="40dp"
    android:layout_marginRight="40dp"
    android:layout_marginBottom="50dp"
    android:background="@drawable/search_button"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/DevicesList" />

<ListView
    android:id="@+id/DevicesList"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_marginTop="85dp"
    android:layout_marginBottom="183dp"
    app:layout_constraintBottom_toTopOf="@+id/search"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

Penggalan program diatas menjelaskan pengaturan tampilan pada halaman search device. *ListView* berfungsi sebagai menampung alamat yang sudah didapatkan oleh *search button*. Berikut ini adalah penulisan kode SearchActivity.java.

```

package com.example.bluetooth;

import androidx.appcompat.app.AppCompatActivity;

import android.Manifest;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.os.Build;
import android.os.Bundle;
import android.bluetooth.BluetoothAdapter;
import android.view.WindowManager;
import android.widget.AdapterView;
import android.widget.Button;

```

```

import android.widget.ListView;
import android.view.View;
import android.bluetooth.BluetoothDevice;
import java.util.ArrayList;
import java.lang.String;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.AdapterView;

public class SearchActivity extends AppCompatActivity
{
    Button vSearchDevices;
    ListView vListDevices;
    ArrayList<String> vArrayList = new ArrayList<String>();
    ArrayAdapter<String> vArrayAdapter;
    BluetoothAdapter vBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
    private static String address = "0";

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);

        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.Lay
outParams.FLAG_FULLSCREEN);
        setContentView(R.layout.activity_search);

        setTitle("Device List");

        vSearchDevices = findViewById(R.id.search);
        vListDevices = findViewById(R.id.DevicesList);

        vSearchDevices.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View view)
            {
                BTSearch();
            }
        });
    }

    public void BTSearch()
    {
        if (vBluetoothAdapter.isDiscovering())
        {
            vBluetoothAdapter.cancelDiscovery();

            checkBT();

            vBluetoothAdapter.startDiscovery();
            IntentFilter discoverDevices = new IntentFilter(BluetoothDevice.ACTION_FOUND);
            registerReceiver(vBroadcastReceiver3, discoverDevices);

```

```

    }

    if (!vBluetoothAdapter.isDiscovering())
    {
        checkBT();

        vBluetoothAdapter.startDiscovery();
        IntentFilter discoverDevices = new IntentFilter(BluetoothDevice.ACTION_FOUND);
        registerReceiver(vBroadcastReceiver3, discoverDevices);
    }
}

public BroadcastReceiver vBroadcastReceiver3 = new BroadcastReceiver()
{
    @Override
    public void onReceive(Context context, Intent intent)
    {
        final String action = intent.getAction();
        if (action.equals(BluetoothDevice.ACTION_FOUND))
        {
            BluetoothDevice device = intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);
            String devicename = device.getName();
            String macaddress = device.getAddress();
            vArrayList.add("Name : "+devicename+"MAC Address : "+macaddress);
            vArrayAdapter = new ArrayAdapter<String>(getApplicationContext(),
            android.R.layout.simple_list_item_1, vArrayList);
            vListDevices.setAdapter(vArrayAdapter);
            vListDevices.setOnItemClickListener(vClickListDevices);
        }
    }
};

public final AdapterView.OnItemClickListener vClickListDevices = new
AdapterView.OnItemClickListener()
{
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id)
    {
        String info = ((TextView) view).getText().toString();
        address = info.substring(info.length() - 17);
        NextActivity();
    }
};

public void NextActivity()
{
    Intent aw = new Intent(SearchActivity.this, MainActivity.class);
    startActivity(aw);
    overridePendingTransition(R.anim.slide_in_right, R.anim.slide_out_left);
}

public static String getVariable()
{

```



```

return address; //oper variabel
}

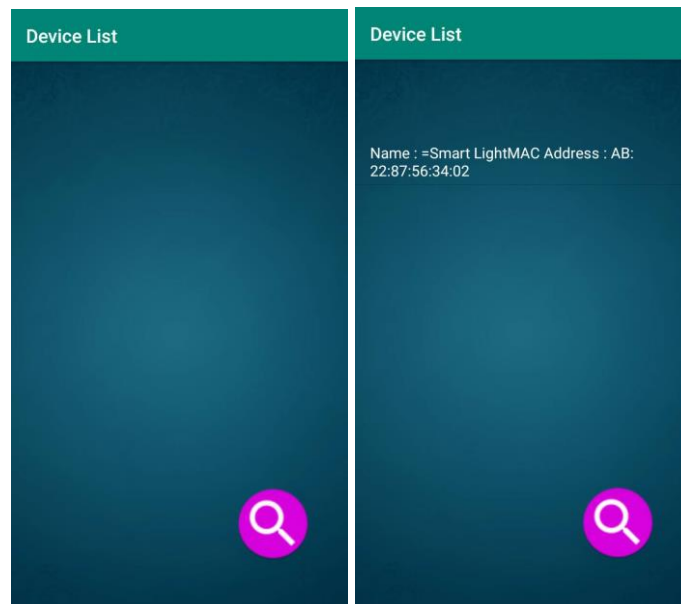
public void checkBT()
{
    if (Build.VERSION.SDK_INT > Build.VERSION_CODES.LOLLIPOP)
    {
        int permissionCheck =
this.checkSelfPermission("Manifest.permission.ACCESS_FINE_LOCATION");
        permissionCheck +=
this.checkSelfPermission("Manifest.permission.ACCESS_COARSE_LOCATION");
        if (permissionCheck != 0)
        {
            this.requestPermissions(new String[]{Manifest.permission.ACCESS_FINE_LOCATION,
Manifest.permission.ACCESS_COARSE_LOCATION}, 1001);
        }

        else
        {
            showMessage("no need to check permission. SDK version < lolipop");
        }
    }
}

public void showMessage(String msg)
{
    Toast.makeText(this, msg, Toast.LENGTH_SHORT).show();
}
}

```

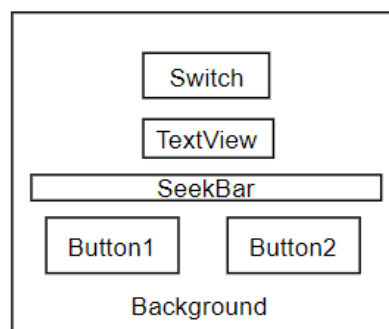
Penggalan program diatas menjelaskan ketika *button* ditekan akan mencari alamat *Bluetooth* yang tersedia. *ListView* akan menampilkan alamat *Bluetooth*. Jika pengguna menekan alamat *Bluetooth* yang tertera pada *ListView* maka program akan menyimpan alamat tersebut kedalam variabel *address* dan mengirim variabel tersebut ke halaman Main Activity. Berikut ini adalah tampilan dari search device.



**Gambar 3.9** Tampilan Halaman Search Device dan Tombol Search Saat Ditekan

#### 3.2.2.4 Perancangan Halaman Main Activity

Perancangan halaman *Main Activity* berisi *Switch*, *TextView*, *SeekBar*, *Button1*, dan *Button2*. *Switch* berfungsi untuk menyalakan atau mematikan lampu. *TextView* berfungsi untuk menampilkan angka sesuai *SeekBar*. *SeekBar* berfungsi untuk mengatur tingkat kecerahan lampu. *Button1* berfungsi untuk mengirimkan data dari *smartphone* ke *arduino uno*. *Button2* berfungsi untuk memutus jaringan *Bluetooth*. Berikut ini merupakan rancangan halaman *main activity*.



**Gambar 3.10** Rancangan Halaman Main Activity

Implementasi Koding :

```
<?xml version="1.0" encoding="utf-8"?>
```

```

<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@drawable/background"
tools:context=".MulaiActivity">

<Button
    android:id="@+id/ButtonDisconnect"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="249dp"
    android:background="@drawable/disconnect_button"
    android:text="DISCONNECT"
    android:textColor="#FBFBFB"
    android:textSize="18sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id/ButtonOk"
    app:layout_constraintTop_toBottomOf="@+id/SeekBar" />

<SeekBar
    android:id="@+id/SeekBar"
    android:layout_width="match_parent"
    android:layout_height="20dp"
    android:layout_marginBottom="55dp"
    android:max="95"
    android:progress="0"
    app:layout_constraintBottom_toTopOf="@+id/ButtonDisconnect"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/SeekBarNumber" />

<TextView
    android:id="@+id/SeekBarNumber"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="25dp"
    android:background="#003F51B5"
    android:text="0"
    android:textColor="FFFFFF"
    android:textSize="25sp"
    app:layout_constraintBottom_toTopOf="@+id/SeekBar"
    app:layout_constraintEnd_toEndOf="@+id/switch1"
    app:layout_constraintStart_toStartOf="@+id/switch1"
    app:layout_constraintTop_toBottomOf="@+id/switch1" />

<Button
    android:id="@+id/ButtonOk"
    android:layout_width="wrap_content"

```

```

        android:layout_height="wrap_content"
        android:layout_marginStart="24dp"
        android:layout_marginLeft="24dp"
        android:layout_marginEnd="2dp"
        android:layout_marginRight="2dp"
        android:background="@drawable/ok_button"
        android:text="OK"
        android:textColor="#FBFBFB"
        android:textSize="18sp"
        android:textStyle="bold"
        app:layout_constraintBaseline_toBaselineOf="@+id/ButtonDisconnect"
        app:layout_constraintEnd_toStartOf="@+id/ButtonDisconnect"
        app:layout_constraintStart_toStartOf="parent" />

<Switch
    android:id="@+id/switch1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="149dp"
    android:layout_marginBottom="34dp"
    android:showText="true"
    android:text="POWER"
    android:textColor="#FFFFFF"
    android:textOff="OFF"
    android:textOn="ON"
    android:textSize="20dp"
    android:textStyle="bold"
    android:thumb="@drawable/switch_thumb"
    android:track="@drawable/switch_track"
    app:layout_constraintBottom_toTopOf="@+id/SeekBarNumber"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

Penggalan program diatas menjelaskan pengaturan tampilan pada halaman main activity. *Button* berfungsi untuk memberikan perintah pada *Arduino Uno* dengan program yang telah ditentukan. *SeekBar* berfungsi sebagai pengatur kecerahan lampu sesuai dengan pengguna. Berikut ini adalah penulisan kode MainActivity.java.

```

package com.example.bluetooth;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.os.AsyncTask;

```

```

import android.bluetooth.BluetoothSocket;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.view.View;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.CompoundButton;
import android.widget.Switch;
import android.widget.Toast;
import android.widget.SeekBar;
import android.widget.TextView;
import java.io.IOException;
import java.util.UUID;

public class MainActivity extends AppCompatActivity
{
    Button vButtonDisconnect, vButtonOk;
    Switch powerSwitch;
    BluetoothAdapter vBluetoothAdapter;
    BluetoothSocket vBluetoothSocket;
    BluetoothDevice vBluetoothDevice;
    SeekBar Brightness;
    TextView BrightnessNumber;
    int cerah, power;
    String address2 = null;
    static final UUID myUUID = UUID.fromString("00001101-0000-1000-8000-00805F9B34FB");

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);

        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.Lay
outParams.FLAG_FULLSCREEN);
        setContentView(R.layout.activity_main);

        address2 = SearchActivity.getVariable();

        setTitle(address2);

        powerSwitch = findViewById(R.id.switch1);
        vButtonDisconnect = findViewById(R.id.ButtonDisconnect);
        vButtonOk = findViewById(R.id.ButtonOk);
        Brightness = findViewById(R.id.SeekBar);
        BrightnessNumber = findViewById(R.id.SeekBarNumber);

        new ConnectBT().execute();

        powerSwitch.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener()
        {
            @Override
            public void onCheckedChanged(CompoundButton buttonView, boolean isChecked)
            {

```

```

        if (isChecked)
        {
            power = 1;
            if (vBluetoothSocket != null)
            {
                try
                {
                    vBluetoothSocket.getOutputStream().write("ON".toString().getBytes());
                    showMessage("ON");
                }

                catch (IOException e)
                {
                    e.printStackTrace();
                }
            }
        }
        else
        {
            power = 0;
        }
    }
});

vButtonDisconnect.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view)
    {
        mDisconnect();
    }
});

vButtonOk.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view)
    {
        if (power == 1)
        {
            if (vBluetoothSocket != null)
            {
                if (cerah == 0)
                {
                    try
                    {
                        vBluetoothSocket.getOutputStream().write("OFF".toString().getBytes());
                        showMessage("OFF");
                    }

                    catch (IOException e)
                    {
                        e.printStackTrace();
                    }
                }
            }
        }
    }
});

```

```

    }
}

else
{
    try
    {
        vBluetoothSocket.getOutputStream().write(String.valueOf(cerah).getBytes());
        showMessage(String.valueOf(cerah));
    }

    catch (IOException e)
    {
        e.printStackTrace();
    }
}

}

else
{
    if (vBluetoothSocket != null)
    {
        try
        {
            vBluetoothSocket.getOutputStream().write("OFF".toString().getBytes());
            showMessage("OFF");
        }

        catch (IOException e)
        {
            e.printStackTrace();
        }
    }
}

});

Brightness.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener()
{
    @Override
    public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser)
    {
        if (fromUser)
        {
            cerah = progress;
            BrightnessNumber.setText(""+cerah);
        }
    }

    @Override
    public void onStartTrackingTouch(SeekBar seekBar)
    {

```

```

    }

    @Override
    public void onStopTrackingTouch(SeekBar seekBar)
    {

    }

    });
}

@Override
public void onBackPressed()
{
    super.onBackPressed();
    try
    {
        vBluetoothSocket.close();
    }

    catch (IOException e)
    {
        e.printStackTrace();
    }
}

public void mDisconnect()
{
    if (vBluetoothSocket != null)
    {
        try
        {
            vBluetoothSocket.close();
            showMessage("Disconnect "+address2);
        }

        catch (IOException e)
        {
            e.printStackTrace();
        }
    }
}

public void showMessage(String msg)
{
    Toast.makeText(this, msg, Toast.LENGTH_SHORT).show();
}

private class ConnectBT extends AsyncTask<String, String, String>
{
    @Override
    protected String doInBackground (String... device)
    {

```



```

        if (vBluetoothSocket == null)
        {
            try
            {
                vBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
                vBluetoothDevice = vBluetoothAdapter.getRemoteDevice(address2);
                vBluetoothSocket = vBluetoothDevice.createRfcommSocketToServiceRecord(myUUID);
                vBluetoothAdapter.cancelDiscovery();
                vBluetoothSocket.connect();
            }

            catch (IOException e)
            {
                e.printStackTrace();
            }
        }
        return null;
    }

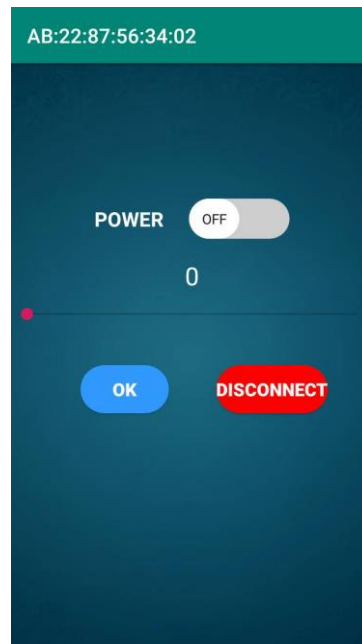
    @Override
    protected void onPostExecute(final String result)
    {
        if (result != null)
        {
            showMessage("fail to connect");
        }

        else
        {
            showMessage("connected to "+address2);
        }
    }

    @Override
    protected void onCancelled()
    {
    }
}

```

Penggalan program diatas menjelaskan variabel address2 berfungsi untuk menerima alamat *Bluetooth* dari halaman search device. Terdapat AsyncTask bernama ConnectBT yang berfungsi untuk menghubungkan koneksi *Bluetooth* dibelakang layar (*background*). Setelah *Bluetooth* terhubung pengguna dapat mengatur kecerahan lampu dengan bantuan *SeekBar* dan *Switch*. Perintah *getOutputStream* digunakan untuk mengirimkan data melalui jaringan *Bluetooth*. Berikut ini adalah tampilan dari main activity.



**Gambar 3.11** Tampilan Halaman Main Activity

### 3.2.2.5 AndroidManifest.xml

*Android Manifest* berfungsi untuk menghubungkan 2 file, yaitu .xml dan .java. Jika tidak dihubungkan maka program tidak akan berjalan. Selain itu, *Android Manifest* berisi *permission* untuk mendapatkan izin dari pengguna sebelum memakai aplikasi. Berikut ini adalah penulisan kode *Android Manifest*.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.bluetooth">

    <uses-permission android:name="android.permission.BLUETOOTH" />
    <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.BLUETOOTH_PRIVILEGED" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/logo"
        android:label="SMART LED"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MulaiActivity"
            android:theme="@style/AppTheme.NoActionBar"></activity>
```

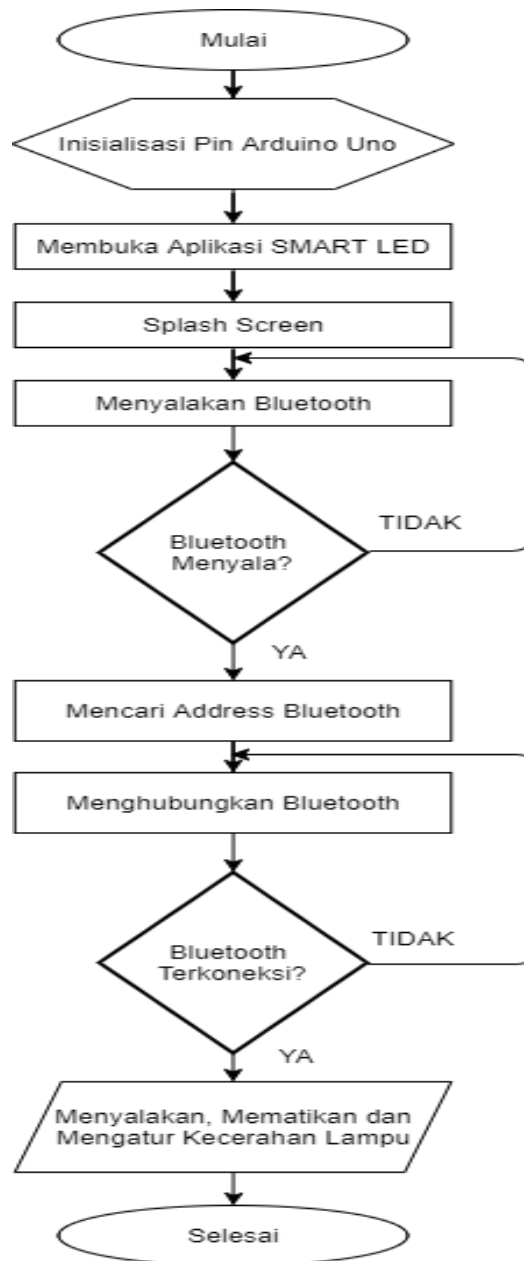
```
<activity android:name=".MainActivity"></activity>
<activity android:name=".SearchActivity"></activity>
<activity android:name=".splash_screen" android:theme="@style/AppTheme.NoActionBar">
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />

    <category android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>

<meta-data
  android:name="preloaded_fonts"
  android:resource="@array/preloaded_fonts" />
</application>
</manifest>
```

Pada penggalan kode diatas aplikasi ini membutuhkan bluetooth permission seperti pada line 3 – 7. Pengguna diharuskan mengizinkan penggunaan bluetooth pada aplikasi ini

### 3.3 Flowchart Program



**Gambar 3.12** Flowchart Program

### Penjelasan flowchart program

Program dimulai dengan menginisialisasi tiap-tiap pin yang digunakan oleh setiap komponen pada *Arduino Uno*. Selanjutnya pengguna membuka aplikasi *SMART LED*. Saat pertama kali membuka aplikasi terdapat *Splash Screen* selama 5 detik dan menuju ke halaman mulai. Pada halaman mulai pengguna diharuskan untuk menghidupkan *Bluetooth*. Jika *Bluetooth* dimatikan maka program akan meminta akses kembali untuk menyalakan *Bluetooth*. Jika *Bluetooth* sudah menyala akan menuju ke halaman *Search Device*. Pada halaman *Search Device* terdapat *button* untuk mencari *Bluetooth Address* dan menghubungkannya. Jika tidak terhubung maka pengguna diharuskan untuk menghubungkan kembali. Jika terhubung akan menuju ke halaman *Main Activity*. Pada halaman *Main Activity* pengguna bisa menyalakan, mematikan dan mengatur kecerahan lampu.

### 3.4 Langkah pembuatan Aplikasi

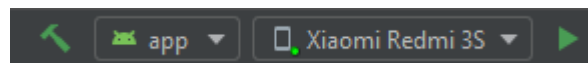
Dalam pembuatan aplikasi ini membutuhkan perangkat seperti *Arduino Uno*, *Android Smartphone*, *Arduino IDE* dan *Android Studio*.

1. Hubungkan *smartphone* dengan komputer melalui kabel USB.



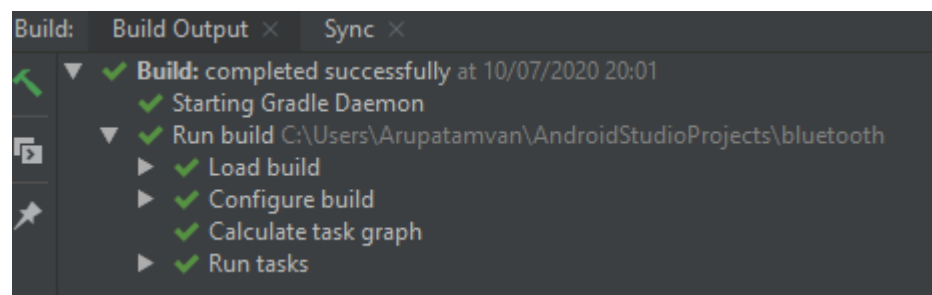
**Gambar 3.13** Menghubungkan Smartphone dengan Komputer

2. Buka aplikasi *Android Studio* dan pilih perangkat *Xiaomi Redmi 3S* lalu tekan tombol *run*.



**Gambar 3.14** Memilih Perangkat Smartphone

3. Tunggu hingga proses *build* selesai dan aplikasi sudah siap untuk digunakan.



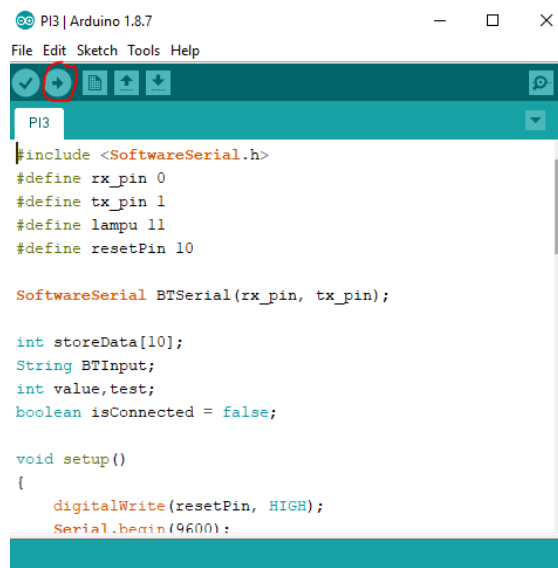
**Gambar 3.15** Tampilan Build

4. Lepaskan kabel *Smartphone* dan hubungkan *Arduino Uno* dengan komputer melalui kabel USB.



**Gambar 3.16** Menghubungkan Arduino Uno dengan Komputer

5. Buka aplikasi *Arduino IDE*, tekan tombol upload, tunggu proses *uploading* selesai dan *Arduino Uno* sudah siap untuk digunakan.



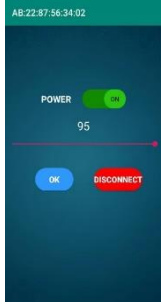





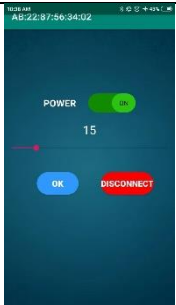


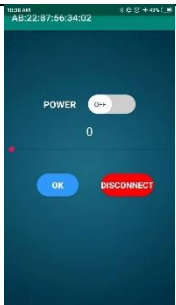


**Gambar 3.17** Tombol Upload pada Arduino IDE

6. Rangkailah *Arduino Uno* seperti pada gambar 3.2 dan tabel 3.1.

### 3.5 Uji Coba

Setelah proses pembuatan aplikasi langkah selanjutnya adalah uji coba. Pada uji coba ini penulis menggunakan *Android Smartphone Xiaomi Redmi 3S*, *Arduino Uno*, lampu bohlam dan *multimeter* digital untuk menguji apakah aplikasi berjalan dengan benar. Kondisi lampu ruangan saat uji coba berlangsung adalah menyala dan jarak 1 meter. Berikut ini adalah tabel hasil pengujian.

**Tabel 3.2** Tabel Hasil Pengujian

| No | Tampilan Android  | Lampu Bohlam  | Tegangan Lampu Bohlam  | Intensitas Cahaya |
|----|---|---|--|-------------------|
| 1. |    |    |    | Tinggi            |
| 2. |   |   |   | Sedang            |
| 3. |  |  |  | Rendah            |
| 4. |  |  |  | Tidak Ada         |



## BAB IV

### PENUTUP

#### 4.1 Kesimpulan

Berdasarkan hasil penelitian yang dilakukan, perangkat ini dapat mengendalikan lampu melalui *android smartphone* dan *arduino uno* yang terintegrasi dengan jaringan *bluetooth* dengan jarak 1 meter.

#### 4.2 Saran

Untuk penelitian selanjutnya disarankan lampu yang digunakan tidak hanya bersifat *dimmerable*. Namun, lampu yang sering digunakan seperti lampu LED. Kemudian, aplikasi dapat berjalan diberbagai jenis sistem operasi lainnya seperti *IOS* dan *Windows*. Selain itu, perangkat kendali lampu dapat ditambahkan dengan modul *wireless* agar pengguna dapat mengendalikan melalui jaringan internet dimana saja dan kapan saja. Hal tersebut dapat dijadikan acuan pengembangan dan penyempurnaan dari aplikasi ini agar lebih baik lagi.

## DAFTAR PUSTAKA

- [1] B, Gustomo. 2015. "Pengenalannya Arduino dan Pemrogramannya". Bandung : Informatika Bandung.
- [2] Setiawardana.dkk.2019."19 Jam Belajar Cepat Arduino Edisi Revisi".Jakarta : Bumi Aksara.
- [3] Supardi, Yuniar.2011."Semua Bisa Menjadi Programmer Android".Jakarta : PT. Elex Media.
- [4] Wright, Michael.2016."HC-05 Bluetooth + Arduino: Includes the ZS-040".Create Space Independent Publishing Platform.
- [5] Yuswanto.2011."Pemrograman Android Untuk Pemula". Jakarta : Pembuka Cakrawala.
- [6] Hariyanto, Bambang, Sistem Manajemen Basis Data: Pemodelan, Perancangan, dan Terapannya, Informatika, Bandung 2004.
- [7] Anonim."Pengertian Arduino Uno".<https://ilearning.me/sample-page-162/arduino/pengertian-arduino-uno/>.Diakses tanggal 02 Juni 2020 pukul 00:12.
- [8] Anonim."HC-05 Bluetooth to Serial Port Module".[https://www.robotshop.com/media/files/pdf/rb-ite-12-bluetooth\\_hc05.pdf](https://www.robotshop.com/media/files/pdf/rb-ite-12-bluetooth_hc05.pdf).Diakses tanggal 07 Juni 2020 pukul 18:32.
- [9] Anonim."AC Light Dimmer Module, 1 Channel, 3.3V\_5V Logic, AC 50\_60Hz, 220V\_110V".[https://github.com/RobotDynOfficial/Documentation/wiki/AC-Light-Dimmer-Module,-1-Channel,-3.3V\\_5V-logic,-AC-50\\_60hz,-220V\\_110V](https://github.com/RobotDynOfficial/Documentation/wiki/AC-Light-Dimmer-Module,-1-Channel,-3.3V_5V-logic,-AC-50_60hz,-220V_110V).Diakses tanggal 07 Juni 2020 pukul 20:30.
- [10] Anonim."BTA16-600B".<http://download.maritex.com.pl/pdfs/sc/BTA16-600B.pdf>.Diakses tanggal 09 Juni 2020 pukul 10:30.
- [11] Kho, Dickson."Pengertian TRIAC dan Aplikasinya".<https://teknikelektronika.com/pengertian-triac-dan-aplikasi-triac-thyristor/#:~:text=Pengertian%20TRIAC%20dan%20Aplikasinya%20%E2%8>

- 0%93%20TRIAC,Trioda%20untuk%20arus%20bolak%20balik).Diakses tanggal 08 Juni 2020 pukul 12:08.
- [12] Pamungkas, Kurnia Agung.dkk.”Aplikasi Android dan Mikrokontroller Arduino pada Kontrol Smarthome dengan Komunikasi Bluetooth”.2016.<https://nero.trunojoyo.ac.id/index.php/nero/article/download/66/68>.Diakses tanggal 01 Juni 2020 pukul 14:32.
- [13] Szeffy, Tamas.”What is Difference Between an Arduino Uno and Arduino Uno R3”.2014.<https://www.quora.com/What-is-the-difference-between-an-Arduino-Uno-and-an-Arduino-Uno-R3>.Diakses tanggal 05 Juni 2020 pukul 11:57.
- [14] Anonim. “The Arduino Popularity Contest”.2015.  
<https://www.sparkfun.com/news/1982>. Diakses tanggal 10 Agustus 2020 pukul 20:00.
- [15] Riyanto, Andi Dwi. “Indonesian Digital Report 2019”.2019.  
<https://andi.link/hootsuite-we-are-social-indonesian-digital-report-2019/>. Diakses tanggal 10 Agustus 2020 pukul 20:30.
- [16] Anonim. “Mobile Operating System Market Share Indonesia”.2020.  
<https://gs.statcounter.com/os-market-share/mobile/indonesia>. Diakses tanggal 11 Agustus 2020 pukul 09:30.
- [17] Kumar, Chandan. “Bluetooth Controlled Light Using Arduino with Single Relay”. <https://create.arduino.cc/projecthub/chandankumarcmsn/bluetooth-controlled-light-using-arduino-with-single-relay-09bb68>. Diakses tanggal 11 Agustus 2020 pukul 10:30.
- [18] Anonim. “Library RBD Dimmer”. <https://github.com/RobotDynOfficial/Lib-RBD-Dimmer-for-Mega-UNO-Leonardo>.Diakses tanggal 15 Agustus 2020 pukul 20:00.
- [19] Anonim. “LAMPU LED, BOHLAM, NEON : PERBEDAAN, PENGGUNAAN, KELEBIHAN SERTA KEKURANGAN”.  
<https://binus.ac.id/malang/2020/06/lampu-led-bohlam-neon-perbedaan-penggunaan-kelebihan-serta-kekurangan/>. Diakses tanggal 15 Agustus 2020 pukul 20:20.

- [20] Heri. “SIMBOL FLOWCHART : Pengertian, Jenis, Fungsi dan Contohnya”.2017. <https://salamadian.com/simbol-simbol-flowchart/>. Diakses tanggal 11 Januari 2021 pukul 20:30.
- [21] Anonim. “Pengertian dan Sejarah Perkembangan Bahasa Pemrograman Java”. 2019.<https://medium.com/@seotog08/pengertian-dan-sejarah-perkembangan-bahasa-pemrograman-java-204f24d661e3>. Diakses tanggal 28 Februari 2021 pukul 04:00.
- [22] Andre. “Pengertian Bahasa C”.2018.<https://www.duniailkom.com/tutorial-belajar-c-pengertian-bahasa-pemrograman-c/>. Diakses tanggal 28 Februari 2021 pukul 04:05.

# LAMPIRAN LISTING PROGRAM

## PI2.ino

```
#include <RBDdimmer.h>

#include <SoftwareSerial.h>

#define rx_pin 0

#define tx_pin 1

#define pwm 3

#define zerocross 2

#define resetPin 9


SoftwareSerial BTSerial(rx_pin, tx_pin);

dimmerLamp lampu(pwm);


String BTInput;

int waiting = 1000;

int minBright = 0;

int maxBright = 95;

int value, counter;

int storeData[300];

boolean isPowered = false;


void setup()

{

    digitalWrite(resetPin, HIGH);

    BTSerial.begin(9600);

    pinMode(resetPin,OUTPUT);

    lampu.begin(NORMAL_MODE,ON);

}


void loop()

{
```

```
arrayCheck();

if (BTSerial.available())

{

    BTInput = BTSerial.readString();

    powerCheck();

    if (value <= maxBright && isPowered == false)

    {

        delay(waiting);

        lampu.setPower(minBright);

    }

    else if (value > minBright && isPowered == true)

    {

        delay(waiting);

        lampu.setPower(value);

    }

}

else if (value == minBright)

{

    delay(waiting);

    lampu.setPower(storeData[counter-1]);

}

}

void powerCheck()

{

    if (BTInput.equals("ON"))

    {

        isPowered = true;

    }

}
```

```
else if (BTInput.equals("OFF"))  
  
    {  
  
        isPowered = false;  
  
    }  
  
else  
  
    {  
  
        value = BTInput.toInt();  
  
        counter = counter + 1;  
  
        storeData[counter] = value;  
  
    }  
}
```

```
void arrayCheck()  
  
{  
  
    if (counter == 295)  
  
    {  
  
        digitalWrite(resetPin, LOW);  
  
    }  
}
```

## Splash\_screen.java

```
package com.example.bluetooth;  
  
  
  
import androidx.appcompat.app.AppCompatActivity;  
  
  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.os.Handler;  
import android.view.WindowManager;  
import android.view.animation.Animation;  
import android.view.animation.AnimationUtils;
```

```

import android.widget.ImageView;

import android.widget.TextView;


public class splash_screen extends AppCompatActivity
{

    private static int splash_time = 5000;


    Animation topAnim,botAnim;

    ImageView gambar;

    TextView first,second;


    @Override

    protected void onCreate(Bundle savedInstanceState)
    {

        super.onCreate(savedInstanceState);

        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,WindowManager.LayoutParams.FLAG_FULLSCREEN);

        setContentView(R.layout.activity_splash_screen);


        topAnim = AnimationUtils.loadAnimation(this,R.anim.top_animation);

        botAnim = AnimationUtils.loadAnimation(this,R.anim.bottom_animation);


        gambar = findViewById(R.id.imageView);

        first = findViewById(R.id.textView);

        second = findViewById(R.id.textView2);


        gambar.setAnimation(topAnim);

        first.setAnimation(botAnim);

        second.setAnimation(botAnim);


        new Handler().postDelayed(new Runnable()

        {

            @Override

```



```
        public void run()
        {
            Intent intent = new Intent(splash_screen.this, MulaiActivity.class);

            startActivity(intent);

            finish();

        }

    },splash_time);

}

}
```

## MulaiActivity.java

```
package com.example.bluetooth;

import androidx.appcompat.app.AppCompatActivity;

import android.bluetooth.BluetoothAdapter;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.os.Bundle;
import android.view.View;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.Toast;
import java.lang.String;

public class MulaiActivity extends AppCompatActivity
{

    Button xMulai;

    BluetoothAdapter mBlueAdapter;
```

```

@Override

protected void onCreate(Bundle savedInstanceState)

{

    super.onCreate(savedInstanceState);

getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,WindowManager.LayoutParams.FLAG_FULLSCREEN);

    setContentView(R.layout.activity_mulai);

    xMulai = findViewById(R.id.Mulai);

    mBlueAdapter = BluetoothAdapter.getDefaultAdapter();

    xMulai.setOnClickListener(new View.OnClickListener()

    {

        @Override

        public void onClick(View view)

        {

            enableBT();

        }

    });

}

public void enableBT()

{

    if (!mBlueAdapter.isEnabled())

    {

        Intent enableBTIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);

        startActivity(enableBTIntent);

        IntentFilter BTIntent = new IntentFilter(BluetoothAdapter.ACTION_STATE_CHANGED);

        registerReceiver(mBroadcastReceiver1, BTIntent);

    }

}

```

```
public BroadcastReceiver mBroadcastReceiver1 = new BroadcastReceiver()

{

    @Override

    public void onReceive(Context context, Intent intent)

    {

        String action = intent.getAction();

        if (action.equals(mBlueAdapter.ACTION_STATE_CHANGED))

        {

            final int state = intent.getIntExtra(BluetoothAdapter.EXTRA_STATE, mBlueAdapter.ERROR);

            switch(state)

            {

                case BluetoothAdapter.STATE_TURNING_ON:

                    showMessage("Activating Bluetooth...");

                    break;

                case BluetoothAdapter.STATE_ON:

                    showMessage("Bluetooth Activated!");

                    nextActivity();

                    break;

            }

        }

    }

};

public void nextActivity()

{

    Intent aw = new Intent(this, SearchActivity.class);

    startActivity(aw);

    overridePendingTransition(R.anim.slide_in_right, R.anim.slide_out_left);

}
```

```
public void showMessage(String msg)

{

    Toast.makeText(this, msg, Toast.LENGTH_SHORT).show();

}

}
```

## SearchActivity.java

```
package com.example.bluetooth;

import androidx.appcompat.app.AppCompatActivity;

import android.Manifest;

import android.content.BroadcastReceiver;

import android.content.Context;

import android.content.Intent;

import android.content.IntentFilter;

import android.os.Build;

import android.os.Bundle;

import android.bluetooth.BluetoothAdapter;

import android.view.WindowManager;

import android.widget.ArrayAdapter;

import android.widget.Button;

import android.widget.ListView;

import android.view.View;

import android.bluetooth.BluetoothDevice;

import java.util.ArrayList;

import java.lang.String;

import android.widget.TextView;

import android.widget.Toast;

import android.widget.AdapterView;

public class SearchActivity extends AppCompatActivity

{
```

```

Button vSearchDevices;

ListView vListDevices;

ArrayList<String> vArrayList = new ArrayList<String>();

ArrayAdapter<String> vArrayAdapter;

BluetoothAdapter vBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();

private static String address = "0";

@Override

protected void onCreate(Bundle savedInstanceState)

{

    super.onCreate(savedInstanceState);

    getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,WindowManager.LayoutParams.FLAG_FULLSCREEN);

    setContentView(R.layout.activity_search);

    setTitle("Device List");

    vSearchDevices = findViewById(R.id.search);

    vListDevices = findViewById(R.id.DevicesList);

    vSearchDevices.setOnClickListener(new View.OnClickListener()

    {

        @Override

        public void onClick(View view)

        {

            BTSearch();

        }

    });

}

public void BTSearch()

{

    if (vBluetoothAdapter.isDiscovering())

```

```

{

    vBluetoothAdapter.cancelDiscovery();

    checkBT();

    vBluetoothAdapter.startDiscovery();

    IntentFilter discoverDevices = new IntentFilter(BluetoothDevice.ACTION_FOUND);

    registerReceiver(vBroadcastReceiver3, discoverDevices);

}

if (!vBluetoothAdapter.isDiscovering())

{

    checkBT();

    vBluetoothAdapter.startDiscovery();

    IntentFilter discoverDevices = new IntentFilter(BluetoothDevice.ACTION_FOUND);

    registerReceiver(vBroadcastReceiver3, discoverDevices);

}

}

public BroadcastReceiver vBroadcastReceiver3 = new BroadcastReceiver()

{

    @Override

    public void onReceive(Context context, Intent intent)

    {

        final String action = intent.getAction();

        if (action.equals(BluetoothDevice.ACTION_FOUND))

        {

            BluetoothDevice device = intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);

            String devicename = device.getName();

            String macaddress = device.getAddress();

            vArrayList.add("Name : "+devicename+"MAC Address : "+macaddress);

```

```

        vArrayAdapter = new ArrayAdapter<String>(getApplicationContext(), android.R.layout.simple_list_item_1, vArrayList);

        vListDevices.setAdapter(vArrayAdapter);

        vListDevices.setOnItemClickListener(vClickListDevices);

    }

}

};

```

```

public final AdapterView.OnItemClickListener vClickListDevices = new AdapterView.OnItemClickListener()

{

    @Override

    public void onItemClick(AdapterView<?> parent, View view, int position, long id)

    {

        String info = ((TextView) view).getText().toString();

        address = info.substring(info.length() - 17);

        NextActivity();

    }

};

```

```

public void NextActivity()

{

    Intent aw = new Intent(SearchActivity.this, MainActivity.class);

    startActivity(aw);

    overridePendingTransition(R.anim.slide_in_right, R.anim.slide_out_left);

}

```

```

public static String getVariable()

{

    return address; //oper variabel

}

```

```

public void checkBT()

{

```

```

        if (Build.VERSION.SDK_INT > Build.VERSION_CODES.LOLLIPOP)
        {

            int permissionCheck = this.checkSelfPermission("Manifest.permission.ACCESS_FINE_LOCATION");

            permissionCheck += this.checkSelfPermission("Manifest.permission.ACCESS_COARSE_LOCATION");

            if (permissionCheck != 0)

            {

                this.requestPermissions(new String[]{Manifest.permission.ACCESS_FINE_LOCATION,
Manifest.permission.ACCESS_COARSE_LOCATION}, 1001);

            }

            else

            {

                showMessage("no need to check permission. SDK version < lolipop");

            }

        }

    }

    public void showMessage(String msg)

    {

        Toast.makeText(this, msg, Toast.LENGTH_SHORT).show();

    }

}

```

## MainActivity.java

```

package com.example.bluetooth;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.os.AsyncTask;

import android.bluetooth.BluetoothSocket;

import android.bluetooth.BluetoothAdapter;

import android.bluetooth.BluetoothDevice;

```



```
import android.view.View;

import android.view.WindowManager;

import android.widget.Button;

import android.widget.CompoundButton;

import android.widget.Switch;

import android.widget.Toast;

import android.widget.SeekBar;

import android.widget.TextView;

import java.io.IOException;

import java.util.UUID;

public class MainActivity extends AppCompatActivity

{

    Button vButtonDisconnect, vButtonOk;

    Switch powerSwitch;

    BluetoothAdapter vBluetoothAdapter;

    BluetoothSocket vBluetoothSocket;

    BluetoothDevice vBluetoothDevice;

    SeekBar Brightness;

    TextView BrightnessNumber;

    int cerah, power;

    String address2 = null;

    static final UUID myUUID = UUID.fromString("00001101-0000-1000-8000-00805F9B34FB");

    @Override

    protected void onCreate(Bundle savedInstanceState)

    {

        super.onCreate(savedInstanceState);

        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.LayoutParams.FLAG_FULLSCREEN);

        setContentView(R.layout.activity_main);

        address2 = SearchActivity.getVariable();
```

```
setTitle(address2);
```

```
powerSwitch = findViewById(R.id.switch1);
```

```
vButtonDisconnect = findViewById(R.id.ButtonDisconnect);
```

```
vButtonOk = findViewById(R.id.ButtonOk);
```

```
Brightness = findViewById(R.id.SeekBar);
```

```
BrightnessNumber = findViewById(R.id.SeekBarNumber);
```

```
new ConnectBT().execute();
```

```
powerSwitch.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener()
```

```
{
```

```
    @Override
```

```
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked)
```

```
    {
```

```
        if (isChecked)
```

```
        {
```

```
            power = 1;
```

```
            if (vBluetoothSocket != null)
```

```
            {
```

```
                try
```

```
                {
```

```
                    vBluetoothSocket.getOutputStream().write("ON".toString().getBytes());
```

```
                    showMessage("ON");
```

```
                }
```

```
            catch (IOException e)
```

```
            {
```

```
                e.printStackTrace();
```

```
            }
```

```
        }
```

```

    }

    else

    {

        power = 0;

    }

}

});

```

```

vButtonDisconnect.setOnClickListener(new View.OnClickListener()

```

```

{

    @Override

    public void onClick(View view)

    {

        mDisconnect();

    }

});

```

```

vButtonOk.setOnClickListener(new View.OnClickListener()

```

```

{

    @Override

    public void onClick(View view)

    {

        if (power == 1)

        {

            if (vBluetoothSocket != null)

            {

                if (cerah == 0)

                {

                    try

                    {

                        vBluetoothSocket.getOutputStream().write("OFF".getBytes());

                        showMessage("OFF");

                    }

                }

            }

        }

    }

});

```

```
    }

    catch (IOException e)

    {

        e.printStackTrace();

    }

}

else

{

    try

    {

        vBluetoothSocket.getOutputStream().write(String.valueOf(cerah).getBytes());

        showMessage(String.valueOf(cerah));

    }

    catch (IOException e)

    {

        e.printStackTrace();

    }

}

}

else

{

    if (vBluetoothSocket != null)

    {

        try

        {

            vBluetoothSocket.getOutputStream().write("OFF".toString().getBytes());

            showMessage("OFF");
```

```

    }

    catch (IOException e)

    {

        e.printStackTrace();

    }

    }

    }

    });

Brightness.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener()

{

    @Override

    public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser)

    {

        if (fromUser)

        {

            cerah = progress;

            BrightnessNumber.setText(""+cerah);

        }

    }

    @Override

    public void onStartTrackingTouch(SeekBar seekBar)

    {

    }

    @Override

    public void onStopTrackingTouch(SeekBar seekBar)

    {

```

```
    }

});

}

@Override

public void onBackPressed()

{

    super.onBackPressed();

    try

    {

        vBluetoothSocket.close();

    }

    catch (IOException e)

    {

        e.printStackTrace();

    }

}

public void mDisconnect()

{

    if (vBluetoothSocket != null)

    {

        try

        {

            vBluetoothSocket.close();

            showMessage("Disconnect "+address2);

        }

        catch (IOException e)

        {
```

```
        e.printStackTrace();
    }
}

}

public void showMessage(String msg)
{
    Toast.makeText(this, msg, Toast.LENGTH_SHORT).show();
}

private class ConnectBT extends AsyncTask<String, String, String>
{
    @Override
    protected String doInBackground (String... device)
    {
        if (vBluetoothSocket == null)
        {
            try
            {
                vBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
                vBluetoothDevice = vBluetoothAdapter.getRemoteDevice(address2);
                vBluetoothSocket = vBluetoothDevice.createRfcommSocketToServiceRecord(myUUID);
                vBluetoothAdapter.cancelDiscovery();
                vBluetoothSocket.connect();
            }

            catch (IOException e)
            {
                e.printStackTrace();
            }
        }
        return null;
    }
}
```

```

    }

    @Override

    protected void onPostExecute(final String result)

    {

        if (result != null)

        {

            showMessage("fail to connect");

        }

        else

        {

            showMessage("connected to "+address2);

        }

    }

    @Override

    protected void onCancelled()

    {

    }

}

```

## Activity\_splash\_screen.xml

```

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:background="@drawable/background"

    tools:context=".splash_screen">

```



```
<ImageView

    android:id="@+id/imageView"

    android:layout_width="291dp"

    android:layout_height="296dp"

    android:src="@drawable/logo"

    app:layout_constraintBottom_toTopOf="@+id/textView"

    app:layout_constraintEnd_toEndOf="parent"

    app:layout_constraintStart_toStartOf="parent"

    app:layout_constraintTop_toTopOf="parent"

    app:layout_constraintVertical_bias="0.616" />
```

```
<TextView

    android:id="@+id/textView"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:text="SMART LED"

    android:textColor="#FFFFFF"

    android:textSize="35sp"

    android:textStyle="bold"

    app:fontFamily="@font/cherry_cream_soda"

    app:layout_constraintBottom_toBottomOf="parent"

    app:layout_constraintEnd_toEndOf="parent"

    app:layout_constraintStart_toStartOf="parent"

    app:layout_constraintTop_toTopOf="parent"

    app:layout_constraintVertical_bias="0.628" />
```

```
<TextView

    android:id="@+id/textView2"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:text="Simple application to control your lamp"
```

```

        android:textColor="#FFFFFF"

        android:textSize="20sp"

        android:textStyle="italic"

        app:fontFamily="@font/assistant_extralight"

        app:layout_constraintBottom_toBottomOf="parent"

        app:layout_constraintEnd_toEndOf="parent"

        app:layout_constraintHorizontal_bias="0.494"

        app:layout_constraintStart_toStartOf="parent"

        app:layout_constraintTop_toTopOf="parent"

        app:layout_constraintVertical_bias="0.714" />

```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

## Activity\_mulai.xml

```

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:background="@drawable/background"

    tools:context=".MulaiActivity">

    <ImageView

        android:id="@+id/imageView"

        android:layout_width="291dp"

        android:layout_height="296dp"

        android:src="@drawable/logo"

        app:layout_constraintBottom_toTopOf="@+id/textView"

        app:layout_constraintEnd_toEndOf="parent"

        app:layout_constraintStart_toStartOf="parent"

        app:layout_constraintTop_toTopOf="parent"

        app:layout_constraintVertical_bias="0.616" />

```

```
<TextView

    android:id="@+id/textView"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:text="SMART LED"

    android:textColor="#FFFFFF"

    android:textSize="35sp"

    android:textStyle="bold"

    app:fontFamily="@font/cherry_cream_soda"

    app:layout_constraintBottom_toBottomOf="parent"

    app:layout_constraintEnd_toEndOf="parent"

    app:layout_constraintStart_toStartOf="parent"

    app:layout_constraintTop_toTopOf="parent"

    app:layout_constraintVertical_bias="0.628" />
```

```
<TextView

    android:id="@+id/textView2"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:text="Simple application to control your lamp"

    android:textColor="#FFFFFF"

    android:textSize="20sp"

    android:textStyle="italic"

    app:fontFamily="@font/assistant_extralight"

    app:layout_constraintBottom_toBottomOf="parent"

    app:layout_constraintEnd_toEndOf="parent"

    app:layout_constraintHorizontal_bias="0.494"

    app:layout_constraintStart_toStartOf="parent"

    app:layout_constraintTop_toTopOf="parent"

    app:layout_constraintVertical_bias="0.714" />
```

```

<Button

    android:id="@+id/Mulai"

    android:layout_width="70dp"

    android:layout_height="70dp"

    android:background="@drawable/go_button"

    android:text="GO"

    android:textColor="#FBFBFB"

    android:textSize="20dp"

    android:textStyle="bold"

    app:layout_constraintBottom_toBottomOf="parent"

    app:layout_constraintEnd_toEndOf="parent"

    app:layout_constraintStart_toStartOf="parent"

    app:layout_constraintTop_toTopOf="parent"

    app:layout_constraintVertical_bias="0.857" />

```

```

</androidx.constraintlayout.widget.ConstraintLayout>

```

## Activity\_search.xml

```

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:background="@drawable/background"

    tools:context=".SearchActivity">

    <Button

        android:id="@+id/search"

        android:layout_width="80dp"

        android:layout_height="80dp"

        android:layout_marginEnd="40dp"

        android:layout_marginRight="40dp"

```

```

        android:layout_marginBottom="50dp"

        android:background="@drawable/search_button"

        app:layout_constraintBottom_toBottomOf="parent"

        app:layout_constraintEnd_toEndOf="parent"

        app:layout_constraintTop_toBottomOf="@+id/DevicesList" />

```

```

<ListView

        android:id="@+id/DevicesList"

        android:layout_width="0dp"

        android:layout_height="0dp"

        android:layout_marginTop="85dp"

        android:layout_marginBottom="183dp"

        app:layout_constraintBottom_toTopOf="@+id/search"

        app:layout_constraintEnd_toEndOf="parent"

        app:layout_constraintStart_toStartOf="parent"

        app:layout_constraintTop_toTopOf="parent" />

```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

## Activity\_main.xml

```

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

        xmlns:app="http://schemas.android.com/apk/res-auto"

        xmlns:tools="http://schemas.android.com/tools"

        android:layout_width="match_parent"

        android:layout_height="match_parent"

        android:background="@drawable/background"

        tools:context=".MulaiActivity">

    <Button

        android:id="@+id/ButtonDisconnect"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

```

```

android:layout_marginBottom="249dp"

android:background="@drawable/disconnect_button"

android:text="DISCONNECT"

android:textColor="#FBFBFB"

android:textSize="18sp"

android:textStyle="bold"

app:layout_constraintBottom_toBottomOf="parent"

app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintStart_toEndOf="@+id/ButtonOk"

app:layout_constraintTop_toBottomOf="@+id/SeekBar" />

```

```
<SeekBar
```

```

    android:id="@+id/SeekBar"

    android:layout_width="match_parent"

    android:layout_height="20dp"

    android:layout_marginBottom="55dp"

    android:max="95"

    android:progress="0"

    app:layout_constraintBottom_toTopOf="@+id/ButtonDisconnect"

    app:layout_constraintEnd_toEndOf="parent"

    app:layout_constraintStart_toStartOf="parent"

    app:layout_constraintTop_toBottomOf="@+id/SeekBarNumber" />

```

```
<TextView
```

```

    android:id="@+id/SeekBarNumber"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:layout_marginBottom="25dp"

    android:background="#003F51B5"

    android:text="0"

    android:textColor="FFFFFF"

    android:textSize="25sp"

```

```

app:layout_constraintBottom_toTopOf="@+id/SeekBar"

app:layout_constraintEnd_toEndOf="@+id/switch1"

app:layout_constraintStart_toStartOf="@+id/switch1"

app:layout_constraintTop_toBottomOf="@+id/switch1" />

```

```
<Button
```

```

    android:id="@+id/ButtonOk"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:layout_marginStart="24dp"

    android:layout_marginLeft="24dp"

    android:layout_marginEnd="2dp"

    android:layout_marginRight="2dp"

    android:background="@drawable/ok_button"

    android:text="OK"

    android:textColor="#FBFBFB"

    android:textSize="18sp"

    android:textStyle="bold"

    app:layout_constraintBaseline_toBaselineOf="@+id/ButtonDisconnect"

    app:layout_constraintEnd_toStartOf="@+id/ButtonDisconnect"

    app:layout_constraintStart_toStartOf="parent" />

```

```
<Switch
```

```

    android:id="@+id/switch1"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:layout_marginTop="149dp"

    android:layout_marginBottom="34dp"

    android:showText="true"

    android:text="POWER"

    android:textColor="#FFFFFF"

    android:textOff="OFF"

```

```

        android:textOn="ON"

        android:textSize="20dp"

        android:textStyle="bold"

        android:thumb="@drawable/switch_thumb"

        android:track="@drawable/switch_track"

        app:layout_constraintBottom_toTopOf="@+id/SeekBarNumber"

        app:layout_constraintEnd_toEndOf="parent"

        app:layout_constraintStart_toStartOf="parent"

        app:layout_constraintTop_toTopOf="parent" />

```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

## Bottom\_animation.xml

```

<?xml version="1.0" encoding="utf-8"?>

<set xmlns:android="http://schemas.android.com/apk/res/android">

    <translate

        android:fromXDelta="0%"

        android:fromYDelta="100%"

        android:duration="1500"/>

    <alpha

        android:fromAlpha="0.1"

        android:toAlpha="1.0"

        android:duration="1500"/>

</set>

```

## Top\_animation.xml

```

<?xml version="1.0" encoding="utf-8"?>

<set xmlns:android="http://schemas.android.com/apk/res/android">

    <translate

```



```
    android:fromXDelta="0%"  
  
    android:fromYDelta="-50%"  
  
    android:duration="1500"/>
```

```
<alpha  
  
    android:fromAlpha="0.1"  
  
    android:toAlpha="1.0"  
  
    android:duration="1500"/>
```

```
</set>
```

## Slide\_out\_right.xml

```
<?xml version="1.0" encoding="utf-8"?>  
  
<set xmlns:android="http://schemas.android.com/apk/res/android">  
  
    <translate  
  
        android:duration="@android:integer/config_mediumAnimTime"  
  
        android:fromXDelta="0"  
  
        android:toXDelta="100%p" />  
  
</set>
```

## Slide\_out\_left.xml

```
<?xml version="1.0" encoding="utf-8"?>  
  
<set xmlns:android="http://schemas.android.com/apk/res/android">  
  
    <translate  
  
        android:duration="@android:integer/config_mediumAnimTime"  
  
        android:fromXDelta="0"  
  
        android:toXDelta="-100%p" />  
  
</set>
```

## Slide\_in\_right.xml

```
<?xml version="1.0" encoding="utf-8"?>  
  
<set xmlns:android="http://schemas.android.com/apk/res/android">  
  
    <translate  
  
        android:duration="@android:integer/config_mediumAnimTime"
```

```

        android:fromXDelta="100%p"

        android:toXDelta="0" />

</set>

```

## Slide\_in\_left.xml

```

<?xml version="1.0" encoding="utf-8"?>

<set xmlns:android="http://schemas.android.com/apk/res/android">

    <translate

        android:duration="@android:integer/config_mediumAnimTime"

        android:fromXDelta="-100%p"

        android:toXDelta="0" />

</set>

```

## Go\_button.xml

```

<?xml version="1.0" encoding="utf-8"?>

<ripple xmlns:android="http://schemas.android.com/apk/res/android" android:color="?android:colorControlHighlight">

    <item>

        <shape android:shape="rectangle">

            <solid android:color="#9acd32"/>

            <corners android:radius="999dp"/>

        </shape>

    </item>

</ripple>

```

## Search\_button.xml

```

<?xml version="1.0" encoding="utf-8"?>

<ripple xmlns:android="http://schemas.android.com/apk/res/android" android:color="?android:colorControlHighlight">

    <item

        android:drawable="@drawable/logo_search"/>

</ripple>

```

## Ok\_button.xml

```

<?xml version="1.0" encoding="utf-8"?>

<ripple xmlns:android="http://schemas.android.com/apk/res/android" android:color="?android:colorControlHighlight">

```

```

<item>

    <shape android:shape="rectangle">

        <solid android:color="#3399FF"/>

        <corners android:radius="50dp"/>

    </shape>

</item>

```

```

</ripple>

```

## Disconnect\_button.xml

```

<?xml version="1.0" encoding="utf-8"?>

<ripple xmlns:android="http://schemas.android.com/apk/res/android" android:color="?android:colorControlHighlight">

    <item>

        <shape android:shape="rectangle">

            <solid android:color="#FF0000"/>

            <corners android:radius="50dp"/>

        </shape>

    </item>

</ripple>

```

## Switch\_track.xml

```

<?xml version="1.0" encoding="utf-8"?>

<selector xmlns:android="http://schemas.android.com/apk/res/android">

    <item android:state_checked="true">

        <shape android:shape="rectangle">

            <solid android:color="#148B00"/>

            <corners android:radius="200dp"/>

        </shape>

    </item>

    <item android:state_checked="false">

```

```
<shape android:shape="rectangle">

    <solid android:color="#CDCDCD"/>

    <corners android:radius="200dp"/>

</shape>

</item>

</selector>
```

## Switch\_thumb.xml

```
<?xml version="1.0" encoding="utf-8"?>

<selector xmlns:android="http://schemas.android.com/apk/res/android">

    <item android:state_checked="true">

        <shape android:shape="rectangle">

            <solid android:color="#FFFFFF"/>

            <size android:height="40dp" android:width="50dp"/>

            <corners android:radius="40dp"/>

            <stroke android:width="20dp" android:color="#2DC413"/>

        </shape>

    </item>

    <item android:state_checked="false">

        <shape android:shape="rectangle">

            <solid android:color="#FFFFFF"/>

            <size android:height="40dp" android:width="50dp"/>

            <corners android:radius="40dp"/>

            <stroke android:width="20dp" android:color="#FFFFFF"/>

        </shape>

    </item>

</selector>
```

## AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

    package="com.example.bluetooth">
```

```

<uses-permission android:name="android.permission.BLUETOOTH" />

<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />

<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />

<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

<uses-permission android:name="android.permission.BLUETOOTH_PRIVILEGED" />


<application

    android:allowBackup="true"

    android:icon="@drawable/logo"

    android:label="SMART LED"

    android:roundIcon="@mipmap/ic_launcher_round"

    android:supportRtl="true"

    android:theme="@style/AppTheme">

    <activity android:name=".MulaiActivity" android:theme="@style/AppTheme.NoActionBar"></activity>

    <activity android:name=".MainActivity"></activity>

    <activity android:name=".SearchActivity"></activity>

    <activity android:name=".splash_screen" android:theme="@style/AppTheme.NoActionBar">

        <intent-filter>

            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />

        </intent-filter>

    </activity>

    <meta-data

        android:name="preloaded_fonts"

        android:resource="@array/preloaded_fonts" />

    </application>

</manifest>

```

## LAMPIRAN OUTPUT PROGRAM

### Tampilan Halaman Splash Screen



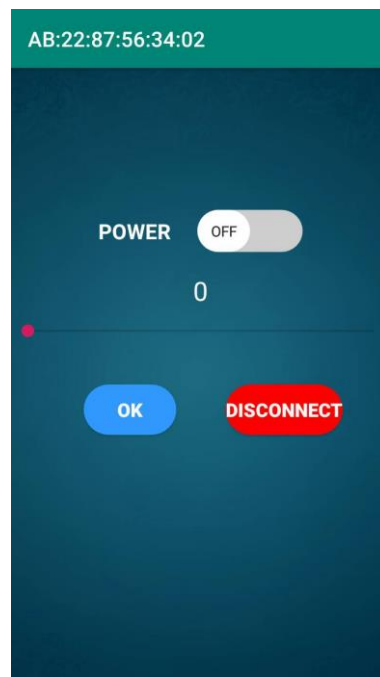
### Tampilan Halaman Mulai



**Tampilan Halaman Search Device**



**Tampilan Halaman Main Activity**



### Intensitas Cahaya Tinggi



### Intensitas Cahaya Sedang





### Intensitas Cahaya Rendah



### Intensitas Cahaya Mati

