

## 1. Pendahuluan

Lampu adalah sebuah perangkat yang mendukung sambungan daya listrik ke sebuah perangkat yang menghasilkan cahaya (disebut lampu, atau bola lampu). Lampu merupakan salah satu perangkat yang sangat penting untuk membantu manusia melakukan aktivitas sehari-hari. Lampu membantu manusia untuk melihat benda atau sesuatu yang terdapat di rumah ketika cahaya alami dirasa intensitasnya kurang memadai untuk melihat sebuah benda. Lampu membutuhkan energi berupa listrik untuk dapat berfungsi sebagai alat penerangan. Untuk menyalakan dan mematikan lampu digunakan saklar pemutus arus listrik[1].

Mikrokontroler dapat mengendalikan lampu tanpa harus menekan saklar fisik pada rangkaian lampu. Mikrokontroler adalah suatu alat elektronika digital yang mempunyai masukan dan keluaran serta kendali dengan program yang bisa ditulis dan dihapus dengan cara khusus[2]. *Arduino* merupakan mikrokontroler yang dapat berkomunikasi dengan perangkat lain melalui protokol *TCP/IP*[3]. Salah satu jenis *smartphone* yang banyak digunakan saat ini adalah *smartphone* yang berbasis *Android*, informasi ini juga didukung oleh data yang dikumpulkan peneliti melalui penyebaran kuisisioner dan data yang menunjukkan bahwa 80,00% responden adalah pengguna *smartphone* berbasis *Android*. *Android* merupakan sebuah sistem operasi yang berbasis *Linux* untuk telepon seluler seperti telepon pintar dan komputer tablet[4].

Penelitian ini juga didasarkan pada hasil pengumpulan data melalui penyebaran kuisisioner yang diberikan kepada 30 responden secara acak. Pada penelitian tersebut didapatkan informasi bahwa 93.33% responden merasa kesulitan jika harus mematikan lampu diseluruh ruangan rumah dengan letak saklar yang berjauhan.

Berdasarkan kajian akan pentingnya kegunaan lampu dalam kehidupan manusia, Maka dilakukan sebuah perancangan untuk membuat aplikasi pengendali lampu dengan mikrokontroler berbasis *Android*. Diharapkan aplikasi ini lebih mudah dan praktis dibandingkan dengan saklar konvensional. Aplikasi ini mengharuskan pengguna terhubung dengan jaringan *WLAN* karena penelitian ini hanya berfokus pada jaringan lokal dan belum memanfaatkan *IP Public* sebagai solusi pengendalian aplikasi dari jarak yang lebih jauh. Hasil dari penelitian ini adalah sebuah aplikasi berbasis *Android* yang bertujuan membantu pengguna untuk mengendalikan lampu rumah sehingga pengendalian lampu menjadi lebih praktis.

## 2. Kajian pustaka

Sebelumnya telah dilakukan penelitian oleh Erwin Pratomo dengan judul Perancangan Sistem Kontrol Kendali Alat Listrik Rumah Tangga Jarak Jauh

Berbasis SMS[5]. Aplikasi tersebut menggunakan SMS dan mikrokontroler sebagai pengendali utama rangkaian elektronik yang digunakan untuk mengendalikan peralatan listrik rumah tangga. Dari penelitian tersebut didapatkan data bahwa aplikasi tersebut mengharuskan pengguna untuk menulis perintah pengendali sesuai dengan format tertentu. Untuk menerjemahkan perintah dari teks yang datang, pengguna membutuhkan sebuah komputer yang selalu menyala. Layanan yang digunakan pada proses pengiriman perintah adalah SMS. Waktu yang digunakan untuk mengirimkan data dan mendapatkan konfirmasi tentang status perangkat rata-rata 8.1 detik. Sedangkan pada penelitian ini perintah untuk mengendalikan lampu berbasis *UDP* dan memiliki rata-rata waktu konfirmasi 3.1 detik. Format dari aplikasi ini bersifat *autogenerated* hal ini dilakukan agar pengguna lebih sedikit melakukan kesalahan dalam penulisan perintah. Perintah dari aplikasi *Android* dapat langsung diterima oleh mikrokontroler melalui *ethernet shield* yang berada pada mikrokontroler. Jika menggunakan jaringan lokal maka aplikasi ini tidak membutuhkan biaya transfer data untuk dapat sampai di tujuan paket.

*Android* adalah sebuah sistem operasi yang berbasis *Linux* untuk telepon seluler seperti telepon pintar dan komputer tablet. *Android* menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. *Android* sudah mengeluarkan banyak versi, sampai pada April 2013, ada 16 versi dari *Android*[6].

Mikrokontroler adalah suatu alat elektronika digital yang mempunyai masukan dan keluaran serta kendali dengan program yang bisa ditulis dan dihapus dengan cara khusus, cara kerja mikrokontroler sebenarnya membaca dan menulis data. Jika diartikan secara harafiah mikrokontroler berarti “pengendali kecil”. Dengan mikrokontroler maka sebuah sistem elektronik bisa menjadi lebih murah dan ringkas[2].

*Arduino* adalah piranti keras berlisensi terbuka yang dimulai pada tahun 2005 dengan harapan dapat membawa dunia elektronik digital ke edukasi, penelitian dan komunitas. *Arduino* memiliki prinsip *ease of use* (kemudahan dalam penggunaan), *openness* (keterbukaan) dan *world-wide availability* (ketersediaan secara global). *Arduino* dimulai sebagai *prototyping circuit board* yang sederhana, sebuah komputer kecil yang berjalan pada 16 Mhz. *Arduino* tidak memiliki layar dan papan tombol dan oleh karena itu, *Arduino* membutuhkan komputer eksternal untuk memprogramnya. Komputer tersebut membutuhkan sebuah software yang harus dijalankan, bernama *Arduino IDE* yang berfungsi untuk menulis, *compiling* dan mengupload program ke *Arduino board*. Setelah itu *Arduino board* berjalan secara otomatis, tidak membutuhkan komputer atau *IDE* untuk menjalankan kode program yang sudah di *upload*. *Arduino* membutuhkan dokumentasi untuk mempelajari hampir semua fiturnya. Maka dari itu *Arduino* memberikan file referensi dan tutorial untuk pengembang, sehingga pengembang dapat belajar menggunakan teknologi digital yang disediakan. Semua dokumentasi dapat dilihat pada situs *Arduino*. Dokumentasi resmi selalu tertulis dalam bahasa Inggris dan diterjemahkan ke bahasa lain oleh sukarelawan. Ada banyak jenis dari *Arduino*

*boards* sehingga bisa menyesuaikan kebutuhan dari pengguna. Sebagai contoh, karena penelitian ini membutuhkan *board* yang bisa diberi modul tambahan *ethernet shield* maka *Arduino* yang digunakan adalah *Arduino Uno R3*[2].

*Wi-Fi (Wireless Fidelity)* adalah koneksi tanpa kabel seperti *smartphone* dengan mempergunakan teknologi radio sehingga pemakainya dapat mentransfer data dengan cepat[7]. Dalam perancangan aplikasi *Android* ini *wi-fi* digunakan untuk melakukan interkoneksi antara mikrokontroler dan *smartphone Android*.

Ada dua buah protokol yang dapat digunakan untuk implementasi penelitian ini, *Transmission Control Protocol(TCP)* dan *User Datagram Protocol(UDP)*. Masing masing protokol memiliki kelebihan dan kekurangan, *TCP* mengharuskan terjadinya koneksi sebelum pengiriman paket terjadi dan memastikan semua pecahan paket diterima oleh alamat tujuan. *User Datagram Protocol(UDP)* merupakan salah satu protokol utama diatas *IP*, merupakan *transport protocol* yang lebih sederhana dibandingkan dengan *TCP*. *UDP* digunakan untuk situasi yang tidak mementingkan mekanisme reliabilitas. *Header UDP* hanya berisi empat *field* yaitu *source port*, *destination port*, *length* dan *UDP checksum* dimana fungsinya hampir sama dengan *TCP*, namun fasilitas *checksum* pada *UDP* bersifat opsional[8].

*Wireshark* merupakan salah satu dari sekian banyak tool Network Analyzer yang banyak digunakan oleh Network administrator untuk menganalisa kinerja jaringannya termasuk protokol didalamnya [9]. Dalam penelitian ini *Wireshark* digunakan untuk meng-capture paket *UDP* yang berasal dan menuju Aplikasi.

### 3. Metodologi Penelitian

Metodologi penelitian pada hakekatnya merupakan operasionalisasi dari epistemologi yang mengkaji perihal urutan langkah-langkah yang ditempuh supaya pengetahuan yang diperoleh memenuhi ciri-ciri ilmiah, dan penelitian itu sendiri adalah merupakan suatu proses mencari sesuatu secara sistimatis dalam waktu yang relatif lama dengan menggunakan metode ilmiah dengan prosedur maupun aturan yang berlaku[10]. Gambar 3.1 menjelaskan tahapan penelitian yang akan dilakukan

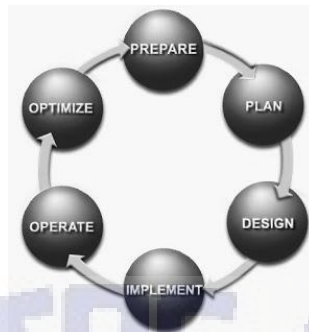


Gambar 1 Tahapan Penelitian[10]

Tahap pertama dalam tahapan penelitian adalah identifikasi masalah yaitu dilakukan penyebaran kuisioner kepada 30 responden yang diambil secara acak dengan tujuan mendapatkan data yang akan dirumuskan pada tahapan penelitian selanjutnya. Tahapan kedua dalam penelitian ini adalah rumusan masalah. Beberapa hasil penelitian yang dapat digunakan sebagai dasar rumusan masalah adalah sebagai berikut: 1) sebanyak 93,33% responden menyatakan bahwa responden merasa kesulitan jika harus mematikan dan menghidupkan lampu diseluruh ruangan rumah dengan letak saklar yang berjauhan 2) 100% responden akan merasa terbantu jika dirancang sebuah aplikasi pengendali lampu dengan perangkat tertentu 3) 60% responden ingin dapat meredupkan lampu ketika ingin tidur 4) 90% responden ingin dapat mematikan lampu teras tanpa harus meninggalkan aktifitas yang sedang dikerjakan. Tahapan ketiga dalam penelitian adalah penelusuran pustaka, Penelusuran pustaka merupakan langkah pertama untuk mengumpulkan informasi yang relevan bagi penelitian. Penelusuran pustaka berguna untuk menghindari duplikasi dari pelaksanaan penelitian. Dengan penelusuran pustaka maka akan dapat diketahui penelitian yang pernah dilakukan[10].

Tahap keempat dalam penelitian ini adalah rancangan penelitian, metode yang digunakan pada tahap perancangan adalah *PPDIOO*. Metode *PPDIOO* adalah salah satu metode yang digunakan untuk perancangan jaringan. Tahap tahap dari metode tersebut dapat dijelaskan seperti berikut ini:

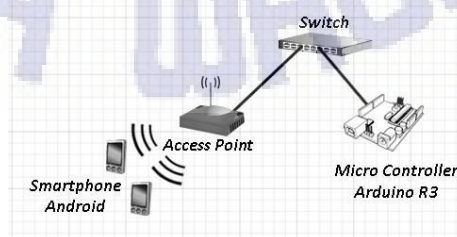
1. Pada tahap *Prepare*, proses yang dilakukan adalah mempersiapkan segala sesuatu. Dimulai dari persiapan kebutuhan untuk jaringan awal agar dapat melakukan analisis awal untuk proses pengiriman dan penerimaan paket *datagram* dari *Android* menuju Mikrokontroler dan sebaliknya.
2. *Plan* dalam tahap ini, yang dilakukan adalah perencanaan jaringan yang dibuat serta menentukan *hardware* dan *software* yang digunakan dalam penelitian ini. Serta skenario yang dilakukan dalam penelitian ini untuk menggambarkan proses penelitian. Tahapan *plan* juga menampilkan *flowchart* diagram yang menjelaskan alur kerja aplikasi dalam penelitian ini.
3. *Design*, dalam tahapan desain ini dibuat suatu topologi jaringan untuk proses *remoting* menggunakan protokol *UDP*. Serta konfigurasi awal yang dilakukan untuk masing masing perangkat.
4. *Implement* pada tahap ini, desain yang telah dibuat diimplementasikan dengan melakukan pengkodean program pada *platform Android* dan Mikrokontroler *Arduino*.
5. *Operate* setelah implementasi perangkat dalam topologi jaringan, langkah selanjutnya adalah proses pengoperasian dengan melakukan konfigurasi yang sudah dirancang dalam tahap desain sebelumnya.
6. *Optimize* tahap optimasi ini dilakukan dengan menganalisis kinerja jaringan yang sudah dibuat apakah sudah berjalan dengan baik.



**Gambar 2** Metode PPDIOO[11]

Pada tahap *prepare* dan *plan* semua kebutuhan yang diperlukan untuk membangun aplikasi, kebutuhan *hardware* maupun *software* yang digunakan dalam penelitian akan dipaparkan. Kebutuhan perangkat lunak dalam penelitian ini adalah *ADT Bundle* yang berfungsi untuk membuat aplikasi pada *platform Android* dan *Arduino SDK* yang berfungsi untuk mengunggah kode program ke *Arduino*. Kebutuhan perangkat keras pada penelitian ini adalah *personal computer (PC)*, *switch*, *Access Point*, kabel *unshielded twisted pair (UTP)*, Mikrokontroler *Arduino Uno R3*, Lampu *Light Emitting Diode (LED)* 5V dan 220V, *NONC Switch Button*, Potensiometer, kabel *Jumper*, dan *Relay*. Tahap selanjutnya adalah menentukan protokol yang akan digunakan, yaitu *UDP* karena protokol *UDP* memungkinkan aplikasi untuk memberikan informasi kepada seluruh *host* secara simultan dan tidak mempedulikan apakah tujuan paket yang dikirim hidup atau mati sehingga mempercepat transfer data.

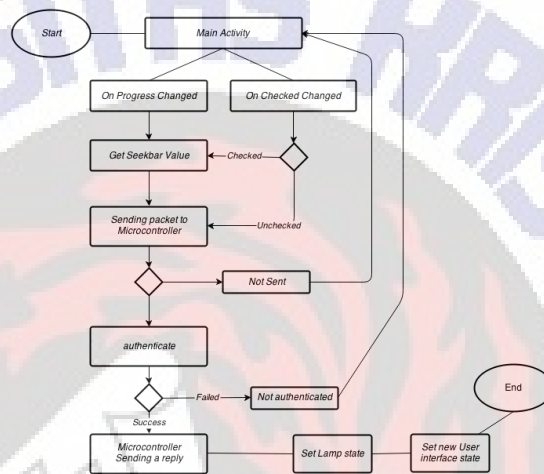
Pada tahap *design* perancangan arsitektur jaringan yang akan digunakan dalam aplikasi pengendali lampu berbasis *Android* dengan mikrokontroler dibuat dan digambarkan, Gambar 3 menunjukkan rancangan arsitektur topologi jaringan *LAN* berdasarkan topologi *star*. Pada gambar 3 ditunjukkan hubungan yang menggambarkan alur pertukaran data antara *Android* dan Mikrokontroler. Topologi pada gambar 3 adalah topologi yang digunakan dalam perancangan aplikasi ini.



**Gambar 3** Topologi jaringan yang digunakan dalam penelitian ini

Diagram alur menggambarkan proses yang terjadi pada aplikasi *Android* menuju mikrokontroler dimulai ketika pengguna membuka aplikasi dan menampilkan antarmuka aplikasi *Android*. Ketika pengguna menekan salah satu

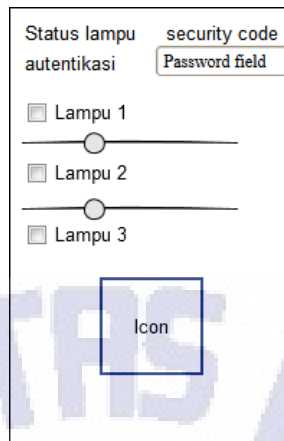
tombol lampu maka proses yang terjadi selanjutnya adalah mengirim semua yang diinputkan oleh pengguna menuju ke mikrokontroler, setelah sampai ke mikrokontroler verifikasi kode keamanan dilakukan untuk memastikan bahwa pengguna berhak untuk mengatur lampu. Ketika verifikasi kode keamanan gagal maka user akan mendapat pemberitahuan bahwa kode yang diinputkan tidak cocok dengan kode verifikasi. Ketika kode verifikasi cocok maka mikrokontroler membaca isi dari paket yang dikirim oleh aplikasi *Android* dan melakukan perintah pengendalian lampu sesuai dengan inputan. Proses selanjutnya adalah mengirim paket balasan ke aplikasi *Android* yang berisi status terkini dari lampu.



**Gambar 4:** Diagram alur pada aplikasi *Android*

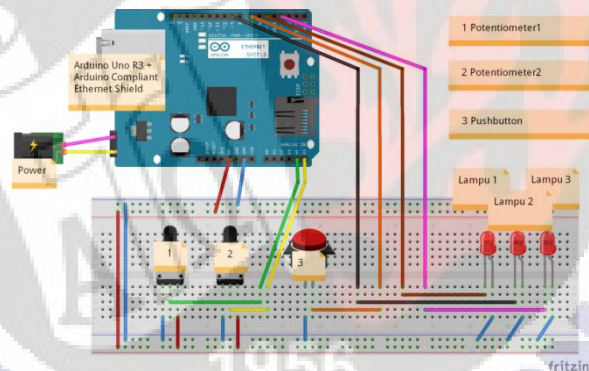
Pada gambar 4 dapat diketahui alur bagaimana aplikasi ini berjalan, dimulai dari menginisialisasi *Main Activity* sehingga *GUI* dapat ditampilkan, setelah itu jika pengguna menekan *checkbox* maka aplikasi akan menentukan apa yang akan dikirimkan ke mikrokontroler berdasarkan status aplikasi saat ini. Setelah paket datagram diterima oleh mikrokontroler maka mikrokontroler akan melakukan apa yang diperintahkan paket datagram. Mikrokontroler selanjutnya akan mengirim paket balasan yang berfungsi untuk memberitahu status terkini dari lampu.

Kemudian pada tahap selanjutnya dibuat *Graphical User Interface* untuk aplikasi pengendali lampu. *Graphical user interface* adalah salah satu bentuk antarmuka yang memungkinkan manusia untuk berinteraksi dengan perangkat lunak selain menggunakan perintah tulisan. Sebuah *GUI* menyediakan ikon grafis dan indikator visual, berlawanan dengan antarmuka berbasis teks. Aksi yang dilakukan pengguna dilakukan melalui manipulasi secara langsung terhadap elemen grafis pada *GUI*[12]. Gambar 5 merupakan desain tampilan aplikasi pengendali lampu pada *Android*.



**Gambar 5** *Graphical User Interface Aplikasi Android*

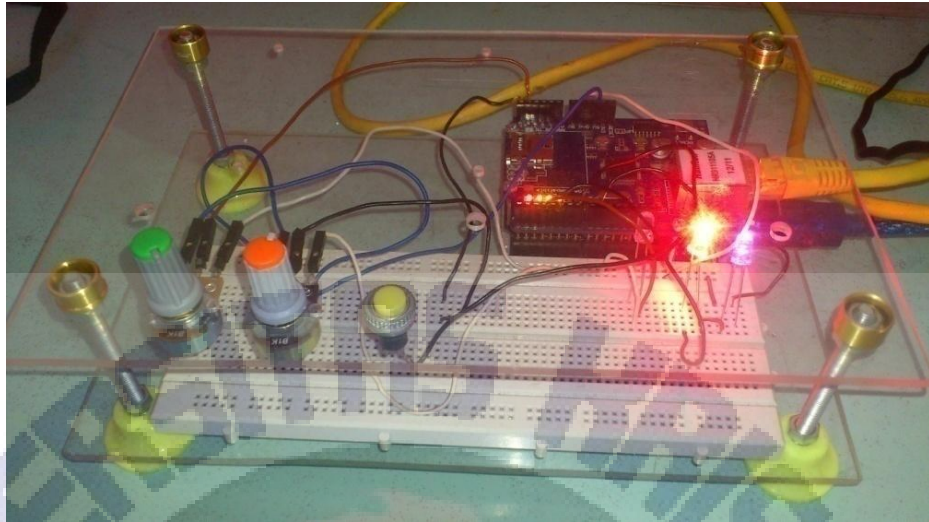
Langkah terakhir pada tahap *design* adalah membuat skema rangkaian Mikrokontroler *Arduino*. Skema Rangkaian pada gambar 6 menggunakan software yang bernama *Fritzing*. *Fritzing* adalah sebuah aplikasi *freeware* yang bisa digunakan untuk membuat *schematic diagram* baik dua dimensi maupun 3 dimensi[13]. Gambar 6 adalah skema rangkaian lampu yang disambungkan dengan mikrokontroler *Arduino*.



**Gambar 6** Skema Rangkaian Mikrokontroler *Arduino*

#### 4. Hasil dan Pembahasan

Setelah *design* sudah dibuat, maka langkah selanjutnya adalah *implementation* pada tahap ini pengkodean pada *Arduino IDE* dan *Eclipse IDE* dilakukan.



**Gambar 7:** Mikrokontroler pada saat dioperasikan

Gambar 7 menunjukkan mikrokontroler ketika dioperasikan, mikrokontroler dapat mengatur intensitas cahaya berdasarkan inputan dari potensiometer ataupun saklar *pushbutton*. Potensiometer membutuhkan tegangan 5v untuk dapat beroperasi dengan baik. Sedangkan saklar *pushbutton* tidak membutuhkan tegangan tambahan selain dari digital pin dari *Arduino*. Lampu 1 dan Lampu 2 diatur menggunakan fitur *Pulse Width Modulation (PWM)* sehingga memungkinkan pengguna untuk mengatur intensitas cahaya sesuai dengan yang dikehendaki oleh pengguna.

**Kode Program 1:** Mengatur nilai Lampu dengan analog pinout berdasarkan Potensiometer

```
1 valKam = analogRead(potenKam);
2 outKam = map(valKam, 0, 1023, 0, 255);
3 if(prevKam ==outKam ||prevKam == outKam + 1 ||prevKam == outKam -1 )
4 { //tidak melakukan apa - apa } else {
5 ledStateK = outKam;
6 analogWrite(ledTam, ledStateK);
7 prevKam = outKam;
```

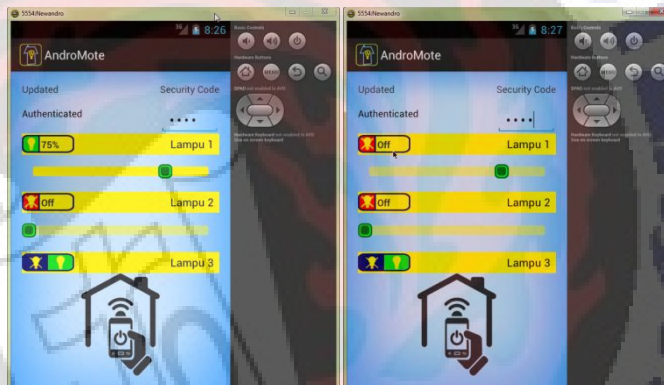
Dalam Kode Program 1 baris pertama pada baris ke-1 digunakan untuk membaca nilai dari potensiometer. Potensiometer memberikan rentang nilai antara 0 – 1023. Baris ke- 2 digunakan untuk mengubah rentang bilangan dari rentang awal menjadi rentang bilangan yang baru, yaitu dari 0 – 1023 menjadi 0 – 255. Baris ke- 3 dan ke-4 digunakan untuk menyaring nilai dari potensiometer. Baris ke- 6 digunakan untuk memberikan nilai secara analog kepada Lampu berdasarkan *re-map* dari nilai potensiometer.



**Kode Program 2:** mengatur nilai Lampu digital pinout berdasarkan Pushbutton

```
1 int reading = digitalRead(buttonPin);
2 if (reading != lastButtonState) {
3   if (reading != buttonState) {
4     buttonState = reading;
5     if (buttonState == HIGH) {
6       ledState != ledState; } } }
7 digitalWrite(ledPin, ledState);
8 lastButtonState = reading;
```

Kode Program 2 berfungsi untuk menghidupkan atau mematikan Lampu 3 berdasarkan kondisi sebelumnya, Lampu 3 adalah lampu yang diatur dengan *input* digital. Baris ke- 4 menunjukkan jika saklar ditekan maka langkah selanjutnya adalah membaca status dari Lampu 3 dan membalik kondisi lampu, jika kondisi Lampu 3 sebelumnya adalah mati maka ketika saklar ditekan Lampu 3 akan menjadi hidup dan sebaliknya



**Gambar 8:** Hidup dan Mati Lampu

Lampu 1 dan Lampu 2 bisa dikendalikan melalui dua cara, yaitu melalui aplikasi yang sudah terpasang di *smartphone Android* atau melalui potensiometer, penggunaan potensiometer ditujukan sebagai alternatif pengendalian lampu jika dalam situasi penggunaan saklar fisik dianggap lebih praktis dari penggunaan aplikasi pengendali. Penggunaan potensiometer juga merupakan langkah antisipasi jika terjadi gangguan pada jaringan *Local Area Network* atau terjadi malfungsi pada *smartphone Android* yang digunakan sebagai pengendali.

Lampu 3 tidak menggunakan fitur *PWM* maka hanya ada dua kondisi yang bisa terjadi pada Lampu 3, yaitu hidup atau mati sehingga Lampu 3 tidak membutuhkan *seekbar* untuk mengatur intensitas lampu. Gambar 9 menunjukkan Lampu 3 dalam keadaan hidup atau mati. Seperti halnya Lampu 1 dan Lampu 2, Lampu 3 juga membutuhkan tombol fisik maka dari itu *pushbutton switch* digunakan dan terkoneksi dengan Lampu 3.

Ketika *checkbox* ditekan ataupun *seekbar* di ubah nilainya aplikasi ini tidak akan secara langsung mengubah nilai yang terlihat di antarmuka aplikasi tetapi aplikasi menunggu balasan paket datagram yang dikirim oleh mikrokontroler ke aplikasi *Android*. Ketika lampu telah diubah nilainya tetapi aplikasi belum

menerima paket balasan dari mikrokontroler. Maka pada teks status aplikasi masih menunjukkan informasi bahwa aplikasi masih menunggu pembaruan status dengan pesan “loading...”. Selama belum terbaru maka tampilan kecerahan lampu masih berada pada nilai sebelumnya.

Ketika tombol pada aplikasi *Android* yang digunakan untuk memerintahkan lampu ditekan maka aplikasi akan menggabungkan *string* yang didapat dari *security code*, *check-box id* dan seekbar value yang kemudian tipe datanya diubah menjadi *byte* sehingga data tersebut dapat dikemas dalam paket datagram yang diberi *header IP address* tujuan yaitu 192.168.12.177 yang merupakan alamat *IP* dari mikrokontroler beserta port tujuan dari paket yaitu 6666 yang merupakan *listening port* dari mikrokontroler. Setelah paket datagram dibuat maka paket tersebut akan dikirim melalui protokol *UDP* dengan datagram *socket*. Setelah paket dikirim langkah selanjutnya adalah menutup *socket* datagram, hal ini dilakukan agar aplikasi tidak menggunakan sumber daya sewaktu tidak ada aktivitas sama sekali dalam hal transfer data.

Ketika aplikasi *Android* dijalankan, maka setiap 3 detik aplikasi akan membuka *socket* untuk menerima paket datagram pada port 6666 jikalau ada, paket tersebut kemudian ditampung dalam bentuk *datagram packet*. Kemudian aplikasi hanya akan mengambil data yang bertipe *byte* tanpa mengambil *packet header* karena yang dibutuhkan oleh aplikasi hanya data yang ada di dalamnya, kemudian data yang berupa *byte* tersebut dikonversi menjadi *String* yang kemudian akan dipecah sesuai dengan format sehingga didapatkan informasi untuk mengubah tampilan aplikasi *Android* sesuai dengan status lampu.

**Kode Program 3:** *Parsing* setiap paket yang datang.

```
1 int packetSize = Udp.parsePacket();
2 if(packetSize) {
3   Serial.print("Received packet of size ");
4   Serial.println(packetSize);
5   Serial.print("From ");
6   IPAddress remote = Udp.remoteIP();
7   for (int i =0; i < 4; i++){
8     Serial.print(remote[i], DEC);
9     if (i < 3){Serial.print(".");}}
10  Serial.print(", port ");
11  Serial.println(Udp.remotePort());
12  Udp.read(packetBuffer,UDP_TX_PACKET_MAX_SIZE);
```

Kode program 3 adalah kode program pada mikrokontroler yang digunakan untuk memeriksa adakah paket yang datang, jika ada paket yang datang maka langkah selanjutnya adalah menulis di *serial interface* tentang pengirim paket dan kemudian menyimpan data yang ada di dalam paket ke dalam *buffer* yang memiliki tipe data *byte*. Mikrokontroler juga mengambil informasi tentang pengirim paket untuk digunakan sebagai header pada paket balasan.

Ketika mikrokontroler menerima paket datagram melalui perintah *UDP.receive*, maka paket yang berupa *byte* akan diubah menjadi *char array* kemudian *array* dibaca setiap indexnya sesuai dengan format yang telah diterapkan sebelumnya, index 0 sampai dengan 3 adalah *security code* yang

digunakan untuk meng-autentikasi pengguna, jika pengguna ter-autentikasi maka index 4 dari *char array* digunakan untuk menentukan lampu mana yang akan diperintah sesuai dengan keinginan pengguna, *Index 5* sampai *7* digunakan untuk menentukan intensitas cahaya yang diinginkan.

Aplikasi ini juga dilengkapi dengan proteksi keamanan sederhana, ketika mengirim perintah dari *Android* maka langkah pertama yang dilakukan ketika paket tiba di mikrokontroler adalah melakukan verifikasi *security code* yang sebelumnya telah dimasukkan oleh pengguna. Ketika *security code* tidak cocok maka aplikasi *Android* hanya akan bisa menerima status lampu, aplikasi tidak akan dapat digunakan untuk mengendalikan lampu.

```
Internet Protocol Version 4, Src: 192.168.12.8 (192.168.12.8), Dst: 192.168.12.177 (192.168.12.177)
  Version: 4
  Header length: 20 bytes
  Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
  Total Length: 33
  Identification: 0x099e (2462)
  Flags: 0x00
  Fragment offset: 0
  Time to live: 128
  Protocol: UDP (17)
  Header checksum: 0x9724 [correct]
  Source: 192.168.12.8 (192.168.12.8)
  Destination: 192.168.12.177 (192.168.12.177)
  User Datagram Protocol, Src Port: 49858 (49858), Dst Port: 6666 (6666)
  source port: 49858 (49858)
  destination port: 6666 (6666)
  Length: 13
  Checksum: 0xbf98 [validation disabled]
  [Good Checksum: False]
  [Bad Checksum: False]
  Data (5 bytes)
  Data: 6331323334
  [Length: 5]
```

Gambar 9: capture paket UDP

Gambar 9 menunjukkan apa isi dari paket yang dikirim dari *Android* menuju ke Mikrokontroler yang telah di-*capture* oleh aplikasi *Wireshark*. Paket tersebut berisi informasi tentang versi *IP* yang digunakan yaitu *Ipv4*, alamat pengirim paket yaitu *IP* dari *smartphone Android* yang diperoleh secara otomatis menggunakan protokol *DHCP* gambar 9 menunjukkan alamat pengirim paket adalah 192.168.12.8, alamat tujuan paket adalah 192.168.12.177 karena *IP address* tersebut adalah *IP* dari mikrokontroler, *source port* adalah *port* asal dari pengirim paket dan dipilih secara acak selama *port* tersebut tidak digunakan oleh layanan lain, *destination port* adalah *port* yang dituju pada aplikasi ini *port* tujuan adalah 6666, protokol yang digunakan untuk mengirim paket yaitu *User Datagram Protocol* karena aplikasi ini menggunakan *UDP* sebagai protokol untuk melakukan lalu lintas data, besarnya data saat pengiriman paket diukur dengan *byte*, gambar 9 menunjukkan besarnya *byte* yang dikirim oleh *smartphone Android* dan diterima oleh mikrokontroler besarnya *byte* yang dikirim adalah 5 *byte*.

Dua tahap terakhir dari metode *PPDIOO* adalah *Operate* dan *Optimize*. Pada tahap ini aplikasi yang sudah dapat berjalan di uji coba dengan cara di operasikan dan melihat apakah semua fungsi sudah sesuai dengan yang diharapkan atau belum, jika dirasa masih kurang optimal maka dilakukan penyesuaian pada tahap *optimize*.

Setelah aplikasi selesai dibuat dan mikrokontroler selesai dikonfigurasi kemudian dilakukan pengumpulan data dengan pengujian *alpha* menggunakan metode *black-box*[14] dengan hasil yang dapat dilihat pada tabel 3.

**Tabel 1.** hasil pengujian *black-box*

Proses	Hasil yang diharapkan	Hasil yang muncul	Waktu rata-rata yang Dibutuhkan	Kesimpulan
Menyalakan Lampu 1	Lampu 1 menyala	Lampu 1 menyala	3,1	Sukses
Menyalakan Lampu 2	Lampu 2 menyala	Lampu 2 menyala	3,1	Sukses
Menyalakan Lampu 3	Lampu 3 menyala	Lampu 3 menyala	3,1	Sukses
Mengatur tingkat cahaya Lampu 1	Intensitas cahaya Lampu 1 berubah	Intensitas cahaya Lampu 1 berubah	3,1	Sukses
Mengatur tingkat cahaya Lampu 2	Intensitas cahaya Lampu 2 berubah	Intensitas cahaya Lampu 2 berubah	3,1	Sukses

Selanjutnya dilakukan pengujian *beta* dengan menggunakan metode Likert kepada 30 responden secara acak dan mendapatkan hasil berdasarkan beberapa pertanyaan yang dapat dilihat dalam Tabel 2.

**Tabel 2.** pengujian Beta  
Keterangan: S = Setuju N = Netral TS = Tidak Setuju

No.	Pertanyaan	Jawaban		
		S	N	TS
1.	Menurut anda aplikasi ini sudah sangat membantu.	80%	20%	0%
2.	Menurut anda aplikasi ini mudah digunakan.	80%	20%	0%
3.	Anda memahami tampilan aplikasi ini.	70%	30%	0%
4.	Dengan menggunakan aplikasi ini pengendalian lampu menjadi lebih praktis.	90%	6,67%	3,33%
5.	Anda tertarik untuk memiliki aplikasi ini	86,6%	13,33%	0%

Berdasarkan pengujian beta diperoleh hasil bahwa aplikasi sudah sangat membantu mempermudah pengendalian lampu di rumah yang memiliki letak saklar berjauhan, karena sebanyak 80% dari jumlah responden menyetujui pertanyaan ini, selain itu aplikasi mudah digunakan, pengguna juga memahami

tampilan aplikasi dan dengan menggunakan aplikasi ini pengguna merasa lebih praktis sehingga 86,6% dari pengguna merasa tertarik untuk memiliki aplikasi ini.

## 5. Simpulan

Dilihat dari pembahasan dalam penelitian ini, maka dapat disimpulkan perancangan ini dibagi menjadi dua bagian utama, yaitu *interface* yang terdapat pada aplikasi *Android*, yang berfungsi untuk mengatur dan melihat status lampu. Dan *UDP server* yang berupa *Arduino* mikrokontroler yang bertugas untuk membaca *input* dari pengguna kemudian melakukan verifikasi keamanan dan memerintahkan *pinout* dari *Arduino* untuk melakukan tugas sesuai dengan perintah serta memberikan status lampu kepada *Android interface*. Berdasarkan penelitian dan pengujian yang telah dilakukan maka dapat disimpulkan bahwa *Android* dapat terhubung dengan mikrokontroler dengan menggunakan protokol *UDP* dan dapat digunakan menjadi pengendali lampu.

Dalam pengembangan maupun penelitian selanjutnya ada beberapa saran yang dapat dijadikan pertimbangan, yaitu: 1) Dengan adanya kemampuan untuk terhubung dengan jaringan listrik maka tidak menutup kemungkinan jika nantinya aplikasi ini dikembangkan untuk dapat menjadi pengendali peralatan lain seperti kunci pintu otomatis, penyiram tanaman otomatis dan rolling door. 2) Jika dilihat dari jangkauan jaringan pengendali yang hanya mencakup *Local Area Network* maka diharapkan peneliti selanjutnya tidak hanya menggunakan jaringan lokal tetapi jaringan internet. 3) Mengingat keterbatasan aplikasi ini yang hanya menggunakan *platform Android* maka diharapkan peneliti selanjutnya dapat menggunakan *platform* selain *Android*. 4) Memikirkan keamanan jaringan dari aplikasi ini yang sama sekali belum dirancang, maka diharapkan pada penelitian selanjutnya dapat mengembangkan aplikasi ini lengkap dengan sistem keamanan.

## 6. Daftar Pustaka

- [1] Nawazir, 2012, *Pengertian Lampu*, <http://id.shvoong.com/exact-sciences/physics/2285162-pengertian-lampu/>, Diakses tanggal 4 juni 2014.
- [2] Perangin-angin, Bisman, 2014, *Penggerak Antena Modem USB Tiga Dimensi Berbasis Mikrokomputer Menggunakan Arduino Uno*, Medan: Universitas Sumatera Utara.
- [3] Goransson, Andreas dan David Cuartielles Ruiz, 2013, *Professional Android Open Accessory Programming with Arduino*, Canada: John Wiley & Sons.
- [4] Risal, Muhammad, 2011, *Apa itu Gadget dan Pengertian Gadget*, <http://www.artikelbagus.com/2011/11/apa-itu-gadget-dan-pengertian-gadget.html>, Diakses tanggal 11 Januari 2014.
- [5] Pratomo, Erwin, 2012, *Perancangan Sistem Kontrol Kendali Alat Listrik Rumah Tangga Jarak Jauh Berbasis SMS*, Salatiga: Universitas Kristen Satya Wacana.
- [6] developer.android.com, 2014, *<Uses SDK>*, <http://developer.android.com/guide/topics/manifest/uses-sdk-element.html>, Diakses tanggal 17 Februari 2014.

- [7] Supriyanto, 2013, *Teknologi WIFI (Wireless Fidelity) dan Implementasinya*, <http://www.vedcmalang.com/pppstkboemlg/index.php/artikel-coba-2/teknologi-informasi/544-teknologi-wifi>, Diakses tanggal 20 Maret 2014.
- [8] Hadianto, Martono, 2010, *Analisis Dan Perancangan Quality Of Service Pada Jaringan Voice Over Internet Protocol Berbasis Initiation Protocol*, Bandung: Universitas Komunikasi Indonesia.
- [9] Aryadi, I Wayan dan Komang Kusuma, 2014, *Mengukur Quality of Service (QoS) Pada Video Conference*, Bali: Universitas Udayana.
- [10] Hasibuan, Zainal A, 2007, *Metodologi Penelitian Pada Bidang Ilmu Komputer Dan Teknologi Informasi*, Fakultas Ilmu Komputer Universitas Indonesia.
- [11] Cisco System, 2007, *PPDIOO Method*, [http://www.cisco.com/global/EMEA/IPNGN/ppdoo\\_method.html](http://www.cisco.com/global/EMEA/IPNGN/ppdoo_method.html), diakses pada 20 Maret 2014.
- [12] Pratama, Arta, 2011, *Perancangan Dan Evaluasi Heuristik Pada Perangkat Lunak Manajemen Proyek Dengan Prinsip Usability Nielsen*, Medan: Universitas Sumatera Utara.
- [13] Zhang, Andy S, 2012, *Development of a Low-Cost Mobile Embedded Handheld AirCasting Device*, New York: NYC College of Technology.
- [14] Riduwan, 2005, *Rumus dan Data dalam Analisis Statistika*. Bandung: Alfabeta.