

Sistem Pengendali *Smart-Kontak* dengan Aplikasi Android dan Web

Suraidi¹

Email: suraidi@ft.untar.ac.id

Program Studi Teknik Elektro Universitas Tarumanagara

Shinta Nathania¹

Program Studi Teknik Elektro Universitas Tarumanagara

ABSTRACTS : *Smart-kontak is a portable power outlet that can be controlled remotely by using Android application and website. Android application can run smart-kontak through Bluetooth, whereas website can manage smart-kontak through Wi-Fi. Both Android application and website can control the power button of the smart-kontak, however there is an additional feature on the website as a timer setting which will set how long the smart-kontak be on. The on/off status and remaining time of the smart-kontak can be seen on Android application and website. Its system design uses microcontroller LOLIN D32 as processing module and Arduino Software (IDE) to write the program. Android application is designed by using Android Studio Software while the website is written with HTML. System testing is done by controlling the smart-kontak through Android application and website. The testing shows that smart-kontak can be run through Android application and website according to the program and all modules function well.*

KEYWORDS: *smart-kontak; portable; timer; bluetooth; website.*

ABSTRAK: *Smart-kontak merupakan stopkontak portable yang dapat dikendalikan dari jarak jauh dengan menggunakan aplikasi Android dan situs web. Aplikasi Android dapat mengendalikan smart-kontak melalui Bluetooth, sedangkan situs web dapat mengendalikan smart-kontak melalui Wi-Fi. Aplikasi Android dan situs web dapat mengatur on/off dari smart-kontak, namun pada situs web terdapat fitur tambahan berupa timer untuk mengatur lamanya smart-kontak menyala. Status on/off dan remaining time dari smart-kontak dapat dilihat pada Aplikasi Android dan situs web. Perancangan sistem ini menggunakan mikrokontroler LOLIN D32 sebagai modul pemrosesnya dan Arduino Software (IDE) untuk menulis program. Aplikasi Android dirancang menggunakan software Android Studio sedangkan situs web ditulis menggunakan HTML. Pengujian sistem ini dilakukan dengan mengendalikan smart-kontak melalui aplikasi Android dan situs web. Pengujian sistem menunjukkan bahwa smart-kontak dapat dikendalikan melalui aplikasi Android dan situs web sesuai dengan program yang dibuat dan semua modul berfungsi sesuai dengan baik.*

KATA KUNCI: *smart-kontak; portable; timer; bluetooth; situs web.*

PENDAHULUAN

Berbagai alat elektronik telah menjadi kebutuhan penting saat ini dan seringkali para pengguna lupa untuk mematikan alat-alat listrik yang sedang terpasang karena kesibukan aktivitas lain yang sedang dilakukan. Salah satu alat elektronik yang banyak digunakan oleh masyarakat saat ini adalah *smartphone*. Lembaga riset *digital marketing E-marketer* memperkirakan pada tahun 2018, jumlah pengguna aktif *smartphone* di Indonesia adalah lebih dari seratus juta orang [1]. Seringkali pengguna *smartphone* meninggalkan *smartphone* dalam keadaan *charging* sementara pemiliknya melakukan aktivitas lain atau bahkan ditinggal tidur sampai pagi yang dapat mengakibatkan baterai pada *smartphone* mengalami *overcharging* [2].

Meskipun saat ini hampir seluruh alat elektronik sudah dilengkapi dengan *circuit* pencegah kerusakan seperti *overcharge protection* yang dengan otomatis memutus arus listrik ke baterai jika baterai sudah penuh, namun hal ini bukan berarti meniadakan penggunaan listrik, karena biasanya *charger* hanya akan berubah menjadi *sleep mode* atau *standby mode* menunggu perintah *circuit board* pada baterai untuk melakukan proses *charging* kembali. Hal ini berarti tidak menutup kemungkinan terjadinya korsleting listrik yang dapat mengakibatkan kebakaran.

Sebagian besar kasus kebakaran pemukiman disebabkan karena korsleting atau hubung singkat kabel listrik. Sebagai contoh, 63,84 % penyebab kebakaran pemukiman di Provinsi DKI Jakarta selama kurun waktu 2017 diakibatkan karena korsleting kabel listrik [3]. Salah satu hal yang paling sering menjadi penyebab kebakaran karena korsleting listrik adalah kebiasaan pengguna yang tidak melepaskan steker dari stopkontak setelah pemakaian alat-alat elektronik seperti *charger handphone*, televisi, VCD, dan sebagainya [4]. Misalnya pada Februari 2016, sebuah rumah di daerah Depok terbakar karena pemilik rumah lupa mencabut *charger handphone* ketika sedang keluar rumah [5]. Pada Juni 2018, kebakaran juga terjadi di Karanganyar, Solo. Kebakaran ini juga disebabkan oleh korsleting akibat pemilik lupa mencabut perangkat elektronik yang tidak sedang digunakan saat pergi berjualan [6]. Tidak hanya di Indonesia, pada tahun 2017 kasus kebakaran serupa

¹ Program Studi Teknik Elektro, Universitas Tarumanagara

terjadi di salah satu rumah susun di Hong Kong. Kebakaran ini terjadi akibat korsleting karena *charger handphone* yang kepanasan karena terus tercolok sepanjang malam sehingga membuat 12 orang harus dilarikan ke rumah sakit [7].

Untuk meminimalkan resiko-resiko tersebut, diperlukan suatu cara untuk mengendalikan stopkontak agar penggunaannya bisa menjadi lebih aman. Umumnya, stopkontak yang dijual di pasaran hanya dapat dikontrol menggunakan *switch on/off* dan kurang efisien karena pengguna harus berada di sekitar stopkontak untuk menyalakan/mematikannya. Terdapat juga beberapa inovasi rancangan untuk mengendalikan stopkontak, yaitu penggunaan *Bluetooth* untuk mengendalikan peralatan elektronik menggunakan *voice recognition* [8] dan pengontrol lampu listrik yang diatur menggunakan Android dengan tambahan saklar manual [9], namun penggunaannya masih terbatas dalam jangkauan *Bluetooth*.

Pada perancangan ini, dibuat suatu stopkontak yang dapat dikendalikan melalui aplikasi Android dan *web*. Semua peralatan elektronik dapat menggunakan stopkontak ini. Teknologi nirkabel seperti *Bluetooth* dan *Wi-Fi* dimanfaatkan untuk mendukung sistem ini agar dapat diakses melalui aplikasi Android dan *web* sehingga memudahkan pengguna untuk mengontrol stopkontak dari jarak dekat atau pun jarak jauh.

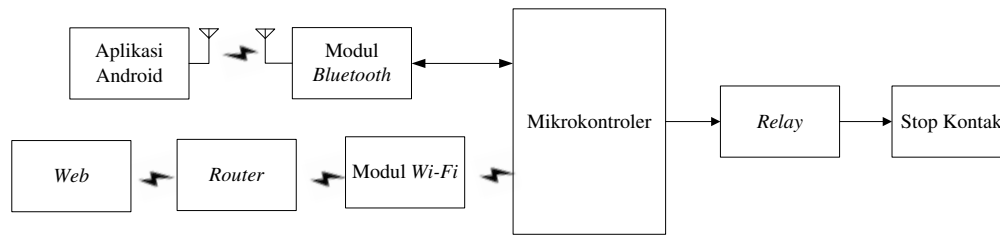
SMART-KONTAK

Smart-kontak merupakan stopkontak yang dapat dikendalikan dari jarak dekat dan jarak jauh dengan aplikasi Android melalui *Bluetooth* dan *web* melalui *Wi-Fi*. *User* dapat mematikan/menyalakan *smart*-kontak melalui aplikasi Android dan situs *web*, pada situs *web*, *user* juga dapat memasukkan *timer* untuk mengatur lamanya *smart*-kontak berada dalam kondisi *on*. Sisa waktu dari *timer* ini dapat dilihat melalui aplikasi Android maupun situs *web*. *Smart*-kontak dapat dilihat pada Gambar 1.

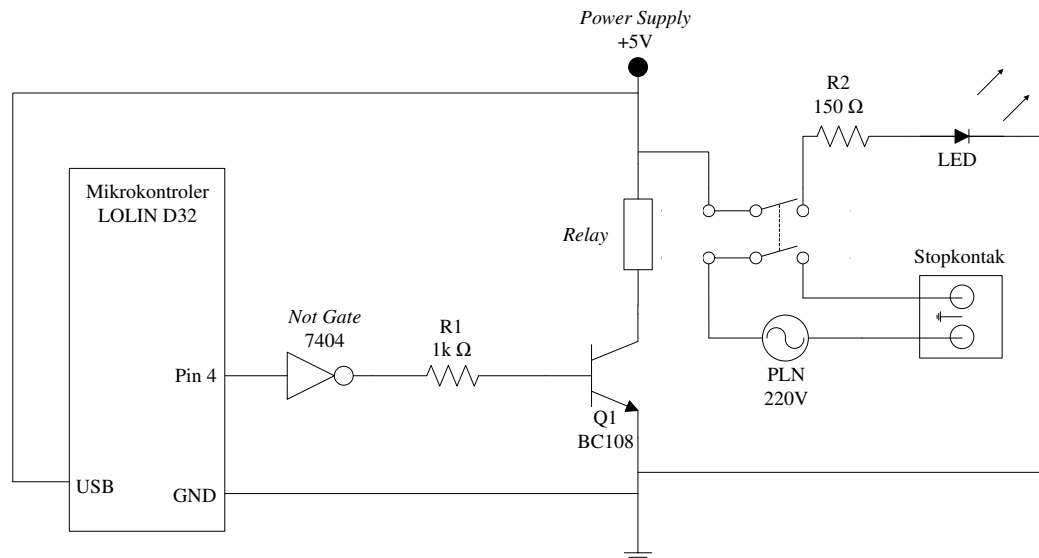


■ Gambar 1. *Smart*-kontak

Mikrokontroler yang digunakan pada perancangan sistem ini memiliki modul *Bluetooth* dan modul *Wi-Fi* yang sudah terintegrasi menjadi satu *board*. Mikrokontroler mengirim data status ke aplikasi Android dan menerima perintah dari aplikasi Android melalui modul *Bluetooth*, sedangkan mikrokontroler menerima perintah dari *web* melalui modul *Wi-Fi*. Modul *Wi-Fi* tersambung dengan *router* agar dapat berkomunikasi dengan *smart device* yang mengakses *web* melalui *Wi-Fi*. Setelah menerima perintah, mikrokontroler melakukan perintah, untuk menyalakan/mematikan stopkontak sesuai dengan perintah yang dikirim dari aplikasi Android maupun *web*. Mikrokontroler tidak secara langsung mengatur stopkontak, namun mikrokontroler mengatur *relay* yang kemudian dihubungkan dengan stopkontak untuk dapat mengatur *on/off* dari stopkontak. Diagram blok dan rangkaian dari perancangan dan implementasi sistem pengendali *smart*-kontak dengan aplikasi Android dan *web* dapat dilihat pada Gambar 2 dan Gambar 3.



■ **Gambar 2.** Diagram Blok Perancangan dan Implementasi Sistem Pengendali *Smart*-kontak dengan Aplikasi Android dan *Web*



■ **Gambar 3.** Rangkaian Sistem Pengendali *Smart*-kontak dengan Aplikasi Android dan *Web*

Aplikasi Android

Aplikasi Android merupakan *software* yang hanya dapat dijalankan pada *hardware* yang berbasis *Operating System Android*. Aplikasi Android dibuat menggunakan *Software Android Studio* dengan menggunakan bahasa Kotlin. Kotlin merupakan bahasa pemrograman modern yang bersifat *statically-typed* yang dapat dijalankan pada *platform Java Virtual Machine (JVM)*. Kotlin juga dapat dikompilasi ke dalam bentuk *JavaScript*. Kode atau *library* yang ditulis dengan bahasa Java dapat dibaca oleh Kotlin, begitu pula sebaliknya, kode atau *library* yang ditulis dengan bahasa Kotlin dapat dibaca oleh Java. Android ditulis menggunakan bahasa pemrograman Java, artinya bahasa pemrograman Kotlin dapat digunakan untuk membuat aplikasi Android.



■ **Gambar 4.** Tampilan Aplikasi Android untuk Memilih Perangkat *Bluetooth*

Aplikasi Android pada *smart-kontak* dapat menampilkan status *on/off* dan *remaining time*, serta memberikan *action* untuk menyalakan/mematikan *smart-kontak*. Pada saat aplikasi Android dibuka, *user* diminta untuk memilih perangkat *Bluetooth* lalu menyambungkannya. Tampilan aplikasi Android untuk memilih perangkat *Bluetooth* dapat dilihat pada Gambar 4. Setelah perangkat *Bluetooth* terhubung, aplikasi Android akan menerima data status dari modul pemroses lalu ditampilkan pada layar (Gambar 5). Jika *timer* sedang aktif, aplikasi Android akan menampilkan *countdown* sampai waktunya habis atau dihentikan. Pada tampilan Gambar 5, di bagian bawah terdapat *virtual button* yang berfungsi untuk menyalakan atau mematikan *smart-kontak*.



■ **Gambar 5.** Tampilan Utama Aplikasi Android

Situs Web

Situs *web* adalah kumpulan halaman *web* yang saling berhubungan yang dapat diakses menggunakan sebuah *browser*. *Browser* mengambil halaman yang diminta oleh pengguna dan menafsirkan kontennya. Protokol yang digunakan untuk mengambil halaman *web* disebut *HyperText Transfer Protocol* (HTTP) [10]. Setiap halaman dapat diidentifikasi dengan memberikan *Uniform Resource Locator* (URL) yang secara efektif berfungsi sebagai nama halaman yang dapat dicakup seluruh dunia. URL memiliki tiga bagian, yaitu protokol, nama domain komputer dimana halaman tersebut ada, dan lokasi yang mengindikasikan halaman yang spesifik.

HyperText Markup Language (HTML) adalah bahasa untuk mendeskripsikan bagaimana dokumen ditampilkan. Sebuah halaman *web* terdiri dari *head* dan *body* yang ada di dalam dokumen HTML. *Head* diartikan sebagai *header* dari tampilan *website* yang biasanya berisi nama *website*, sedangkan *body* merupakan isi utama dari halaman tersebut.

Pengguna dapat memberi *input* pada halaman *web* melalui *forms* yang ditulis dalam HTML. *Forms* menampilkan kotak-kotak dan tombol yang menjadi media pengguna untuk mengisi informasi atau membuat pilihan yang kemudian dikirim ke *server* menggunakan *web browser*. Protokol yang digunakan *web browser* untuk berkomunikasi dengan *web server* adalah HTTP. Metode *GET* adalah salah satu metode pengiriman *input* ke *web server*. *Web server* mengambil *input* dari URL, kemudian *input* disimpan atau diproses.

Situs web pada *smart*-kontak menampilkan status *on/off* dari *smart*-kontak, *set timer*, dan *action* untuk menyalakan/mematikan *smart*-kontak. Situs web akan menampilkan *homepage* ketika fitur *timer* sedang tidak aktif. *Homepage* akan menampilkan status dari *smart*-kontak dan *virtual button* untuk mengaktifkan *timer*, *set on*, dan *set off*. Ketika *timer* diaktifkan, situs web akan beralih ke halaman *countdown*. Pada halaman ini, sisa waktu akan ditampilkan dan dihitung mundur sampai waktunya habis atau dihentikan. Ketika waktu telah habis atau terhenti, web akan menampilkan pesan “*TIME OUT*”. User dapat mematikan *timer* dengan cara menekan *virtual button* “*ON*” atau “*OFF*” yang ada di bagian bawah. Tampilan *homepage* dapat dilihat pada Gambar 6 dan tampilan halaman *countdown* dapat dilihat pada Gambar 7.



■ Gambar 6. Tampilan Homepage



■ Gambar 7. Tampilan Halaman Countdown

Mikrokontroler LOLIN D32

LOLIN D32 dipilih sebagai modul pemroses pada perancangan sistem pengendali *smart*-kontak dengan aplikasi Android dan web karena pada mikrokontroler ini, modul *Bluetooth* dan *Wi-Fi* telah terintegrasi ke dalam satu *board* sehingga menjadi lebih praktis dibandingkan mikrokontroler lainnya. LOLIN D32 merupakan mikrokontroler yang berbasis ESP-32. LOLIN D32 memiliki 22 *digital I/O pin*, 6 *analog input pin*, dan 2 *analog output pin*. Semua pin I/O beroperasi pada tegangan 3,3V. LOLIN D32 memiliki memori sebesar 4 MB *flash* yang dapat digunakan untuk menyimpan program. LOLIN D32 dapat dilihat pada Gambar 8.



■ Gambar 8. LOLIN D32

Pemrograman Sistem

LOLIN D32 *compatible* dengan Arduino sehingga *user* dapat menggunakan *Arduino Software (IDE)* untuk menulis program. Program yang dibuat dalam perancangan sistem pengendali *smart*-kontak dengan aplikasi Android dan *web* adalah program untuk mengirim dan menerima data dari perangkat-perangkat yang terhubung, menyalakan/mematikan *smart*-kontak, serta melakukan perhitungan mundur ketika diberi perintah *set timer*.

Pertama kali, program menyalakan *relay* yang telah dihubungkan ke pin 4. Program mengaktifkan *Bluetooth* dan *Wi-Fi* pada modul pemroses dan memberikan nama "SMART-KONTAK". Potongan program pengaturan sistem dapat dilihat pada Gambar 9.

```
pinMode(4, OUTPUT);
//PIN 4 is connected to the relay switch that will be controlled.
digitalWrite(4, LOW);
//Switch is on at startup.
BTDevice.begin("SMART-KONTAK");
//Start Bluetooth service with name "SMART-KONTAK".
WiFi.softAP("SMART-KONTAK");
//Start Wi-Fi access point with name "SMART-KONTAK".
```

■ Gambar 9. Potongan Program Pengaturan Sistem

Modul pemroses dapat menyalakan/mematikan *smart*-kontak berdasarkan perintah yang didapatkan lewat situs *web* dan aplikasi Android. Ketika modul pemroses menerima data *set on* dari *web*, modul pemroses menyalakan *relay* dan membuat *smart*-kontak menjadi *on*. Potongan program pengaturan *relay set on* melalui situs *web* dapat dilihat pada Gambar 10. Modul pemroses juga dapat mematikan *relay* dan membuat *smart*-kontak menjadi *off* apabila modul pemroses menerima data *set off* dari *web*. Potongan program pengaturan *relay set off* melalui situs *web* dapat dilihat pada Gambar 11.

```
server.on("/SET_ON", HTTP_GET, [] (AsyncWebServerRequest *request){
//Web server action when web browser request force switch on page.
    enable = false; //Set timer status to disabled.
    digitalWrite(4, LOW); //Turn on switch.
    BTDevice.write('N'); //Tell Bluetooth device that the switch is on.
    delay(20);
    BTDevice.write('\r'); //End of message.
    request->send(200, "text/html", SET_ON); //Send the Set On page.
});
```

■ **Gambar 10.** Potongan Program Pengaturan *Relay Set On* Melalui Situs Web

```
server.on("/SET_OFF", HTTP_GET, [] (AsyncWebServerRequest *request){
//Web server action when web browser request force switch off page.
  enable = false; //Set timer status to disabled.
  digitalWrite(4, HIGH); //Turn off switch.
  BTDevice.write('F'); //Tell Bluetooth device that the switch is off.
  delay(20);
  BTDevice.write('\r'); //End of message.
  delay(20);
  request->send(200, "text/html", SET_OFF); //Send the Set Off page.
});
```

■ **Gambar 11.** Potongan Program Pengaturan *Relay Set Off* Melalui Situs Web

Ketika modul pemroses terhubung dengan perangkat *Bluetooth*, modul pemroses memeriksa perintah yang dikirim oleh perangkat *Bluetooth* sekaligus mengirim data status. Apabila fitur *timer* sedang aktif, modul pemroses juga mengirimkan data *remaining time* ke perangkat *Bluetooth*. Potongan program pengiriman data *remaining time* ke perangkat *Bluetooth* dapat dilihat pada Gambar 12, sedangkan potongan program pengiriman status *on/off* ke perangkat *Bluetooth* dapat dilihat pada Gambar 13.

Jika perangkat *Bluetooth* mengirimkan perintah *set off*, modul pemroses mematikan *relay* dan mengirimkan status *off* kepada perangkat *Bluetooth*. Namun, jika perangkat *Bluetooth* mengirimkan perintah *set on*, modul pemroses menyalakan *relay* dan mengirimkan status *on* kepada perangkat *Bluetooth*. Potongan program pengaturan *relay set off* dari perangkat *Bluetooth* dapat dilihat pada Gambar 14, sedangkan potongan program pengaturan *relay set on* dari perangkat *Bluetooth* dapat dilihat pada Gambar 15.

```
void sendTime() { //Send remaining time to Bluetooth device.
  int hh = 0, mm = 0;
  if(enable) { //If the timer enabled,
    hh = t_count / 60; //send the remaining time with
    mm = t_count % 60; //hours and minutes.
  } //If not, all zeros will be sent as hours and minutes.
  sprintf(hh_send, "%02d", hh); //Convert remaining hours to string.
  sprintf(mm_send, "%02d", mm); //Convert remaining minutes to string.
  BTDevice.write(hh_send[0]);
  delay(20);
  BTDevice.write(hh_send[1]);
  delay(20);
  BTDevice.write(mm_send[0]);
  delay(20);
  BTDevice.write(mm_send[1]);
  delay(20);
  //Send in format hhmm.
  BTDevice.write('\r'); //End of message.
  delay(20);}
}
```

■ **Gambar 12.** Potongan Program Pengiriman Data *Remaining Time* ke Perangkat *Bluetooth*

```

else if(command == 's') {
  if(digitalRead(4)) {
    BTDevice.write('F'); //Tell Bluetooth device that the switch is off.
    delay(20);
    BTDevice.write('\r'); //End of message.
    delay(20);}
  else {
    BTDevice.write('N'); //Tell Bluetooth device that the switch is on.
    delay(20);
    BTDevice.write('\r'); //End of message.
    delay(20);}}

```

■ **Gambar 13.** Potongan Program Pengiriman Status *On/Off* ke Perangkat *Bluetooth*

```

if(command == 'F') { //The switch off command.
  enable = false;
  digitalWrite(4, HIGH);
  BTDevice.write('F'); //Tell Bluetooth device that the switch is off.
  delay(20);
  BTDevice.write('\r'); //End of message.
  delay(20);}

```

■ **Gambar 14.** Potongan Program Pengaturan *Relay Set Off* dari Perangkat *Bluetooth*

```

else if(command == 'N') { //The switch on command.
  enable = false; //Set timer status to disabled.
  digitalWrite(4, LOW);
  BTDevice.write('N'); //Tell Bluetooth device that the switch is on.
  delay(20);
  BTDevice.write('\r'); //End of message.
  delay(20); }

```

■ **Gambar 15.** Potongan Program Pengaturan *Relay Set On* dari Perangkat *Bluetooth*

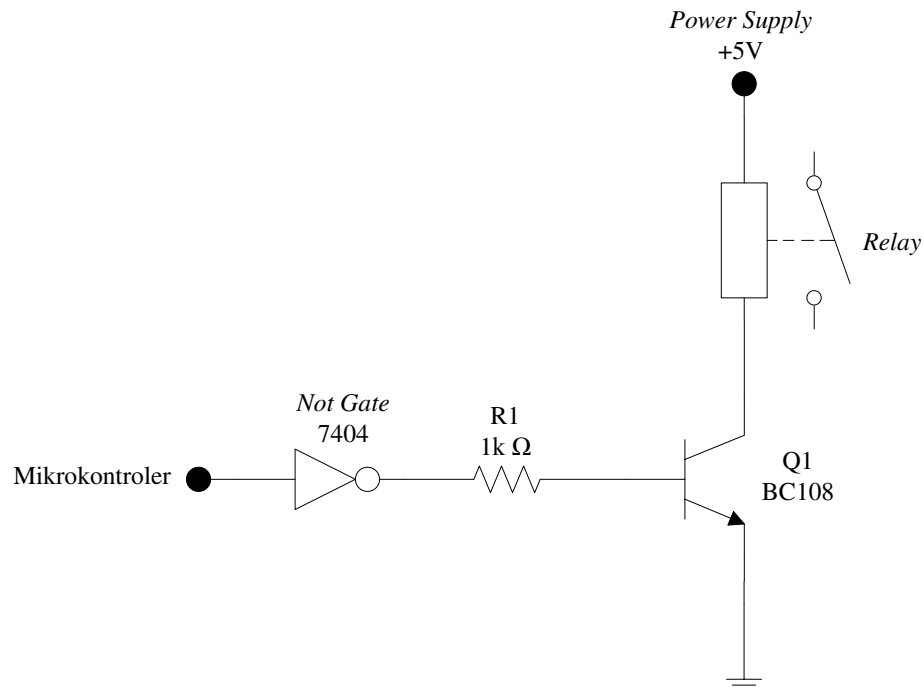
Pada saat fitur *timer* pada situs *web* aktif, modul pemroses melakukan *countdown* untuk menentukan lamanya *relay* berada dalam kondisi *on*. Selama fitur *timer* aktif, modul pemroses memeriksa perintah dari perangkat *Bluetooth*. Ketika modul pemroses menerima perintah *set on* maupun *set off* dari perangkat *Bluetooth*, *countdown* dihentikan dan modul pemroses menyalakan/mematikan *relay* sesuai dengan perintah yang dikirimkan oleh perangkat *Bluetooth*. Potongan program *countdown* pada modul pemroses dapat dilihat pada Gambar 16.

```
void loop() {
  BTComm(); //Check command from Bluetooth device [Timer disabled].
  while(enable) { //Countdown while timer is enabled.
    digitalWrite(4, LOW); //Switch is on while timer is enabled.
    while(t_count > 0) { //Countdown if remaining time not reached zero.
      //This block will be done in one minute.
      for(int sec = 0;sec < 60;sec++){
        BTComm(); //Check command from Bluetooth device [Timer enabled],
        delay(1000);} //Every one second.
      //If this block is done,
      t_count--; //Decrease remaining time by one minute.
      if(!enable) { //If the timer is suddenly disabled by user,
        break;}} //Stop the countdown by breaking the loop.
    if(!enable) { //Still ...,
      break;} //need to break the loop to stop the countdown.
    digitalWrite(4, HIGH); //Switch is switched off when countdown is over.
    enable = false; //Set timer status to disabled.
    BTDevice.write('F'); //Tell Bluetooth device that the switch is off.
    delay(20);
    BTDevice.write('\r'); //End of message.
    delay(20);}}
```

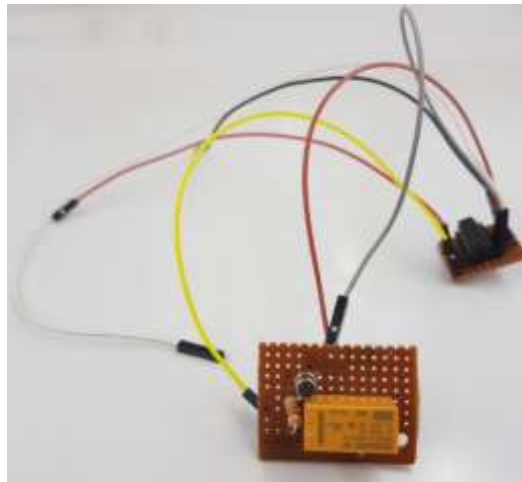
■ **Gambar 16.** Potongan Program *Countdown* pada Modul Pemroses

Driver Relay

Driver relay dibuat dengan rangkaian seperti pada Gambar 17. IC 7404 *not gate* digunakan pada modul *driver relay* sehingga rangkaian menjadi *active-low* yang berarti rangkaian menjadi aktif jika diberi *input low*. Modul *driver relay* dapat dilihat pada Gambar 18.



■ **Gambar 17.** Rangkaian *Driver Relay*



■ **Gambar 18.** Modul Driver Relay

Relay yang digunakan pada perancangan sistem pengendali *smart*-kontak dengan aplikasi Android dan *web* adalah *relay* dengan kode HRS2H-S DC5V. Relay HRS2H-S DC5V merupakan *relay* yang memiliki tegangan *coil* 5V_{DC} dan arus maksimal 1A pada tegangan 120V_{AC} dan 24V_{DC}. Relay ini merupakan *relay* tipe DPDT yang memiliki 2 pasang *output* sehingga dapat digunakan untuk mengontrol stopkontak dan LED. Relay HRS2H-S DC5V dapat dilihat pada Gambar 19.



■ **Gambar 19.** Relay HRS2H-S DC5V

IC 7404 *not gate* adalah IC yang memiliki 6 *inverter* yang berfungsi untuk membalikkan suatu keadaan logika. Jika *input* logika bernilai 1, *output* yang dihasilkan bernilai 0, dan ketika *input* logika bernilai 0, *output* yang dihasilkan bernilai 1. Tegangan *supply* dari IC ini adalah 5V dan 0V (*ground*). IC 7404 *not gate* dapat dilihat pada Gambar 20.



■ **Gambar 20.** IC 7404 Not Gate

Power Supply

Power supply Samsung digunakan dalam perancangan sistem pengendali *smart*-kontak dengan aplikasi Android dan *web*. *Power supply* ini berfungsi untuk memberi daya pada modul *driver relay* dan mikrokontroler. *Power supply* Samsung dapat mengubah tegangan 220V_{AC} dari PLN menjadi 5V_{DC} dengan arus maksimal 2A sehingga dapat digunakan sebagai sumber tegangan untuk mengaktifkan *relay* dan mikrokontroler. *Power supply*

Samsung merupakan *power supply* jenis *switching* yang ukurannya relatif lebih kecil dibandingkan *linear power supply*. *Power supply* Samsung dapat dilihat pada Gambar 21.



■ **Gambar 21.** *Power Supply* Samsung

Pengujian Sistem

Pengujian dilakukan setelah koneksi Bluetooth dan wifi, dengan beberapa kondisi:

- *On/off* menggunakan koneksi Bluetooth
- *On/off* menggunakan web
- Set timer untuk meng-*off*-kan melalui web
- *Timer* sedang berjalan, di-*off*-kan melalui Bluetooth
- *Timer* sedang berjalan, lalu diset ulang *timer*-nya

Semua pengujian berjalan dengan lancar dan sesuai dengan rencana rancangan, semua berjalan dengan baik.

KESIMPULAN

Kesimpulan yang dapat diambil dari perancangan dan implementasi sistem pengendali *smart*-kontak dengan aplikasi Android dan *web* adalah *smart*-kontak dapat dikendalikan melalui aplikasi Android dan situs *web*. Sistem ini dapat memberikan informasi berupa status dan *remaining time* pada aplikasi Android maupun situs *web*. Berdasarkan hasil pengujian semua modul dan keseluruhan sistem yang digunakan pada sistem ini berfungsi sesuai dengan yang direncanakan.

DAFTAR PUSTAKA

- [1]. W. Hendriani, *Menumbuhkan Online Resilience pada Anak di Era Teknologi Digital*, Prosiding Temu Ilmiah X Ikatan Psikologi Perkembangan Indonesia, 22-24 Agustus 2017, Semarang, Indonesia.
- [2]. A. Prastiantari, F. Hermin, dan Mulyono, “*Skopin (Stopkontak Pintar) Pengendali Arus Listrik Menggunakan Timer pada Stopkontak Berbasis Arduino*”, Program Studi Ilmu Komputer, FMIPA UNJ, 2018.
- [3]. B. Setiyo, “*Korsleting Listrik Penyebab Kebakaran Pada Rumah Tinggal Atau Gedung*”, Edu Elekrika Journal, Vol.3, Oktober 2014.
- [4]. I. Rachman dan L. Subiyanto, “*Implementasi Murray-Varley Bridge Berbasis Mikrokontroler untuk Mendeteksi Letak Hubung Singkat (Short Circuit) Kabel Listrik*”, Seminar MASTER, Surabaya, 2017.
- [5]. Pos Kota, diakses pada tanggal 6 Oktober 2018 pukul 20.34 WIB, <http://poskotanews.com/2016/02/14/hp-lupa-dicabut-dari-charger-rumah-ludes-terbakar/>.
- [6]. Solo Pos. “Rumah Warga Winong Karanganyar Terbakar Saat Ditinggal Jualan”, 2018. Internet: <http://soloraya.solopos.com/read/20180620/494/923258/rumah-warga-winong-karanganyar-terbakar-saat-ditinggal-jualan>. [6 Oktober 2018]
- [7]. Bertuah Pos. “Gara-gara *Charger* Telefon Rusun di Hongkong Hangus”, 2017.

Internet : <https://www.bertuahpos.com/international/garagara-charger-telefon-rusun-di-hong-kong-hangus.html>. [6 Oktober 2018]

- [8]. M. Rusdi dan A. Yani, “*Sistem Kendali Peralatan Elektronik Melalui Media Bluetooth Menggunakan Voice Recognition*”, Journal of Electrical Technology, Vol.3 No.1, Februari 2018.
- [9]. Suryono dan Supriati, “*Rancang Bangun Pengontrol Lampu Listrik Menggunakan Android Dilengkapi dengan Saklar Manual*”, Orbith, Vol. 13 No. 2, Juli 2017.
- [10]. A. Tanenbaum and D. Wetherall, *Computer Networks*, 5 ed., Boston: Pearson Education, Inc., 2011.