

Pengendali Saklar Listrik Melalui Ponsel Pintar Android

Vidy Masinambow (1), Meicsy E.I Najoan, ST,MT.(2), Arie S.M. Lumenta, ST,MT. (3)
 (1)Mahasiswa (2)Pembimbing 1 (3)Pembimbing 2
 yagavidy@gmail.com(1) meksy_najoan@unsrat.ac.id (2) arie.lumenta@gmail.com (3)
 Jurusan Teknik Elektro-FT, UNSRAT, Manado-95115

Abstract— *With the development of technology such as smart phones that are already widely owned as well as the development of network technology can be used to overcome the problem of electricity consumption. Electrical switches can be replaced by using the equipment controlled through a relay and network-based mikrokontler equipment so that it can connect to the smart phones which is mounted control program will be used to resolve the issue. Presence wifi shield equipment that can communicate with the microcontroller through serial communication so that information in the form of commands on / off switch that is sent from a smart phone with wireless connection to the equipment shield is a choice of solutions to address the replacement power switch. From the experimental results with the use of such technology, the Android-based smart phones that installed the control program can turn off the power switch or turn on electrical appliances from a considerable distance (as far as wifi connection is available).*

Keywords: *Android Smartphone, Electrical Switches, Microcontroller, Relays, Wifi Shield*

Abstrak- Dengan perkembangan teknologi seperti ponsel pintar yang sudah banyak dimiliki orang serta perkembangan teknologi jaringan dapat digunakan untuk mengatasi masalah penghematan listrik. Saklar listrik dapat diganti dengan menggunakan peralatan *relay* dan dikendalikan melalui peralatan mikrokontroler berbasis jaringan sehingga dapat terhubung ke ponsel pintar yang sudah terpasang program pengendali akan digunakan untuk mengatasi masalah tersebut. Adanya peralatan *Wifi Shield* yang dapat berkomunikasi dengan mikrokontroler melalui komunikasi serial sehingga informasi berupa perintah on/off saklar yang dikirim dari ponsel pintar dengan koneksi *wireless* ke peralatan *wifi shield* merupakan pilihan solusi untuk mengatasi pengganti saklar listrik. Dari hasil percobaan dengan menggunakan teknologi tersebut, ponsel pintar berbasis *android* yang terpasang program pengendali saklar listrik dapat mematikan atau menghidupkan peralatan listrik dari jarak yang cukup jauh (sejauh koneksi *wifi* yang tersedia).

Kata kunci: Mikrokontroler, Ponsel Pintar *Android*, *Relay*, Saklar Listrik, *Wifi Shield*.

I. PENDAHULUAN

Sebagian besar peralatan yang tercipta baik untuk keperluan rumah tangga, perkantoran, pertokoan maupun industri pemakaiannya menggunakan tenaga listrik, yang juga berarti kebutuhan akan listrik terus meningkat. Tak lepas dari itu persediaan listrik saat ini sangatlah terbatas, hal itu

menuntut kita untuk menghemat penggunaan listrik, itu dapat kita lakukan dengan menggunakan secara optimal sesuai dengan kebutuhan, yang sekaligus akan menghemat biaya pengeluaran penggunaan listrik kita.

Begitu juga dengan perkembangan ponsel pintar dengan sistem operasi *android* yang mendominasi pasaran ponsel dunia semakin banyak tersedia di pasaran dengan harga yang semakin terjangkau. Sistem operasi *android* sendiri bersifat sistem operasi *open source* yang dapat dimodifikasi sesuai dengan keperluan. Hal ini menumbuhkan minat *developer software* mobile untuk dapat membuat perangkat lunak yang bermanfaat dalam memenuhi kebutuhan manusia sehari-hari. Dengan memanfaatkan perangkat keras yang sudah terdapat di ponsel pintar *android* berupa pengaturan *wifi* maka *developer* dapat membuat aplikasi berdasarkan fungsi tersebut. Selain itu dalam proses pembuatan aplikasi tentunya tidak berbayar dan dapat dilakukan dengan bebas atau *open source*.

Perkembangan mikrokontroler yang semakin pesat membuat beberapa pengembang membuat suatu proyek *arduino* sebagai design *system* minimum mikrokontroler yang di buka secara bebas dengan modul yang menggunakan mikrokontroler AVR dan menggunakan seri yang lebih canggih, sehingga dapat digunakan untuk membangun sistem elektronika berukuran minimalis namun handal dan cepat. Hal ini menarik minat penulis untuk menggunakan mikrokontroler ini dan ponsel pintar *android* untuk membuat suatu sistem pengendali saklar listrik.

II. LANDASAN TEORI

A. Arduino

Arduino adalah sebuah produk design *system* minimum mikrokontroler yang di buka secara bebas. *arduino* menggunakan bahasa pemrograman C yang telah dimodifikasi dan sudah ditanamkan programmer *bootloader* yang berfungsi untuk menyambatkan antara *software compiler arduino* dengan mikrokontroler. Untuk koneksi dengan komputer menggunakan RS232 to TTL Converter atau menggunakan *Chip USB* ke serial converter seperti *FTDI FT232*. *Arduino* membuka semua sourcenya mulai dari diagram rangkain, jalur PCB, *software compiler*, dan *bootloadernya*.

Arduino Duemilanove merupakan salah satu jenis papan *arduino* yang mempunyai otak mikroprosesor *Atmega 328*, memiliki 14 digital *input/output* pin (dimana 6

diantaranya dapat digunakan sebagai *output* PWM), 6 *input* analog, *osilator* 16 MHz *kristal*, koneksi USB, soket listrik, header ICSP, dan tombol *reset*. Ini berisi semua yang diperlukan untuk mendukung *mikrokontroler*, hanya dengan menghubungkan ke komputer dengan kabel USB atau menghidupkannya dengan adaptor AC-DC atau baterai untuk memulainya.

B. Komunikasi Serial

Dua tipe dasar dari komunikasi serial adalah komunikasi secara *synchronous* dan *asynchronous*. Sebuah sistem komunikasi *synchronous*, kedua alat yang saling bertukar data harus selalu melakukan sinkronisasi untuk mengetahui status keaktifan *bit* data yang dikirim/diterima. Proses sinkronisasi ini akan terus berjalan walaupun pada saat itu tidak terdapat data yang sedang dikirim/diterima. Komunikasi serial *synchronous* ini dapat menghasilkan jumlah bit per *second*(bps) yang lebih besar dibandingkan *asynchronous* karena tidak memerlukan *start bit* maupun *stop bit*, namun tidak sepopuler komunikasi *asynchronous* karena memerlukan resource yang lebih besar dan IBM PC yang telah menjadi standar *Personal Computer* pada umumnya hanya mempunyai support untuk *Asynchronous Serial Communication*. *Asynchronous* yang berarti 'tanpa sinkronisasi' tidak memerlukan *line* untuk sinkronisasi, dan dengan ini dapat melakukan komunikasi full duplex dengan jumlah I/O line yang sangat minim. Sebagai pengganti proses sinkronisasi, sebuah *start bit* dan sebuah *stop bit* diperlukan untuk menandai awal dan akhir dari transmisi. Perlunya ditambahkan kedua bit ini membuat komunikasi secara *asynchronous* menjadi lebih lambat dibandingkan dengan komunikasi serial *synchronous*, tetapi dapat merupakan sebuah kelebihan dimana prosesor tidak akan terbebani dengan proses sinkronisasi. Sebuah *line asynchronous* yang sedang tidak melakukan pengiriman/penerimaan data akan mempertahankan nilai '1' yang juga disebut sebagai '*mark state*'. Dengan menggunakan nilai ini, sebuah sistem dapat melakukan pengenalan antara sebuah *line* yang sedang *idle* (tidak melakukan pengiriman/penerimaan data) dan sebuah *line* yang sedang tidak tersambung/*disconnected*. Setiap proses pengiriman akan dilakukan, sebuah *start bit* yang bernilai '0' akan dikirim, dan ketika terjadi perubahan dari nilai '1' ke '0', sistem penerima akan mengenali awal dimulainya penerimaan data. Protokol komunikasi serial *asynchronous* dapat dilihat pada gambar 1.

C. Transmission Control Protocol/Internet Protocol (TCP/IP)

Standar komunikasi data yang digunakan oleh komunitas internet dalam proses tukar-menukar data dari satu komputer ke komputer lain di dalam jaringan *Internet*. Protokol ini tidaklah dapat berdiri sendiri, karena memang protokol ini berupa kumpulan protokol (*protocol suite*). Protokol ini juga merupakan protokol yang paling banyak digunakan saat ini. Data tersebut diimplementasikan dalam bentuk perangkat lunak (*software*) di sistem operasi. Seperti pada perangkat

lunak, TCP/IP dibentuk dalam beberapa lapisan (*layer*). Seperti pada gambar model dari arsitektur TCP/IP (gambar 2) Dengan dibentuk dalam *layer*, akan mempermudah untuk pengembangan dan pengimplementasian. Antar *layer* dapat berkomunikasi ke atas maupun ke bawah dengan suatu penghubung *interface*. Tiap-tiap *layer* memiliki fungsi dan kegunaan yang berbeda dan saling mendukung *layer* diatasnya.

Model Client/Server

TCP adalah *peer-to-peer*, protokol yang bersifat *connection-oriented*. Tidak ada hubungan tuan dan budak (master/slave), tetapi banyak aplikasi yang bersifat client/server. *Server* adalah aplikasi yang memberikan pelayanan kepada user internet. *Client* adalah yang meminta pelayanan. Aplikasi bisa memiliki bagian *server* dan bagian *client*, dimana dapat berjalan secara bersamaan dalam 1 sistem. *Server* merupakan program yang dapat menerima permintaan (*request*), melakukan pelayanan yang diminta, kemudian mengembalikan sebagai *reply*. *Server* dapat melayani multi request bersamaan.

D. Wireless Local Area Network (WLAN)

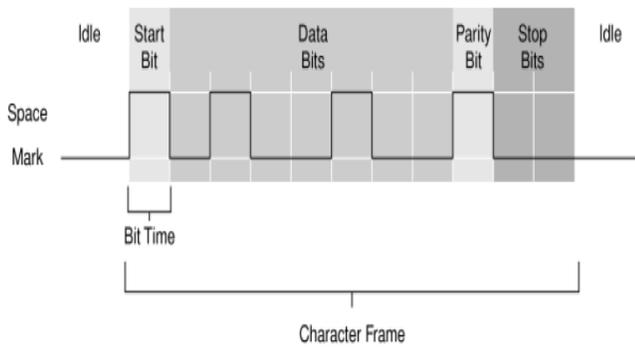
Teknologi *wireless* LAN melakukan proses pengiriman data dengan menggunakan frekuensi radio sebagai media perantaranya. Ada tiga pita (Band) frekuensi yang dapat digunakan secara bebas dalam dunia industri, medis, dan ilmiah, antara lain frekuensi 900 HZ, 2,4 GHz, dan 5,2 GHz. Diantara ketiga band, perangkat-perangkat *wireless* saat ini banyak menggunakan frekuensi 2,4 Ghz.

Wireless LAN kebanyakan memiliki peran sebagai *access layer*, sehingga digunakan sebagai *entry poin* ke dalam jaringan kabel. *Wireless* LAN bekerja pada *layer data Link* seperti umumnya *device access layer* lainnya.

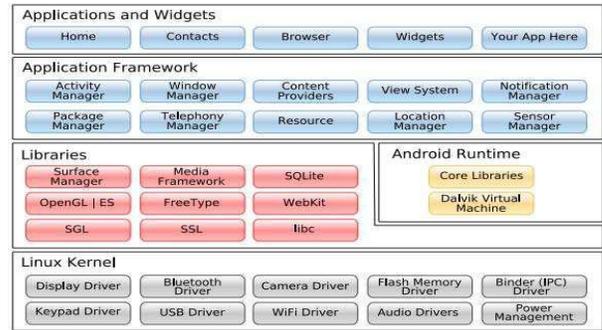
IEEE telah menetapkan protokol standar yang digunakan pada *device wireless*, yakni IEEE 802.11. Saat ini, ada beberapa standar 802.11, antara lain: 802.11a, teknologi menggunakan frekuensi 5 GHz dan dapat menghasilkan kecepatan 54 Mbps. 802.11b, Teknologi menggunakan frekuensi 2,4 GHz dan memiliki kemampuan transmisi hingga 11 Mbps. 802.11g, teknologi sama dengan 802.11b, menggunakan frekuensi 2,4 GHz, dan memiliki kemampuan transmisi 54 54 Mbps.

E. Android

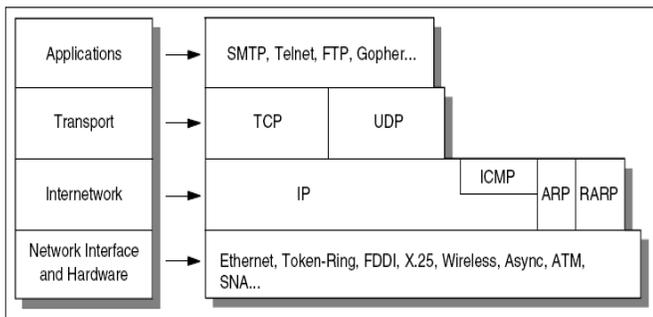
Android merupakan suatu sistem operasi ponsel yang berbasis linux. Android menyediakan *platform* yang bersifat *open source* bagi para pengembang untuk menciptakan sebuah aplikasi. Segi arsitektur sistem (gambar 3) , *android* merupakan sekumpulan *framework* dan *virtual mesin* yang berjalan di atas kernel linux. *Virtual machine* android bernama *dalvik virtual machine (DVM)*, *engine* ini berfungsi untuk menginpresentasikan dan menghubungkan seluruh kode mesin yang digunakan oleh setiap aplikasi dengan kernel linux. Sementara untuk *framework* aplikasi sebagian besar dikembangkan oleh google dan sebagian lain dikembangkan oleh pihak ketiga (*developer*). Beberapa *framework* yang



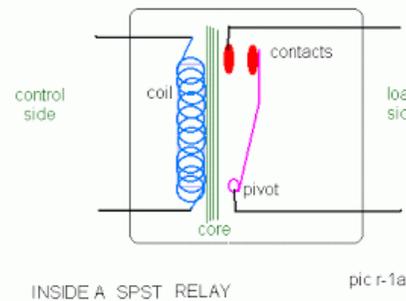
Gambar 1. Protokol Komunikasi Serial Asynchronous



Gambar 3. Arsitektur Sistem Operasi Android



Gambar 2. Detail dari model arsitektur TCP/IP



Gambar 4. Prinsip kerja relay

dikembangkan oleh *android* sendiri misalnya fungsi untuk *telephoni* seperti panggilan telepon, *sms*, dan *video call*.

Aplikasi yang dikembangkan diatas *Android* dibuat dengan kode *java* seperti halnya *Java 2 Mobile Edition* yang telah lama digunakan pada platform perangkat seluler umumnya. Namun secara siklus program memiliki perbedaan mendasar antara *Java 2 Mobile Edition* dengan *java* yang ada pada *android*. Kode *java* pada *android* lebih dekat dengan *Java 2 Standart Edition*. Dengan dukungan *software Development Kit (SDK)* dan *Application Programming Interface (API)* dari *google* memberikan kemudahan bagi pihak ketiga (*developer*) untuk membangun aplikasi yang dapat berjalan pada sistem operasi *Android*.

Selain itu terdapat metode baru dalam mengembangkan aplikasi di dalam sistem operasi *android* menggunakan *Native Development Kit (NDK)*. *NDK* ini memungkinkan *developer* untuk mengembangkan aplikasi di dalam sistem operasi *Android* menggunakan bahasa pemrograman *C* atau *C++*.

Android memiliki empat komponen. Meliputi *activity*, *Broadcast Receiver*, *service* dan *content provider*. Komponen aplikasi dapat disebut juga sebagai elemen-elemen aplikasi yang bisa dikembangkan pada *platform android*.

F. Relay

Relay adalah komponen elektronika berupa saklar elektronik yang digerakkan oleh arus listrik. *Relay* memiliki sebuah kumparan tengah-rendah yang lilitkan pada sebuah inti, terdapat sebuah *armatur* besi yang akan tertarik menuju inti apabila arus mengalir melewati kumparan. *Armatur* ini terpasang pada sebuah tuas pegas. Ketika *armatur* tertarik

menuju ini, kontak jalur bersama akan merubah posisinya dari kontak normal tertutup ke kontak normal terbuka.

Secara prinsip, *relay* merupakan tuas saklar dengan lilitan kawat pada batang besi (*solenoid*) di dekatnya (Gambar 4). Ketika *solenoid* dialiri arus listrik, tuas akan tertarik karena adanya gaya magnet yang terjadi pada *solenoid* sehingga kontak saklar akan menutup. Pada saat arus dihentikan, gaya magnet akan hilang, tuas akan kembali ke posisi semula dan kontak saklar kembali terbuka. *Relay* biasanya digunakan untuk menggerakkan arus/tegangan yang besar (misalnya peralatan listrik 4 ampere AC 220 V) dengan memakai arus/tegangan yang kecil (misalnya 0.1 ampere 12 Volt DC). *Relay* yang paling sederhana ialah *relay* elektromekanis yang memberikan pergerakan mekanis saat mendapatkan energi listrik. Secara sederhana *relay* elektromekanis ini didefinisikan sebagai alat yang menggunakan gaya elektromagnetik untuk menutup (atau membuka) kontak saklar dan sebagai Saklar yang digerakkan (secara mekanis) oleh daya/energi listrik.

Dalam pemakaiannya *relay* yang digerakkan dengan arus DC dilengkapi dengan sebuah dioda yang di-paralel dengan lilitannya dan dipasang terbalik yaitu *anoda* pada tegangan (-) dan *katoda* pada tegangan (+). Ini bertujuan untuk mengantisipasi sentakan listrik yang terjadi pada saat *relay* berganti posisi dari on ke off agar tidak merusak komponen di sekitarnya.

Konfigurasi dari kontak-kontak *relay* ada tiga jenis, yaitu: *Normally Open (NO)*, apabila kontak-kontak tertutup saat *relay* dicatu. *Normally Closed (NC)*, apabila kontak-kontak terbuka saat *relay* dicatu dan *Change Over (CO)*, *relay* mempunyai kontak tengah yang normal tertutup, tetapi ketika

relay dicatu kontak tengah tersebut akan membuat hubungan dengan kontak-kontak yang lain.

Penggunaan *relay* perlu memperhatikan tegangan pengontrolnya serta kekuatan *relay* men-switch arus/tegangan. Biasanya ukurannya tertera pada *body relay*. Misalnya *relay 12VDC/4 A 220V*, artinya tegangan yang diperlukan sebagai pengontrolnya adalah *12Volt DC* dan mampu men-switch arus listrik (maksimal) sebesar 4 ampere pada tegangan *220 Volt*. Sebaiknya *relay* difungsikan 80% saja dari kemampuan maksimalnya agar aman, lebih rendah lagi lebih aman. *Relay* jenis lain ada yang namanya *reedswitch* atau *relay lidi*. *Relay* jenis ini berupa batang kontak terbuat dari besi pada tabung kaca kecil yang dililitin kawat. Pada saat lilitan kawat dialiri arus, kontak besi tersebut akan menjadi magnet dan saling menempel sehingga menjadi saklar yang *on*. Ketika arus pada lilitan dihentikan medan magnet hilang dan kontak kembali terbuka (*off*).

III. METODOLOGI PENELITIAN

A. Tempat Penelitian

Dalam pelaksanaan tugas akhir ini penulis mengambil tempat penelitian pada Ruang Laboratorium Sistem Komputer (LSK), Jurusan Teknik Elektro, Fakultas Teknik Universitas Sam Ratulangi (UNSRAT) Manado

B. Bahan dan Peralatan

Alat dan peralatan yang digunakan dalam penelitian tugas akhir ini meliputi Perangkat Keras (*Hardware*) berupa *Arduino Duemilanove (Atmega 328)*, *Arduino wifi shield*, *Relay*, Adapter Daya, Kabel listrik, Kabel USB, Stop Kontak, Notebook Asus K45D. Untuk perangkat lunak (*Software*) yang digunakan berupa *Operating system Windows 7*, *Arduino Integrated Development Environment (Arduino IDE)*, *Eclipse bundle Android Development Tools (Eclipse ADT)*.

C. Prosedur Penelitian

Prosedur penelitian yang dilakukan dalam membuat pengendali saklar listrik melalui ponsel *android* adalah melakukan studi literatur melalui pencarian materi-materi yang berhubungan dengan pengendalian saklar listrik dan pembuatan aplikasi *android* ponsel *android* melalui buku-buku di perpustakaan dan internet. Selanjutnya penulis merancang sistem pengendali saklar listrik melalui ponsel pintar *android* secara umum. Kemudian penulis merancang komunikasi antara mikrokontroler dan Ponsel Pintar *Android*. Penulis juga merancang rangkaian saklar dan merancang program mikrokontroler. Selanjutnya merancang program aplikasi Pengendali saklar listrik di ponsel *android*. Prosedur terakhir adalah melakukan pengujian sistem dan evaluasi perangkat keras maupun perangkat lunak.

D. Perancangan Sistem

Sistem yang akan dibuat adalah mengendalikan saklar listrik melalui ponsel *android* dengan menggunakan jaringan

wireless. Jaringan *wireless* sebagai media pentransmisi *input* data dari ponsel *android* ke *mikrokontroler* yang telah terhubung dengan rangkaian *relay* yang berfungsi sebagai saklar listrik (Gambar 6). Ponsel pintar *Android* yang terinstal aplikasi pengendali saklar listrik akan mengirimkan perintah melalui jaringan *wireless* ke *arduino wifi shield*, kemudian *arduino wifi shield* akan mengirimkan kembali data ke *mikrokontroler* melalui komunikasi serial, data yang diterima *mikrokontroler* akan diproses *mikrokontroler* yang kemudian akan mengeluarkan output berupa tegangan 5 volt melalui pin digital *arduino* yang telah terhubung dengan *relay* dan akan memicu *relay* untuk mengalirkan listrik. Bagian-bagian dari perancangan sistem ini meliputi perancangan *Hardware* dan perancangan *Software*. Blok diagram sistem pengendalian dapat dilihat pada gambar 5.

Komunikasi data Ponsel Android dengan Wifi Shield

Komunikasi data ini menggunakan *Wireless LAN* dengan Protokol jaringan menggunakan *TCP* dengan IP *192.168.1.1* dan socket yang dibuka untuk komunikasi port *3000*. Protokol *TCP* digunakan karena mendukung komunikasi port yang mengizinkan sebuah server untuk mendukung beberapa sesi koneksi dengan *client* di satu jaringan. Port dapat mengidentifikasi aplikasi dan layanan yang menggunakan koneksi di dalam jaringan *TCP/IP*.

Komunikasi Serial Antara Wifi Shield Dengan Mikrokontroler

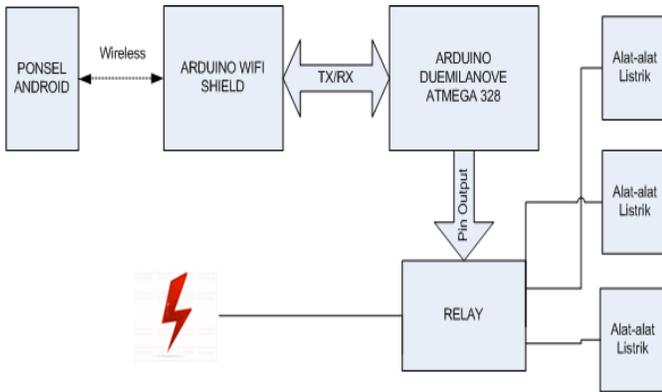
Proses komunikasi *wifi shield* dengan *mikrokontroler* menggunakan komunikasi serial dengan *baut rate 115200 bit/second*. Data yang diterima *wifi shield* berupa karakter *ASCII* diubah menjadi *Biner* oleh prosesor *wifi shield* agar dapat dikirim secara serial ke *mikrokontroler*. *Mikrokontroler* akan membaca data *ASCII* yang dikirim sebagai 1 byte data dalam bentuk *biner*. Data kemudian di proses sesuai dengan instruksi pada code program yang terdapat di *RAM flash memory*. Untuk pengontrolan saklar listrik ini digunakan sebanyak 18 karakter *ASCII* (TABEL I) sebagai bentuk instruksi ke mikrokontroler.

Rangkaian Saklar Dengan Mikrokontroler

Dalam implementasi saklar (gambar 6) pada penelitian ini, pin digital 8, 9, 10 pada *arduino* akan mengeluarkan *output* berupa tegangan 5 volt yang terhubung pada pin *NO* atau *normally open* pada *relay* sebagai input mengalirkan listrik.

Perancangan Program Mikrokontroler

Perancangan program *mikrokontroler* penulis menggunakan *software IDE arduino* untuk menulis program, mengkompilasi program dan mengunggah program ke dalam *mikrokontroler*. Program ditulis dengan bahasa *C* dan memanggil *library* program *IDE Arduino* dengan menggunakan fungsi *#include*. Pemanggilan *Library IDE arduino* yang digunakan adalah: *#include <SoftwareSerial.h>* berfungsi untuk memanggil *library* komunikasi serial pada *software arduino*, *#include <Time.h>* berfungsi untuk memanggil *library* perhitungan waktu pada *software arduino*, Dan *#include <TimeAlarms* berfungsi untuk memanggil *library alarm* pada *software arduino*. Gambar 7 menunjukkan flowchart pembuatan program mikrokontroler.



Gambar 5. Blok Diagram Sistem

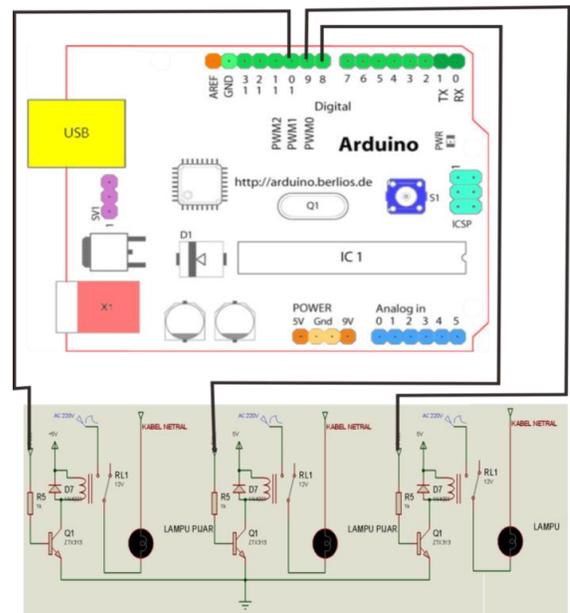
TABEL I
KARAKTER ASCII YANG DIGUNAKAN DALAM INSTRUKSI
MIKROKONTROLER

Char	Heksa	Biner	Instruksi
a	61	1100001	Pin 8 HIGH
b	62	1100010	Pin 9 HIGH
c	63	1100011	Pin 10 HIGH
d	64	1100100	Timer 30, Pin 8 HIGH
e	65	1100101	Timer 60, Pin 8 HIGH
f	66	1100110	Timer 30, Pin 9 HIGH
g	67	1100111	Timer 60, Pin 9 HIGH
h	68	1101000	Timer 30, Pin 10 HIGH
i	69	1101001	Timer 60, Pin 10HIGH
A	41	1000001	Pin 8 LOW
B	42	1000010	Pin 9 LOW
C	43	1000011	Pin 10 LOW
D	44	1000100	Timer 30, Pin 8 LOW
E	45	1000101	Timer 60, Pin 9 LOW
F	46	1000111	Timer 30, Pin 10 LOW
G	47	1001000	Timer 60, Pin 8 LOW
H	48	1001001	Timer 30, Pin 9 LOW
I	49	1001010	Timer 60, Pin 10 LOW

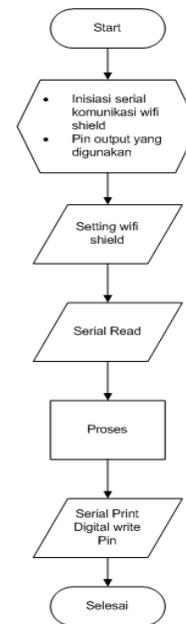
Perancangan Program Aplikasi Pengendali Ponsel Android

Pada program ponsel android dibuat suatu aplikasi yang dapat memudahkan interaksi antara ponsel android dengan mikrokontroller, aplikasi ini diberi nama Pengendali saklar listrik. Aplikasi dibuat dengan menggunakan program IDE Eclipse Android Development Tools, tahapan pembuatan aplikasi meliputi pembuatan Layout aplikasi dan pembuatan activity aplikasi.

Pada perancangan layout (gambar 8) aplikasi ini dibuat sebanyak empat buah layout, Layout pertama merupakan layar pembuka dan diberi nama Home dengan satu imagebutton yang ketika ditekan akan menuju ke layout kedua yaitu layout Menu. Pada Layout Menu memiliki dua imagebutton yaitu



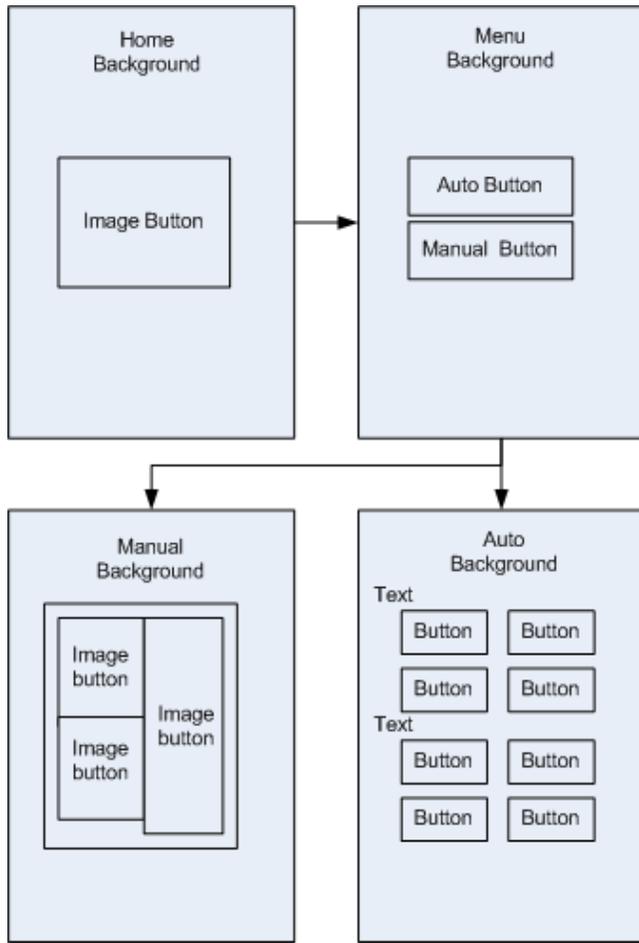
Gambar 6. Rangkaian Saklar Listrik



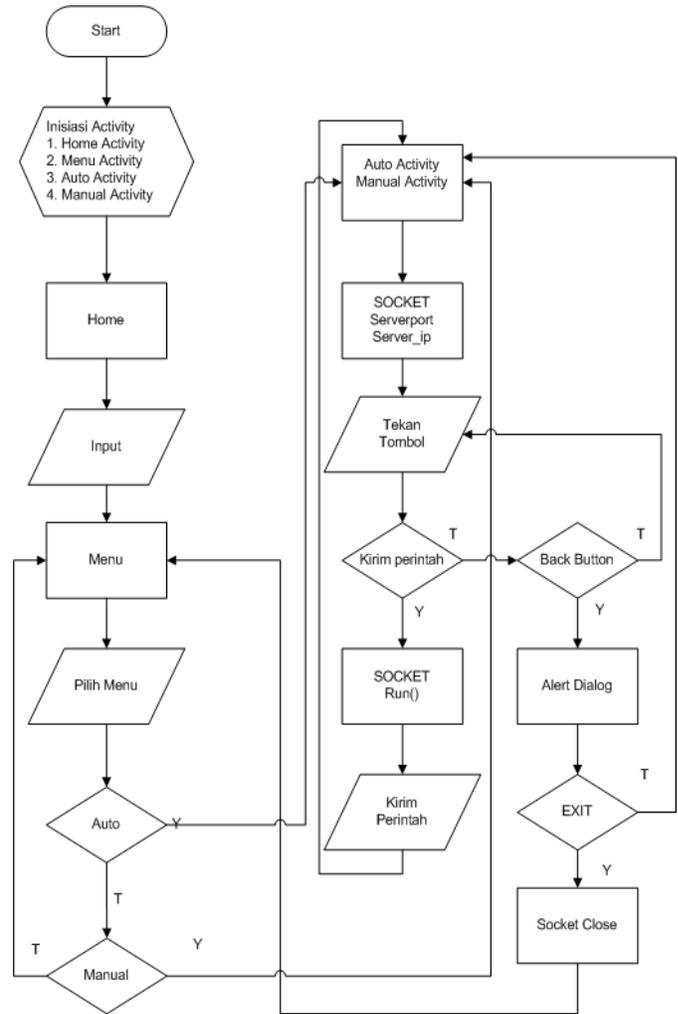
Gambar 7. Flowchart Program Mikrokontroler

Auto dan Manual, masing-masing imagebutton yang akan ditekan akan menuju ke layout berikutnya yang merupakan pilihan pengendalian yang akan digunakan. Layout Auto merupakan mode pengendalian saklar berbasis waktu, dimana pilihan waktu yang digunakan telah ditetapkan seberapa lama saklar akan menyala dan akan mati. Sedangkan pada layout manual merupakan mode pengendalian biasa dengan menggunakan Togglebutton. Togglebutton dipilih karena merupakan button berbasis switch yang dalam penggunaannya akan memiliki dua pilihan pengiriman data.

Pada perancangan activity ini dibuat sebanyak empat buah activity dengan menggunakan bahasa pemrograman java



Gambar 8. Perancangan Tampilan Aplikasi



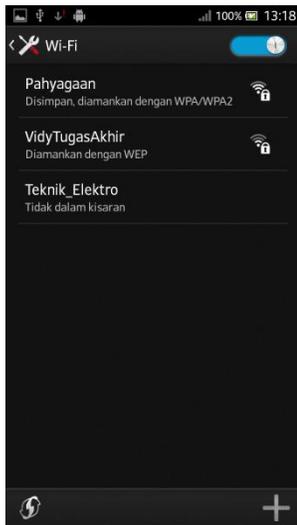
Gambar 9. Flowchart Aplikasi Pengendali pada Android

(Gambar 9). Setiap activity akan berjalan pada satu *layout* dengan perintah `setContentView(R.layout.manual)` pada `onCreate` atau pada saat aplikasi pertama dijalankan. Untuk perpindahan *activity* digunakan perintah pindah intent pada pendeklarasian `onClick` atau pada saat *button* di klik . Untuk *activity* utama yaitu *Auto Activity* dan *Manual Activity* dengan `mengimport libraries` pemrograman java yang terdapat *Java Application Programming Interface (API)* . Dasar pemrograman yang dipakai adalah *Socket Programming java* yang bertindak sebagai client. *Libraries java* yang digunakan dalam *activity* ini adalah sebagai berikut: `import java.io.BufferedWriter;`, `import java.io.IOException;`, `import java.io.OutputStreamWriter;`, `import java.io.PrintWriter;`, `import java.net.InetAddress;`, `import java.net.Socket;`, `import java.net.UnknownHostException;`.

Sedangkan *libraries android* yang digunakan dalam *auto activity* ini adalah `import android.app.Activity;`, `import android.app.AlertDialog;`, `import android.content.DialogInterface;`, `import android.os.Bundle;`, `import android.view.View;`, `import android.view.animation.Animation;`, `import android.view.animation.AnimationUtils;`, `import android.widget.Button;`, `import android.widget.TextView;`, `import android.widget.Toast;`.

IV. HASIL DAN PEMBAHASAN

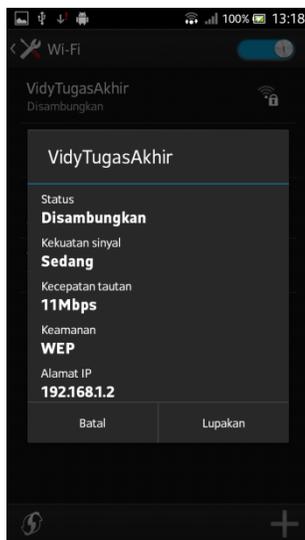
Untuk dapat mengendalikan *mikrokontroler* melalui ponsel pintar *android*, Ponsel *android* terlebih dahulu dihubungkan dengan jaringan *wireless* yang telah dibuat oleh *mikrokontroler*. Dengan mengakses pengaturan *wifi* pada ponsel *android*, jaringan *wireless* yang dibuat *mikrokontroler* akan dapat dilihat pada ponsel *android* dengan nama jaringan *VidyTugasAkhir*. Gambar 10 menunjukkan jaringan *wireless* yang telah dibuat oleh *mikrokontroler*. Jaringan *wireless* yang dibuat memiliki tingkat keamanan *Wired Equivalent Privacy (WEP)* sehingga untuk dapat terkoneksi dengan jaringan *wireless* ini harus memasukan sandi yang sesuai dengan yang dibuat *wifi shield* untuk proses autentifikasi menuju ke *wifi shield*. Setelah proses *autentifikasi* selesai maka ponsel *android* sudah terkoneksi dengan ponsel *android* dengan status tersambung dan memiliki IP yang diberikan oleh *wifi shield*. Gambar 11 menunjukkan informasi status jaringan yang telah terkoneksi dengan *wifi shield*, sedangkan pada gambar 12,13,14 dan 15 merupakan tampilan program pada saat *running* di ponsel pintar *android*



Gambar 10. Jaringan wireless VidyTugasAkhir yang dibuat mikrokontroler



Gambar 13. Tampilan Menu Activity



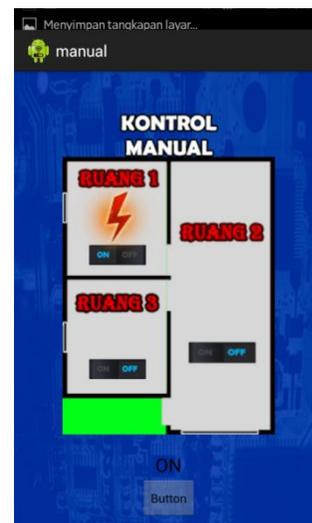
Gambar 11. Ponsel Pintar Android terkoneksi dengan Mikrokontroler



Gambar 14. Tampilan Auto Activity



Gambar 12. Tampilan Home Activity



Gambar 15. Tampilan Manual Activity

Auto Mode

Auto mode adalah pengendalian saklar listrik dengan menggunakan *timer* untuk menghubungkan atau memutuskan aliran listrik, hanya ada dua pilihan waktu yaitu 30 detik ON/OFF dan 60 detik ON/OFF sebagai bentuk demonstrasi pengendalian ini. Ketika file *button* di klik maka data akan dikirim ke *mikrokontroler*, dan diproses untuk kemudian dalam waktu 30 detik (Gambar 16) atau 60 detik (gambar 17) *mikrokontroler* akan mengeluarkan input yang memicu relay untuk mengalirkan atau memutuskan arus listrik. Pada percobaan ini lampu biru dan lampu merah masing-masing tombol yang ditekan 30 detik untuk biru dan 60 detik merah. Tampilan *button* saat di klik memakai animasi scale dan toast *button* sebagai tanda bahwa *button* di klik.



Gambar 16. Lampu biru 30 detik on yang menyala

Manual Mode

Manual Mode adalah pengendalian saklar listrik biasa tanpa *timer*, ketika layar disentuh arus listrik akan langsung dialirkan atau diputuskan oleh relay. Pengujian ini dilakukan pada saat tombol pada auto mode 60 ditekan kemudian langsung berpindah ke Manual mode, karna dalam coding program *mikrokontroler timer* yang digunakan tidak menggunakan *delay* maka *mikrokontroler* bisa mendapat instruksi lainnya. Dari hasil pengujian lampu yang dinyalakan tidak akan langsung menyala namun ada jeda satu detik dikarenakan proses pengiriman data dari media *wireless* diubah menjadi komunikasi serial agar dapat diterima oleh *mikrokontroler*. Gambar 18 menunjukkan lampu biru dan lampu merah bawah menyala dan gambar 19 semua lampu dinyalakan.



Gambar 18. Dua Lampu yang dinyalakan



Gambar 17. Lampu merah 60 detik on yang menyala



Gambar 19. Semua Lampu yang dinyalakan

V. KESIMPULAN

Dari penelitian diatas didapatkan kesimpulan bahwa pengendali saklar listrik dilakukan oleh *mikrokontroler* yang sudah diprogram dan terhubung ke Ponsel Pintar *Android* melalui koneksi *wireless*. Fungsi *relay* menggantikan saklar listrik, bagian pemacu terhubung ke mikrokontroler sehingga bisa memutuskan atau meneruskan arus listrik yang mengalir ke peralatan listrik (lampu). Perangkat Lunak *IDE Arduino* yang merupakan bawaan dari papan *arduino (board arduino)* dimana *mikrokontroler* berada berfungsi sebagai tempat menulis program, melakukan kompilasi dan mengunggah program hasil kompilasi ke *mikrokontroler*. Perangkat Lunak *Eclipse Bundle Android* adalah tools untuk membuat program aplikasi ponsel pintar berbasis *Android*. Program ini bila dijalankan di ponsel *Android*, akan mengirim sinyal ke *Wifi Shield* atau modul *wireless* kemudian *wifi shield* meneruskan sinyal ini ke mikrokontroler melalui komunikasi serial. Pengendalian dibuat dalam dua mode, yaitu *auto* dan *manual*. *Mode Auto* akan mengendalikan saklar listrik (*relay*) dengan menggunakan waktu (*timer*) sedangkan mode *manual* bersifat *real time* (jika tombol di tekan lampu menyala/mati).

- [2] D.A. Mellis, *Blink With Out Delay* , tersedia di: <http://www.arduino.cc/en/Tutorial/BlinkWithoutDelay>, diakses 7 Juli 2013.
- [3] F.R Tambaani, Perancangan waktu sistem tercepat perlombaan balap mobil, Skripsi Program S1 Teknik Elektro Universitas Sam Ratulangi, Manado, 2012.
- [4] Java Code Geeks., *Android Socket Example* , tersedia di: <http://examples.javacodegeeks.com/android/core/socket-core/android-socket-example/>, diakses 15 Juni 2013.
- [5] M. Margolis, *Arduino Cookbook*, O'Reilly Media, Sebastapol, 2011.
- [6] N.H. Safaat, *Pemrograman Aplikasi Android Smartphone dan Tablet PC Android*, Informatika, Bandung, 2012.
- [7] R. Tamada, *Android Working with XML Animation*, tersedia di <http://www.androidhive.info/2013/06/android-working-with-xml-animations/> , diakses 7 Juli 2013.
- [8] W. Simon, *30 Arduino Project Evil Jenius*, The McGraw-Hill Companies, Inc.Ebook, 2010.
- [9] W. Budiharto, *Aneka Proyek Mikrokontroler*, Graha Ilmu, Yogyakarta, 2011.

DAFTAR PUSTAKA

- [1] A.A. Huda , *24 Jam Pintar Pemrograman Android*, Andi Offset, Yogyakarta., 2012.