

LAMPU PIJAR PADA ANALOGI INSTALASI LISTRIK FASE-TIGA TERKENDALI MELALUI *SMARTPHONE* BERBASIS ANDROID TERHUBUNG *INTERNET* BERBANTUAN MIKROKONTROLER

Arief Goeritno^{1*}, Febby Hendrian², Ritzkal²

¹Jurusan/Program Studi Teknik Elektro, Fakultas Teknik, Universitas Ibn Khaldun Bogor

²Program Studi Teknik Informatika, Fakultas Teknik, Universitas Ibn Khaldun Bogor
Jl. Sholeh Iskandar km2, Kedung Badak, Tanah Sareal, Kota Bogor 16162 Jawa Barat
Telepon: (0251) 7160993/8380993

*Email: arief.goeritno@ft.uika-bogor.ac.id

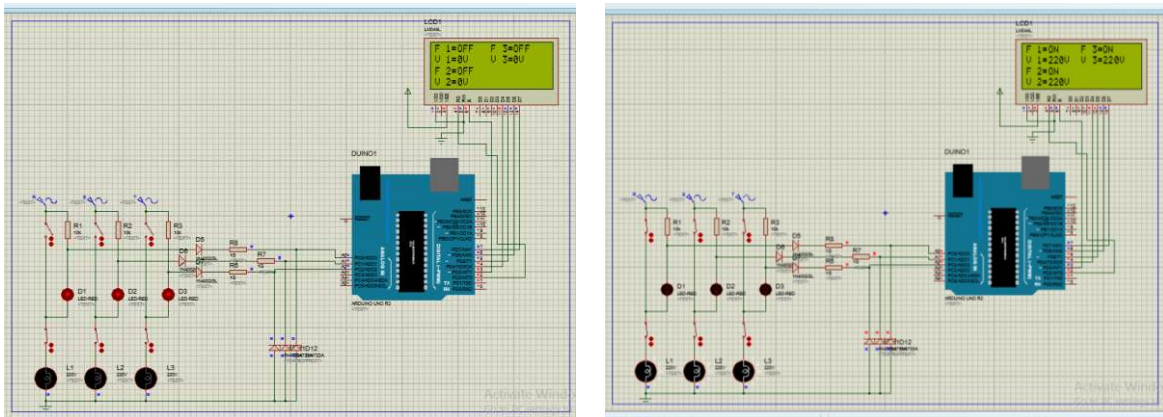
Abstrak

Telah dilakukan (a) perakitan (*assembling*) untuk perolehan bentuk fisis minimum system melalui keberhasilan *handshaking* secara (i) hardware dan (ii) software dan (b) pengukuran kinerja minimum system melalui (i) pembuatan tampilan kebutuhan sistem secara hardware dan software dan (ii) pemberian kondisi untuk pemantauan dan pengendalian pada analogi instalasi listrik fase-tiga. *Handshaking* secara hardware diperoleh melalui: (i) integrasi sejumlah peranti elektronika, yaitu: i) mikrokontroler Arduino UNO R3, ii) Arduino ethernet shield W5100, iii) Mikrotik RouterBrand, iv) modul relai, dan v) *smartphone* berbasis Android dan analogi instalasi listrik fase-3. *Handshaking* secara software diperoleh dengan tahapan berupa 4 (empat) langkah, yaitu: (i) pemasangan Arduino IDE versi 1.8.1 berbasis bahasa C untuk *compiling* dan *uploading* ke peranti mikrokontroler Arduino UNO R3, (ii) pembuatan aplikasi untuk *smartphone* berbasis Android, (iii) pengunduhan software untuk pemasangan aplikasi pada *smartphone* berbasis Android, dan (iv) pemasangan file dengan format apk pada *smartphone* berbasis Android. Minimum system terpasang dengan pengontrol mikro pada mikrokontroler Arduino UNO R3 dan perangkat *smartphone* berbasis Android sebagai pusat kendali. Keberadaan *tampikan* untuk pelaksanaan pengamatan terhadap pengukuran kinerja minimum system didasarkan kepada 2 (dua) kebutuhan system, sistem hardware dan software. Pengukuran kinerja dilakukan dengan pemantauan dan pengendalian pada analogi instalasi listrik fase-tiga, melalui 3 (tiga) pemberian kondisi berbeda pada jalur fase-R atau fase-S atau fase-T. Pemantauan dan pengendalian dengan pemberian kondisi telah sesuai dengan keberadaan minimum system.

Kata-kata kunci:, analogi instalasi listrik fase-tiga, Internet, lampu pijar, mikrokontroler, *smartphone* berbasis Android.

1. PENDAHULUAN

Keberadaan sistem elektronis berbasis sejumlah komponen elektronika atau mikrokontroler (pengontrol mikro), maupun sistem elektronis berbantuan komputer personal telah dimanfaatkan untuk pemantauan instalasi listrik, melalui *miniature circuit breaker* atau *MCBs* (Goeritno, 2016). Tahapan uji verifikasi terhadap prototipe sistem elektronis berbasis mikrokontroler Arduino UNO R3 telah dilakukan untuk pemantauan instalasi listrik fase-tiga (Johan, 2016), sebagai upaya perolehan bentuk fisis rangkaian elektronika terintegrasi. Uji verifikasi dilakukan melalui simulasi terhadap rangkaian terintegrasi (Johan, 2016) berbantuan aplikasi *Proteus* (Proteus2000, 1998). Diagram pengawatan integrasi sistem berbantuan aplikasi *Proteus* untuk pemantauan kondisi pada analogi instalasi listrik fase-tiga (Johan, 2016), seperti ditunjukkan pada Gambar 1.



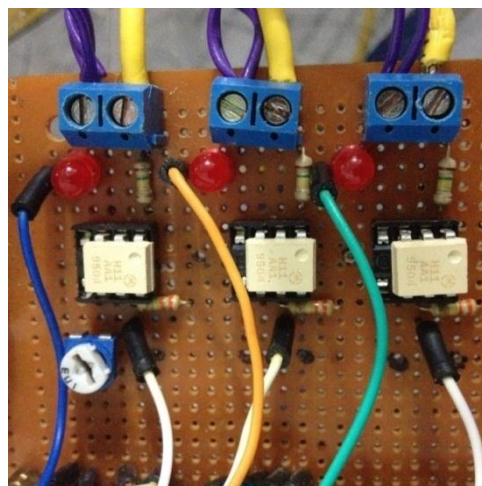
(a) fase-R, fase-S, dan fase-T kondisi “off”

(b) fase-R, fase-S, dan fase-T kondisi “on”

Gambar 1 Diagram pengawatan integrasi sistem berbantuan aplikasi *Proteus* untuk pemantauan kondisi pada analogi instalasi listrik fase-tiga

Berdasarkan Gambar 1 ditunjukkan, bahwa pemantauan terhadap analogi instalasi listrik fase-tiga didasarkan kepada keberadaan algoritma dan sintaks (Johan, 2016) untuk pemrograman terhadap mikrokontroler Arduino UNO R3 (Banzi, 2011; Durfee, 2011; Margolis, 2011; Barrett, 2012; Banzi and Shiloh, 2015). Algoritma tersebut meliputi tiga hal utama, yaitu tampilkan indikasi sensor-transduser, ambil dan kirim data, dan keluaran (Johan, 2016; Goeritno, 2016).

Penyempurnaan terhadap subsistem sensor-transduser pada prototipe sistem elektronis berupa pemanfaatan terhadap sejumlah komponen elektronika (Goeritno, 2016), yaitu *optocoupler*, resistor, transistor BD139, *Light Emiting Diode (LED)*, dan *solid state relay (SSR)*. Subsistem sensor-transduser hasil penyempurnaan tersebut diintegrasikan terhadap subsistem mikrokontroler Aduino UNO R3 dan *Liquid Crystal Display (LCD)* untuk pembentukan prototipe sistem elektronis (Goeritno, 2016). Bentuk fisis sensor-transduser pada sistem elektronis (Goeritno, 2016), seperti ditunjukkan pada Gambar 2.

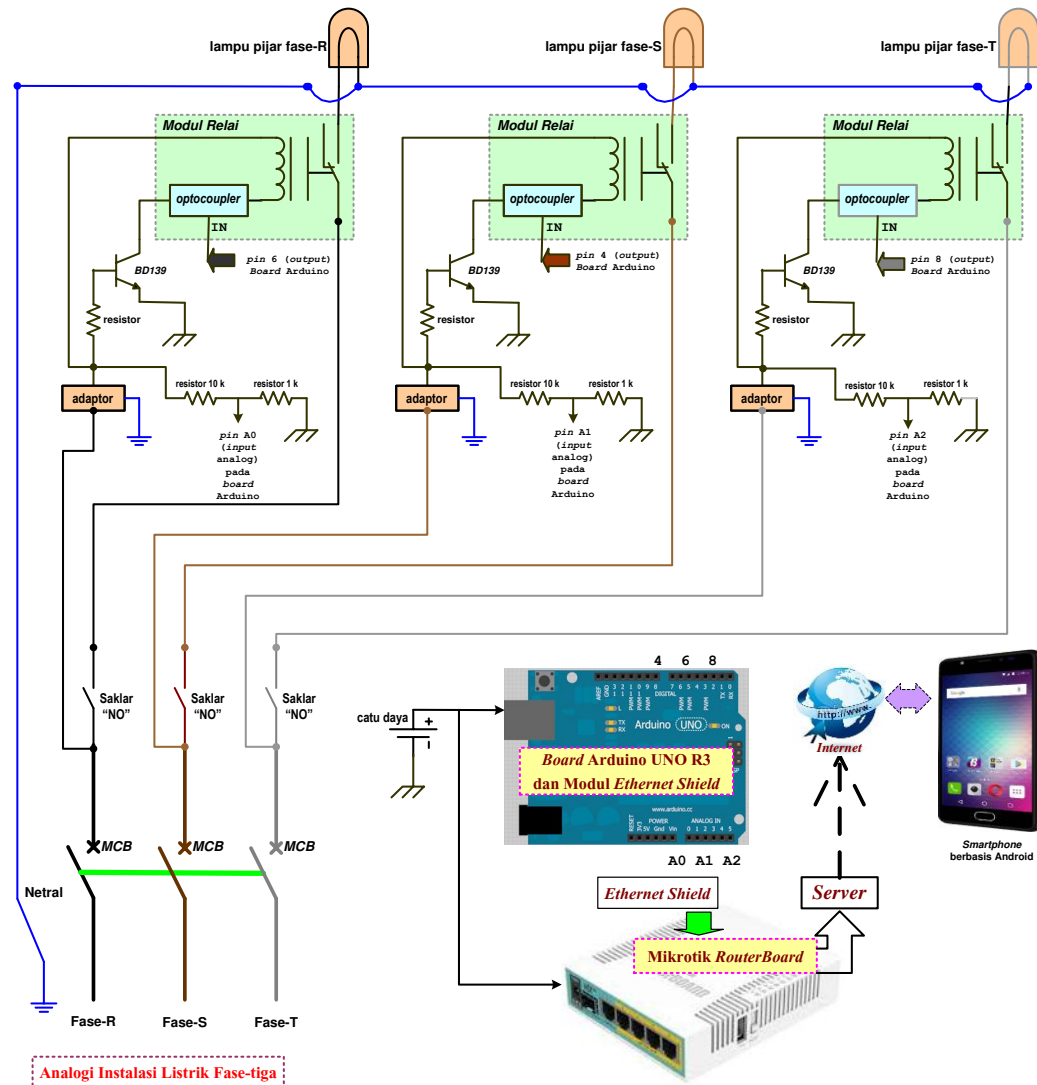


Gambar 2 Bentuk fisis sensor-transduser pada sistem elektronis

Berdasarkan Gambar 2 ditunjukkan, bahwa sensor-transduser pendeteksi kondisi difungsikan untuk setiap perubahan yang terjadi di *miniature circuit breaker (MCB)* satu *pole* sebagai analogi masing-masing fase pada sistem instalasi listrik fase-tiga. Indikator pada subsistem sensor-transduser berupa *LED*. Saat kondisi *LED* menyala (ON), kondisi *MCB* dalam keadaan tidak terhubung (OFF), sedangkan saat kondisi *LED* tidak menyala (OFF), kondisi *MCB* dalam keadaan terhubung (ON). Keluaran sensor-transduser untuk masing-masing fase-tunggal pada analogi instalasi listrik fase-tiga sebagai masukan yang tersambung ke sistem mikrokontroler Arduino UNO R3 dan untuk ditampilkan pada *LCD*.

Berdasarkan uraian tersebut, diperlukan rumusan masalah pada penelitian ini, yaitu (a) perakitan (*assembling*) terhadap sejumlah perangkat elektronis dan (b) pengukuran kinerja terhadap *minimum system*. Perakitan merupakan upaya untuk perolehan bentuk fisis *minimum system* melalui keberhasilan *handshaking* secara *hardware* dan *software*. Keberhasilan *handshaking* secara *hardware* dilakukan melalui pengintegrasian sejumlah perangkat elektronika. Keberhasilan *handshaking* secara *software* dilakukan melalui pemasangan perangkat lunak (*software*) pada board Arduino (Evans, 2011; Purdum, 2012) berbasis bahasa pemrograman C (Ritchie, 1993; Kernighan and Ritchie, 1988) dan aplikasi monitoring berbasis *app inventor* pada komputer personal, pemasangan aplikasi monitoring pada *smartphone* berbasis Android (Purdy, 2011) dengan sistem pengoperasian (*Operating System, OS*) *mobile* (Nosrati, 2012), pembuatan aplikasi untuk ditanamkan pada board Arduino UNO R3 dan *smartphone* berbasis Android melalui jaringan Internet (Forouzan, 2010, Hermawan, 2016) untuk pemantauan dan pengendalian beban listrik pada analogi instalasi listrik fase-tiga. Perolehan kinerja *minimum system* untuk pemantauan dan pengendalian terhadap beban listrik (dipilih lampu pijar) pada analogi instalasi listrik fase-tiga, dilakukan melalui (i) pemenuhan terhadap kebutuhan sistem secara *hardware* dan *software* dan (ii) pelaksanaan pengamatan terhadap pemberian kondisi pada fase-R atau fase-S atau fase-T sebagai bagian dari analogi instalasi listrik fase-tiga.

Diagram pengawatan analogi instalasi listrik fase-tiga dan *minimum system*, seperti ditunjukkan pada Gambar 1.



Gambar 1 Diagram pengawatan analogi instalasi listrik fase-tiga dan *minimum system*

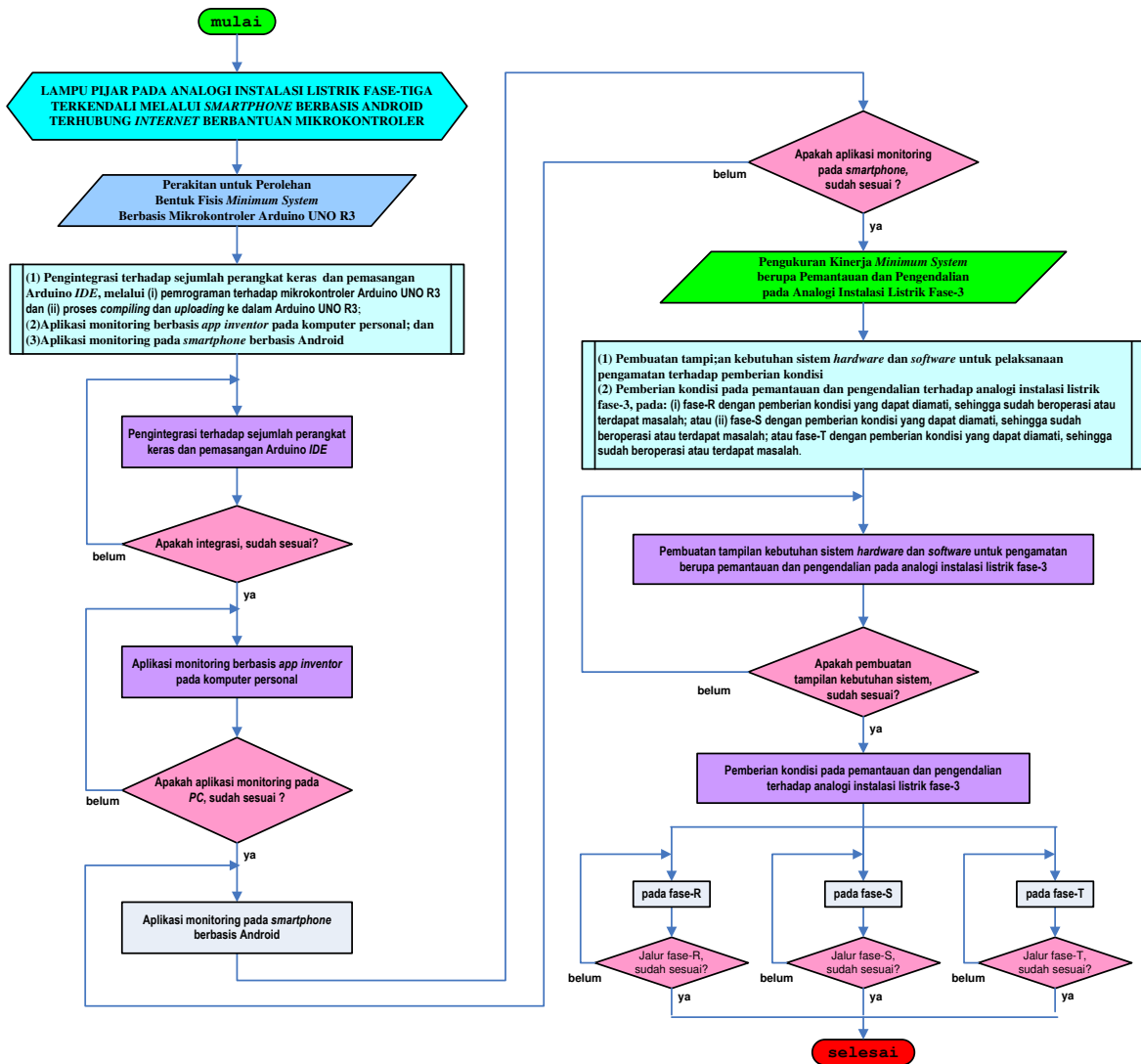
Berdasarkan Gambar 1 ditunjukkan, bahwa terdapat sejumlah perangkat elektronika penting yang meliputi *board* Arduino UNO R3, *ethernet shield*, mikrotik *RouterBoard* untuk koneksi ke *Internet*, dan *smartphone* berbasis Android. *Board* Arduino UNO R3 dengan *chip* mikrokontroler Atmega328 (ATMEL Corporation, 2009) dengan 14 *pin* masukan/keluaran (*input/output*, *I/O*) digital dan 6 *pin* masukan analog. Untuk pemrograman cukup digunakan koneksi *Universal Serial Bus (USB)* tipe A ke tipe B (Axelson, 2005, Axelson, 2009), sebagaimana banyak digunakan untuk koneksi ke pencetak (*printer*). Keberadaan komunikasi antara *board* Arduino dan telepon pintar (*smartphone*) pun sudah sangat mudah dan banyak cara, salah satu cara praktis, adalah penggunaan kabel *USB* dengan aplikasi Android atau Android *USB Host Application Programming Interface (API)* (di Cerbo, 2011), agar *smartphone* berbasis Android dapat saling bertukar data atau berkomunikasi dengan mikrokontroler (Monperrus, et al., 2012; Maalej and Robillard, 2013). Cara konvensional untuk penggunaan *tablet* atau *smartphone* berbasis Android, adalah sebagai periferil *USB* yang terhubung ke *host USB* (misalnya *PC*) dan penyinkronan data dan *file* media. Namun perangkat berbasis Android juga dapat berperilaku sebagai *host USB* sejak level *API* 12 (Android versi 3.1). Penggunaan terhadap fitur tersebut, maka dapat dilakukan keterhubungan dengan *mouse USB*, *keyboard*, atau *memory stick* dengan *smartphone* berbasis Android (Monperrus, et al., 2012; Maalej and Robillard, 2013). Untuk pemasangan secara fisis pada perangkat periferil, diperlukan adaptor kabel khusus yang disebut *USB On The Go (USB OTG)*. Bentuk fisis yang paling umum, tercakup konektor *USB* tipe-A bentuk *female* di salah satu ujung tempat perangkat periferil tersambung dan konektor *USB* tipe *micro* bentuk *male* di ujung lain terhubung ke perangkat berbasis Android (Monperrus, et al., 2012; Maalej and Robillard, 2013).

Arduino *Ethernet* adalah papan mikrokontroler berdasarkan ATmega328 (ATMEL Corporation, 2009), dengan 14 *pin* masukan/keluaran (*input/output*) digital, 6 masukan analog diberi label A0-A5, osilator kristal 16 MHz., konektor RJ45, kabel koneksi ke catu daya, *header In-Circuit Serial Programming (ICSP)*, dan tombol *reset* (Arduino, 2012). Keberadaan *IP* telah secara aktual sebagai pembawa data, namun *IP* ini telah serahkan penyampaian data tersebut kepada protokol lebih tinggi (Forouzan, 2010), yaitu *Transmission Control Protocol (TCP)*. *Transmission Control Protocol (TCP)* merupakan sebuah protokol standar sebagai pengatur segala pertukaran data. Keberadaan *TCP* merupakan upaya, agar setiap perangkat komputer dengan alamat *IP* atau *IP address* yang dapat diketahui dan dikenali, sehingga *IP* ini dengan dua cakupan berupa *source* dan *destination Internet Protocol Address* (Forouzan, 2010). Berdasarkan keberadaan sejumlah peranti elektronika yang diperlukan, sehingga diperlukan tujuan penelitian, yaitu: (1) memperoleh bentuk fisis *minimum system* berbasis mikrokontroler Arduino UNO R3 dan (2) memperoleh hasil pengukuran kinerja *minimum system* berupa pemantauan dan pengendalian terhadap analogi instalasi listrik fase-tiga melalui *smartphone* berbasis Android.

2. METODOLOGI

Untuk keberlangsungan metode penelitian, sejumlah bahan penelitian diperlukan berupa perangkat keras dan lunak. Perangkat keras, meliputi: (i) *board* Arduino UNO R3, (ii) modul *ethernet shield*, (iii) sejumlah modul relai, (iv) sejumlah resistor dan transistor, (v) sejumlah kabel *pin*, (vi) *breadboard*, (vii) sejumlah adaptor, (viii) sejumlah lampu pijar, (ix) *miniature circuits breakers (MCBs)*, (x) catu daya (*power supply*), dan (xi) mikrotik *RouterBoard*. Perangkat lunak berupa Arduino *Integrated Development Environment (Arduino IDE)* versi 1.8.1 yang tersedia secara gratis dan dapat diperoleh secara langsung pada halaman resmi Arduino di <https://www.arduino.cc/en/main/software>, bahasa pemrograman C, dan aplikasi *online* berbasis *app inventor* di <http://ai2.appinventor.mit.edu>. Selain bahan penelitian, diperlukan alat-alat penelitian berupa: (a) *web browser* Google Chrome, (b) *smartphone* berbasis Android merek Samsung Note 3 SM-N900, (c) *laptop* Core2Duo P7370 2.00 GHz., dan (d) alat ukur terhadap tegangan listrik (*voltmeter*).

Langkah-langkah pada metode penelitian didasarkan kepada tujuan penelitian. Diagram alir metode penelitian, seperti ditunjukkan pada Gambar 2.



Gambar 2 Diagram alir metode penelitian

Berdasarkan Gambar 2 ditunjukkan, bahwa langkah-langkah berkaitan dengan perakitan untuk bentuk fisis *minimum system* berbasis mikrokontroler Arduino UNO R3 dan pengukuran kinerja *minimum system* untuk pemantauan dan pengendalian terhadap analogi instalasi listrik fase-tiga. Proses perakitan untuk perolehan bentuk fisis *minimum system*, yaitu: (a) pengintegrasian terhadap sejumlah perangkat keras dan pemasangan aplikasi Arduino atau Arduino IDE (Banzi, 2011; Durfee, 2011; Margolis, 2011; Barrett, 2012; Banzi and Shiloh, 2015), dimana pengintegrasian ini dilakukan melalui pemrograman terhadap mikrokontroler Arduino UNO R3 dan proses *compiling* dan *uploading* ke dalam Arduino UNO R3; (b) pembuatan aplikasi monitoring berbasis *app inventor* dengan *web browser* secara *online*; dan (c) pengiriman *file* dari komputer personal dan pemasangan aplikasi monitoring pada *smartphone* berbasis Android. Pelaksanaan pengukuran kinerja *minimum system* untuk pemantauan dan pengendalian terhadap analogi instalasi listrik fase-tiga, berupa: (a) pembuatan tampilan kebutuhan sistem *hardware* dan *software* untuk pelaksanaan pengamatan terhadap pemberian kondisi dalam upaya pemantauan dan pengendalian pada analogi instalasi listrik fase-tiga dan (b) pelaksanaan pengamatan berupa pemantauan dan pengendalian terhadap analogi instalasi listrik pada: (i) fase-R atau (ii) fase-S atau fase-T dengan pemberian kondisi yang dapat diamati, sehingga dapat diketahui sistem sudah beroperasi atau terdapat masalah.

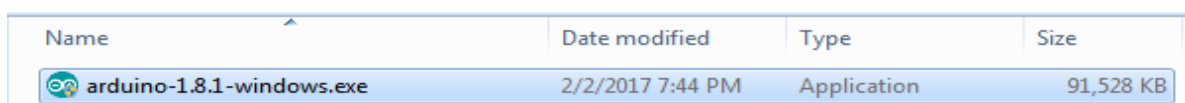
3. HASIL DAN PEMBAHASAN

3.1. Bentuk Fisis *Minimum System* Berbasis Mikrokontroler Arduino UNO R3

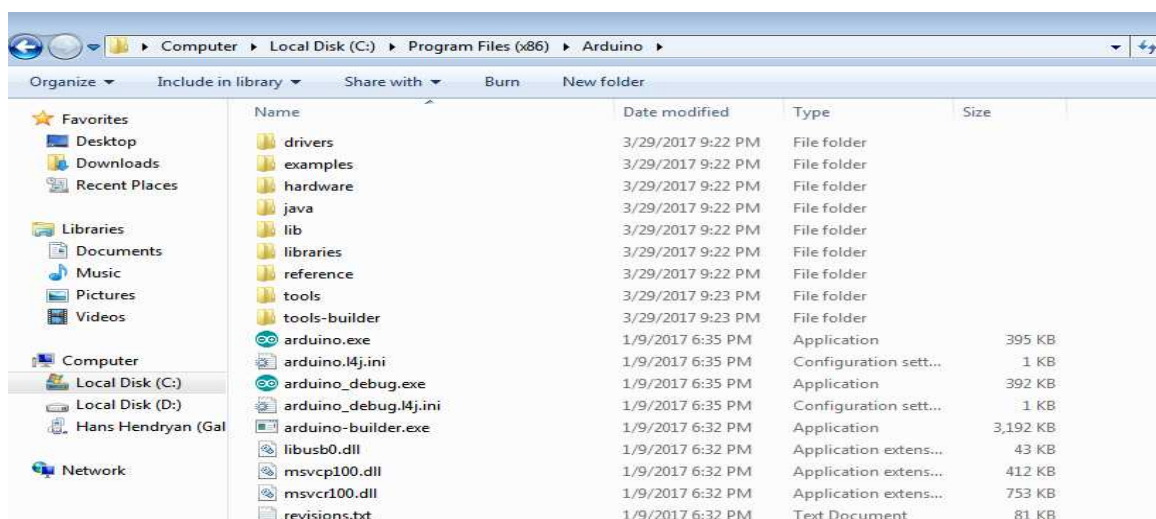
Bentuk fisis *minimum system* pada penelitian ini didasarkan kepada tujuan penelitian, sedangkan hasil diperoleh didasarkan kepada metode penelitian. Hasil diperoleh berupa i) integrasi terhadap sejumlah perangkat keras dan pemasangan perangkat lunak (*software*, aplikasi) Arduino atau Arduino *IDE*, ii) aplikasi pemantauan (monitoring) berbasis *app inventor* pada komputer personal, iii) dan aplikasi monitoring pada *smartphone* berbasis Android.

3.1.1 Integrasi terhadap sejumlah perangkat keras dan pemasangan aplikasi Arduino *IDE*

Integrasi terhadap sejumlah perangkat keras dan pemasangan aplikasi Arduino merupakan bentuk keberhasilan *handshaking* secara *hardware* dan *software* yang dilakukan melalui sejumlah tahapan, dimulai dari pengunduhan berupa *file* Arduino 1.8.1 melalui *web* (www.arduino.cc) yang digunakan untuk proses *uploading* dan *compiling* sistem pemantauan terhadap analogi instalasi listrik fase-3. Tampilan hasil pengunduhan *file* dan pemasangan Arduino *IDE* versi 1.8.1, seperti ditunjukkan pada Gambar 3.



(a) hasil pengunduhan *file* Arduino *IDE* versi 1.8.1



(b) pemasangan Arduino *IDE* versi 1.8.1

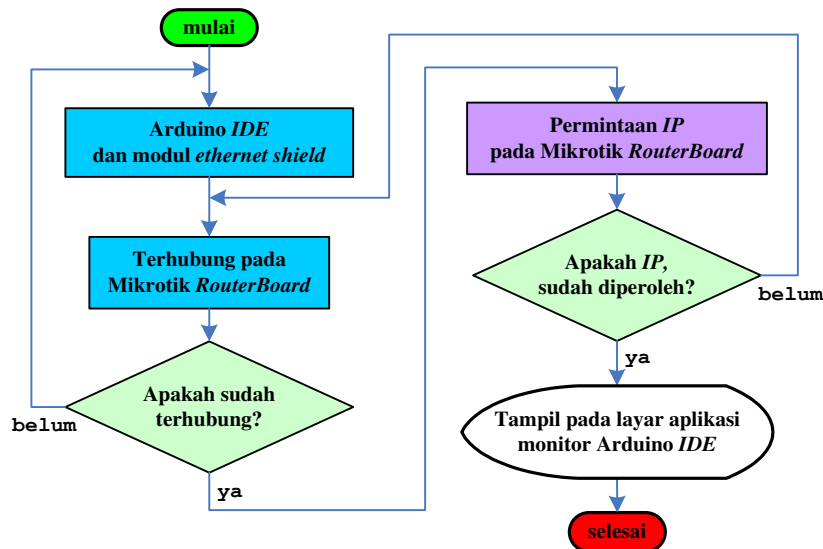
Gambar 3 Tampilan hasil pengunduhan *file* dan pemasangan Arduino *IDE* versi 1.8.1

a) Pemrograman terhadap mikrokontroler Arduino UNO R3

Tahapan setelah *software* Arduino terpasang pada *PC*, maka langkah-langkah lanjutan berupa penentuan algoritma dan penulisan sintaks yang dilanjutkan dengan proses *compiling* dan *uploading* terhadap *source code* dari komputer personal (*personal computer*, *PC*) ke dalam mikrokontroler Arduino UNO R3. Pembuatan sintaks untuk pemantauan sekaligus pengendalian analogi instalasi listrik fase-3, berupa tahapan pengunduhan sintaks pada Arduino untuk pengaktifan modul *ethernet shield*.

i) Algoritma dan sintaks

Penentuan algoritma dapat berbentuk diagram alir yang dilanjutkan dengan penulisan sintaks berbasis bahasa *C*. Penulisan sintaks merupakan upaya untuk perolehan sejumlah *source code* sebagai inti pengoperasian mikrokontroler Arduino UNO R3. Perolehan sebuah *minimum system* dengan *source code* berbasis bahasa *C*, merupakan salah satu bentuk minimalis sebuah sistem tertanam (*embedded system*). Diagram alir untuk pengaktifan *ethernet shield* dan pembuatan sintaks, seperti ditunjukkan pada Gambar 4.



Gambar 4 Diagram alir untuk pengaktifan ethernet shield

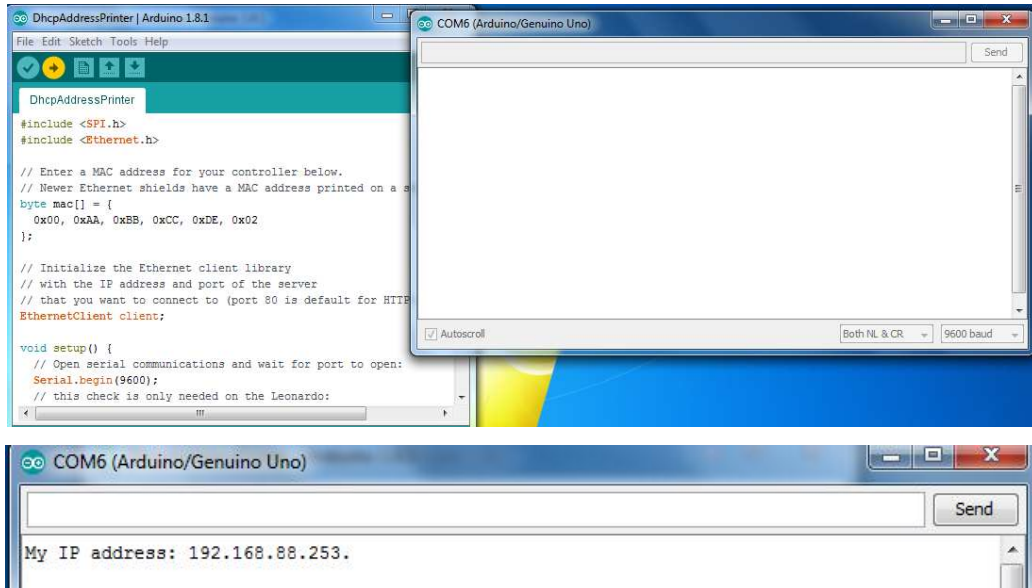
Berdasarkan Gambar 4 ditunjukkan, bahwa pembuatan diagram alir berkenaan dengan permintaan pengalamatan IP secara *Dynamic Host Control Protocol (DHCP)* pada Mikrotik RouterBoard, sekaligus sebagai bentuk penetapan ethernet shield pada Mikrotik RouterBoard. Untuk kondisi dimana modul ethernet shield beroperasi dengan baik, maka IP tertampilkan pada monitor Arduino IDE, sedangkan jika kondisi modul ethernet shield terdapat masalah, maka tertampilkan error pada monitor Arduino IDE.

Susunan sintaks untuk pengaktifan ethernet shield, yaitu:

```

#include <SPI.h>
#include <Ethernet.h>
byte mac[] = {
  0x00, 0xAA, 0xBB, 0xCC, 0xDE, 0x02 };
EthernetClient client;
void setup() {
  Serial.begin(9600);
  while (!Serial) { ; }
  if (Ethernet.begin(mac) == 0) {
    Serial.println("Failed to configure Ethernet using DHCP");}
  printIPAddress(); }
void loop() {
  switch (Ethernet.maintain()){
  Serial.println("Error: renewed fail");
    break;
    Serial.println("Renewed success");
    printIPAddress();
    break;
  Serial.println("Error: rebind fail");
    break;
    Serial.println("Rebind success");
    printIPAddress();
    break;
  default:
    break; }}
void printIPAddress() {
  Serial.print("My IP address: ");
  for (byte thisByte = 0; thisByte < 4; thisByte++) {
    Serial.print(Ethernet.localIP()[thisByte], DEC);
    Serial.print("."); }
  Serial.println();
}
  
```

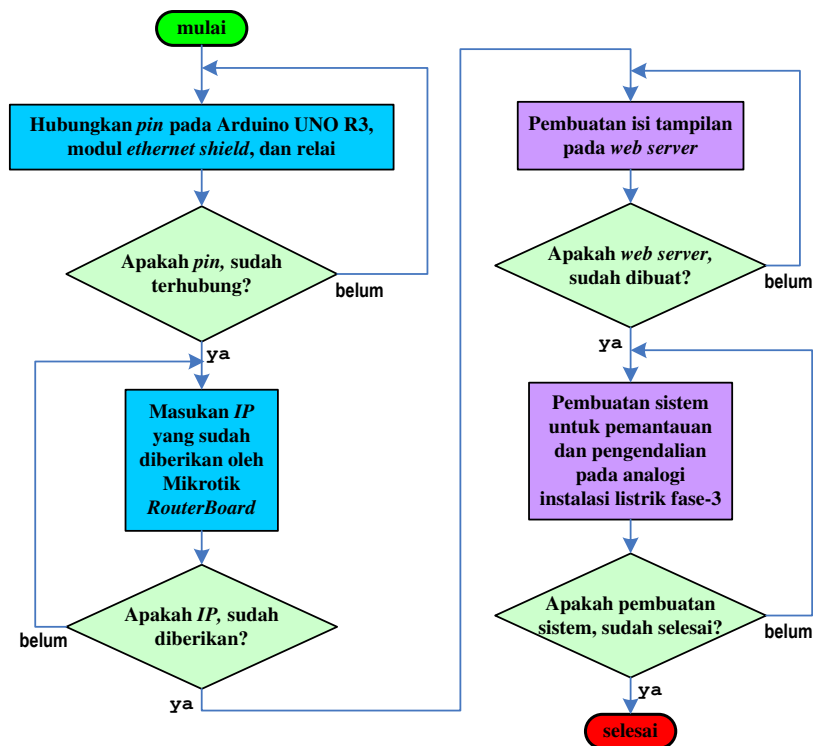
Setelah penyelesaian susunan sintaks, diperlukan *compiling* pada Arduino IDE yang telah berisi *script* pengaktifan. Tampilan hasil pengaktifan *ethernet shield* pada Arduino IDE, seperti ditunjukkan pada Gambar 5.



Gambar 5 Tampilan hasil pengaktifan *ethernet shield* pada Arduino IDE

Berdasarkan Gambar 5 ditunjukkan, bahwa setelah pengaktifan modul *ethernet shield* pada Arduino IDE, dapat dilakukan proses lanjutan berupa pembuatan *source code* tersebut, agar Arduino UNO R3 tersebut dapat berfungsi untuk pemantauan dan pengendalian kondisi terhadap analogi instalasi listrik fase-tiga.

Tampilan penentuan algoritma untuk pemantauan dan pengendalian terhadap analogi instalasi listrik fase-3, seperti ditunjukkan pada Gambar 6.



Gambar 6 Tampilan penentuan algoritma untuk pemantauan dan pengendalian terhadap analogi instalasi listrik fase-tiga

Berdasarkan Gambar 6 ditunjukkan, bahwa pengaktifan beberapa *pin*, pengaktifan *IP* yang sudah diberikan oleh Mikrotik *RouterBoard*, dan pengaktifan tampilan *web server* melalui *Arduino IDE* yang digunakan pada sistem pemantauan dan pengendalian terhadap analogi instalasi listrik fase-tiga. Susunan sintaks untuk pengaktifan sistem pemantauan dan pengendalian, yaitu:

```
#include <SPI.h>
#include <Ethernet.h>

int lampu1 = 6; // pin yang digunakan lampu
int lampu2 = 4; // pin yang digunakan lampu
int lampu3 = 8; // pin yang digunakan lampu

int Analog1 = A0;
int Analog2 = A1;
int Analog3 = A2;

boolean statusR1, statusR2, statusR3 = false; // informasi pada tabel status

String readString;
char c;
byte mac[] = {0xDA, 0xAD, 0xBE, 0xEB, 0xFE, 0xFB};
IPAddress ip(192, 168, 1, 252); // ip yang diberikan pada router
EthernetServer server(80);

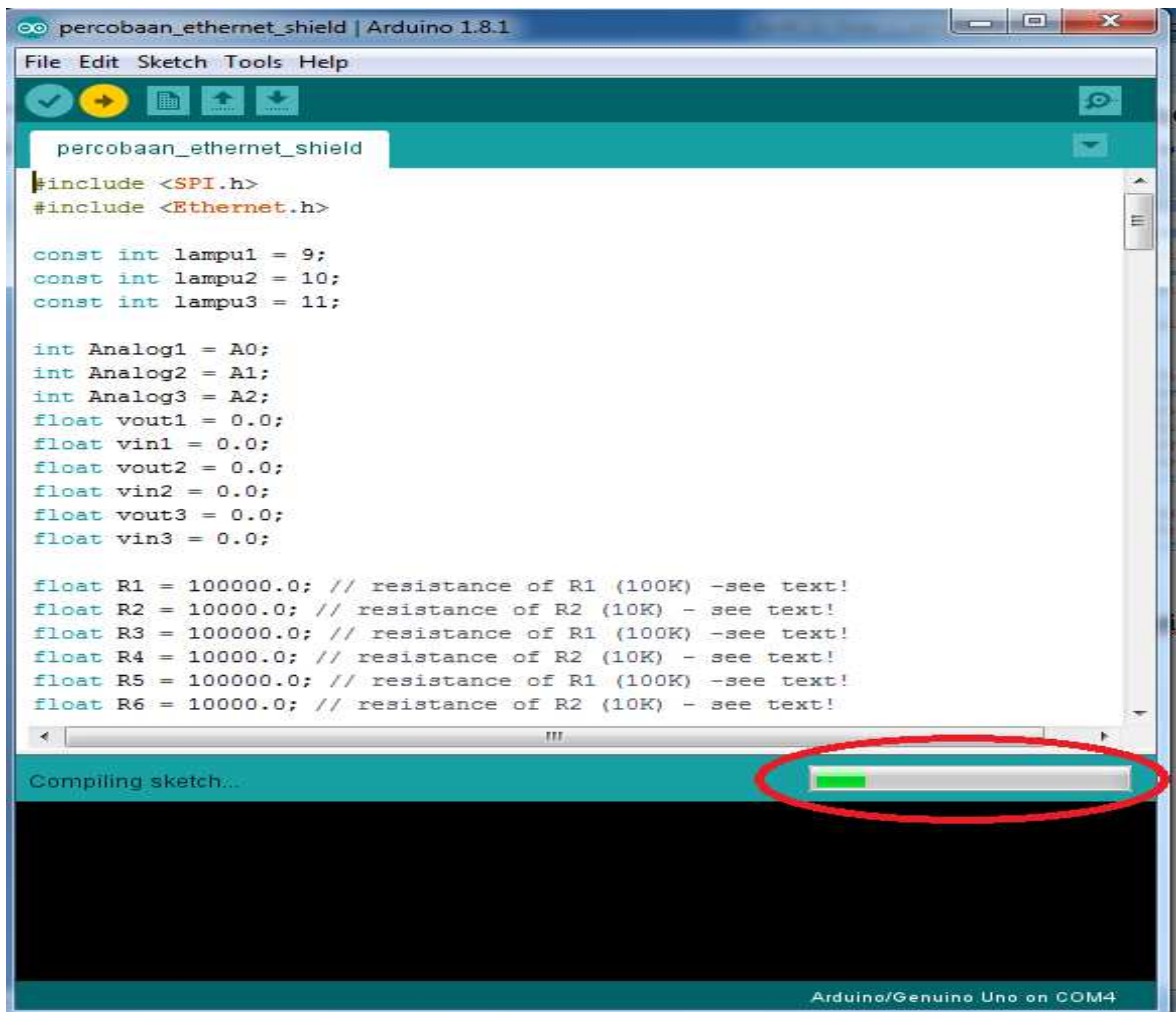
void setup() {
  pinMode(Analog1, INPUT); // mngaktifkan pin yang digunakan
  pinMode(Analog2, INPUT); // mngaktifkan pin yang digunakan
  pinMode(Analog3, INPUT); // mngaktifkan pin yang digunakan
  pinMode(lampu1, OUTPUT); // mngaktifkan pin yang digunakan
  boolean statusR1, statusR2, statusR3 = false; // informasi pada tabel status
}

void loop() {
  client.println("<html>");
  client.println("<head>");
  client.println("<title>Febby Hendrian</title>");
  client.println("<meta http-equiv=\"refresh\" content=\"5\">");
  client.println("</head>");
  client.println("<body align=\"center\" style=\"background-color: \"blue\">");
  client.println("<h1>PEMANTAUAN INSTALASI LISTRIK</h1>");
  client.println("");
  client.println("<hr />");
  client.println("<br />");
  client.println("<br />");
  client.println("<table border=\"1\" cellpadding=\"12\" align=\"center\">");
  client.println("<tr><td><center>FASE</center></td></tr>"); // tabel untuk fase
  client.println("<td><center>STATUS</center></td></tr>"); // tabel untuk status
  client.println("<td><center>SAKLAR</center></td></tr>"); // tabel untuk saklar
  client.println("<td><center>VOLT</center></td></tr>"); // tabel untuk volt
  client.println("<tr class=\"trh1\"><td align=\"center\">");
  client.println(" ");
  client.println(" ");
  client.println("1 (R)"); // tabel pada fase
  client.println("</td><td align=\"center\">");
  if (statusR1 == false) { //STATUS AWAL
    client.println("<b>PADAM</b>"); //JIKA KEADAAN STATUS PADAM
  } else {
    client.println("<b>MENYALA</b>"); // JIKA KEADAAN STATUS NYALA
  }
  client.println("</td><td align=\"center\">");
  if (statusR1 == false) {
    client.println("<a href=\"/1on\" />ON</a>"); // button on
  } else {
    client.println("<a href=\"/1off\" />OFF</a>"); //button off
  }
}
```

Berdasarkan susunan sintaks tersebut yang disesuaikan terhadap proses algoritma pada sebuah sistem tertanam (*embedded system*) tertunjukkan, bahwa terdapat penentuan konfigurasi *pin*, deklarasi variabel, inisialisasi, program utama (dengan kemunculan tampilan), ambil dan kirim data, dan keluaran.

ii) *Compiling dan uploading terhadap source code ke dalam mikrokontroler Arduino UNO R3*

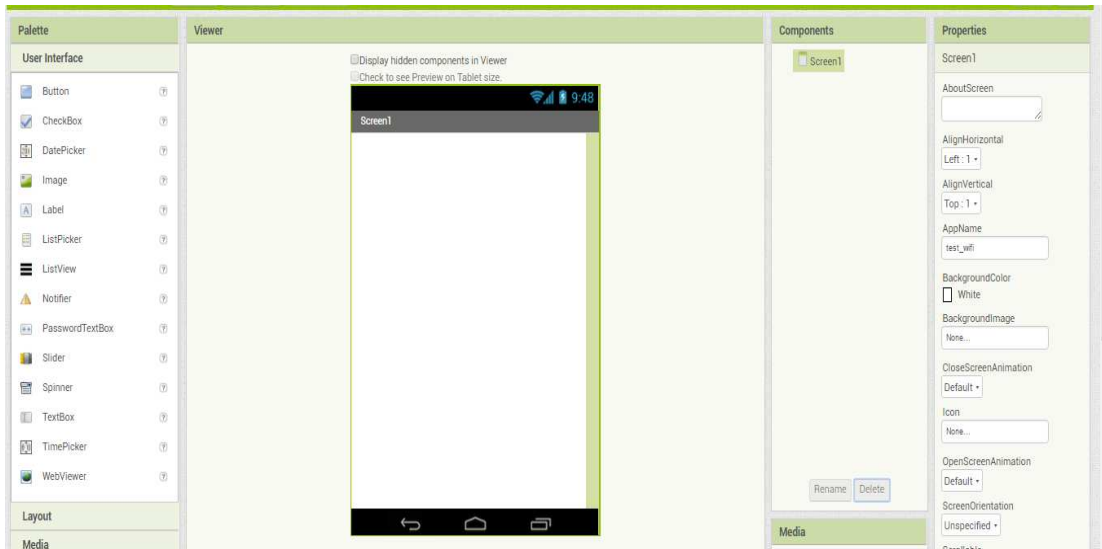
Tahapan *compiling* dan *uploading* terhadap *source code* ke dalam mikrokontroler Arduino UNO R3 merupakan tahapan setelah penentuan algoritma dan penulisan sintaks untuk perolehan sejumlah *source code*. Proses *compiling* dan *uploading* terhadap *source code* ke dalam mikrokontroler Arduino UNO R3 dari PC dilakukan dengan bantuan kabel USB. Tampilan proses *compiling* dan *uploading* terhadap *source code* dari PC ke mikrokontroler Arduino UNO R3, seperti ditunjukkan pada Gambar 7.



Gambar 7 Tampilan proses *compiling* dan *uploading* terhadap *source code* dari PC ke mikrokontroler Arduino UNO R3

3.1.2 Aplikasi monitoring berbasis *app inventor* pada komputer personal

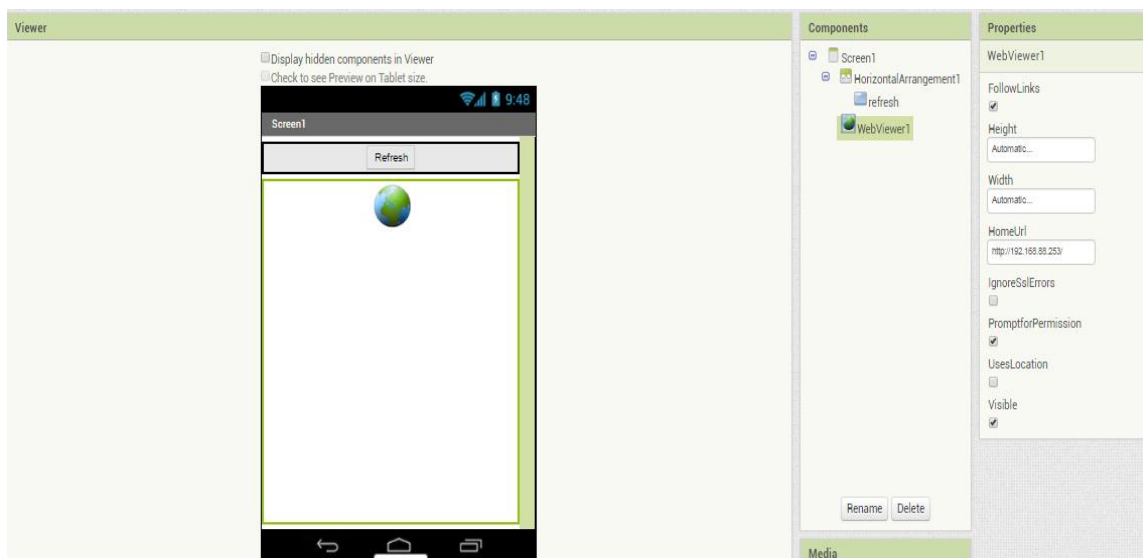
Aplikasi monitoring berbasis *app inventor* digunakan pada komputer personal untuk pembuatan sistem pemantauan dan pengendalian terhadap analogi instalasi listrik fase-tiga. Pembuatan aplikasi monitoring dilakukan secara *online* melalui *web browser* (<http://ai2.appinventor.mit.edu>). Tampilan aplikasi berbasis *app inventor*, seperti ditunjukkan pada Gambar 8.



Gambar 8 Tampilan aplikasi berbasis *app inventor*

Berdasarkan Gambar 8 ditunjukkan, bahwa setelah pembuatan aplikasi pada *Arduino IDE*, dilakukan pembuatan aplikasi untuk pemantauan dan pengendalian terhadap analogi instalasi listrik fase-3 yang dipasang pada *smartphone* berbasis Android berbantuan *app inventor*, berupa aplikasi dengan ekstensi (.apk) yang sebelumnya telah di “build” melalui *website app inventor* yang secara otomatis *file* tersebut tersimpan pada *PC* dan sudah siap untuk dipasang pada *smartphone* berbasis Android.

Pembuatan aplikasi monitoring melalui *app inventor* dilakukan dengan layar *design* dan *blocks* pada komputer personal. Tampilan pembuatan aplikasi monitoring melalui *app inventor* dengan layar *design* pada komputer personal, seperti ditunjukkan pada Gambar 9.



Gambar 9 Tampilan pembuatan aplikasi monitoring melalui *app inventor* dengan layar *design* pada komputer personal

Berdasarkan Gambar 9 ditunjukkan, bahwa tampilan rancangan pembuatan aplikasi monitoring pada layar *design*, sedangkan menu-menu pada layar *blocks* digunakan untuk pemberian perintah-perintah dalam bentuk *puzzle* yang sudah disediakan sebagai keperluan aplikasi yang dibuat pada layar *block*. Tampilan pembuatan aplikasi monitoring melalui *app inventor* dengan layar *blocks* pada komputer personal, seperti ditunjukkan pada Gambar 10.



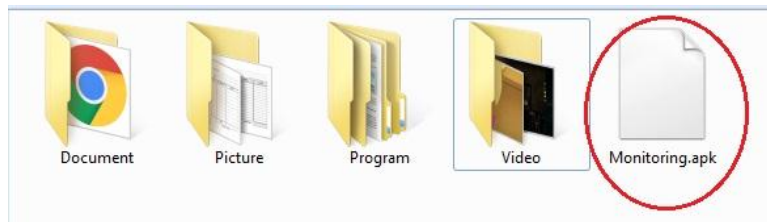
Gambar 10 Tampilan pembuatan aplikasi monitoring melalui *app inventor* dengan layar *blocks* pada komputer personal

Berdasarkan Gambar 10 ditunjukkan, bahwa tampilan pembuatan aplikasi monitoring dengan layar *blocks* pada komputer personal dapat dilakukan untuk pemakaian melalui langkah-langkah:

- a) ketika aplikasi monitoring di”buka”, maka tertampil hasil dari *web server*;
- b) pada layar *smartphone* ditampilkan hasil dari *web server* berupa tabel, dimana tabel tersebut berisikan tabel fase yang digunakan sebagai pemantauan, tabel status digunakan sebagai pemantauan terhadap fase, tabel saklar digunakan sebagai *button* untuk pengendalian terhadap lampu, dan “volt” berupa informasi nilai tegangan yang masuk pada saat saklar diaktifkan; dan
- c) *refresh button* digunakan untuk *reloading* isi dari *web server*.

3.1.3 Aplikasi monitoring pada *smartphone* berbasis Android

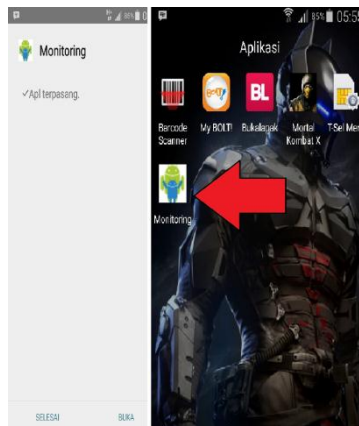
Setelah pembuatan aplikasi monitoring pada *web browser* berbasis *app inventor*, maka dilakukan pengunduhan terlebih dahulu untuk pemasangan aplikasi monitoring dengan melakukan transfer data melalui komputer personal. Tampilan pengunduhan aplikasi monitoring pada komputer personal, seperti ditunjukkan pada Gambar 11.



Gambar 11 Tampilan pengunduhan aplikasi monitoring pada komputer personal

Berdasarkan Gambar 11 ditunjukkan, bahwa setelah dilakukan pengunduhan aplikasi berbasis *app inventor*, dilakukan langkah selanjutnya berupa pemasangan aplikasi monitoring pada *smartphone* berbasis Android.

Tampilan proses pemasangan aplikasi monitoring pada *smartphone* berbasis Android, seperti ditunjukkan pada Gambar 12.



Gambar 12 Tampilan proses pemasangan aplikasi monitoring pada *smartphone* berbasis Android

Berdasarkan Gambar 12 ditunjukkan, bahwa proses pemasangan aplikasi monitoring telah berhasil terpasang. Perolehan hasil selanjutnya berupa pembukaan pada *smartphone* berbasis Android brmerek Samsung Note 3 SM-N900.

Tampilan hasil pemasangan aplikasi monitoring pada *smartphone* berbasis Android, seperti ditunjukkan pada Gambar 13.



FASE	STATUS	SAKLAR	VOLT
1 (R)	PADAM	<u>ON</u>	0.00
2 (S)	PADAM	<u>ON</u>	0.00
3 (T)	PADAM	<u>ON</u>	0.00

Gambar 13 Tampilan hasil pemasangan aplikasi monitoring pada *smartphone* berbasis Android

Berdasarkan Gambar 13 ditunjukkan, bahwa tampilan hasil pemasangan aplikasi monitoring pada *smartphone* berbasis Android berupa 4 (empat) kolom dengan pencantuman (i) fase, (ii) status, (iii) saklar, dan (iv) “volt”.

3.2. Pengukuran Kinerja *Minimum System* pada Pemantauan dan Pengendalian terhadap Analogi Instalasi Listrik Fase-tiga

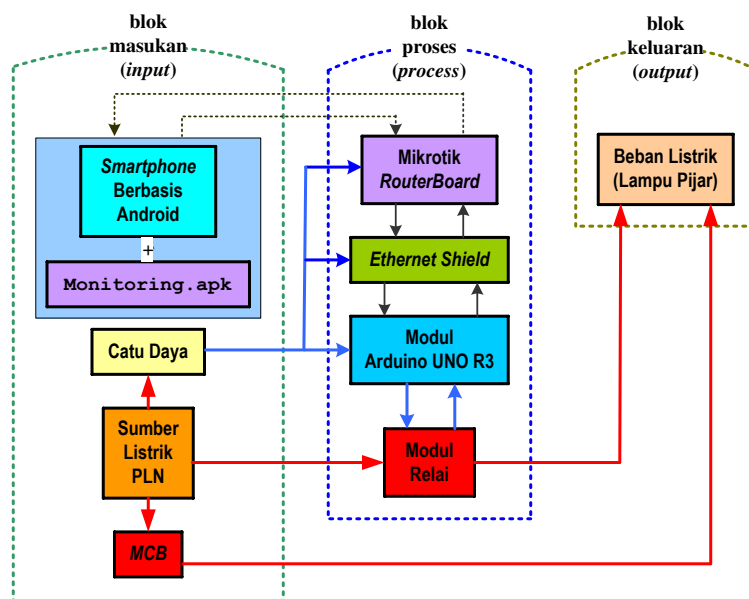
Pengukuran kinerja terhadap *minimum system* untuk pemantauan dan pengendalian analogi instalasi listrik fase-tiga dilakukan untuk perolehan hasil, setelah dilakukan pemasangan dan pembuatan aplikasi pada Arduino *IDE*, dan *smartphone* berbasis Android, selanjutnya diperoleh tampilan kebutuhan sistem untuk pelaksanaan pengamatan terhadap pemberian kondisi terhadap analogi instalasi listrik fase-R atau fase-S atau fase-T.

3.2.1 Kebutuhan sistem *hardware* dan *software* untuk pengamatan terhadap pemberian kondisi

Terdapat 2 (dua) kebutuhan sistem untuk pengamatan pada langkah pemantauan dan pengendalian terhadap analogi instalasi listrik pada fase-tiga, yaitu tampilan kebutuhan sistem untuk *hardware* dan *software*.

a) Kebutuhan sistem *hardware*

Diagram blok tampilan kebutuhan sistem *hardware*, seperti ditunjukkan pada Gambar 14.



Gambar 14 Diagram blok tampilan kebutuhan sistem hardware

Berdasarkan Gambar 14 ditunjukkan, bahwa *bread board* digunakan untuk tampilan sistem *hardware* berbantuan kabel *jumper* secukupnya dengan tujuan, untuk kemudahan proses pembuatan dan simulasi untuk penghubungan komponen satu dengan yang lain melalui penancapan kabel *jumper* pada jalur yang sudah disediakan, sehingga diperoleh kemudahan dalam proses pelepasan saat terjadi kesalahan teknis atau *trouble*. *Smartphone* berbasis Android yang digunakan untuk penelitian ini, yaitu Samsung Note 3 SM-N900, RAM 3 GB, Android versi 5.0 *Lollipop*.

Keberadaan diagram blok sistem *hardware* berupa tiga blok, yaitu (1) blok masukan, (2) blok proses, dan (3) blok keluaran.

i) Blok masukan

Blok ini terdapat 5 masukan, yaitu (1) *smartphone* berbasis Android, (2) aplikasi *monitoring.exe* digunakan untuk pemantauan dan pengendalian pada analogi instalasi listrik fase-3, (3) *power supply* digunakan untuk catu daya pada Arduino UNO R3, (4) sumber listrik PLN digunakan untuk catu daya untuk *minimum system* dan analogi instalasi listrik fas-3, (5) *MCB* digunakan untuk pengaman dan pemutus/penghubung lampu dari sumber listrik PLN.

ii) Blok proses

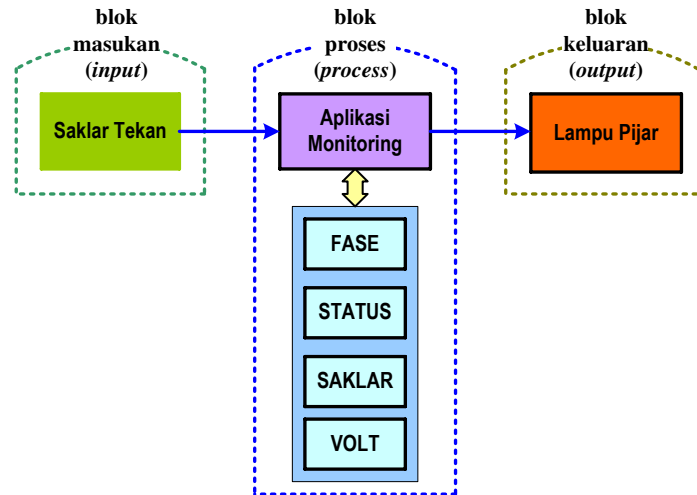
Blok proses berupa terdapat 4 proses, yaitu (1) Mikrotik *RouterBoard* digunakan, agar *smartphone* berbasis Android dapat terhubung dengan *web server*, (2) *ethernet shield* digunakan, agar Arduino dapat terhubung ke Mikrotik *RouterBoard*, (3) Arduino UNO R3, perangkat utama pada sistem yang digunakan untuk *compiling* dan *uploading*, agar sistem beroperasi secara sempurna, dan (4) modul relai adalah perangkat pendukung, agar Arduino dapat kendalikan lampu pijar.

iii) Blok keluaran

Blok keluaran berupa satu komponen, yaitu lampu pijar yang berfungsi sebagai tanda, apakah sistem telah beroperasi dengan baik atau terjadi *trouble*.

b) Kebutuhan sistem software

Diagram blok tampilan kebutuhan sistem *software*, seperti ditunjukkan pada Gambar 15.



Gambar 15 Diagram blok tampilan kebutuhan sistem software

Berdasarkan Gambar 15 ditunjukkan, bahwa *software* yang digunakan pada *smartphone* berbasis Android dirancang melalui sebuah aplikasi berbasis *web*, yaitu <http://ai2.appinventor.mit.edu>. Aplikasi berbasis *web* tersebut dikhususkan untuk pengembangan dan pembuatan aplikasi berbasis Android. Dalam aplikasi tersebut digunakan sebuah *laptop* dengan minimal spesifikasi pada *processor* Core2Duos P7370, CPU 2.00GHz dan RAM 2GB dengan sistem pengoperasian Windows 7 Ultimate 32-Bit. Untuk akses aplikasi berbasis *web* <http://ai2.appinventor.mit.edu> digunakan sebuah *browser* Google Chrome dengan versi 58.0.3029.110, sedangkan untuk keterhubungan dengan *Internet* digunakan Mikrotik *RouterBoard*. Terdapat 3 blok diagram pada sistem *software*, yaitu: i) blok *input*, ii) blok *process*, dan iii) blok *output*.

i) Blok input

Blok ini terdapat 1 (satu) masukan, yaitu *button* digunakan untuk *me-refresh* atau *me-reolad* ulang pada saat *webserver* dalam keadaan *error*.

ii) Blok process

Blok *process* sistem *software* ini berupa tampilan hasil *web server* pada aplikasi *monitoring.exe*, dimana terdapat beberapa tabel, yaitu (1) tabel fase berfungsi sebagai suatu informasi, (2) status digunakan sebagai informasi sistem pemantauan pada analogi instalasi listrik fase-3, (3) saklar digunakan untuk pengendalian jalur fase pada lampu, dan (4) volt digunakan untuk sebagai informasi nilai tegangan yang dihasilkan pada saat jalur fase dengan lampu dalam keadaan menyala.

iii) Blok output

Blok ini terdapat 1 keluaran, yaitu lampu yang berfungsi sebagai tanda apakah aplikasi yang dibuat beroperasi dengan baik atau terjadi *trouble*.

3.2.2 Pengamatan dengan pemberian kondisi

Setelah perolehan tampilan kebutuhan sistem, dilakukan pengamatan dengan pemberian kondisi untuk pemantauan dan pengendalian terhadap analogi instalasi listrik fase-3, yaitu untuk fase-R atau fase-S atau fase-T.

a) Pemantauan dan pengendalian terhadap analogi instalasi listrik pada fase-R

Tampilan hasil pemantauan dan pengendalian terhadap analogi instalasi listrik pada fase-R, seperti ditunjukkan pada Tabel 1.

Tabel 1 Tampilan hasil pemantauan dan pengendalian terhadap analogi instalasi listrik pada fase-R

Pengamatan	Android	Analogi Instalasi Listrik Fase-3																
<p>Hasil pada Fase-R (kesatu) saat saklar ditekan melalui tombol pada <i>smartphone</i></p>	 <table border="1"> <thead> <tr> <th>FASE</th> <th>STATUS</th> <th>SAKLAR</th> <th>VOLT</th> </tr> </thead> <tbody> <tr> <td>1 (R)</td> <td>MENYALA</td> <td>OFF</td> <td>213.08</td> </tr> <tr> <td>2 (S)</td> <td>PADAM</td> <td>ON</td> <td>0.00</td> </tr> <tr> <td>3 (T)</td> <td>PADAM</td> <td>ON</td> <td>0.00</td> </tr> </tbody> </table>	FASE	STATUS	SAKLAR	VOLT	1 (R)	MENYALA	OFF	213.08	2 (S)	PADAM	ON	0.00	3 (T)	PADAM	ON	0.00	
FASE	STATUS	SAKLAR	VOLT															
1 (R)	MENYALA	OFF	213.08															
2 (S)	PADAM	ON	0.00															
3 (T)	PADAM	ON	0.00															

Berdasarkan Tabel 1 ditunjukkan, bahwa lampu pijar pada fase-R menyala dan lampu pijar pada fase-S dan fase-T tidak menyala, pada fase-R terdapat (1) tabel status berfungsi sebagai informasi pemantauan analogi instalasi listrik terhadap fase-R, untuk kondisi dimana saklar ditekan ON, maka keadaan status berubah dari padam menjadi menyala dan berlaku untuk sebaliknya; (2) tabel saklar berfungsi sebagai pengendalian analogi instalasi listrik terhadap fase-R (pertama); dan (3) volt berfungsi sebagai informasi nilai tegangan pemantauan analogi instalasi listrik terhadap fase-R, untuk kondisi dimana saklar ditekan ON, maka pada kolom volt berisikan nilai tegangan dan berlaku untuk sebaliknya.

b) Pemantauan dan pengendalian terhadap analogi instalasi listrik pada fase-S

Tampilan hasil pemantauan dan pengendalian terhadap analogi instalasi listrik pada fase-S, seperti ditunjukkan pada Tabel 2.

Tabel 2 Tampilan hasil pemantauan dan pengendalian terhadap analogi instalasi listrik pada fase-S



Pengamatan	Android	Analogi Instalasi Listrik Fase-3																
<p>Hasil pada Fase-S (kedua) saat saklar ditekan melalui tombol pada <i>smartphone</i></p>	 <table border="1"> <thead> <tr> <th>FASE</th> <th>STATUS</th> <th>SAKLAR</th> <th>VOLT</th> </tr> </thead> <tbody> <tr> <td>1 (R)</td> <td>MENYALA</td> <td>OFF</td> <td>212.35</td> </tr> <tr> <td>2 (S)</td> <td>MENYALA</td> <td>OFF</td> <td>210.15</td> </tr> <tr> <td>3 (T)</td> <td>PADAM</td> <td>ON</td> <td>00.0</td> </tr> </tbody> </table>	FASE	STATUS	SAKLAR	VOLT	1 (R)	MENYALA	OFF	212.35	2 (S)	MENYALA	OFF	210.15	3 (T)	PADAM	ON	00.0	
FASE	STATUS	SAKLAR	VOLT															
1 (R)	MENYALA	OFF	212.35															
2 (S)	MENYALA	OFF	210.15															
3 (T)	PADAM	ON	00.0															

Berdasarkan Tabel 2 ditunjukkan, bahwa lampu pijar pada fase-R masih menyala dan pada fase-T tidak menyala, pada fase-S terdapat (1) tabel status berfungsi sebagai informasi pemantauan analogi instalasi listrik terhadap fase-S), untuk kondisi dimana saklar ditekan ON, maka keadaan status berubah dari padam menjadi menyala dan berlaku untuk sebaliknya; (2) tabel saklar berfungsi sebagai pengendalian analogi instalasi listrik terhadap fase-S; dan (3) “volt” berfungsi sebagai informasi nilai tegangan pemantauan analogi instalasi listrik terhadap fase-S (kedua), untuk kondisi dimana saklar ditekan ON, maka pada kolom “volt” berisikan nilai tegangan dan berlaku untuk sebaliknya.

c) Pemantauan dan pengendalian terhadap analogi instalasi listrik pada fase-T

Tampilan hasil pemantauan dan pengendalian terhadap analogi instalasi listrik pada fase-T, seperti ditunjukkan pada Tabel 3.

Tabel 3 Tampilan hasil pemantauan dan pengendalian terhadap analogi instalasi listrik pada fase-T

Pengamatan	Android	Analogi Instalasi Listrik Fase-3																
<p>Hasil pada Fase-T (ketiga) saat saklar ditekan melalui tombol pada <i>smartphone</i></p>	 <table border="1" style="margin: auto; border-collapse: collapse;"> <thead> <tr> <th>FASE</th> <th>STATUS</th> <th>SAKLAR</th> <th>VOLT</th> </tr> </thead> <tbody> <tr> <td>1 (R)</td> <td>MENYALA</td> <td>OFF</td> <td>212.35</td> </tr> <tr> <td>2 (S)</td> <td>MENYALA</td> <td>OFF</td> <td>210.15</td> </tr> <tr> <td>3 (T)</td> <td>MENYALA</td> <td>OFF</td> <td>207.25</td> </tr> </tbody> </table>	FASE	STATUS	SAKLAR	VOLT	1 (R)	MENYALA	OFF	212.35	2 (S)	MENYALA	OFF	210.15	3 (T)	MENYALA	OFF	207.25	
FASE	STATUS	SAKLAR	VOLT															
1 (R)	MENYALA	OFF	212.35															
2 (S)	MENYALA	OFF	210.15															
3 (T)	MENYALA	OFF	207.25															

Berdasarkan Tabel 3 ditunjukkan, bahwa lampu pijar pada fase-R dan fase-S masih menyala, pada fase-T terdapat (1) tabel status berfungsi sebagai informasi pemantauan analogi instalasi listrik terhadap fase-T. Untuk kondisi dimana saklar ditekan ON, maka keadaan status berubah dari padam menjadi menyala dan berlaku untuk sebaliknya, (2) tabel saklar berfungsi sebagai pengendalian analogi instalasi listrik terhadap fase-T, dan (3) volt berfungsi sebagai informasi nilai tegangan pemantauan analogi instalasi listrik terhadap fase-T (ketiga). Untuk kondisi dimana saklar ditekan ON, maka pada kolom volt berisikan nilai tegangan dan berlaku untuk sebaliknya.

4. KESIMPULAN

Berdasarkan bahasan, maka ditarik kesimpulan sesuai tujuan penelitian.

- 1) Perolehan bentuk fisis *minimum system* untuk pemantauan dan pengendalian terhadap analogi instalasi listrik fase-3, dilakukan melalui *handshaking* secara *hardware* dan *software*. Integrasi sistem secara *hardware*, berupa integrasi sejumlah peranti elektronika, yaitu: i) mikrokontroler Arduino UNO R3, ii) Arduino *ethernet shield* W5100, iii) Mikrotik *RouterBrand*, iv) modul relai, dan v) *smartphone* berbasis Android dan analogi instalasi listrik fase-3. Integrasi sistem secara *software* berupa 4 (empat) langkah, yaitu: i) pemasangan Arduino *IDE* versi 1.8.1 berbasis bahasa pemrograman C untuk *comfiling* dan *uploading* ke peranti mikrokontroler Arduino UNO R3, ii) pembuatan aplikasi untuk *smartphone* berbasis Android, iii) pengunduhan *software* untuk pemasangan aplikasi pada *smartphone* berbasis Android, dan iv) pemasangan *file* dengan format apk pada *smartphone* berbasis Android. Mikrokontroler Arduino UNO R3 dan *smartphone* berbasis Android sebagai pusat kendali *minimum system* terpasang.
- 2) Keberadaan tampilan untuk pelaksanaan pengamatan terhadap pengukuran kinerja *minimum system* didasarkan kepada 2 (dua) kebutuhan sistem, sistem *hardware* dan *software*. Pengukuran kinerja dilakukan dengan pemantauan dan pengendalian terhadap analogi instalasi listrik fase-3, melalui 3 (tiga) pemberian kondisi berbeda pada jalur fase-R atau fase-S atau fase-T. Pemantauan dan pengendalian dengan pemberian kondisi telah sesuai dengan keberadaan *minimum system*.

DAFTAR PUSTAKA

Arduino. (2012, October). *Arduino Ethernet*. [Online] Available: <https://www.arduino.cc/en/Main/arduinoBoardEthernet> (accessed: January, 30 2017).
 ATMEL Corporation. (2009, May). *8-bit Microcontroller with 32 Kbytes In-System Programmable Flash*. [Online]. Available:

- <http://site.gravitech.us/Arduino/NANO30/ATMEGA328datasheet.pdf> (accessed: 30 January, 30 2014).
- Axelson, Jan. (2005, August). *USB Complete: Everything You Need to Develop Custom USB Peripherals*, Third Edition, Lakeview Research LLC, Madison, WI, pp. 1-32.
- Axelson, Jan. (2009). *USB Complete: The Developer's Guide*, Fourth Edition, Lakeview Research LLC, Madison, WI, pp. 10-20.
- a. Banzi, Massimo. (2011). *Getting Started with Arduino: the Open Source Electronics Prototyping Platform* (2nd Edition), O'Reilly Media. Sebastopol, CA, pp. 17-24.
- Banzi, Massimo and Michael Shiloh. (2015). *Getting Started with Arduino: the Open Source Electronics Prototyping Platform*, (3rd Edition), Maker Media, Sebastopol, pp. 15-22.
- Barrett, Steven F., *Arduino Microcontroller: Processing for Everyone!*, Second Edition, Morgan & Claypool, San Rafael, CA, 2012, pp. 47-50.
- di Cerbo, Manuel. (2011, September 16). *Creating a Serial to USB driver using the Android USB-Host API*. [Online]. Available: <http://android.serverbox.ch/?p=370> (accessed: January, 30 2016).
- Durfee, William. (2011, October). *Arduino Microcontroller Guide*. Course Material, University of Minnesota, Minneapolis, pp. 1-17.
- Evans, Brian. (2011). *Beginning Arduino Programming: Writing Code for the Most Popular Microcontroller Board in the World*, Apress, New York, NY, pp. 1-11.
- Forouzan, Behrouz A., (2010). *TCP/IP Protocol Suite* (Fourth Edition), McGraw-Hill, New York, pp. 7-8.
- Goeritno, Arief, Ritzkal, Ayumi Johan, (2016) "Kinerja Prototipe Sistem Elektronik Berbasis Mikrokontroler Arduino Uno R3 Untuk Pemantauan Analogi Instalasi Listrik," di *Jurnal Ilmiah SETRUM*, Volume 5, No.2, hlm. 94-99, Desember 2016.
- Hermawan, Cepi dkk., "Penggunaan Protokol *Internet* dan *Bluetooth* Untuk Sistem Penggerakan Kunci Pintu Berbantuan Arduino UNO R3 Terkendali melalui *Smartphone* Berbasis Android 4.4.2 KitKat," di *The 4th National Conference on Industrial Electrical and Electronics (NCIEE)* Jurusan Teknik Elektro Universitas Sultan Ageng Tirtayasa (Untirta), Cilegon, BT, 2016, hlm. 113-124.
- Johan, Ayumi, Arief Goeritno, Ritzkal, "Prototipe Sistem Elektronik Berbasis Mikrokontroler Untuk Pemantauan Instalasi Listrik," di *Prosiding Seminar Nasional Teknologi Industri (SNTI FTI-Usakti V-2016)*, Jakarta, JK, 2016, hlm. 324-330.
- Kernighan, Brian W. and Dennis M. Ritchie. (1988). *The C programming Language*, Second Edition, Prentice Hall, New Jersey, pp. 1-4.
- Maalej, Walid and M.P. Robillard, (2013). "Patterns of Knowledge in API Reference Documentation," in *IEEE Transactions on Software Engineering*, Vol. 39, No. 9, September 2013, pp. 1264-1282.
- Margolis, Michael. (2011, November). *Arduino Cookbook*. O'Reilly Media, Inc., Sebastopol, CA, pp. 1-15.
- Monperrus, Martin et al. (2012). "What Should Developers Be Aware Of? An Empirical Study on the Directives of API Documentation," in *Empirical Software Engineering*, Vol. 17, No. 6, December 2012, pp. 703-737.
- Nosrati, M., R. Karimi, H.A. Hasanvand, (2012). "Mobile Computing: Principles, Devices and Operating Systems," in *World Applied Programming*. Vol 2, No. 7, July 2012, pp. 399-408.
- Purdum, Jack. (2012). *Beginning C for Arduino: Learn C Programming for the Arduino and Compatible Microcontrollers*, (1st ed.), Apress, New York, NY, pp. 1-19.
- Purdy, Kevin (2011). *The Complete Android Guide*, 3Ones, Inc., San Diego, pp. 4-66.
- Ritchie, Dennis M. (1993). "The Development of the C Language," in *Presented at Second History of Programming Languages Conference*, pp. 1-16.