

EKSPLORASI INSTRUKSI BOOLEAN DALAM BAHASA C UNTUK MEMPROGRAM PLC ARDUINO

Zaiyan Ahyadi

Politeknik Negeri Banjarmasin

z.ahyadi@poliban.ac.id

ABSTRACT

PLC arduino is a low cost arduino system that is used to substitute high cost PLC system. The main problem to use arduino as PLC is programming language of arduino is high level language C, while PLC mostly is programmed by diagram ladder. This paper explores metode to use boolean sintaxis in C, thus ladder diagram of PLC system easily translated to arduino system. The translation method shows clear equivalence between ladder diagram and boolean sintaxis. The survey result shows that the method is easy to be understood for people who has lack skills in programming.

Keywords: *Arduino, PLC, ladder diagram, boolean*

ABSTRAK

PLC arduino adalah sistem arduino dengan harga murah yang digunakan untuk menggantikan sistem PLC yang berbiaya mahal. Masalah utama agar arduino dapat digunakan seperti PLC adalah bahasa pemrograman arduino adalah bahasa tingkat tinggi C, sedangkan PLC umumnya diprogram dengan diagram ladder. Makalah ini mengeksplorasi metode instruksi boolean dalam bahasa C, sehingga sistem PLC dengan bahasa pemrograman diagram ladder dengan mudah dapat ditranslasi ke sistem arduino. Metode translasi memperlihatkan kesamaan yang jelas antara ladder diagram dengan sintaks boolean. Hasil survey memperlihatkan metode ini dapat dengan mudah difahami bahkan oleh orang yang mempunyai dasar pemrograman lemah.

Kata Kunci: *Arduno, PLC, diagram ladder, boolean*

PENDAHULUAN

PLC merupakan peralatan kontrol otomatis yang umum dipakai dalam dunia industri. Harga PLC yang mahal menyebabkan PLC masih sedikit digunakan oleh industri kecil. Bahkan dalam dunia pendidikan pun, harga masih menjadi kendala utama bagi institusi pendidikan untuk dapat menyediakan sarana praktikum PLC yang memadai untuk mahasiswa.

Komponen utama PLC adalah mikrokontroler, yang disebut juga *computer in single chip*, yaitu mikroprosesor yang telah dilengkapi dengan interface input output dan memori utama dalam satu chip. Pemrograman PLC berdasarkan standar IEC 1131-3 (International Electrotechnical Commission) ada 5 cara: *Ladder Diagram (LD)*, *Instruction List (IL)*, *Function Block Diagram (FBD)*, *Sequential Function Chart (SFC)* dan *Structured Text (ST)*. Di antara kelima bahasa pemrograman tersebut yang paling populer adalah Ladder Diagram, setelah itu Instruction List. Pemrograman LD menjadi populer karena bentuknya serupa dengan diagram pengawatan listrik 2 kawat. Sehingga teknisi listrik dengan mudah memahaminya. Structured text merupakan bahasa yang paling jarang dipakai oleh programmer PLC karena bahasa ini adalah bahasa pemrograman komputer seperti bahasa C atau Pascal. Sebenarnya bahasa ini merupakan bahasa yang umum dipakai untuk memprogram mikroprosesor, namun kebanyakan teknisi PLC tidak terbiasa dengan bahasa pemrograman komputer.

Arduino adalah sistem mikrokontroler AVR yang umum dipakai dalam sistem embedded sekarang ini. Arduino diprogram dengan menggunakan IDE arduino dengan dasar bahasa pemrograman bahasa C++ (superset dari bahasa C). Namun bahasa C untuk arduino telah dimodifikasi sehingga dapat mengenali komponen arduino. Arduino dilengkapi dengan banyak *library* (pustaka program) sehingga orang tanpa pengetahuan mendalam tentang komponen elektronika dapat menggunakannya. Dengan banyaknya produksi *clone* arduino di pasaran menyebabkan harganya menjadi sangat murah. Harga terendah di pasaran untuk clone arduino nano cuma Rp 23.000,-. Bandingkan dengan harga PLC yang pada umumnya di atas satu juta.

Secara teori karena PLC pada dasarnya adalah mikrokontroler, seharusnya arduino yang menggunakan mikrokontroler AVR dapat digunakan dan diprogram seperti PLC. Sehingga masalah biaya otomasi yang mahal dapat diatasi.

Rumusan masalah

Arduino belum digunakan untuk menggantikan PLC karena kelemahan para teknisi PLC dalam hal pemrograman mikrokontroler. Pada umumnya PLC diprogram dengan diagram ladder yang tidak dapat digunakan untuk memprogram arduino.

Tujuan penelitian

Penelitian ini bertujuan untuk menemukan metode yang praktis dan mudah dipahami agar sistem arduino dapat bekerja seperti PLC dengan logika pemrograman berbasis diagram ladder. Metode terutama harus mudah dipahami oleh pengguna PLC meskipun belum pernah punya pengalaman menggunakan arduino atau mikrokontroler lainnya. Pada penelitian ini, sistem arduino yang digunakan sebagai pengganti PLC disebut dengan PLC Arduino.

Penelitian lain dalam bidang PLC berbasis arduino

Beberapa peneliti telah menggagas ide untuk menggunakan mikrokontroler generik seperti AVR dan MCS 51 untuk dapat menggantikan PLC. Kebanyakan dari mereka mengusulkan pemrograman mikrokontroler menggunakan fungsi-fungsi yang serupa dengan bahasa instruction list pada PLC dengan menggunakan pustaka PLClib (Granvillano, 2014). Dian Artanto juga mengembangkan pustaka PLC untuk arduino dengan metode yang sama. Translasi diagram ladder adalah memanggil fungsi yang ada pada pustaka seperti menggunakan bahasa Instruction List.

Terdapat juga program ladder berbasis grafis buatan Jonathan Westhues untuk mikrokontroler yang diberi nama LDmicro yang dapat didownload free pada situs laman web <http://cq.cx/ladder-tutorial.pl>. Namun hasil dari kompilasi program adalah susunan fungsi serupa dengan bahasa Instruction List.

METODE PENELITIAN

Penelitian ini dilakukan dengan tiga tahapan. Tahapan pertama adalah mempelajari salah satu PLC dengan teknik pemrogramannya yang berbasis diagram ladder. Di pasaran tersedia banyak jenis PLC. Untuk lebih praktisnya maka akan dipilih salah satu jenis PLC yang akan dijadikan obyek. Dalam hal ini penulis memilih Zelio logic dari Schneider. Meskipun oleh pabriknya zelio disebut sebagai smart relay, namun prinsip kerjanya hampir sama dengan PLC. Perbedaan antara smart relay dan PLC terletak pada pembaharuan logika output, dimana PLC pembaharuan output dilakukan pada siklus program yang sama sedangkan smart relay pembaharuan output dilakukan pada siklus program berikutnya (Putro, Agfianto Eko. 2007).

Tahap kedua membuat program ekivalensi (kesamaannya) dengan menggunakan bahasa C. Pada tahap ini akan dibuat *template* (kerangka) program untuk sistem yang telah ditentukan hardwarenya. Penulis menggunakan arduino nano dengan 7 buah input terhubung ke D0 sampai dengan D6, dan 7 buah output yang terhubung ke D7 sampai dengan D13, seperti yang ditunjukkan pada gambar 1. Selain template logika kontrol PLC akan dituliskan dalam fungsi dengan nama *process()*. Program yang dibuat berdasarkan diagram ladder sederhana untuk melihat sesuai tidaknya program berjalan dengan jalannya PLC. Dalam hal ini penulis masih menggunakan simulator zelio dan simulator arduino dibantu dengan program Proteus.

Tahap ketiga menerapkan hasil yang didapat pada tahap kedua untuk sistem PLC lain yang lebih kompleks. Pada tahap ini verifikasi sudah menggunakan hardware PLC dan arduino sebenarnya. Pada tahap ini juga dilakukan survey terhadap beberapa orang untuk mengetahui tingkat kemudahan metode pemrograman arduino yang diusulkan sehingga dapat dilihat tingkat kemudahannya.

Dasar Teori PLC

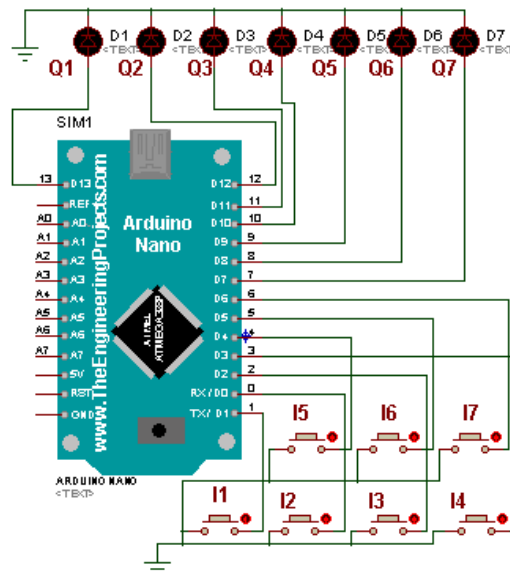
Port pada PLC telah ditetapkan terlebih dahulu input outputnya. Ini sebagai konsekuensi bahwa PLC adalah peralatan kontrol siap pakai. Port PLC telah dilengkapi komponen optocoupler sebagai peralatan pengaman. Untuk jenis PLC yang dapat langsung mengontrol beban dengan tegangan besar (di atas 24 V) biasanya Port outputnya telah dilengkapi dengan komponen relay.

Software PLC bekerja dalam 3 tahapan secara terus menerus:

- Baca seluruh input
Pada tahapan ini status logika input pada port akan dibaca dan disimpan dalam memori PLC
- Proses input
Pada proses ini kondisi logika input yang telah disimpan dalam memori diproses berdasarkan dengan logika yang telah diprogram pada PLC. Dimana biasanya logika tersebut diprogram dengan menggunakan bahasa diagram ladder
- Kirim output
Pada tahap ini nilai logika output hasil dari proses sebelumnya dikirimkan ke port output.

Sistem PLC Arduino

Sistem arduino yang digunakan sebagai PLC arduino dapat dilihat pada gambar 1.



Gambar 1. Sistem arduino yang digunakan sebagai PLC

Untuk sistem hardware arduino pada gambar 1 di atas, dapat dibuat template programnya sebagai berikut:

Kode1 . Template program

```
#define Q1p 13
#define Q2p 12
#define Q3p 11
#define Q4p 10
#define Q5p 9
#define Q6p 8
```

```
#define Q7p 7

#define I1p 0
#define I2p 1
#define I3p 2
#define I4p 3
#define I5p 4
#define I6p 5
#define I7p 6

bool Q1,Q2,Q3,Q4,Q5,Q6,Q7,I1,I2,I3,I4,I5,I6,I7;
bool Q1old, Q1lvl, Q1tog, Q1set, Q1rst;
bool Q2old, Q2lvl, Q2tog, Q2set, Q2rst;
bool Q3old, Q3lvl, Q3tog, Q3set, Q3rst;
bool Q4old, Q4lvl, Q4tog, Q4set, Q4rst;
bool Q5old, Q5lvl, Q5tog, Q5set, Q5rst;
bool Q6old, Q6lvl, Q6tog, Q6set, Q6rst;
bool Q7old, Q7lvl, Q7tog, Q7set, Q7rst;

void setup() {
  pinMode(I1p, INPUT_PULLUP); pinMode(I2p, INPUT_PULLUP);
  pinMode(I3p, INPUT_PULLUP); pinMode(I4p, INPUT_PULLUP);
  pinMode(I5p, INPUT_PULLUP); pinMode(I6p, INPUT_PULLUP);
  pinMode(I7p, INPUT_PULLUP);
  pinMode(Q1p, OUTPUT); pinMode(Q2p, OUTPUT);
  pinMode(Q3p, OUTPUT); pinMode(Q4p, OUTPUT);
  pinMode(Q5p, OUTPUT); pinMode(Q6p, OUTPUT);
  pinMode(Q7p, OUTPUT);
}

void loop() {
  // read input
  I1 = not digitalRead(I1p); I2 = not digitalRead(I2p);
  I3 = not digitalRead(I3p); I4 = not digitalRead(I4p);
  I5 = not digitalRead(I5p); I6 = not digitalRead(I6p);
  I7 = not digitalRead(I7p);
  //proses
  process () ;

  // write output
  digitalWrite(Q1p,Q1); digitalWrite(Q2p,Q2);
  digitalWrite(Q3p,Q3); digitalWrite(Q4p,Q4);
  digitalWrite(Q5p,Q4); digitalWrite(Q6p,Q4);
  digitalWrite(Q7p,Q4);}
```

Dapat dilihat pada kode 1 yang merupakan program template untuk sistem arduino pada gambar 1 terdiri dari 3 bagian. Bagian pertama adalah deklarasi seluruh variabel yang akan digunakan pada sistem. Pada bagian ini juga dituliskan perintah direktive *define* untuk membuat *alias* dari nama pin input / output arduino.

Bagian kedua adalah fungsi *setup()* yang dijalankan hanya sekali dan merupakan template dasar arduino. Fungsi ini melakukan inisialisasi terhadap sistem arduino, antara lain menetapkan pin sebagai input atau output. Port input diset menjadi mode *input pull-up*, karena pada rangkaiannya input push button jika ditekan akan terhubung dengan ground. Resistor pullup yang digunakan merupakan resistor internal pada chip AVR.

Bagian ketiga adalah fungsi *loop()* yang yang dikerjakan berulang kali terus menerus selama sistem diberikan catu daya. Fungsi loop menjalankan tiga tahapan cara kerja PLC: baca seluruh input, dilanjutkan dengan memproses logika input, kemudian mengirim hasil output ke port output.

Pada tahapan baca input, logika pada port input yang terhubung secara pull-up dibaca secara terbalik dengan menambahkan perintah *not*. Ini berguna agar pembacaan logika tidak terbalik dengan aksi penekanan tombol input. Hasil pembacaan status input disimpan pada variabel dengan nama yang bersesuaian dengan nama pin input.

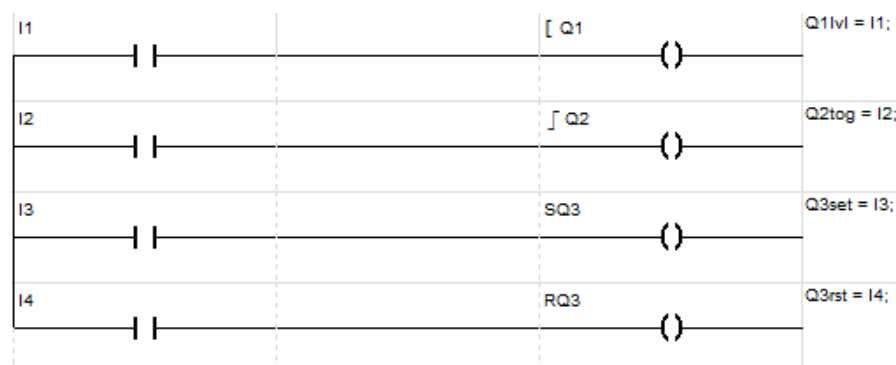
Pada tahapan proses hanya merupakan perintah pemanggilan fungsi *proses()*. Fungsi ini didefinisikan pada bagian lain program agar pengguna dengan mudah menuliskan logika ladder, dan tidak mengubah program template yang telah dibuat. Fungsi proses inilah yang menjadi kunci kesamaan logika ladder pada PLC dengan persamaan boolean dalam program bahasa C. Tahapan terakhir pada fungsi loop adalah mengirimkan logika variabel output hasil dari proses ke port output yang bersesuaian.

Output

Kunci utama menggunakan program arduino yang sudah mempunyai template adalah fungsi proses yang harus diisi program yang ekuivalen dengan diagram ladder. Output pada zelio logic mempunyai 4 mode:

1. Mode kontaktor
Pada mode koil akan aktif jika sambungan terhubung, kalau tidak makan koil akan mati.
2. Mode relay impulse
Pada mode ini koil akan berubah status menjadi *toggle* jika mendapat pulsa rising edge
3. Mode latch (set)
Pada mode ini koil akan aktif seketika sambungan terhubung, dan akan tetap terus aktif sampai ada perintah reset.
4. Mode unlatch (reset)
Pada mode ini koil akan mati jika sambungan reset terhubung.

Gambar berikut memperlihatkan contoh diagram ladder untuk keempat fungsi tersebut.



Gambar 2. Diagram ladder untuk 4 mode output pada zelio

Persamaan boolean dalam bahasa C untuk diagram ladder di atas dapat dituliskan sebagai berikut:

Kode 2.

```
{  
  Q1lv1 = I1; outQ1lv1(Q1lv1);  
  Q2tog = I2; outQ2tog(Q2tog);  
  Q3set = I3; outQ3set(Q3set);  
  Q3rst = I4; outQ3rst(Q3rst);  
}
```

Dapat dilihat persamaan boolean dalam bahasa C pada kode 2 sangat sederhana dan terlihat jelas kesamaannya dengan diagram ladder pada gambar 2. Setiap satu output akan menjadi satu baris fungsi boolean. Setelah instruksi boolean terdapat instruksi memanggil fungsi yang bersesuaian dengan nama outputnya. Fungsi ini dituliskan pada bagian lain di program dan akan menjadi template karena tidak akan berubah selama rangkaian hardware tidak berubah. Fungsi-fungsi tersebut dapat dilihat pada kode 3. Pada kode tersebut hanya diperlihatkan template untuk empat mode output Q1. Untuk Q2 sampai dengan Q7 formatnya sama, hanya berbeda nomor outputnya saja.

Kode 3. Fungsi template output.

```
void outQ1lv1(bool Qin) {  
  Q1 = Qin;  
}  
  
void outQ1tog(bool Qin) {  
  if(not Q1old and Qin) Q1 = not Q1;  
  Q1old = Qin;  
}  
  
void outQ1set(bool Qin) {  
  if(Qin) Q1 = 1;  
}  
  
void outQ1rst(bool Qin) {  
  if(Qin) Q1 = 0;  
}
```

Counter

Counter pada zelio diberi nama dengan inisial C, misalnya C1 untuk counter 1. Setiap counter mempunyai atribut nilai counter aktif(CC), nilai pembanding(diset dengan menu window), arah counter (DC) dan reset counter (RC). Untuk itu program dalam bahasa C harus mempunyai komponen tersebut yang dideklarasikan pada awal program dan menjadi bagian template, sebagai berikut:

```
bool C1,CC1,CC1old, DC1,RC1;
```

dan juga dibuatkan template untuk fungsi outputnya pada kode 4 berikut

Kode 4. Fungsi untuk counter

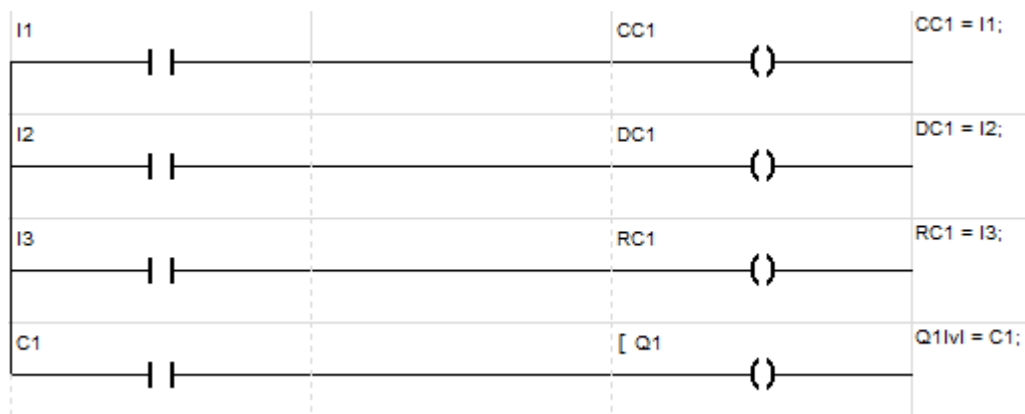
```
void outCC1(bool CC1in) {  
  if(not CC1old and CC1) {
```

```
    if(DC1)  C1ctr--;  
    else  C1ctr++;  
    C1=(C1ctr>=C1reg);  
  }  
  CC1old = CC1;  
}  
  
void outDC1(bool DCin){  
  DC1 = DCin;  
}  
  
void outRC1(bool RCin){  
  if(RCin) {  
    C1ctr=0;  
    C1 = 0;  
  }  
}
```

Pada program zelio nilai pembanding diset dengan menu berbasis *window*, sedangkan pada program arduino harus diset dengan menambahkan pemanggilan fungsi *setCounter()* pada fungsi *setup*. Adapun isi fungsi *setCounter* untuk satu counter aktif (Counter1) adalah sebagai berikut:

```
C1reg=4;
```

Contoh diagram ladder penggunaan counter pada zelio dapat dilihat pada gambar 3 berikut.



Gambar 3. Diagram ladder untuk Counter

Program arduino ekivalen dengan rangkaian pada diagram ladder tersebut adalah sebagai berikut. Potongan program ini ada pada fungsi proses.

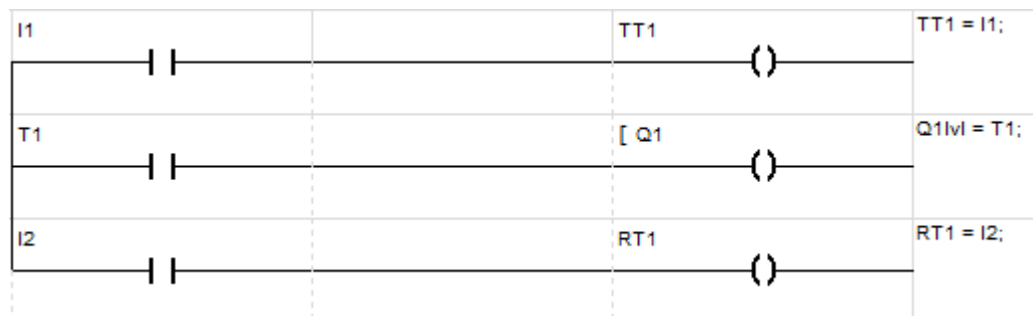
Kode 5.

```
void process()  
{  
  CC1 = I1; outCC1(CC1);  
  DC1 = I2; outDC1(DC1);  
  RC1 = I3; outRC1(RC1);  
  Q1/v1 = C1; outQ1/v1(Q1/v1);  
}
```


Timer

Timer pada zelio mempunyai banyak mode. Pada penelitian ini hanya menggunakan satu mode saja. Pada dasarnya timer mempunyai cara kerja yang sama dengan counter, bedanya kenaikan nilai counter-nya secara periodic. Untuk itu pada program arduino ditambahkan pustaka *TimerOne.h*. yang telah disediakan oleh IDE arduino. Pustaka ini menyediakan fungsi interupsi timer 1.

Contoh diagram ladder timer 1 (T1) pada zelio dapat dilihat pada gambar 4. Sedangkan program boolean C dapat dilihat pada kode 5.



Gambar 4. Diagram ladder fungsi timer pada zelio

Kode 5

```
void process()  
{  
  TT1 = I1; outTT1(TT1);  
  Q1lv1 = T1; outQ1lv1(Q1lv1);  
  TR1 = I2; outTR1(TR1);  
}
```

HASIL DAN PEMBAHASAN

Metode pemrograman sistem arduino dengan menggunakan fungsi boolean dalam bahasa C yang diusulkan pada penelitian ini memperlihatkan kesamaan atau ekuivalensi yang jelas dengan program ladder standar PLC. Persamaan boolean yang dituliskan berdasarkan diagram ladder sederhana memperlihatkan program tersebut bekerja dengan baik pada sistem PLC arduino. Lebih jauh peneliti telah mencoba dengan sistem yang lebih kompleks seperti sistem kuis bel cepat- tepat dan sistem lampu lalu lintas berdasarkan timer, memperlihatkan metode tersebut bekerja dengan benar.

Untuk membuktikan bahwa metode juga dapat dimengerti dengan mudah, peneliti telah melakukan survey dengan sampel sekitar 50 mahasiswa dengan dasar pengetahuan PLC dan diagram ladder serta dasar pemrograman bahasa C yang sangat minim. Sampel terlebih dahulu diajari cara membaca diagram ladder kemudian diajari membuat persamaan boolean dalam bahasa C. Hasil ujicoba memperlihatkan lebih dari 90% sampel mengerti dengan mudah cara memprogram sistem PLC arduino, asal telah tersedia program ladder-nya.

KESIMPULAN

Penelitian ini telah mengusulkan sebuah metode yang praktis dan mudah difahami agar sistem arduino dapat digunakan sebagai pengganti PLC. Program untuk arduino merupakan program Structured Text dalam istilah PLC, namun pengguna hanya perlu menuliskan bentuk boolean dalam bahasa C. Dimana persamaan boolean tersebut bisa langsung terlihat kesamaan dan keteraturannya dengan bahasa diagram ladder.

Beberapa fungsional PLC tidak dibuatkan bentuk boolean dalam bahasa C. Penulis beranggapan arduino (yang murah ini) belum saatnya digunakan untuk menggantikan sistem PLC yang sangat kompleks. Apabila suatu sistem otomasi sangat kompleks tentulah dimiliki oleh industri besar, yang berarti mempunyai dana yang cukup untuk menggunakan PLC dalam sistem otomasinya.

Pengembangan lebih lanjut

Beberapa fungsional PLC belum dibuatkan *templat*nya dalam bahasa C arduino, seperti komparator. Demikian juga mode untuk unit fungsional counter timer masih satu mode. Pada zelio logic counter dan timer mempunyai banyak mode.

Penelitian ini dapat dikembangkan dengan membuat program compiler grafis dari ladder diagram menjadi persamaan boolean dalam bahasa C, sehingga arduino dapat langsung diprogram dengan menggunakan bahasa diagram ladder.

Hal lain yang dapat didalami lebih lanjut adalah membandingkan hasil pada penelaitan ini dengan hasil pada penelitian lain untuk melihat efektifitas nya pada beberapa parameter, misalnya ukuran kode program, kecepatan eksekusi dan lainnya.

DAFTAR PUSTAKA

- Granvillano, Christian, 2014, *Arduino arduino as aprogrammable logic controller (PLC)*, Open Electronic article, <http://www.open-electronics.org/arduino-as-a-programmable-logic-controller-plc/>
- Westhues, Jonathan, An LDmicro Tutorial, <http://cq.cx/ladder-tutorial.pl>
- Artanto, Dian., *Ladder Diagram + C = Ladder Diagram Teks -> PLCArduino*, <http://plcarduino.blogspot.co.id/>
- Putra, Agfianto Eko, 2007, *PLC, konsep pemrograman dan aplikasi*, Gava Media, Yogyakarta
- Bolton, W., 2006, *Programmable Logic Controller*, Elsevier
- Arduino Nano Tutorial, <https://www.arduino.cc/en/Guide/ArduinoNano>