

Simulasi Komunikasi Serial Dengan Protokol I²C Menggunakan Arduino IDE dan Proteus 8

¹ Candra Pradhana, ² Mochamad Sulaiman

¹ Teknik Elektro, Universitas Islam Raden Rahmat, Malang

² Teknik Elektro, Universitas Islam Raden Rahmat, Malang

¹candraphysics@gmail.com, ² sule.kmp@gmail.com

Abstract – The main goal of this research is to understanding serial communication between microcontroller especially Arduino UNO R3 (hardware) and Arduino IDE (software), this communication is very important concerned with data transmission due to bit data per time. With I²C protocol, serial communication is easier dan simple, Arduino R3 support this feature with its A4 and A5 pins. After designed has finished, Arduino as *Master* and *Slave* were coded and after compilation we should find the .Hex extention file and should uploaded into Proteus 8 (Simulino Arduino). The simulation was interfaced by virtual terminal in Proteus 8 or LCD 16x2 I²C.

Keywords — Arduino IDE; Serial Communication, I²C, Proteus 8, Bit Data

Abstrak—Tujuan dari penelitian ini adalah untuk memahami komunikasi serial antar mikrokontroler terutama Arduino UNO R3, komunikasi serial sangat penting terkait dengan transmisi data. Data yang dikirim adalah per bit per waktu. Dengan protokol I²C komunikasi data serial akan lebih cepat dan rangkaian lebih sederhana terutama pada kasus digunakan lebih dari satu mikrokontroler (*Master*) ataupun banyak sensor, pin A4 dan A5 Arduino merupakan fitur ini. Dilakukan perancangan dan simulasi komunikasi serial dengan Arduino UNO R3 (Berupa Simulino pada Proteus 8), perancangan dimulai dengan Arduino IDE untuk *Master* dan *Slave*, setelah kompilasi, kode pada Arduino IDE berupa file.Hex akan di unggah ke proteus 8 sehingga simulasi dapat dijalankan dan tampilan bisa dilihat melalui virtual terminal ataupun LCD 16x2 I²C.

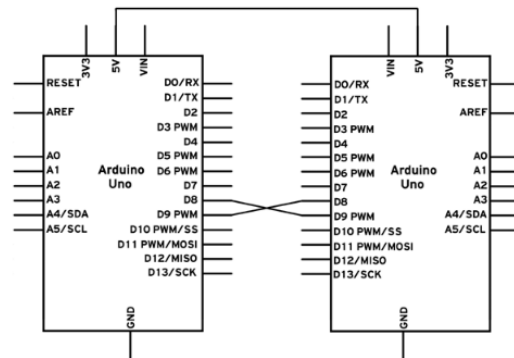
Kata Kunci—Arduino IDE; Komunikasi Serial; I²C; Proteus 8, Bit Data

I. PENDAHULUAN

Dalam era revolusi industri 4.0 tentunya penggunaan teknologi begitu pesat dan terus berkembang, dalam transmisi data komunikasi serial sangat berperan peting dan tak terpisahkan. Secara umum komunikasi serial menjadi pelopor dalam platform digital [1]. Komunikasi Serial

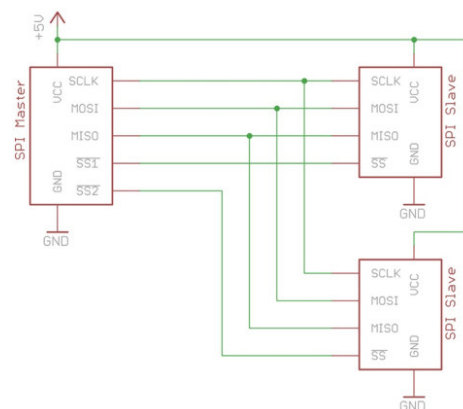
adalah proses pengiriman data per bit dalam waktu tertentu, Pada umumnya mikrokontroler mengirimkan data memakai protokol UART,SPI, dan I²C. Ketiga protokol komunikasi serial tersebut mempunyai keunggulan tersendiri, akan tetapi untuk banyak sensor dan mikrokontroler semisal kita ingin menghubungkan dua *Master* yaitu dua Arduino Ataupun Arduino dengan Raspberry maka protokol I²C lebih tepat. Berikut adalah tiga skema komunikasi serial yang masing-masing mempunyai kekurangan dan kelebihan tersendiri. Dari ketiga protokol hanya I²C lah yang mampu mempunyai banyak *Master* (mikrokontroler).

UART (*Universal Unsynchronous Receiver Transmitter*)



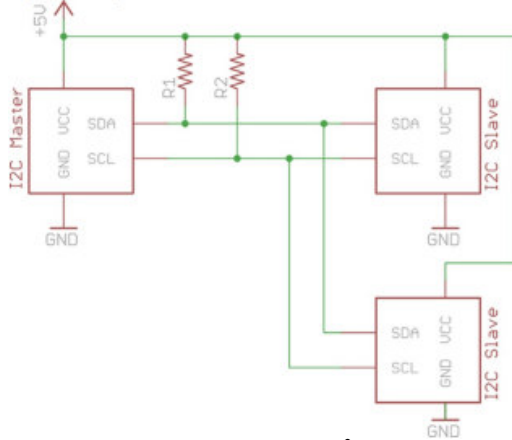
Gambar 1. Protokol UART[2]

SPI (*Serial Peripheral Interface*)



Gambar 2. Protokol SPI[3]

I²C (Inter Integrated Circuit)



Gambar 3. Protokol I²C[3]

Dalam Protokol I²C ada dua kabel bernama SDA (*serial data line*) dan SCL (*serial clock line*), *master* bertugas memulai komunikasi dan *clock*, sedangkan *slave* yang dituju oleh *master*[4]. Kesuksesan dalam membangun protokol komunikasi I²C adalah integritas desain elektronika dan kesederhanaan dalam perangkat lunak[5].

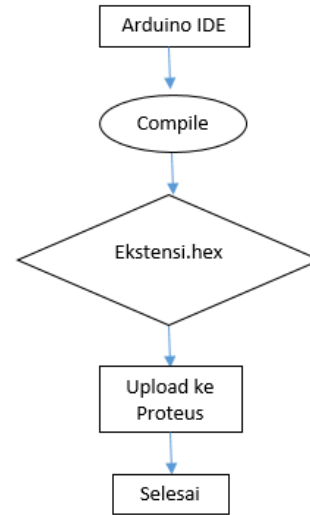
Arduino IDE (*Integraterd Development Enviroment*) adalah perangkat lunak untuk Mikrokontroler yang menggunakan bahasa C atau C++, yang mana pengguna dapat mengupload program melalui kabel USB [6]. Dari Arduino IDE kita bisa merancang program untuk komunikasi serial dan setelah dilakukan proses kompilasi maka Arduino IDE akan menampilkan *file* ekstensi.Hex (Lakukan Setting Arduino IDE untuk menampilkan proses kompilasi). *Library* Arduino IDE pertama-tama dibutuhkan untuk simulasi dengan Proteus 8. Dalam Penelitian ini memakai Proteus 8 yang merupakan perangkat lunak elektronika yang memiliki SPICE untuk simulasi interaktif [7].

METODE PENELITIAN

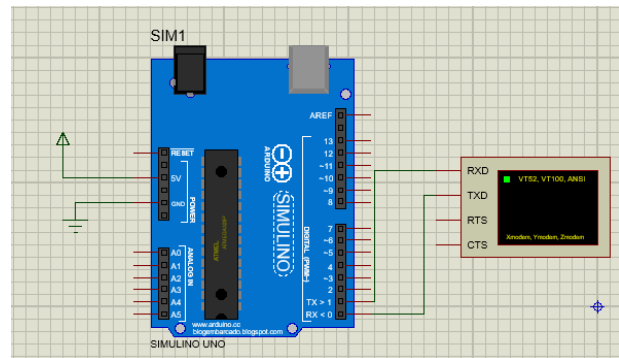
A. Metode

Penulisan kode program pada Arduino IDE dimaksudkan untuk *Master-Slave* yang mana *Master* akan memerintah *Slave*, program Arduino IDE pada *Master* dan *Slave* akan diimplementasikan pada proses simulasi dengan Proteus 8. Pada gambar 4 setelah proses *compile*, akan nampak proses pada Arduino IDE seperti pada gambar 6, Proteus 8 membutuhkan *file* dalam format hexadesimal (.Hex),

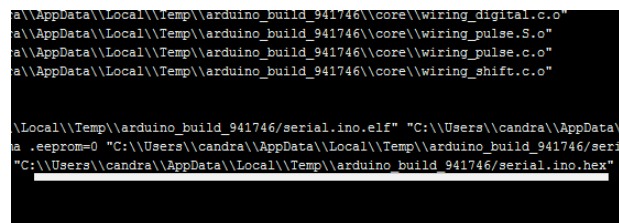
sehingga kita terlebih dahulu mencari file.Hex pada Arduino IDE.



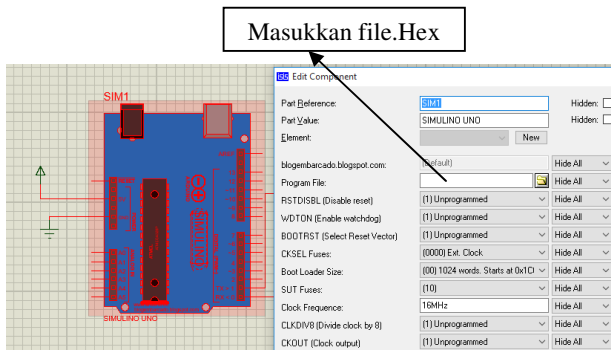
Gambar 4. Diagram Alir Simulasi



Gambar 5 . Rangkaian Arduino Proteus



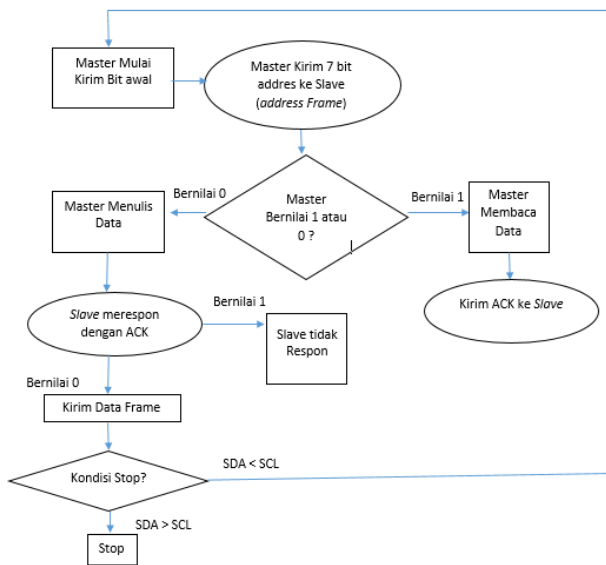
Gambar 6 . Proses Mencari File.Hex



Gambar 7 . Masukkan file dari Arduino ke Proteus

B. Protokol I²C

Protokol I²C (*Inter-Integrated Circuit*) memungkinkan komunikasi dua arah secara lebih cepat dengan menggunakan pin input/output dengan minimal [6]. Pada penelitian ini digunakan Library wire Arduino dan pin A4 dan A5 agar bisa berkomunikasi secara I²C.



Gambar 8. Protokol I²C Master-Slave

Library Wire pada Arduino IDE mempunyai beberapa fungsi sebagai berikut:

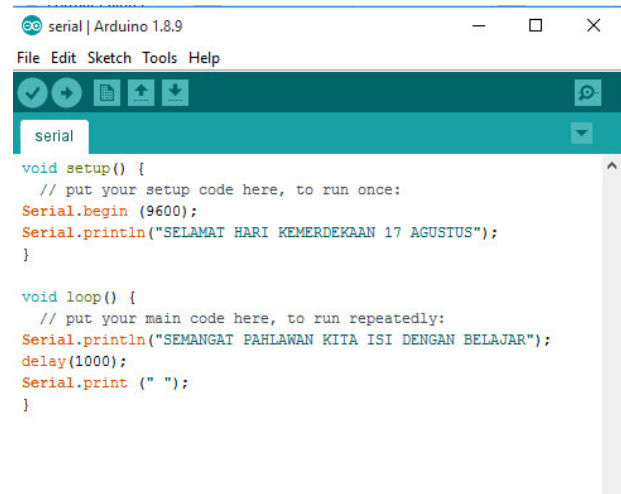
- begin()* - Setting Arduino sebagai *Master* atau *Slave*
- requestFrom()* – Digunakan oleh *Master* untuk meminta data dari *Slave*
- beginTransmission()* – *Master* mengirim data ke *Slave* tertentu

- endTransmission()* – digunakan oleh *Master* untuk mengakhiri
- write()* – bisa digunakan oleh *Master/Slave* untuk mengirim data
- available()* – untuk menentukan jumlah Byte
- read()* - membaca byte data dari bus I²C
- SetClock()* – digunakan *Master* untuk mengatur frekuensi
- OnReceive()* - digunakan *Slave* apabila data diterima *Master*
- OnRequest()* – digunakan *Slave* apabila *Master* meminta

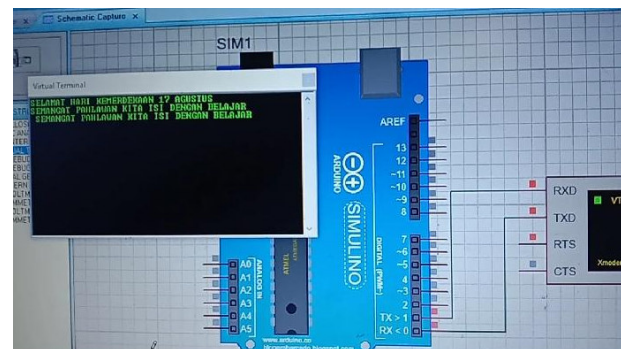
II. HASIL DAN PEMBAHASAN

C. Simulasi Komunikasi Serial dengan Proteus 8

Virtual Terminal akan menampilkan output dari simulasi pada proteus 8, hal ini mirip dengan serial monitor pada Arduino IDE.



Gambar 9 . Arduino IDE



Gambar 10 . Simulasi dengan Virtual Terminal

D. Simulasi Komunikasi Serial dengan I²C
 Listing Kode Arduino IDE untuk Master

```
#include <Wire.h>

void setup() {

    Wire.begin(); // Protokol I2c, alamat optional untuk
    Master

}

byte x = 0;

void loop() {

    Wire.beginTransmission(9); // Kirim ke Alamat #9

    Wire.write("x is "); // Kirim string

    Wire.write(x); // Kirim dalam bit

    Wire.endTransmission(); // Berhenti

    x++; //Penambahan

    delay(500);

}

Listing Kode Arduino IDE untuk Slave
```

```
#include <Wire.h>

void setup() {

    Wire.begin(9); // Bus alamat I2C #9

    Wire.onReceive(receiveEvent);

    Serial.begin(9600); // Mulai Serial Output

}

void loop() {

    delay(100);

}

// function that executes whenever data is received from
Master

// this function is registered as an event, see setup()

void receiveEvent(int howMany) {
```

```
while (1 < Wire.available()) { // loop through all but the
last

    char c = Wire.read(); // Ubah Byte sebagai karakter

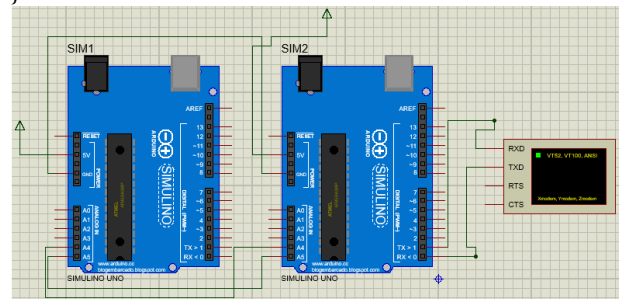
    Serial.print(c); // Tampilkan Karakter

}

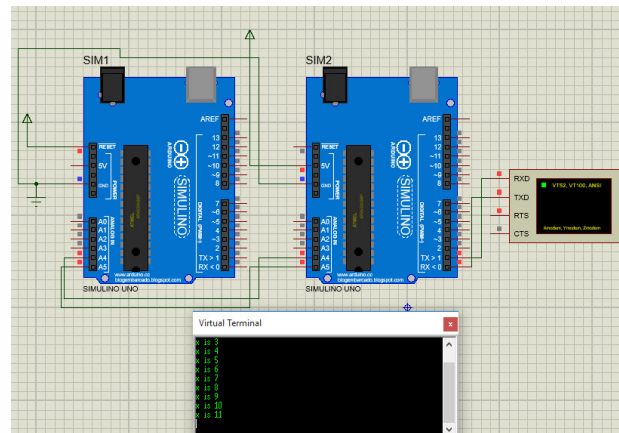
int x = Wire.read(); // Terime Byte integer

Serial.println(x); // Tampilkan integer

}
```

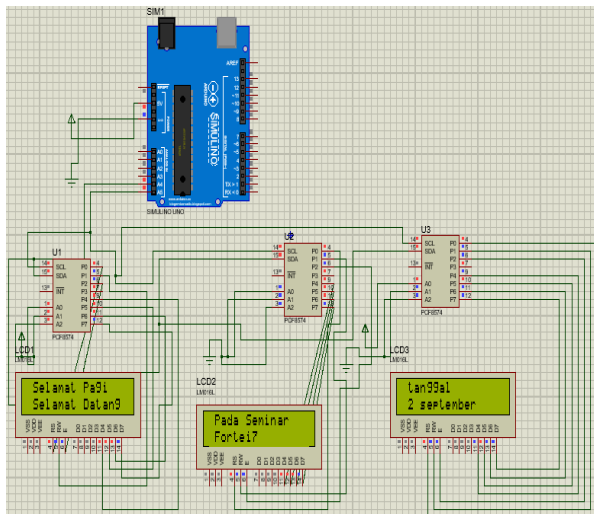


Gambar 11 . Simulasi Arduino sebagai Master-Slave



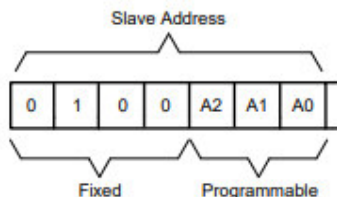
Gambar 12. Hasil Tampilan Simulasi Dua Arduino

Mikrokontroler umumnya sebagai *Master*, pada gambar 11 Arduino yang bertindak sebagai *Master* dan sebagai *Slave*, sebagai *Master* berarti sebagai pemberi perintah. Pada Gambar 11 dan 12 bertujuan untuk menampilkan *string* dan bit data. Berupa kata (“x is...”) dengan penambahan (*increment*). Kode program dibutuhkan baik untuk *Master* ataupun *slave*, yaitu dengan mengunggah tiap file.Hex ke masing-masing Arduino pada Proteus 8 (Simulino).



Gambar 13. Simulasi I²C LCD 16x2

Pada gambar 13 dilakukan simulasi LCD 16x2 dengan PCF8574 sebagai *extender I/O*, terlihat bahwa hanya butuh dua kabel yaitu SCL dan SDA ke Mikrokontroler Arduino, hal ini akan lebih meringkas kabel yang masuk ke Arduino. Yang penting adalah pengalamatan (*addressing*) dari PIC8574, karena jika alamatnya sama maka akan mengalami *error*, salah satu cara mencari alamat *slave I²C* adalah dengan melihat *datasheet* dari komponen selain memakai cara *I²C scanner*.



Gambar 14. Alamat dari PCF8574 [8]

Dari gambar14 kita bisa membuat alamat dengan A2,A1,A0 pada PCF8574, total kemungkinan adalah $2^3 = 8$ alamat. Pada gambar 13 menggunakan 3 alamat saja yaitu:

- a) 0100111=0x27 hex
- b) 0100000=0x20 hex
- c) 0100001=0x21 hex

Listing Program 3 LCD I²C Arduino IDE:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd1 (0x27,16,2);
```

```
LiquidCrystal_I2C lcd2 (0x20,16,2);
LiquidCrystal_I2C lcd3 (0x21,16,2);
void setup()
lcd1.begin();
lcd2.begin();
lcd3.begin();
lcd1.backlight();
lcd2.backlight();
lcd3.backlight();
lcd1.setCursor (1,0);
lcd1.print ("Selamat Pagi");
lcd1.setCursor(1,1);
lcd1.print ("Selamat Datang");
lcd2.setCursor (1,0);
lcd2.print ("Pada Seminar");
lcd2.setCursor(1,1);
lcd2.print ("Fortei7");
lcd3.setCursor (1,0);
lcd3.print ("tanggal");
lcd3.setCursor(1,1);
lcd3.print ("2 september");
}
void loop() {
}
```

III. KESIMPULAN

Simulasi Komunikasi Serial dengan Arduino+Proteus telah berjalan sebagaimana mestinya, data yang dikirim dapat berupa *string*, angka dapat ditampilkan melalui *virtual terminal* ataupun LCD I²C, hal yang perlu diperhatikan dalam Protokol ini adalah pengalamatan dari sensor ataupun piranti eksternal yang merupakan *slave*. Pengalamatan (*addressing*) tiap piranti berbeda sehingga perlu melihat *datasheet* dahulu. Protokol I²C mempunyai keunggulan yaitu mampu menampung 128 piranti luar dan hanya butuh dua kabel ke mikrokontroler sehingga berguna jika membutuhkan banyak sensor ataupun mikrokontroler, perangkat lainnya, kedepannya diharapkan dapat menambah pemahaman tentang komunikasi serial dengan menggunakan simulasi secara *hardware* dengan banyak sensor semisal RTC,Gyroscope,Ultrasonik I²C karena keterbatasan *library* simulasi pada Proteus 8

IV. DAFTAR PUSTAKA

[1] L. Louis, "WORKING PRINCIPLE OF ARDUINO AND USING IT AS A TOOL FOR STUDY AND RESEARCH," *Int. J. Control. Autom. Commun. Syst.*, vol. 1, no. 2, pp. 21–29, 2016, doi: 10.5121/ijcacs.2016.1203.

[2] C. Amariei, *Arduino Development Cookbook*. Birmingham-Mumbai: PACKT, 2015.

- [3] J. Blum, *Exploring Arduino*. Wiley.
- [4] V. K. Pandey, "A Review Paper on I2C Communication Protocol," *Int. J. Adv. Res. Ideas Innov. Technol.*, vol. 4, no. 2, pp. 340–343, 2018.
- [5] A. Jacob, W. Nurshazwani, W. Zakaria, M. Razali, and B. Tomari, "EVALUATION OF I2C COMMUNICATION PROTOCOL IN DEVELOPMENT OF MODULAR CONTROLLER BOARDS," *ARPJ J. Eng. Appl. Sci.*, vol. 11, no. 8, pp. 4991–4996, 2016.
- [6] A. Wibowo, "Communication Concept Between Bluetooth As a Master and Slave To Exchange Digital Information," *Int. J. Eng. Adv. Technol.*, vol. 9, no. 2, p. ISSN: 2249 – 8958, 2019, doi: 10.35940/ijeat.B3222.129219.
- [7] S. Fuada and U. P. Indonesia, "Perancangan sistem Kontrol Pada Prototip Pengeriing Kerupuk Berbasis IC Digital Menggunakan Software Proteus 7.0," *J. Ilm. Setrum*, no. September, 2017, doi: 10.36055/setrum.v6i1.531.
- [8] T. Instrument, "PCF8574 Remote 8-Bit I / O Expander for I 2 C Bus," 2015.