

Algoritma Komunikasi USART dengan Metode *Normal Speed dan Double Speed*

Andre Mochammad Satrio¹, Mohammad Mujirudin¹, Ahmad Kadarisman²
Masbah Rotuanta Tagore Siregar², Latifah Sarah Supian³, Harry Ramza¹

¹Program Studi Teknik Elektro, Fakultas Teknik
Universitas Muhammadiyah Prof. Dr. HAMKA
Jalan Tanah Merdeka No. 6, Kp Rambutan, Jakarta 13830, INDONESIA
Email : andremochammadsatrio81@gmail.com, mujirudin@uhamka.ac.id, hramza@uhamka.ac.id

²Program Studi Teknik Elektro, Fakultas Pascasarjana
Institut Sains dan Teknologi Nasional
Jalan Moh. Kahfi II, Srengseng Sawah, Jagakarsa Jakarta 12640, INDONESIA
Email : ariskig@gmail.com, mrtsiregar2012@istn.ac.id

³Department of Electrical and Electronics Engineering, Faculty of Engineering
National Defence University of Malaysia,
Kem Sg. Besi, 57000, Kuala Lumpur, MALAYSIA
Email : cawa711@gmail.com

Abstrak

Tulisan ini menjelaskan tentang algoritma komunikasi serial *Universal Synchronous Asynchronous Transmitter and Receiver* (USART) pada mikrokontroler dengan perangkat lunak MATLAB. Agar komunikasi serial tersebut dapat bekerja, diperlukan konfigurasi yang sesuai. Sehingga diperoleh waktu yang diperlukan untuk transmisi data berdasarkan beban data. Metode yang digunakan untuk mendapatkan nilai waktu adalah pengukuran waktu menggunakan fungsi waktu ("tic" dan "toc") pada MATLAB. Pengukuran tersebut dilakukan ketika data dikirim dari MATLAB ke mikrokontroler dan kembali ke MATLAB. Jenis transmisi serial yang digunakan pada pengukuran adalah *Asynchronous Normal Speed* dan *Asynchronous Double Speed*. Jenis transmisi serial *Asynchronous Double Speed* memiliki konfigurasi *Baud Rate* dua kali lebih besar dari *Asynchronous Normal Speed*. Nilai *Baud Rate* tersebut sebesar 2 Mbaud dan Waktu yang diperoleh ketika jenis transmisi serial *Asynchronous Double Speed* yang digunakan dan beban data sebesar 20 byte adalah selama 0,0117192 detik. Ketika beban data sebesar 1000 byte, waktu yang dibutuhkan transmisi adalah selama 0,031719 detik. Namun ketika jenis transmisi serial *Asynchronous Normal Speed* yang digunakan dan beban data sebesar 20 byte, waktu yang dibutuhkan untuk transmisi adalah selama 0,012719 detik. Ketika beban data mencapai 1000 byte, waktu yang dibutuhkan untuk transmisi adalah selama 0,031394 detik. Sehingga waktu yang dibutuhkan untuk transmisi data meningkat seiring dengan bertambahnya beban data.

Kata kunci—Komunikasi Serial, USART, *Asynchronous*, Mikrokontroler, MATLAB.

Abstract

This paper describes a *Universal Synchronous Asynchronous Transmitter and Receiver* (USART) serial communication algorithm on the microcontroller with the software MATLAB. The appropriate configuration is necessary for serial communication and it's obtained the time requirements of the data transmission based on the data load. The methodology of this paper applies the time measurement of the MATLAB function ("tic" and "toc"). It was sent from MATLAB to microcontroller and returned to MATLAB. The serial transmission mode were used on this measurements are *Asynchronous Normal Speed* and *Asynchronous Double Speed*. The transmission of *Double Speed Asynchronous* mode having *Double Baud Rate* configuration than *Normal Speed*. The value of *Baud Rate* amount of 2 MBaud and the *Asynchronous* mode on the normal speeds are 20 data byte and 1000 data byte, it requires of transmission timing around 0,012719 seconds and 0,031394 seconds respectively. Moreover on the normal speed asynchronous mode of 20 data byte and 1000 data byte, it requires of transmission timing around 0,0117192 seconds and 0,031719 seconds respectively. Therefore, it can get the short statement that the time requirements is proportional comparative with the data load.

Keywords—Serial Communication, USART, *Asynchronous*, Microcontroller, MATLAB.

Received 1 November 2020; Accepted 11 Desember 2020

1. Pendahuluan

Seiring dengan perkembangan zaman, teknologi komunikasi berkembang pesat. Terutama komunikasi

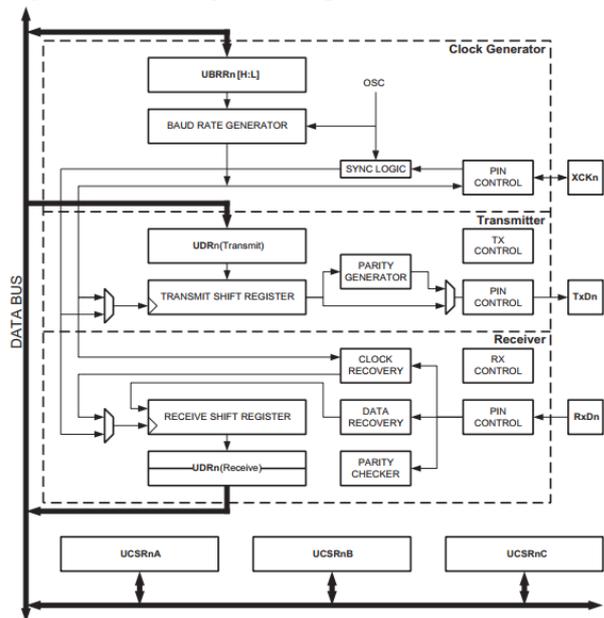
antar perangkat elektronik yang telah banyak menggunakan komunikasi serial. Dibanding dengan komunikasi paralel, komunikasi serial memiliki beberapa

keunggulan seperti biaya yang rendah dan perawatan yang rendah. Salah satu perangkat elektronik yang menggunakan komunikasi serial yaitu USART. USART merupakan protokol komunikasi serial dua arah [1]. Protokol komunikasi tersebut digunakan untuk komunikasi data antar perangkat dengan jarak transmisi yang pendek serta kecepatan transmisi yang rendah. USART dapat mengubah data dari transmisi paralel ke transmisi serial pada *Transmitter* menggunakan *shift register*. Selain itu, pada *Receiver* data diubah dari transmisi serial ke transmisi paralel [2]. Proses tersebut dilakukan tiap bit dalam satu waktu.

Berdasarkan kecepatan transmisi data, jenis transmisi *Asynchronous* USART di beberapa perangkat memiliki dua jenis transmisi. Jenis transmisi *Asynchronous* tersebut yaitu *Normal Speed* dan *Double Speed*. Jenis transmisi *Asynchronous Double Speed* memiliki kecepatan transmisi data dua kali dari jenis transmisi *Normal Speed* [4]. Agar sistem dari kedua jenis transmisi tersebut bekerja, diperlukan suatu algoritma yang sesuai. Kemudian, sistem tersebut dirancang dan diujicobakan menggunakan fungsi waktu berdasarkan beban data (*byte*). Sehingga diperoleh nilai waktu yang dibutuhkan untuk transmisi data.

2. Komunikasi Serial USART

Komunikasi serial USART memiliki tiga komponen utama. Komponen tersebut terdiri dari *Clock Generator*, *Transmitter* dan *Receiver* [3][4]. Berdasarkan [4], blok diagram USART dapat dilihat pada Gambar 1.



Gambar 1. Blok Diagram USART Pada Mikrokontroler

Clock generator berfungsi sebagai *synchronization logic* untuk masukkan *clock* eksternal yang digunakan pada operasi *synchronous slave*. Selain itu, *clock generator* juga berfungsi sebagai *Baud Rate generator*. Komponen *Transmitter* terdiri dari berbagai *logic* seperti *buffer* tulis tunggal untuk menyimpan data sementara, *Transmit Shift Register* untuk menggeser bit per bit keluar, *Parity Generator* untuk membangkitkan *bitparity* dan *Control*

logic untuk mengolah *frame format* yang berbeda. Komponen *Receiver* terdiri berbagai *logic* yaitu *Control logic*, *Parity Checker* untuk memeriksa galat pada data yang diterima, *Data Recovery* untuk memulihkan data, *Clock Recovery* untuk memulihkan sinyal *clock*, *Receive Shift Register* untuk menggeser bit masuk dan *buffer* baca tunggal untuk menyimpan data sementara. [4]

2. 1. Clock Generation

Clock generation logic merupakan basis *clock* yang digunakan untuk transmisi data. Pada *clock generator* terdapat *Baud Rate Generator logic*. *Baud Rate Generator* merupakan pembagi frekuensi sistem [2]. Agar *Baud Rate generator* pada *clock logic* bekerja, diperlukan pengaturan pada register USART *Baud Rate Register n* (*UBRRn*). Register *UBRRn* terdiri atas dua *byte* register yaitu USART *Baud Rate Register n Low* (*UBRRnL*) yang memiliki bit ke-0 sampai ke-7 dan USART *Baud Rate Register n High* (*UBRRnH*) yang memiliki bit ke-8 sampai ke-12. Pada beberapa mikrokontroler, komunikasi serial USART memiliki dua jenis transmisi *Asynchronous* yang berbeda. Jenis transmisi tersebut terdiri dari *Asynchronous Normal Speed* dan *Asynchronous Double Speed*. Jenis transmisi tersebut dapat diatur pada register USART *Control and Status Register n A* (*UCSRnA*) bit ke-1 yaitu *Double The USART Transmission Speed* (*U2Xn*). [4][2]

Persamaan yang digunakan untuk menemukan nilai *UBRRn* pada jenis Transmisi *Asynchronous* adalah dengan membagi frekuensi osilator dengan konstanta 16 dan nilai *Baud Rate* yang dikehendaki. Namun pada jenis transmisi *Asynchronous Double Speed*, konstanta pembagi tersebut bernilai setengah dari *Asynchronous Normal Speed* yaitu 8. Hasil dari perhitungan tersebut kemudian dikurangi dengan konstanta satu. Formula tersebut dapat dilihat pada persamaan berikut [4]:

$$UBRRn = \frac{F_{osc}}{k * BAUD} - 1 \tag{1}$$

Dimana,

BAUD = *Baud Rate* (bit per detik atau bps)

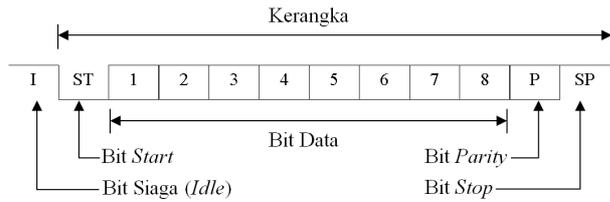
F_{osc} = Frekuensi osilator sistem (Hz)

UBRRn = Nilai dari register *UBRRnL* dan *UBRRnH* (0-4095)

k = konstanta pembagi yang memiliki nilai 8 atau 16.

2. 2. Kerangka Data

Kerangka data pada komunikasi serial memiliki beberapa kombinasi bit. Kombinasi tersebut terdiri dari bit data, bit untuk sinkronisasi data yaitu *bitstart* dan *bitstop*, serta *bitparity* untuk memeriksa galat pada data. Bit data memiliki konfigurasi 5, 6, 7, 8 atau 9 bit data. Bit sinkronisasi memiliki 1 *bitstart* dengan sinyal rendah dan 1 atau 2 *bitstop* dengan sinyal tinggi. *Bitparity* memiliki konfigurasi 1 bit yaitu bit ganjil atau bit genap. Kerangka data dapat dilihat pada gambar berikut [4][5][6][2]:

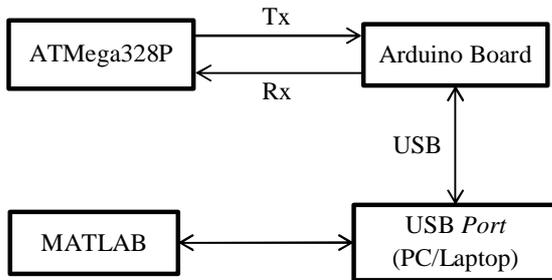


Gambar 2. Kerangka Data pada Komunikasi USART

Berdasarkan Gambar 2, kerangka data USART dimulai dari bitstart kemudian bit data. Setelah bit data, terdapat bitstop. Namun jika bitparity diatur, maka bit tersebut disisipkan setelah bit data dan sebelum bitstop. Ketika kerangka data tersebut ditransmisikan, kerangka data bisa diikuti oleh kerangka data baru secara langsung atau bisa diatur ke dalam kondisi siaga.

3. Perancangan Sistem dan Pembahasan

Sistem yang dirancang merupakan program komunikasi USART pada mikrokontroler dengan perangkat lunak MATLAB. Diagram alir sistem dapat dilihat pada Gambar 3.

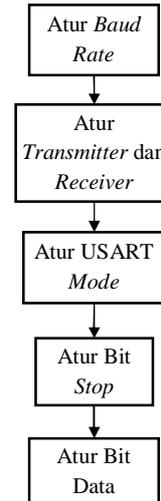


Gambar 3. Perancangan Sistem Komunikasi USART

Berdasarkan diagram alir yang ditunjukkan oleh Gambar 3 bahwa data dapat ditransmisikan dari IC ATmega328P ke MATLAB atau sebaliknya. Data tersebut melalui Arduino Board untuk dikonversi ke USB dari serial. Kemudian dari USB Arduino Board masuk ke USB Port PC/Laptop. Jarak USB tetap yaitu sepanjang 50 cm.

3. 1. Diagram Alir Program Transmisi Serial

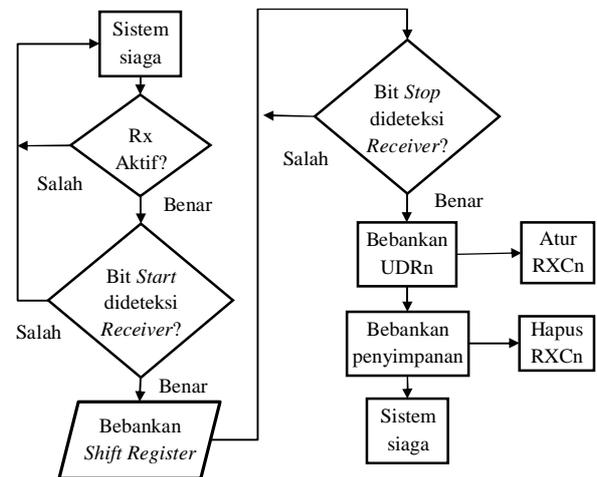
Pada penelitian ini terdapat tiga proses pada transmisi mikrokontroler yaitu Inisialisasi sistem, Transmit dan Receive. Pada Proses inisialisasi sistem, diperlukan beberapa konfigurasi. Konfigurasi tersebut dapat dilihat pada Gambar 4.



Gambar 4. Konfigurasi Komunikasi USART

Berdasarkan gambar tersebut, proses konfigurasi sistem dimulai dari menentukan nilai Baud Rate yang akan digunakan. Baud Rate dapat diatur pada register UBRRn byte UBRRnL dan UBRRnH. Setelah Baud Rate diatur, kemudian diaktifkan Transmitter dan/atau Receiver. Transmitter dan Receiver dapat diatur pada register USART Control and Status Register n B (UCSRnB) bit Receiver Enable n (RXENn) untuk Receiver dan Transmitter Enable n (TXENn) untuk Transmitter. Jenis transmisi USART, bit Stop dan bit Data dapat diatur pada register USART Control and Status Register n C (UCSRnC). Setelah inisialisasi sistem diatur, mikrokontroler berada pada kondisi siaga menerima atau mengirim data[4].

Proses menerima data pada komunikasi serial USART dapat dilihat pada Gambar 5[7][4].

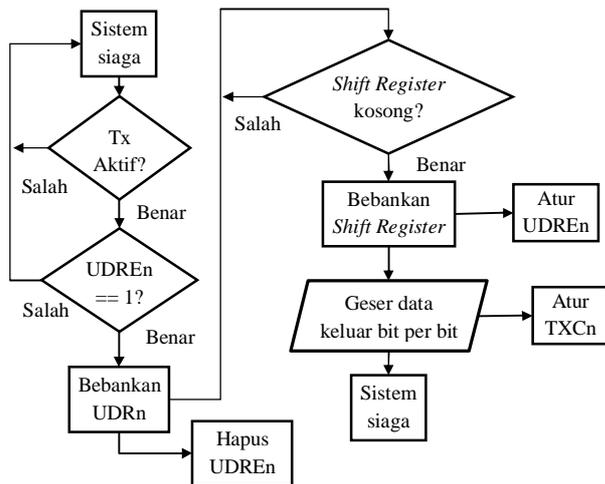


Gambar 5. Proses Menerima Data Pada Komunikasi USART

Proses menerima data pada diagram alir yang ditunjukkan pada Gambar 5 diawali dengan sistem siaga menerima data. Kemudian sistem memeriksa apakah Receiver aktif atau tidak. Sistem juga memeriksa bitstart pada data yang diterima. Jika bitstart dideteksi, Shift Register akan dibebankan oleh data yang diterima. Hal

tersebut berlangsung sampai bit $stop$ dideteksi sistem. Jika bit $stop$ telah terdeteksi, $buffer$ atau USART $Input/Output$ Data Register n (UDR n) akan dibebankan oleh data yang diperoleh dari $Shift$ Register. Kemudian atur bit USART $Receive$ Complete n (RXC n) pada Register UCSR n A. Data akan dipindahkan ke penyimpanan sementara. Kemudian bit RXC n dihapus dan sistem akan kembali ke kondisi siaga.

Proses mengirim data pada komunikasi serial USART dilihat pada Gambar 6[7][4].



Gambar 6. Proses Mengirim Data Pada Komunikasi USART

Gambar 6 menunjukkan bahwa proses transmisi data diawali dengan sistem siaga. Kemudian sistem akan memeriksa apakah Transmitter enable atau tidak, jika tidak sistem tidak melanjutkan proses jika iya sistem ke proses berikutnya. Proses selanjutnya yaitu sistem memeriksa apakah bit USART Data Register Empty n (UDREN) pada Register UCSR n A bernilai 1 atau tidak. Register tersebut akan bernilai 1 jika $buffer$ kosong. Kemudian bebaskan UDR n dengan data yang hendak dikirim. Proses tersebut bersamaan dengan menghapus bit UDREN. $Shift$ Register juga diperiksa apakah kosong atau tidak. Jika kosong maka data dari UDR n akan dibebankan ke dalam $Shift$ Register. Hal tersebut bersamaan dengan set bit UDREN. Geser data keluar bit per bit melalui transmitter. Kemudian bit USART $Transmit$ Complete n (TXC n) diatur.

3. 2. Program Transmisi Serial

Berdasarkan diagram alir yang telah dipaparkan, program transmisi terdiri dari tiga bagian utama. Bagian pertama pada program transmisi USART adalah konfigurasi atau inisialisasi USART. Konfigurasi tersebut mengacu pada Gambar 4 dan memiliki beberapa tahap konfigurasi. Tahap awal untuk konfigurasi USART adalah mengatur $Baud$ Rate yang digunakan. Konfigurasi $Baud$ Rate dapat dilihat pada $coding$ berikut:

```
uint32_t BAUDRATE = 1000000;
unsignedint UBRR_Async = ((F_CPU / (BAUDRATE * 16UL)) - 1);
UBRR0H = (uint8_t)(UBRR_Async >> 8);
UBRR0L = (uint8_t)(UBRR_Async);
```

$Coding$ tersebut akan menghasilkan kecepatan transmisi data tercepat pada jenis transmisi $Asynchronous$ $Normal$ $Speedy$ yaitu sebesar 1 $Mbaud$. Berbeda dengan jenis transmisi $Asynchronous$ $Double$ $Speedy$ yaitu memiliki nilai $Baud$ Rate sebesar 2 $Mbaud$. Selain itu, konstanta pada persamaan yang ditunjuk $coding$ tersebut adalah bernilai 8 untuk jenis transmisi $Asynchronous$ $Double$ $Speed$.

Kemudian Transmitter dan Receiver dapat diaktifkan pada $coding$ berikut:

```
UCSR0B = (1 << RXEN0) | (1 << TXEN0);
```

Tahap selanjutnya yaitu mengatur jenis transmisi serial USART dan kerangka data. Jenis transmisi dan kerangka data tersebut dapat diatur pada $coding$ berikut:

```
UCSR0A |= 0;
UCSR0C = (1 << UCSZ01) | (1 << UCSZ00);
```

Baris kedua pada $coding$ tersebut akan mengaktifkan jenis transmisi $Asynchronous$ dan konfigurasi default dari kerangka data. Kerangka data tersebut terdiri atas 1 bit $start$, 8 bit data, tanpa bit $parity$ dan 1 bit $stop$. Selain itu, pada baris pertamam $coding$ akan mengaktifkan jenis transmisi $Asynchronous$ $Normal$ $Speed$. Untuk jenis transmisi $Asynchronous$ $Double$ $Speed$, nilai register baris pertama $coding$ perlu diubah menjadi 2.

Setelah dilakukan konfigurasi pada mikrokontroler, maka dilakukan perintah untuk menerima data. Perintah tersebut dapat dilihat pada $coding$ berikut:

```
unsignedchar TerimaData(void){
while(!(UCSR0A & (1 << RXC0)));
return UDR0;
}
```

Berdasarkan Gambar 5, sistem akan menunggu sampai bit RXC0 bernilai 1. Jika RXC0 bernilai 1, maka data yang diterima telah berada pada $buffer$. $Coding$ tersebut akan menerima dan menyimpan data tersebut di $buffer$.

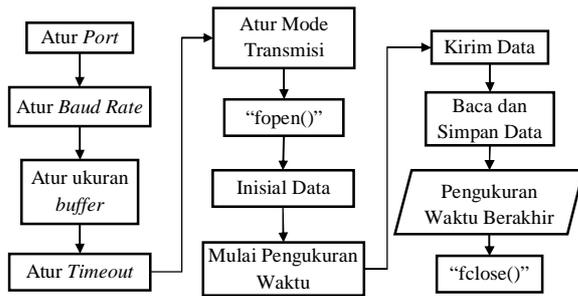
Selain perintah untuk menerima data, terdapat perintah untuk mengirim data. Perintah untuk mengirim data dapat dilihat pada $coding$ berikut:

```
void KirimData(unsignedchar indata){
while(!(UCSR0A & (1 << UDRE0)));
UDR0 = indata;
}
```

$Coding$ dapat digunakan untuk mengirim satu byte data. Sebelum data dikirim, sistem akan menunggu sampai $buffer$ kosong. Jika $buffer$ kosong, maka data tersebut akan dibebankan ke $buffer$ dan data akan dikirim menggunakan $shift$ register.

3. 3. Diagram Alir Program Transmisi Serial MATLAB

Diagram alir program transmisi serial pada perangkat lunak MATLAB yang digunakan pada penelitian ini dapat dilihat pada Gambar 7.



Gambar 7. Konfigurasi USART dan Proses Pengukuran Pada MATLAB

Berdasarkan Gambar 7, tahap awal transmisi serial data pada MATLAB yaitu konfigurasi komunikasi serial pada MATLAB disesuaikan dengan konfigurasi komunikasi serial pada mikrokontroler. Konfigurasi tersebut meliputi COM Port, Baud Rate, Buffer Size, Time Out dan jenis transmisi. Tahap selanjutnya adalah mengatur kondisi MATLAB dan mikrokontroler kedalam kondisi siaga dengan fungsi "fopen()". Kemudian, inisial data dari ukuran data 20 byte sampai 1000 byte dengan lompatan data sebesar 20 byte. Pengukuran waktu dimulai menggunakan fungsi "tic". Setelah waktu mulai diukur, data ditransmisikan ke mikrokontroler dengan fungsi "fprintf()" dan tunggu sampai data diterima dengan fungsi "fscanf()". Setelah data diterima, pengukuran waktu diakhiri dengan fungsi "toc". Proses diakhiri dengan fungsi "fclose()" yang menandakan bahwa komunikasi antar perangkat diputus.

3. 4. Program Transmisi Serial MATLAB

Konfigurasi yang ditunjukkan pada coding USART di mikrokontroler harus sesuai dengan konfigurasi komunikasi serial pada perangkat lunak MATLAB. Konfigurasi tersebut ditunjukkan pada coding berikut:

```
s=serial('COM3','BaudRate',1000000,'InputBuffer Size',10000,...
'OutputBuffer Size',10000,Timeout,30,'ReadAsyncMode','manual');
fopen(s);
```

Konfigurasi yang dipaparkan coding tersebut akan mengatur Baud Rate 1 Mbaud, Input Buffer 10 Kbyte, Output Buffer 10 Kbyte, time out 30 detik dan jenis transmisi Asynchronous. Selain itu pada baris terakhir coding, akan membuka jalur komunikasi antar mikrokontroler dengan MATLAB berdasarkan konfigurasi yang telah diatur.

Setelah konfigurasi pada perangkat lunak MATLAB, dilakukan perintah untuk mengirim data. Perintah tersebut dapat dilihat pada coding:

```
fprintf(s,dataTx);
```

Coding tersebut dapat mengirim data yang memiliki bentuk simbol atau karakter berdasarkan konfigurasi USART yang diatur. Data yang dikirim diperoleh dari variable "dataTx".

Selain program untuk mengirim data, terdapat program untuk menerima data. Program tersebut dapat dilihat padacoding berikut:

```
readasync(s)
[dataRx,countRx]=fscanf(s);
stopasync(s)
```

Coding tersebut dapat menerima data dan mengubah data ke bentuk simbol atau karakter. Data yang diterima akan disimpan pada variable "dataRx". Sedangkan variable "countRx" akan menampilkan informasi tentang jumlah dari data yang diterima. Pada jenis asynchronous diperlukan coding pada baris pertama. Coding tersebut dapat memulai menerima data dan membaca data dari buffer. Selain itu, diperlukan baris terakhir pada coding agar proses mengirim dan menerima data berhenti serta isi buffer dibersihkan.

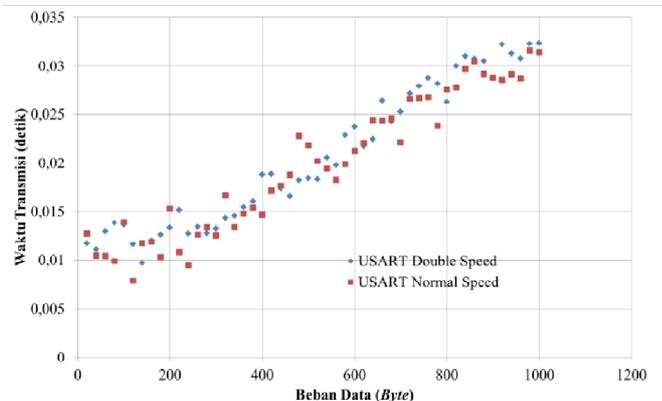
Pengukuran waktu pada sistem dilakukan menggunakan fungsi waktu. Fungsi tersebut dapat dilihat pada coding dibawah ini:

```
tic
%% User Program%%
WaktuTotal=toc
```

Baris pertama pada coding tersebut akan mengaktifkan dan memulai pengukuran waktu. Berdasarkan Gambar 7, program yang diukur merupakan program untuk mengirim, menerima dan menyimpan data pada suatu variable. Sedangkan baris terakhir pada coding akan membaca nilai waktu dari program yang diukur. Kemudian nilai tersebut disimpan kedalam variable "WaktuTotal".

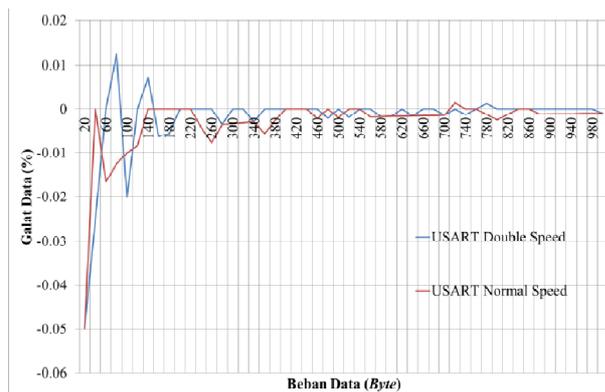
4. Temuan Penelitian

Berdasarkan penelitian yang telah dilakukan, ditemukan nilai waktu yang dibutuhkan komunikasi serial USART untuk transmisi data berdasarkan beban data dan jenis transmisi. Temuan tersebut diperoleh berdasarkan konfigurasi komunikasi serial USART pada mikrokontroler dan MATLAB. Sehingga diperoleh nilai waktu yang dibutuhkan komunikasi serial USART untuk mengirim dan menerima data. Nilai tersebut dapat dilihat pada Gambar 8.



Gambar 8. Perolehan Waktu Komunikasi USART

Pada gambar 8 tersebut dapat diketahui bahwa semakin besar beban data yang ditransmisikan, semakin lama waktu yang dibutuhkan USART untuk melakukan transmisi data tersebut. Ketika jenis transmisi serial USART *Asynchronous Double Speed* dan beban data 20 *byte*, waktu yang dibutuhkan untuk transmisi data adalah selama 0,0117192 detik. Namun ketika beban data mencapai 1000 *byte*, maka waktu yang dibutuhkan transmisi selama 0,032377 detik. Nilai tersebut tidak berbeda jauh dengan jenis transmisi serial USART *Asynchronous Normal Speed*. Hal tersebut dapat disebabkan oleh berbagai faktor, salah satunya yaitu pada perangkat USB ke pengubah serial. Ketika beban data 20 *byte* waktu dibutuhkan transmisi selama 0,0127109 detik. Namun pada beban data 1000 *byte*, waktu transmisi mencapai 0,031394 detik. Selain itu, ditemukan galat pada data yang telah dikirim maupun diterima. Galat dapat ditemukan dengan cara jumlah data yang diterima dibagi oleh jumlah data yang dikirim pada perangkat lunak MATLAB. Hasil tersebut kemudian dikurang oleh konstanta satu dan setelah itu dikali 100%. Galat dikatakan baik ketika bernilai 0%. Galat tersebut dapat dilihat pada Gambar 9.



Gambar 9. Perolehan Galat Komunikasi USART

Berdasarkan gambar tersebut ketika beban data sedikit, terlihat bahwa galat yang ditimbulkan berbeda. Nilai galat akan diatas 0% ketika beban data 80 *byte* dan 140 *byte* untuk jenis transmisi USART *Double Speed*. Namun ketika data berada 60 sampai 120 nilai galat dibawah 0% untuk jenis transmisi USART *Normal Speed*.

Berdasarkan hasil temuan yang telah dipaparkan, diperoleh bahwa perbandingan USART *Double Speed* dan USART *Normal Speed* pada konfigurasi kecepatan maksimal tidak terlalu signifikan. Waktu yang diperoleh ketika jenis transmisi serial *Asynchronous Double Speed* yang digunakan dan beban data sebesar 20 *byte* adalah selama 0,0117192 detik. Ketika beban data sebesar 1000 *byte*, waktu yang dibutuhkan transmisi adalah selama 0,031719 detik. Namun ketika jenis transmisi serial *Asynchronous Normal Speed* yang digunakan dan beban

data sebesar 20 *byte*, waktu yang dibutuhkan untuk transmisi adalah selama 0,012719 detik. Ketika beban data mencapai 1000 *byte*, waktu yang dibutuhkan untuk transmisi adalah selama 0,031394 detik. Sehingga, waktu yang dibutuhkan USART untuk transmisi data meningkat seiring dengan meningkatnya beban data. Hal tersebut berbeda dengan galat yang ditimbulkan. Nilai galat akan besar jika data yang ditransmisikan oleh USART sedikit. Namun, nilai galat akan mendekati nol jika data yang ditransmisikan besar. Galat bernilai -0.05% ketika beban data 20 *byte*. Namun ketika beban data sebesar 1000 *byte*, galat mendekati 0%.

5. Kesimpulan

Berdasarkan hasil temuan yang telah dipaparkan, diperoleh bahwa perbandingan USART *Double Speed* dan USART *Normal Speed* pada konfigurasi kecepatan maksimal tidak terlalu signifikan. Waktu yang diperoleh ketika jenis transmisi serial *Asynchronous Double Speed* yang digunakan dan beban data sebesar 20 *byte* adalah selama 0,0117192 detik. Ketika beban data sebesar 1000 *byte*, waktu yang dibutuhkan transmisi adalah selama 0,031719 detik. Namun ketika jenis transmisi serial *Asynchronous Normal Speed* yang digunakan dan beban data sebesar 20 *byte*, waktu yang dibutuhkan untuk transmisi adalah selama 0,012719 detik. Ketika beban data mencapai 1000 *byte*, waktu yang dibutuhkan untuk transmisi adalah selama 0,031394 detik. Sehingga, waktu yang dibutuhkan USART untuk transmisi data meningkat seiring dengan meningkatnya beban data. Hal tersebut berbeda dengan galat yang ditimbulkan. Nilai galat akan besar jika data yang ditransmisikan oleh USART sedikit. Namun, nilai galat akan mendekati nol jika data yang ditransmisikan besar. Galat bernilai -0.05% ketika beban data 20 *byte*. Namun ketika beban data sebesar 1000 *byte*, galat mendekati 0%.

References

- [1] M. Jokitulppo, "Arduino-controlled Robot," Thesis, Jank University of Applied Sciences, Kotkansalo, Jouko, 2015.
- [2] P. A. P. T. Ms.Neha R. Laddha, "Implementation of serial communication using UART with configurable baud rate," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 1, no. 4, pp. 263–268, 2013.
- [3] U. Nanda and S. K. Pattnaik, "Universal Asynchronous Receiver and Transmitter (UART)," *ICACCS 2016 - 3rd Int. Conf. Adv. Comput. Commun. Syst. Bringing to Table, Futur. Technol. from Around Globe*, 2016, doi: 10.1109/ICACCS.2016.7586376.
- [4] A. Corporation, "8-bit AVR Microcontroller with 4/8/16/32K Bytes In-System Programmable Flash Datasheet." San Jose, California, United States, pp. 1–23, 2009.
- [5] A. P. Godse and D. A. Godse, *Microcontrollers & microprocessors*, vol. 50, no. 1. India: Technial Publication Pune, 2002.
- [6] P. A. P. T. Ms.Neha R. Laddha, "A Review on Serial Communication by UART," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 3, no. 1, pp. 366–369, 2013.
- [7] S. Evan, Sarotaga, and Calif, "Universal Protocol Programmable Communication Interface," U.S. Patent 5371736, 1994.