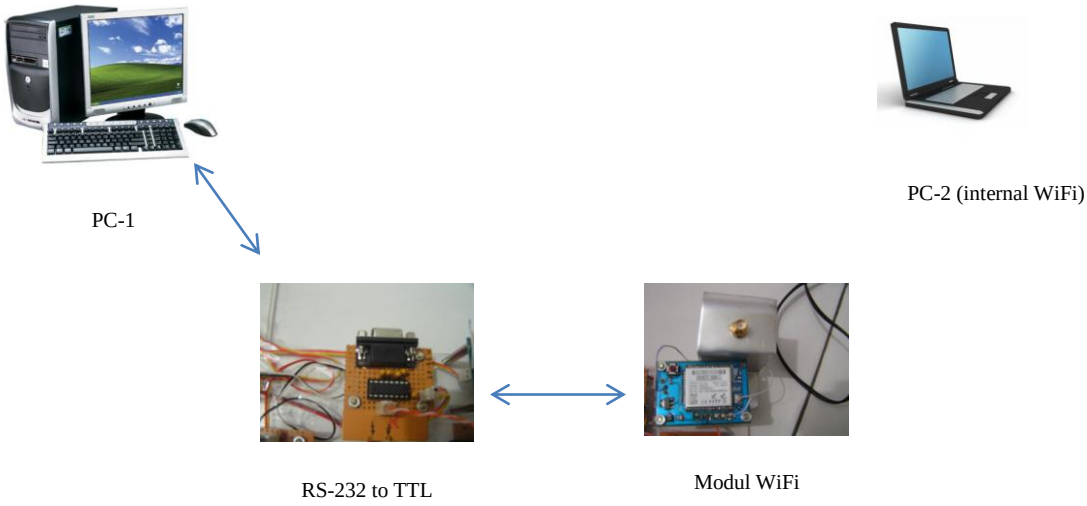
BAB IV
HASIL DAN PEMBAHASAN

4.1 Pengujian

Pengujian dilakukan terhadap semua blok yang ada pada sistem secara mandiri, kemudian dilakukan pengujian terhadap sistem secara keseluruhan.

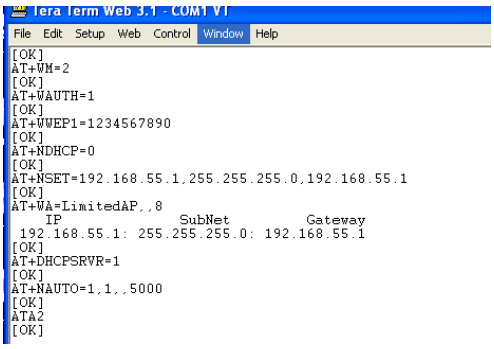
4.1.1 Modul WiFi

Pengujian Modul WiFi dilakukan dengan cara melakukan pengaturan dan uji kinerja dari perangkat tersebut. Gambar 4.1 memperlihatkan ilustrasi skema pengujian yang dilakukan.

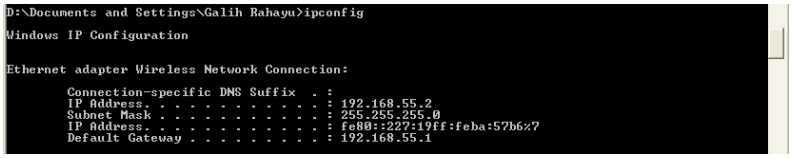


Gambar 4.1. Skema pengujian Modul WiFi

Pengaturan modul WiFi dilakukan di PC-1 melalui *Terminal emulator* pada Sistem operasi Windows XP dengan menggunakan AT *command*. Kemudian PC-2 memeriksa konfigurasi jaringan yang didapat dari Modul WiFi.



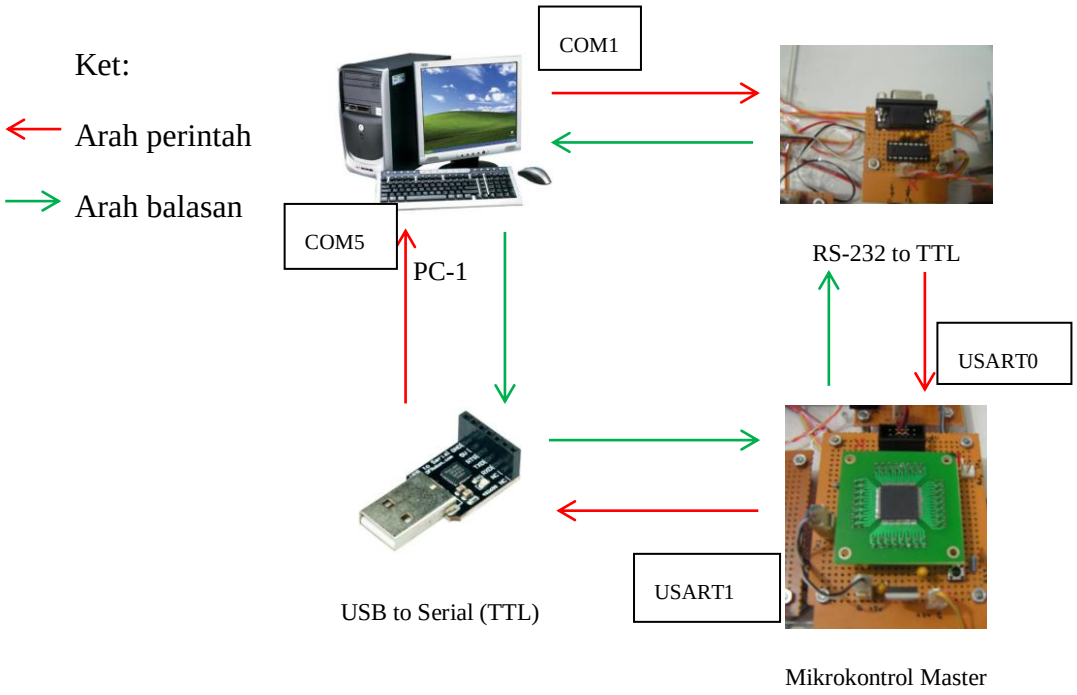
Gambar 4.2. Konfigurasi Modul Wifi dengan AT *Command*



Gambar 4.3. IP *address* pada PC-2 yang didapat dari Modul Wifi

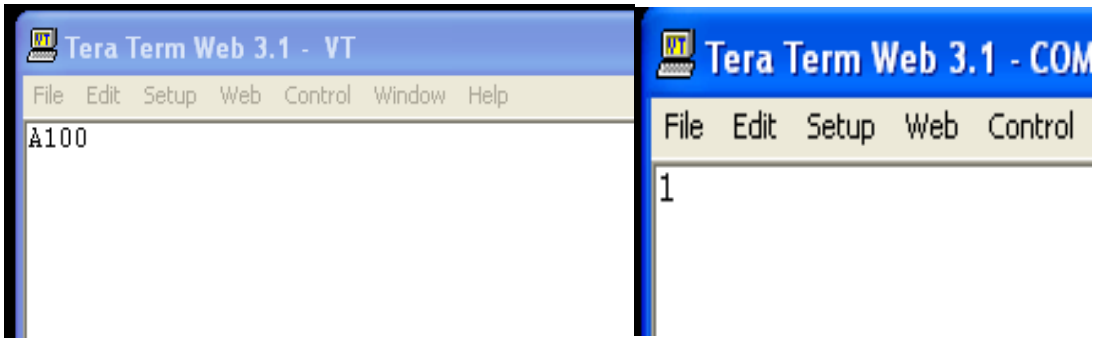
4.1.2 Mikrokontroler Master

Mikrokontroler *Master* diuji dengan cara mengirimkan perintah pada kedua USART.



Gambar 4.4. Skema pengujian Mikrokontroler *Master*

Berdasarkan konfigurasi pada Gambar 4.4 maka didapat keluaran pada *Terminal Emulator* seperti yang diperlihatkan Gambar 4.5.

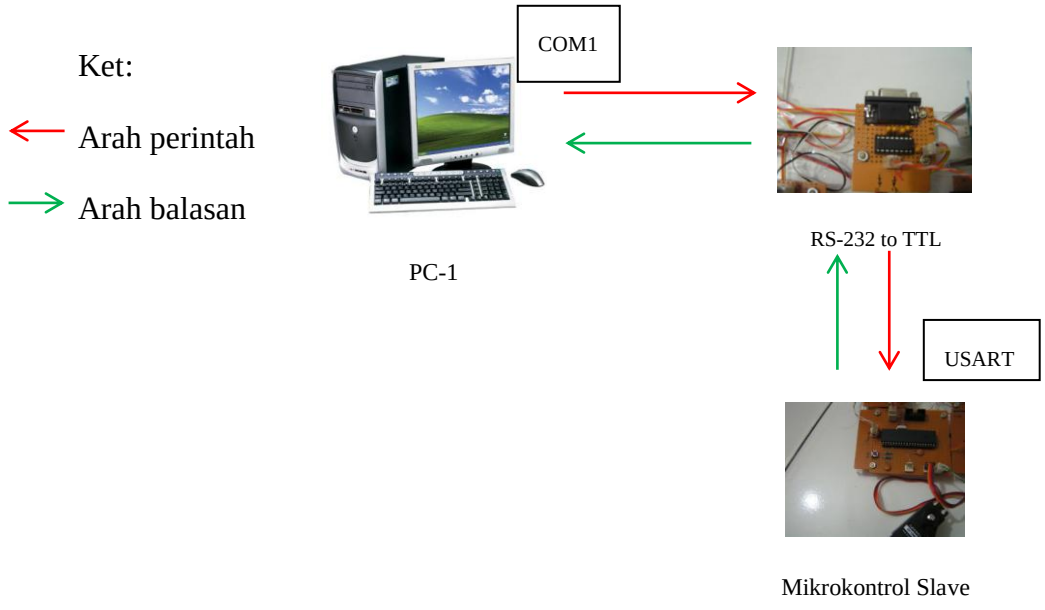


Keluaran pada terminal (COM5) Keluaran pada terminal (COM5)

Gambar 4.5. Keluaran pada *Terminal Emulator*

4.1.3 Mikrokontroler *Slave*

Mikrokontroler *Slave* diuji dengan cara mengirimkan perintah untuk mengatur posisi motor *servo* dan membaca nilai sensor suhu.



Gambar 4.6. Skema pengujian

4.1.3.1 Pengujian Sensor Suhu LM35

Untuk menguji kinerja ADC Mikrokontroler *Slave* dan sensor suhu maka dilakukan pengukuran suhu menggunakan termometer air raksa, kemudian hasil dibandingkan dengan data dari ADC.

ADC pada Mikrokontroler *Slave* adalah mode 10-bit *Single Ended Coversion* dengan $V_{ref} = 2.56V$ maka nilai resolusi dari ADC adalah:

$$Resolusi = \frac{2.56V}{2^{10}} = \frac{2.56V}{1024} = 2.50 \text{ mV}$$

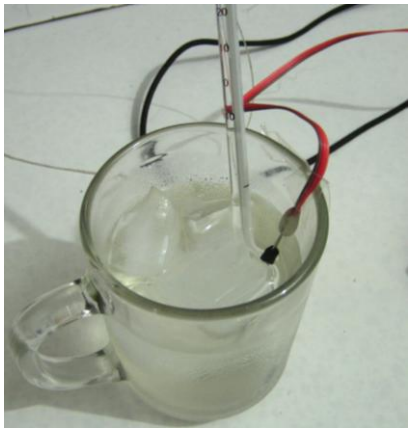
Sedangkan keluaran dari IC LM35 yang menjadi masukan (V_{in}) bagi ADC adalah 10 mV per derajat Celcius, maka nilai ADC bisa dihitung yakni:

Misal, Suhu = 50°C

$$V_{in} = V_{outLM35} = 50°\text{C} \times 10mV = 500mV$$

$$ADC = \frac{V_{in}}{Resolusi} = \frac{500\ mV}{2.50\ mV} = 200 = 0011001000_2$$

Untuk merubah ke nilai suhu maka nilai ADC dibagi empat, namun karena hasil dalam mode *Left Adjust* maka nilai pada register ADCH $= 00110010_2$ dan ADCL $= 00000000_2$, sehingga untuk mendapatkan nilai suhu hanya perlu membaca nilai ADCH yakni 50.



Gambar 4.7. Sensor mengukur suhu air dingin

Tabel 4.1. Perbandingan Suhu Termometer dengan Data ADC

No	Suhu	Data ADC
1	6℃	7℃
2	9℃	12℃
3	14℃	12℃
4	14℃	13℃
5	23℃	23℃
6	26℃	26℃
7	26℃	29℃
8	45℃	43℃
9	49℃	49℃
10	51℃	52℃
11	51℃	55℃

- Pengukuran Suhu menggunakan Termmometer Air Raksa.

Pada Tabel 4.1 data ADC merupakan hasil bacaan ADC ATmega8535 yang didapat dari V_{out}LM35 ,sedangkan data Suhu didapatkan dari termometer air raksa yang menjadi nilai yang dijadikan acuan nilai benar. Dari Tabel 4.1 bisa disimpulkan bahwa hasil bacaan sensor suhu sudah berhasil, adapun hasil bacaan yang tidak sesuai dengan nilai benar bisa diperbaiki misalnya dengan penggunaan sensor suhu yang lebih akurat.

4.1.3.2 Pengujian Motor Servo

Dalam pengujian digunakan dua buah motor *servo* yang berbeda yakni GWS S03N STD dan DFRobot DF15MG. Karena kurangnya dokumentasi dari kedua motor *servo* tersebut maka penulis melakukan *trial and error* dalam menentukan PWM yang tepat untuk posisi minimal (0°) dan maksimal (90°), pemilihan sudut maksimal 90° dikarenakan kebanyakan keran ukuran 0.5 inch mempunyai sudut bukaan sebesar 90°. Pada umumnya motor *servo* bekerja pada PWM dengan frekuensi 50 Hz.

Dengan konfigurasi PWM pada Mikrokontroler *Slave* dalam 16-bit *Timer/Counter mode Fast* PWM Prescaler 64 dan F_CPU 14.7456 MHz.Untuk menghasilkan frekuensi 50 Hz, maka:

$$f_{pwm} = 50Hz = \left(\frac{F_CPU}{PRESCALER \times (1 + TOP)}\right) = \left(\frac{11745600\ Hz}{64 \times (1 + TOP)}\right)$$

$$64 \times (1 + TOP) = \frac{11745600\ Hz}{50\ Hz} = 294912$$

$$TOP = \frac{294912}{64} - 1 = 4607$$

Dimana TOP adalah nilai untuk register ICR1,

Untuk mencari PWM yang tepat maka proses *trial and error* dilakukan dengan cara merubah isi register OCR1A hasil dari proses tersebut ditampilkan pada Tabel 4.2.

Tabel 4.2. Nilai OCR1A untuk motor *servo*

No	Merk Motor	Nilai OCR1A	
		0°	90°
1	GWS S03N STD	130	322
2	DFRobot DF15MG	190	412

Maka untuk mencari nilai frekuensi dari nilai OCR1A Motor servo No. 1 pada Tabel 4.2 ,adalah:

$$f_{pwm0derajat} = \left(\frac{130}{4607}\right) x\ 50\ Hz = 1.41\ Hz$$

$$f_{pwm90derajat} = \left(\frac{322}{4607}\right) x\ 50\ Hz = 3.45\ Hz$$

Dari tabel 4.2 bisa dicari nilai untuk menggerakan motor *servo* perderajat yakni:

$$A_1 = \frac{NS_{MAX} - NS_{MIN}}{S_{MAX}} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (4.1)$$

$$A_2 = NS_{MIN} + (A_1 * Setpoint) \dots\dots\dots\dots\dots\dots (4.2)$$

Keterangan:

$A_1 = Pengali\ nilai\ Setpoint$

$A_2 = Nilai\ Register\ OCR1A$

$NS_{MAX} = Nilai\ OCR1A\ untuk\ Sudut\ Maksimum\ (90°)$

$NS_{MIN} = Nilai\ OCR1A\ untuk\ Sudut\ Minimum\ (0°)$

$S_{MAX} = Nilai\ Sudut\ Maksimum\ (90\ untuk\ 90°)$

$Setpoint = Nilai\ Sudut\ dari\ pengguna$

Berdasarkan persamaan 4.1 dan 4.2, maka untuk Motor *Servo* DFRobot DF15MG:

$$A_1 = \frac{412 - 190}{90} = 2.47$$

Dan untuk GWS S03N STD:

$$A_1 = \frac{322 - 130}{90} = 2.14$$

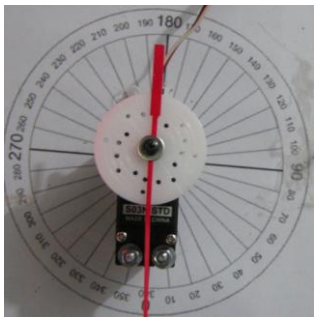
Dari hasil percobaan didapatkan data pergerakan motor *servo* GWS S03N STD yang ditampilkan pada Tabel 4.3.

Tabel 4.3. Pergerakan motor *servo* dengan s*etpoint* dari aplikasi Android

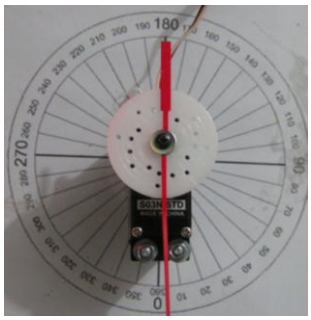
No	*Setpoint*	Pergerakan Motor *servo*
1	0°	0°

2	1°	1°
3	10°	10°
4	20°	20°
5	30°	30°
6	40°	40°
7	50°	50°
8	60°	60°
9	70°	70°
10	80°	80°
11	90°	90°

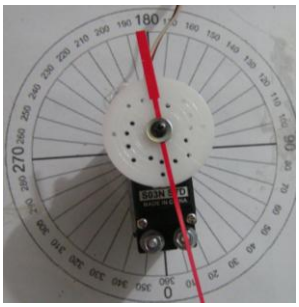
0°



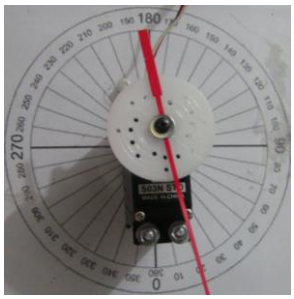
1°



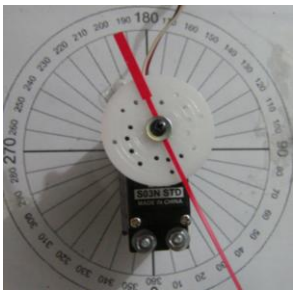
10°



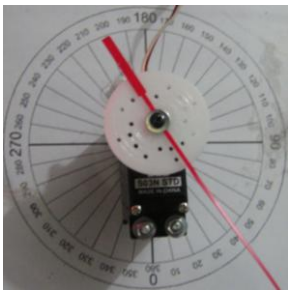
20°



30°



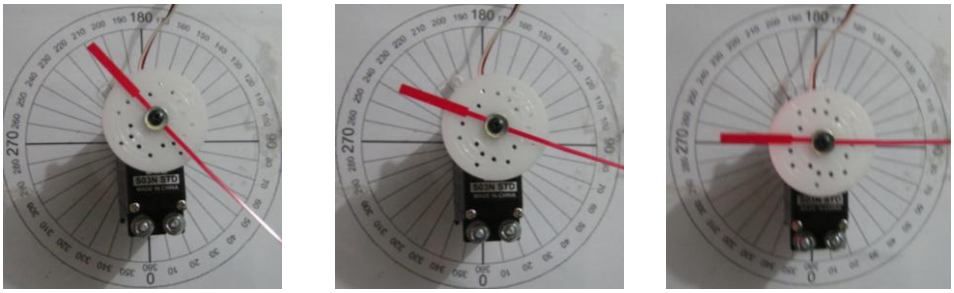
40°



60°

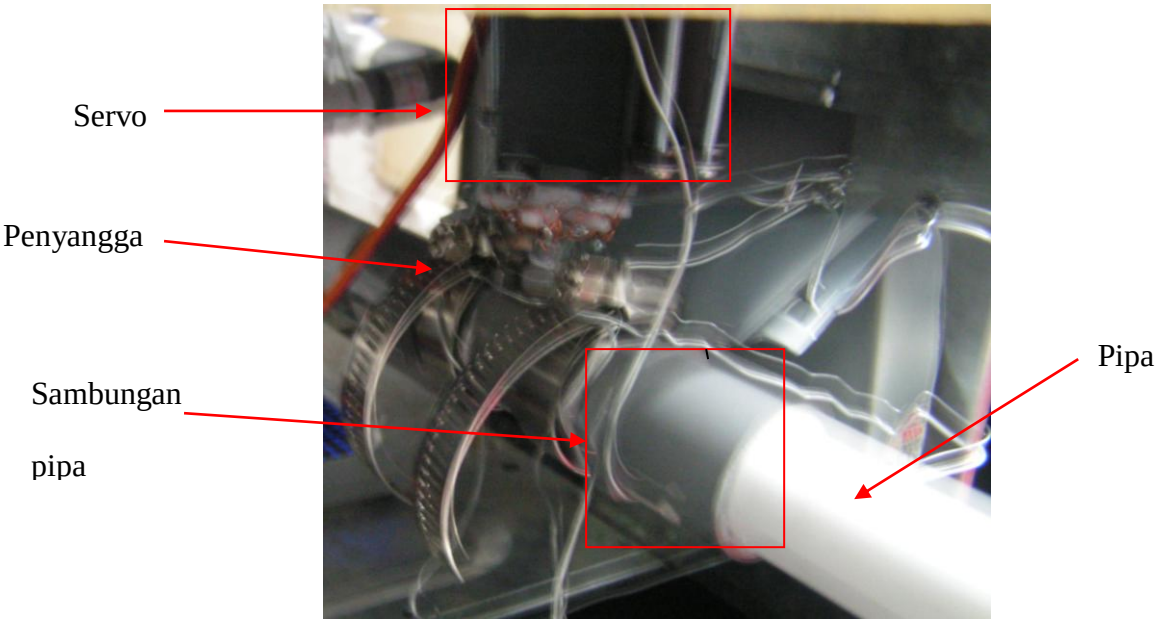
80°

90°



Gambar 4.8. Pergerakan motor *servo* sesuai *setpoint*

4.1.4 Bagian Mekanik



Gambar 4.9. Gambar bagian mekanik

Untuk bagian mekanik tata letak disesuaikan dengan tempat keran. Untuk motor servo DFRobot DF15MG, tidak diperlukan *gear* tambahan, karena torsi dari motor *servo* sudah cukup untuk memutar keran ukuran 0.5 inch. Sedangkan untuk motor *servo* GWS S03N STD, *servo* tersebut tidak kuat untuk memutar keran, untuk itu diperlukan tambahan gear .

Dengan referensi nilai torsi dari motor *servo* DFRobot DF15MG, maka diperlukan *gear* sebagai berikut:

Torsi DFRobot DF15MG = 10 kg.cm

Torsi GWS S03N STD = 2.40 kg.cm

$$\frac{\text{Torsi DFRobot DF15MG}}{\text{Torsi GWS S03N STD}} = \frac{10\ \text{kg. cm}}{2.40\ \text{kg. cm}} = 4.17$$

Untuk menghitung jumlah gigi hasil dibulatkan jadi 4. Dari perbandingan kedua motor servo, servo GWS S03N STD harus mempunyai torsi 4x dari torsi yang ada atau dengan kata lain *gear ratio* adalah 4. Maka berdasarkan persamaan 2.1:

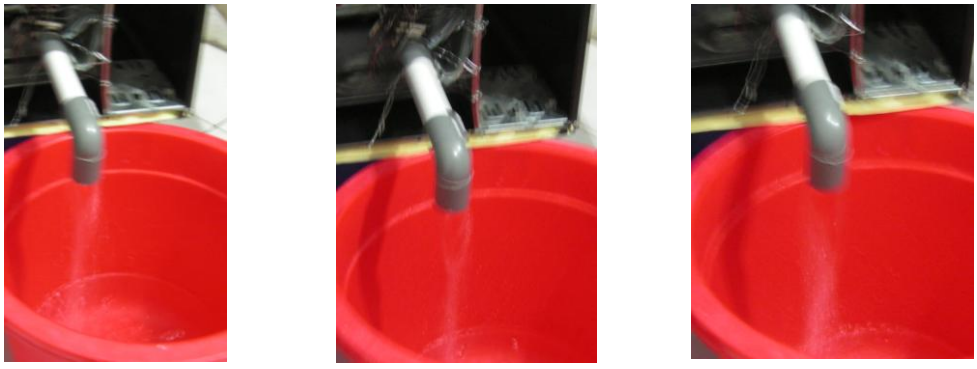
Misal jumlah $Z_1 = 10$

$$u = \frac{Z_2}{Z_1}$$

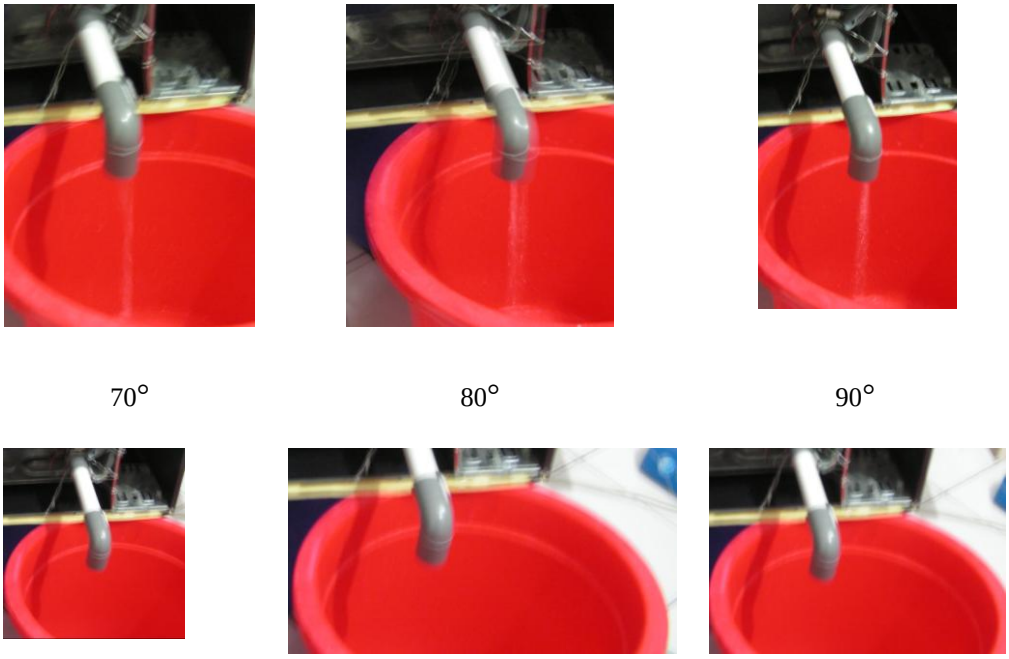
$$Z_2 = u \times Z_1 = 4 \times 10 = 40$$

Gambar 4.10 memperlihatkan bukaan keran saat mengalirkan air. Berdasarkan ujicoba maka untuk keran, posisi menutup penuh ada pada 90° dan terbuka penuh pada 0°.

0° 10° 20°



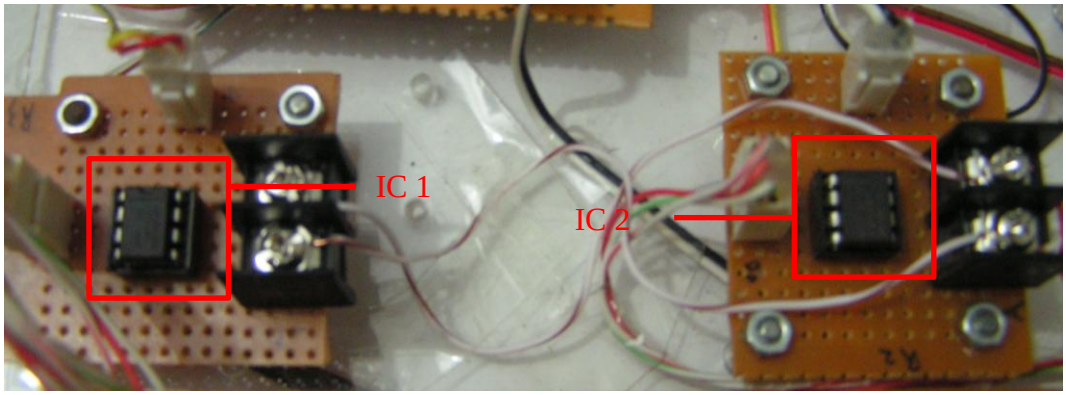
30° 40° 60°



Gambar 4.10. Bukaan keran

4.1.5 Komunikasi RS-485

Blok RS-485 diuji dengan cara mengukur level tegangan pada kaki MAX485. Tabel 4.4 memperlihatkan hasil pengukuran.



Gambar 4.11. Blok RS-485

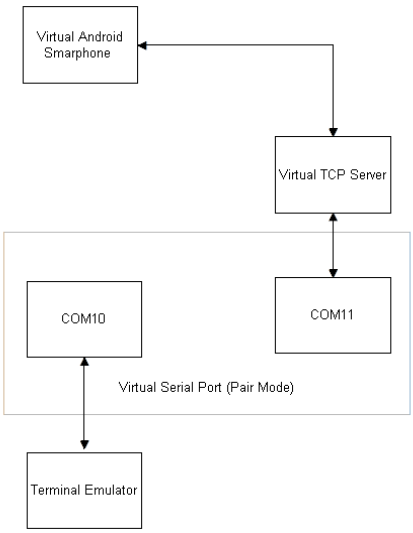
Tabel 4.4. Hasil pengukuran IC MAX485

No.	IC 1			AB	A	B	IC2		
	DE/ $\overline{RE}$	DI	RO				DE/ $\overline{RE}$	DI	RO
1	0V	0.05V	0V	-3.24	0.92V	4.38V	4.90V	0V	0.05V
2	0V	0.02V	4.87V	0.84V	0.92V	0.06V	4.90V	4.90V	0.03V
3	4.90V	0V	0.02V	-4.02V	0.38V	4.41V	0V	0.01V	0V
4	4.90V	4.90V	0.01	4.32V	4.44V	0.07V	0V	0V	4.04V

- Pengukuran menggunakan DT830B Digital Multimeter.

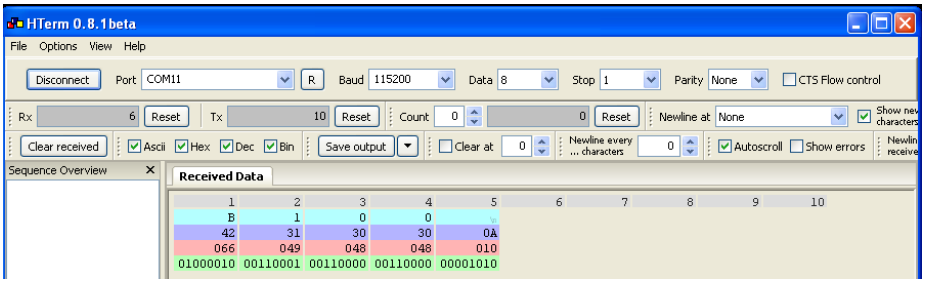
4.1.6 Pengujian Perangkat Lunak pada Android *Smartphone*

Pengujian dilakukan dengan cara fasilitas *Emulator* pada Android SDK *Tools.* Pengujian dilakukan untuk melihat format data yang dikirimkan oleh perangkat Android ke alat pengendali.



Gambar 4.12. Skema Pengujian Perangkat Lunak Android

Gambar 4.12 memperlihatkan konfigurasi dari perangkat lunak pada PC yang digunakan untuk menangkap format data dari perangkat lunak android. Pada Gambar 4.12 *Virtual Serial port* COM10 dan COM11 dikonfigurasi dalam *Pair mode* dalam mode ini maka masukan/keluaran dari COM10 akan ditampilkan juga pada COM11.



Gambar 4.13. Format data yang diterima pada *Terminal Emulator*

Gambar 4.13 memperlihatkan data yang dikirimkan oleh perangkat lunak pada Android *smartphone*. Data merupakan String pada kode sumber yang kemudian ditambahkan karakter *Newline*(‘\n’) oleh fasilitas Socket pada Java.