

BAB IV

HASIL DAN PEMBAHASAN

Pada Bab ini dapat diketahui apakah seluruh sistem bekerja sesuai dengan perancangan. Pengujian perangkat keras dan perangkat lunak dilakukan pada masing-masing komponen dalam sistem. Pengujian yang dilakukan adalah sebagai berikut.

1. Pengujian sensor *Tipping Bucket*,
2. Pengujian sensor Kecepatan Air
3. Pengujian sensor ultrasonik HC-SR04
4. Pengujian modul nRF 24L01+
5. Pengujian keseluruhan sistem
6. Pengujian konsumsi arus

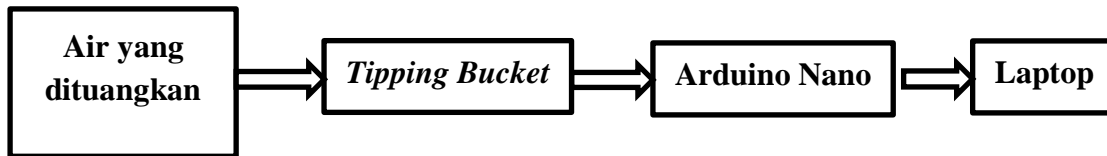
4.1 Pengujian Sensor *Tipping Bucket*

Pengujian sensor ini bertujuan untuk membuktikan apakah sensor *Tipping Bucket* yang telah dirancang dapat mendeteksi kondisi curah hujan yang dideteksi dengan ketelitian 0.5 mm. Adapun alat yang digunakan selama pengujian sensor meliputi

1. Sensor *Tipping Bucket*
2. Laptop
3. Mikrokontroler Arduino Nano
4. Serial Monitor
5. Air \pm 200 ml

Pada pengujian sensor *Tipping Bucket*, metode yang digunakan yaitu dengan mensimulasikan air hujan menggunakan 200 ml air, yang akan menghasilkan 10 kali jungkitan pada sensor. Air yang terbuang pada setiap jungkitan akan dihitung volumenya.

Diagram blok pengujian sensor *Tipping Bucket*.



Gambar 4.1 Blok diagram pengujian sensor *tipping bucket*

Prosedur pengujian sensor *Tipping Bucket* adalah sebagai berikut.

1. Rangkai rangkaian pengujian sensor *Tipping Bucket* sesuai dengan blok diagram seperti Gambar 4.1.
2. Tuangkan air melalui corong *Tipping Bucket*.
3. Baca nilai keluaran sensor melalui *serial monitor* yang ada pada *software Arduino IDE*.
4. Hitung volume air yang keluar melalui buangan air pada *Tipping Bucket*.
5. Hitung nilai *error* yang dihasilkan oleh sensor *Tipping Bucket*.

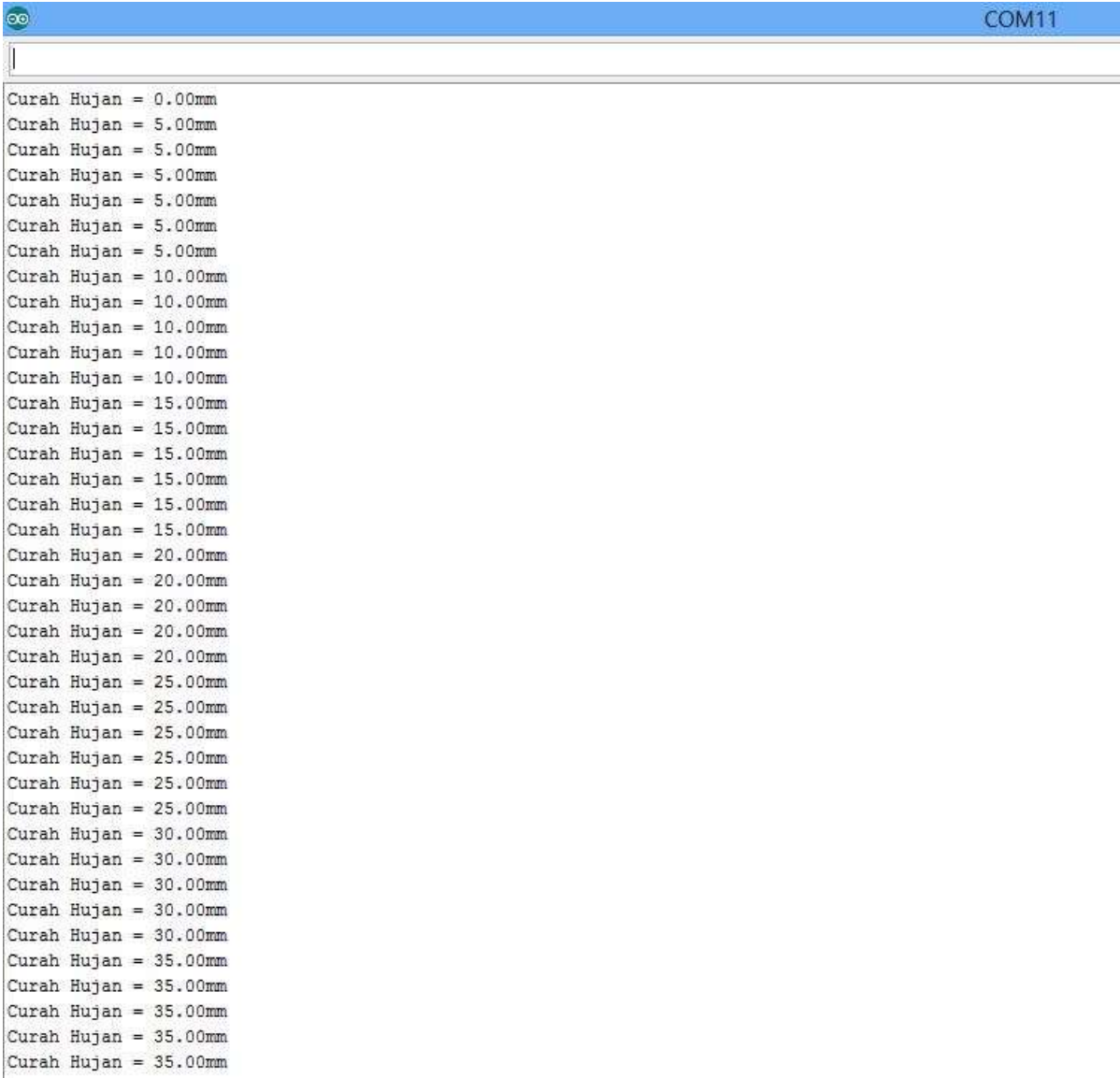
Dari pengujian ini dapat diketahui pembacaan sensor dan *error* sensor *Tipping Bucket* terhadap kondisi air hujan yang disimulasikan dengan 200 ml air. Pengujian dilakukan dengan melakukan penghitungan volume air yang terbuang pada setiap jungkitan *Tipping Bucket* dan pembacaan pada *serial monitor software Arduino IDE*. Hasil pengujian akan ditampilkan pada Tabel 4.1.

Pada tabel 1 dapat dilihat bahwa hasil pengujian alat ukur curah hujan menunjukkan nilai *error* sebesar 10 % pada jungkitan pertama, dan ketiga, dan 5 % pada jungkitan ke 4, 6, dan 7. Pengujian ini menghasilkan rata – rata *error* alat ini sebesar 5 %, hal ini disebabkan karena kalibrasi alat yang masih manual. Namun nilai *error* yang terjadi tidak terlalu berpengaruh pada kinerja sistem, dikarenakan resolusi sensor yang ingin dicapai adalah 0,5 mm.

Tabel 4.1 Hasil kalibrasi alat ukur curah hujan

Jungkitan	Volume Air Perhitungan	Volume Air Buangan	<i>Error</i> (%)
1	20 ml	18 ml	10%
2	20 ml	20 ml	0%
3	20 ml	18 ml	10%
4	20 ml	19 ml	5%
5	20 ml	20 ml	0%
6	20 ml	19 ml	5%
7	20 ml	19 ml	5%
Rata – rata <i>Error</i>			5%

Tabel diatas menunjukkan bahwa sensor *Tipping Bucket* dapat mengukur curah hujan paling sedikit sebesar 0.5 mm dengan rata –rata *error* sebesar 5%. Dimana setiap 20 ml air yang dituangkan mewakili 0.5 mm curah hujan. Setiap detik bila terjadi jungkitan maka akan terjadi perubahan sinyal elektrik berupa pulsa, dimana tiap pulsa mewakili volume air 50 ml. Selain pada alat juga dapat dilihat dengan munculnya data baru pada *serial monitor Arduino IDE*,



```

COM11
Curah Hujan = 0.00mm
Curah Hujan = 5.00mm
Curah Hujan = 5.00mm
Curah Hujan = 5.00mm
Curah Hujan = 5.00mm
Curah Hujan = 5.00mm
Curah Hujan = 5.00mm
Curah Hujan = 10.00mm
Curah Hujan = 10.00mm
Curah Hujan = 10.00mm
Curah Hujan = 10.00mm
Curah Hujan = 10.00mm
Curah Hujan = 15.00mm
Curah Hujan = 15.00mm
Curah Hujan = 15.00mm
Curah Hujan = 15.00mm
Curah Hujan = 15.00mm
Curah Hujan = 15.00mm
Curah Hujan = 20.00mm
Curah Hujan = 20.00mm
Curah Hujan = 20.00mm
Curah Hujan = 20.00mm
Curah Hujan = 20.00mm
Curah Hujan = 25.00mm
Curah Hujan = 25.00mm
Curah Hujan = 25.00mm
Curah Hujan = 25.00mm
Curah Hujan = 25.00mm
Curah Hujan = 25.00mm
Curah Hujan = 30.00mm
Curah Hujan = 30.00mm
Curah Hujan = 30.00mm
Curah Hujan = 30.00mm
Curah Hujan = 35.00mm
Curah Hujan = 35.00mm
Curah Hujan = 35.00mm
Curah Hujan = 35.00mm

```

Gambar 4. 2 Hasil pengukuran curah hujan oleh sensor *tipping bucket*

Pada Gambar 4.2 menunjukkan hasil pengukuran sensor ditampilkan pada serial monitor *software* Arduino. Hal ini berarti pengukuran curah hujan yang dilakukan secara mekanik oleh *Tipping Bucket* sudah dapat diubah ke dalam bentuk sinyal elektrik dengan memanfaatkan sensor *Hall Effect* A3144 yang kemudian diolah oleh mikrokontroler Arduino Nano sehingga bisa ditampilkan hasil pengukuran curah hujan lewat *serial monitor*.

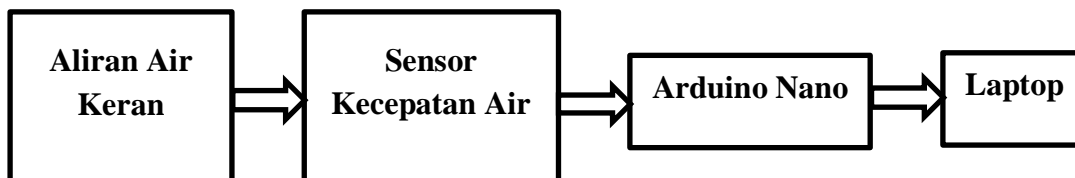
4.2 Pengujian Sensor Kecepatan Air

Tujuan dari pengujian Sensor Kecepatan Air adalah untuk mengetahui apakah sensor dapat membaca kecepatan suatu aliran air dan menampilkan hasilnya melalui *serial monitor* pada *Arduino IDE*.

Adapun alat yang digunakan selama pengujian sensor kecepatan air meliputi:

1. Sensor Kecepatan Air
2. Mikrokontroler Arduino Nano
3. Laptop
4. Serial Monitor

Prosedur pengujian sensor kecepatan air yakni dengan menggerakkan baling – baling pada sensor menggunakan aliran air. Putaran tersebut akan memutar magnet yang akan memicu sensor *Hall Effect A1344* yang kemudian akan mengkonversi putaran tersebut menjadi kecepatan dan dibaca oleh mikrokontroler Arduino.



Gambar 4.3. Diagram blok pengujian sensor kecepatan air

Prosedur pengujian sensor kecepatan air sebagai berikut.

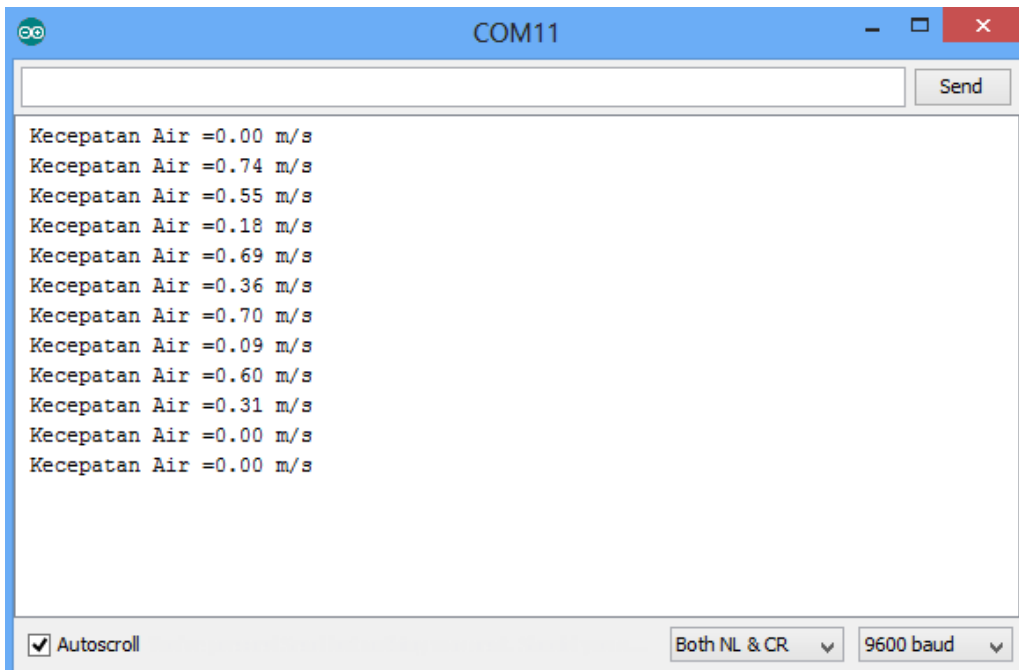
1. Rangkai rangkaian
2. Dekatkan baling – baling pada sensor dengan aliran air
3. Baca nilai keluaran sensor melalui *serial monitor* yang ada pada *software Arduino IDE*, bandingkan dengan alat ukur standar.

Pengujian ini dilakukan untuk mengetahui hasil pembacaan sensor *Hall Effect A3144* terhadap putaran baling – baling. Putaran pada baling – baling dipengaruhi oleh kecepatan aliran air yang menggerakkan baling – baling tersebut.

Tabel 4.2 Hasil Pengujian Alat Ukur Kecepatan Air

Kecepatan Terukur	Kecepatan Pada Alat Ukur Standar	Error
0.11	0.11	0%
0.18	0.18	0%
0.24	0.25	4%
0.36	0.37	2.7%
0.74	0.74	0%

Hasil pengukuran menunjukkan *error* yang cenderung kecil, yaitu dibawah 4%. Sehingga alat dapat dikatakan dapat menghasilkan nilai kecepatan yang sudah mendekati nilai kecepatan pada alat ukur standar.



Gambar 4.4 Nilai kecepatan air

Gambar 4.4 menunjukkan bahwa putaran baling – baling yang digerakkan oleh air berhasil dikonversi menjadi kecepatan oleh Arduino. Ketika keran mati dan baling – baling tidak berputar, maka nilai kecepatan air adalah 0,00 m/s, sedangkan ketika keran perlahan lahan dinyalakan, nilai kecepatan air meningkat hingga 0,70 m/s. Hasil pengukuran yang keluar membuktikan bahwa program telah berhasil bekerja sesuai keinginan.

4.3 Pengujian Sensor Ketinggian Air

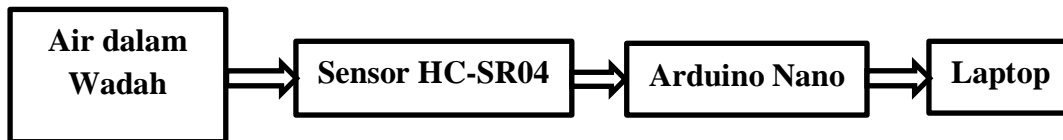
Pengujian sensor ultrasonik ketinggian air dilakukan untuk membuktikan apakah sensor dapat mendeteksi ketinggian air. Melalui pengujian ini pula dapat diketahui besarnya *error* dari hasil pengukuran sensor dengan membandingkan hasil pengukuran dengan alat ukur standar. Alat ukur standar yang digunakan dalam pengujian ini adalah meteran. Adapun alat yang digunakan selama pengujian sensor kecepatan air meliputi:

1. Sensor Ultrasonik HC-SR04
2. Mikrokontroler Arduino Nano
3. Laptop
4. Serial Monitor

5. Meteran

6. Air

Prosedur pengujian sensor ketinggian air terhadap permukaan air. Pengujian terhadap permukaan air dilakukan dengan cara menghadapkan sensor ultrasonik ke permukaan air dalam wadah dengan jarak tertentu. Jarak antara permukaan air dan sensor kemudian diukur menggunakan mistar lalu dicatat hasilnya mulai 2 cm sampai 200 cm.



Gambar 4.5 Diagram blok pengujian sensor ketinggian air

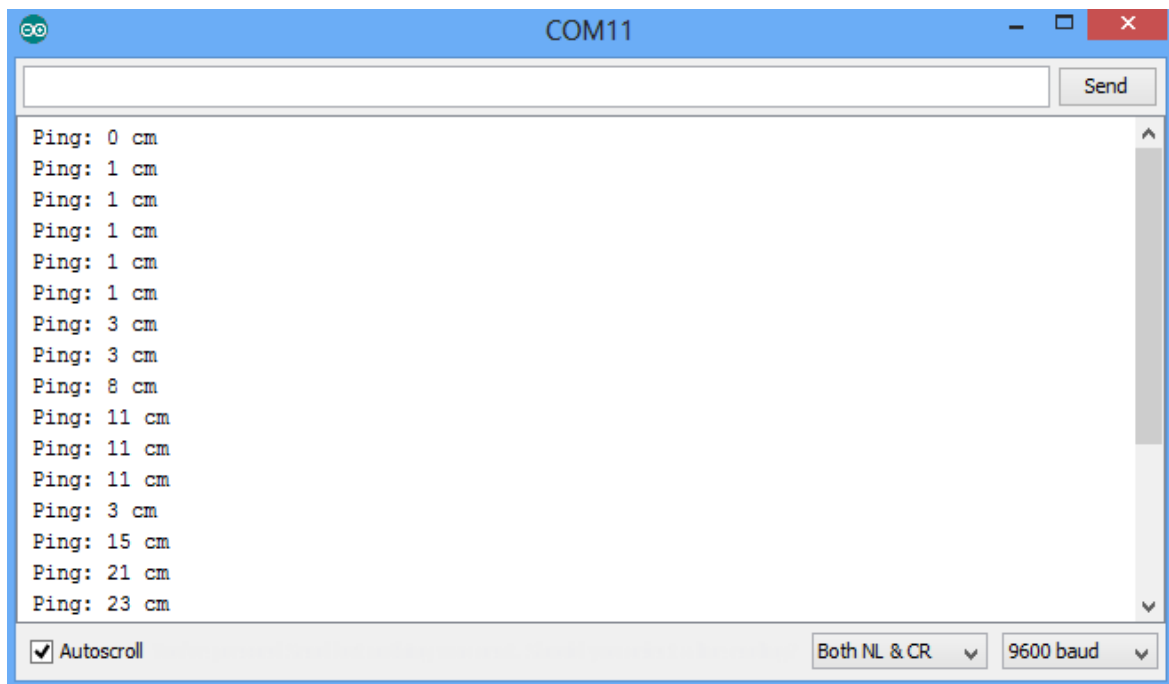
Prosedur pengujian sensor ultrasonik dengan mikrokontroler adalah sebagai berikut:

1. Rangkai rangkaian pengujian sensor seperti blok diagram pada Gambar 4.3
2. Hadapkan sensor ultrasonik HC-SR04 pada permukaan air dengan jarak 200 cm
3. Baca nilai keluaran sensor melalui *serial monitor* yang ada pada *software* Arduino *IDE*.
4. Bandingkan dengan jarak yang terukur menggunakan alat ukur.
5. Ubah jarak sensor terhadap air.
6. Baca nilai keluaran sensor melalui *serial monitor* yang ada pada *software* Arduino *IDE*.
7. Bandingkan dengan jarak yang terukur menggunakan alat ukur.
8. Ulangi langkah 6 dan 7 hingga mencapai jarak 2 cm.

Tabel 4.3 Hasil Pengujian Sensor Ultrasonik HC-SR04 Terhadap Permukaan Air

Jarak Sebenarnya (cm)	Hasil Pengukuran (cm)			<i>Error</i>
	Ke-1	Ke-2	Ke-3	
200	199	199	200	0.003%
180	180	180	179	0.003%
160	159	160	160	0.002%
140	140	140	140	0%
120	120	120	120	0%
100	100	99	100	0.003%
80	79	79	79	0%
60	59	60	60	0.003%
40	40	40	40	0%
20	20	20	21	0.003%
2	1	1	1	0.5%

Hasil pengujian sensor ultrasonik menunjukkan bahwa sensor pada jarak 2 cm memiliki *error* 100% dari 3 kali pengukuran, hal ini terjadi karena sensor memiliki batas kemampuan pembacaan minimal jarak yaitu 2 cm. Pada jarak 20, 60, 100, 160, 180, dan 200 memiliki *error* sebesar kurang dari 0.01% dari 3 kali pengukuran. *Error* sebesar 1 cm tidak terlalu berpengaruh terhadap performansi sistem, hal ini karena pada kondisi sebenarnya air mengalami riak sehingga perbedaan data sebesar 1 cm dapat diabaikan.



Gambar 4.6 Hasil pembacaan sensor ultrasonik hc-sr04 pada serial monitor

4.4 Pengujian Modul nRF 24L01+

Pengujian modul nRF 24L01+ bertujuan untuk membuktikan apakah modul nRF 24L01+ dapat mengirimkan data pada jarak tertentu, dan menampilkan data pada *serial monitor* pada *Arduino IDE*.

Adapun alat yang digunakan selama pengujian sensor kecepatan air meliputi :

1. 2 buah modul nRF 24L01+
2. 2 buah Mikrokontroler Arduino Nano
3. Laptop
4. *Power Bank*
5. Serial Monitor

Salah satu modul nRF 24L01+ berperan sebagai *transmitter*, dan modul nRF 24L01+ lainnya berperan sebagai *receiver*. Setelah itu dapat dikirimkan data sambil menjauhkan

kedua modul sampai mencapai jarak maksimal data dapat ditransmisikan. Diagram blok pengujian nRF 24L01+ dapat dilihat pada gambar dibawah.



Gambar 4.7 Diagram blok pengujian nRF 24L01+

Prosedur pengujian nRF 24L01+ dengan Arduino Nano adalah sebagai berikut.

1. Rangkailah rangkaian pengujian nRF 24L01+ sesuai dengan Gambar 4.5
2. Menggunakan salah satu Arduino Nano yang telah diprogram, kirimkan data secara urut kepada Arduino lainnya melalui nRF 24L01+.
3. Lihat data yang diterima oleh Arduino Nano melalui *serial monitor* yang ada pada *software Arduino IDE*
4. Ubah jarak transmisi nRF 24L01+ secara bervariasi yaitu, 90 meter, 140 meter, 260 meter, dan 330 meter.
5. Pada setiap perubahan jarak perhatikan data yang diterima oleh Arduino Nano.

Pengujian ini dilakukan untuk mengetahui seberapa jauh modul nRF 24L01+ dapat mentransmisikan data. Pada pengujian ini variasi jarak ialah mulai 90 m, 140 m, 260 m, dan 330 m.

Tabel 4.4. Hasil Pengujian Modul nRF24L01+

Jarak	Keterangan	
	Tanpa Halangan	Ada Halangan (Tembok)
90 meter	Terjangkau	Terjangkau
100 meter	Terjangkau	Terjangkau
110 meter	Terjangkau	Terjangkau ada <i>error</i>
120 meter	Terjangkau	Terjangkau ada <i>error</i>
130 meter	Terjangkau	Terjangkau ada <i>error</i>
140 meter	Terjangkau	Tidak Terjangkau
260 meter	Terjangkau ada <i>error</i>	Tidak Terjangkau
330 meter	Tidak Terjangkau	Tidak Terjangkau

Dari Tabel 4.3 menunjukkan bahwa modul nRF 24L01+ dapat mentransmisikan data sampai jarak 260 meter, namun tidak sempurna. Yakni terdapat data yang hilang ketika diamati dari Serial Monitor pada *receiver*. Sedangkan untuk jarak 330 meter seluruh data yang dikirim tidak dapat diterima oleh modul nRF 24L01+ *receiver*.

4.5 Pengujian Konsumsi Arus

Pengujian konsumsi arus bertujuan untuk mengetahui arus yang dikonsumsi oleh *node* sensor saat kondisi aktif dan *sleep*. Pengukuran arus yang dikonsumsi dilakukan menggunakan multimeter digital. Pengujian konsumsi arus menggunakan power bank 5V sebagai catu daya.

Tabel 4.5 Hasil Pengujian Konsumsi Arus *Node* Sensor

No	Kondisi	Arus (mA)
		VCC = 5V
1	Aktif	33,4
2	<i>Sleep</i>	17,61

Berdasarkan hasil pengujian didapatkan arus yang dikonsumsi oleh *node* sensor dengan menggunakan power bank 5V ketika kondisi aktif adalah 33,4 mA dan ketika kondisi *sleep* adalah 17,61 mA .

Dengan menggunakan data hasil pengujian konsumsi arus dapat diketahui masa hidup daripada *node* sensor.

Catu daya *Power Bank* = 5V, dengan kapasitas 5000 mAh.

$$\text{Konsumsi arus selama 1 jam} = \frac{(I_{\text{aktif}} \cdot t_{\text{aktif}}) + (I_{\text{sleep}} \cdot t_{\text{sleep}})}{3600} \times 200 \dots\dots\dots (4-1)$$

$$\text{Konsumsi arus selama 1 jam} = \frac{(33,4 \cdot 5) + (17,61 \cdot 12)}{3600} \times 200 \dots\dots\dots (4-2)$$

$$\text{Konsumsi arus selama 1 jam} = 21,0177 \text{ mA} \dots\dots\dots (4-3)$$

$$\text{Masa hidup} = \frac{5000 \text{mAh}}{21,0177 \text{ mA}} = 237,894 \text{ Jam} = 9,912 \approx \pm 10 \text{ Hari} \dots\dots\dots (4-4)$$

Berdasarkan hasil perhitungan diatas dapat diketahui bahwa *node* sensor dapat bertahan selama 237 jam atau kurang lebih 10 hari dengan catu daya *power bank* 5000 mAh.

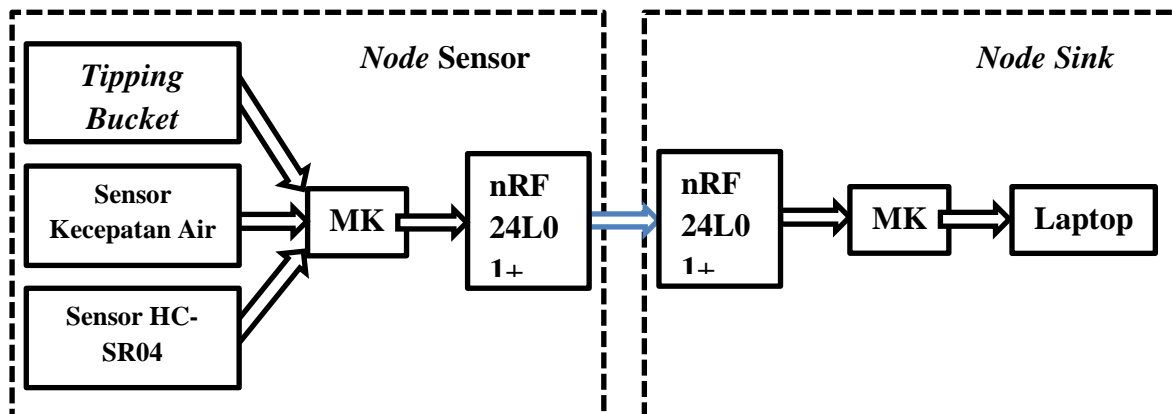
4.6 Pengujian Keseluruhan Sistem

Tujuan pengujian keseluruhan adalah untuk mengetahui apakah sistem yang telah dibuat dapat bekerja sesuai dengan perancangan. Dalam pengujian parameter yang diamati adalah sistem keseluruhan mampu menampilkan data curah hujan, kecepatan air, dan ketinggian air pada *serial monitor* secara berkala dan *node* sensor dapat bekerja dengan konsumsi daya yang kecil.

Alat yang digunakan dalam pengujian keseluruhan sistem adalah:

1. 2 Buah mikrokontroler Arduino Nano
2. Sensor Tipping Bucket
3. Sensor Kecepatan Air
4. Sensor Ultrasonik HC-SR04
5. 2 Buah nRF 24L01+
6. *Power Bank*
7. Laptop

Pengujian dilakukan dengan menjalankan perangkat keras dan perangkat lunak, kemudian dilakukan pemantauan pada *serial monitor* pada *Arduino IDE*.

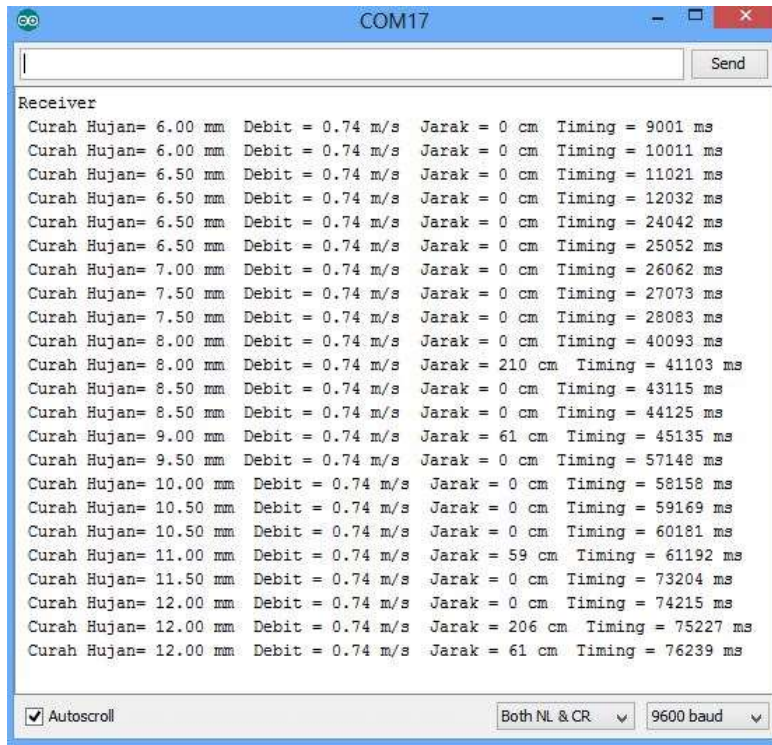


Gambar 4.8 Diagram Blok Pengujian Keseluruhan Sistem

Prosedur pengujian keseluruhan sistem adalah sebagai berikut.

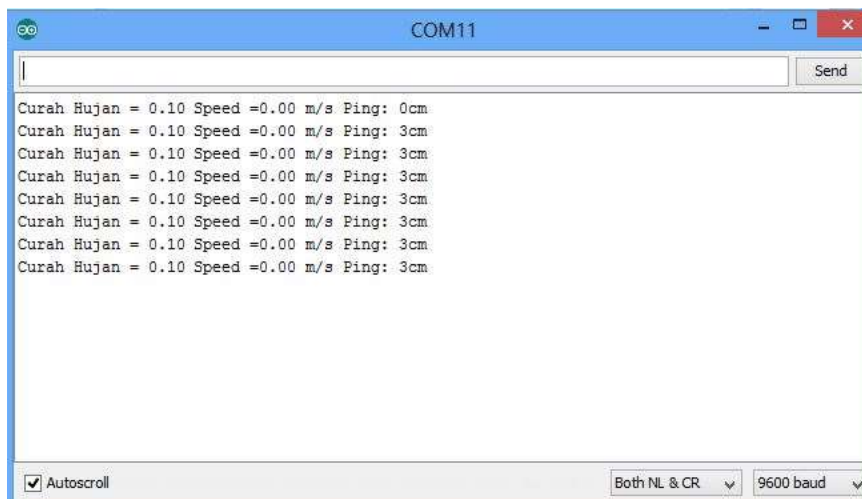
1. Rangkai rangkaian pengujian keseluruhan sistem sesuai Gambar 4.5.
2. Jalankan program pada *node sensor* dan *node sink*.
3. Berikan perlakuan pada masing – masing sensor pada *node sensor*.
4. Lihat hasil pembacaan sensor pada *Serial Monitor node sink*.

Pengujian keseluruhan menunjukkan bahwa program dan sistem telah mampu menampilkan informasi yang dikirim dari *node sensor* ke *node sink*. Data yang diterima dapat dibaca pada *serial monitor node sink*.



Gambar 4.9 Pembacaan pada Serial Monitor pada *Node Sink*

Pada *node sink* setiap 5 data yang diterima, terdapat perbedaan waktu selama kurang lebih 12 detik. Hal ini terjadi karena program pada *node* sensor mengaktifkan modul sampai data dikirim, setelah itu modul akan memasuki mode *sleep*. Lamanya *sleep* dapat diatur pada program *node* sensor itu sendiri.



Gambar 4.10 Hasil Pembacaan pada Serial Monitor pada pengujian *Node Sensor*

Halaman Ini Sengaja Dikosongkan