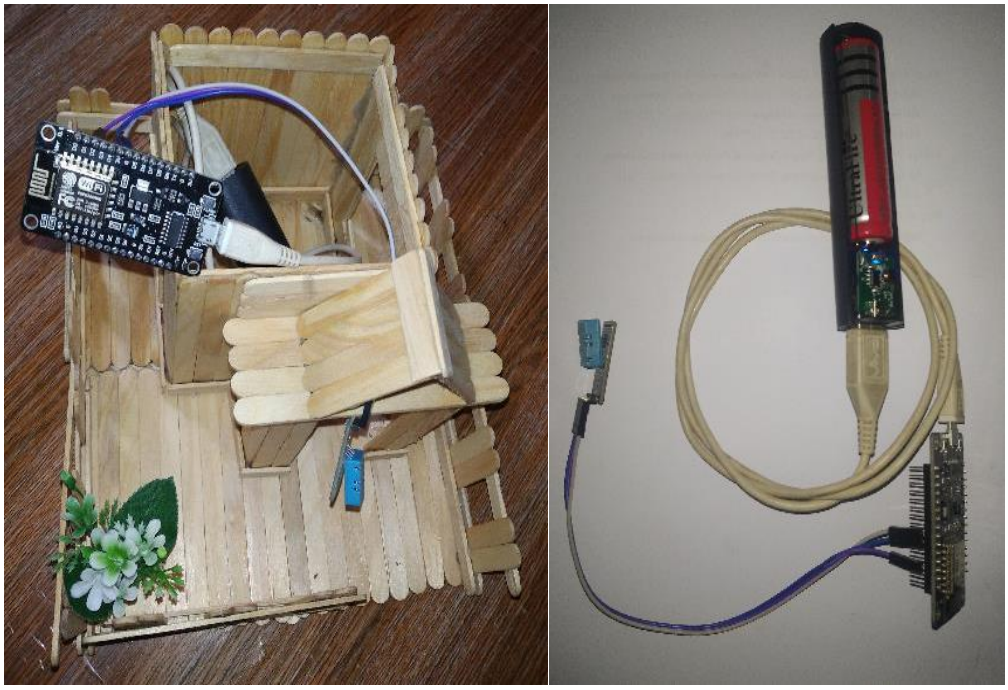
BAB IV
HASIL DAN PEMBAHASAN

Bab ini berisi tentang hasil uji coba dari sistem yang telah dirancang pada bab sebelumnya. Pengujian dimulai dengan memastikan setiap komponen pada perangkat keras serta perangkat lunak dapat bekerja sesuai dengan rancangan sebelumnya, setelah menguji perangkat keras dan perangkat lunak selanjutnya melakukan uji coba terhadap enkripsi serta dekripsi data.

4.1 Realisasi Perangkat Keras



Gambar 4.1 Node Sensor

Komponen utama perangkat keras yaitu modul sensor DHT11 dan mikrokontroller NodeMCU ESP8266. Modul sensor DHT11 dihubungkan ke microcontroler NodeMCU melalui pin – pin menggunakan kabel jumper sesuai dengan tabel 4.1 dibawah ini.

Tabel 4.1 Koneksi Pin Sensor dan Mikrokontroller

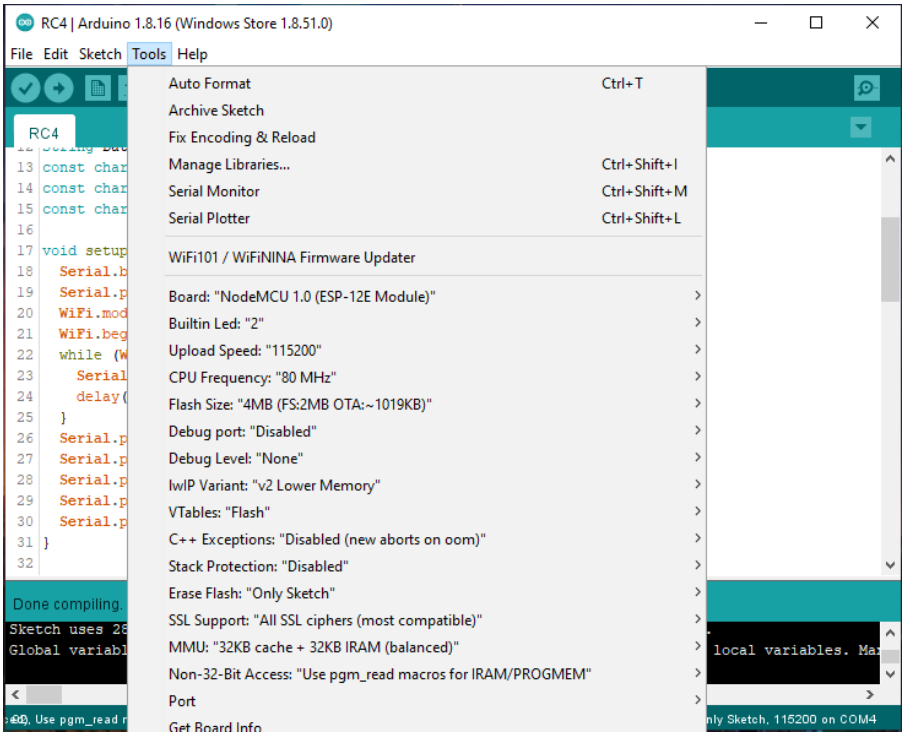
Pin NodeMCU ESP8266	Pin Modul Sensor DHT11
3V / VCC	VCC / +
G / GND	GND / -
D2	OUT / DATA

NodeMCU terhubung dengan laptop menggunakan kabel USB sebagi sumber daya dan dapat melakukan unggah kode program. Client dan server terhubung melalui koneksi internet dari smartphone (portable hotspot).

4.2 Realisasi Perangkat Lunak

Berikut ini merupakan realisai dari perancangan dari client dan server

4.2.1 Konfigurasi Arduino IDE (Client)

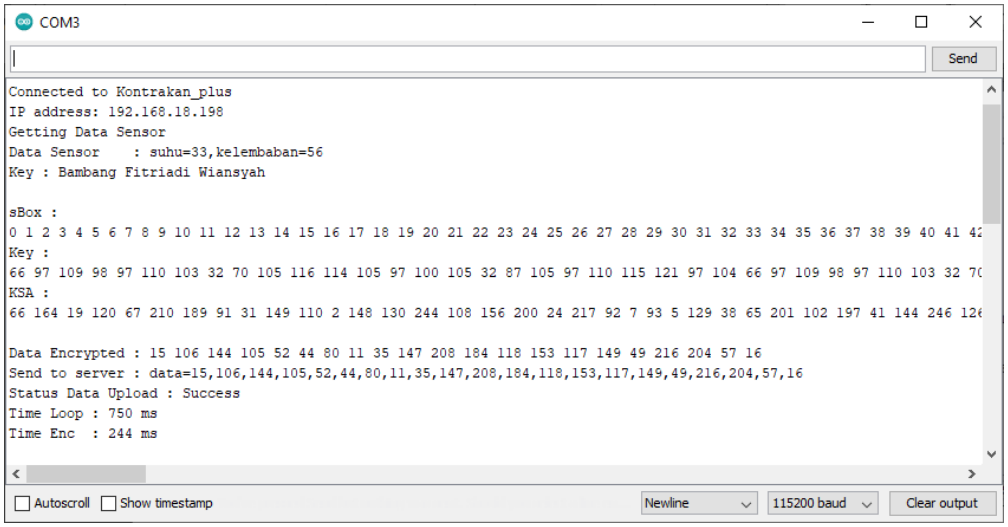


Gambar 4.2 Konfigurasi Arduino IDE untuk NodeMCU

Pada realisasi perangkat lunak ini menggunakan Arduino IDE yang di pergunakan untuk menulis kode program yang dijalankan pada mikrokontroller. Diperlukan adanya konfigurasi pada beberapa board yang ada di Arduino IDE agar dapat mengunggah kode program dengan benar tanpa terjadi error.

1. Board : NodeMCU 1.0 (ESP-12E Module)
2. Upload Speed : 115200
3. Debug Port : Disable
4. Flash Size : 4MB (FS: 2MB OTA: ~ 1019KB)

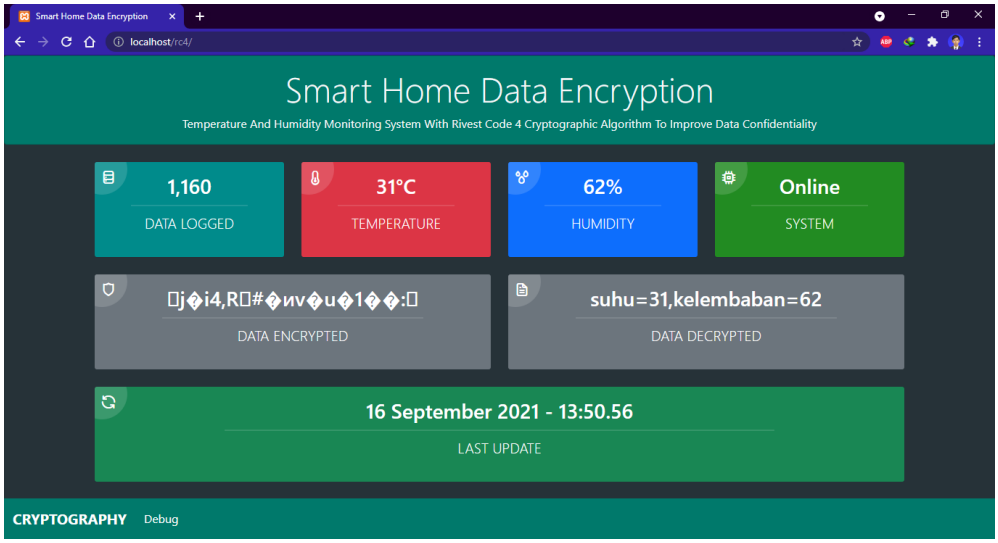
4.2.2 Hasil Program Pada Serial Monitor (Client)



Gambar 4.3 Hasil Proses Enkripsi Node Sensor pada Serial Monitor

Pada gambar 4.3 menunjukan proses yang di jalankan oleh node sensor melalui serial monitor pada Arduino IDE. Node sensor terhubung pada jaringan wifi dengan alamat IP “192.168.18.198”, selanjutnya sensor DHT11 mengukur suhu dan kelembaban. Data suhu dan kelembaban kemudian di enkripsi menggunakan algoritma Rivest Code 4 menjadi *ciphertext*. Lalu data yang berupa ciphertext dikirim ke server.

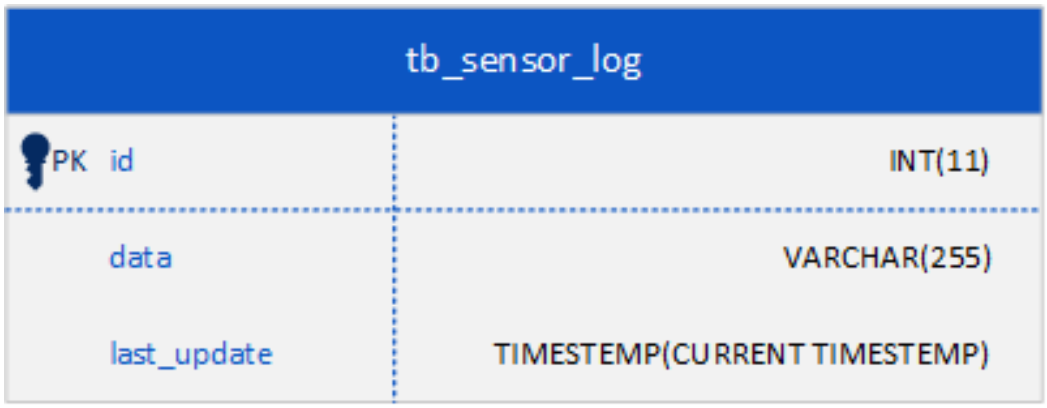
4.2.3 Tampilan pada Web Server



Gambar 4.4 Tampilan Antarmuka Web Server

Pada gambar 4.4 server menampilkan jumlah data sensor yang disimpan pada database, suhu, kelembaban, dan status node sensor secara realtime. Serta ditampilkan data yang masih terenkripsi dan yang telah di dekripsi.

4.2.4 Database pada Server



Gambar 4.5 Database pada Web Server

Gambar 4.5 merupakan tampilan database dengan nama database “iotech”, nama tabel “sensorlog” yang berisikan “data” untuk menyimpan data enkripsi yang di kirim oleh node sensor dan “lastupdate” untuk menyimpan waktu data saat dikirimkan.

4.2.5 Implementasi Algoritma Rivest Code 4

Algoritma Rivest Code 4 di implementasikan pada node sensor untuk melakukan enkripsi dan pada server untuk melakukan dekripsi. Pada node sensor enkripsi dilakukan setelah pembacaan suhu dan kelembaban selesai. Enkripsi pada node sensor melalui tahap sebagai berikut :

1. Enkripsi dimulai dengan melakukan inisalisasi variabel “sBox” dan “sKey”. Variabel “sBox” digunakan untuk permutasi deret angka pada variabel tersebut. Dan variabel “sKey” digunakan untuk menyimpan kunci dengan panjang 256-byte. Kunci dengan panjang kurang dari 256-byte akan di ulang hingga panjangnya mencapai 256-byte. Proses diatas dapat dilihat pada gambar 4.6.

```
71  key = (unsigned char *)"Bambang Fitriadi Wiansyah";
72  Serial.print("Key : ");
73  Serial.println((char *)key);
74  int sbox[256], skey[256];
75  for (i = 0; i < 256; i++) {
76    sbox[i] = i;
77    skey[i] = key[(i % strlen((char*)key))];
78  }
```

Gambar 4.6 Proses Inisialisasi Key dan SBox

2. Tahap kedua yaitu *Key Scheduling Algorithm* (KSA) dimana dilakukan permutasi pada variabel “sBox” dengan perhitungan “sKey”. *State automaton* diberi nilai awal berdasarkan kunci enkripsi. *State* yang diberi nilai awal berupa array yang merepresentasikan suatu deret permutasi dengan 256 elemen, sehingga hasil dari KSA adalah permutasi awal dengan indeks 0 sampai 255 dinamakan State. Proses tersebut dapat dilihat pada gambar 4.7.

```
89   Serial.println("\nKSA :");
90   for (i = 0; i < 256; i++) {
91     j = (j + sbox[i] + skey[i]) % 256;
92     temp = sbox[i];
93     sbox[i] = sbox[j];
94     sbox[j] = temp;
95   }
```

Gambar 4.7 Proses Key-Scheduling Algorithm

3. Tahap ketiga adalah *Pseudo-random Generation Algorithm* (PGRA) dimana pada proses ini bertujuan untuk menghasilkan *keystream*. Setiap putaran, bagian *keystream* sebesar 1 byte dengan nilai antara 0 sampai dengan 255 dihasilkan oleh PRGA berdasarkan state. Keystream yang dihasilkan pada proses PGRA akan di lakukan operasi XOR dengan plaintext untuk menghasilkan ciphertext. Proses tersebut dapat dilihat pada gambar 4.8.

```
101  Serial.print("\n\nData Encrypted : ");
102  for (i = j = n = 0; n < (int)strlen((char *)plaintext); n++) {
103    i = (i + 1) % 256;
104    j = (j + sbox[i]) % 256;
105    temp = sbox[i];
106    sbox[i] = sbox[j];
107    sbox[j] = temp; //swap
108    t = (sbox[i] + sbox[j]) % 256;
109    keyStream = sbox[t];//keyStream
110    Data += (plaintext[n] ^ keyStream);
111    Serial.print((plaintext[n] ^ keyStream));
112    if (n < (int)strlen((char *)plaintext) - 1) {
113      Data += ",";
114      Serial.print(" ");
115    }
116  }
```

Gambar 4.8 Pseudo-random Generation Algorithm

4.3 Pengujian Sistem

4.3.1 Pengujian Fungsional

Pada pengujian fungsionalitas dilakukan analisis terhadap kesesuaian fungsi hasil implementasi yang dibuat pada client dan server dengan perancangan. Peran client yang berupa node sensor adalah melakukan enkripsi data dan

melakukan pengiriman data berupa data “suhu dan kelembaban” yang didapat dari sensor DHT11 dalam bentuk ciphertext ke server. Server bertugas menerima request dan data dari client dan menampilkan data yang telah di dekripsi. Berikut tabel 4.2 merupakan hasil pengujian fungsionalitas.

Tabel 4.2 Hasil Pengujian Fungsional Sistem

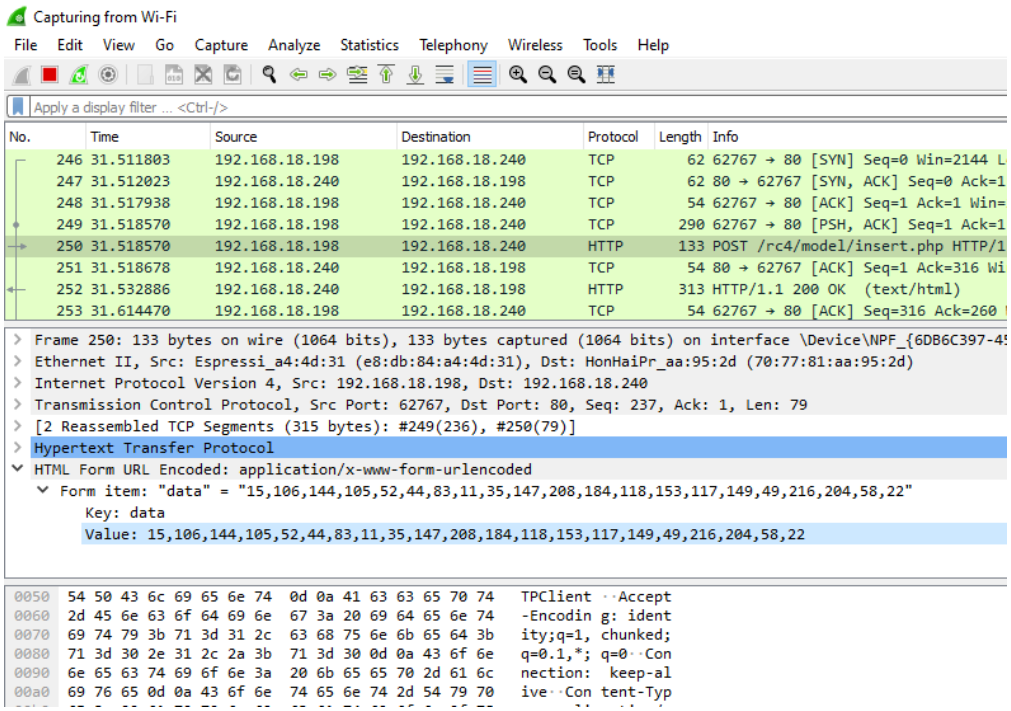
No	Pengujian	Hasil
1.	Menyambungkan ke jaringan wifi.	Berhasil
2.	Membaca data dari sensor DHT11.	Berhasil
3.	Mengenkripsi data dengan algoritma Rivest Code 4.	Berhasil
4.	Client mengirim HTTP request dan ciphertext ke server.	Berhasil
5.	Server menerima HTTP request dan ciphertext dari client.	Berhasil
6.	Mendekripsi ciphertext menggunakan algoritma Rivest Code 4.	Berhasil
7.	Menampilkan Hasil dekrip ciphertext pada web server	Berhasil

Node sensor berhasil terkoneksi ke jaringan wifi yang terhubung dengan internet. Pengujian sensor DHT11 dapat membaca suhu dan kelembaban dengan memanfaatkan library dht11 yang ada pada Arduino IDE. Dilakukan enkripsi pada data yang dibaca oleh sensor menjadi ciphertext menggunakan algoritma RC4. Data yang telah di enkripsi kemudian dikirimkan ke server menggunakan metode post pada protokol http. Data yang berupa ciphertext lalu di terima server dan di simpan pada database. Data ciphertext pada database akan di dekripsi dan di tampilkan pada *web page*.

4.3.2 Pengujian Packet Sniffing

Pada pengujian *packet sniffing* dilakukan dengan *capturing data* menggunakan aplikasi wireshark. Data yang didapat kemudian dianalisis untuk mengetahui bentuk data yang dikirim node sensor ke server. Ketika proses pengiriman data dari node sensor ke server dan sebaliknya, disinilah waktu yang paling rawan

terjadi tindak kejahatan *sniffing*. Sehingga data yang di transmisikan ke server bisa saja di lihat oleh orang lain bila tidak di enkripsi terlebih dahulu. Sehingga pada proses transmisi data sangat rawan terhadap kejahatan pencurian data. Data yang telah di enkripsi tetap dapat terlihat saat dilakukan pengujian packet sniffing, namun data yang di peroleh adalah data yang telah di enkripsi menjadi kode – kode tertentu sehingga bentuk data yang sebenarnya tidak diketahui. Berikut gambar 4.9 yang memperlihatkan proses *capturing data* pada aplikasi wireshark.



Gambar 4.9 Data enkripsi yang di capture oleh wiresharak

4.3.3 Pengujian Waktu Enkripsi

Pada pengujian waktu enkripsi dilakukan menghitung lama waktu yang dibutuhkan node sensor untuk melakuakan enkripsi dengan panjang key yang berbeda. Penggunaan key yang berbeda bertujuan untuk mengetahui apakah panjang key berpengaruh terhadap kecepatan enkripsi data. Digunakan 3 key yang berbeda, dimana setiap key digunakan untuk mengenkripsi 5 data. Hasil pengujian waktu enkripsi dapat dilihat pada tabel 4.3, tabel 4.4, dan tabel 4.5 dibawah ini.

Tabel 4.3 Hasil Pengujian dengan panjang Key 25-byte

Key : Bambang Fitriadi Wiansyah		
Data : suhu=29,kelembaban=65		
No	**Lama Waktu Enkripsi (ms)**	**Lama Waktu Proses Loop (ms)**
1	246	320
2	245	697
3	245	365
4	244	367
5	245	578

Tabel 4.4 Hasil Pengujian dengan Panjang Key 64-byte

Key : QfQyFD2SyraWCCsC94W9QKAPnM3gKSHaQfQyFD2SyraWCCsC94W9QKAPnM3gKSHa		
Data : suhu=29,kelembaban=65		
No	**Lama Waktu Enkripsi (ms)**	**Lama Waktu Proses Loop (ms)**
1	241	475
2	241	439
3	241	368
4	241	390
5	241	786

Tabel 4.5 Hasil Pengujian dengan Panjang Key 4-byte

Key : FpHx		
Data : suhu=29,kelembaban=65		
No	**Lama Waktu Enkripsi (ms)**	**Lama Waktu Proses Loop (ms)**
1	241	573
2	242	625
3	243	480
4	243	557
5	241	458

Lama waktu enkripsi dihitung dari inisialisasi variable key dan sbox hingga didapatkan ciphertext. Sedangkan lama waktu proses loop adalah keseluruhan proses mengambil data sensor, melakukan enkripsi data, dan mengirim ke server yang dihitung dalam satuan *millisecond*.

4.3.4 Pengujian Performa Sistem

Pada pengujian ini betujuan untuk menyandingkan performa node sensor yang tidak menggunakan enkripsi dan yang menggunakan enkripsi algoritma Rivest Code 4. Hasil pengujian performa sistem dapat dilihat pada tabel 4.6:

Tabel 4.6 Hasil Pengujian Performa Node Sensor

No.	Lama Waktu Total Proses Pengiriman Data ke Server	
	Dengan RC4 (ms)	**Tanpa Algotirma RC4 (ms)**
1	674	322
2	551	278
3	367	237
4	466	167
5	565	298
6	431	255
7	544	241
8	451	244
9	655	231
10	412	288
Avg	511	256

Rata – rata waktu proses pengiriman data yang dilakukan node sensor ke web server memerlukan waktu 511 ms pada sistem yang menggunakan algoritma RC4. Sedangkan pada sistem yang tidak menggunakan algoritma RC4 memerlukan waktu rata – rata 256 ms. Sistem yang menggunakan algoritma RC 4 memerlukan waktu tambahan rata – rata 250 ms untuk melakukan enkripsi data sebelum dikirimkan ke web server.

4.3.5 Pengujian Black Box

Pada pengujian ini dilakukan untuk melihat apakah sistem secara konsistem menghasilkan keluaran yang valid dan konsisten. Akan dilakukan enkripsi dan dekripsi pada sistem dan memastikan sistem bekerja sebagaimana mestinya. Dimana data yang di enkripsi node sensor dapat di dekripsi oleh server dengan keluaran yang valid dan konsisten. Hasil pengujian terdapat pada tabel 4.7.

Tabel 4.7 Hasil Pengujian Black Box

Key : Bambang Fitriadi Wiansyah			
Data : suhu=29,kelembaban=62			
No	Hasil Enkripsi Node Sensor	Hasil Dekripsi Server	Hasil
1	15, 106, 144, 105, 52, 45, 90, 11, 35, 147, 208, 184, 118, 153, 117, 149, 49, 216, 204, 58, 20	suhu=29,kelembaban=62	Valid
2	15, 106, 144, 105, 52, 45, 90, 11, 35, 147, 208, 184, 118, 153, 117, 149, 49, 216, 204, 58, 20	suhu=29,kelembaban=62	Valid
3	15, 106, 144, 105, 52, 45, 90, 11, 35, 147, 208, 184, 118, 153, 117, 149, 49, 216, 204, 58, 20	suhu=29,kelembaban=62	Valid
4	15, 106, 144, 105, 52, 45, 90, 11, 35, 147, 208, 184, 118, 153, 117, 149, 49, 216, 204, 58, 20	suhu=29,kelembaban=62	Valid
5	15, 106, 144, 105, 52, 45, 90, 11, 35, 147, 208, 184, 118, 153, 117, 149, 49, 216, 204, 58, 20	suhu=29,kelembaban=62	Valid
6	15, 106, 144, 105, 52, 45, 90, 11, 35, 147, 208, 184, 118, 153, 117, 149, 49, 216, 204, 58, 20	suhu=29,kelembaban=62	Valid
7	15, 106, 144, 105, 52, 45, 90, 11, 35, 147, 208, 184, 118, 153, 117, 149, 49, 216, 204, 58, 20	suhu=29,kelembaban=62	Valid
8	15, 106, 144, 105, 52, 45, 90, 11, 35, 147, 208, 184, 118, 153, 117, 149, 49, 216, 204, 58, 20	suhu=29,kelembaban=62	Valid
9	15, 106, 144, 105, 52, 45, 90, 11, 35, 147, 208, 184, 118, 153, 117, 149, 49, 216, 204, 58, 20	suhu=29,kelembaban=62	Valid

10	15, 106, 144, 105, 52, 45, 90, 11, 35, 147, 208, 184, 118, 153, 117, 149, 49, 216, 204, 58, 20	suhu=29,kelembaban=62	Valid
11	15, 106, 144, 105, 52, 45, 90, 11, 35, 147, 208, 184, 118, 153, 117, 149, 49, 216, 204, 58, 20	suhu=29,kelembaban=62	Valid
12	15, 106, 144, 105, 52, 45, 90, 11, 35, 147, 208, 184, 118, 153, 117, 149, 49, 216, 204, 58, 20	suhu=29,kelembaban=62	Valid
13	15, 106, 144, 105, 52, 45, 90, 11, 35, 147, 208, 184, 118, 153, 117, 149, 49, 216, 204, 58, 20	suhu=29,kelembaban=62	Valid
14	15, 106, 144, 105, 52, 45, 90, 11, 35, 147, 208, 184, 118, 153, 117, 149, 49, 216, 204, 58, 20	suhu=29,kelembaban=62	Valid
15	15, 106, 144, 105, 52, 45, 90, 11, 35, 147, 208, 184, 118, 153, 117, 149, 49, 216, 204, 58, 20	suhu=29,kelembaban=62	Valid

Hasil pengujian di ambil dengan membandingkan ciphertext dan plaintext yang ada pada node sensor dan data pada web server. Plaintext yang di enkripsi menjadi ciphertext akan di dekripsi pada web server menjadi plaintext. Kemudian di bandingkan antara data plaintext pada node sensor dan plaintext pada web server, jika data plaintext tersebut sama maka data valid.

4.4 Analisis Kerja Sistem

4.4.1 Kelebihan Sistem

1. Mengamankan data sensor yang dikirim melalui protokol HTTP.
2. Sistem dapat melakukan enkripsi dan dekripsi sesuai rancangan.

4.4.2 Kekurangan Sistem

1. Hanya melakukan enkripsi data yang ditransmisikan melalui protokol HTTP.
2. Tidak melakukan autentikasi data yang diterima oleh server sebelum menyimpannya ke database.