

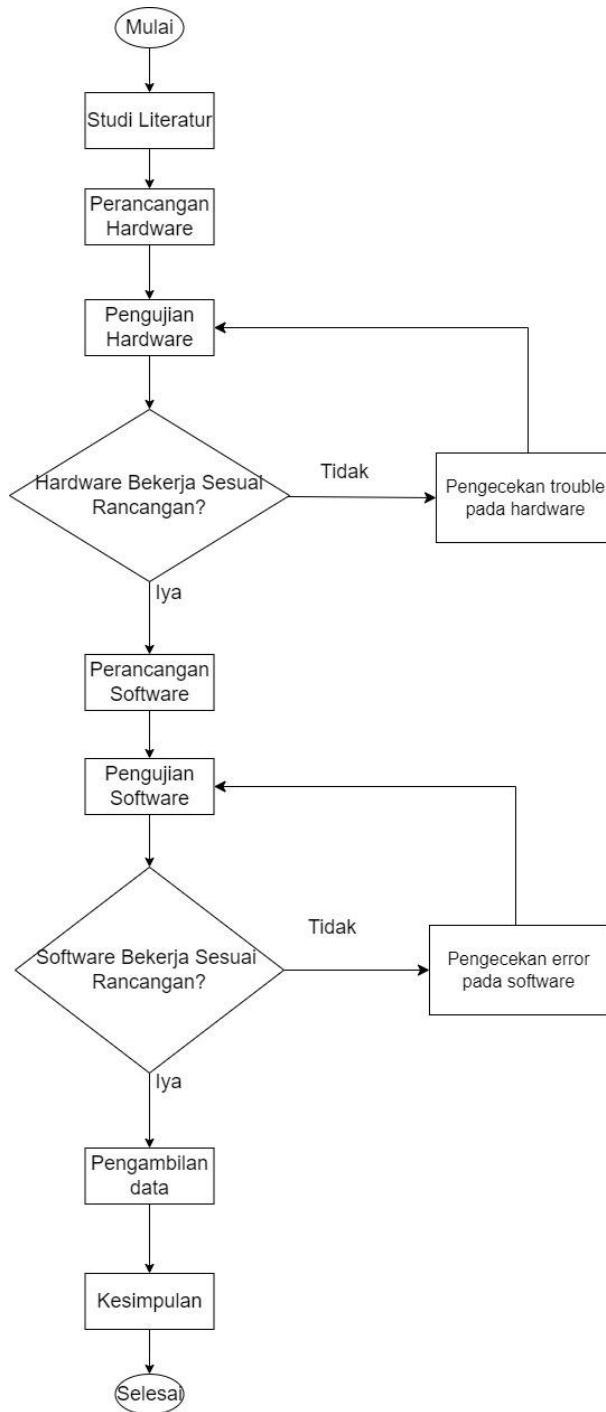
BAB III

METODE PENELITIAN

Aplikasi yang dirancang untuk alat *monitoring* kualitas garam akan mampu memantau kualitas garam di tambak garam secara *real-time* dengan bantuan teknologi LoRa. Informasi kualitas garam akan diketahui pada saat sensor mengambil data yang hasilnya akan ditampilkan pada layar monitor. Berikut metodologi pada penulisan penelitian ini.

3.1. ALUR PENELITIAN

Sistematika sistem ini dilakukan dalam berbagai tahap yaitu dengan melakukan literatur yang banyak, implementasi perancangan perangkat keras, melakukan perancangan perangkat lunak, melakukan pengujian sesuai parameter, dan yang terakhir adalah penyusunan hasil data dari hasil pengujian sistem. Dalam sebuah perancangan suatu penelitian diperlukan adanya alur penelitian yang bertujuan untuk menyelesaikan masalah yang akan diteliti secara terstruktur, sebagaimana dijelaskan pada Gambar 3.1.

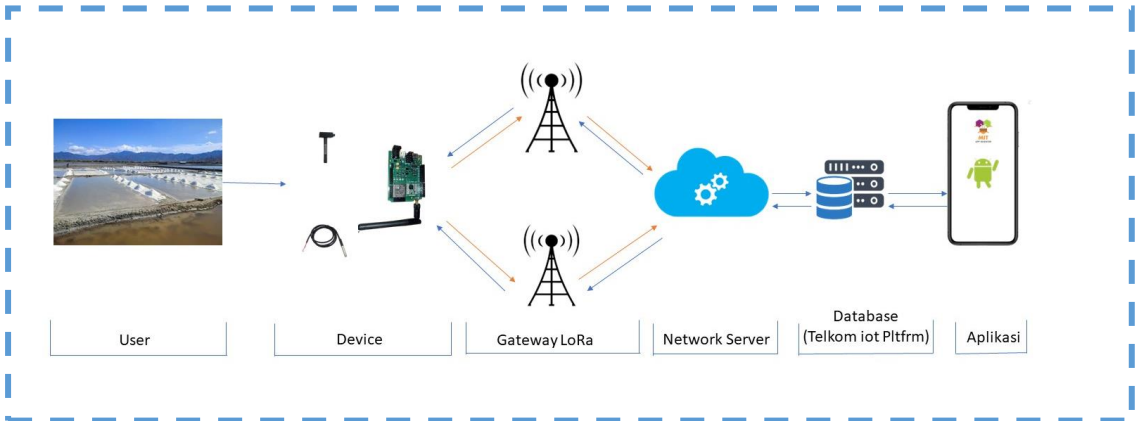


Gambar 3.1 *Flowchart* Penelitian

Berdasarkan *flowchart* penelitian pada Gambar 3.1 dapat diketahui bahwa untuk skema penelitiannya yaitu diawali dengan melakukan studi literatur dengan membaca terkait permasalahan-permasalahan garam di Indonesia dan mencari jurnal yang satu tema dengan penelitian penulis. Setelah melakukan studi literatur dan mengetahui solusi yang akan dibuat maka proses selanjutnya yaitu

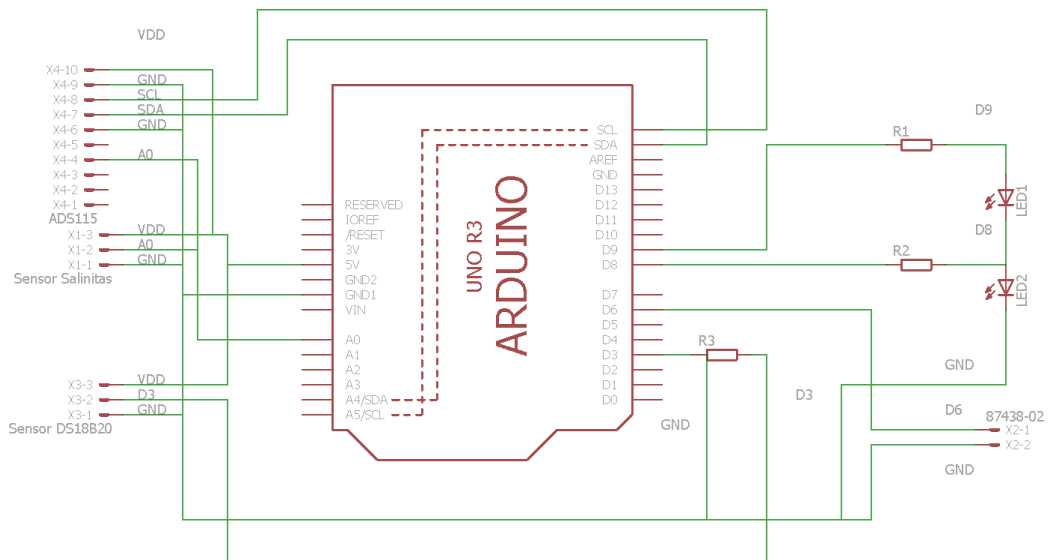
perancangan *hardware*. Pada proses perancangan *hardware* ini peneliti menyusun semua komponen yang sudah disediakan sesuai dengan skematik alat yang sudah dibuat. Proses selanjutnya yaitu proses pengujian sesuai dengan parameter, pada proses pengujian sensor-sensor akan dikalibrasi dengan alat ukur yang sudah sering digunakan oleh industrial dan untuk proses kalibrasi ini menggunakan metode regresi linier. Pada metode regresi linier menggunakan dua variabel yaitu variabel bebas (data hasil baca sensor) dan variabel terikat (data hasil baca alat ukur) dengan persamaan $Y = a + bX$, untuk a dan b merupakan koefisien dari regresi liniernya dan untuk menghasilkan nilai a dan b maka perlu melakukan perhitungan seperti pada persamaan rumus (2) dan (3). Alat ukur yang digunakan untuk mengkalibrasi yaitu *refractometer* salinitas dan alat ukur suhu air dan pada proses kalibrasi akan menggunakan sebanyak 3 sampel garam yang berbeda dengan pengambilan data per sampelnya sebanyak 10 kali percobaan dengan massa garam yang berbeda namun volume air tetap yaitu 1150 ml. Jika pengujiannya ditemukan ketidaksesuaian dan terjadi kesalahan maka dilakukan pengecekan kesalahan dan perancangan kembali pada *hardware*-nya sampai pengujian berhasil. Jika berhasil, peneliti dapat melakukan perancangan *software*, yaitu dengan melakukan perancangan *design* aplikasi MIT *App Inventor* yang nantinya digunakan untuk menampilkan hasil dari pembacaan sensor pada alat. Setelah proses perancangan selesai akan dilakukan proses pengujian yang mana sama halnya perancangan *hardware* jika terjadi kesalahan maka akan dilakukan pengecekan dan perancangan ulang sampai pengujian tersebut berhasil. Setelah pengujian *software* berhasil maka peneliti melakukan pengambilan data dan menganalisa hasil data tersebut, dari analisa tersebut penulis akan membuat suatu simpulan.

3.2. PERANCANGAN *HARDWARE*



Gambar 3.2 Arsitektur Sistem

Berdasarkan Gambar 3.2 dapat diketahui bahwa arsitektur dari sistem *monitoring* yang akan dibuat yaitu suatu *device* yang terdiri dari satu mikrokontroler dan sensor-sensor akan mengambil data dari tambak garam yang kemudian mengirimkannya melalui jaringan LoRa, kemudian disimpan pada database Telkom IoT Platform dan diteruskan ke aplikasi pada *handphone user*. Dari arsitektur tersebut dapat diketahui beberapa komponen penting yang digunakan sehingga sebelum dilakukan perancangan *wiring* pada *hardware* maka perlu merancang skematik rangkaiannya. Sebagaimana dijelaskan pada gambar dibawah ini untuk skematik rangkaiannya.



Gambar 3.3 Skematik Rangkaian

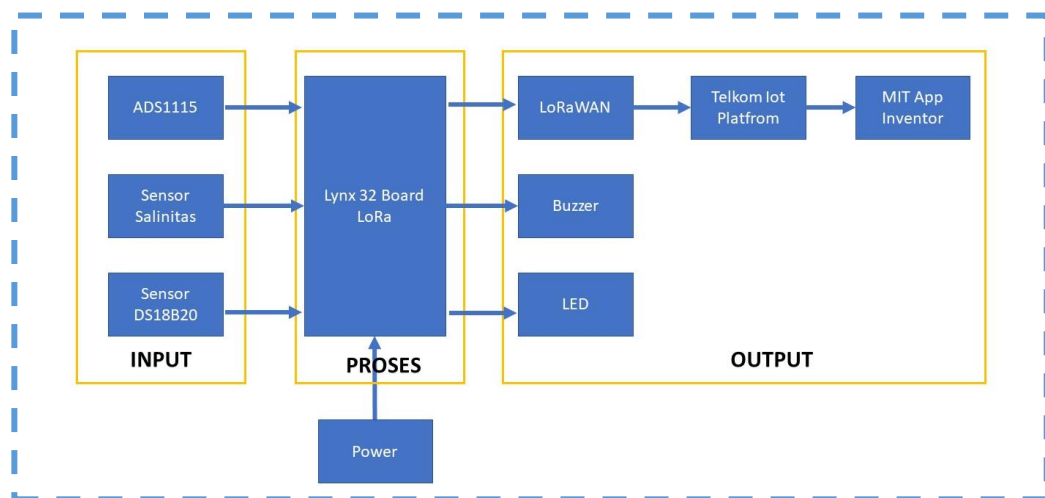
Berdasarkan gambar di atas dapat diketahui mikrokontroler yang digunakan yaitu Arduino Uno dikarenakan komponen tersebut memiliki besar yang sama

dengan lynx 32 *board* LoRa sehingga sebagai penggantinya menggunakan Arduino tersebut, sensor yang digunakan yaitu sensor salinitas dan sensor suhu/DS18B20, kemudian ada komponen tambahan seperti ADS115 sebagai pembaca pin analog, kemudian buzzer, LED sebagai *output* pemberi status kondisi pada alat.

Dibutuhkan beberapa komponen pelengkap lainnya untuk perancangan alat ini. Adapun alat dan bahan yang digunakan penulis sebagaimana dijelaskan Tabel 3.1.

Tabel 3.1 Alat dan Bahan

No	Hardware	Jumlah
1	Lynx32 LoRa <i>Development Board</i>	1
2	Salinitas TDS	1
3	DS18B20	1
4	<i>PCB double layer</i>	1
5	<i>Kabel Header</i>	20
6	Buzzer	1
7	LED	2
8	Laptop	1
9	Kabel USB	1
10	<i>Handphone</i>	1



Gambar 3.4 Diagram Blok Sistem

Berdasarkan Gambar 3.4 alur tahapan ini yaitu dimulai dari pembacaan beberapa sensor sampai pada proses pembacaan di aplikasi dan *website*-nya. Pada Gambar 3.4 dapat diketahui bahwa *lynx32 board LoRa* berfungsi sebagai mikrokontroler untuk menggerakkan sistem sekaligus terkoneksi ke jaringan. Sensor salinitas dan DS18B20 berfungsi sebagai *input* yang mengirim data pembacaan parameter salinitas dan suhu air ke *board LoRa*. Setelah menerima data, *board LoRa* akan mendapatkan daya dari *power supply* setelah mendapatkan daya maka *board* akan terhubung ke LoRa yang kemudian memproses data tersebut dan diteruskan menuju *output*. *Output* dari sistem ini yaitu *website* Telkom IoT Platform dan aplikasi MIT pada Android.

Pada *screen website* dan aplikasi di Android akan menampilkan hasil pembacaan sensor-sensor. *Lynx32 board LoRa* akan mengirim data pembacaan sensor-sensor ke Telkom IoT platform, di mana data tersimpan secara *real-time* dan dikirim ke aplikasi. Aplikasi yang telah jadi dan terinstal di Android digunakan untuk melakukan pemantauan perubahan pada tiap parameter kualitas garam. Sehingga petambak garam dapat dengan mudah mendapatkan informasi dari tiap kualitas garam yang diproduksi.

a. *Lynx32 Board LoRa*

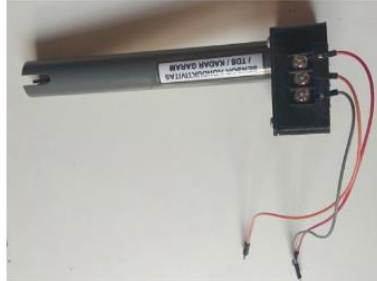
Penulis menggunakan *board* ini sebagai mikrokontroler dari sistem dan sebagai pengantar koneksi dari *device* ke jaringan LoRanya.



Gambar 3.5 Board LoRa

b. Sensor Salinitas/Konduktivitas

Penulis menggunakan sensor salinitas untuk membaca salinitas dari larutan garam yang digunakan sebagai sampel penelitian.



Gambar 3.6 Sensor Salinitas/Konduktivitas

Tabel 3.1 Konektivitas Pin Sensor Salinitas dengan *Board LoRa*

Sensor Salinitas	<i>Lynx32 Board LoRa</i>
VCC	5 V
GND	GND
DQ	A0

c. Sensor DS18B20

Penulis menggunakan sensor DS18B20 untuk membaca suhu air pada larutan garam yang digunakan sebagai sampel penelitian.



Gambar 3.7 Sensor DS18B20

Tabel 3.2 Konektivitas Pin DS18B20 dengan *Board LoRa*

Sensor DS18B20	<i>Lynx32 Board LoRa</i>
VCC	3,3V
GND	GND
DQ	IO13

d. LED

LED digunakan penulis sebagai notifikasi dalam keadaan alat sedang dalam kondisi menyala, kondisi kualitas di bawah standar, kondisi sesuai standar. Maupun kondisi alat sedang *trouble*.



Gambar 3.8 LED

Tabel 3.2 Konektivitas Pin LED dengan Board Lora

Warna	LED	<i>Lynx32 Board LoRa</i>
Merah	GND	GND
	Out	IO17
Hijau	GND	GND
	Out	IO26
Kuning	GND	GND
	Out	IO27

e. Buzzer

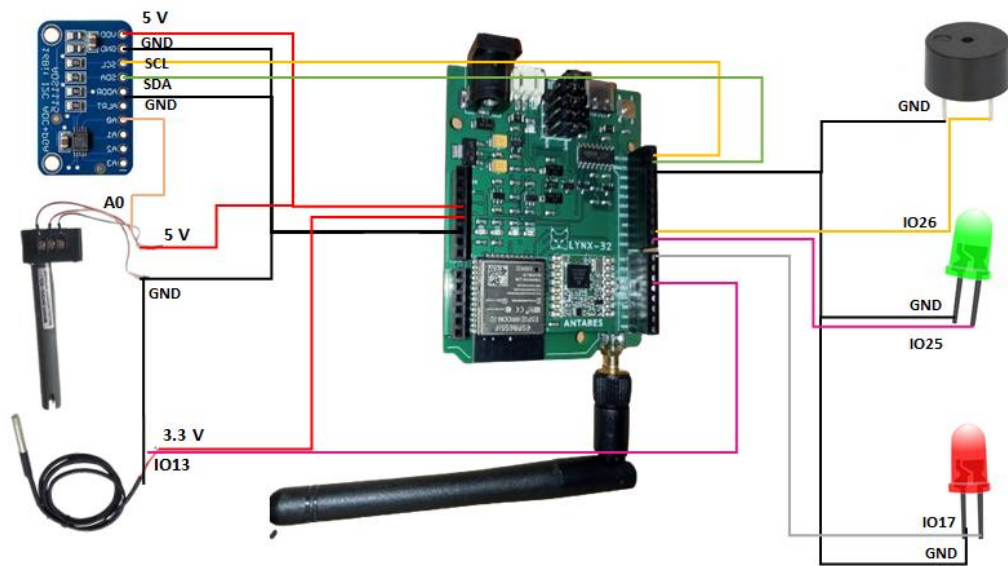
Buzzer digunakan penulis sebagai notifikasi dalam beberapa keadaan alat baik itu notifikasi kualitas garam maupun kondisi alat.



Gambar 3.9 Buzzer

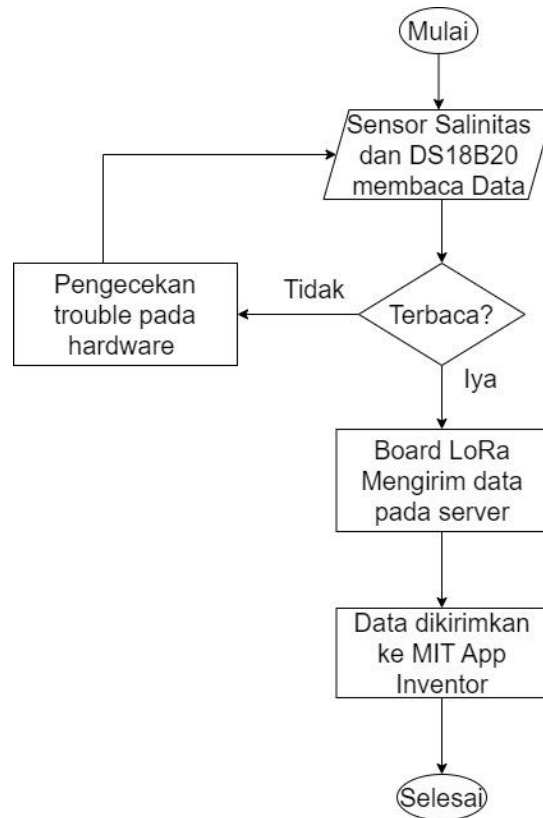
Tabel 3.3 Konektivitas Pin *Buzzer* dengan *Board LoRa*

Buzzer	<i>Lynx32 Board LoRa</i>
GND	GND
Out	IO25



Gambar 3.10 *Wiring Diagram*

Berdasarkan Gambar 3.10 dapat diketahui mengenai *wiring* atau jalur pengkabelan antar komponen-komponen pada alat *monitoring* ini sensor salinitas terhubung pada GND, pin 5V pada *board* LoRa dan untuk *output* terhubung pin A0 pada ADS1115, selanjutnya ada sensor DS18B20 di mana sensor ini terhubung pada pin GND, pin 3,3V, dan *output*-nya pada pin IO13. Pada LED merah terhubung pada GND dan *output*-nya pada pin IO17, LED hijau terhubung pada pin IO25, dan Buzzer terhubung pada pin IO26.



Gambar 3.11 *Flowchart Device Monitoring* Kualitas Garam

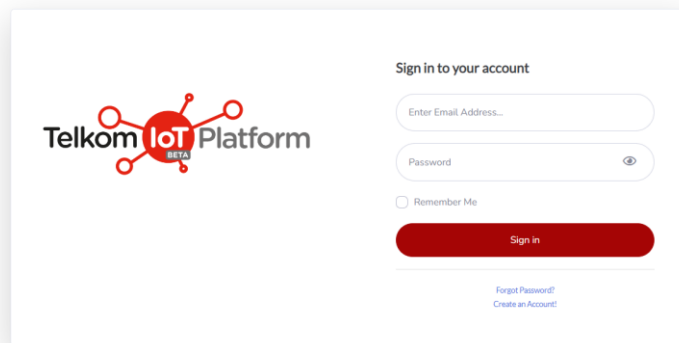
Berdasarkan Gambar 3.10 menunjukkan alur kerja dari alat *monitoring* kualitas garam. *Board* LoRa digunakan sebagai pengendali seluruh komponen yang terhubung pada *board* ini. Sensor-sensor akan membaca data dari sampel larutan garam, kemudian apabila sensor sudah berhasil membaca data maka dilanjutkan ke *board* LoRa untuk menerima data dan melakukan cek koneksi ke jaringan, jika sudah berhasil terkoneksi maka *board* LoRa akan meneruskan data tersebut ke *database*, dari *database* tersebut akan diteruskan ke *website* dan aplikasi yang sudah dibuat.

3.3. PERANCANGAN *SOFTWARE*

Perancangan *software* dilakukan dengan membuat perancangan program pada perangkat *hardware* dan perancangan *design* aplikasi pada perangkat *handphone*.

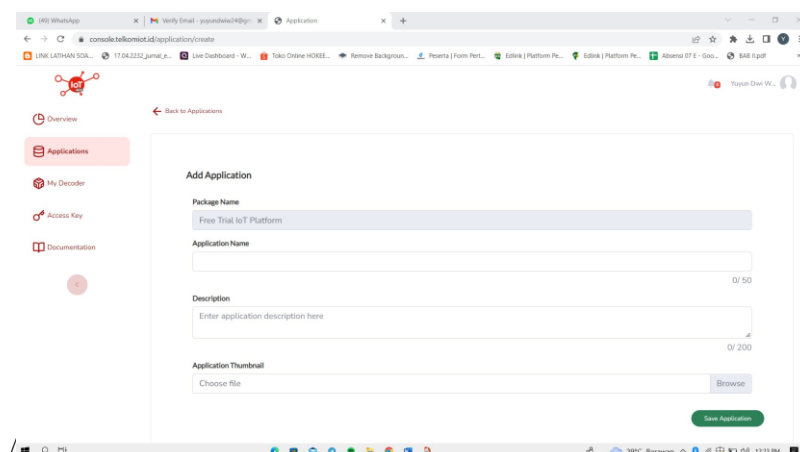
a. Telkom IoT Platform

Telkom IoT platform digunakan oleh penulis sebagai *database* yang memiliki konsep *real-time* sehingga data dari *device* akan tersimpan di *website* ini. Telkom IoT platform menggunakan *protocol* MQTT, karena *protocol* tersebut mempunyai metode yang efisien dalam mengambil data dan juga sudah tersedia pada platform Telkom IoT Platform.



Gambar 3.12 Halaman Depan Telkom IoT Platform

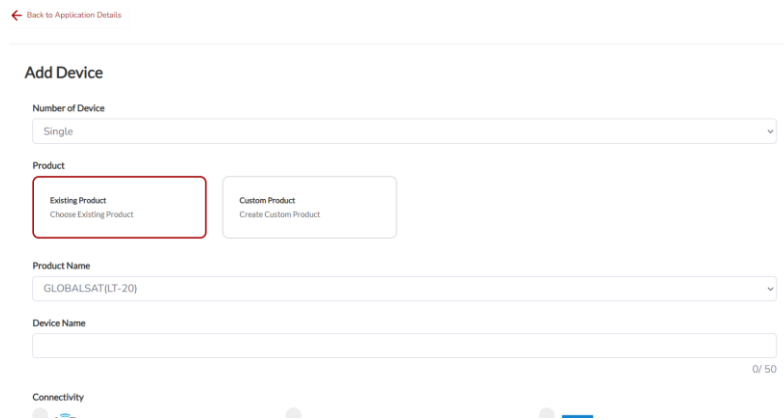
Pada halaman depan pengguna akan dianjurkan untuk mengisi email yang sudah terdaftar di Telkom IoT Platform dan *password*. Jika belum melakukan pendaftaran maka melakukan registrasi terlebih dahulu.



Gambar 3.13 Halaman Aplikasi

Pada halaman aplikasi pengguna akan dianjurkan untuk mengisi nama dari aplikasi yang akan dibuat dan paket yang digunakan itu apa.

Halaman ini bertujuan untuk membuat *device* yang akan dihubungkan dengan *hardware* yang sudah dibuat.

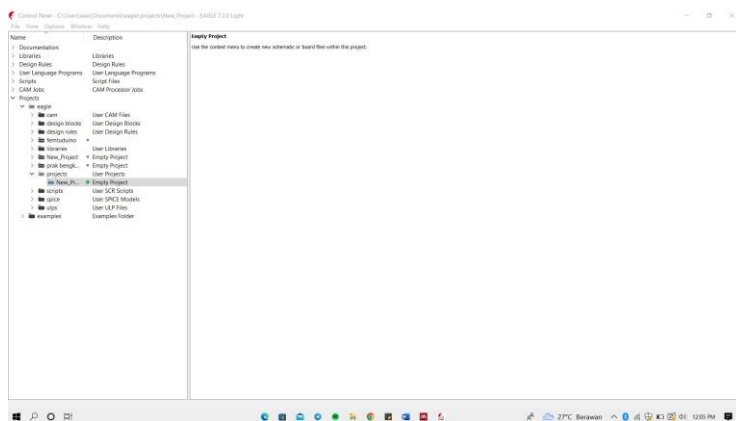


Gambar 3.14 Halaman *Device*

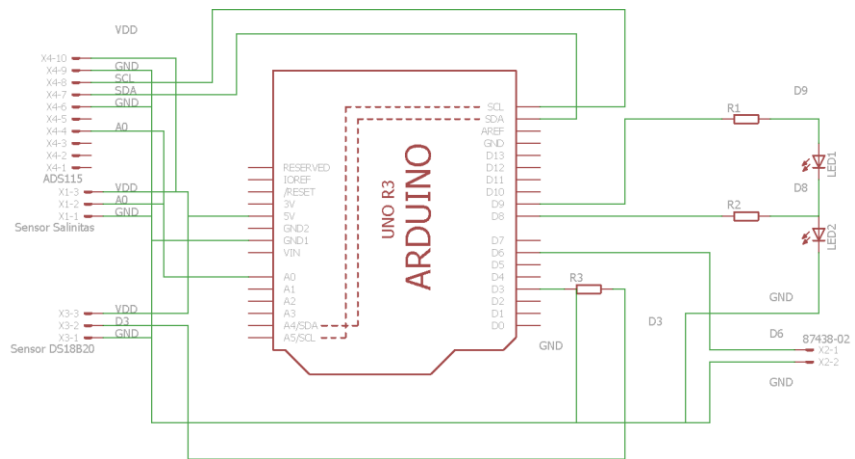
Pada halaman *device* pengguna akan mengetahui data yang didapatkan dari *hardware* dan pengguna dapat melihat histori tiap pengambilan data dari sensornya.

b. Eagle

Eagle digunakan oleh penulis untuk *men-desgin* skematik dari rangkaian *device* yang akan dibuat. Aplikasi ini mudah digunakan karena komponen-komponen telah tersedia pada aplikasi tersebut dan jika tidak tersedia maka pengguna hanya menambahkan *library* dari komponen yang akan ditambahkan tersebut.



Gambar 3.15 Halaman Utama Eagle

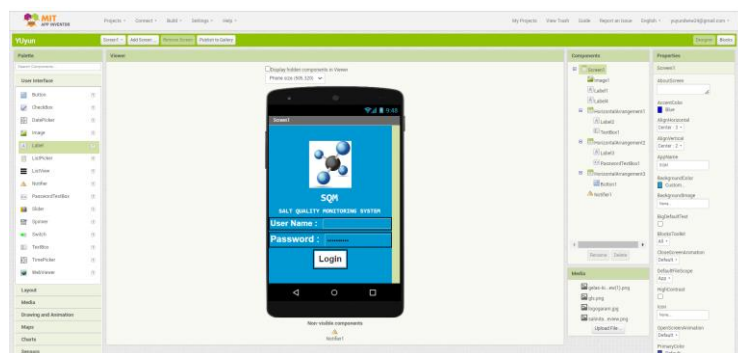


Gambar 3.16 *Design Skematik*

Berdasarkan Gambar 3.16 untuk dapat *men-design* maka penulis menambahkan semua komponen yang akan digunakan dalam membuat alat *monitoring*-nya. Pada gambar tersebut mikrokontroler yang digunakan yaitu Arduino bukan lynx32, hal itu dikarenakan *board* lynx32 belum ada library komponennya dan *board* Arduinolah yang hampir mirip dengan lynx32 tersebut. Penempatan sensor hanya menggunakan gambaran dari pin-pinnya saja sehingga tidak menggunakan komponen dari sensor itu sendiri.

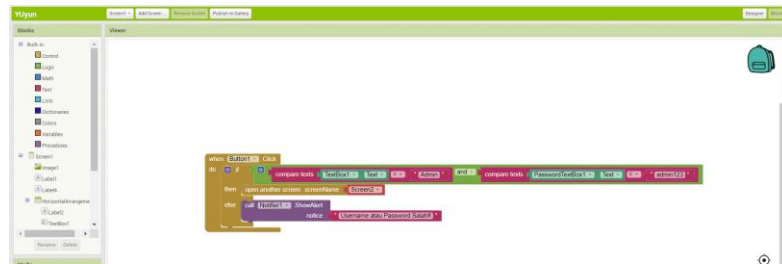
c. MIT App Inventor

Guna *men-design* aplikasi *monitoring*-nya penulis menggunakan *website* MIT App ini. App inventor ini mudah digunakan karena kodingan program sudah disediakan dalam bentuk blok-blok dan hanya disusun sesuai dengan yang akan dirancang.



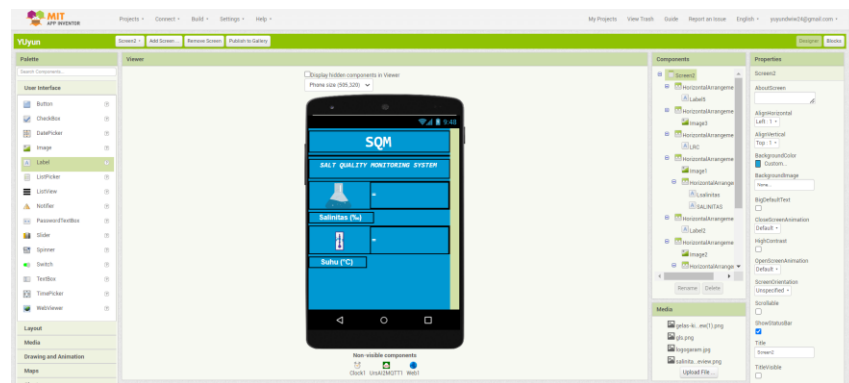
Gambar 3.17 *Design Halaman Depan Aplikasi*

Pada halaman depan ini pengguna akan diperintahkan untuk mengisi *username* dan *password*, ketika *username* atau *password* salah maka akan ada notifikasi bahwa *username/password* yang diisi salah. Disini pengguna dapat melihat logo dari aplikasi dan nama aplikasinya yaitu SQM yang mana kependekan dari “Salt Quality Monitoring System”.



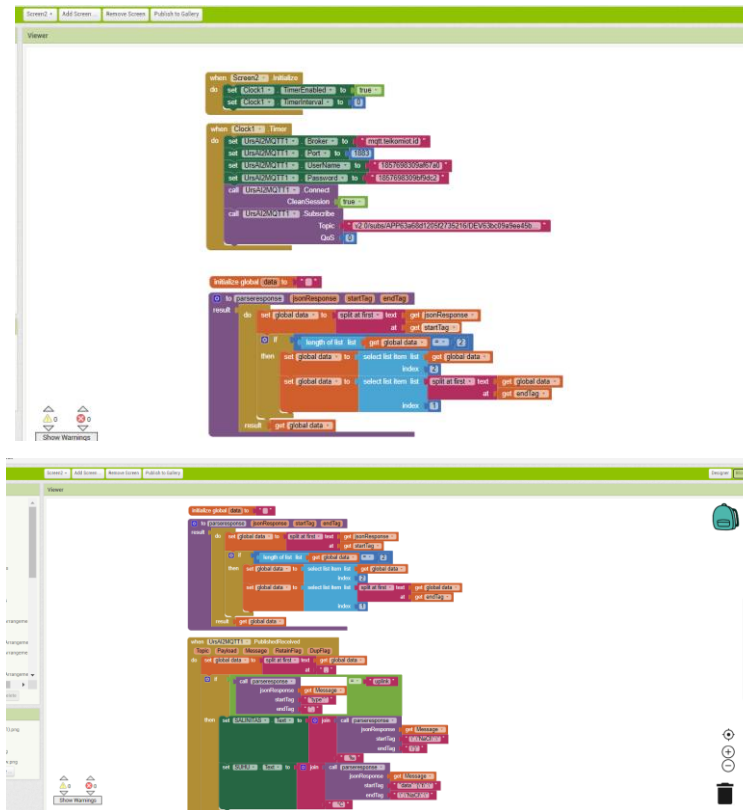
Gambar 3.18 Program Halaman Depan Aplikasi

Pada gambar 3.18 terdapat pemrograman halaman depan aplikasi. Pemrograman ini bertujuan untuk menjadikan tampilan awal pada aplikasi itu berjalan sesuai dengan yang diinginkan. Pemrograman berupa kode blok sehingga hanya menyatukan antara perintah satu dengan perintah lainnya.



Gambar 3.19 Halaman Kedua Aplikasi

Pada halaman kedua aplikasi pengguna dapat melihat secara *real-time* hasil pembacaan dari *device* yang sudah dibuat. Data yang ditampilkan berupa hasil pembacaan dari sensor salinitas dan sensor suhu DS18B20.



Gambar 3.20 Pemrograman Pada Halaman Kedua

Berdasarkan Gambar 3.20 pemrograman pada halaman kedua aplikasi SQM ini dirancang untuk menampilkan pembacaan data oleh sensor secara *real-time*, untuk menampilkan data tersebut maka diperlukan inisiasi variabel dan parsing data di mana data yang diambil hanyalah data *uplink* dari salinitas dan suhu, sehingga ketika *website* Telkom IoT platform *uplink* data maka secara langsung aplikasi akan menampilkan data tersebut. Protokol yang digunakan yaitu MQTT karena menyesuaikan protokol yang digunakan oleh *website* Telkom IoT platformnya. Jika melihat pemrograman tersebut terdapat beberapa perintah seperti “*when screen2.initialize*” pemrograman tersebut digunakan supaya ketika halaman kedua tampil maka timer akan hidup atau secara otomatis mengikuti waktu yang berlangsung dan *timeinterval* dibuat 0 supaya data yang terupdate langsung terupdate tanpa adanya jeda data tersebut harus hilang. Perintah selanjutnya yaitu” *when clock1. Timer*”, perintah tersebut digunakan untuk ketika waktu sudah berlangsung atau *real-time* maka halaman kedua ini akan

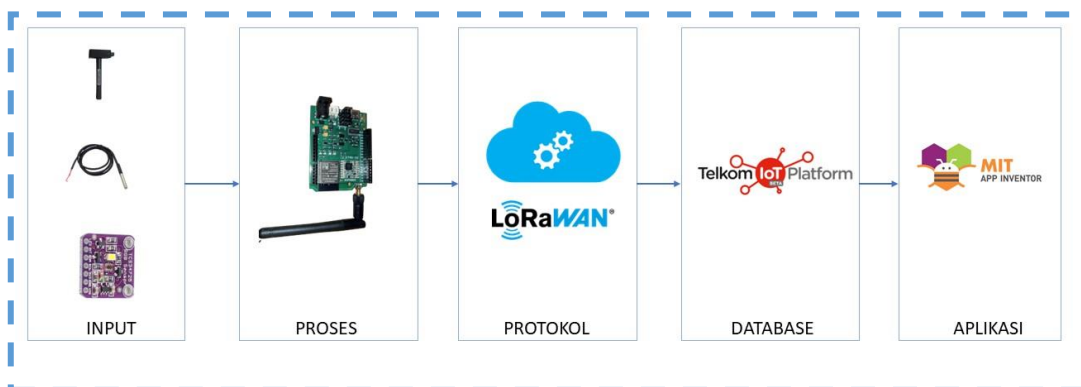
mengambil data dari Telkom IoT platform dengan menggunakan protokol MQTT. Selanjutnya ada inisialisasi global yaitu variabel data, untuk respon dari pengambilan data Telkom IoT platform ini berupa data json, dan parsing datanya dengan perintah “*when UrsAIMQTT1.publishedreceived*” yang mana berarti pada saat MQTT menerima data maka data tersebut akan diparsing atau difilter ke data *uplink* saja, ketika sudah terfilter maka data tersebut akan tampil pada kolom hasil data salinitas maupun suhu. Dengan perintah-perintah tersebut halaman kedua pada aplikasi dapat berjalan sesuai dengan rancangan penulis.

3.4. SKENARIO PENGUJIAN

Pada sub bab ini, penulis akan menjelaskan skema pengujian sistem *monitoring* kualitas garam berbasis lynx32 board LoRa.

3.4.1. Pengujian *Hardware*

Pengujian ini akan dilakukan sebanyak 90 kali dengan cara menguji alat ke beberapa sampel larutan garam yang terjual dipasaran. Hasil pengujian alat akan terbaca pada sistem *dashboard* Telkom IoT Platform dan MIT App Inventor dengan menampilkan data berupa suhu dan salinitas. Pengujian ini bertujuan untuk mengetahui alat kualitas garam sudah berjalan secara baik atau masih ada kendala lainnya.



Gambar 3.21 Diagram Blok Pengujian *Device*

- a. Alat yang digunakan:
 1. *Board* Lynx32 LoRa
 2. ADS1115

3. Sensor DS18B20
 4. Sensor Salinitas
 5. LED merah dan hijau
 6. Buzzer
- b. Prosedur Pengujian:
1. Melakukan penghubungan *jack male* DC ke *power* daya.
 2. Mengambil data larutan garam sampai berhasil menampilkan hasil pembacaannya.
 3. Menampilkan data di sampaikan atau bisa diakses di aplikasi Mit App Inventor dan Telkom IoT Platform.
 4. Mengulangi langkah pertama pada setiap pemantauan kualitas garam.

3.4.2. Pengujian Keakuratan Sistem Alat

Uji keakuratan sistem alat dilakukan dengan membandingkan hasil data pembacaan sensor dengan alat ukur standar pengukuran yang ada. Pengujian keakuratan sensor dilakukan pada sepuluh sampel yang berbeda. Besarnya nilai presentase kesalahan alat dapat dihitung dengan menggunakan persamaan berikut:

$$\text{Presentase kesalahan alat} = \frac{\text{Nilai alat standar} - \text{Nilai baca sensor}}{\text{Nilai alat standar}} \times 100\% \quad [8] \quad (4)$$

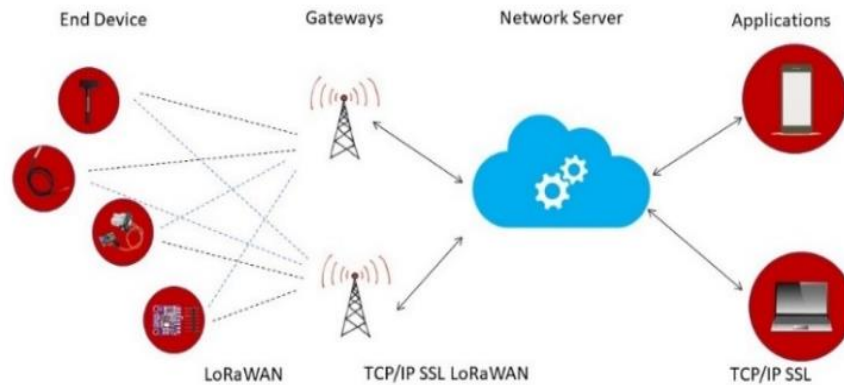
Sedangkan untuk menghitung nilai rata-rata presentase kesalahan alat dapat dihitung menggunakan persamaan berikut:

$$\text{Rata-rata presentase kesalahan alat} = \frac{\text{Presentase kesalahan alat}}{n \text{ (jumlah data)}} \quad [8]. \quad (5)$$

Selanjutnya untuk menghitung akurasi alat digunakan persamaan berikut:

$$\text{Akurasi} = 100\% - \% \text{Kesalahan alat} \quad [8]. \quad (6)$$

3.4.3. Pengujian *Latency*



Gambar 3.22 Alur Komunikasi Data Jaringan

Pengujian *latency* ini dilakukan dengan secara *end to end* dengan membandingkan antara waktu proses pengiriman data dari mikrokontroler sebagai *publisher* menuju server Telkom IoT Platform dengan waktu pengiriman kembali dari server menuju ke mikrokontroler sebagai *subscriber*. Hasil perbedaan waktu di tampilkan pada serial monitor, selanjutnya data akan diolah untuk melihat rata-rata dari keseluruhan *latency* jaringan yang didapat.

Tabel 3.4 Kriteria Nilai Delay

Kategori	<i>Delay</i>
<i>Very Good</i>	< 150 ms
<i>Good</i>	150 - 300 ms
<i>Fair</i>	300 - 450 ms
<i>Bad</i>	> 450 ms