BAB III
METODE PENELITIAN

3.1 ALAT DAN BAHAN

Dalam rencana pemeriksaan ini akan membutuhkan beberapa perangkat dan bahan untuk memesan peralatan (*equipment*) dan pemrograman (*programming*). Pada peralatan (*equipment*) akan dijelaskan tentang gambar rencana dan rangka yang akan dibuat. Sedangkan pada perencanaan pemograman (*programming*) akan dibuat pengertian tentang *stream chart programming* pada regulator dan *stream graph* pada aplikasi *android* untuk menampilkan informasi dari *firebase*. Daftar alat dan bahan untuk menyusun perangkat *hardware* dan perangkat *software* dapat dilihat pada Tabel 3.1.

Tabel 3.1 Daftar Alat dan Bahan

No	Alat Dan Bahan	Jumlah
1	Tempat Penyimpanan Pupuk	1
2	Laptop	1
3	*Smartphone*	1
4	ESP32	1
5	Sensor DS18B20	1
6	Sensor YL69	1
7	DC *Fan*	2
8	*Water Pump*	1
9	*Software* Arduino IDE	1
10	*Software* MIT App *Inventor*	1
11	*Software Wireshark*	1
12	*Google Firebase*	1

3.1.1 Tempat Penyimpanan Pupuk

Pada penelitian ini menggunakan tempat untuk menampung pupuk dengan ukuran 40x30x25 dengan menggunakan bahan dasar triplex. Tempat penyimpanan pupuk ini berfungsi sebagai wadah peletakan pupuk kandang yang akan diproses.

3.1.2 Laptop

Pada penelitian ini, mengunakan laptop sebagai alat untuk diguanakn mengakses aplikasi serta *software* perancangan sistem yang akan dibuat. Spesifikasi laptop yang digunakan yaitu menggunakan prosesor *ryzen* 5 dengan RAM sebagai 4 GB serta menggunakan kapasitas memori 1TB.

3.1.3 *Smartphone*

Pada penelitian ini menggunakan *smartphone* untuk menunjang penggunaan atau demo aplikasi *android*, dengan menggunakan spesifikasi *smartphone* sebagai berikut. *Smartphone version* 12 dengan ram 6 GB.

3.1.4 ESP32

Pada penelitian ini, ESP32 digunakan untuk melakukan proses pengolahan serta pengirim dan penerimaan data hasil baca sensor dari 32 ke *google firebase* dengan menggunakan modul *WiFi* ESP32 yang terpasang.

3.1.5 Sensor DS18B20

Pada penelitian ini sensor DS18B20 digunakan untuk memantau nilai suhu pada pupuk kendang pengkomposan yang nantinya data tersebut dikirim ke *database google firebase*.

3.1.6 Sensor YL69

Pada penelitian ini sensor YL69 digunakan memantau nilai kelembapan pupuk kandang dalam satuan persen. Tingkat kelembapan nantinya akan dikirim ke aplikasi pada *android* melalui *google firebase*.

3.1.7 DC *Fan*

Pada penelitian ini, DC *Fan* digunakan sebagai pendingin pupuk kandang menggunakan tegangan 9 V.

3.1.8 *Water Pump*

Pada penelitian ini, mengguankan *water pump* untuk memberikan air kepada pupuk kandang disaat pupuk dalam keadaan kering menggunakan tegangan 5 V.

3.1.9 *Software* Arduino IDE

Pada penelitian ini, menggunakan *software* Arduino ide untuk melakukan *programming* pada mikrokontroler ESP32. Dan sebagai aplikasi untuk memantau perancangan sistem dilihat menggunakan tampilan *serial* monitor.

3.1.10 *Software* MIT App *Inventor*

Pada penelitian ini, menggunakan *platform* MIT app *inventor* untuk perancangan aplikasi *android* untuk mempermudah rancangan menggunakan sistem *drag and drop* desain.

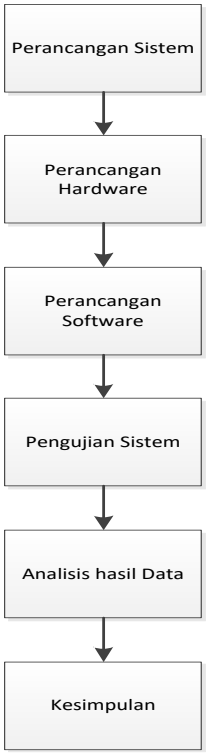
3.1.11 *Software Wireshark*

Pada penelitian ini, membutuhkan *wireshark* untuk mengukur serta men*capture* nilai dari kualitas jaringan yang dirancang, dari segi *delay*, *paket loss* dan *thoughput* pengiriman paket data suhu dan kelembaban serta kontroling untuk menyalakan dan mematikan komponen *output*.

3.1.12 *Google Firebase*

Pada penelitian ini, *Google Firebase* digunakan untuk *database* menyimpan hasil data dari perancangan sistem secara sistematis, kemudian data akan diteruskan ke aplikasi *android*.

3.2 ALUR PENELITIAN

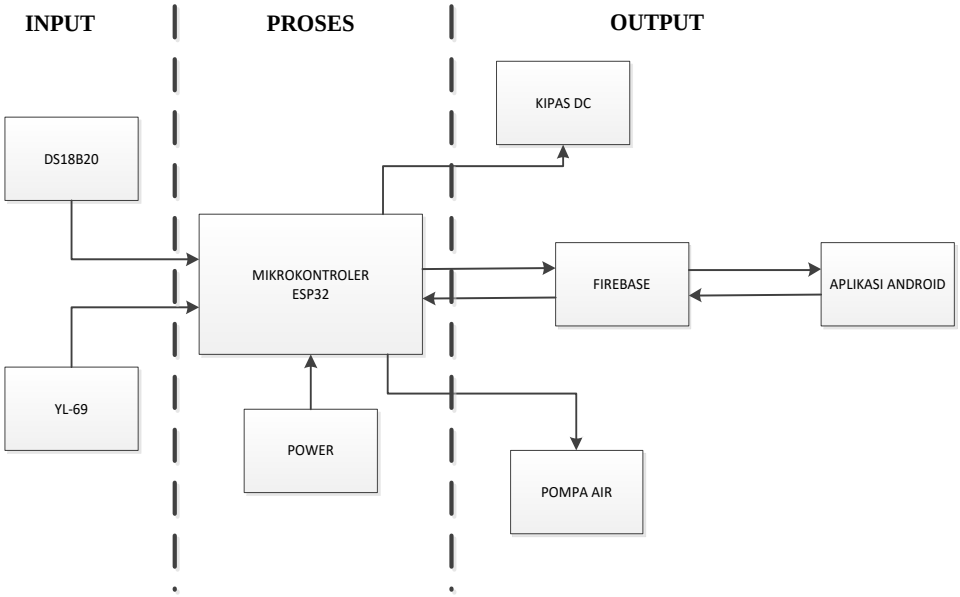


Gambar 3.1. Alur Penelitian

Pada Gambar 3.1 merupakan alur eksplorasi yang akan diselesaikan pada *review* ini. Pada tahap utama, khususnya rencana kerangka kerja, konfigurasi kerangka kerja adalah tahap mengenali bagian mana yang akan digunakan dalam rencana dan bagaimana menunjukkan masalah yang akan ditangani oleh kerangka kerja. Pada tahap selanjutnya, yaitu perencanaan peralatan tertentu, pada tahap ini pencipta merencanakan suatu alat dengan melihat bagian-bagian mana yang digunakan untuk membuat alat tersebut. Pada tahap ketiga khususnya perencanaan pemrograman, pada tahap ini pembuat melakukan konfigurasi pemrograman dengan merancang *script* yang akan diselesaikan pada setiap bagian yang telah dirancang pada produk di Arduino IDE. Pada tahap keempat, khususnya pengujian kerangka perangkat rencana, jika pada masa percobaan terjadi kesalahan dengan batas-batas, maka pada saat itu, peningkatan rencana peralatan dan konfigurasi pemrograman akan dilakukan di masa mendatang sampai pengujian efektif. Dengan asumsi tahap pengujian kerangka kerja telah efektif, berbagai informasi

akan diselesaikan untuk pengujian sehingga pembuat dapat membuat keputusan dari pengujian kerangka kerja.

3.2.1 Blok Diagram Sistem



Gambar 3.2 Blok Diagram Sistem

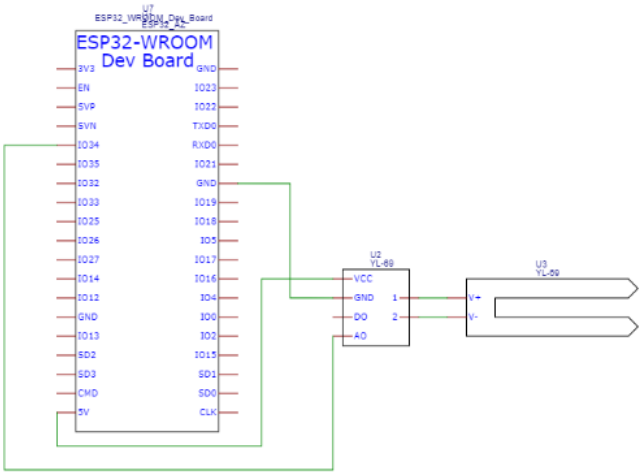
Pada Gambar 3.2 merupakan blok diagram perancangan yang akan dibuat dimana ESP32 berfungsi sebagai mikrokontroler untuk menggerakan sistem. Pada penelitian ini ada beberapa *input* yag digunakan seperti sensor DS18B20 untuk mengukur suhu pupuk dan YL-69 untuk mengatur kelembapan pupuk. Kemudian data yang terbaca oleh sensor dikirimkan ke ESP32 untuk diproses. Data yang diproses oleh mikrokontroler ESP32 diteruskan ke *output* sistem. *Output* sistem yang dihasilkan pada penelitian ini yaitu DC *Fan* yang digunakan untuk mendinginkan pupuk dan *water pump* yang digunakan untuk melembab kan pupuk. Kemudian *google firebase* sebagai *realtime database* yang akan diteruskan ke *platform* MIT App *Inventor* dan ditampilkan dalam bentuk aplikasi *android*. Aplikasi *android* akan menampilkan kondisi pembacaan sensor seperti suhu pupuk, dan tingkat kelembapan pada pupuk. Selain itu aplikasi android berfungsi untuk memberikan notifikasi apabila nilai suhu, dan kelembapan tidak pada kondisi normal.

3.3 PERANCANGAN *HARDWARE*

Pada perancangan *hardware* berisikan tentang perancangan *hardware* pada alat yang akan di buat pada penelitian.

3.3.1 Perancangan Sensor Kelembaban

Pada Gambar 3.3 Menjelaskan skematik rangkaian ESP32 dengan YL-69 dimana letak *port*/pin yang digunakan adalah pin VCC 5V, GND, dan pin 34.

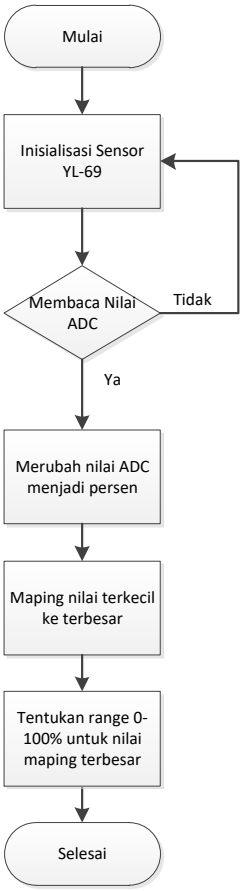


Gambar 3.3 Skematik rangkaian ESP32 dengan *YL*-69

Tabel 3.2 Koneksi Pin Sensor YL-69 Dengan ESP32

No	Pin ESP 32	Pin YL-69	Fungsi
1.	34	A0	Pembacaan Sensor *YL-69*
2.	5V	VCC	Memberi Daya 5 V
3.	GND	GND	*Grounding*

Pada Tabel 3.2 menunujukan letak posisi pin YL-69 yang dihubungkan dengan ESP32, dimana pin VCC menggunakan tegangan 5V dikarenakan tegangan kerja yang dibutuhkan oleh YL-69 berada pada kisaran 3,3 sampai 5 V. GND dihubungkan dengan GND sebagai *grounding*. Dan yang terkhir pin 34 sebagai pin *input* dari sensor menuju ke ESP32.

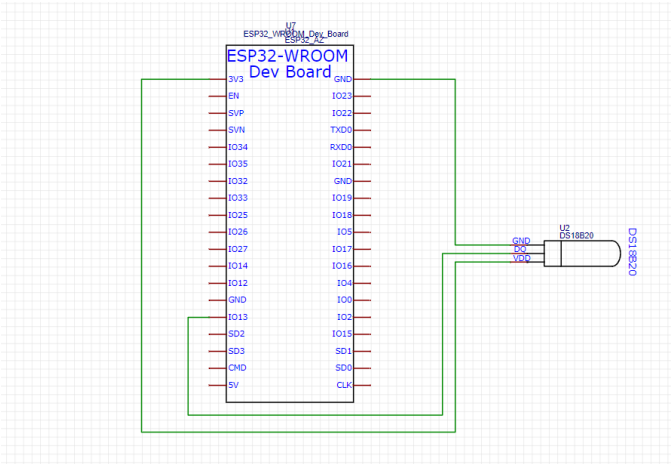


Gambar 3.4 *Flowchart* Pembacaan sensor YL-69

Pada Gambar 3.4 menunjukan *flowchart* dari cara pembacaan sensor YL-69 untuk memperoleh nilai. Sensor YL-69 membaca data dengan memberikan keluaran nilai analog yang dimana nilai tersebut tidak dapat diartikan kedalam nilai kelembaban. Oleh karena itu dibutuhkannya ADC untuk mengubah nilai pembacaan analog tadi kenilai digital kelembaban yaitu persen. Cara mengubahnya yaitu dari pembacaan ESP32 menggunakan nilai bit analog 12 bit, dimana nilai tersebut mempunyai nilai ADC sebesar 4095. Nilai tersebut dilakukan *mapping* untuk menentukan nilai persentase dari pembacaan sensor, dengan cara memasukan rentang 0-100%. Dengan nilai ADC terendah yang dibaca sensor sebesar 4095, nilai tersebut dijadikan nilai 0% dan nilai ADC tertinggi sebesar 1400 yang di peroleh dari pembacaan sensor maksimal ADC untuk nilai 100%. Untuk mendapatkan nilai persen, nilai ADC terendah di kurang nilai sensor di bagi nilai rentang pembacaan sensor dikali 100.

3.3.2 Perancangan Sensor Suhu

Pada Gambar dibawah ini Menjelaskan skematik rangkaian esp 32 dengan DS18B20 dimana letak port/pin yang digunakan adalah pin VCC 3 V, GND, dan pin 13.

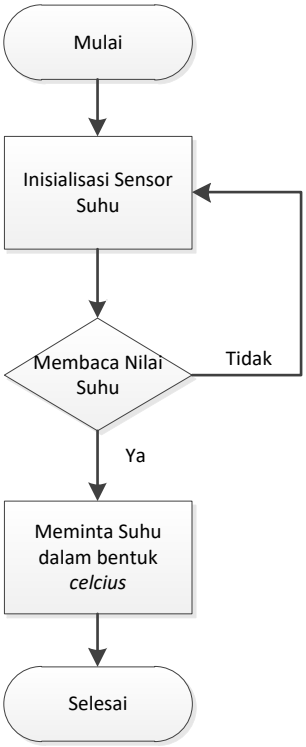


Gambar 3.5 Skematik Rangkaian ESP32 dengan DS18B20

Tabel 3.3 Koneksi Pin Sensor DS18B20 Dengan ESP32

No	Pin ESP 32	Pin DS18B20	Fungsi
1	13	A0	Pembacaan data sensor *dallas*
2	3V	VCC	Memberikan daya 3 V
3	GND	GND	*Grounding*

Pada Tabel 3.3 menunujukan letak posisi pin DS18B20 yang dihubungkan dengan ESP32, dimana pin VCC menggunakan tegangan 3 V dikarenakan tegangan kerja yang dibutuhkan oleh DS18B20 berada pada 3 V. GND dihubungkan dengan GND sebagai grounding. Dan yang terkhir pin 13 sebagai pin masukan dari sensor menuju ke ESP32.

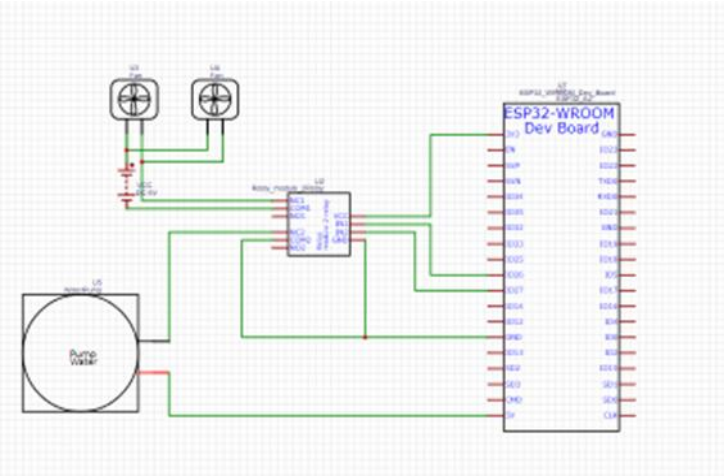


Gambar 3.6 *Flowchart* Pembacaan sensor DS18B20

Pada Gambar 3.6 menunjukan *flowchart* dari cara pembacaan sensor DS18B20, dimana cara pembacaan sensor DS18B20 dengan menggunakan nilai analog yang ditranslatekan menggunakan *library* DS18B20. Pada saat penginisialisasi sensor dan belum dapat membaca nilai suhu, maka proses lanjut ke penginialisasi sensor suhu sampai dapat menerima nilai suhu. Peneliti hanya perlu memberikan perintah untuk pengambilan data nilai suhu dalam *celcius* untuk memperoleh nilai suhu dalam *celcius*.

3.3.3 Perancangan *Relay*

Pada Gambar dibawah ini menunjukan skematik hubungan antara ESP32 dengan *relay*, yang dimana *relay* dihubungkan dengan ESP32 menggunakan pin 26 dan pin 27. Pin tersebut digunakan untuk memberikan keluaran dari ESP32 menuju ke *relay* untuk mengaktifkan atau mematikan *relay*.

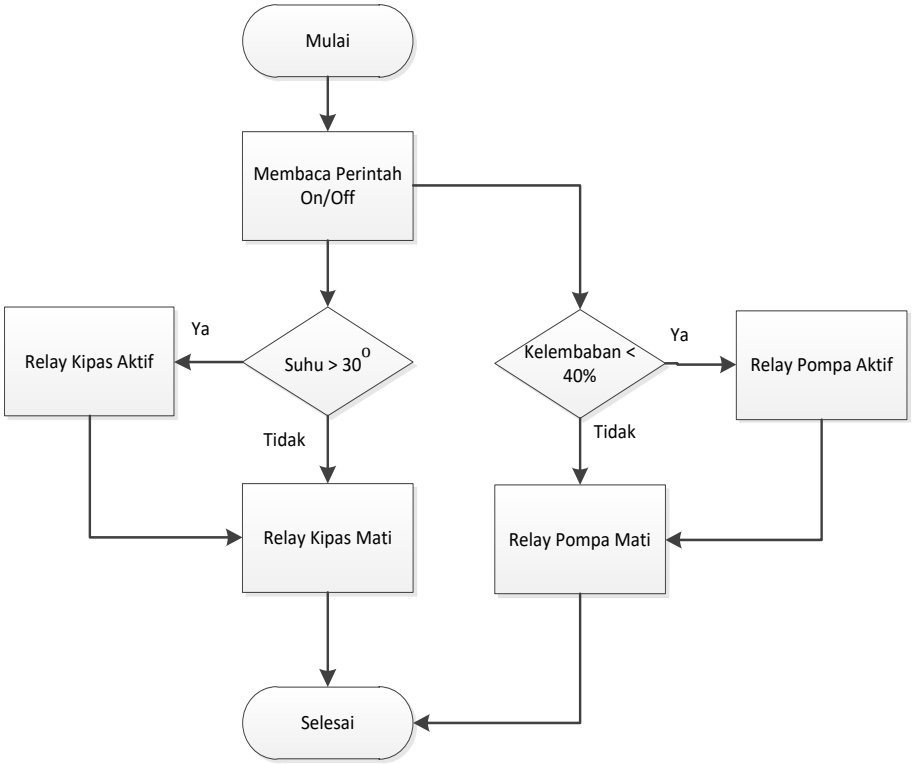


Gambar 3.7 Skematik Rangkaian ESP32 dengan *Relay*

Tabel 3.4 Koneksi Pin *Relay* Dengan ESP32

No	Pin ESP 32	Pin *Relay*	Fungsi
1.	3.3 V	VCC	Memberi Daya 3.3 V
2.	GND	GND	*Grounding*
3.	26	IN1	Inputan logika *relay* 1
4.	27	IN2	Inputan logika *relay* 2

Pada Tabel 3.4 menunujukan letak posisi pin *relay* yang dihubungkan dengan ESP32, dimana pin 26 dijadikan pin untuk memberi perintah kepada kipas, dan pin 27 dijadikan pin untuk memberi peritah kepada *water pump*. Tegangan yang digunakan untuk menyalakan *relay* berasal dari ESP32 sebesar 3,3V, tegangan untuk *water pump* berasal dari ESP32 sebesar 5V dan tegangaan untuk kipas berasal dari baterai 9V.

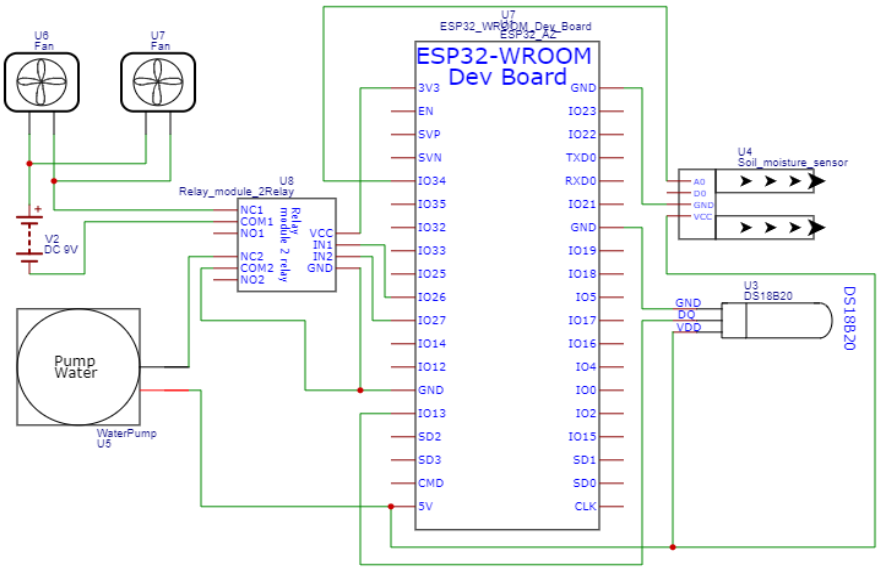


Gambar 3.8 *Flowchart* kerja *Relay*

Pada Gambar 3.8 menunjukan *flowchart* kerja *relay*. Dimana *relay* akan aktif jikamana menerima perintah dari ESP32. Baik perintah *on* maupun perintah *off*. Fungsi dari *relay* sendiri untuk memutuskan arus yang akan mengaliri kipas maupun pompa air, sehingga kipas dan pompa air dapat mati dan nyala sesuai dengan perintah yang diberikan melalui *relay*. *Flowchart* Gambar 3.8 menampilkan jika mana nilai suhu lebih dari 30°C maka *relay* kipas akan aktif, dan juga jika kelembaban kurang dari 40% maka *relay* pompa akan aktif.

3.3.4 Perancangan Sistem

Pada Gambar di bawah ini merupakan desain skematik perangkat keras yang akan digunakan dalam penelitian ini.



Gambar 3.9 Desain Skematik Perangkat Keras

Pada Gambar 3.9 penelitian ini menggunakna beberapa komponen seperti sensor DS18B20, sensor YL-69, *relay*, pompa air dan DC *Fan.* Dalam penelitian ini *mikrokontroller* yang digunakan adalah ESP32 yang berfungsi sebagai sebagai otak untuk menjalankan sistem. Berikut daftar tabel beberapa komponen yang terhubung dengan ESP32 :

Tabel 3.5 Koneksi Pin *Relay* Dengan ESP32

No	Pin ESP 32	Pin *Relay*	Fungsi
1.	3.3 V	VCC	Memberi Daya 3.3 V
2.	GND	GND	*Grounding*
3.	26	IN1	*Input* logika *relay* 1
4.	27	IN2	*Input* logika *relay* 2

Tabel 3.6 Koneksi Pin DC *Fan*

No	Pin Baterai	Pin Kipas	Fungsi
1.	Positif 9V	VCC	Memberi Daya 9 V
2.	Negatif 9V	GND	*Grounding* yang terhubung dengan *relay*, untuk mengatur putus tidaknya aliran listrik

Tabel 3.7 Koneksi Pin *Water Pump*

No	Pin ESP 32	Pin Pompa Air	Fungsi
1.	5 V	VCC	Memberi Daya 5 V
2.	GND	GND	*Grounding* yang terhubung dengan *relay*, untuk mengatur putus tidaknya aliran listrik

Tabel 3.8 Koneksi Pin Sensor YL-69 Dengan ESP32

No	Pin ESP32	Pin YL-69	Fungsi
1.	34	A0	Pembacaan Sensor *YL-69*
2.	5V	VCC	Memberi Daya 5 V
3.	GND	GND	*Grounding*

Tabel 3.9 Koneksi Pin Sensor *Dallas* DS18B20 Dengan ESP32

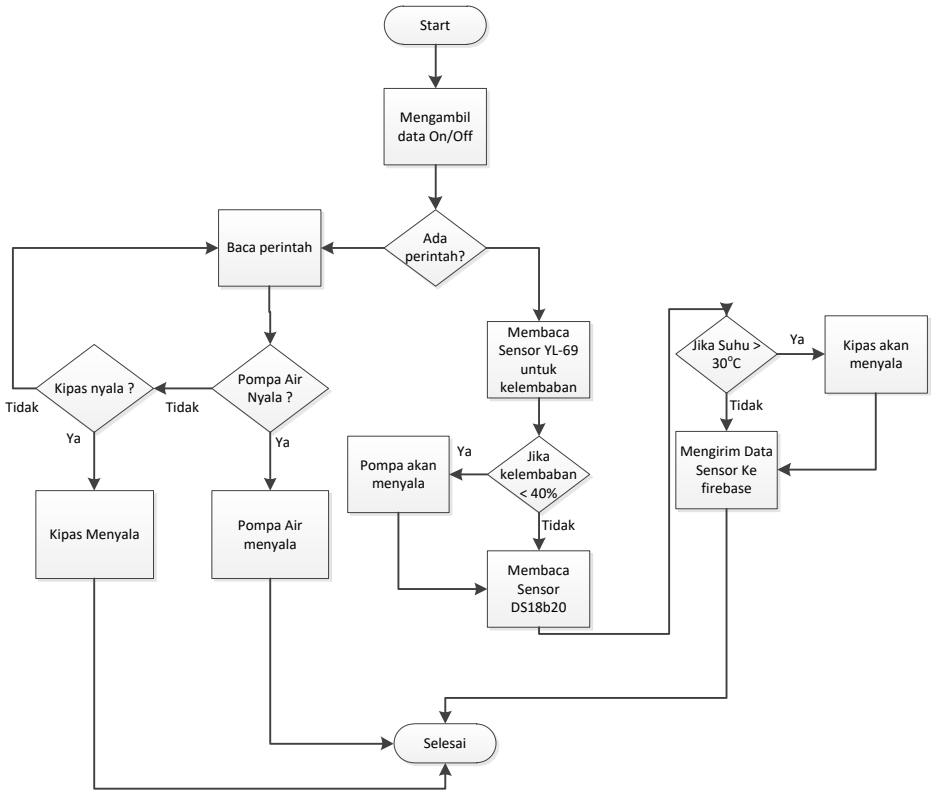
No	Pin ESP 32	Pin DS18B20	Fungsi
1	13	A0	Pembacaan data sensor *dallas*
2	3V	VCC	Catu daya 3.3 V dari ESP32
3	GND	GND	*Grounding*

3.4 PERANCANGAN *SOFTWARE*

Pada perancangan *software* berisikan tentang perancangan *software* pada alat yang akan di buat pada penelitian.

3.4.1 *Flowchart* Sistem *Mikrokontroller* Dan Perangkat Lunak

Flowchart sistem pada penelitian ini ditunjukan pada gambar 3.10 :

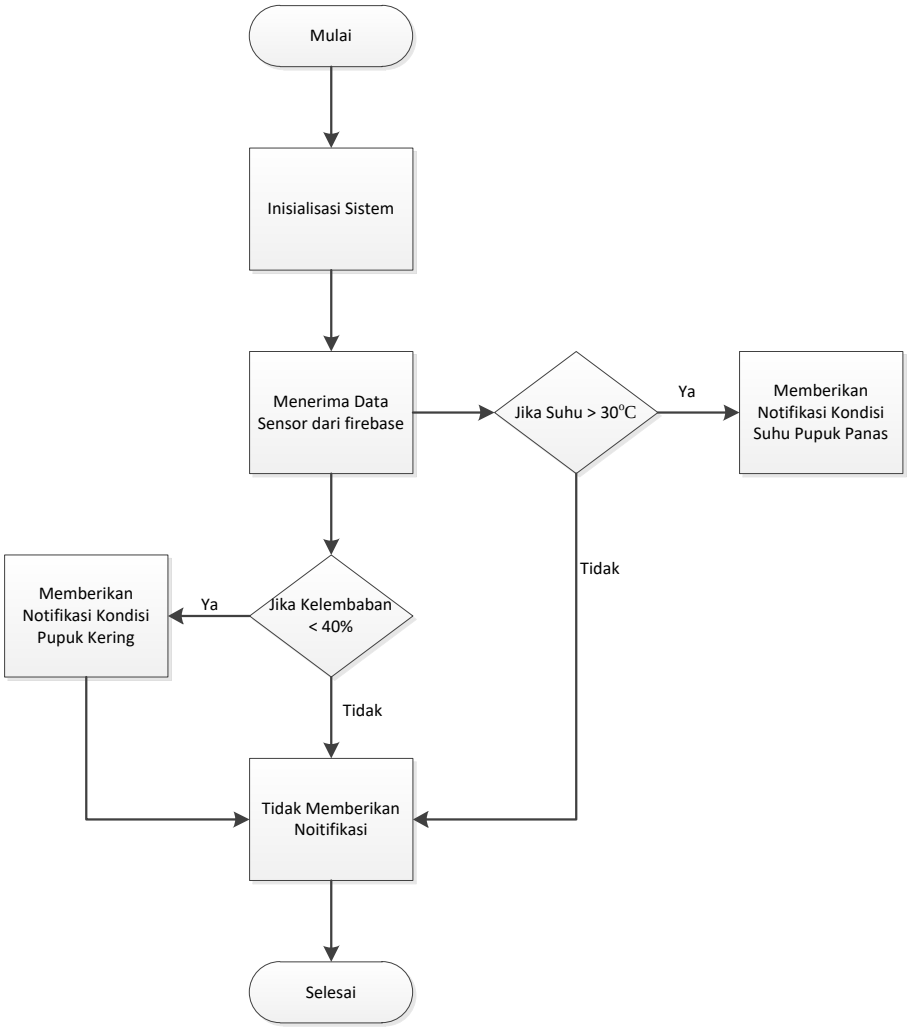


Gambar 3.10 *Flowchart* Sistem *Mikrokontroller*

Pada Gambar 3.10 merupakan *flowchart* sistem *mikrokontroller* menunjukan alur kerja *mikrokontroller* sesuai dengan perangkat lunak yang akan dirancang. *Mikrokontroller* akan melakukan perintah sesuai dengan program yang dimasukkan, dimana semua perangkat yang terhubung akan dikendalikan oleh *mikrokontroller*. Proses pertama yang dimulai yaitu membaca sensor YL-69 untuk mengetahui kelembaban pupuk, jika kondisi kelembaban pupuk berada dibawah 40%, maka pompa yang telah dirancang akan menyala untuk menyirami pupuk agar lembab. Kemudian dilanjutkan membaca sensor *dallas* DS18B20 untuk mengetahui kondisi suhu pupuk, jika kondisi suhu pupuk berada diatas 30°C, maka kipas akan menyala untuk mendinginkan pupuk. Dan dilanjutkan dengan

pengiriman data sensor ke *google firebase* sebagai *realtime database* menggunakan komunikasi *WiFi* dan jaringan internet.

Flowchart perangkat lunak pada sistem pada penelitian ini ditunjukan pada gambar 3.11 :



Gambar 3.11 *Flowchart* Sistem Aplikasi *Android*

Pada gambar 3.11 merupakan alur *flowchart* aplikasi *android* dimana proses pertama yaitu inisialisasi sistem untuk pemberian nilai awal kondisi yang dilakukan saat deklarasi variabel. Kemudian menerima data sensor dari *google firebase*. Apabila nilai data sudah didapatkan dan nilai suhu diatas 30°C, maka akan menampilkan notifikasi pada aplikasi *android* “Suhu Pupuk Panas”. Apabila nilai data sudah didapatkan dan nilai kelembaban di bawah 40%, maka akan menampilkan notifikasi pada aplikasi *android* “Kelembapan Pupuk dalam kondisi

kering”. Jika kondisi berada pada kondisi normal, maka tidak akan menampilkan notifikasi.

3.4.2 Aplikasi *Software*

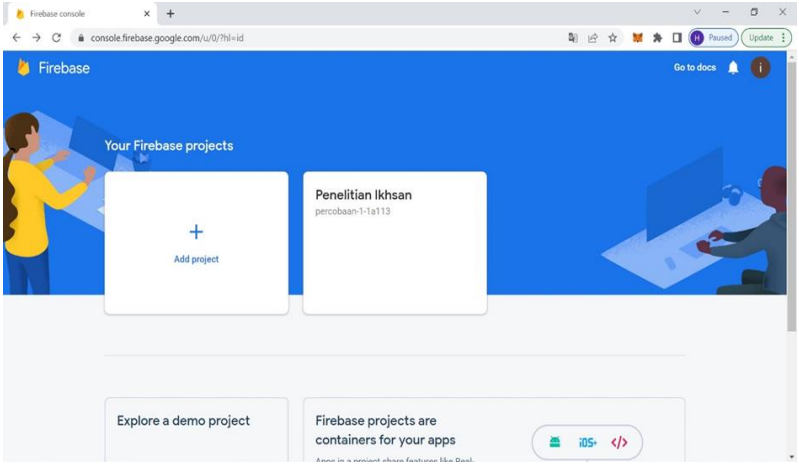
Dalam perancangan perangkat lunak dibagi menjadi 2 bagian yaitu perancangan *google firebase* dan MIT App *Inventor*.

a. Perancangan *Google Firebase*

Google firebase disini berfungsi sebagai *realtime database* untuk penjunjang pembacaan sensor pada aplikasi *android*. Berikut merupakan Langkah-langkah perancangan *google firebase*:

1. Pembuatan projek *database*

Pembuatan projek *database* untuk memisahkan *database* projek yang satu dengan yang lain.

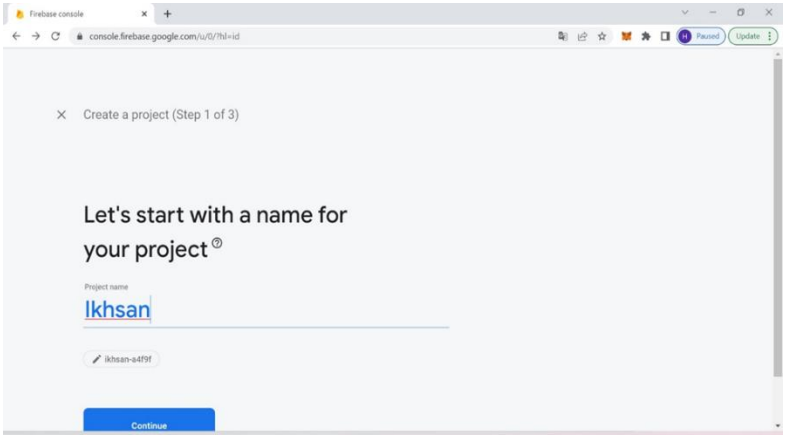


Gambar 3.12 Tampilan *Google Firebase* Membuat Projek Baru

Pada Gambar 3.12 merupakan tampilan awal membuat projek baru pada *google firebase*. Apabila dalam perancangan database tidak dipisah-pisah dapat menyebabkan *database* projek satu dengan yang lain akan bentrok dan menyebabkan *database* susah dimengerti, karena database pada projek satu dengan yang lain bercampur. Maka pentingnya pemisahan projek dengan membuat projek baru.

2. Penamaan projek *database*

Penamaan projek *database* untuk mempermudah *creator* mencari *database* sesuai dengan projek yang dirancang.

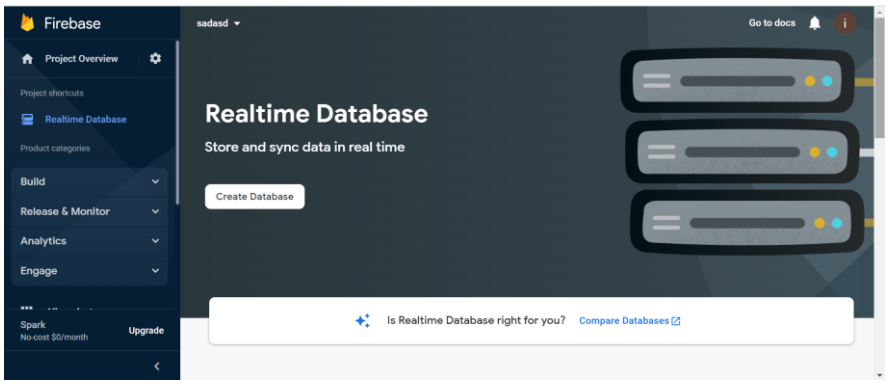


Gambar 3.13 Penamaan Projek Pada *Firebase*

Kemudian pada Gambar 3.13 merupakan pemberian nama projek *firebase* yang akan dibuat. Penamaan projek sangatlah penting dibuat sesuai dengan nama projek yang dirancang, agar creator dari database bisa mengetahui mana database yang sesuai dengan projek yang dirancang, agar tidak kesulitan dalam mencari.

3. Pembuatan *database*

Pada tahap pembuatan *database*, penulis menggunakan *realtime database* sebagai fitur yang digunakan.

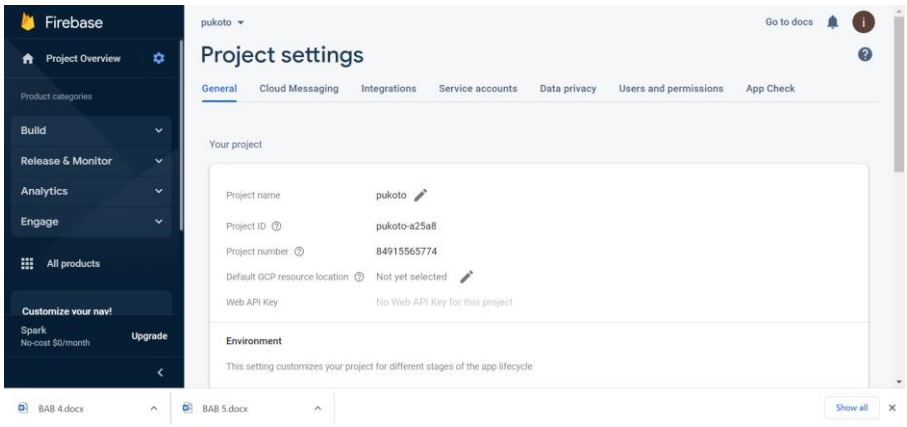


Gambar 3.14 Membuat *Realtime Database*

Gambar 3.14 menampilkan tampilan awal untuk pembuatan *realtime database* yang akan dirancang. Fungsi dari *realtime database* yaitu sebagai *database* yang membaca nilai *real* dengan keaadan dilapangan melalui sensor yang dikirimkan dengan modul *WiFi* yang terpasang di ESP32.

4. Seting *database*

Setingan projek berfungsi untuk melakukan konfigurasi sesuai dengan perancangan yang diharapkan oleh *creator*.

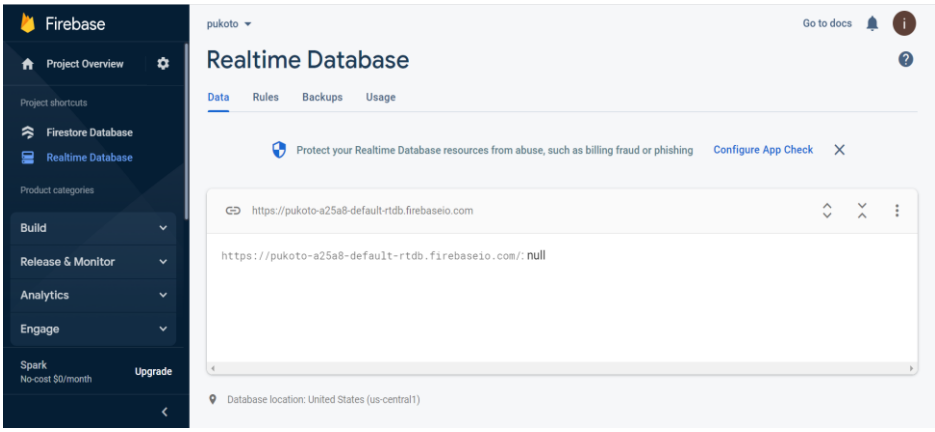


Gambar 3.15 Tampilan seting *firebase*

Pada Gambar 3.15 berisikan tampilan untuk melakukan konfigurasi dari projek yang dirancang di *firebase*. Konfigurasi projek berisikan sentingan untuk mengunakan mode terkunci dan tidak terkunci. Fungsi mode ini untuk membuat *database* menjadi *privat*, hanya orang yang mempunyai akses yang dapat menggunakan.

5. Tampilan *database*

Tampilan *database* setelah selesai perancangan.



Gambar 3.16 Tampilan *Database*

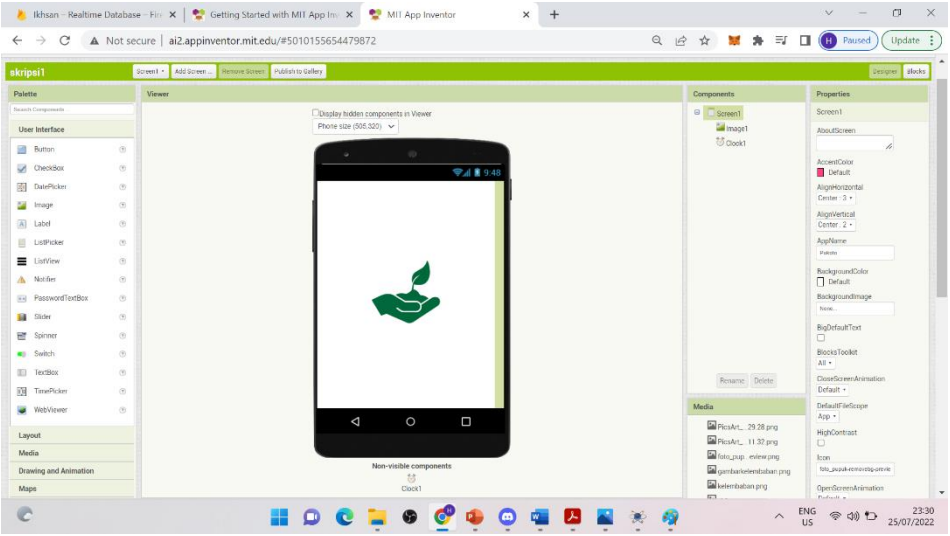
Pada Gambar 3.16 berisikan tampilan database setalah melakukan tahap-tahap perancangan *database*.

b. Perancangan MIT App *Inventor*

MIT App *Inventor* disini berfungsi sebagai *platform* pembantu untuk membuat aplikasi berbasis *android*. Berikut merupakan Langkah-langkah perancangan aplikasi menggunakan MIT App *Inventor*:

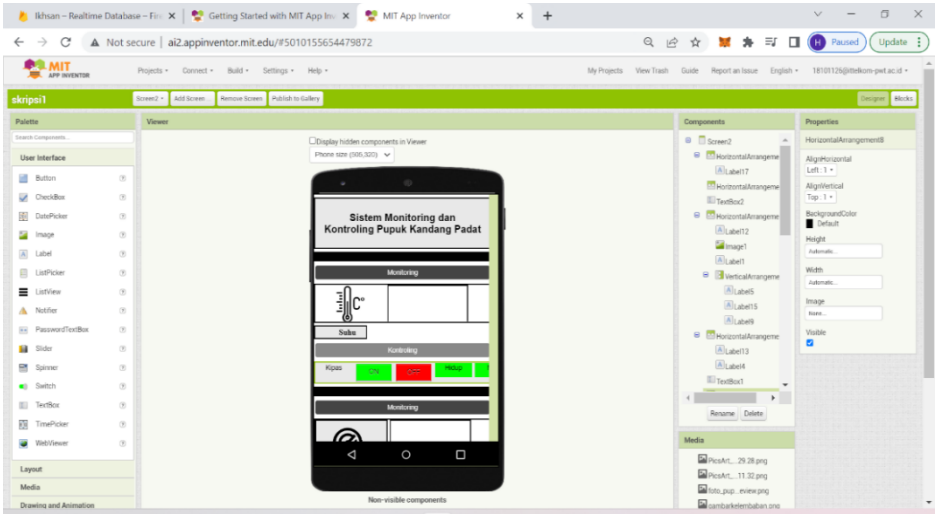
1. Tampilan *Designer* MIT App *Inventor*

Perancangan design pada MIT App *Inventor* bertujuan untuk membuat tampilan yang akan ditunjukan pada aplikasi.



Gambar 3.17 Tampilan *Designer* Pada *Screen* 1

Pada Gambar 3.17 merupakan tampilan *designer* pada *screen* 1, dimana desain aplikasi *android* yang telah dibuat terdapat *icon* serta nama aplikasi.



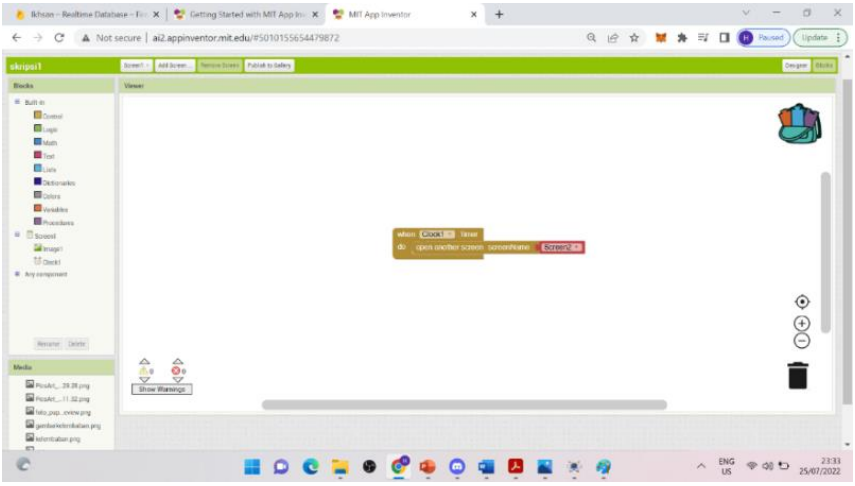
Gambar 3.18 Tampilan *Designer* Pada *Screen* 2

Pada Gambar 3.18 merupakan tampilan *designer screen* 2 yang telah dibuat, dimana pada *screen* 2 ini berfungsi sebagai monitoring kondisi pupuk

kandang padat. Pada *screen* 2 terdapat informasi mengenai menyalakan kipas angin dan pompa air secara manual dan otomatis, waktu dari *network time protocol* dan status bar yang berisi nilai suhu, serta kelembapan pupuk yang terbaca oleh sensor dan diperoleh dari *realtime google firebase*. Dan pada *screen* 2 digunakan untuk menampilkan sebuah notifikasi dari beberapa sensor antara lain sensor *dallas* DS18B20 dan sensor YL-69. Notifikasi berupa suhu, dan kelembapan pupuk yang akan dipantau pada sistem ini.

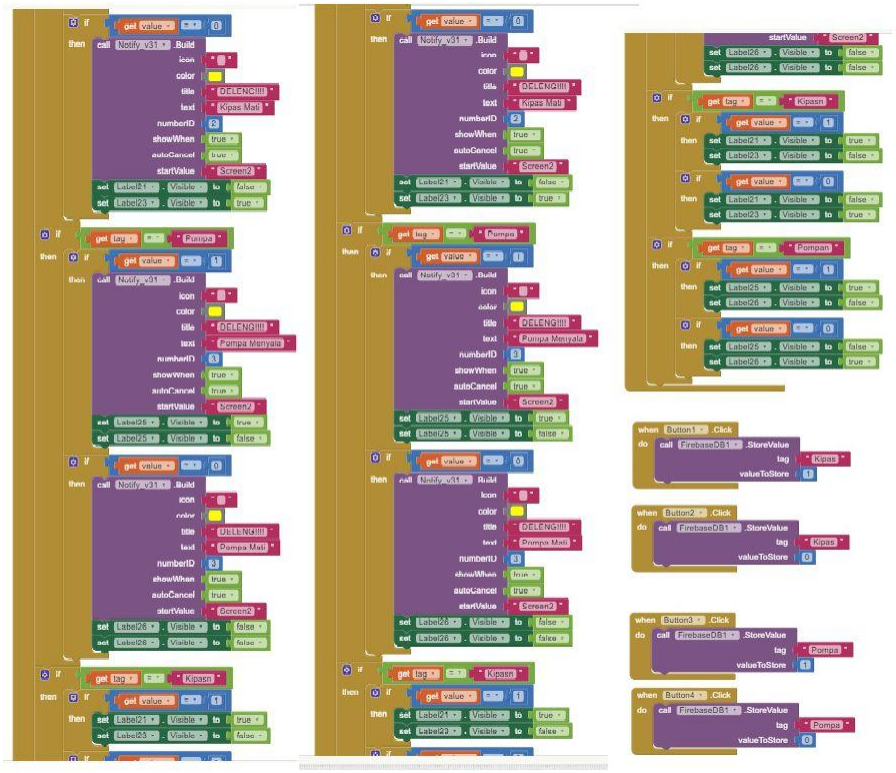
2. Tampilan blok program pada *MIT App Inventor*

Perancangan blok program pada MIT App *Inventor* bertujuan untuk membuat logika pemrograman agar aplikasi berjalan sesuai dengan logika *creator*.



Gambar 3.19 Tampilan *Blocks* Pada *Screen* 1

Pada Gambar 3.19 merupakan *block* dari *screen* 1 yang terdapat *clock*, dimana *clock* berfungsi untuk membuka *screen* 2 dengan durasi 2 detik.



Gambar 3.20 Tampilan *Block* Pada *Screen* 2

Pada Gambar 3.20 merupakan tampilan *block* pada *screen* 2 untuk mendapatkan nilai sensor suhu *dallas* DS18B20 dan nilai sensor kelembapan YL-69 dari *realtime database google firebase.* Membuat notifikasi peringatan untuk suhu serta kelembapan pupuk jika tidak sesuai dengan standar yang ditentukan dan notifikasi informasi waktu penyalaan kipas angin dan pompa air secara manual maupun otomatis.

3.5 PENGUJIAN SISTEM

Pengujian sistem dilakukan untuk mengetahui apakah sistem yang dirancang diimplementasikan atau tidak. Sistem dapat dinyatakan beroperasi degan benar apabila semua komponen yang digunakan dapat beroperasi sesuai dengan tujuan. Beberapa proses pengujian yang akan dilakukan sebagai berikut:

3.5.1 Pengujian Sensor DS18B20

Pada pengujian sensor DS18B20 dilakukan sebanyak 15 kali dengan rentang uji suhu 11°C sampai 65°C, pengujian akurasi sensor untuk mendapatkan nilai *error* dengan membandingkan nilai pengukuran sensor DS18B20 yang tertampil pada *serial* monitor Arduino IDE dengan nilai *actual* alat ukur. Besarnya nilai *error* yang diuji menggunakan rumus pada (2.1).

3.5.2 Pengujian Sensor YL69

Pada pengujian sensor YL-69 dilakukan sebanyak 15 kali dengan rentang uji kelmbapan aktual 0 sampai 64.4 % , pengujian akurasi sensor untuk mendapatkan nilai *error* dengan membandingkan nilai pengukuran sensor YL-69 yang tertampil pada *serial* monitor Arduino IDE dengan nilai *actual* alat ukur. Besarnya nilai *error* yang diuji menggunakan rumus pada (2.2).

3.5.3 Pengujian DC *Fan*

Pada pengujian DC *Fan* dilakukan sebanyak 10 kali, untuk menguji kecocokan program dengan perangkat *output* apakah program dapat menjalankan *Fan* atau tidak. Cara pengujian dilakukan dengan dua cara, yaitu secara manual menggunakan aplikasi yang berada pada *smartphone Android* dan secara otomatis menggunakan nilai keluaran dari sensor suhu yang dimana berada pada suhu yang diprogram.

3.5.4 Pengujian *Water Pump*

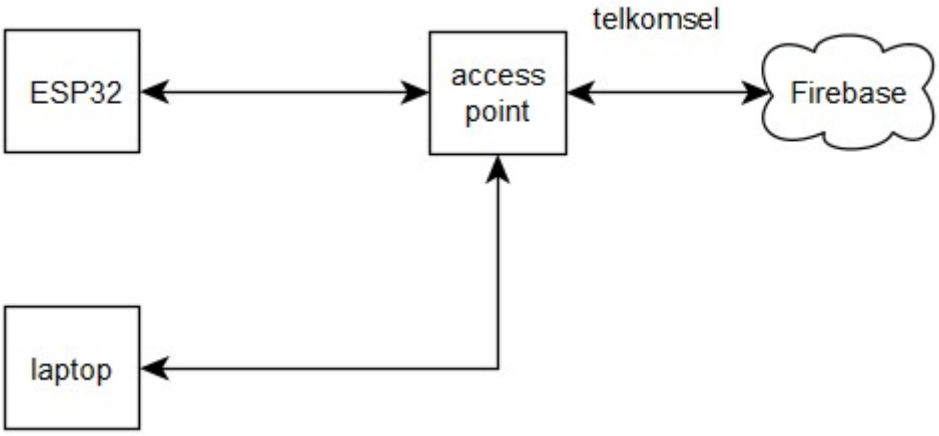
Pada pengujian *Water Pump* dilakukan sebanyak 10 kali, untuk menguji kecocokan program dengan perangkat *output* apakah program dapat menjalankan *water pump* atau tidak. Cara pengujian dilakukan dengan dua cara, yaitu secara manual menggunakan aplikasi yang berada pada *smartphone* Android dan secara otomatis menggunakan nilai keluaran dari sensor DS18B20 yang dimana berada pada kelembaban yang diprogram.

3.5.5 Pengujian Keseluruhan

Pada pengujian keseluruhan, penelitian ini menguji notifikasi dari aplikasi yang telah dirancang dan sistem monitoring yang telah dirancang untuk memonitoring suhu dan kelembaban pengkomposan pupuk kandang padat. Pengujian notifikasi meliputi pengujian tiap notifikasi yang telah dirancang sebanyak 30 kali. Pengujian monitoring meliputi pengujian notifikasi jika suhu tinggi, notifikasi jika kelembaban rendah, notifikasi jika pompa menyala, dan notifikasi jika kipas menyala. Pengujian selanjutnya yaitu pengujian monitoring suhu dan kelembaban pupuk kandang padat. Pada pengujian tersebut, peneliti melakukan pengujian dari jam 04.00 sampai 23.00. pengujian dilakukan setiap jamnya untuk melakukan pemantauan kondisi pengkomposan pupuk kandang padat.

3.5.6 Pengujian *Quality of Service* (QoS)

Pengujian parameter dari *Quality of Service* (QoS) digunakan untuk mengukur nilai dari kualitas jaringan tertentu pada suatu layanan. Pada pengujian QoS ini ada 3 parameter yang akan diujikan pada penelitian ini yaitu *delay*, *throughput* dan *packet loss* dengan menggunakan standar tiphon pada protokol HTTP. Untuk mendapatkan data nilai dari QoS saat pengujian menggunakan aplikasi *wireshark* sebagai media untuk menampilkan paket data. Pada pengujian QoS dilakukan, penelitian ini menggunakan waktu yang berbeda dengan pengiriman data yang sama. Untuk waktu yang digunakan selama 1 menit, 5 menit, 10 menit, 15 menit, 20 menit dan 25 menit pada jarak 2m dengan *obstacle* dinding tembok. Berikut merupakan skenario jaringan pada pengujian QoS.



Gambar 3.21 Skenario jaringan pengujian QoS