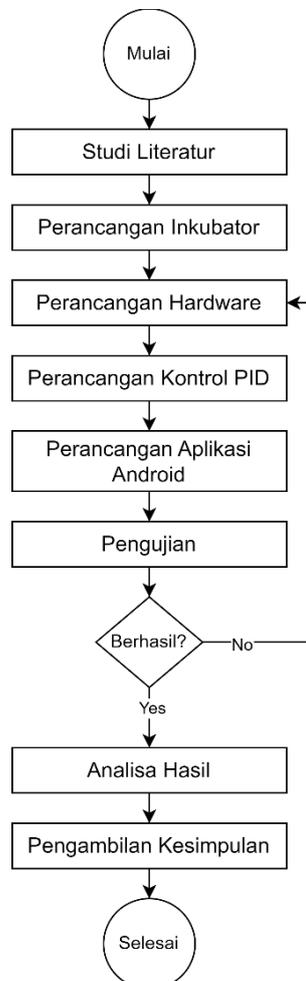


BAB III METODE PENELITIAN

3.1 Metode Penelitian

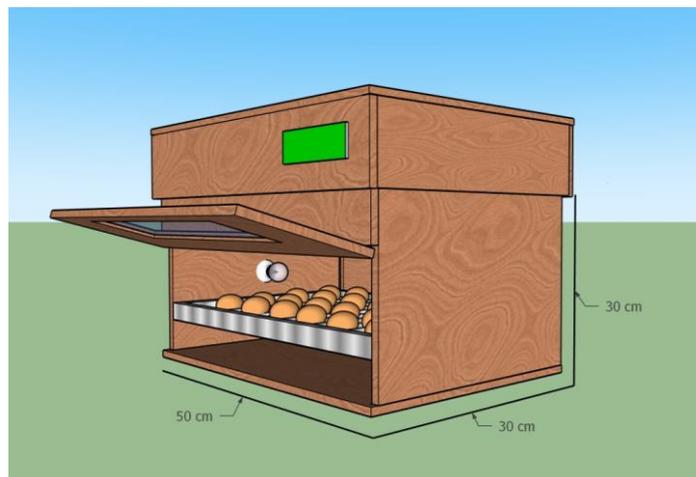
Metode yang diterapkan dalam pelaksanaan penelitian ini adalah metode eksperimen, tahapan awal dimulai dengan membuat rancangan terinci untuk membuat sistem yang diharapkan. Adapun sistem yang diharapkan adalah sebuah inkubator penetas telur berbasis *Internet of Thing* (IoT) dengan metode kontrol PID dan aplikasi android. Sistem ini terdiri dari sensor DHT11 sebagai sensor suhu dan kelembapan yang terintegrasi dengan ESP32, rangkaian lampu sebagai sumber pemanas dan rangkaian kipas yang digunakan sebagai pemerataan suhu pada inkubator. Pada Gambar 3.1 digambarkan diagram alur penelitian yang dilakukan.



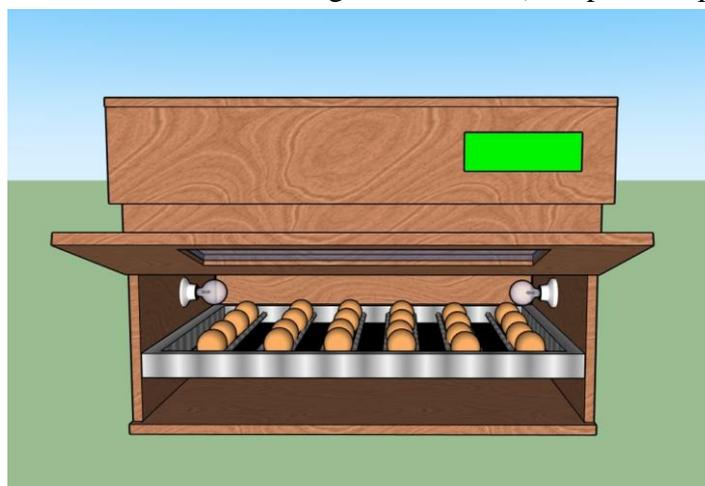
Gambar 3.1 Diagram Alur Penelitian

Pada tahapan awal melakukan studi literatur untuk mendapatkan informasi mengenai data yang diperlukan dalam penelitian ini, informasi tersebut yaitu nilai suhu yang dibutuhkan dalam penetasan telur, penggunaan sensor DHT11 sebagai masukan nilai suhu dan kelembapan, mikrokontroler esp32, penggunaan *synchron* motor dan penerapan PID untuk mengontrol suhu agar stabil sesuai target suhu untuk penetasan telur.

Kemudian tahapan berikutnya adalah perancangan inkubator menggunakan tripleks kayu sebagai bahan dasar pembuatan Box inkubator. Inkubator telur ini berukuran panjang 50 cm, lebar 30 cm dan tinggi 30 cm. Pada gambar 3.2 dan gambar 3.3 merupakan sketsa desain perancangan inkubator.



Gambar 3.2 Sketsa Perancangan Inkubator (Tampak Samping)



Gambar 3.3 Sketsa Perancangan Inkubator (Tampak Depan)

Pada tahapan selanjutnya dilakukan perancangan *hardware* yang akan digunakan pada inkubator, ESP32 akan berfungsi sebagai mikrokontroler dan

DHT11 akan berfungsi sebagai sensor suhu dan kelembapan. Mikrokontroler ESP32 akan mengontrol lampu pijar dan kipas pada sistem ini. Inkubator juga dilengkapi dengan 1 lampu pijar 25watt sebagai pemanas yang akan dikontrol menggunakan metode kontrol PID dan kipas pendingin 12volt sebanyak 2 buah digunakan untuk meratakan suhu pada inkubator. *Synchronous* motor juga digunakan untuk menggerakkan rak pada inkubator, sehingga memungkinkan untuk melakukan pemutaran telur secara otomatis. Selain itu, inkubator juga dilengkapi dengan kamera pengawas ESP32Cam, yang terintegrasi dengan aplikasi android, sehingga memungkinkan untuk melakukan pemantauan kondisi inkubator secara langsung.

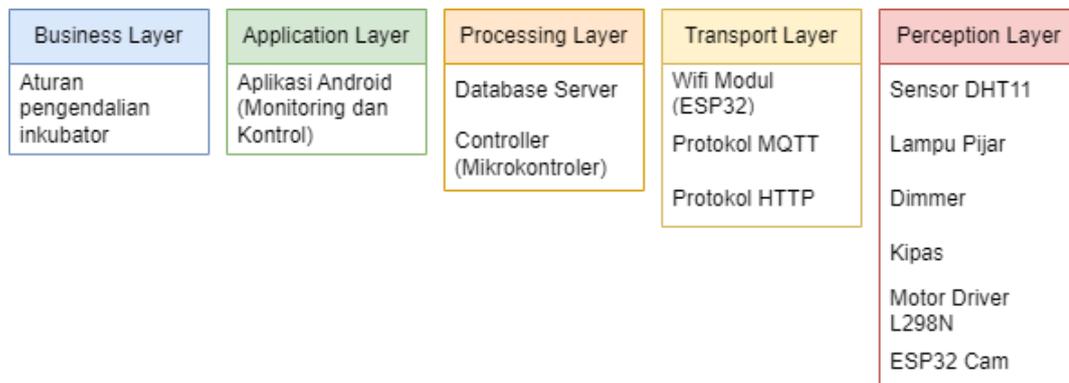
Tahapan selanjutnya yaitu merancang kontrol PID, tujuan dari perancangan kontrol PID untuk mengontrol suhu inkubator, memastikan bahwa suhu tetap stabil dan sesuai dengan target suhu yang akan diinginkan. PID merupakan singkatan dari *Proportional Integral* dan *Derivative* yang memiliki tiga komponen. Komponen *proporsional* (P) berfungsi mengurangi kesalahan berdasarkan besarnya kesalahan saat ini, komponen *integral* (I) berfungsi mengurangi kesalahan kumulatif dari waktu ke waktu, dan komponen *derivatif* (D) mengantisipasi kesalahan yang akan datang dengan mempertimbangkan laju perubahan kesalahan. Jika digunakan pada inkubator, sensor suhu akan mencatat nilai suhu sebenarnya dan mengirimkannya ke pengendali PID. Pengendali PID kemudian menghitung sinyal kontrol berdasarkan ketiga komponen tersebut. Sinyal kontrol ini mengontrol elemen pemanas untuk menjaga suhu inkubator pada tingkat yang diinginkan.

Selanjutnya tahapan perancangan aplikasi android, pada tahapan ini aplikasi Android sebagai media untuk memantau suhu dan kelembapan dalam inkubator agar proses pemantauan lebih mudah dan efektif. Sensor yang terpasang di dalam inkubator untuk mengukur suhu dan kelembapan akan terintegrasi dengan aplikasi ini. Sensor akan mengirimkan data yang dikumpulkannya ke *database* online yang nantinya akan diakses oleh aplikasi android menggunakan API. Aplikasi ini memiliki antarmuka yang menampilkan data suhu dan kelembapan dalam bentuk grafik yang mudah dipahami. Kemudian terdapat juga fitur *live cam* yang

terintegrasi dengan ESP32 cam sehingga dalam melakukan pemantauan langsung kondisi pada inkubator.

Pada tahap akhir proses perancangan, selanjutnya dilakukan tahap uji coba. pada tahap ini sistem diuji untuk mengukur suhu dan kelembapan, memantau situasi inkubator dengan kamera melalui aplikasi android, dan menguji *synchronous* motor agar bisa memutar telur sesuai dengan *timer*. kemudian data pengukuran akan dikirimkan ke *server database*. Lalu hasil data yang terdapat pada *server* akan diintegrasikan dengan aplikasi Android.

Dalam proses perancangan *Internet of Things* (IoT), terdapat berbagai ketentuan yang harus dipenuhi untuk memastikan sistem berjalan dengan baik, termasuk pemahaman yang mendalam tentang lapisan-lapisan yang ada dalam arsitektur IoT. Pada contoh inkubator penetas telur berbasis IoT yang menggunakan kontrol PID dan aplikasi Android, struktur lapisan IoT memainkan peran penting dalam memastikan integrasi dan fungsionalitas yang optimal. Struktur Lapisan IoT Inkubator dapat dilihat pada gambar 3.4.



Gambar 3.4 Struktur Lapisan IoT Inkubator Penetas Telur

Gambar 3.4 Struktur Lapisan IoT Inkubator di atas menjelaskan berbagai lapisan dalam arsitektur IoT inkubator penetas telur. Setiap lapisan memiliki peran penting dalam memastikan sistem berjalan dengan efisien dan efektif. Bagian *Business Layer* pada arsitektur IoT inkubator penetas telur bertanggung jawab pada bagian pengelolaan aturan operasional sistem. Protokol dan aturan pengendalian inkubator akan mengatur berbagai aspek operasional seperti pengendalian suhu inkubator melalui kontrol PID dan penjadwalan penggerak telur agar dapat memutar telur secara otomatis.

Lapisan *Application Layer* mencakup aplikasi Android yang digunakan untuk *monitoring* dan kontrol inkubator. Aplikasi ini memungkinkan pengguna untuk memantau kondisi suhu dan kelembaban secara berkala, serta memberikan kontrol penuh untuk mengatur sistem *on* dan *off* PID pada sistem inkubator, serta mengatur nilai PWM lampu pijar dan kipas. Aplikasi ini juga menyajikan data historis yang dapat digunakan untuk analisis lebih lanjut berupa grafik suhu dan kelembaban yang diatur berdasarkan tampilan harian dan bulanan.

Kemudian pada lapisan *Processing Layer* terdapat dua bagian proses pada inkubator yang terdiri dari *server database* dan *controller* (mikrokontroler dengan PID). *Server database* akan menyimpan data yang dikumpulkan dari sensor DHT11, seperti suhu dan kelembaban, Intensitas nilai PWM pada lampu dan kipas, serta status *relay* pada *synchronous* motor. *Controller* bertugas menjalankan algoritma PID untuk menjaga suhu dalam inkubator sesuai dengan *set point* yang ditentukan. *Controller* ini menerima *input* dari sensor suhu DHT11 dan menyesuaikan *output* ke lampu pijar untuk memastikan suhu tetap stabil.

Network Layer pada inkubator mencakup modul jaringan Wi-Fi (ESP32) yang menghubungkan sistem inkubator dengan internet. Protokol MQTT juga digunakan untuk mengirim dan menerima perintah kontrol *On/Off* untuk sistem kendali PID, memungkinkan pengguna untuk mengontrol sistem secara langsung dari aplikasi Android. Selain itu, protokol HTTP digunakan untuk pengiriman data dari mikrokontroler ke *server*, memastikan bahwa data suhu dan kelembaban yang diukur oleh sensor DHT11 dapat dipantau dan disimpan di *server database*.

Pada inkubator bagian *Perception Layer* terdiri dari sensor DHT11 yang berfungsi mengukur suhu dan kelembaban, lampu pijar yang berfungsi sebagai sumber utama untuk menghasilkan panas, kipas yang membantu meratakan suhu, dan *synchronous* motor yang menggerakkan rak telur. Sensor DHT11 memberikan *feedback* suhu ke mikrokontroler, yang kemudian menggunakan informasi ini untuk menyesuaikan *output* ke lampu pijar dan memastikan suhu dalam inkubator sesuai dengan *set point* yang diinginkan. Kipas membantu menjaga distribusi suhu yang merata pada setiap ruangan inkubator, sementara *synchronous* motor

menggerakkan rak telur secara otomatis sesuai waktu penjadwalan untuk memastikan setiap telur mendapatkan panas yang cukup.

3.2 Perancangan *Hardware*

Perancangan *hardware* inkubator telur difungsikan agar dapat menghasilkan suhu yang ideal dan stabil untuk proses penetasan telur. Inkubator sendiri terdiri dari beberapa komponen seperti mikrokontroler ESP32, sensor DHT11, lampu pijar, kipas, *synchronous* motor dan, kamera ESP32. Berikut merupakan beberapa tahapan perancangan *hardware* pada inkubator.

3.2.1 Perangkat Penunjang

Perangkat penunjang penelitian terdiri dari dua bagian yaitu, perangkat keras dan perangkat lunak. Berikut ini adalah perangkat keras yang digunakan:

1. Laptop
2. *Smartphone*
3. ESP32
4. ESP32 Cam
5. Sensor DHT11
6. Kabel Jumper
7. Lampu Pijar
8. Kipas
9. *Breadboard*
10. *Synchron* Motor
11. Dimer
12. *Driver* Motor L298N
13. LCD 16x2
14. *Relay*
15. *Step Up*

Kemudian untuk perangkat lunak yang digunakan adalah sebagai berikut:

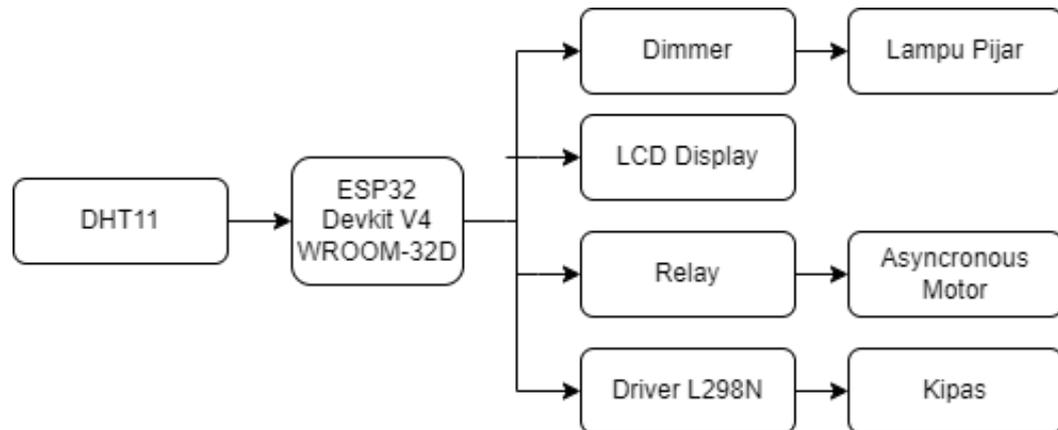
1. Visual Studio Code (PlatformIo)
2. Google Chrome (*Browser*)
3. Android Studio
4. Arduino IDE

5. Matlab

6. Figma

3.2.2 Design Sistem Inkubator Penetas Telur

Blok diagram inkubator terdiri dari beberapa komponen penting yang saling berhubungan untuk menciptakan lingkungan dengan suhu yang stabil dan ideal. Blok diagram inkubator telur dapat dilihat pada gambar 3.5.



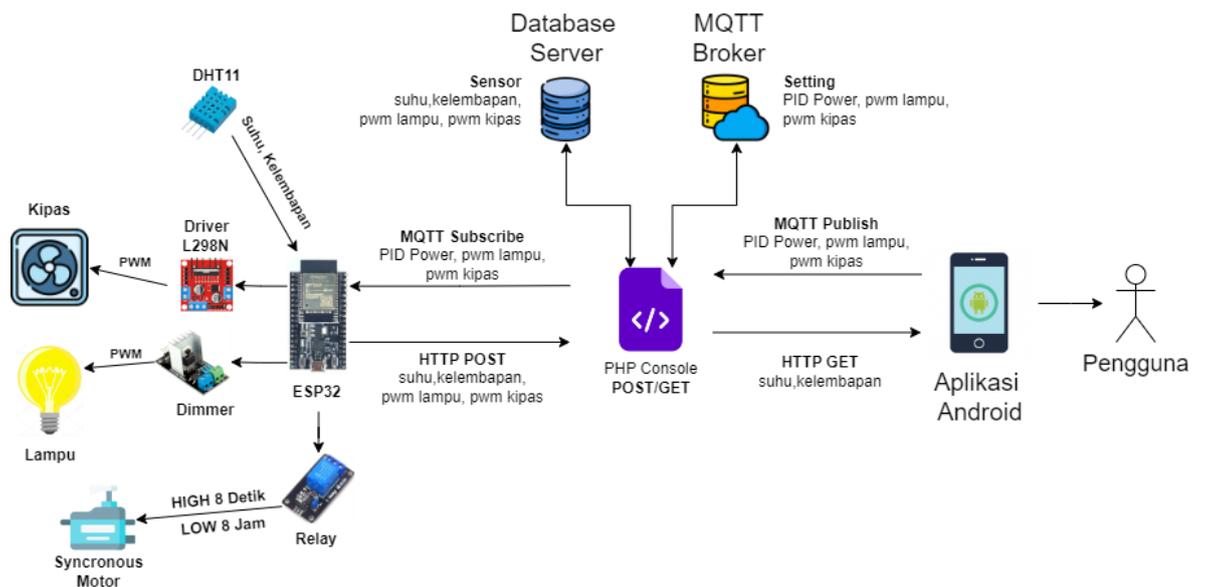
Gambar 3.5 Blok Diagram Inkubator Penetas Telur

Blok Diagram yang ditunjukkan pada Gambar 3.5 menjelaskan bahwa inkubator terdiri dari beberapa komponen utama yang terintegrasi dengan baik untuk mendukung fungsi penetasan telur. Komponen-komponen utama dalam inkubator ini meliputi sistem pemanas, sensor suhu dan kelembapan, mikrokontroler, kipas, dan *synchronous* motor. Sistem pemanas pada inkubator menggunakan lampu pijar yang berfungsi untuk menjaga suhu pada tingkat yang diinginkan secara stabil dan konsisten. Penggunaan kipas dalam inkubator sangat penting karena kipas bertanggung jawab untuk mengalirkan sirkulasi udara dan panas di dalam inkubator. Kipas juga berperan dalam menjaga keseimbangan atau kestabilan suhu, sehingga kondisi suhu yang ideal dapat dipertahankan untuk penetasan telur.

Sensor suhu dan kelembapan juga merupakan komponen penting dalam inkubator ini. Sensor DHT11 yang digunakan berfungsi untuk mengukur suhu dan kelembapan di dalam inkubator. Data yang diperoleh oleh sensor tersebut kemudian akan diolah lebih lanjut oleh mikrokontroler. Mikrokontroler yang digunakan adalah ESP32 yang bertanggung jawab untuk mengatur semua proses yang terjadi,

termasuk pengendalian suhu, kelembapan, dan sirkulasi udara, sehingga memastikan kondisi lingkungan yang optimal untuk penetasan telur. Integrasi yang baik antara komponen-komponen ini sangat penting untuk keberhasilan proses penetasan telur yang efektif dan efisien.

Arsitektur Diagram inkubator penetas telur yang terintegrasi *Internet of Things* dapat dilihat pada gambar 3.6 berikut.

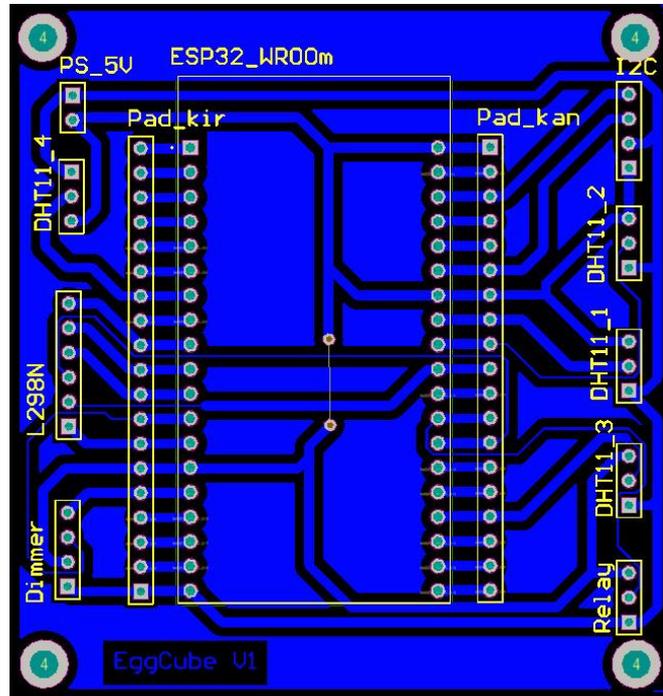


Gambar 3.6 Arsitektur Diagram

Gambar 3.6 arsitektur diagram menunjukkan sensor DHT11 berfungsi sebagai *input* utama yang akan menghasilkan data pembacaan suhu yang akan diproses oleh mikrokontroler ESP32, data suhu tersebut akan diolah menggunakan kontrol PID untuk menghasilkan keluaran berupa sinyal PWM. Setelah data suhu diproses dan menghasilkan sinyal PWM, nilai Suhu dan sinyal PWM tersebut juga disimpan pada *database server* agar dapat melakukan *monitoring* melalui aplikasi android dan analisis terhadap kondisi inkubator. Inkubator yang terintegrasi dengan internet akan mengirimkan data ke *database server* melalui protokol HTTP, aplikasi android yang dirancang memungkinkan untuk mengakses data pada *database server* untuk melihat data inkubator pada proses *monitoring*.

Mikrokontroler akan dirancang untuk melakukan tugas khusus yang diprogram dengan tujuan utama menjaga suhu inkubator agar tetap ideal dan stabil sesuai dengan target suhu yang diperlukan untuk penetasan telur. Dengan adanya

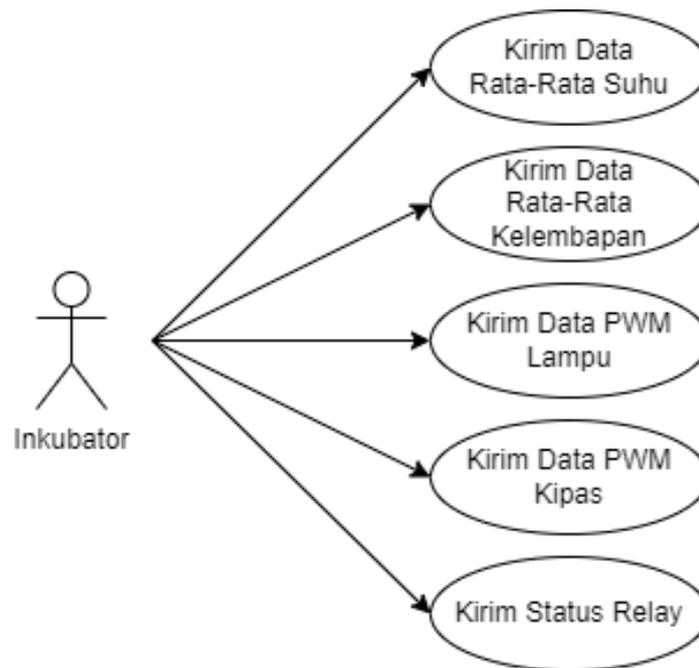
penerapan mikrokontroler ini, inkubator mampu beroperasi secara otomatis sesuai dengan instruksi program yang telah ditentukan, yaitu menjaga kestabilan suhu serta mengirimkan data pembacaan sensor suhu dan kelembapan ke *server database*.



Gambar 3.7 Design PCB

Desain PCB dari sistem inkubator juga dapat diamati melalui gambar 3.7, yang menunjukkan tata letak dan koneksi antar komponen untuk memastikan kinerja yang optimal. Dalam gambar tersebut, dapat terlihat bagaimana setiap komponen dihubungkan secara strategis untuk meminimalkan gangguan.

Untuk proses pengiriman data pada inkubator ke *database server*, terdapat *use case* diagram yang dapat menggambarkan interaksi antara inkubator dan *server*. Pengiriman data dilakukan setiap interval selama 20 detik sekali. *Use case* diagram inkubator dapat dilihat pada gambar 3.8.

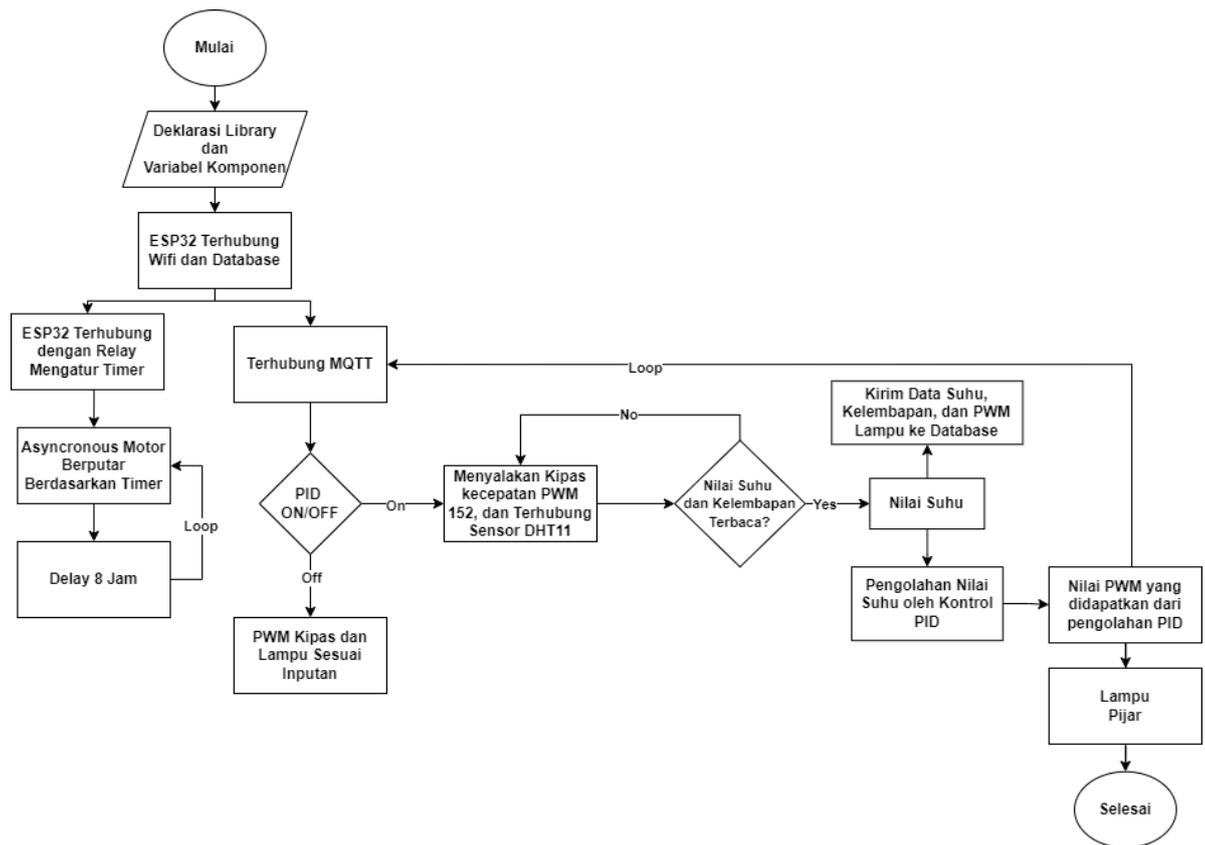


Gambar 3.8 Use Case Diagram Inkubator

Pada gambar 3.8, *use case diagram* inkubator menggambarkan bahwa aktor utama dalam proses ini adalah sistem inkubator itu sendiri. Sistem ini secara otomatis bertanggung jawab untuk mengumpulkan data dari sensor DHT11 yang telah dipasang di dalamnya. Data yang dikumpulkan mencakup informasi penting seperti suhu, kelembapan, PWM lampu, PWM kipas dan status *relay*. Setiap 20 detik, sistem inkubator akan mengirimkan data tersebut ke *server* menggunakan protokol komunikasi HTTP. *Server* yang menerima data ini kemudian akan menyimpannya dalam *database* yang telah disiapkan. Data yang disimpan dalam *database* ini bukan hanya untuk *monitoring*, tetapi juga berfungsi sebagai catatan historis yang dapat digunakan untuk analisis lebih lanjut.

3.2.3 Perinsip Kerja Inkubator Penetas Telur

Proses kerja sistem inkubator secara menyeluruh, mulai dari pengumpulan data, proses pengendalian suhu, hingga tindakan sistem untuk menjaga kondisi suhu yang ideal dan stabil dapat dilihat pada gambar 3.9.



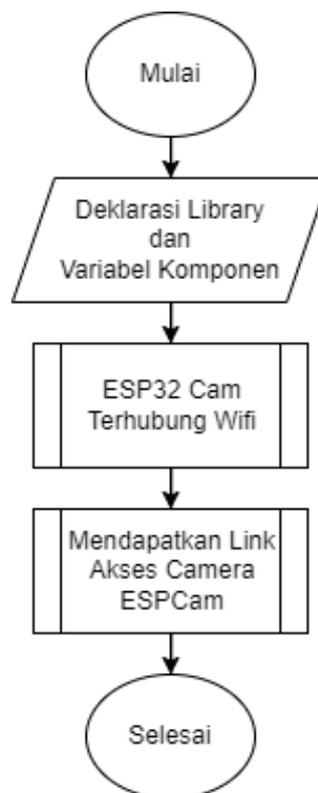
Gambar 3.9 Diagram Alir Perangkat Keras Inkubator

Pada gambar 3.9 diagram alir perangkat keras, terdapat 2 proses yang terjadi pada sistem, tahapan pertama mikrokontroler akan terhubung dengan WiFi, membaca *library* dan variabel, serta konfigurasi yang digunakan untuk melakukan perintah yang diterima oleh mikrokontroler. Mikrokontroler akan melakukan pengecekan secara berulang terhadap protokol MQTT untuk memastikan sistem kendali PID *ON* atau *OFF*, apabila nilai MQTT yang didapatkan *OFF* maka kipas dan lampu akan menyala sesuai dengan masukan PWM yang dikirim oleh pengguna ke *database server*, namun apabila sistem kendali *ON* maka kipas akan menyala dengan kecepatan yang konstan dengan besaran PWM sebesar 255 dan PWM lampu akan dikendalikan oleh PID sesuai kebutuhan dan bacaan dari sensor, selanjutnya sensor DHT11 akan menghasilkan data *input* nilai pembacaan suhu, apabila proses pembacaan suhu tidak dapat terbaca maka sistem akan otomatis melakukan pengulangan proses dari awal untuk melakukan pengecekan kondisi. Data nilai suhu tersebut akan dilakukan pengolahan oleh kontrol PID secara otomatis pada mikrokontroler, hasil dari pengolahan data nilai suhu menggunakan

kontrol PID akan menghasilkan keluaran berupa nilai PWM yang akan digunakan untuk menghidupkan lampu pijar sebagai sumber pemanas. kemudian data suhu tersebut juga akan ditampilkan pada LCD *display* dan dikirimkan pada *database server* dengan data lainnya seperti kelembapan dan PWM lampu. Proses ini akan dilakukan secara terus menerus hingga menghasilkan suhu yang sesuai dengan *set point* target secara ideal dan stabil.

Pada bagian kontrol *synchronous* motor terdapat *relay* yang terhubung dengan mikrokontroler yang berfungsi mengatur tegangan masuk dan keluar untuk menggerakkan *synchronous* motor sesuai dengan waktu yang telah ditetapkan pada program mikrokontroler, *synchronous* motor akan berputar selama 8 detik untuk menggerakkan rak telur, sehingga memungkinkan telur untuk berputar secara otomatis.

Kemudian pada perangkat keras lainnya, terdapat juga prinsip kerja ESP32 cam yang digunakan sebagai *monitoring* kamera pengawas yang dapat dilihat pada gambar 3.10.

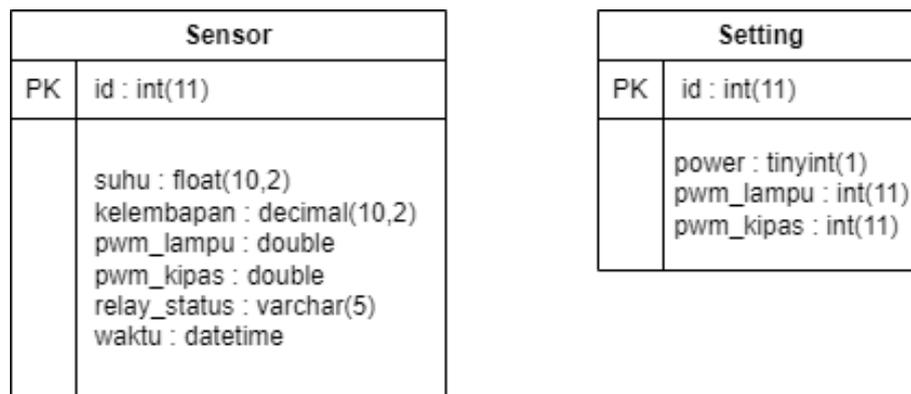


Gambar 3.10 Diagram Alir ESP32 Cam

Berdasarkan gambar 3.10 diagram alir ESP32 Cam, tahapan pertama yang dilakukan oleh sistem adalah proses deklarasi *library* dan variabel yang akan digunakan untuk integrasi pada komponen, kemudian ESP32 Cam akan terhubung dengan WiFi berdasarkan variabel SSID dan *Password* yang telah ditentukan pada kode program. Apabila ESP32 Cam telah terhubung dengan WiFi, maka pada bagian *serial monitor* akan terdapat *link url* yang dapat digunakan untuk mengakses kamera. *Link url* tersebut dapat digunakan pada web *browser* dan juga aplikasi android untuk melakukan *streaming video*.

3.2.4 Database Server

Database server dirancang pada sistem inkubator adalah untuk menyimpan dan mengelola data penting yang dikumpulkan dari berbagai kontroler dan sensor yang terpasang di dalamnya. MySQL adalah *platform database* yang dipilih untuk perancangan ini karena stabilitasnya dan kemudahan integrasinya dengan berbagai bahasa pemrograman. Semua data yang diperlukan disimpan dalam satu tabel utama dalam skema *database*. Diagram *database server* dapat dilihat pada gambar 3.11.



Gambar 3.11 Diagram Database Server

Pada gambar 3.11 diagram *database* sensor terdapat tabel sensor berfungsi untuk menampung dan menyimpan data yang didapatkan dari sensor pada inkubator, data yang terdapat pada tabel sensor sendiri terdiri dari 'id' untuk identifikasi setiap data yang masuk pada *database server*, 'suhu' untuk menyimpan nilai suhu, 'kelembapan' untuk menyimpan nilai kelembapan, 'pwm_kipas' untuk menyimpan nilai PWM yang digunakan untuk menyalakan kipas, 'pwm_lampu'

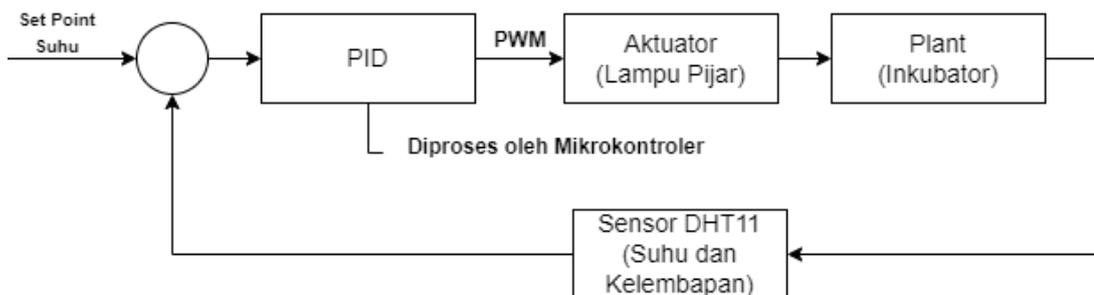
digunakan untuk menyimpan nilai PWM lampu, 'relay_status' untuk menyimpan data status *relay* dan 'waktu' menyimpan data waktu masuk setiap data yang ada.

Selanjutnya tabel *Setting* digunakan untuk menyimpan data dari aplikasi android berdasarkan *input* data dari pengguna. Data yang terdapat pada tabel *setting* adalah 'id' untuk identifikasi setiap data yang masuk pada *database server*, 'power' untuk menyimpan nilai status kendali PID berupa nilai *boolean*, 'pwm_lampu' menyimpan nilai PWM untuk menyalakan lampu, dan 'pwm_kipas' berfungsi untuk menyimpan data untuk menyalakan kipas.

Dengan adanya perancangan sistem *database* untuk inkubator dan aplikasi android ini diharapkan dapat beroperasi dengan optimal, menyediakan data yang akurat dan mudah diakses untuk analisis lebih lanjut.

3.3 Perancangan Kontrol PID

Perancangan kontrol PID terdiri dari beberapa ketentuan dalam proses pengerjaannya, salah satu hal yang perlu diperhatikan adalah bagaimana kontrol PID dapat bekerja sesuai fungsi dan mendapatkan hasil tujuan yang diharapkan, pada gambar 3.13 terdapat blok diagram kontrol PID.



Gambar 3.12 Blok Diagram Kontrol PID

Pada Gambar 3.12, ditampilkan blok diagram kontrol PID yang digunakan dalam inkubator penetas telur. Proses penetasan telur dimulai dengan pengaturan titik suhu target, atau dikenal sebagai *set point*, yang diinginkan untuk mencapai kondisi ideal dalam inkubator. Sistem kontrol menggunakan *set point* ini sebagai referensi utama untuk memastikan suhu dalam inkubator tetap stabil dan sesuai dengan yang diperlukan. Dalam blok kontrol PID, terdapat mekanisme yang bertugas untuk mengontrol suhu dengan cara mengubah intensitas cahaya dari lampu pijar. Intensitas lampu pijar diatur menggunakan tegangan PWM (*Pulse*

Width Modulation) yang dihasilkan dari kalkulasi kontrol PID. Hal ini bertujuan untuk memastikan bahwa suhu dalam inkubator sesuai dengan titik *set point* yang telah ditentukan sebelumnya.

Untuk menghasilkan panas yang dibutuhkan dalam proses penetasan, lampu pijar berperan sebagai aktuator utama. Lampu pijar ini dipasang di dalam inkubator, tempat dimana telur-telur disimpan selama proses penetasan. Selain itu, sistem ini juga dilengkapi dengan sensor DHT11 yang berfungsi untuk mengukur suhu aktual di dalam inkubator. Sensor DHT11 memberikan *feedback* berupa nilai suhu yang terukur kepada blok kontrol PID. *Feedback* ini sangat penting karena blok kontrol PID akan membandingkan nilai suhu aktual yang diterima dari sensor dengan *set point* yang telah ditetapkan. Jika terdapat perbedaan antara suhu aktual dan *set point*, kontrol PID akan melakukan penyesuaian dengan mengubah intensitas lampu pijar melalui tegangan PWM.

Agar kontrol PID yang dirancang memiliki keakuratan yang sesuai dengan implementasi pada inkubator, perlu dilakukannya penentuan nilai *transfer function* terlebih dahulu. Penentuan nilai *transfer function* ini dilakukan menggunakan *System Identification Toolbox* yang merupakan salah satu fitur pada aplikasi Matlab. Berbagai metode disediakan dalam *system identification toolbox* ini untuk mengidentifikasi model matematis sistem dari data *input* dan *output* berdasarkan data yang telah dikumpulkan. Langkah pertama dalam proses ini adalah mengumpulkan data yang mencakup respons sistem terhadap berbagai sinyal uji, nantinya lampu pijar pada inkubator akan diberikan beberapa tegangan sinyal PWM sebagai uji sistem untuk mendapatkan data awal yang nantinya akan digunakan untuk menentukan nilai *transfer function*. Data yang telah didapatkan tersebut akan di *input* pada Matlab untuk diolah dan dianalisis menggunakan fungsi pada *System Identification Toolbox*. Identifikasi ini menghasilkan *transfer function* yang secara akurat menunjukkan dinamika terhadap sistem inkubator yang sebenarnya. Dengan menggunakan MATLAB dan *toolbox* ini, proses identifikasi menjadi lebih terstruktur dan terotomatisasi, yang memungkinkan peneliti untuk mendapatkan model yang presisi.

Setelah nilai *transfer function* dari sistem inkubator diperoleh, langkah berikutnya adalah menentukan nilai parameter PID, salah satu cara menentukan parameter kontrol PID yaitu menggunakan metode pendekatan FOPDT (*First Order Plus Dead Time*) untuk menentukan nilai parameter K_p , K_i , dan K_d . Metode ini banyak digunakan karena mudah dan hampir mendekati sistem yang kompleks.

FOPDT adalah suatu metode pendekatan yang banyak digunakan untuk menentukan parameter yang akan diterapkan pada kontroler jenis PID. Pada metode FOPDT (*First Order Plus Dead Time*) terdapat parameter waktu yang dibutuhkan *output* untuk mencapai 63.2% dari *peak overshoot response system*. Kemudian terdapat komponen parameter θ sebagai nilai *dead time* yang berarti waktu yang dibutuhkan oleh *output* untuk mulai berubah (Arfittariah., 2023). Berdasarkan percobaan yang dilakukan oleh Hedengren pada *Apmonitor.com* (2020) dengan menggunakan metode pendekatan yang serupa yaitu FOPDT, langkah-langkahnya dimulai dengan mengidentifikasi perubahan *output* (Δy) sebagai perbedaan antara nilai *steady-state* awal dan *steady-state* akhir *output*, perubahan *input* (Δu) sebagai perbedaan antara nilai *steady-state* awal dan *steady-state* akhir *input*. Setelah Mengetahui nilai perubahan *input* dan *output*, maka tahap selanjutnya menghitung *steady-state gain* (K_p) dengan membagi perubahan *output* (Δy) dengan perubahan *input* (Δu) seperti pada persamaan 1.

$$K_p = \left(\frac{\Delta y}{\Delta u} \right) \dots \dots \dots (1)$$

Kemudian menentukan waktu tunda (θ_p) dengan cara mengidentifikasi perubahan antara waktu saat *input* dan *output* mulai menunjukkan perubahan yang signifikan, ini bisa dilihat pada plot data *step response*. Langkah selanjutnya identifikasi Grafik untuk menentukan waktu dimana *output* mencapai 63.2% dari Δy , sesuai dengan persamaan 2.

$$0.632 \times \Delta y \dots \dots \dots (2)$$

Setelah mendapatkan nilai $0.632\Delta y$ menggunakan persamaan 2, langkah selanjutnya perlu menentukan terlebih dahulu *output* berdasarkan $t_{0.632}$ di mana nilai *output* mencapai 63.2% dari total perubahan *steady-state*. Untuk menentukan *output* ketika mencapai 63.2% dapat menggunakan persamaan 3.

$$y(t_{0.632}) = \text{Nilai Awal Suhu} + 0.632\Delta y \dots (3)$$

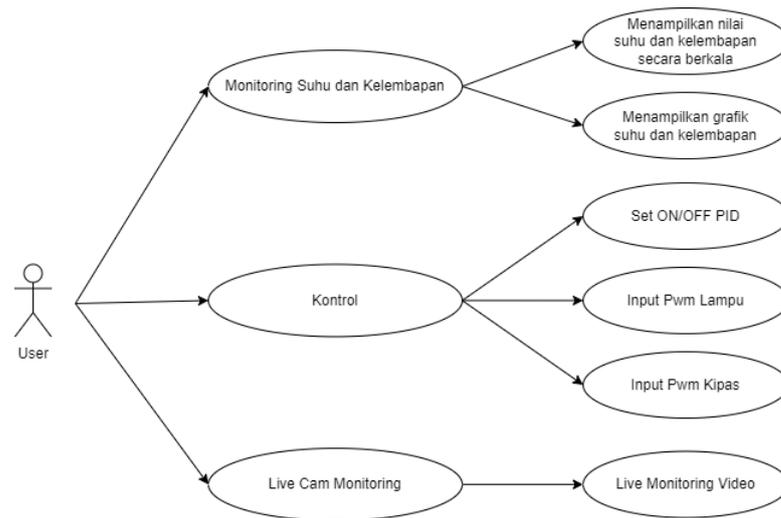
setelah menentukan nilai *output* $y(t_{0.632})$, maka langkah selanjutnya yaitu menentukan konstanta waktu (T) dengan cara mengukur waktu yang dibutuhkan oleh *output* untuk mencapai sekitar nilai $y(t_{0.632})$ dari total perubahan nilai awal menuju nilai *steady-state* setelah waktu tunda. langkah terakhir adalah menentukan kalkulasi nilai T_p yang didapatkan dengan cara mengurangi antara nilai waktu (T) dengan θ_p seperti pada persamaan 4.

$$T_p = (T) - \theta_p \dots (4)$$

Dari hasil analisis model FOPDT dengan data respons yang telah dilakukan sebelumnya maka akan didapatkan nilai K_p , T_p dan θ_p yang nantinya akan berfungsi sebagai parameter K_p , K_i dan K_d pada kontrol PID. Untuk memastikan kebutuhan parameter sesuai dengan hasil dan prediksi yang lebih akurat, diperlukannya tuning parameter K_p , K_i dan K_d terhadap respon *set point* target, agar dapat menyesuaikan dengan kebutuhan dan hasil yang diharapkan.

3.4 Perancangan Aplikasi Android

Proses perancangan aplikasi Android memerlukan sebuah *Use Case Diagram* yang menjelaskan interaksi antara pengguna dengan berbagai fitur utama aplikasi. Diagram ini terdiri dari beberapa fitur yang menjelaskan secara rinci bagaimana pengguna berinteraksi dengan setiap bagian dari aplikasi tersebut. Aktor utama dalam diagram ini adalah pengguna (*user*), yang memiliki kemampuan untuk mengakses tiga fitur utama. *Use Case Diagram* untuk aplikasi Android ini dapat dilihat pada gambar 3.13.



Gambar 3.13 Use Case Diagram Aplikasi Android

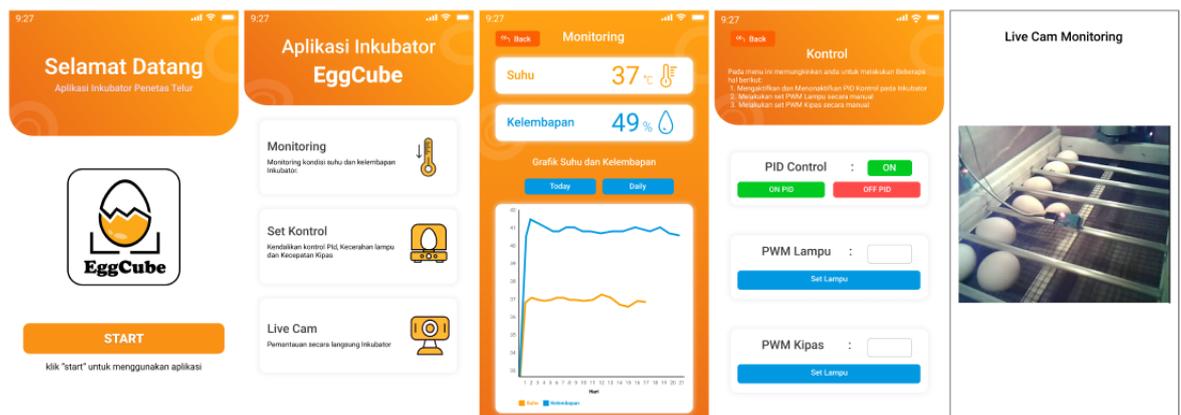
Pada gambar 3.14 *use case* diagram aplikasi android dirancang memiliki beberapa fitur seperti *monitoring*, kontrol dan *live cam monitoring* sesuai fungsi dan kebutuhannya. Fitur *monitoring* berfungsi untuk menunjukkan kondisi inkubator berupa suhu dan kelembapan yang ditampilkan secara berkala. Data ini dikumpulkan dari sensor yang terhubung dengan inkubator dan disimpan dalam *database server*. Dengan fitur ini, pengguna dapat dengan mudah untuk memastikan kondisi sesuai dengan suhu yang diperlukan untuk proses penetasan telur.

Fitur kontrol memungkinkan pengguna untuk mengatur berbagai parameter yang ada pada inkubator. Pengguna dapat mengatur sistem PID (*Proportional-Integral-Derivative*) berupa *ON* atau *OFF* untuk mengontrol suhu secara otomatis dan juga dapat mengatur nilai PWM (*Pulse Width Modulation*) untuk lampu dan kipas. Mengatur nilai PWM memungkinkan pengguna untuk mengontrol intensitas lampu dan kecepatan kipas. Pengaturan ini dikirim ke *database server*, sehingga dapat diterapkan secara langsung pada mesin inkubator.

Melalui fitur *live cam* inkubator, pengguna dapat melihat kondisi inkubator secara langsung melalui *streaming* video. ESP32 Cam yang terpasang di inkubator mengirimkan video secara langsung ke aplikasi melalui IP *streaming* yang diberikan, sehingga pengguna dapat memantau kondisi telur dan memastikan proses penetasan berjalan dengan baik. Fitur ini sangat bermanfaat karena dapat

memberikan informasi visual tambahan yang tidak dapat diperoleh hanya dari data suhu dan kelembapan.

Pada tahapan berikutnya, diperlukan aplikasi seperti Figma dan Android Studio. Figma adalah sebuah perangkat lunak yang memungkinkan pengguna untuk merancang antarmuka aplikasi dengan desain yang menarik, serta dilengkapi dengan fitur untuk membuat prototipe interaktif. Desain antarmuka aplikasi Android yang telah dibuat dapat dilihat pada Gambar 3.14.



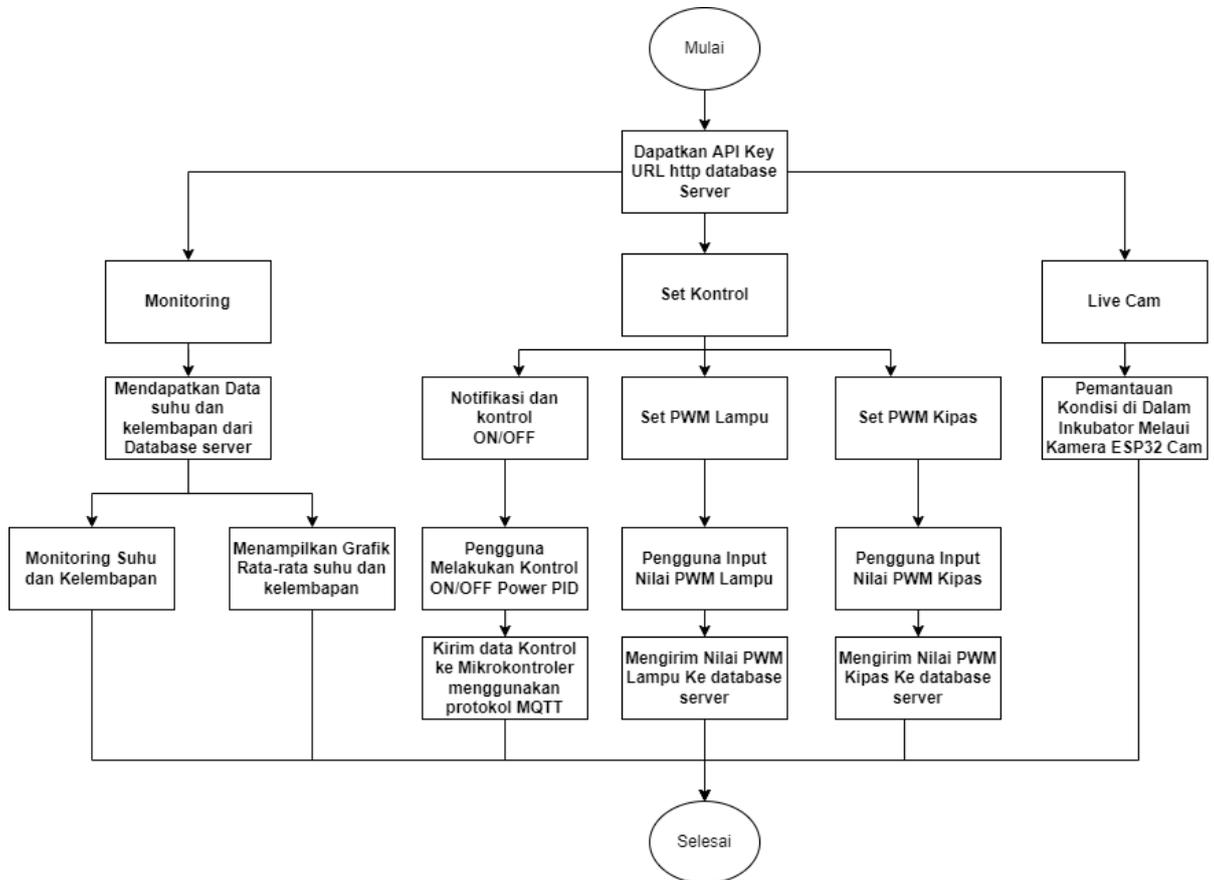
Gambar 3.14 *Mockup* Aplikasi Android

Setelah desain antarmuka aplikasi selesai, langkah berikutnya adalah proses perancangan aplikasi menggunakan Android Studio. Dengan menggabungkan ketiga fitur utama ini, aplikasi Android inkubator penetas telur diharapkan dapat memberikan kemudahan dan kenyamanan pada proses *monitoring* dan pengendalian dalam penetasan telur.

Aplikasi ini memungkinkan pengguna untuk melihat kondisi inkubator secara berkala seperti suhu dan kelembapan, mengatur parameter yang diperlukan untuk mengendalikan kipas dan lampu pijar, serta memungkinkan untuk melihat kondisi visual inkubator secara langsung melalui kamera *monitoring*. Dengan demikian, pengguna dapat meningkatkan tingkat keberhasilan penetasan telur melalui pemantauan yang lebih terperinci dan pengaturan yang lebih presisi.

Aplikasi ini dirancang dengan tujuan agar dapat digunakan dengan mudah oleh berbagai kalangan pengguna, dari pemula hingga yang berpengalaman. Untuk memastikan bahwa alur penggunaan aplikasi ini berjalan dengan efektif dan efisien, dibuatlah sebuah diagram alir penggunaan aplikasi. Diagram ini menggambarkan secara rinci langkah-langkah dan proses yang harus dilalui oleh pengguna, mulai

dari saat pertama kali mereka membuka aplikasi hingga mereka mencapai tujuan akhir yang diinginkan. Detail lebih lanjut mengenai diagram alir perangkat lunak ini dapat dilihat pada Gambar 3.15.



Gambar 3.15 Diagram Alir Perangkat Lunak

Pada gambar 3.15 diagram alir perangkat lunak, tahapan awal dari penggunaan aplikasi android adalah, perangkat melakukan konfigurasi dengan *API Key database server* untuk mendapatkan izin akses data yang akan digunakan pada aplikasi. Terdapat 3 fitur utama pada aplikasi android, fitur pertama adalah *Monitoring* yang memberikan informasi kepada pengguna mengenai kondisi suhu dan kelembapan pada inkubator dengan data yang ditampilkan pada aplikasi akan diperbaharui secara berkala, pada fitur tersebut juga terdapat tampilan grafik yang memberikan informasi mengenai rata-rata suhu yang didapatkan dari inkubator selama periode waktu tertentu, terdapat 2 pilihan mode grafik yaitu *today* dan *monthly*, pada mode *today* grafik akan menampilkan rata-rata suhu yang didapatkan sensor pada hari tersebut dan akan diperbaharui lagi di hari selanjutnya, kemudian

mode *monthly* akan memberikan tampilan grafik rata-rata suhu berdasarkan hari selama 1 bulan, pada mode ini data grafik akan di perbaharui pada bulan berikutnya. Hal ini dapat memberikan manfaat kepada pengguna untuk menganalisis perilaku suhu inkubator.

Fitur kedua yaitu set kontrol, pada fitur ini pengguna dapat secara langsung menghidupkan ataupun mematikan sistem kendali PID pada inkubator menggunakan protokol MQTT yang dikirimkan dari jarak jauh melalui aplikasi android, hal ini bertujuan untuk mengurangi risiko yang tidak diinginkan ketika mesin inkubator dalam keadaan menyala. Selain itu pengguna dapat mengatur sendiri intensitas cahaya lampu pijar dan kecepatan kipas dengan memberikan nilai PWM sesuai kebutuhan.

Fitur ketiga adalah *live cam*, merupakan fitur *monitoring* secara langsung melalui *streaming* video menggunakan ESP32 Cam yang telah terintegrasi antara aplikasi android dan *link url* ESP32 Cam. Dengan adanya fitur ini pengguna dapat secara langsung melihat kondisi yang terjadi pada inkubator dari jarak jauh menggunakan aplikasi android.

3.5 Pengujian

Pada tahapan pengujian, akan dilakukan beberapa pembuktian sistem secara keseluruhan. Ini dilakukan untuk menentukan apakah inkubator dapat melakukan fungsinya sesuai dengan harapan. Pengujian ini meliputi pengujian perangkat lunak (*software*) dan pengujian sistem secara keseluruhan. Setelah rancangan pengujian selesai, diharapkan sistem dapat beroperasi sesuai dengan fungsinya.

3.5.1 Pengujian Perangkat Lunak

Pengujian perangkat lunak akan menggunakan metode Black Box Testing, yang bertujuan untuk mengevaluasi fungsi aplikasi tanpa memeriksa struktur internal atau kode sumber aplikasi tersebut (Fahrezi dkk., 2022). Pendekatan ini berfokus pada *input* dan *output* yang dihasilkan oleh aplikasi untuk memastikan setiap fitur berfungsi sesuai dengan spesifikasinya. Dengan metode ini, pengujian dilakukan dari perspektif pengguna akhir, yang memungkinkan identifikasi kesalahan atau kekurangan yang mungkin tidak terdeteksi dari sisi pengembang, serta memastikan kualitas dan keandalan aplikasi sesuai tujuan yang diinginkan.

Tahapan pengujian perangkat lunak dengan metode Black Box Testing dapat dilihat pada tabel 3.1. Proses ini melibatkan berbagai skenario uji untuk mengidentifikasi potensi kesalahan fungsi dalam aplikasi.

Tabel 3.1
Pengujian Perangkat Lunak

Fitur yang Diuji	Jenis Fitur	Kebutuhan	Hasil yang Diharapkan
<i>Monitoring</i>	<ol style="list-style-type: none"> 1. Tampilan data suhu dan kelembapan. 2. Grafik suhu dan kelembapan. 	<p>Pengguna dapat mengetahui informasi suhu dan kelembapan pada inkubator.</p>	<ol style="list-style-type: none"> 1. Aplikasi android menampilkan suhu dan kelembapan inkubator secara berkala. 2. Grafik menampilkan suhu dan kelembapan dari beberapa percobaan.
Kontrol	<ol style="list-style-type: none"> 1. Notifikasi Kontrol PID. 2. Kontrol ON/OFF PID. 3. Set PWM Lampu 4. Set PWM Kipas 	<p>Memudahkan pengguna untuk mengendalikan inkubator dari jarak jauh dan menyesuaikan pengaturan inkubator sesuai dengan kebutuhan.</p>	<ol style="list-style-type: none"> 1. Aplikasi Android mengontrol sistem kendali PID inkubator. 2. Aplikasi android mengirim perintah untuk menetapkan nilai PWM lampu dan kipas.
Live Cam	<ol style="list-style-type: none"> 1. Live video <i>streaming</i>. 	<p>Pengguna dapat melihat kondisi inkubator ketika tidak berada di lokasi inkubator.</p>	<ol style="list-style-type: none"> 1. Aplikasi mampu menampilkan video <i>streaming</i> ESP32 Cam dan melihat kondisi inkubator.

3.5.2 Pengujian Perangkat Keras

Proses pengujian perangkat keras bertujuan untuk memastikan bahwa semua komponen fisik dari sistem berfungsi sesuai dengan tujuan yang diharapkan. Proses ini mencakup pengujian untuk setiap komponen guna mengidentifikasi kelebihan dan kekurangannya. Pengujian dilakukan secara mandiri untuk mengevaluasi performa, keandalan, dan kompatibilitas masing-masing komponen dalam berbagai kondisi operasional. Tahapan pengujian perangkat keras dapat dilihat pada tabel 3.2.

Tabel 3.2
Pengujian Perangkat Keras

No.	Komponen	Hasil yang Diharapkan
1	Sensor DHT11	Mampu mengukur data suhu dan kelembapan.
2	Lampu Pijar	Lampu pijar mampu menyala dengan memberikan variasi tegangan PWM.
3	Kipas	Kipas dapat berputar secara stabil dengan memberikan <i>input</i> tegangan PWM
4	Synchronous Motor	Dapat menggerakkan rak telur dengan sistem penggeseran.
5	LCD	Dapat menampilkan sejumlah data berbentuk Text dengan jelas.
6	ESP32 Cam	Dapat menampilkan kondisi inkubator secara langsung.

3.5.3 Pengujian Keseluruhan Sistem

Pengujian keseluruhan sistem inkubator melibatkan integrasi antara proses inkubator dengan sistem kontrol PID, serta penggunaan aplikasi Android sebagai media *monitoring* dan kontrol. Proses pengujian ini dilakukan untuk memastikan bahwa sistem dapat mengatur suhu dan kelembapan dengan tepat, serta menyajikan informasi tersebut kepada pengguna. Sistem kontrol PID digunakan untuk

mengendalikan suhu di dalam inkubator dengan mengatur *output* dari elemen pemanas dan kipas berdasarkan parameter yang telah ditetapkan (nilai Kp, Ki, dan Kd). Selama pengujian, data suhu dan kelembapan dikumpulkan dan disimpan dalam *database*, kemudian diakses dan ditampilkan melalui aplikasi Android.

3.6 Analisis Hasil

Setelah data percobaan dan pengujian berhasil dikumpulkan, tahap selanjutnya adalah menganalisis hasil tersebut secara mendetail. Analisis hasil ini sangat penting untuk memahami kinerja dan efektivitas sistem yang telah diuji. Teknik yang digunakan untuk menganalisis hasil adalah teknik statistik deskriptif. Menurut Sutanapong (2015), teknik ini merupakan suatu metode analisis data yang bertujuan untuk menggambarkan atau mendeskripsikan data sebagaimana adanya, teknik statistik deskriptif memungkinkan peneliti untuk menyajikan data dalam bentuk yang lebih terstruktur dan informatif, sehingga dapat memberikan gambaran yang jelas tentang kondisi atau fenomena yang sedang diteliti. Dengan menggunakan teknik ini, data yang diperoleh dari percobaan dan pengujian dapat diorganisir dan dijelaskan dengan cara yang lebih mudah dipahami (Fauzi., 2020).