

BAB III

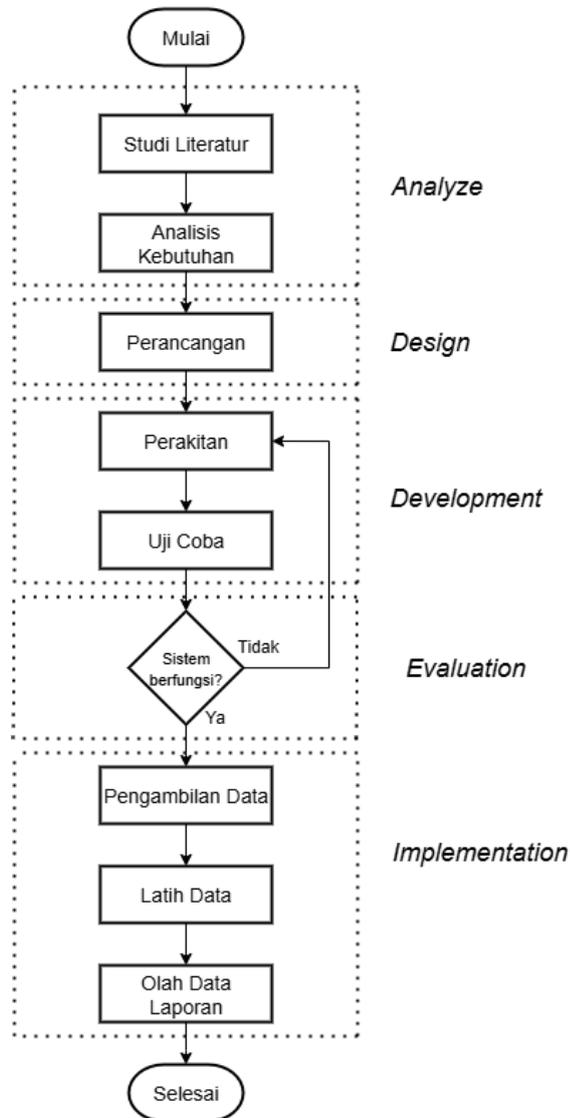
METODOLOGI PENELITIAN

Pada Bab III ini akan membahas secara rinci metode yang digunakan dalam penelitian ini, mulai dari perancangan sistem, pengumpulan data, hingga proses pengolahan dan analisis. Metode penelitian yang digunakan adalah *Research and Development* (R&D) dengan model pengembangan ADDIE yang mencakup tahap *Analysis, Design, Development, Implementation, dan Evaluation*. Pemilihan metode ini bertujuan untuk memastikan bahwa sistem yang dikembangkan tidak hanya berfungsi sesuai rancangan, tetapi juga memenuhi kebutuhan pengguna dan menghasilkan data penguapan yang akurat. Pada bab ini, dijelaskan pula spesifikasi perangkat keras dan perangkat lunak, prosedur pengujian alat, serta metode analisis data yang digunakan, termasuk penerapan algoritma regresi linear untuk memodelkan hubungan antara variabel lingkungan dan tingkat penguapan.

3.1 Jenis Penelitian

Jenis penelitian yang digunakan dalam penelitian ini adalah penelitian dan pengembangan atau yang biasa disebut *Research and Development* (R&D). Penelitian R&D bertujuan untuk mengembangkan dan menghasilkan produk yang inovatif serta efektif. Pada penelitian ini, model yang digunakan adalah model ADDIE yang merupakan salah satu pendekatan populer dalam proses pengembangan. Model ADDIE pertama kali dikembangkan pada tahun 1970-an dan hingga kini sering digunakan dalam berbagai bidang terutama untuk merancang dan mengembangkan produk yang berorientasi pada kinerja (Waruwu, 2024).

Diagram Alir Penelitian



Gambar 3.1 Diagram alir penelitian

Seperti pada Gambar 3.1 merupakan model ADDIE terdiri dari lima tahapan utama yaitu Analisis, Desain, Pengembangan, Implementasi, dan Evaluasi. Setiap tahap memiliki peran yang saling berhubungan dan saling mempengaruhi. Pada tahap Analisis, dilakukan identifikasi kebutuhan dan masalah yang menjadi dasar pengembangan. Tahap Desain mencakup perencanaan detail produk atau sistem seperti pembuatan kerangka kerja atau prototipe awal. Kemudian tahap Pengembangan fokus pada pembuatan produk atau sistem berdasarkan desain yang

telah dirumuskan. Tahap Implementasi adalah proses penerapan produk dalam lingkungan nyata, sementara tahap Evaluasi bertujuan untuk menilai efektivitas dan efisiensi produk yang dikembangkan.

Meskipun tahap Evaluasi ditempatkan di akhir, proses evaluasi sebenarnya dilakukan secara berkelanjutan di setiap tahapan. Evaluasi di tiap langkah bertujuan untuk memastikan keberlanjutan dan efektivitas dari proses pengembangan. Dengan adanya evaluasi yang berkesinambungan, setiap kekurangan atau masalah dapat segera diidentifikasi dan diperbaiki sebelum melanjutkan ke tahap berikutnya. Model ADDIE yang fleksibel dan sistematis ini menjadi salah satu metode yang sangat cocok untuk menghasilkan produk berbasis teknologi, termasuk dalam penelitian ini.

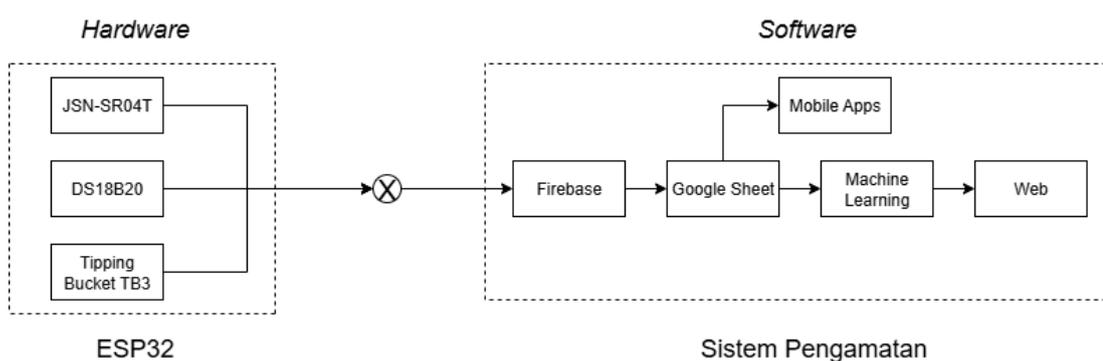
3.2 Deskripsi Umum Sistem

Sistem pemantauan panci penguapan otomatis berbasis IoT dan algoritma *Random Forest* adalah inovasi teknologi yang dirancang untuk memantau tingkat penguapan air secara otomatis dan *real-time*. Sistem ini menggunakan sensor yang terhubung ke perangkat IoT untuk mengukur parameter lingkungan seperti suhu dan ketinggian air dalam panci penguapan serta curah hujan yang terjadi. Data yang dikumpulkan dikirimkan ke *database* melalui jaringan internet untuk analisis lebih lanjut. Dengan memanfaatkan teknologi IoT, sistem ini memungkinkan pengguna untuk memantau penguapan dari jarak jauh melalui aplikasi atau *dashboard* berbasis web.

Selain pengukuran *real-time*, sistem ini juga menggunakan algoritma *Random Forest* untuk menganalisis pola data dan memprediksi tingkat penguapan berdasarkan variabel lingkungan yang relevan. Hasil prediksi ini membantu memberikan wawasan yang lebih mendalam mengenai faktor-faktor yang mempengaruhi penguapan sehingga dapat digunakan untuk keperluan penelitian iklim, manajemen sumber daya air, atau pertanian. Integrasi IoT dan algoritma *Random Forest* menjadikan sistem ini tidak hanya canggih dalam pengukuran tetapi juga andal dalam memberikan prediksi berbasis data.

Penelitian ini menggunakan beberapa komponen untuk membangun sistem pemantauan panci penguapan otomatis. ESP32 berperan sebagai mikrokontroler utama yang bertanggung jawab untuk memproses seluruh data. Data lingkungan dikumpulkan oleh tiga sensor utama yaitu DS18B20 yang berfungsi sebagai sensor suhu air, JSN-SR04T yang digunakan untuk mengukur ketinggian air, dan *Tipping Bucket* TB3 yang bertugas menghitung curah hujan. Perakitan rangkaian menggunakan *bread board* sebagai tempat *wiring* alat dengan kabel *jumper* yang berfungsi untuk mengalirkan tegangan antar komponen. Seluruh komponen ini diaktifkan menggunakan sumber tegangan AC yang menyediakan daya listrik yang diperlukan untuk operasional sistem.

Arsitektur sistem pada Gambar 3.2 menunjukkan integrasi antara panci penguapan otomatis dan sistem pemantauan berbasis IoT. Bagian pertama yaitu sistem berbasis IoT menggunakan ESP32 yang berfungsi sebagai pusat pengumpulan data dari berbagai sensor. Sensor yang digunakan diantaranya JSN-SR04T, DS18B20, dan *Tipping Bucket* TB3. Data yang diperoleh dari sensor dikirimkan ke sistem pemantauan melalui koneksi internet menggunakan ESP32 sebagai mikrokontroler utama. ESP32 bertanggung jawab untuk mengumpulkan, memproses, dan mengirim data ke Firebase.



Gambar 3.2 Arsitektur Sistem Pemantauan Panci Penguapan

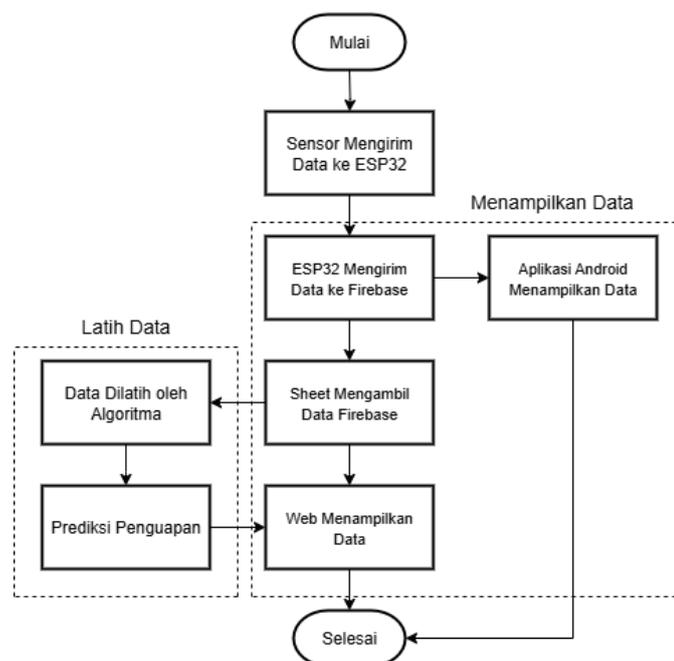
Bagian kedua adalah sistem pemantauan yang terhubung dengan Firebase sebagai *backend* penyimpanan data. Data dari Firebase dapat diolah lebih lanjut dan diekspor ke Google Sheet untuk mempermudah analisis dan visualisasi data. Sistem pemantauan ini menyediakan antarmuka berbasis web dan *mobile*, sehingga

pengguna dapat memantau hasil pengukuran sensor secara *real-time* melalui perangkat yang terhubung dengan internet. Arsitektur ini menciptakan sistem IoT yang efisien, memungkinkan pengawasan parameter lingkungan secara terpusat dan fleksibel.

3.2.1 Prinsip Kerja Alat

Sistem pemantauan panci penguapan berbasis IoT ini hampir seluruh prosesnya berjalan secara otomatis. Sistem ini dirancang agar seluruh proses pemantauan panci penguapan serta proses perhitungan penguapan setiap harinya dilakukan dengan otomatis. Dengan adanya otomatisasi ini diharapkan dapat menggantikan pemantauan manual agar lebih efektif dan efisien.

Diagram Alir Sistem



Gambar 3.3 Diagram Alir Sistem Pemantauan Panci Penguapan

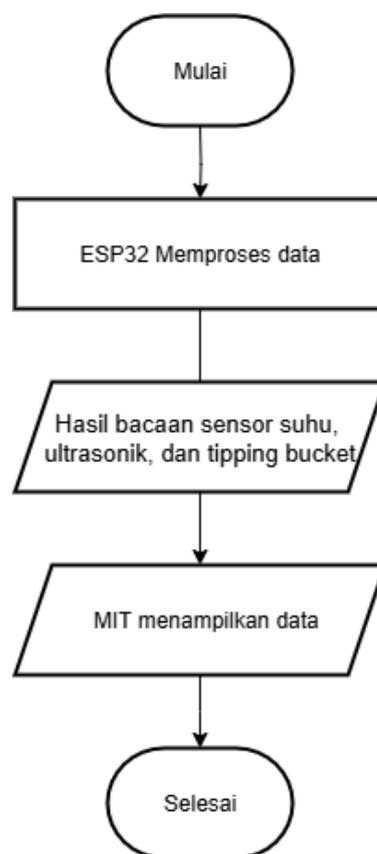
Berdasarkan Gambar 3.3 sistem dimulai dari seluruh sensor yang ada mengirim data hasil bacaannya ke mikrokontroler ESP32. Kemudian ESP32 mengirim data tersebut ke Firebase agar data dapat dilihat secara *online*. Setelah itu aplikasi android yang telah dibuat mengambil data dari Firebase dan menampilkannya secara *real-time*. Selain itu, Google Sheet pun mengambil data

dari Firebase dan menyimpan data tersebut setiap jamnya. Data yang telah terkumpul kemudian dilatih oleh algoritma agar sistem mampu memprediksi penguapan kedepannya. Diakhiri oleh web menampilkan data *real-time*, data prediksi, dan juga data dari Google Sheet sebagai basis data.

3.3 Perancangan *Hardware*

Perancangan perangkat keras dalam penelitian ini difokuskan pada pengembangan sistem berbasis IoT yang mampu mengukur parameter lingkungan secara otomatis dan akurat. Perangkat keras utama yang digunakan merupakan mikrokontroler ESP32 yang dipilih karena kemampuannya dalam memproses data sensor dan konektivitas nirkabel yang handal. Sistem ini dilengkapi dengan sensor JSN-SR04T untuk mengukur ketinggian air, DS18B20 untuk membaca suhu, dan *Tipping Bucket* TB3 untuk mengukur curah hujan. Komponen-komponen ini dirancang untuk bekerja secara sinergis, di mana data yang dikumpulkan dari sensor dikirimkan ke sistem pemantauan melalui koneksi internet. Proses perancangan juga mempertimbangkan efisiensi daya, kompatibilitas antar-komponen, dan perlindungan perangkat terhadap kondisi lingkungan sehingga memastikan sistem dapat beroperasi secara optimal dalam jangka waktu yang lama.

ESP32 mengumpulkan data dari berbagai sensor yang terhubung termasuk sensor ultrasonik, sensor suhu, dan tipping bucket. Sensor ultrasonik mengukur ketinggian permukaan air. Sensor suhu mencatat suhu air, sementara tipping bucket mencatat jumlah curah hujan yang terjadi. Data yang diperoleh dari setiap sensor diproses oleh ESP32 untuk menghasilkan informasi yang lebih terstruktur dan siap digunakan seperti ditunjukkan pada Gambar 3.4.



Gambar 3.4 *Flowchart Hardware*

Setelah data diproses, ESP32 mengirimkan hasil pengukuran ke aplikasi MIT App Inventor yang telah dirancang untuk menampilkan data secara *real-time*. Pengguna dapat memantau data seperti tinggi air, suhu air, dan jumlah penguapan melalui antarmuka aplikasi yang mudah digunakan. Sistem ini memungkinkan pemantauan kondisi lingkungan dengan lebih praktis dan efisien serta mendukung kebutuhan pengawasan jarak jauh.

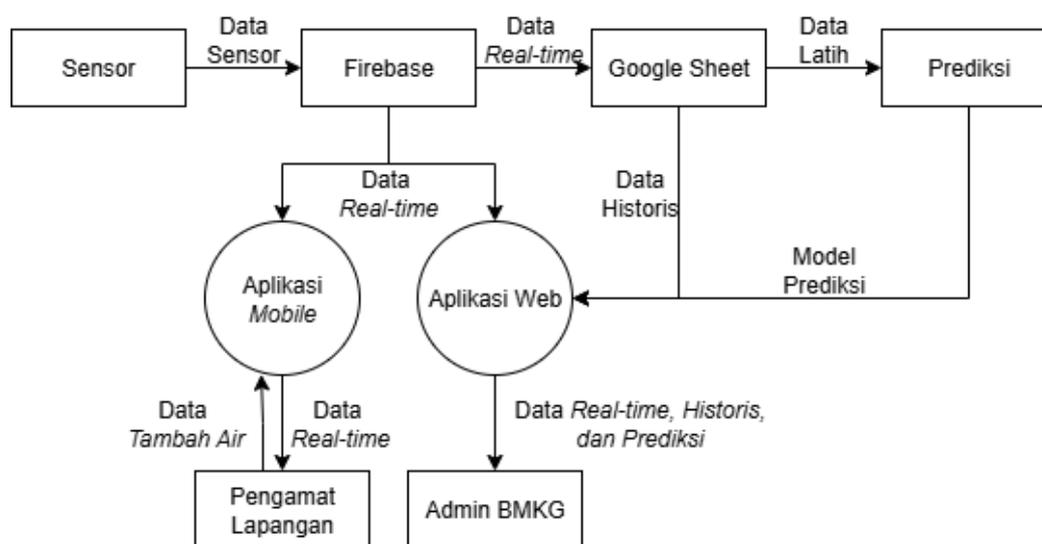
3.4 Perancangan *Software* Sistem Pemantauan Panci Penguapan

Perancangan perangkat lunak dalam penelitian ini bertujuan untuk membangun sistem pemantauan berbasis IoT yang dapat mengintegrasikan pengukuran data lingkungan dari perangkat keras ke *platform* digital secara *real-time*. Proses perancangan dimulai dengan pengembangan *firmware* pada mikrokontroler ESP32 yang bertugas mengatur sensor, membaca data, dan mengirimkannya ke Firebase melalui koneksi internet. Selanjutnya, data yang tersimpan di Firebase diolah untuk

ditampilkan dalam antarmuka pengguna berbasis web dan mobile yang dirancang menggunakan *framework* modern untuk memastikan kemudahan akses dan penggunaan. Perancangan juga mencakup integrasi dengan Google Sheet untuk mendukung penyimpanan dan analisis data secara terstruktur sehingga hasil pengukuran dapat digunakan untuk pengambilan keputusan secara efektif.

3.4.1 Prinsip Kerja *Software* pada Sistem Pemantauan Panci Penguapan

Sebagai modul WiFi, ESP32 mengumpulkan data dari sensor-sensor yang terhubung dan mengolahnya serta mengirimkan hasil olahan ke Firebase seperti pada Gambar 3.5. Firebase merupakan layanan *backend* yang berfungsi sebagai penyimpanan data berbasis *cloud*. Proses ini memungkinkan data dari berbagai sensor dikumpulkan secara *real-time* dan disimpan dengan aman di *server* Firebase untuk keperluan lebih lanjut.



Gambar 3.5 Data Flow Diagram Sistem Pemantauan Panci Penguapan

Firestore kemudian menyuplai data *real-time* ke dua arah, yaitu ke Google Sheet sebagai arsip dan ke aplikasi berbasis *mobile* dan web yang digunakan oleh pengamat lapangan untuk melihat dan menambahkan data manual seperti volume air tambah. Selain itu, aplikasi web menerima data *real-time* dari Firestore dan juga mengakses data historis dari Google Sheet untuk ditampilkan kepada admin BMKG

sebagai laporan lengkap yang mencakup data *real-time*, historis, dan hasil model prediksi.

Di sisi lain, data historis dari Google Sheet juga digunakan sebagai data latih (*training*) untuk proses prediksi dan hasil dari model prediksi dikirim ke aplikasi web agar dapat dimanfaatkan kembali oleh admin maupun pihak pengelola data. Dengan arsitektur ini, sistem memungkinkan pertukaran data yang terintegrasi dan *real-time* antara perangkat IoT, pengamat, model prediktif, dan pengguna administratif.

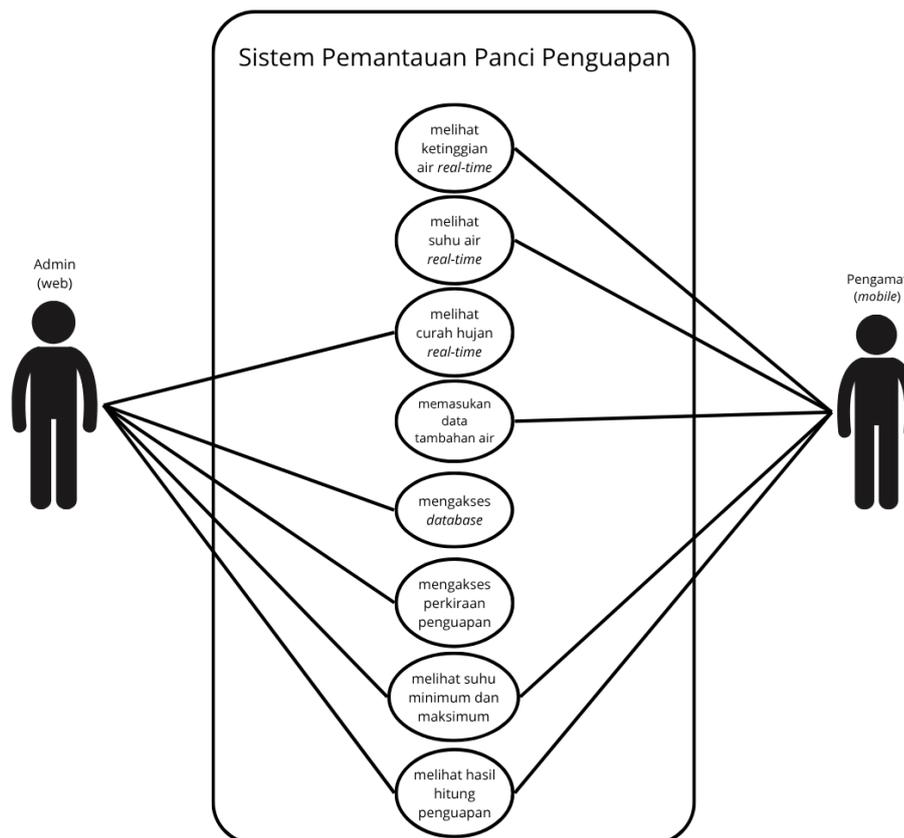
Proses *forecasting* dalam sistem ini bekerja pada bagian Prediksi yang terhubung langsung dengan Google Sheet sebagai sumber data latih. Sistem *forecasting* ini merupakan komponen *software* berbasis *machine learning* yang berjalan terpisah dari sistem pengumpulan data sensor. *Input* utama dari proses *forecasting* adalah data historis yang tersimpan di Google Sheet, yang sebelumnya dikirim dari Firebase secara *real-time*. Data ini mencakup nilai-nilai penguapan harian, suhu, curah hujan, dan lama penyinaran dalam periode waktu tertentu yang digunakan sebagai fitur (*variabel input*) untuk melatih model. Selain itu, model juga memerlukan struktur data *time series* seperti nilai H-1 (hari sebelumnya) sebagai bagian dari fitur prediktif.

Setelah proses pelatihan dan prediksi dijalankan, output dari sistem *forecasting* berupa model prediksi terlatih. Model prediksi terlatih ini kemudian digunakan oleh aplikasi web untuk ditampilkan kepada admin BMKG, serta dapat dikombinasikan dengan data *real-time* pada tampilan sistem pemantauan. Dengan demikian, komponen prediksi menjadi bagian krusial yang mengubah data historis menjadi informasi prediktif yang mendukung pengambilan keputusan dalam pengelolaan data klimatologi.

3.4.2 User Interface

Sistem pemantauan penguapan otomatis yang dikembangkan dalam penelitian ini memiliki dua antarmuka utama, yaitu aplikasi berbasis mobile dan aplikasi berbasis web. Pembagian ini didasarkan pada kebutuhan pengguna dengan

dua peran berbeda, yaitu pengamat lapangan dan admin BMKG. Dengan adanya dua antarmuka, sistem menjadi lebih fleksibel dan efisien karena setiap jenis pengguna dapat mengakses fungsi-fungsi yang relevan sesuai tanggung jawabnya, tanpa dibebani oleh fitur yang tidak diperlukan.



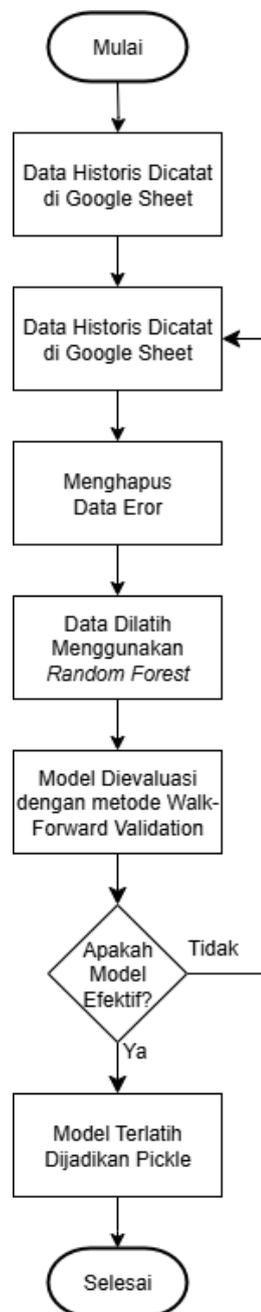
Gambar 3.6 Use Case Diagram Panci Penguapan

Berdasarkan Gambar 3.6, aplikasi *mobile* dirancang khusus untuk digunakan oleh petugas pengamat lapangan BMKG. Antarmuka ini dibuat sederhana dan ringan agar mudah dioperasikan di lapangan, bahkan dengan keterbatasan jaringan. Melalui aplikasi ini, petugas dapat melihat data sensor secara *real-time*, serta menambahkan data manual seperti volume air tambahan jika diperlukan. Tujuan utama dari aplikasi *mobile* adalah mempermudah proses pemantauan dan pelaporan harian sehingga pengumpulan data dapat dilakukan dengan lebih cepat, akurat, dan efisien di lapangan.

Sementara itu, aplikasi web ditujukan untuk digunakan oleh admin BMKG yang memiliki wewenang lebih dalam mengelola dan menganalisis data. Aplikasi web ini menampilkan data historis hasil pemantauan, yang sifatnya lebih sensitif karena merupakan bagian dari data komersial BMKG. Selain itu, aplikasi web juga terintegrasi dengan sistem prediksi penguapan berbasis machine learning, sehingga admin dapat melihat hasil proyeksi nilai penguapan untuk keperluan analisis lanjutan atau pengambilan keputusan. Dengan demikian, pembagian dua antarmuka ini mendukung alur kerja yang lebih terstruktur dan sesuai dengan kebutuhan masing-masing peran pengguna dalam sistem.

3.5 Perancangan Model Prediksi

Algoritma *Random Forest* dipilih sebagai metode *machine learning* untuk melakukan pemodelan prediktif terhadap nilai penguapan. Pemilihan ini didasarkan kepada perbandingan hasil performa dengan algoritma lainnya seperti *Decision Tree* dan *Linear Regression*, serta penelitian terdahulu yang membandingkan dengan *Decision Tree* dan *Naive Bayes*. Pendekatan ini bertujuan untuk mengidentifikasi dan memahami secara mendalam hubungan kompleks yang mungkin bersifat non-linear antara berbagai variabel *input* dengan variabel *output* yaitu nilai penguapan. Variabel-variabel input yang dipertimbangkan dalam model ini mencakup data meteorologis seperti suhu maksimum dan suhu minimum harian yang tercatat, total curah hujan yang terjadi, lama penyinaran matahari yang diterima, serta faktor waktu atau urutan temporal data. Dengan menganalisis pola interaksi antar variabel-variabel ini, algoritma *Random Forest* diharapkan mampu menghasilkan model prediksi yang akurat dan andal untuk memperkirakan tingkat penguapan di masa mendatang berdasarkan kondisi lingkungan yang terukur.



Gambar 3.7 Diagram Alur Pelatihan Data

Proses pembuatan sistem *forecasting* seperti pada Gambar 3.7 pada penelitian ini dilakukan melalui beberapa tahapan yang sistematis, dimulai dari pengumpulan data (*data collecting*). Data dikumpulkan secara otomatis dari perangkat IoT berupa sensor penguapan, suhu, curah hujan, dan lama penyinaran matahari yang dikirimkan ke Firebase Realtime Database. Data real-time ini kemudian diekspor

ke Google Sheets dan diunduh untuk dilakukan pemrosesan lebih lanjut di lingkungan Google Colab. Dalam penelitian ini, tidak dilakukan tahap eksplorasi data (*Exploratory Data Analysis*) secara menyeluruh karena fitur-fitur yang digunakan dalam model sudah ditentukan di awal berdasarkan standar variabel yang digunakan oleh BMKG dalam perhitungan penguapan seperti suhu minimum, suhu maksimum, curah hujan, dan lama penyinaran.

Setelah data terkumpul, dilakukan tahap *preprocessing* yaitu membersihkan data dari nilai kosong atau tidak logis, mengatur format tanggal, serta membuat fitur tambahan seperti lag (H-1) agar data time series bisa diproses dalam bentuk *supervised learning*. Selain itu, dilakukan pula proses *augmentasi time series* untuk memperbanyak data latih dengan teknik penggeseran jendela waktu (*sliding window*) sehingga model memiliki lebih banyak sampel untuk belajar dan dapat meningkatkan generalisasi. Data kemudian dipisahkan menjadi data latih dan data uji dengan tetap menjaga urutan waktu agar valid dalam konteks *forecasting*. Tahap berikutnya adalah pelatihan model menggunakan algoritma *Random Forest*.

Setelah model dilatih, dilakukan tahap evaluasi dengan menggunakan metrik regresi seperti MAE, RMSE, dan R^2 untuk menilai kinerja model terhadap data uji. Dalam penelitian ini, model dievaluasi menggunakan teknik *Walk-Forward Validation*, yaitu metode evaluasi yang membagi data secara berurutan berdasarkan waktu sehingga dapat meniru kondisi sistem prediksi dunia nyata secara lebih akurat dan mencegah kebocoran informasi dari masa depan. Teknik ini memungkinkan model dilatih pada data masa lalu, lalu diuji pada satu titik waktu ke depan secara bertahap hingga seluruh data uji selesai diprediksi. Model dengan performa terbaik disimpan dalam bentuk *file .pkl* sebagai model terlatih yang kemudian digunakan untuk melakukan prediksi nilai penguapan secara berkelanjutan dalam sistem. Pendekatan ini memastikan bahwa proses *forecasting* tidak hanya akurat, tetapi juga sesuai dengan karakteristik data *time series* dan kebutuhan operasional sistem pemantauan penguapan otomatis.