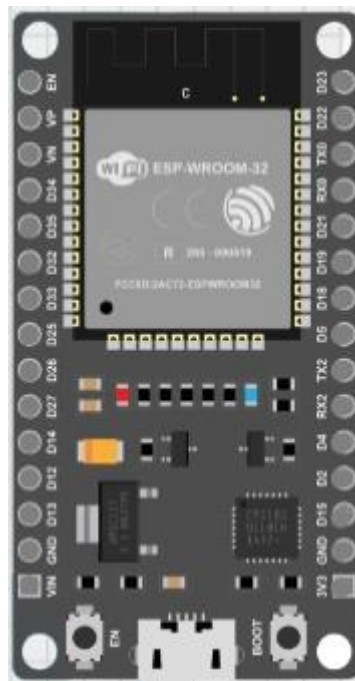


BAB 3

METODE PENELITIAN

3.1 ALAT YANG DIGUNAKAN

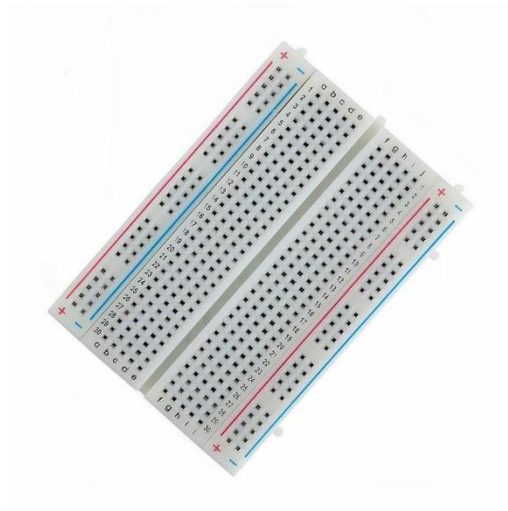
ESP32 dipilih sebagai komponen utama karena kemampuannya yang handal dalam mengintegrasikan sensor dan perangkat serta koneksi WiFi untuk pengiriman dan penerimaan data nirkabel. Pemahaman tentang cara kerja ESP32 akan mencakup konfigurasi perangkat keras, seperti pengaturan pin input/output, dan pengembangan perangkat lunak, seperti skrip untuk membaca data sensor dan mengirimkan instruksi kontrol ke perangkat rumah tangga yang terhubung, serta akan membahas bagaimana ESP32 diintegrasikan dengan Arduino Cloud sebagai platform manajemen dan kontrol yang memungkinkan akses jarak jauh ke perangkat smart home.



Gambar 3.1 ESP 32

Breadboard adalah papan yang biasa digunakan untuk membuat prototipe rangkaian elektronik. *Breadboard* ini nantinya akan dihubungkan dengan ESP32, Karena pin pada ESP32 terbatas maka perlu adanya *breadboard* yang memiliki pin 400 titik. Dengan banyaknya titik masukan maka semua komponen dapat

terhubung dan pada tahap perancangan dapat berjalan dengan baik. Nantinya hanya ESP32 akan dipasangkan pada bagian kiri *breadboard* karena bagian kanan akan banyak pin yang diperlukan.



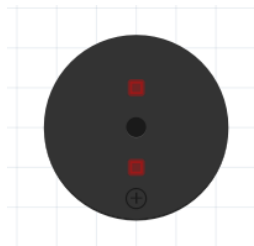
Gambar 3.2 BreadBoard

Dalam pengembangan sistem *smart home* berbasis *Internet of Things* (IoT) menggunakan perangkat lunak Arduino Cloud, sensor MQ-7 dipilih sebagai komponen dalam mendeteksi kadar gas LPG di sekitar. Pemahaman tentang prinsip kerja sensor MQ-7 akan mencakup mekanisme deteksi gas, termasuk jenis bahan bakar yang terbakar dan respons sensor terhadapnya untuk menghasilkan sinyal *output* yang relevan. Penelitian juga akan mempertimbangkan pengaturan fisik sensor, termasuk lokasi pemasangan dan integrasi dengan mikrokontroler, serta pengembangan skrip perangkat lunak untuk memproses data sensor dan mengambil langkah yang sesuai berdasarkan informasi yang diperoleh. Evaluasi terhadap keandalan sensor MQ-7 dalam sistem *smart home* akan menjadi bagian penting dalam perancangan ini, bertujuan untuk memastikan deteksi gas LPG berlangsung dengan akurat dan dapat diandalkan.



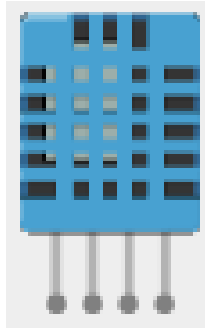
Gambar 3.3 Sensor MQ-7

Supaya penghuni rumah lebih mendapat kemudahan maka pada perancangan ini diperlukan buzzer guna memberitahu ketika kadar gas LPG yang deteksi oleh sensor MQ-7 melebihi dari standardisasi yang ada nanti buzzer ini akan berbunyi seperti layaknya alarm, memberikan notifikasi berupa suara jika terjadi kebocoran pada dapur rumah.



Gambar 3.4 Buzzer

Selain menggunakan buzzer yang berbunyi ada juga sensor DHT11. Sensor DHT11 dipilih sebagai komponen kunci karena kemampuannya dalam mengukur suhu dan kelembaban di sekitar. Pemahaman tentang operasi sensor ini akan mencakup prinsip kerjanya, termasuk jenis data yang dihasilkan (suhu dan kelembaban) dan bagaimana sensor ini merespons lingkungan sekitarnya. akan mempertimbangkan aspek konfigurasi perangkat keras, seperti pemasangan sensor dan koneksi dengan mikrokontroler, serta pengembangan perangkat lunak untuk menganalisis data sensor dan mengambil tindakan yang sesuai. Kinerja sensor DHT11 dalam aplikasi *smart home* dengan tujuan untuk memverifikasi keakuratan dan keandalannya dalam memberikan informasi suhu dan kelembaban yang akurat.



Gambar 3.5 Sensor DHT11

Pemahaman tentang cara kerja relay mencakup prinsip kerja perangkat tersebut, yang melibatkan kumparan elektromagnetik yang menghasilkan medan magnet saat diberi tegangan, serta kontak *switch* yang berubah posisi sebagai respons terhadap medan magnet tersebut. Ini juga akan mempertimbangkan aspek konfigurasi perangkat keras, termasuk cara pemasangan relay dan koneksi dengan perangkat listrik yang akan dikendalikan, serta pengembangan skrip perangkat lunak untuk mengatur aktivasi dan deaktivasi relay berdasarkan instruksi dari Arduino Cloud. Relay dalam sistem smart home menjadi cukup krusial dalam perancangan *smarthome* berbasis *internet of things*, bertujuan untuk memastikan bahwa relay dapat mengendalikan perangkat listrik dengan efektif sesuai perintah yang diberikan melalui internet.

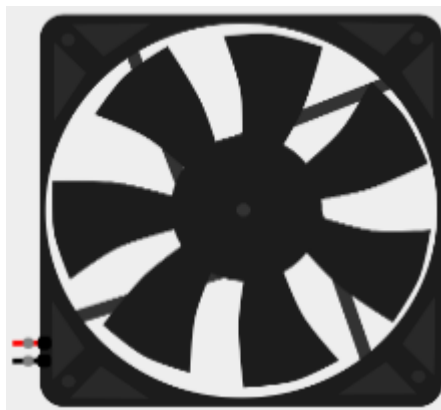


Gambar 3.6 Relay

Dengan menggunakan perangkat lunak Arduino Cloud, kipas angin dipilih untuk dikendalikan secara otomatis dalam sistem. Prinsip kerja motor kipas yang

bergantung pada aliran listrik melalui kumparan kawat di medan magnet, yang menghasilkan putaran kipas. juga akan memperhatikan konfigurasi perangkat keras, termasuk instalasi dan penyambungan kipas angin dengan sumber listrik, serta pengembangan skrip perangkat lunak untuk mengatur operasi kipas berdasarkan instruksi dari Arduino Cloud.

Dalam rumah pintar yang berbasis *Internet of Things* (IoT), kipas angin exhaust memiliki peran penting dalam menjaga kualitas udara dalam ruangan. Prosesnya dimulai dengan sensor suhu yang terpasang di ruangan untuk mengukur kondisi udara. Jika suhu atau kelembaban melewati batas yang telah ditentukan, informasi tersebut dikirim ke perangkat pengontrol yang terhubung ke internet. Perangkat tersebut kemudian mengirim perintah kepada kipas angin exhaust untuk mulai beroperasi. Motor kipas angin *exhaust* menerima sinyal tersebut dan memulai putaran, yang menghisap udara kotor dari dalam ruangan melalui saluran ventilasi dan mengeluarkannya ke luar rumah. Selama kipas angin exhaust aktif, sistem secara terus-menerus memantau kondisi udara secara *real-time*. Setelah suhu atau kelembaban kembali ke level yang nyaman, perangkat pengontrol mengirim perintah untuk mematikan kipas angin *exhaust*.



Gambar 3.7 Exhaust Fan

Dari semua perangkat keras di atas nantinya akan dikendalikan melalui smartphone maupun laptop dengan menggunakan *software* Arduino IoT Cloud. Arduino IoT Cloud merupakan sebuah platform cloud yang digunakan untuk memungkinkan penggunaan mikrokontroler Arduino secara terhubung ke internet. *Software* Arduino IoT Cloud berperan sebagai pusat pengendalian dan

pemantauan sistem *smart home* yang dikembangkan. Penggunaan Arduino IoT Cloud memungkinkan pengguna untuk mengontrol dan memantau perangkat-perangkat di rumah secara *remote* melalui koneksi internet.

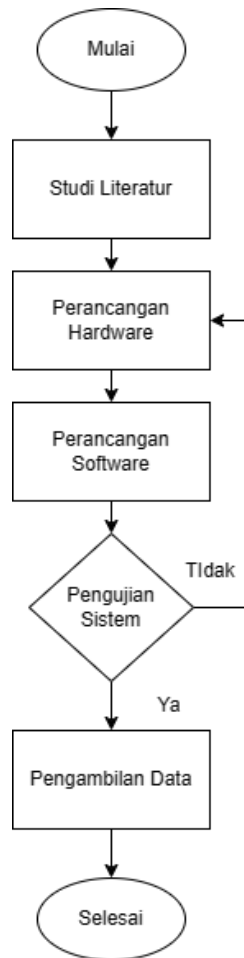
Cara kerja Arduino IoT Cloud dimulai dengan registrasi dan login pengguna ke platform tersebut melalui laptop atau *smartphone*. Setelah login, pengguna dapat membuat proyek baru untuk sistem *smart home* mereka dan menambahkan perangkat-perangkat yang terhubung ke mikrokontroler Arduino, seperti sensor suhu, relay, dan exhaust fan. Kemudian, pengguna dapat membuat aturan atau skenario pengendalian berdasarkan kondisi yang diinginkan, seperti mengatur suhu ruangan berdasarkan nilai suhu yang terdeteksi oleh sensor, atau mengaktifkan perangkat tertentu saat mendeteksi gerakan. Arduino IoT Cloud juga menyediakan fitur pemantauan yang memungkinkan pengguna untuk melihat data yang dikirim oleh sensor-sensor dan status perangkat dalam sistem *smart home* mereka secara *real-time*.



Gambar 3.8 Platform Arduino IoT Cloud

3.2 ALUR PENELITIAN

Penelitian dilakukan dalam beberapa tahap yaitu tahap perancangan sistem, tahap pembuatan simulasi, tahap pengujian simulasi, dan yang terakhir adalah tahap analisis dari hasil pengujian simulasi



Gambar 3.9 Alur Penelitian

3.3 TAHAP STUDI LITERATUR

Studi literatur akan mengeksplorasi konsep dasar *Internet of Things* (IoT) dan bagaimana IoT memfasilitasi koneksi dan pertukaran data antara objek fisik melalui internet. Perkembangan terbaru dalam teknologi IoT, termasuk peran dari Arduino sebagai platform pengembangan perangkat keras. Selanjutnya, akan menyelidiki konsep *smarthome* dan bagaimana IoT telah mengubah cara memandang dan mengelola rumah tangga. Ini melibatkan pemahaman tentang perangkat yang dapat diotomatisasi dalam rumah, seperti exhaust fan, serta bagaimana integrasi IoT memungkinkan pengendalian dan pemantauan dari jarak jauh terhadap perangkat tersebut.

Pada studi literatur ini akan membahas peran Arduino IoT Cloud dalam pengembangan sistem *smarthome*. Ini mencakup pemahaman tentang fitur dan fungsi yang disediakan oleh platform tersebut, serta kelebihan dan kekurangannya

dalam konteks pengembangan *smarthome*. Selanjutnya, dalam studi literatur akan menelusuri penelitian sebelumnya yang relevan dengan topik skripsi ini, termasuk implementasi *smarthome* berbasis IoT menggunakan platform Arduino atau platform lainnya, serta evaluasi terhadap kesuksesan dan hambatan dalam proyek-proyek semacam itu. Dengan menganalisis temuan dari studi literatur ini, penelitian ini akan membangun dasar teoritis yang kokoh untuk merancang dan melaksanakan sistem *smarthome* berbasis IoT menggunakan *software* Arduino IoT Cloud.

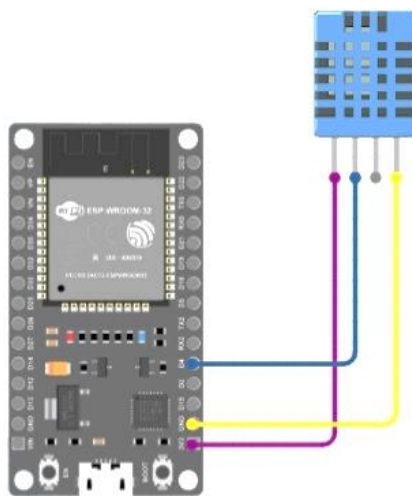
3.4 TAHAP PERANCANGAN *HARDWARE*

Dalam tahap perancangan hardware fokus utamanya mengintegrasikan komponen-komponen kunci yang diperlukan dalam sistem *smart home*. Pertama, sensor DHT11 akan diintegrasikan untuk mengukur suhu dan kelembaban di dalam ruangan, memberikan pemahaman yang mendalam tentang kondisi lingkungan. Selanjutnya, relay akan digunakan sebagai saklar untuk mengontrol perangkat listrik, seperti exhaust fan, dengan pengaturan yang terkait dengan suhu dan kelembaban yang terdeteksi oleh sensor DHT11. Pemasangan exhaust fan menjadi penting untuk meningkatkan kualitas udara dalam ruangan. Sensor MQ7 juga akan dipasang untuk mendeteksi gas LPG yang berpotensi berbahaya, memastikan keamanan penghuni rumah. Semua komponen ini akan terhubung ke mikrokontroler ESP32, yang bertindak sebagai otak sistem, mengumpulkan data dari sensor, mengontrol relay, dan berkomunikasi dengan *software* Arduino Cloud untuk memungkinkan pengendalian jarak jauh dan pemantauan melalui internet. Tahap perancangan *hardware* ini akan mencakup pemilihan dan penempatan yang tepat dari komponen-komponen tersebut, koneksi yang benar antara sensor, relay, dan mikrokontroler, serta pengujian untuk memastikan kinerja yang optimal dan terintegrasi dari keseluruhan sistem *smart home*.

3.4.1 PERANCANGAN PADA BAGIAN PENDETEKSI SUHU

Dalam tahap perancangan perangkat pada bagian pendeteksi suhu pada penelitian ini fokus utamanya adalah merancang dan mengimplementasikan sistem pendeteksi suhu dengan sensor DHT11. Cara mengaitkan sensor suhu

DHT11 dengan ESP32. Tahapan koneksi yang diperlukan, seperti menyambungkan pin VCC (pin 2) sensor DHT11 ke pin 3.3V pada ESP32, juga mengaitkan pin Data (pin 1) sensor DHT11 ke pin GPIO2 pada ESP32, serta pin GND (pin 4) sensor DHT11 ke pin GND pada ESP32. Selain itu, diuraikan fungsi setiap komponen, dimana sensor DHT11 bertugas mengukur suhu, sementara ESP32 bertanggung jawab atas pembacaan dan pemrosesan data dari sensor DHT11. Secara keseluruhan, gambar tersebut memberikan pedoman yang jelas dalam menghubungkan sensor suhu DHT11 dengan ESP32.

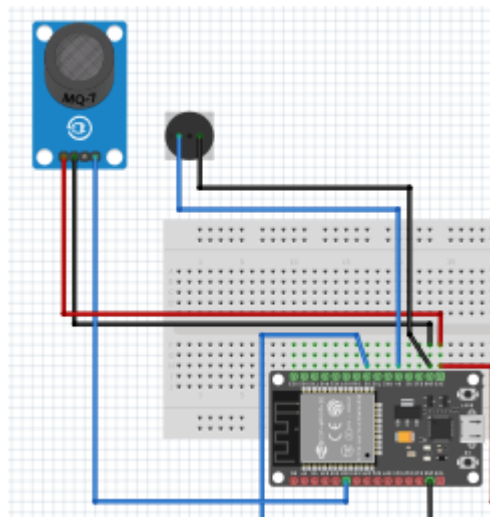


Gambar 3.9 Rancangan ESP 32 dengan sensor DHT11

3.4.2 PERANCANGAN PADA BAGIAN PENDETEKSI GAS

Komponen yang terdiri dari ESP32, sebuah papan pengembangan mikrokontroler ESP32, dan Sensor MQ-7, sebuah sensor gas yang difungsikan untuk mendeteksi gas LPG, ditunjukkan dalam diagram. Koneksi yang diperlukan antara kedua komponen tersebut adalah Vcc, yang mengalirkan tegangan 5V dari ESP32 ke pin Vcc sensor, GND, yang menghubungkan GND dari ESP32 ke pin GND sensor, dan Analog *Input*, dengan pin analog A0 dari ESP32 disambungkan ke pin *Aout* sensor. Fungsi dari Sensor MQ-7 adalah mendeteksi keberadaan gas LPG di udara. Sensor ini dilengkapi dengan elemen pemanas internal yang bertugas untuk memanaskan bahan sensor. Saat gas LPG terdeteksi, bahan sensor akan bereaksi dan menghasilkan perubahan resistansi yang berbeda. ESP32 kemudian mengukur perbedaan resistansi ini dan mengonversinya menjadi nilai

konsentrasi LPG. Kombinasi ESP32 dan Sensor MQ-7 termasuk pemantauan kualitas udara untuk mendeteksi kebocoran gas LPG di rumah.



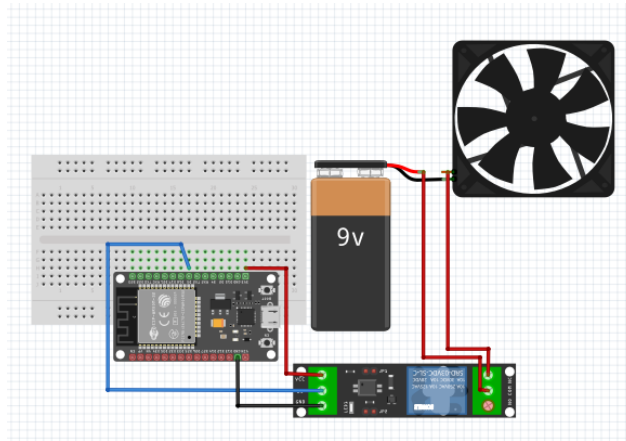
Gambar 3.10 Rancangan ESP32 dengan sensor MQ-7

3.4.3 PERANCANGAN PADA BAGIAN EXHAUST FAN

Pada tahap komponen ESP32, sebuah papan pengembangan mikrokontroler ESP32, Relay satu channel, sebuah modul relay yang digunakan untuk mengendalikan perangkat elektronik, dan Fan (kipas), sebuah kipas 12V yang berfungsi untuk mendinginkan perangkat, dijelaskan dalam deskripsi. Koneksi yang diperlukan antara komponen-komponen tersebut adalah Vcc, yang mengalirkan tegangan 5V dari ESP32 ke pin Vcc relay, GND, yang menghubungkan GND dari ESP32 ke pin GND relay, dan IN, dengan pin GPIO 16 dari ESP32 disambungkan ke pin IN relay. Terminal NO relay dihubungkan dengan kipas.

Fungsi utama dari ESP32 adalah mengontrol relay untuk menghidupkan dan mematikan kipas. Ketika ESP32 mengirimkan sinyal *HIGH* ke pin IN relay, relay akan aktif dan membuat koneksi antara terminal NO dan COM, sehingga mengalirkan arus ke kipas dan membuatnya berputar. Sebaliknya, saat ESP32 mengirimkan sinyal *LOW* ke pin IN relay, relay akan nonaktif dan memutuskan hubungan terminal NO dan COM, menghentikan aliran arus ke kipas dan membuatnya berhenti berputar. ESP32, relay, dan kipas ini termasuk pengendalian kipas dengan mengatur kecepatannya berdasarkan suhu, waktu, atau sensor lainnya, membangun sistem pendinginan untuk perangkat elektronik

seperti Raspberry Pi, Arduino, dan lainnya, serta otomasi rumah dengan mengendalikan kipas ventilasi, exhaust fan, dan sejenisnya secara otomatis. ESP32 dapat dihubungkan dengan relay dan kipas untuk mengendalikannya.



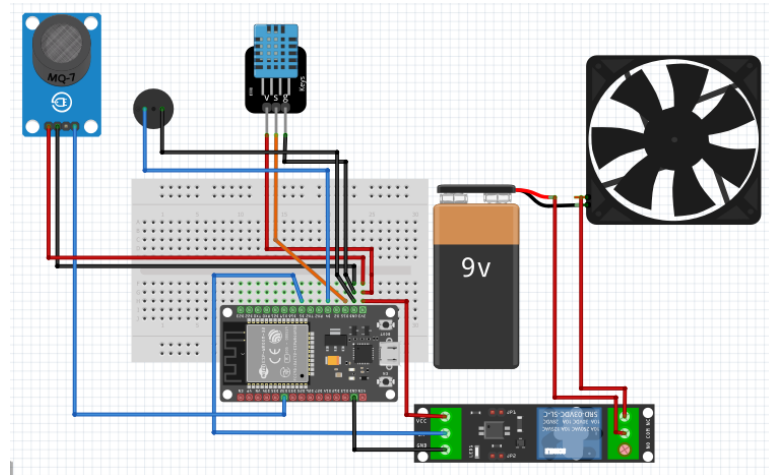
Gambar 3.11 Rancangan ESP32 dengan exhaust fan

3.4.4 PERANCANGAN KESELURUHAN ALAT

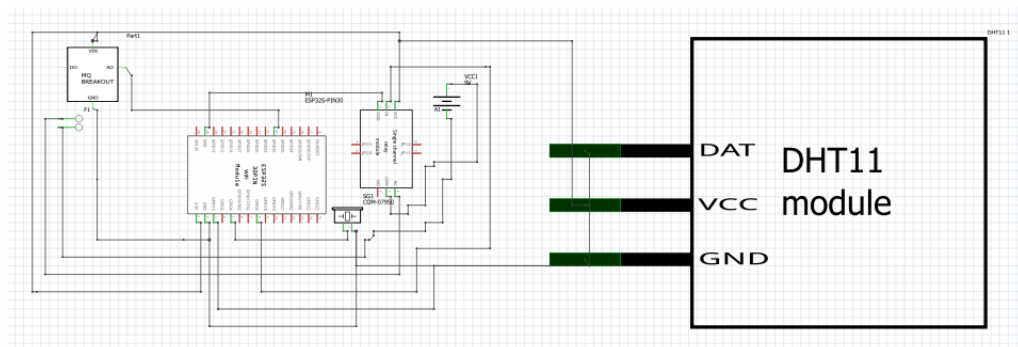
Untuk perancangan keseluruhan komponen terdiri dari sensor DHT11 yang berfungsi sebagai sensor suhu dan kelembaban, serta sensor MQ-7 untuk mendeteksi asap. Sedangkan untuk aktuator, terdapat kipas yang digunakan untuk ventilasi dan pendinginan, dan relay yang bertugas mengendalikan kipas. Mikrokontroler yang digunakan adalah ESP32, sebuah mikrokontroler yang dilengkapi dengan fitur Wi-Fi dan Bluetooth. Perangkat lunak yang digunakan mencakup Arduino IDE, yang digunakan untuk menulis dan mengunggah kode ke ESP32, dan Arduino IoT Cloud, sebuah platform cloud untuk memantau dan mengontrol perangkat IoT.

Sistem ini berfungsi untuk memantau dan mengendalikan kondisi rumah, seperti suhu, kelembaban, keberadaan asap, dan exhaust fan. Sensor DHT11 dan MQ-7 bertanggung jawab untuk membaca data tersebut, yang kemudian dikirim ke ESP32. Berdasarkan data sensor, ESP32 akan mengendalikan kipas melalui relay. Data sensor dan status kipas dapat diakses dan dikontrol melalui platform Arduino IoT Cloud. Keuntungan utama dari sistem ini adalah memungkinkan pemantauan dan kontrol *real-time* dari jarak jauh melalui koneksi internet. Selain itu, sistem juga menawarkan otomatisasi, di mana kipas dapat dihidupkan atau

dimatikan secara otomatis berdasarkan kondisi lingkungan yang terdeteksi. Fleksibilitas sistem memungkinkan penambahan sensor dan aktuator lain dengan mudah.



Gambar 3.12 Rancangan keseluruhan alat

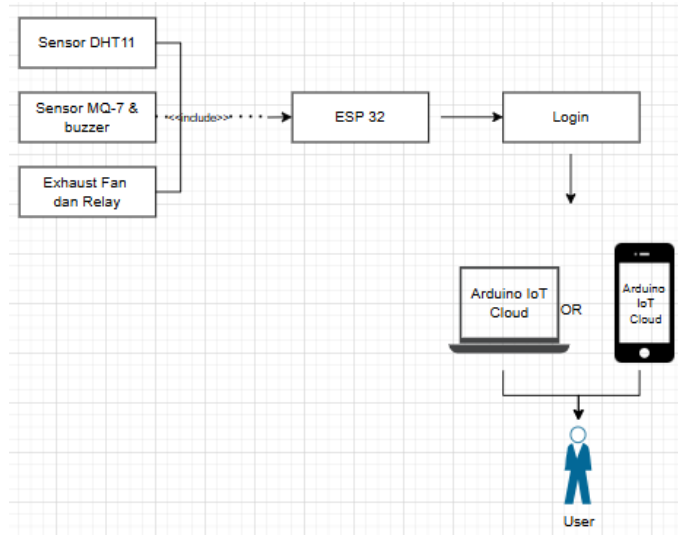


Gambar 3.13 Schematic rancangan keseluruhan

3.5 TAHAP PERANCANGAN *SOFTWARE*

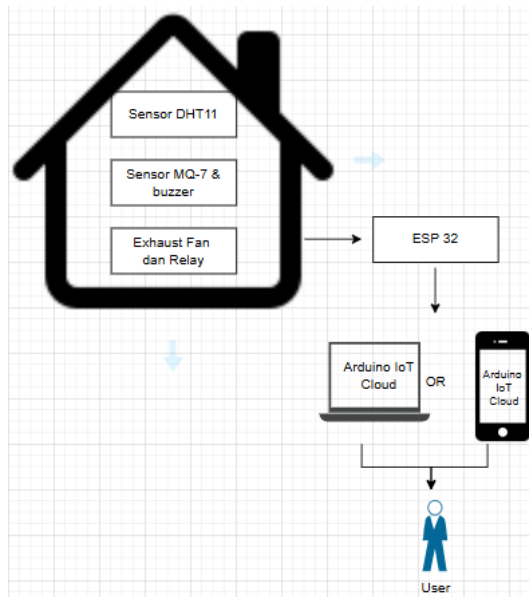
Pada tahap perancangan *software*, langkah awal melibatkan perancangan gambar *use case* yang menggambarkan interaksi antara berbagai komponen sistem, seperti sensor DHT11, sensor MQ-7, dan *exhaust fan* yang terhubung ke mikrokontroler ESP32. Selanjutnya, dalam gambar *use case* tersebut dijelaskan bahwa ESP32 harus melakukan proses *login* terlebih dahulu pada platform Arduino IoT Cloud sebelum pengguna dapat mengendalikannya. Melalui *software* ini yang dapat diakses melalui perangkat seperti laptop atau *smartphone*,

pengguna diberi kemampuan untuk memantau dan mengontrol sistem *smart home* dari jarak jauh, sesuai dengan kebutuhan dan preferensi *user*.



Gambar 3.14 Use Case Arduino IoT Cloud

Pada tahap perancangan arsitektur komunikasi data, langkah pertama adalah merancang sebuah diagram yang menggambarkan bagaimana komponen-komponen dalam sistem saling berkomunikasi. Diagram ini akan memperlihatkan bagaimana sensor DHT11, sensor MQ-7, dan *exhaust fan* terhubung ke mikrokontroler ESP32 dalam sistem smarthome. Selanjutnya, dalam diagram arsitektur komunikasi data tersebut, ESP32 akan terhubung dengan software Arduino IoT Cloud. Penting untuk dicatat bahwa software ini dapat diakses melalui berbagai perangkat, baik itu laptop maupun *smartphone*, sehingga pengguna memiliki fleksibilitas dalam memantau dan mengontrol sistem dari jarak jauh. Setelah mengaksesnya, pengguna dapat berinteraksi dengan sistem. Dalam implementasinya, *software* Arduino IoT Cloud menggunakan bahasa pemrograman C++, yang merupakan bahasa pemrograman yang umum digunakan dalam pengembangan perangkat lunak berbasis mikrokontroler seperti ESP32.



Gambar 3.15 Arsitektur Komunikasi Data *Smarthome*

3.6 TAHAP PENGUJIAN SISTEM

Pada tahap pengujian sistem Rancang Bangun *Smarthome* Berbasis *Internet of Things* Menggunakan *Software* Arduino IoT Cloud dengan Mikrokontroler ESP32 dan Sensor DHT11, Sensor MQ-7 & Buzzer, Relay, dan *Exhaust Fan*, dilakukan serangkaian pengujian untuk memverifikasi kinerja dan kelayakan sistem yang telah dirancang. Langkah awal melibatkan pengujian sensor DHT11 dan sensor MQ-7 untuk menilai akurasi dalam mengukur suhu, dan deteksi gas LPG. Proses ini melibatkan pengumpulan data dan perbandingan dengan standar atau referensi yang ada. Dan juga untuk sensor DHT11 dan MQ7 untuk mengetahui akurasi setiap sensor maka perlu adanya pengujian error. Sensor DHT11 akan dibandingkan dengan menggunakan hydrometer termometer digital, sedangkan sensor MQ7 menggunakan gas detektor sebagai pembanding.

Selanjutnya, dilakukan pengujian terhadap relay yang berfungsi mengontrol perangkat elektronik seperti *exhaust fan*. Pengujian ini mencakup dalam mengendalikan perangkat. Tahap berikutnya adalah pengujian integrasi antara semua komponen, termasuk mikrokontroler ESP32, sensor DHT11, sensor MQ-7&Buzzer, relay, dan exhaust fan. Pengujian ini bertujuan untuk memastikan keselarasan antar komponen dan operasional sistem sesuai harapan.

Selain itu, dilakukan pengujian terhadap koneksi antara ESP32 dengan software Arduino IoT Cloud, yang mencakup proses *login*, transfer data, serta kontrol dan pemantauan sistem melalui platform cloud baik pada laptop maupun *smartphone*. Hasil dari pengujian akan dievaluasi untuk menilai apakah sistem memenuhi spesifikasi yang telah ditetapkan dan apakah sistem tersebut siap untuk digunakan secara praktis. Dengan demikian, tahap pengujian sistem merupakan langkah krusial dalam memastikan bahwa sistem *smarthome* berbasis IoT yang telah direncanakan dapat beroperasi dengan baik dan dapat diandalkan dalam penggunaan sehari-hari.

3.7 TAHAP PENGAMBILAN DATA

Untuk tahap pengambilan data dilakukan proses pengumpulan data untuk menganalisis dan menguji sistem. Pertama-tama, sensor DHT11 digunakan untuk mengukur suhu di dalam rumah. Informasi yang diperoleh dari sensor ini akan memberikan gambaran tentang kondisi lingkungan dalam ruangan. Serta sensor ini akan diuji keakuratannya dengan menggunakan sensor hygrometer. Selanjutnya, sensor MQ-7 digunakan untuk mendeteksi keberadaan gas LPG di udara. Pengambilan data dari sensor ini krusial untuk memantau tingkat keamanan udara di dalam rumah. Untuk menguji keakuratan dari sensor ini sensor MQ-7 dibandingkan dengan gas detektor, dengan demikian akan mendapatkan hasil keakuratannya. Selanjutnya mengambil data pada exhaust yang mana hanya merancang supaya exhaust fan ini dapat di *on/off* (hanya sekedar menguji dapat hidup atau mati). Relay berfungsi untuk mengontrol perangkat elektronik, seperti exhaust fan, berdasarkan data yang diperoleh dari sensor DHT11 dan MQ-7. Data dari kedua sensor ini menjadi dasar dalam pengambilan keputusan untuk mengaktifkan atau menonaktifkan relay.

Data yang terkumpul dari sensor-sensor tersebut kemudian akan diproses oleh mikrokontroler ESP32. Mikrokontroler ini bertugas untuk mengambil data dari sensor, melakukan analisis data, dan mengirimkan instruksi kepada relay untuk mengendalikan perangkat elektronik. Software Arduino IoT Cloud difungsikan sebagai platform untuk mengumpulkan dan menyimpan data yang dikirimkan oleh ESP32. Dengan pengumpulan data yang tepat dan akurat dari

sensor-sensor yang terhubung pada sistem, diharapkan dapat menghasilkan sistem smarthome yang efisien serta dapat meningkatkan kenyamanan dan keamanan di dalam rumah.