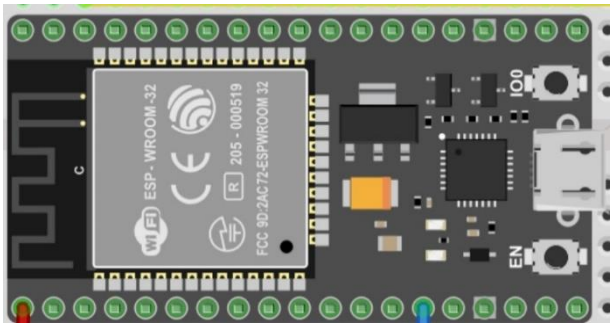
BAB 3
METODE PENELITIAN

3.1 Alat Yang Digunakan

ESP32 yang digunakan dalam sistem monitoring kapabilitas air memiliki spesifikasi ESP32-WROOM-32 untuk memenuhi kebutuhan aplikasi. Ditenagai oleh mikrokontroler *Tensilica Xtensa Dual-Core* memberikan kinerja yang handal dengan kecepatan prosesor hingga 240 MHz, memungkinkan pelaksanaan aplikasi dengan responsivitas tinggi. Kapasitas memori *flash* sebesar 4 MB dan RAM sebesar 520 KB efisien untuk menyimpan program dan mengelola data. Modul *Wi-Fi* 802.11 b/g/n dan *Bluetooth* 4.2 mendukung konektivitas nirkabel yang handal, sedangkan antena *Wi-Fi* internal memastikan kekuatan sinyal yang optimal.

ESP32 ini menyediakan sejumlah pin I/O, termasuk pin khusus untuk menghubungkan sensor kapabiltas air dan perangkat eksternal. Dengan kemampuan pengelolaan daya yang efisien melalui mode operasional dan mode *sleep*, sistem ini dapat mengoptimalkan konsumsi energi. Antarmuka sensor disesuaikan dengan sensor ketingian air yang diintegrasikan, sementara desainnya memudahkan integrasi dengan sistem pemantauan. Selain itu, ESP32 dirancang untuk terhubung dan berkomunikasi dengan Arduino *Cloud*, mempermudah pengiriman data monitoring kapabilitas air. Dimensi fisik yang ringkas menjadikannya cocok untuk integrasi pada gerbong kereta api, sesuai dengan aplikasinya dalam sistem monitoring ketinggian air.



Gambar 3. 1 ESP32

Sensor *water level* K-0135 yang memanfaatkan jalur PCB (*Printed Circuit Board*) dalam sistem monitoring kapabilitas air memiliki sejumlah spesifikasi. Desain PCB (*Printed Circuit Board*) disesuaikan dengan kebutuhan sistem, dengan jalur yang dirancang untuk memaksimalkan akurasi pengukuran kapabilitas air.

Bahan PCB (*Printed Circuit Board*) berkualitas tinggi dipilih untuk menjamin ketahanan terhadap kondisi lingkungan di dalam *tank capacity* toilet pada gerbong kereta api, termasuk resistensi terhadap korosi dan air.

Dimensi fisik sensor disesuaikan dengan kebutuhan integrasi dalam *tank capacity* air toilet gerbong kereta api, dirancang dengan bentuk yang memudahkan proses pemasangan. Material sensor dipilih dengan pertimbangan khusus untuk mempertahankan ketahanan terhadap cairan dan perubahan kondisi lingkungan yang mungkin terjadi. Jumlah jalur PCB yang memadai telah disiapkan untuk mengakomodasi koneksi sensor *water level* K-0135 dan perangkat elektronik lainnya.

Konektor yang handal hadir untuk menyederhanakan proses penghubungan antara sensor dan sistem pemantauan, sekaligus menjamin keandalan transmisi data. Implementasi pemrosesan sinyal yang efisien dilakukan untuk mengubah data dari sensor menjadi informasi yang dapat dimengerti oleh mikrokontroler, seperti ESP32. Terdapat pula fitur proteksi terhadap gangguan, seperti filter untuk mengurangi *noise* atau interferensi yang dapat memengaruhi akurasi pengukuran.

Selain itu, sensor ini memiliki kompatibilitas yang optimal dengan sistem pemantauan ketinggian air yang menggunakan ESP32, memungkinkan integrasi yang lancar dan efisien. Dengan kombinasi fitur-fitur ini, diharapkan sensor dapat memberikan kinerja yang unggul dalam mengukur ketinggian air pada gerbong kereta api.

Gambar 3. 2 Sensor *Water Level* K-0135

Integrasi ESP32 dengan Arduino *cloud* memerlukan konfigurasi perangkat keras yang melibatkan beberapa komponen dan pengaturan khusus. Penggunaan modul ESP32 dalam bentuk *development board*, seperti NodeMCU atau Wemos D1 Mini, menjadi inti perangkat keras yang esensial. Pastikan modul ini memiliki dukungan *Wi-Fi* agar dapat terhubung secara langsung ke Arduino *cloud*. Sensor *water level* yang diintegrasikan dengan ESP32 harus dipilih dengan cermat, memastikan kemampuannya untuk memberikan data yang akurat dan kemudahan dalam penghubungannya dengan ESP32. Persiapkan jalur PCB dengan desain yang optimal untuk menghubungkan sensor *water level* K-0135 dan ESP32, memastikan transfer data yang efisien antara sensor dan mikrokontroler. Jika perangkat akan digunakan di area dengan sinyal *Wi-Fi* yang kurang optimal, pertimbangkan penggunaan antena *Wi-Fi* eksternal untuk meningkatkan jangkauan dan stabilitas koneksi.

Pastikan ketersediaan konektor dan kabel yang diperlukan untuk menghubungkan sensor *water level*, ESP32, dan komponen lainnya, dengan memilih konektor yang handal dan kabel yang sesuai panjangnya. Sediakan sumber daya listrik yang stabil untuk ESP32 dan sensor, dapat berupa baterai isi ulang atau sumber daya listrik eksternal, disesuaikan dengan kebutuhan dan kondisi penggunaan. Pastikan koneksi internet stabil di lokasi pemasangan ESP32, baik melalui *Wi-Fi* maupun menggunakan modul SIM untuk konektivitas seluler. Daftarkan akun di Arduino *cloud* dan peroleh token otentikasi yang diperlukan untuk mengonfigurasi ESP32 agar dapat terkoneksi ke *cloud*. Instal Arduino IDE di komputer pengembangan dan unduh serta pasang *library* yang diperlukan untuk ESP32 dan Arduino *cloud*, seperti ArduinoOTA dan ArduinoIoT*Cloud*, guna mendukung fungsionalitas yang diinginkan.

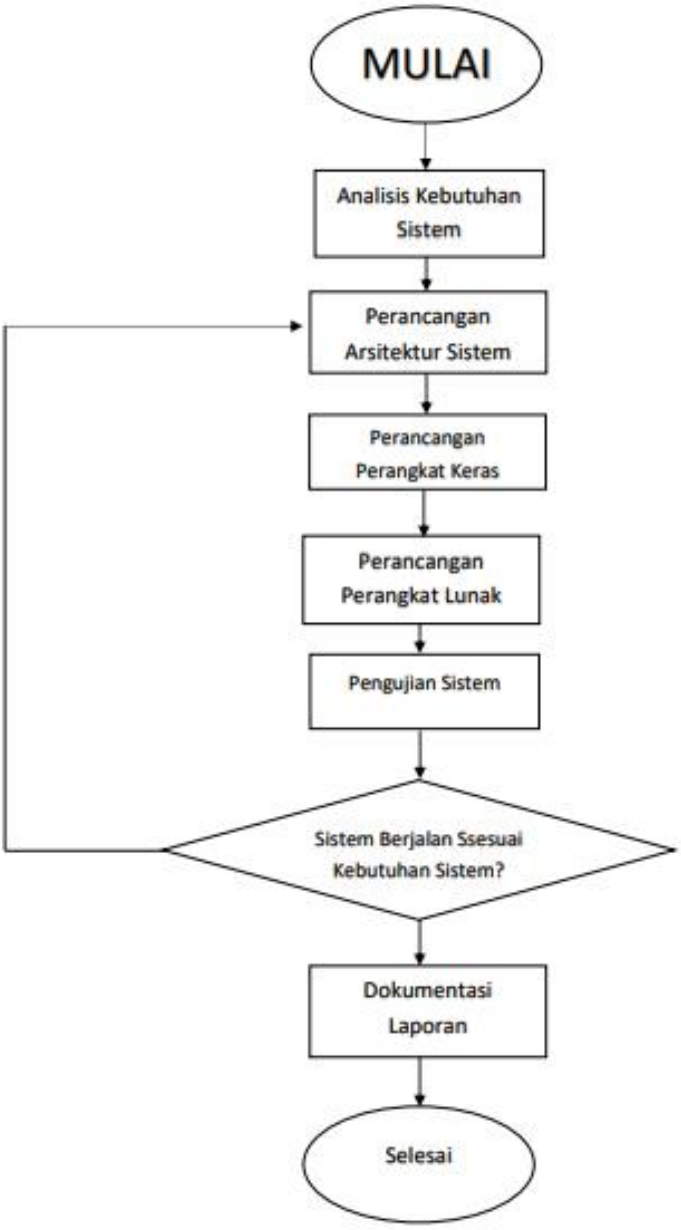
Gambar 3. 3 Tampilan *Widget Arduino Cloud*

3.2 Alur Penelitian

Penelitian ini dimulai dengan tahap analisis kebutuhan sistem, yang bertujuan untuk mengidentifikasi kebutuhan dari sistem yang akan dikembangkan. Setelah tahap analisis, langkah berikutnya adalah merancang arsitektur sistem berdasarkan hasil analisis kebutuhan. Proses perancangan kemudian melibatkan perancangan perangkat keras, termasuk desain perangkat IoT (*Internet of Thigs*) seperti sensor dan mikrokontroler yang digunakan pada implementasi sistem monitoring kapabilitas air pada *tank capacity* toilet gerbong kereta api dengan ESP32 berbasis sensor *water level* dan Arduino *Cloud*. Perancangan perangkat keras ini berfokus pada rancangan perangkat IoT.

Selanjutnya, tahap perancangan perangkat lunak berkaitan dengan pengembangan protokol sensor *water level* untuk mentransmisikan data ke Arduino *Cloud* sebagai antarmuka tempat menampilkan informasi. Setelah perancangan, proses implementasi dilakukan untuk membangun sistem kontrol dan monitoring berdasarkan tahap-tahap sebelumnya. Kualitas implementasi ini sangat menentukan kinerja sistem secara keseluruhan. Setiap komponen, mulai dari perangkat keras hingga perangkat lunak, harus diintegrasikan dengan cermat untuk mencapai tujuan monitoring kapabilitas air dengan efektif.

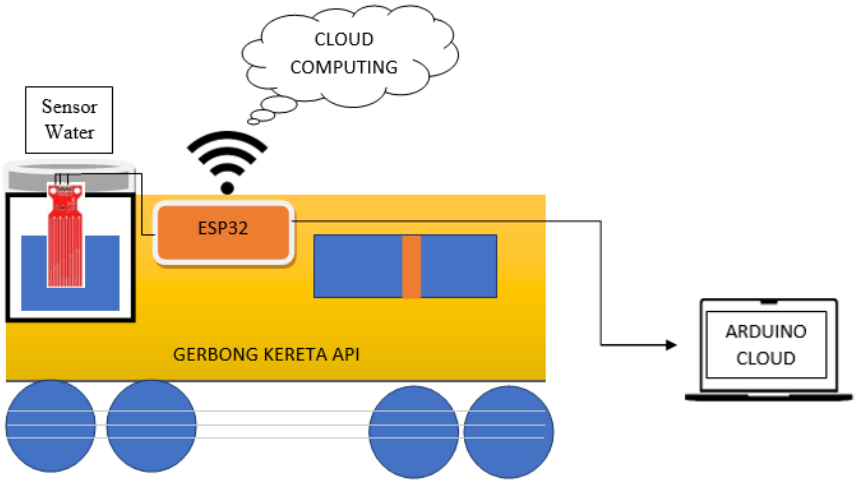
Tahap berikutnya adalah pengujian sistem, proses ini mencakup pengujian sensor *water level* untuk memastikan keakuratannya, evaluasi kinerja protokol sensor *water level*, dan pengujian terhadap platform Arduino *Cloud*. Hasil dari pengujian ini menjadi penentu apakah sistem telah memenuhi kebutuhan yang telah ditetapkan sejak awal. Jika ada ketidaksesuaian, penelitian akan kembali ke tahap perancangan arsitektur sistem untuk melakukan perbaikan dan penyempurnaan. Namun, jika hasil sesuai, proses dilanjutkan ke tahap dokumentasi laporan. Tahap ini melibatkan penyusunan laporan secara komprehensif yang mencakup semua aspek penelitian, mulai dari analisis kebutuhan, perancangan arsitektur sistem, implementasi, hingga hasil pengujian. Seluruh alur tahap penelitian ini dapat dilihat dalam Gambar 3.4 Alur Penelitian, memberikan pandangan menyeluruh mengenai proses penelitian yang dilakukan serta memberikan gambaran keseluruhan alur sistem ini dapat memberikan gambaran yang jelas tentang bagaimana teknologi IoT (*Internet of Things*) digunakan untuk meningkatkan efisiensi dan monitoring *tank capacity* toilet pada gerbong kereta api. Dengan memanfaatkan teknologi ini, informasi mengenai kapabilitas air dapat diperoleh dengan lebih mudah dan akurat, memastikan pengelolaan sumber daya air yang lebih efisien dalam konteks transportasi publik.



Gambar 3. 4 Alur Penelitian

3.3 Perancangan Perangkat Keras

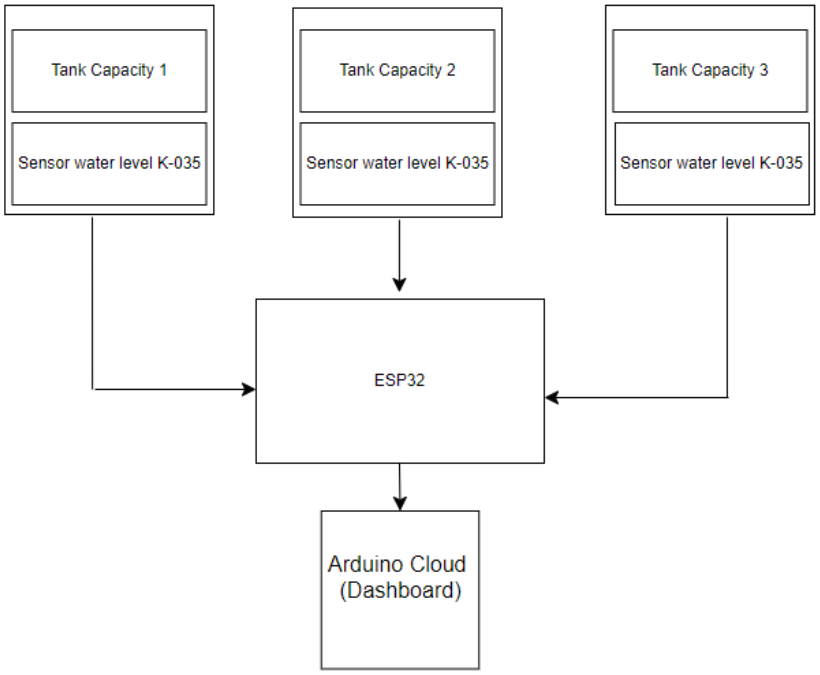
Pada tahap perancangan perangkat keras, akan dilakukan perancangan arsitektur sistem dan gambaran rangkaian ESP32 beserta sensor *water level*. Ilustrasi arsitektur sistem yang akan dikembangkan dapat ditemukan pada Gambar 3.5 Arsitektur Sistem.



Gambar 3. 5 Arsitektur Sistem

Dalam analisis gambar 3.5 Arsitektur Sistem, terlihat jelas bahwa alur sistem dimulai dari pipa distribusi yang mengalirkan air ke setiap *tank capacity* toilet yang terdapat pada gerbong kereta api. Fokus utama dari alur ini adalah memastikan bahwa kapabilitas air pada setiap toilet dapat dipantau dan dikontrol dengan efektif. Proses ini mencakup dua tahap utama yang saling terkait, yaitu pemanfaatan Arduino *Cloud* sebagai sistem informasi dan pemasangan sensor serta mikrokontroler pada miniatur *tank capacity* toilet.

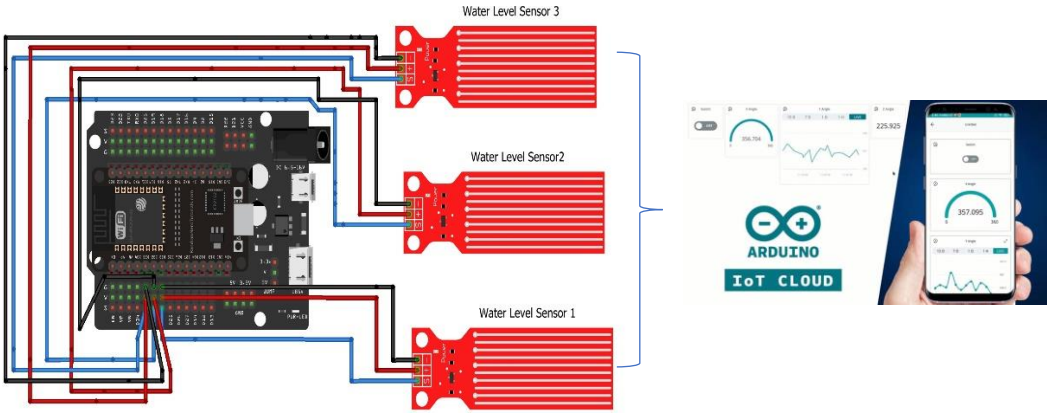
Sensor *water level* K-0135 dihubungkan ke ESP32 untuk mendapatkan informasi kapabilitas air *tank capacity* toilet pada gerbong kereta api menggunakan sensor *water level* K-0135. Dari data ini akan didapatkan jumlah kapabilitas air *tank capacity* toilet pada gerbong kereta api menggunakan sensor *water level* K-0135, serta dapat digunakan untuk melakukan pengecekan jumlah air yang berada pada *tank capacity* toilet pada gerbong kereta api dengan batasan jumlah air yang sudah ditentukan. Memasangkan ESP32 dan sensor *water level* K-0135 sebanyak tiga buah pada *tank capacity* toilet gerbong kereta api yang terhubung pada *software Arduino Cloud*. Platform ini memiliki antarmuka pengguna yang ramah pengguna, memungkinkan pengguna untuk mengelola perangkat, melihat data, dan mengonfigurasi pengaturan dengan mudah melalui *web browser*.



Gambar 3. 6 Blok Diagram Sistem

Sensor dan mikrokontroler dipasang pada miniatur *tank capacity* toilet pada gerbong kereta api. Proses ini bertujuan untuk memonitor ketinggian air pada setiap *tank capacity* toilet. Data mengenai jumlah kapabilitas air dari setiap *tank capacity* toilet akan dihubungkan ke *Arduino Cloud*. *Arduino Cloud*, sebagai sistem informasi, akan menampilkan data kapabilitas air yang digunakan oleh setiap *tank capacity* toilet pada gerbong kereta api. Selain itu, *Arduino Cloud* juga akan terhubung ke mikrokontroler yang terdapat pada *tank capacity* toilet gerbong dua untuk melakukan kontrol dan monitoring secara efisien.

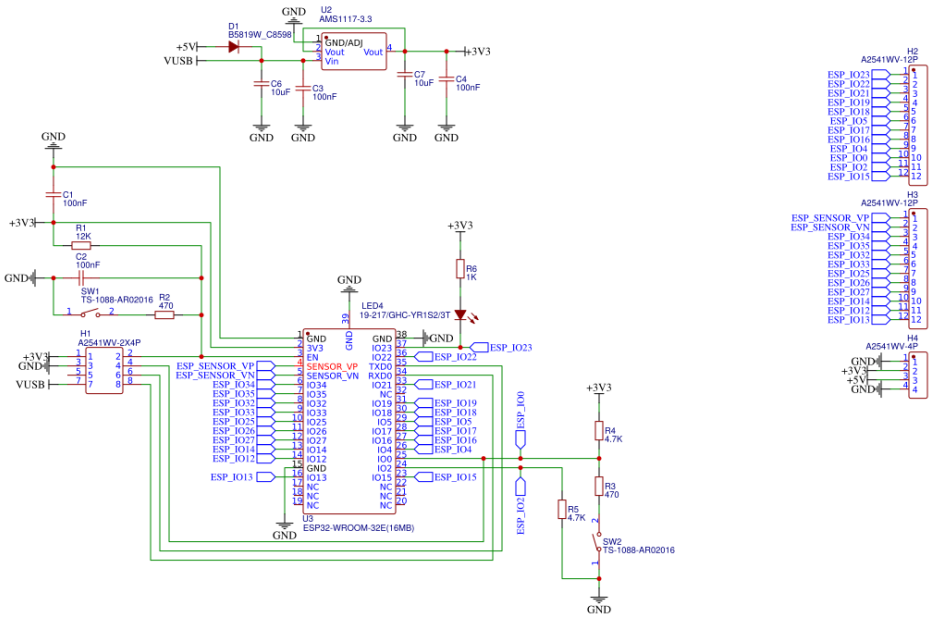
Seluruh sistem ini dirancang untuk beroperasi secara terintegrasi, memastikan bahwa informasi kapabilitas air dapat diakses dengan cepat dan akurat. Penggunaan Arduino *Cloud* sebagai pusat informasi memberikan keleluasaan dalam mengakses data, sehingga dapat diakses dari berbagai lokasi secara *real-time*. Selain itu, pemasangan sensor dan mikrokontroler pada setiap *tank capacity* toilet memungkinkan pengumpulan data yang spesifik dan terperinci, memberikan pemahaman mendalam tentang penggunaan air pada masing-masing toilet. Rancangan ESP32 beserta sensor *water level* K-0135 dapat dilihat pada gambar berikut:



Gambar 3. 7 Rancangan ESP32 Beserta Sensor *Water Level*

ESP32 memulai deteksi sensor *water level* K-0135 dengan mengukur tegangan analog yang dihasilkan oleh sensor. Inisialisasi ESP32 dan konfigurasi pin untuk membaca nilai analog dari sensor merupakan langkah pertama dalam proses ini. Kemudian, ESP32 membaca tegangan analog yang terukur di titik tengah rangkaian pembagi tegangan yang terhubung dengan sensor. Nilai tegangan analog tersebut dikonversi menjadi nilai digital melalui Analog-*to*-Digital *Converter* (ADC) pada ESP32. Nilai digital yang diperoleh dari konversi ini diinterpretasikan sebagai nilai resistansi sensor, yang kemudian diubah menjadi level air yang terukur sesuai dengan karakteristik sensor yang telah diketahui sebelumnya. Hasil level air yang terdeteksi dapat dimonitor melalui antarmuka serial atau aplikasi yang sesuai, dan ESP32 dapat diatur untuk mengambil tindakan tertentu berdasarkan level air yang terdeteksi, seperti memberikan peringatan atau mengontrol ketersdian air. Dengan proses ini, ESP32 dapat efektif dalam mendeteksi sensor water level K-0135 dan mengambil tindakan yang sesuai berdasarkan data yang diperoleh.

Board shield ESP32 digunakan untuk memperluas fungsionalitas mikrokontroler ESP32 dengan mudah. *Shield* ini menambahkan modul seperti sensor, aktuator, atau komunikasi tanpa banyak pengkabelan. Misalnya, sensor *shield* untuk memantau lingkungan, motor *shield* untuk kontrol robotika, *atau relay shield* untuk mengendalikan perangkat listrik. Pemasangannya sederhana, cukup dipasang di atas ESP32, dan diprogram menggunakan IDE seperti Arduino atau ESP-IDF. Shield ini mempercepat prototyping dan pengembangan aplikasi IoT dengan menyediakan *plug-and-play* fungsionalitas tambahan.



Gambar 3. 8 Skematik *Board Shield* ESP32

Sensor *water level* K-0135 mengonversi data dengan mengubah perubahan level air menjadi perubahan resistansi. Sensor ini bekerja berdasarkan prinsip resistif, di mana resistansinya berubah sesuai dengan ketinggian air. Untuk membaca resistansi ini, sensor dihubungkan dalam rangkaian pembagi tegangan bersama dengan sebuah resistor tetap, yang kemudian dihubungkan ke sumber tegangan dan *ground*. Tegangan yang dihasilkan pada titik tengah rangkaian, yang dipengaruhi oleh perubahan resistansi sensor, diukur oleh pin ADC pada mikrokontroler ESP32. Tegangan ini kemudian diubah menjadi nilai digital oleh ADC ESP32. Nilai digital yang dihasilkan ini diinterpretasikan sebagai nilai resistansi sensor. Berdasarkan nilai resistansi ini, level air yang terukur dapat dihitung menggunakan karakteristik sensor yang telah diketahui sebelumnya. Mikrokontroler ESP32 kemudian memproses data resistansi ini untuk menentukan level air dan mengambil tindakan yang diperlukan, seperti menampilkan data melalui antarmuka serial atau aplikasi untuk pemantauan *real-time* dan pengambilan keputusan. Dengan demikian, sensor *water level* K-0135 mengonversi perubahan level air menjadi data yang dapat diolah oleh ESP32 untuk mendeteksi dan memantau level air.

Untuk mendeteksi resistansi dari sensor *water level* K-0135 menggunakan mikrokontroler ESP32, dapat digunakan rangkaian pembagi tegangan. Dalam rangkaian ini, sensor dihubungkan secara seri dengan sebuah resistor tetap, dan

kedua ujung rangkaian ini dihubungkan ke sumber tegangan (seperti 3.3V dari ESP32) dan *ground*. Tegangan di titik tengah rangkaian pembagi ini, yang dipengaruhi oleh perubahan resistansi sensor, dihubungkan ke salah satu pin ADC pada ESP32. Melalui fungsi pembacaan analog pada ESP32, tegangan ini dibaca dan kemudian dikonversi menjadi nilai resistansi sensor. Dengan menulis program untuk membaca nilai ADC, menghitung tegangan yang terukur, dan kemudian menghitung resistansi sensor, dapat dipantau perubahan resistansi dari sensor *water level* K-0135. Program tersebut akan menampilkan nilai resistansi yang terhitung melalui komunikasi serial, memungkinkan monitoring level air secara *real-time* berdasarkan perubahan resistansi yang terdeteksi oleh sensor.

Pada penelitian ini terdapat tiga keterangan pada sensor antarai lain adalah S, +, dan -, dimana Simbol "S" pada sensor *water level* K-0135 biasanya mengacu pada pin keluaran analog yang akan dihubungkan ke salah satu masukan analog ESP32, kabel yang terhubung pada simbol "S" pada penelitian ini pin A0, A1, dan A2 pada sensor *water level* K0135 mengukur level air pada berbagai titik. Pin A0 menghubungkan sensor *water level* 1, pin A1 menghubungkan sensor *water level* 2, dan pin A2 menghubungkan sensor *water level* 3. Pin ini menghasilkan nilai analog berdasarkan ketinggian air yang terdeteksi. Nilai-nilai ini kemudian diteruskan ke pin analog pada ESP32, yaitu pin 33, 32, dan 35. Pada tanda "+" pada sensor *water level* K-0135 menandakan terminal positif atau terminal daya. Ini adalah tempat di mana tegangan daya yang diperlukan untuk mengoperasikan sensor disediakan. Misalnya, jika sensor membutuhkan tegangan 5V, maka kabel positif dari sumber daya harus terhubung ke terminal "+" pada sensor, sedangkan Tanda "-" pada sensor *water level* menandakan terminal negatif atau *ground*. Ini adalah tempat di mana kabel *ground* atau tanah dari sumber daya harus terhubung untuk melengkapi sirkuit listrik dan memastikan koneksi yang baik serta stabilitas operasi sensor.

Tabel 3. 1 Data Skema Sensor *Water Level* K-0135 Dan ESP32

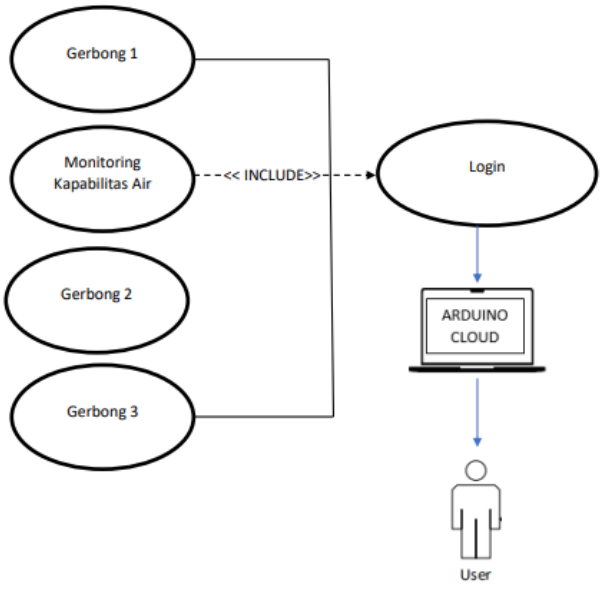
Sensor *water level* K0135	ESP32
A0	33
A1	32
A2	35

Gambar 3. 9 Sensor *Water Level* K-0135

3.4 Perancangan Perangkat Lunak

Pada fase perancangan perangkat lunak, dilakukan desain menggunakan *software* Arduino *Cloud*. Selain melakukan desain dengan Arduino *Cloud*, tahap ini juga melibatkan pemrograman pada mikrokontroler dan desain alur komunikasi data antara perangkat keras dan perangkat lunak sistem. Dalam hal ini, terdapat kebutuhan untuk menyelaraskan secara efisien antara perangkat keras dan perangkat lunak agar dapat berkomunikasi dengan lancar dan akurat.

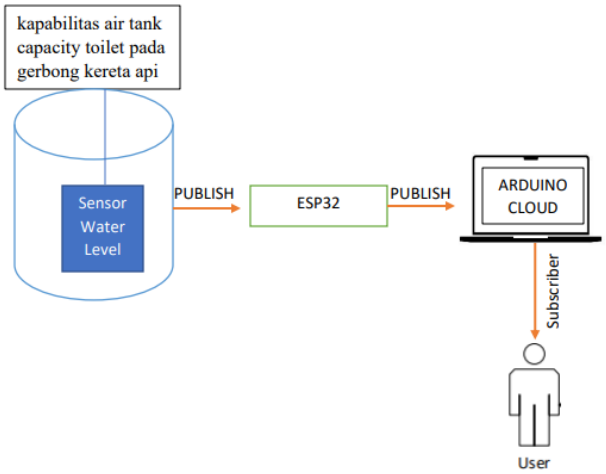
Use case pada sistem monitoring berbasis *Arduino Cloud* yang akan dibangun yaitu *user* dapat melakukan aktivitas monitoring kapabilitas air pada tiga *tank capacity* toilet pada gerbong kereta api. Aktivitas tersebut hanya dapat dilakukan setelah *user* melakukan aktivitas *login*. Adapun rancangan *use case web* Arduino *Cloud* yang akan dibangun dapat dilihat pada Gambar 3.8 *Use Case* Arduino *Cloud*.



Gambar 3. 10 *Use Case* Arduino *Cloud*

Selain merancang perangkat yang terhubung ke Arduino *Cloud*, pada fase ini juga akan dilakukan pemrograman pada mikrokontroler untuk mengambil data nilai dari sensor yang digunakan dan membuat keputusan untuk memonitor kapabilitas air. Selain itu, mikrokontroler akan diprogram agar dapat terhubung ke jaringan internet dan mengirimkan data yang telah diperoleh ke Arduino *Cloud*. Proses pemrograman mikrokontroler ini akan dilakukan menggunakan aplikasi Arduino IDE dengan bahasa pemrograman C++ dan algoritma *switch case*. Algoritma *switch case* akan digunakan untuk membuat keputusan berdasarkan nilai variabel, memberikan opsi yang lebih jelas dan terorganisir daripada serangkaian pernyataan *if-else* yang panjang. Dalam konteks Arduino, penggunaan *switch case* umumnya dilakukan untuk mengevaluasi nilai dari suatu variabel dan menjalankan blok kode yang sesuai dengan nilai tersebut.

Dalam proses arsitektur komunikasi data sensor *water level*, mengikuti alur yang dimulai dari sensor *water level* yang menggunakan jalur PCB. Ketika sensor menyentuh air, data kapabilitas air yang diperoleh dikirimkan (dipublikasikan) ke ESP32 untuk kemudian ditampilkan pada Arduino *Cloud*. Sistem informasi ini menggunakan Arduino *loud* sebagai media penyimpanan untuk status kapabilitas air yang tercatat. Detail dari alur komunikasi data sensor *water level* melalui jalur PCB dapat ditemukan dalam Gambar 3.9 Arsitektur Komunikasi Data Sensor *Water Level*.



Gambar 3. 11 Arsitektur Komunikasi Data Sensor *Water Level*

3.5 Implementasi

Pada fase implementasi, terdapat tiga langkah yang akan dilakukan, yakni:

1. Menyusun perangkat keras, yang melibatkan penggabungan mikrokontroler ESP32 dan sensor *water level* K-0135 menjadi satu kesatuan perangkat keras sistem. Penyusunan ini disesuaikan dengan desain yang telah dibuat pada tahap perancangan perangkat keras sistem.
2. Membangun sistem pemantauan berbasis *web* dengan memanfaatkan Arduino *Cloud*. Pengembangan ini melibatkan pemrograman menggunakan bahasa C++.
3. Membangun saluran komunikasi data antara sistem *web* dan perangkat keras yang telah disusun sebelumnya. Protokol komunikasi data yang digunakan adalah berdasarkan protokol sensor *water level*.

3.6 Pengujian Dan Evaluasi Sistem

Pada fase pengujian dan evaluasi sistem, dilakukan uji coba pada kedua aspek, yakni perangkat keras dan perangkat lunak. Uji coba pada sisi perangkat keras dilakukan untuk mengevaluasi kinerja perangkat keras yang terdiri dari mikrokontroler ESP32 dan sensor *water level* K-0135. Sementara itu, uji coba pada sisi perangkat lunak dilakukan untuk menguji kinerja sistem informasi berbasis *web* dan memeriksa efektivitas protokol sensor *water level* sebagai bagian dari mekanisme pengiriman data.