

IMPLEMENTASI ESP32 CAM DAN KODULAR BERBASIS ANDROID UNTUK MONITORING SMART GARDEN

Ine Agustine Cahyaningtyas, Arnisa Stefanie, Ibrahim

Program Studi Teknik Elektro S1, Fakultas Teknik
Universitas Singaperbangsa Karawang, Karawang, Indonesia
Ine.agustine19067@student.unsika.ac.id

ABSTRAK

Penelitian ini bertujuan untuk mengimplementasikan alat *monitoring smart garden* menggunakan perangkat ESP32 Cam dan platform Kodular berbasis Android. *Smart garden* atau kebun pintar adalah konsep gabungan antara teknologi dan pertanian, yang memanfaatkan teknologi untuk mengontrol dan memantau pertumbuhan tanaman. Metode perancangan alat ini melibatkan penggunaan ESP32 Cam untuk merekam video secara *real time* dan mengirimkannya melalui jaringan WiFi. Platform Kodular digunakan untuk membuat aplikasi Android yang memungkinkan pengguna memonitor tanaman melalui *live streaming* video dari ESP32 Cam. Hasil pengujian menunjukkan bahwa sistem *monitoring smart garden* menggunakan ESP32 Cam dan Kodular berhasil berjalan dengan baik. Fitur *live streaming* video dari ESP32 Cam dapat diakses melalui aplikasi Android yang dibuat dengan kodular. Pembudidaya dapat memantau pertumbuhan tanaman secara *real time*.

Kata kunci: ESP32 Cam, Kodular, Smart Garden

1. PENDAHULUAN

Kemajuan teknologi adalah hal yang tak dapat dihindari dalam kehidupan kita. Ini karena perkembangan teknologi sejalan dengan kemajuan ilmu pengetahuan. Setiap kali ada inovasi, tujuannya adalah memberikan manfaat positif bagi manusia. Teknologi juga membawa banyak kemudahan dan menjadi cara baru dalam melakukan berbagai aktivitas. Manusia telah merasakan banyak manfaat dari berbagai inovasi teknologi yang muncul dalam dekade terakhir. Dengan kemajuan teknologi yang semakin cepat, orang terus berinovasi, termasuk dalam menciptakan sistem kebun pintar [1].

Kebun pintar atau *smart garden* adalah gabungan antara teknologi dan tanaman yang dikendalikan oleh komputer. Di zaman sekarang, pertanian telah ditingkatkan dengan bantuan komputer dan perangkat elektronik selama bertahun-tahun. Meskipun masih ada beberapa tantangan kecil, *smart garden* membawa manfaat seperti memberikan pengingat kapan waktu yang tepat untuk memberi nutrisi, melakukan penyiraman, dan jenis perawatan lainnya kepada pengguna [2]. Manfaat lain yang dapat dirasakan oleh para pembudidaya dengan adanya *smart garden* adalah para pembudidaya dapat memonitor tanamannya dengan memanfaatkan aplikasi yang dapat diakses secara *real time live streaming* menggunakan perangkat android. Ini memungkinkan pembudidaya untuk mengambil tindakan yang diperlukan dengan cepat dan tepat saat ada masalah pada tanaman.

Dari penjelasan diatas maka peneliti terinspirasi untuk melakukan sebuah inovasi pembuatan alat *monitoring smart garden* dengan memanfaatkan ESP32 Cam dan Kodular dalam pengaplikasiannya. ESP32 Cam merupakan salah satu mikrokontroler yang memiliki fasilitas tambahan berupa bluetooth, WiFi, kamera, bahkan sampai ke slot mikro SD [3]. Dalam penelitian ini ESP32 Cam menjadi pilihan yang tepat untuk mengembangkan alat *monitoring smart*

garden. Dengan bantuan kamera yang dimiliki oleh ESP32 Cam, alat *monitoring* ini dapat menghasilkan *output* berupa video *live streaming* yang bisa diakses oleh pengguna melalui perangkat Android.

Kodular adalah situs web yang menyediakan *tools* atau alat untuk membuat aplikasi Android dengan menggunakan *block programming* [4]. *Block programming* adalah metode pengembangan aplikasi dengan menyusun blok-blok kode yang terhubung satu sama lain, sehingga memudahkan dalam proses pembuatan aplikasi tanpa harus menulis kode secara manual. Dengan Kodular, pembuatan aplikasi untuk mengakses dan memonitor *smart garden* dengan fitur *live streaming* dari ESP32 Cam dapat dilakukan dengan lebih mudah dan cepat.

Dengan menggabungkan teknologi ESP32 Cam dan Kodular, peneliti akan mencoba untuk menghasilkan sebuah alat *monitoring smart garden* yang dapat memberikan manfaat dan kemudahan bagi para pembudidaya. Alat ini diharapkan dapat membantu meningkatkan efisiensi dan produktivitas dalam bercocok tanam serta memberikan solusi untuk mengatasi masalah yang mungkin muncul dalam perawatan tanaman.

2. TINJAUAN PUSTAKA

2.1. Smart Garden

Smart garden berasal dari bahasa Inggris yang artinya kebun atau taman pintar, yaitu sebuah rancangan sistem yang dibuat untuk memudahkan pekerjaan dalam hal perawatan tanaman [5]. Atau dengan kata lain *Smart garden* adalah aplikasi sistem kendali dan monitoring penyiraman/perawatan tanaman yang memanfaatkan teknologi cerdas [6].

2.2. Kodular

Kodular adalah sebuah platform daring yang menyajikan alat untuk menciptakan aplikasi Android menggunakan prinsip *block programming* dengan

metode *drag and drop*. Fitur utama Kodular adalah *block programming*, yang memungkinkan kita untuk menghindari keharusan menulis kode program secara manual dalam pembuatan aplikasi Android [7].

2.3. Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi *middleware*, dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Android adalah sistem operasi yang menghidupkan lebih dari satu miliar *smartphone* dan tablet [4]. Dalam penelitian ini, peneliti menggunakan versi Android 11 (*Red Velvet Cake*).

2.4. ESP32 CAM dan Modul kamera

ESP32 *Cam* fungsinya sebagai mikrokontroler yang dapat terhubung ke *WiFi* sebagai mikrokontroler ini akan bersistem *Internet Of Things*.



Gambar 1. ESP32 *cam* dan modul kamera

Modul kamera pada ESP32 *Cam* berfungsi untuk merekam video secara *real time* dari *Smart Garden*. Modul kamera ini memungkinkan pengambilan video dari lingkungan taman dan kemudian mengirimkan data tersebut ke mikrokontroler ESP32 *Cam*. Data video yang dihasilkan oleh modul kamera akan digunakan untuk memantau keadaan *Smart Garden* dengan lebih detail. Melalui koneksi *WiFi*, data video ini akan dikirimkan ke aplikasi di *smartphone*, sehingga pengguna dapat melihat kondisi taman pintar secara langsung.

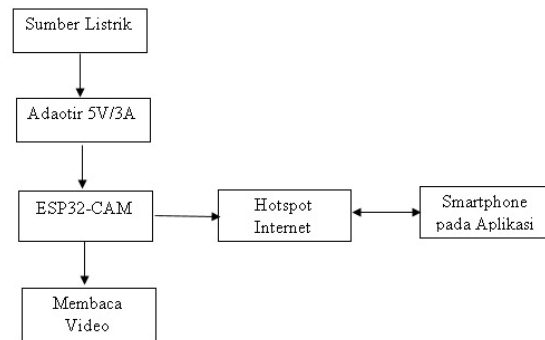
3. METODE PENELITIAN

3.1. Perancangan Sistem

Dalam penelitian ini, perancangan alat pemantauan *smart garden* menggunakan ESP32 *Cam* memiliki peran yang sangat krusial. Untuk mencapai hasil terbaik, diperlukan perencanaan sistem yang tepat, mempertimbangkan semua komponen alat dan aspek pemrograman. Keakuratan dalam perencanaan menjadi kunci karena langkah pertama dalam merancang sistem menentukan hasil akhir dari proses keseluruhan. Perencanaan sistem pemantauan *smart garden* meliputi beberapa tahap, seperti pembuatan blok diagram dan *flowchart*.

3.1.1. Blok Diagram

Blok diagram digunakan untuk memberikan gambaran visual tentang bagaimana komponen utama dalam sistem berinteraksi satu sama lain. Dalam contoh blok diagram, ESP32 *Cam* berfungsi sebagai mikrokontroler yang mengendalikan kamera, *Wifi* berperan dalam menyediakan koneksi internet, dan Android adalah aplikasi di perangkat seluler yang digunakan untuk memonitor *smart garden*.



Gambar 2. Blok diagram dan blok diagram alir

Penjelasan blok diagram pada perancangan alat di atas sebagai berikut:

1. Tenaga listrik
Berfungsi sebagai sumber tenaga (*power*) untuk komponen rangkaian alat.
2. Adaptor *power supply*
Berfungsi untuk mengubah bentuk tegangan AC yang bersumber dari arus listrik menjadi bentuk tegangan DC dengan nilai 5V dan memberikan tegangan yang dibutuhkan pada setiap rangkaian.
3. ESP32 *Cam*
Berfungsi sebagai mikrokontroler yang dapat terhubung ke *WiFi*
4. Hotspot Internet
Berfungsi sebagai jaringan *WiFi*
5. *Smartphone* pada Aplikasi
Berfungsi sebagai pemantau keadaan *smart garden* melalui aplikasi
6. Membaca Video
Aplikasi android pada *smartphone* membaca video yang diterima dan menampilkan video secara *live streaming* kepada pengguna.

Batasan aplikasi *Smart garden* dengan sistem yang dikembangkan:

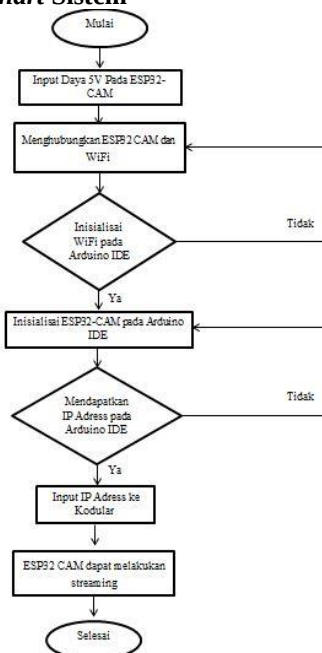
1. Jarak *WiFi* : Sistem ini akan berfungsi dengan baik dalam jangkauan jaringan *WiFi*. Jika jarak antara ESP32 *Cam* dan *router WiFi* terlalu jauh atau ada hambatan fisik, kualitas koneksi dan *live streaming* video dapat terpengaruh.
2. Jaringan : ESP32 *Cam* hanya dapat diakses dalam satu jaringan.
3. Kualitas koneksi internet: Kualitas koneksi internet akan mempengaruhi kualitas *live streaming* video. Jika koneksi internet tidak stabil atau lambat, video mungkin mengalami *buffering* atau tidak dapat diputar secara lancar.

4. Keamanan: Karena sistem ini menggunakan jaringan *WiFi*, perlu memastikan bahwa jaringan *WiFi* aman dan dilindungi dengan kata sandi yang kuat untuk mencegah akses yang tidak sah ke sistem.

Dengan demikian, blok diagram pada perancangan alat *monitoring smart garden* tersebut menunjukkan hubungan dan fungsi masing-masing komponen. Tenaga listrik sebagai sumber daya utama disediakan oleh adaptor *power supply* untuk memasok tegangan DC yang dibutuhkan. ESP32 *Cam* berperan sebagai otak utama alat yang mengendalikan kamera yang terhubung ke *WiFi* untuk berkomunikasi dengan aplikasi di *smartphone* pengguna. *Smartphone* sebagai pemantau keadaan tanaman yang memiliki peran penting dalam menerima data *monitoring smart garden*. Dengan keterhubungan dan keterpaduan antara komponen-komponen ini, alat *monitoring smart garden* diharapkan dapat berfungsi dengan baik untuk memberikan pemantauan secara efisien.

Dalam konteks penelitian ini, blok diagram digunakan untuk menggambarkan bagaimana komponen-komponen utama dalam sistem (seperti ESP32 *Cam*, *WiFi*, dan Android App) saling berinteraksi dan berhubungan satu sama lain dalam pembuatan alat *monitoring smart garden*.

3.1.2 Flowchart Sistem



Gambar 3. Flowchart sistem

Flowchart berperan dalam menggambarkan bagaimana algoritma atau proses berjalan, dengan menunjukkan langkah-langkahnya melalui simbol kotak atau urutan yang dihubungkan oleh panah [8]. Contohnya, tahapan awal adalah dengan memasok daya sebesar 5 V pada ESP32 *Cam* untuk mengaktifkan perangkat. Selanjutnya hubungkan ESP32 *Cam* ke jaringan *WiFi*, jika berhasil terkoneksi dengan jaringan *WiFi* maka proses dapat dilanjutkan

pada tahap berikutnya. Tahap berikutnya adalah melakukan inisialisasi koneksi *WiFi* pada Arduino IDE, sehingga ESP32 *Cam* dapat berkomunikasi dengan jaringan *WiFi* yang telah terhubung. Setelahnya, perlu dilakukan inisialisasi pada ESP32 *Cam* melalui Arduino IDE, jika inisialisasi ini berhasil, maka perangkat siap untuk melanjutkan ke langkah berikutnya, namun jika inisialisasi gagal proses tidak dapat dilanjutkan.

Setelah semua berhasil, langkah selanjutnya adalah mendapatkan alamat IP yang diberikan kepada ESP32 *Cam* melalui Arduino IDE. Alamat ini merupakan tanda pengenal unik yang digunakan untuk menghubungkan perangkat dengan aplikasi kodular. Setelah memperoleh alamat IP, langkah selanjutnya adalah memasukan alamat IP tersebut kedalam aplikasi yang telah dibuat menggunakan kodular. Ini memungkinkan aplikasi untuk mengetahui dimana ESP32 *Cam* berada dalam jaringan. Setelah semua langkah diatas dilakukan, ESP32 *Cam* siap untuk memulai proses *streaming*. Dengan demikian, pengguna dapat mengakses *live streaming* video dari ESP32 *Cam* melalui aplikasi yang telah dibuat dengan kodular.

Dalam konteks penelitian ini, *flowchart* digunakan untuk menggambarkan langkah-langkah yang dilakukan oleh sistem atau perangkat untuk mencapai tujuan tertentu, seperti menginisialisasi ESP32 *Cam*, menghubungkan ke *WiFi*, dan menampilkan *live streaming* video melalui aplikasi Android.

3.2. Skenario Pengujian

3.2.1. Skenario Pengujian Kamera

Skenario pengujian modul kamera dilakukan untuk mengevaluasi kualitas dan fungsi kamera pada perangkat ESP32 *Cam*. Pengujian ini melibatkan beberapa skenario:

1. Kondisi Terkoneksi WiFi

Pada kondisi ini ESP32 *Cam* diharapkan dapat terhubung ke jaringan *WiFi*. Pengujian ini mengevaluasi apakah modul kamera dapat merekam video secara normal saat terkoneksi ke *WiFi*. Hasil yang diharapkan modul kamera merekam video dengan baik dan dapat mengirimkan data melalui jaringan *WiFi*.

2. Pengujian Respon Terhadap Gerakan

Modul kamera diuji untuk melihat bagaimana ia merespon gerakan objek di sekitarnya. Pengujian ini mengukur kemampuan modul kamera untuk mendeteksi dan merekam gerakan. Hasil yang diharapkan modul kamera merespon gerakan dengan baik dan merekam video sesuai dengan gerakan yang terjadi.

3.2.2. Skenario Pengujian Arduino IDE

Skenario pengujian ini dilakukan pada perangkat ESP32 *Cam* untuk memastikan bahwa program yang ditulis untuk mengontrol dan mengirim data melalui

jaringan berjalan dengan baik. Pengujian ini melibatkan beberapa skenario:

1. Inisialisasi ESP32 Cam

ESP32 Cam dihubungkan ke sumber daya dan diinisialisasi. Pengujian ini mengevaluasi apakah ESP32 Cam dapat menginisialisasi IP dengan benar. Hasil yang diharapkan ESP32 Cam berhasil diinisialisasi dengan alamat IP yang sesuai.

2. Unggah Program

Program yang telah ditulis untuk ESP32 Cam diunggah melalui Arduino IDE. Pengujian ini memverifikasi apakah program berhasil diunggah tanpa masalah dan dapat berjalan di perangkat. Hasil yang diharapkan program berhasil diunggah tanpa kesalahan.

3. Mendapatkan Alamat IP Addresss

Pengguna memasukkan alamat IP ESP32 Cam yang telah diinisialisasi ke dalam aplikasi Android versi 11. Pengujian ini mengecek apakah android dapat berkomunikasi dengan ESP32 Cam melalui alamat IP. Hasil yang diharapkan aplikasi android versi 11 dapat terhubung ke ESP32 Cam melalui alamat IP dan menampilkan kontrol *live streaming*.

4. Streaming Video

Pengguna memilih opsi “start stream” pada aplikasi android versi 11. Pengujian ini mengevaluasi apakah perangkat ESP32 Cam dapat mengirim video *live streaming* dengan sukses. Hasil yang diharapkan ESP32 Cam berhasil mengirim video *live streaming* yang dapat diakses melalui aplikasi android versi 11

3.3. Kodular

Kodular Kodular adalah sebuah platform daring yang menyajikan alat untuk menciptakan aplikasi Android menggunakan prinsip *block programming* dengan metode *drag and drop*. Fitur utama Kodular adalah *block programming*, yang memungkinkan kita untuk menghindari keharusan menulis kode program secara manual dalam pembuatan aplikasi Android [7].



Gambar 4. Tampilan aplikasi



Gambar 5. Kode blok pada kodular

Pada gambar 5. Kode blok pada kodular terlihat beberapa kode blok pada *platform* kodular yang digunakan untuk mengatur dan mengontrol menu *streaming* dalam aplikasi Android versi 11 untuk alat *monitoring smart garden*.

Penjelasan mengenai penggunaan kode blok diatas adalah sebagai aturan yang dibuat oleh peneliti untuk menghasilkan *output* pada aplikasi kodular, seperti terdapat tampilan utama yang memiliki menu berupa gambar atau logo kamera yang bertuliskan “streaming”. Jika pengguna melakukan klik pada menu tersebut, maka aplikasi akan mengarahkan pengguna ke tampilan khusus yang disebut sebagai pengaturan kamera ESP32 Cam. Di tampilan pengaturan ini, pengguna memiliki akses penuh untuk mengkonfigurasi berbagai aspek kamera sesuai dengan preferensi mereka. Fitur-fitur yang dapat diatur mencakup resolusi kamera, kualitas gambar atau video, efek visual kamera, serta kemampuan untuk memutar gambar secara *horizontal* dengan fitur “mirror”. Pengguna juga memiliki kontrol atas proses *streaming*, dengan opsi untuk memulai dan menghentikannya. Perlu diingat bahwa penggunaan kamera.

4. HASIL DAN PEMBAHASAN

Penelitian ini menerapkan konsep *smart garden* dengan menggabungkan perangkat ESP32 Cam dan platform Kodular berbasis Android. Tujuan utama penelitian adalah mengembangkan alat *monitoring* yang memungkinkan pengguna untuk memantau pertumbuhan tanaman secara *real-time*. Implementasi sistem dimulai dengan merancang blok diagram yang menggambarkan interaksi antara ESP32 Cam, WiFi, dan aplikasi Android. ESP32 Cam, sebagai mikrokontroler, mengambil peran dalam merekam video dari lingkungan taman secara *real time* melalui modul kamera, dan kemudian mengirimkan data video tersebut melalui jaringan WiFi. Platform Kodular digunakan untuk menciptakan aplikasi Android yang memberikan akses *live streaming* video dari ESP3 Cam kepada pengguna.

Pengujian dilakukan untuk memastikan kinerja dan fungsionalitas sistem. Modul kamera pada ESP32 Cam diuji dalam berbagai skenario, termasuk kondisi terkoneksi dan tidak terkoneksi dengan WiFi, serta respon terhadap gerakan. Hasil pengujian memverifikasi bahwa modul kamera berfungsi dengan

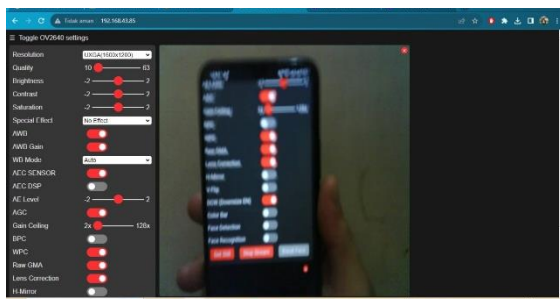
baik dan mampu merekam video serta mengirimkannya melalui jaringan WiFi.

Selanjutnya, pengujian pada Arduino IDE dilakukan untuk memverifikasi bahwa program yang ditulis untuk ESP32 Cam berjalan dengan baik. Pengujian melibatkan langkah-langkah seperti menghubungkan perangkat, mengunggah program, mengakses alamat IP yang diinisialisasi oleh ESP32 Cam, dan memantau *live streaming* video. Hasil pengujian menunjukkan bahwa program berjalan sesuai yang diharapkan, memungkinkan pengguna untuk mengakses *live streaming* video melalui aplikasi Android yang dibuat dengan Kodular.

Dalam kesimpulannya, penelitian ini berhasil menciptakan alat *monitoring* smart garden yang menggabungkan ESP32 Cam dan Kodular. Alat ini memberikan solusi efektif bagi para pembudidaya untuk memonitor pertumbuhan tanaman secara real-time. Implementasi sistem ini memiliki potensi untuk meningkatkan efisiensi dalam bercocok tanam dan memfasilitasi tindakan cepat dalam perawatan tanaman. Meskipun ada beberapa keterbatasan seperti jarak jaringan WiFi dan kualitas koneksi internet, penelitian ini berhasil membuka peluang untuk pengembangan lebih lanjut dalam mengoptimalkan sistem *monitoring smart garden*.

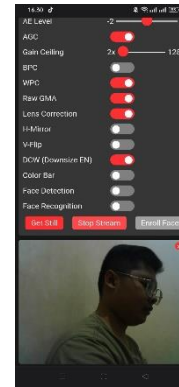
4.1. Pengujian Modul Kamera

Pengujian modul kamera adalah proses evaluasi dan pengujian kualitas kamera pada suatu perangkat, seperti *smartphone*. Modul kamera yang terintegrasi dalam ESP32 ini memungkinkan perangkat untuk merekam video secara *real time*.



Gambar 6. Setting pengujian kamera

Pada gambar 6 diatas terdapat *setting* pengujian kamera pada Web ESP32 Cam dengan menerapkan beberapa fitur seperti Resolusi sebesar UXGA (1600x1200), *Quailty* 10/63, *Brightness* -2/2, *Contrast* -2/2, *Saturation* -2/2, *Spesial Effect* (No effect), AWB diaktifkan, AWB Gain diaktifkan, WB Mode (Auto), AEC Sensor diaktifkan, AEC DSP tidak diaktifkan, AE Level -2/2, AGC diaktifkan, Gain Ceiling 2x/128x, BPC tidak diaktifkan, WPC diaktifkan, Raw GMA diaktifkan, *Lens Correction* diaktifkan, dan H-Mirror tidak diaktifkan. Dimana fitur-fitur tersebut dapat disesuaikan dengan kebutuhan pengguna.



Gambar 7. Hasil Pengujian tampilan video menggunakan android versi 11 pada aplikasi

Pada gambar 7. Hasil pengujian tampilan video menggunakan android versi 11 pada aplikasi, terdapat beberapa fitur didalamnya dimana pengguna memiliki akses penuh untuk mengkonfigurasi berbagai aspek kamera sesuai dengan preferensi mereka. Fitur-fitur yang dapat diatur mencakup resolusi kamera, kualitas gambar atau video, efek visual kamera, serta kemampuan untuk memutar gambar secara *horizontal* dengan fitur "*mirror*". Pengguna juga memiliki kontrol atas proses streaming, dengan opsi untuk memulai dan menghentikannya. Perlu diingat bahwa penggunaan kamera.

Tujuan dari pengujian ini adalah untuk memastikan bahwa kamera bekerja sesuai harapan, menghasilkan video berkualitas tinggi, serta berfungsi dengan baik dalam berbagai kondisi penggunaan. Modul kamera ini menghubungkan langsung ke tempat ESP32 Cam melalui WiFi perangkat.

Tabel 1. Pengujian modul kamera ESP32 Cam

Wifi	Gerakan	Kamera ESP32 Cam
Tidak terdeteksi	Tidak terdeteksi	Tidak menampilkan video
Terkoneksi	Terdeteksi	Menampilkan video
Terkoneksi	Terdeteksi	Menampilkan video

Pada tabel 1. Pengujian modul kamera ESP32 Cam dilakukan berbagai skenario pengujian. Hasil diatas menunjukan bagaimana modul kamera berperilaku dalam situasi yang berbeda, terkait dengan koneksi WiFi dan respon terhadap gerakan. Berikut adalah penjelasan tentang skenario pengujian dan bagaimana hasil tersebut diperoleh:

1. Pengujian Pertama

Dalam pengujian pertama ini ESP32 Cam tidak dapat terhubung ke jaringan WiFi karena kemungkinan terdapat masalah pada koneksi atau konfigurasi. Sehingga gerakan tidak terdeteksi karena kamera tidak berfungsi. Karena kamera tidak berfungsi dan tidak terkoneksi dengan jaringan, tidak ada video yang dihasilkan atau ditampilkan.

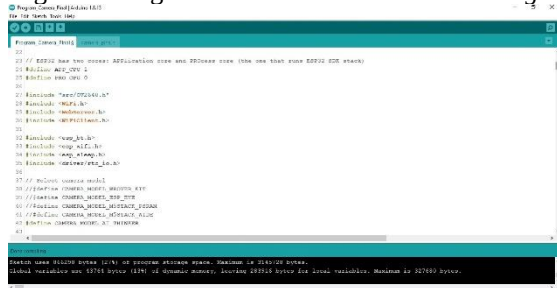
2. Pengujian Kedua dan Ketiga

Dalam pengujian kedua ini ESP32 Cam dapat terhubung ke WiFi ini memungkinkan akses ke kamera melalui jaringan WiFi. Dimana kamera mendeteksi gerakan, sehingga menunjukkan bahwa modul kamera berfungsi dengan baik dan dapat mendeteksi gerakan. Dengan modul kamera yang berfungsi dan terhubung ke jaringan, video dari lingkungan sekitar direkam dan ditampilkan dengan sukses.

Pada pengujian kedua dan ketiga ini menunjukkan bahwa ketika ESP32 Cam berhasil terkoneksi ke jaringan WiFi, modul kamera berfungsi dengan baik dalam mendeteksi gerakan dan merekam video. Hasil ini memberikan indikasi bahwa sistem berhasil berjalan dengan baik dalam skenario di mana semua komponen berinteraksi dengan benar dan modul kamera dapat menghasilkan data video yang diharapkan.

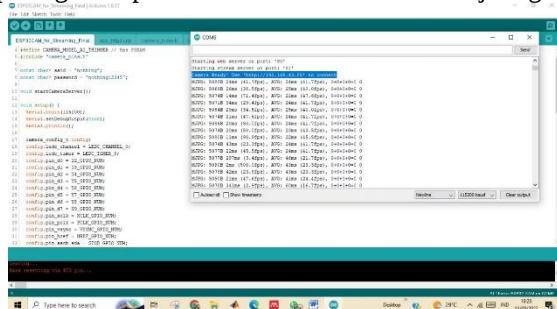
4.2. Pengujian Pada Arduino Ide

Pengujian pada Arduino IDE adalah proses memastikan bahwa program yang telah ditulis untuk papan Arduino berfungsi dengan baik dan sesuai dengan rancangan untuk melakukan *live streaming*.



Gambar 8. Arduino IDE

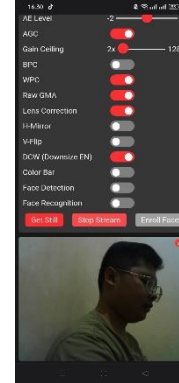
Tahapan awal dalam pengujian alat ESP32 Cam yaitu dengan menghubungkan adaptor atau USB kabel ke sumber listrik, seperti colokan listrik atau port USB komputer. Ini memberikan daya yang diperlukan untuk menjalankan ESP32 Cam. Setelah ESP32 Cam diberi daya, perangkat akan menginisialisasi alamat IP-nya. Ini berarti ESP32 Cam akan mendapatkan alamat IP yang unik di jaringan tempatnya terhubung. Inisialisasi ini memungkinkan perangkat dapat dikenali dan diakses melalui jaringan.



Gambar 9. Tampilan inisialisasi alamat IP

Program untuk ESP32 Cam dibuat menggunakan Arduino IDE dan diunggah (*upload*) ke perangkat melalui koneksi serial seperti kabel USB.

Program ini mungkin mencakup kode untuk mengakses *fitur* pada kamera. Setelah program berhasil diunggah, pengguna perlu mengetahui alamat IP ESP32 Cam yang telah diinisialisasi. Kemudian, alamat IP tersebut dimasukkan ke dalam aplikasi menggunakan android versi 11 yang digunakan untuk mengakses *fitur* pada kamera perangkat.



Gambar 10. Hasil tampilan pada aplikasi menggunakan android versi 11

Setelah berhasil masuk ke dalam aplikasi dengan memasukkan alamat IP perangkat, pengguna akan memiliki akses ke *fitur* “*streaming Video*.” Fitur ini memungkinkan perangkat ESP32 Cam untuk mengirimkan aliran video langsung melalui jaringan, yang dapat ditampilkan dan diakses oleh pengguna melalui aplikasi. Dalam aplikasi, pengguna dapat memilih menu “*Start Stream*” untuk mengaktifkan video *live* dari ESP32 Cam. Setelah diaktifkan, perangkat akan mulai mengirim video *live* melalui jaringan. Untuk menghentikan streaming video *live*, pengguna dapat memilih menu “*stop stream*” pada aplikasi.

5. KESIMPULAN

Berdasarkan pengujian pada android versi 11, dapat disimpulkan bahwa pengujian ESP32 Cam untuk alat *monitoring smart garden* sudah berjalan dengan baik dan menghasilkan hasil sesuai dengan apa yang diharapkan. Hal tersebut menunjukkan bahwa ESP32 Cam telah berhasil menghadirkan kemampuan *live streaming* yang dapat diakses melalui aplikasi dengan memasukkan alamat IP perangkat. Hal ini memungkinkan pembudidaya untuk secara *real time* memonitor kondisi tanaman dengan mudah dan praktis

DAFTAR PUSTAKA

- [1] Ghito, R.K. and Nurdiana, N., 2018, October. Rancang Bangun *Smart Garden System* Menggunakan Sensor Soil Moisture Dan Arduino Berbasis Android (Studi Kasus: Di Gerai Bibit Narnea Cikijing). In *Prosiding Industrial Research Workshop and National Seminar* (Vol. 9, pp. 166-170).
- [2] Sari, D.M. and Hasanuddi, Z.B., 2018. Sistem Kontrol Dan *Monitoring* Pertumbuhan Tanaman Hortikultura Pada *Smart Garden*. *JURNAL IT: Media Informasi STMIK Handayani Makassar*, 8(1).

- [3] Ardiansyah, M., Febryan, A., Adriani, A. and Rahmania, R., 2023. Rancang Bangun Sistem Keamanan Rumah Berbasis Telegram Menggunakan ESP 32 CAM. *Vertex Elektro*, 15(1), pp.64-71.
- [4] Kumala, A. and Winardi, S., 2020. Aplikasi Pencatatan Perbaikan Kendaraan Bermotor Berbasis Android. *Jurnal Intra Tech*, 4(2), pp.112-120.
- [5] M. Saiqul Umam, S. Adi Wibowo, and Y. Agus Pranoto, "IMPLEMENTASI PROTOKOL MQTT PADA APLIKASI SMART GARDEN BERBASIS IOT (INTERNET OF THINGS)," 2023.
- [6] Darmawan, I., Kumara, I. and Khrisne, D.C., 2021. Smart Garden Sebagai Implementasi Sistem Kontrol dan Monitoring Tanaman Berbasis Teknologi Cerdas. *Jurnal SPEKTRUM* Vol, 8(4), pp.161-70.
- [7] Purnamasari, A.I. and Setiawan, A., 2019. Pengembangan *Passive Infrared Sensor* (PIR) HC-SR501 dengan Microcontrollers ESP32-CAM Berbasiskan *Internet of Things* (IoT) dan *Smart Home* sebagai Deteksi Gerak untuk Keamanan Perumahan. *Prosiding SISFOTEK*, 3(1), pp.148-154.
- [8] Hutabarat, L. and Susanti, E., 2020. Perancangan Sistem *Monitoring* Rumah Dengan Sensor *Passive Infra Red* (Pir) Menggunakan Nodemcu Berbasis *Internet of Things* (Iot). *SIGMA TEKNIKA*, 3(2), pp.139-147.