



Smart System Identifikasi Objek Menggunakan Kamera Esp32 dan OpenCV

Atika Dessy Rahmadani¹, Febrian Wahyu Christanto^{*2}, Aldian Umbu Tamu Ama³,

Eryan Ahmad Firdaus⁴

^{1,2}Universitas Semarang

³Universitas Pignatelli Triputra

⁴Universitas Pertahanan

E-mail: ¹atikaredha@gmail.com, ^{*2}febrrian.wahyu.christanto@usm.ac.id, ³umbu@upitra.ac.id,

⁴eryan.firdaus@idu.ac.id

Abstract

Technology used to help complete work is a must. In Buluroto Village, Blora there are 5 hamlets with 6357 residents. There are still many places with poor lighting, so theft still occurs frequently. Ordinary CCTV has difficulty distinguishing objects because of poor lighting, so a system is needed to identify objects with poor lighting. This system was created using Prototype development method with Esp32 Camera, Micro-USB Cable, Esp32-MB, and OpenCV. It was concluded that the smart object identification system was able to detect 20 objects in one frame and store 80 objects in the library. This system can work optimally with an accuracy level reaching 50% - 98%. The delay test results showed average time was 2.18 seconds and around 87.17% of random respondents agreed with this system because it can identify objects like humans, animals, moving and immovable objects well so it is hoped can increase the level of environmental safety.

Keywords : Esp32, Object Identification, OpenCV.

Abstrak

Penggunaan teknologi dalam membantu menyelesaikan pekerjaan merupakan hal yang menjadi keharusan. Di Desa Buluroto, Kabupaten Blora terdapat 5 dukuhan dengan 6357 warga. Keadaan desa masih terdapat banyak tempat dengan pencahayaan kurang sehingga tingkat pencurian masih sering terjadi. CCTV biasa kesulitan untuk membedakan objek karena pencahayaan kurang sehingga dibutuhkan sistem yang mampu mengidentifikasi objek dengan pencahayaan yang kurang. Sistem ini dibuat dengan metode pengembangan Prototype dengan implementasi menggunakan Kamera Esp32, Micro-USB Cable, Esp32-MB, dan OpenCV. Diperoleh kesimpulan bahwa smart system identifikasi objek mampu mendeteksi 20 objek di dalam satu frame dan menyimpan 80 objek di library. Sistem ini dapat bekerja secara optimal dengan tingkat keakuratan identifikasi objek mencapai 50% - 98%. Hasil pengujian delay didapatkan waktu rata-rata selama 2,18 detik serta sekitar 87,17% responden acak setuju terhadap implementasi sistem ini karena dapat mengidentifikasi objek berupa manusia, hewan, benda bergerak, dan tidak bergerak dengan baik sehingga diharapkan dapat menambah tingkat keamanan lingkungan.

Kata Kunci : Esp32, Identifikasi Objek, OpenCV.

I. PENDAHULUAN

Perkembangan teknologi yang sangat pesat di era globalisasi saat ini telah memberikan banyak manfaat dalam kemajuan diberbagai aspek sosial. Perkembangan teknologi ini juga harus diikuti dengan perkembangan pada Sumber Daya Manusia (SDM) [1]. Seperti sistem identifikasi objek yang nantinya

diharapkan bisa diterapkan di berbagai bidang. Dengan memanfaatkan teknologi tersebut, maka diharapkan mampu mempermudah pekerjaan manusia seperti memantau keadaan jalan raya, penyortiran barang, dan sebagainya. Adanya sebuah sistem yang dapat menangkap suatu objek yang ada di depan kamera bisa mengidentifikasi



jenis objek serta melakukan *tracking* objek tentu sangat dibutuhkan dalam mempermudah pekerjaan manusia serta sebagai pemanfaatan teknologi khususnya dibidang identifikasi objek [2]. Sistem identifikasi nantinya bukan hanya dapat mempermudah pekerjaan manusia diberbagai bidang namun diharapkan sebagai cara untuk memperkenalkan teknologi masa kini kepada masyarakat yang kiranya belum terlalu mengenal teknologi sistem identifikasi objek.

Identifikasi objek merupakan suatu bidang keilmuan dari komputer vision yang menggambarkan suatu objek yang didasarkan pada sifat utama dari objek tersebut. Identifikasi objek pada citra digital membutuhkan teknik dan metode yang mampu untuk mengekstraksi dan mengidentifikasi fitur-fitur yang terdapat pada citra digital, dimana komponen utamanya adalah warna sebagai dasar dari representasi objek pada citra digital. Salah satu metode yang mampu menerapkan pengelompokan warna – warna objek pada citra digital sehingga dapat menjadi fitur utama dari objek [3].

Terdapat 5 dukuh dan dihuni sebanyak 6357 warga di Desa Buluroto, Kabupaten Blora, Jawa tengah. Disana masih banyak tempat dengan pencahayaan yang kurang sehingga tingkat pencurian masih sering terjadi.

CCTV biasa kesulitan untuk membedakan objek dengan pencahayaan kurang [4]. Maka dibutuhkan sebuah sistem yang mampu mengidentifikasi objek dengan pencahayaan yang kurang. Identifikasi objek adalah langkah awal untuk melakukan identifikasi yang bertujuan untuk memperoleh akurasi yang baik sebab deteksi objek menyediakan batas lokasi dan skala dari setiap objek yang dapat terdeteksi [5]. Pemrosesan gambar memiliki banyak tujuan dalam pengolahan *image* ini ada banyak diantaranya untuk memperbaiki kualitas gambar atau untuk identifikasi gambar [6]. Metode pengenalan objek adalah salah satu metode untuk sistem keamanan [7]. Terdapat banyak alat otomatis yang dibuat untuk mendukung keamanan rumah sebagai alat *monitoring* dan pencegah tindak pencurian. Beberapa contohnya yaitu dengan adanya CCTV dan *face detection* atau teknologi pengenalan wajah memanfaatkan *library* yang ada pada OpenCV [8] dan memanfaatkan bahasa pemrograman Python sebagai instrumen sistem [9]. Python juga didukung oleh komunitas yang besar [10].

II. METODE PENELITIAN

2.1. Tinjauan Pustaka

Untuk mendukung penelitian ini, maka penelitian ini perlu mengacu pada



penelitian terdahulu agar terjadi sinergi dan pengembangan penelitian ke arah yang lebih baik. Penelitian-penelitian

terdahulu terdapat dalam Tabel 1 dibawah ini.

Tabel 1. Penelitian Terdahulu

No.	Nama Penelitian	Objek	Tools	Hasil penelitian
1.	Pengolahan Citra Digital menggunakan <i>Hierarchical Agglomerative Clustering</i> untuk Identifikasi Objek [11]	Objek Benda	Identifikasi Objek, algoritma <i>Hierarchical Agglomerative Clustering</i> , Citra Digital	Sebuah alat untuk mengidentifikasi objek menggunakan Citra Digital dengan algoritma <i>Hierarchical Agglomerative Clustering</i>
2.	Rancang Bangun Kit Pembelajaran Mikrokontroler Arduino untuk Menunjang Pelajaran Jarak Jauh Menggunakan Raspberry Pi [12]	Objek Manusia	Ciri warna, <i>image clustering</i> , metode <i>K-Means</i> , pengenalan buah	Sebuah aplikasi untuk mengidentifikasi objek manusia berdasarkan ciri warna (<i>Red</i> , <i>Green</i> , dan <i>Blue</i>) untuk mendapatkan nilai <i>centroid</i> data yang akan disimpan ke dalam basis data dengan label nama tertentu.
3.	Alat Pemantau Keamanan Rumah Berbasis Esp32-cam [13]	Objek Benda	ESP32-CAM, <i>Internet of Things</i> , <i>Web Browser</i>	Implementasi ESP32-CAM melalui program Arduino IDE dan <i>web browser</i> yang kemudian menampilkan hasil pada <i>web browser</i> pada <i>smartphone</i> sehingga pemilik rumah dapat memantau dan menggunakan kamera dari jarak jauh melalui <i>remote</i> .
4.	Sistem Pendekripsi Wajah Dengan Metode <i>Viola Jones</i> Menggunakan Esp32-Cam [14]	Objek Benda	ESP32, Modul CAM	Sebuah alat deteksi objek dengan menggunakan ESP32 CAM yang mendekripsi objek-objek disekitar lokasi penerapan alat yang dapat mendekripsi wajah.
5.	Perancangan Alat Penangkap Gambar Pelaku Kejahatan Berbasis Node MCU ESP32 CAM [15]	Objek Wajah	ESP32 CAM; PIR Motion; Penangkap Gambar	Sebuah alat penangkap gambar berbasis ESP 32-CAM yang hasil penelitian berupa rancangan rangkaian alat penangkap gambar yang bekerja pada ruang gelap dan memiliki sumber tegangan dari baterai yang dapat diisi ulang.

Penelitian terdahulu diatas mendekripsi objek berupa benda, buah, warna, dan wajah namun belum dengan keseluruhan objek sehingga terkadang terjadi kendala dalam implementasi di berbagai bidang. Maka dari itu pada dalam penelitian ini akan

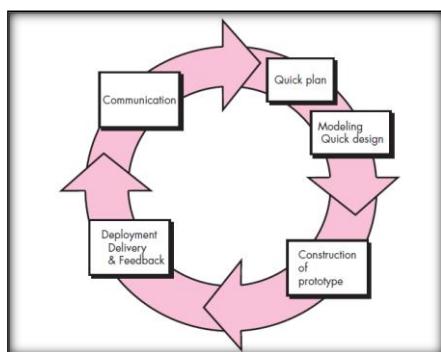
diimplementasikan sebuah sistem cerdas identifikasi objek yang dapat mendekripsi objek bergerak maupun tidak bergerak sehingga diharapkan dapat membantu pekerjaan manusia. Sistem otomatis ini dapat mendekripsi objek apa saja yang masuk dalam *frame* kamera



sehingga dapat memantau keadaan sekaligus memberikan fitur keamanan tambahan di dalam rumah. Perangkat yang dibutuhkan dalam metode indentifikasi objek adalah kamera Esp32 sebagai perangkat pengenalan wajah dan untuk mendeteksi objek.

2.2. Metode Pengembangan Sistem

Penelitian ini menggunakan metode pengembangan *Prototype* dalam perancangan sistem yang akan nantinya akan dikembangkan. Metode ini sangat cocok untuk mengembangkan sebuah perangkat yang akan dikembangkan kembali [16]. Dalam *Prototype* terdapat tahapan *Communication*, *Quick Plan*, *Modelling Quick Design*, *Construction of Prototype*, serta *Development Delivery and Feedback* [17]. Metode ini dimulai dengan pengumpulan pengguna kemudian membuat sebuah rancangan yang selanjutnya akan dievaluasi kembali. Metode pengembangan *Prototype* yang digunakan seperti ditunjukkan pada Gambar 1 dibawah.



Gambar 1. Metode Pengembangan *Prototype*

Adapun penjelasan mengenai metode pengembangan *Prototype* dijelaskan pada Gambar 1 adalah sebagai berikut. *Communication* (Komunikasi) merupakan tahap awal pada model pengembangan *Prototype*. Pada tahap ini *developer* bertemu dengan *user* untuk menentukan tujuan umum, kebutuhan yang diinginkan, dan gambaran bagian-bagian awal yang akan dibutuhkan. Pada tahap ini dilakukan *requirement gathering* atau pengumpulan kebutuhan. Definisi sasaran keseluruhan untuk diterapkan pada smart system yang nantinya akan dikembangkan. Proses selanjutnya adalah proses pengembangan sistem untuk mendeskripsikan fitur - fitur yang terdapat pada kamera Esp32 serta fungsi dari identifikasi objek pada kelas utama sesuai yang diinginkan.

Tahapan *Quick Plan* merupakan tahapan kedua dari pembuatan desain sederhana sebagai langkah dasar memberikan gambaran singkat terkait sistem yang akan dibuat. Pada tahap ini merupakan proses perancangan cepat dan berfokus pada representasi dari aspek-aspek perangkat lunak yang akan terlihat oleh *user* dan tahap ini menjadi dasar pembuatan *Prototype*. Tahapan *Modelling Quick Design* merupakan desain cepat yang mengarah pada pembangunan *Prototype*. Pada tahap ini dirancang aspek yang terlihat oleh

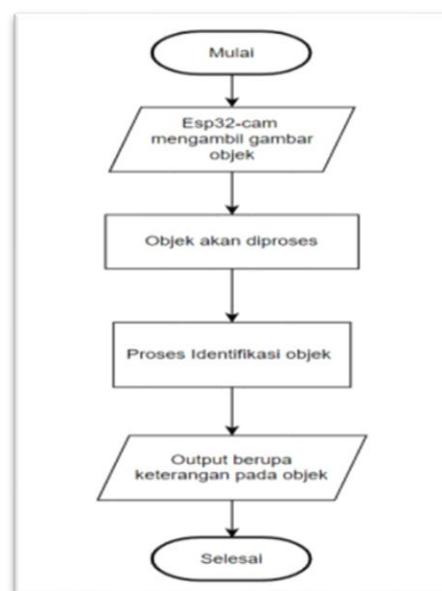


pengguna akhir. Model konstruksi aplikasi digambarkan dalam bentuk rancangan *flowchart* dan rancangan diagram blok nantinya digunakan untuk memulai konstruksi pembuatan *Prototype* yang akan dikembangkan.

Construction of Prototype merupakan pembangunan kerangka atau rancangan *Prototype* dari sistem yang akan dibangun. Dalam langkah ini terdapat juga kontruksi perangkat lunak yang menjelaskan tentang implementasi dari sistem yang akan dibuat atau dikembangkan dimana dalam langkah *Construction of Prototype* ini dilakukan setelah menemukan ide untuk pembuatan *smart system* identifikasi objek. Pengembangan ide yang telah didapat kemudian akan diimplementasikan kedalam sistem. Tahapan terakhir dari metode ini adalah *Deployment Delivery and Feedback* merupakan konstruksi perangkat lunak yang menjelaskan tentang implementasi dari sistem dalam penerapan, pengiriman, dan umpan balik terdapat pengujian perangkat lunak yang menjelaskan evaluasi dari sistem. Pada tahap ini merupakan tahap akhir dalam metode *Prototype*. Tahap ini merupakan tahap pengujian sistem yang didalamnya terdapat penjelasan evaluasi dari sistem yang dibuat.

2.3. Flowchart Desain Sistem

Flowchart juga berguna sebagai fasilitas untuk berkomunikasi antara pemrogram dalam tim suatu proyek [18]. Flowchart desain sistem dimulai dengan kamera Esp32 yang mulai aktif dalam mengambil gambar objek baik diam ataupun bergerak kemudian gambar objek tersebut akan diproses dan diidentifikasi sesuai dengan *data training* dan *data testing* yang sudah disimpan pada *library* sistem. Proses berikutnya adalah proses final dalam sistem yaitu penentuan identifikasi objek dengan menggunakan OpenCV untuk kemudian menghasilkan *output* sistem berupa keterangan pada objek yang telah teridentifikasi dalam bentuk tulisan dan kemudian proses dalam *flowchart* selesai. Gambaran desain *flowchart* dalam penelitian ini terdapat dalam Gambar 2 dibawah ini.

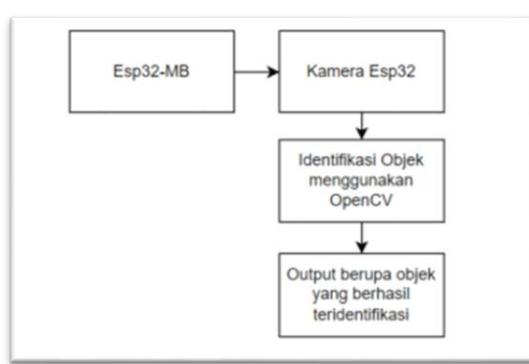


Gambar 2. *Flowchart* Desain Sistem



2.4. Perancangan Diagram Blok

Diagram blok adalah suatu penyajian bergambar dari fungsi yang dilakukan oleh tiap komponen dan aliran sinyalnya. Dengan adanya diagram blok maka semuanya akan terlihat jelas seperti komponen yang digunakan seperti *input* dan *output* sistem. Gambar 3 dibawah merupakan diagram blok dalam penelitian ini.



Gambar 3. Diagram Blok Model Perancangan

Gambar 3 diatas adalah skema desain diagram blok. Esp32-MB digunakan sebagai *port micro USB* kamera Esp32 ke komputer yang dihubungkan dengan *micro USB cable* [19]. Kamera Esp32 digunakan sebagai alat mengambil gambar objek yang nantinya diproses ke OpenCV. Identifikasi OpenCV gambar objek yang diambil dengan kamera Esp32 lalu akan diproses [20]. *Output* setelah proses identifikasi berhasil akan ada tampilan berupa gambar objek beserta keterangan objek tersebut serta akurasi presentasi dari objek.

2.5. Skema Keseluruhan Rangkaian

Skema ini digunakan untuk mengetahui dan menerangkan keseluruhan model dari *smart system* identifikasi objek menggunakan kamera Esp32 dan OpenCV. Pada Gambar 4 terdapat rangkaian kamera Esp32 dan Esp32-MB yang disatukan kemudian *micro USB cable* sebagai penghubung ke *laptop*. Berikut dalam Gambar 4 adalah skema rangkaian yang dirancang.



Gambar 4. Skema Keseluruhan Rangkaian

Skema keseluruhan rangkaian pada Gambar 4 diatas diawali dengan Esp32-MB yang digunakan sebagai *port micro USB* kamera Esp32 yang dihubungkan ke *laptop*. Kamera Esp32 digunakan sebagai kamera pengambil gambar objek yang nantinya akan diidentifikasi. *micro USB cable* digunakan sebagai penghubung Esp32-MB yang disatukan dengan kamera Esp32 kemudian dihubungkan menggunakan *micro USB cable* ke *laptop*. *Laptop* digunakan sebagai *library* utama dan menampilkan gambar objek yang teridentifikasi.



Hardware atau perangkat keras berfungsi untuk memasukan data ke processor atau untuk menyimpan dan menghasilkan data. Pada implementasi perangkat keras ini terdiri dari kamera Esp32 sebagai komponen utama dalam sistem identifikasi, Esp32-MB sebagai *port* USB untuk kamera Esp32, dan *micro USB cable*. Gambar 5 dibawah merupakan gambar kombinasi perangkat keras pendukung kamera dalam penelitian ini. Kombinasi perangkat pada gambar tersebut antara lain adalah Esp32-MB yang berfungsi untuk menghubungkan kamera Esp32 atau biasa disebut sebagai *port micro USB*. Dikarenakan kamera Esp32 tidak memiliki *port micro USB*, maka dibutuhkan Esp32-MB sebagai penghubung dan nantinya akan disatukan dengan kamera Esp32.



Gambar 5. Kamera Esp32

Kamera Esp32 merupakan komponen utama dalam sistem identifikasi yang digunakan sebagai alat penangkap atau pengambil gambar objek. Gambar objek yang diambil kamera Esp32 nantinya akan diolah oleh Python, Arduino IDE, dan OpenCV untuk

kemudian akan diidentifikasi oleh sistem. Program yang ditulis dengan Arduino IDE disebut *sketch* digunakan untuk *compile* ke *logic-board* Arduino ataupun mikrokontroler lainnya [21]. *Micro USB cable* digunakan sebagai penghubung komponen kamera Esp32 ke *laptop*. Kabel ini akan menghubungkan dua komponen tersebut sehingga proses koding dapat berjalan dengan baik..

2.6. Pengujian dan Evaluasi

Pada tahap ini dilakukan pengujian sistem menggunakan pengujian *Beta*. Pengujian *Beta* dilakukan dengan pembagian kuisioner di lapangan dihitung menggunakan metode skala *Likert* untuk mengetahui tingkat kepuasan pengguna terhadap sistem identifikasi yang telah dibuat. Dalam pengujian *Beta* didapat 23 data respon. Kemudian untuk mengetahui *delay* pada sistem dilakukan pengujian *delay* menggunakan *stopwatch*. Hasil penelitian dan pengujian yang telah didapatkan nantinya akan dievaluasi hasilnya. Perbaikan akan dilakukan apabila terdapat kekurangan dari hasil penelitian yang telah dicapai dan tentunya akan menjadi saran penelitian berikutnya yang dapat dijalankan dan dikembangkan.

III. HASIL DAN PEMBAHASAN

Implementasi sistem dalam penelitian ini dimulai dengan integrasi

antara *library* OpenCV, Arduino IDE, dan perangkat keras kamera Esp32. OpenCV (*Open Source Computer Vision Library*) sendiri adalah sebuah perangkat lunak yang ditujukan untuk pengolahan citra digital secara *real-time* yang dibuat oleh Intel menggunakan bahasa C++ [6]. Kamera Esp32 ini nantinya dimasukkan pula kode program yang nantinya dapat mengidentifikasi objek-objek yang tersimpan dalam basisdata berupa *data training* dan *data testing*. Kode program yang akan diterapkan kedalam sistem menggunakan bahasa pemrograman Python.

Proses konfigurasi IP sistem identifikasi objek nantinya diproses yang mana program akan meminta data yang diperlukan misalnya dalam proses identifikasi objek tertentu diperlukan data berupa titik koordinat objek, konfigurasi IP, pengambilan gambar objek, dan variabel-variabel lainnya dari objek yang terdeteksi. Konfigurasi IP diperlukan untuk menghubungkan kamera Esp32 dalam mengambil gambar objek yang akan diolah dan dicari koordinat target kemudian ditentukan nama objek yang terdeteksi.

Gambar 6 dibawah ini merupakan proses konfigurasi IP pada kamera Esp32. konfigurasi IP kamera Esp32 yang dihubungkan ke sistem identifikasi objek kemudian akan muncul keterangan “CAMERA OK” yang berarti

kamera Esp32 telah terintegrasi dan setelah sistem terhubung dengan koneksi *internet* maka akan muncul alamat IP seperti pada Gambar 7.

```
75 }
76 serial.print("http://");
77 serial.println(WiFi.localIP());
```

Output Serial Monitor x

Message (Enter to send message to ESP32 Wever Module on 'COM3')

camera: 4, 255, 255, 255
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
memSize:1000, clock_div1:
load:0x0fff20030, len:13444
load:0x007f0000, len:13964
load:0x40000400, len:9400
entry:0x00000050

Gambar 6. Proses Konfigurasi pada Kamera Esp32

```
entry 0x400805f0

CAMERA OK
http://192.168.43.191
/cam-lo.jpg
/cam-hi.jpg
/cam-mid.jpg
```

Gambar 7. Alamat IP Kamera Esp32

Pada Gambar 7 telah muncul alamat IP http://192.168.43.191 sebagai contoh yang berarti kamera Esp32 sudah berhasil terhubung ke sistem. Namun sebelum proses identifikasi berjalan, maka harus ditambahkan alamat IP pula kedalam pemrograman Python pada Arduino IDE seperti pada Gambar 8 dibawah ini.



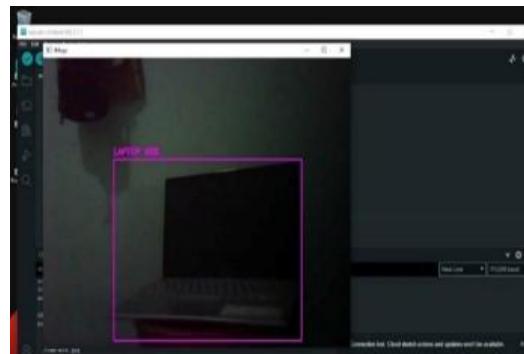
```
camera.py - C:\Users\ES11\Documents\espdetek\camera.py (3.10.11)
File Edit Format Run Options Window Help
import cv2
import numpy as np
import urllib.request

url = 'http://192.168.43.191/cam-hi.jpg'

cap = cv2.VideoCapture(url)
whT=320
confThreshold = 0.5
```

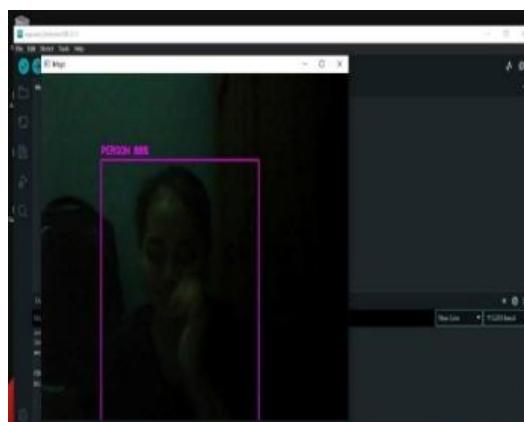
Gambar 8. Proses Penambahan IP pada Arduino IDE

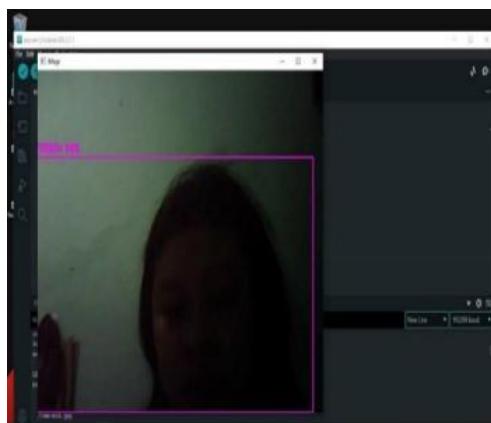
Pengujian sistem dilakukan dengan tujuan untuk mengetahui sistem berkerja dengan baik sesuai yang diharapkan atau tidak dan dapat berjalan sesuai dengan fungsinya serta bebas dari kesalahan maupun *error*. Oleh karena itu diperlukan sebuah pengujian untuk menghindari sistem bermasalah pada saat digunakan serta nantinya sistem identifikasi objek bisa terealisasikan dengan baik. Dalam proses pengujian identifikasi objek akan diuji tentang seberapa baik sistem dapat mengidentifikasi objek, berapa banyak gambar yang dapat teridentifikasi dalam satu *frame tracking* dari pengujian, dan seberapa besar persentase kebenaran pada objek ini. Gambar 9 dan Gambar 10 dibawah ini adalah hasil identifikasi pada objek.



Gambar 9. Identifikasi Objek Diam

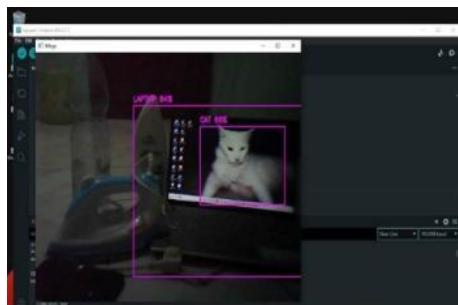
Pengujian identifikasi objek pada Gambar 9 dan 10 adalah identifikasi terhadap objek diam dan objek bergerak. Pada objek diam teridentifikasi bahwa objek berupa *laptop* dan akurasi objek yang teridentifikasi sebesar 98%. Sedangkan pada identifikasi objek bergerak pencahayaan objek sangat berpengaruh dalam proses sistem mengidentifikasi objek. Dengan pencahayaan yang kurang pada objek bergerak, keakuratan objek yang teridentifikasi hanya sebesar 88%, sedangkan dengan pencahayaan objek yang cukup maka keakuratan objek yang teridentifikasi meningkat menjadi 94%.





Gambar 10. Identifikasi Objek Bergerak

Pengujian sistem dilakukan pula untuk multi objek yang terdapat di dalam satu frame yaitu objek yang akan diidentifikasi melebihi dari satu objek. Hasil pengujian multi objek terdapat di dalam Gambar 11 dan Gambar 12 di bawah ini.



Gambar 11. Identifikasi Multi Objek Laptop dan Kucing



Gambar 12. Identifikasi Multi Objek Botol dan Gelas

Pengujian yang dilakukan terhadap multi objek seperti pada Gambar 11 dan Gambar 12 berjalan dengan baik. Dalam hal ini sistem dapat melakukan identifikasi yang akurat terhadap multi objek dengan tingkat keakuratan mencapai lebih dari 50%.

Pada penelitian ini dilakukan pula pengujian terhadap sistem identifikasi objek menggunakan perangkat dengan spesifikasi CPU yang sama tetapi dengan spesifikasi memori yang berbeda-beda sehingga ditemukan proses *delay* yang berbeda-beda pada setiap perangkatnya. Hasil pengujian ditunjukkan pada Tabel 2 dibawah ini. Pengujian *delay* menggunakan beberapa laptop dengan spesifikasi yang berbeda-beda. Pengujian dilakukan pada sistem identifikasi objek agar mengetahui *delay* pada saat sistem dijalankan. Dari pengujian ini didapatkan rata-rata delay adalah 2,18 detik dan semakin besar spesifikasi memori pada sebuah perangkat akan mempengaruhi proses terhadap sistem identifikasi objek.

Tabel 2. Pengujian *Delay Smart System* Identifikasi Objek

Spesifikasi CPU	Spesifikasi Laptop	Pengujian	Identifikasi Objek	Delay Identifikasi Objek (detik)
Core I3	RAM 2GB	Pengujian 1	Berhasil teridentifikasi	3.54
Core I3	RAM 4GB	Pengujian 2	Berhasil teridentifikasi	2.02
Core I3	RAM 6GB	Pengujian 3	Berhasil teridentifikasi	0.98
Rata – Rata				2,18

Dilakukan pula pengujian *Beta* dalam penelitian ini. Pengujian *Beta* dilakukan dengan pembagian kuesioner sebanyak 23 responden secara acak di Desa Buluroto, Kabupaten Blora dan dihitung menggunakan metode skala Likert untuk mengetahui tingkat kepuasan pengguna terhadap implementasi sistem identifikasi objek [22]. Berikut Tabel 3 adalah tabel skor pernyataan.

Tabel 3. Skor Kuesioner

No	Pernyataan	Skor untuk
		Pernyataan
1	Sangat setuju	5
2	Setuju	4
3	Kurang Setuju	3
4	Tidak Setuju	2

Pada Tabel 3 diatas setelah dilakukan pengujian dengan skala *Likert* terhadap tanggapan responden, maka hasil tabulasi data tersebut dilakukan perhitungan skor sebagai berikut :

- a. Jawaban Sangat Setuju (SS) = 50 responden x 5 = 250

- b. Jawaban Setuju (S) = 31 responden x 4 = 124
- c. Jawaban Kurang Setuju (KS) = 7 responden x 3 = 21
- d. Jawaban Tidak Setuju (TS) = 3 responden x 2 = 6
- e. Total hasil skor = 401
- f. Skor Maksimum = $23 \times 4 \times 5 = 460$ (jumlah responden x jumlah pertanyaan x skor tertinggi likert)
- g. Skor Minimum = $23 \times 4 \times 2 = 184$ (jumlah responden x jumlah pertanyaan x skor terendah likert)
- h. Indeks (%) = (Total Skor / Skor Maksimum) x 100
- i. Indeks (%) = $(401 / 460) \times 100$
- j. Indeks (%) = 87,17 %.

Dari hasil perhitungan menggunakan skala *Likert* diperoleh tingkat kepuasan responden terhadap implementasi smart system identifikasi objek menggunakan kamera Esp32 dan OpenCV adalah mencapai 87,17 %.



IV. KESIMPULAN

Berdasarkan pada hasil penelitian dan pengujian *smart system* identifikasi objek diperoleh beberapa kesimpulan bahwa sistem ini dapat mengidentifikasi objek diam maupun objek bergerak dengan baik. Tingkat keakurasiannya antara 50% - 98% tergantung pada baik dan buruknya pencahayaan terhadap objek. Dari pengujian yang telah dilakukan didapat bahwa sistem mampu mendeteksi hingga 20 objek sekaligus dalam satu *frame* dan sistem ini mampu menyimpan sebanyak kurang lebih 80 objek pada library OpenCV. Pengujian *delay* yang dilakukan dengan spesifikasi CPU yang sama tetapi dengan spesifikasi memori yang bervariasi didapatkan hasil waktu *delay* rata-rata sebesar 2,18 detik dalam proses identifikasi objek dan semakin besar spesifikasi memori pada sebuah perangkat akan mempengaruhi proses terhadap sistem identifikasi objek. Berdasarkan hasil pengujian Beta didapatkan tingkat kepuasan 23 responden acak terhadap terhadap *smart system* identifikasi objek menggunakan OpenCV dan kamera Esp32 ini mencapai 87,17 % yang berarti sistem ini mendapatkan respon positif dari pengguna.

V. SARAN

Smart system identifikasi objek ini dapat dikembangkan lebih lanjut. Untuk pengembangan perangkat keras, saran yang diberikan antara lain bahwa sistem ini dapat dikembangkan dengan menambahkan sistem pengenalan citra wajah atau *face-id*. Dari kamera penangkap objek dapat menggunakan spesifikasi kamera dengan resolusi yang lebih baik agar hasil yang didapatkan lebih baik dan jelas. Sistem ini dapat diintegrasikan pula dengan Bot Telegram sehingga gambar yang ditangkap dan teridentifikasi dapat dikirim melalui *smartphone* pengguna. Diharapkan dari penelitian ini dapat pula menambahkan sistem keamanan rumah dan lingkungan khususnya pada lingkungan Desa Buluroto, Kabupaten Blora.

DAFTAR PUSTAKA

- [1] F. W. Christanto, S. Susanto, and B. A. Pramono, "NodeMCU dan Kontrol Pengukuran Ph Air Berbasis Android untuk Menentukan Tingkat Kejernihan Pada Air Tawar," *J. Pengemb. Rekayasa dan Teknol.*, vol. 16, no. 1, p. 1, 2020, doi: 10.26623/jprt.v16i1.1895.
- [2] E. Didik, "Mengidentifikasi Object," *Univ. Bina Nusant.*, pp. 7–10, 2017.
- [3] A. Willis and K. Hasan, "Simple Computer Vision Algorithm Production using OpenCV for



- 'Virtual Ecosystem' Project," *IJNMT (International J. New Media Technol.)*, vol. 7, no. 1, pp. 23–27, 2020, doi: 10.31937/ijnmt.v7i1.1660.
- [4] K. Burugu, V. Doddha, V. Rajmoor, S. C. Yerra, and S. Bhutada, "Security Surveillance Monitoring System Using Open CV," *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, vol. 8, no. 3, pp. 402–407, 2022, doi: https://doi.org/10.32628/IJSRCS_EIT.
- [5] M. F. Wicaksono and M. D. Rahmatya, "Implementasi Arduino dan ESP32 CAM untuk Smart Home," *J. Teknol. dan Inf.*, vol. 10, no. 1, pp. 40–51, 2020, doi: 10.34010/jati.v10i1.2836.
- [6] S. Mulia, "Mengenal OpenCV Dalam Python: Pengertian, Sejarah, Dukungan pada OS, Fitur-fitur," <https://idmetafora.com/news/read/1177/Mengenal-OpenCV-Dalam-Python-Pengertian-Sejarah-Dukungan-pada-OS-Fitur-fitur.html>. Accessed: Feb. 25, 2024. [Online]. Available: <https://idmetafora.com/news/read/1177/Mengenal-OpenCV-Dalam-Python-Pengertian-Sejarah-Dukungan-pada-OS-Fitur-fitur.html>
- [7] R. Monitawati, F. T. Elektro, and U. T. Bandung, "Tracking Seragam Militer Berbasis Image Processing Secara Real Time," *e-Proceeding Eng.*, vol. 6, no. 1, pp. 1364–1369, 2019.
- [8] P. G. Krihsna, Y. Subbarayudu, S. S. S. Kumar, D. Naveen, A. Srivastava, and K. Thangamani, "IoT Sensor-based Sustainable Smart Home Management for Human Needs through Micro Controller," in *E3S Web of Conferences*, S. S. Kumar, Ed., Newcastle, England: EDP Sciences, 2023, pp. 1–9. doi: 10.1051/e3sconf/202343001079.
- [9] T. C. A.-S. Zulkhaidi, E. Maria, and Y. Yulianto, "Pengenalan Pola Bentuk Wajah dengan OpenCV," *JURTI (Jurnal Rekayasa Teknol. Informasi)*, vol. 3, no. 2, pp. 181–186, 2020, doi: 10.30872/jurti.v3i2.4033.
- [10] A. N. Syahrudin and T. Kurniawan, "Input dan Output pada Bahasa Pemrograman Python," *J. Dasar Pemrograman Python STMIK*, no. June 2018, pp. 1–7, 2018.
- [11] J. Jumadi, Y. Yupianti, and D. Sartika, "Pengolahan Citra Digital Untuk Identifikasi Objek Menggunakan Metode Hierarchical Agglomerative Clustering," *JST (Jurnal Sains dan Teknol.)*, vol. 10, no. 2, pp. 148–156, 2021, doi: 10.23887/jstundiksha.v10i2.3363 6.
- [12] J. Utama, T. Rahajoeningroem, and Y. Firmansyah, "Rancang Bangun Kit Pembelajaran Mikrokontroler Arduino untuk Menunjang Pelajaran Jarak Jauh Menggunakan Raspberry Pi," *J. Teknol. dan Inf.*, vol. 12, no. 2, pp. 131–148, 2022, doi: 10.34010/jati.v12i2.6963.
- [13] W. Yulita and A. Afriansyah, "Alat Pemantau Keamanan Rumah Berbasis Esp32-Cam," *JTST (Jurnal Teknol. dan Sist. Tertanam)*, vol. 3, no. 2, pp. 23–31, 2022, doi: 10.33365/jtst.v3i2.2197.
- [14] I. M. Y. C. Putra, I. M. O. Widhyantara, and I. G. A. K. D. Djuni H, "Sistem Pendekripsi Wajah Dengan Metode Viola Jones Menggunakan Esp32-Cam," *J. SPEKTRUM*, vol. 9, no. 1, p. 94, 2022, doi: 10.24843/spektrum.2022.v09.i01. p11.



- [15] I. Purwata, M. Ashari, W. Bagye, and Saikin, "Perancangan Alat Penangkap Gambar Pelaku Kejahatan Berbasis Node MCU ESP32 CAM," *Jambura J. Electr. Electron. Eng.*, vol. 5, no. 1, pp. 36–40, 2023, doi: 10.37905/jjeee.v5i1.16871.
- [16] S. Andrianita, "Pengembangan dan Analisis Kualitas Aplikasi Panduan Shalat Jenazah pada Handphone Berbasis Android," Universitas Negeri Yogyakarta, 2015.
- [17] R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner's Approach*, 9th ed. New York: McGraw Hill, 2020.
- [18] Syamsiah, "Perancangan Flowchart dan Pseudocode Pembelajaran Mengenal Angka dengan Animasi untuk Anak PAUD Rambutan," *STRING (Satuan Tulisan Ris. dan Inov. Teknol.)*, vol. 4, no. 1, pp. 86–93, 2019, doi: 10.30998/string.v4i1.3623.
- [19] Y. B. Pratama and N. P. Dalimunthe, "Implementasi Teknik Computer Vision Untuk Deteksi Viridiplantae Pada Lahan Pasca Tambang," *Bull. Comput. Sci. Res.*, vol. 3, no. 1, pp. 64–72, 2022, doi: 10.47065/bulletincsr.v3i1.193.
- [20] N. Mehendale, "Object Detection using ESP 32 CAM," *SSRN Electron. J.*, pp. 1–9, 2022, doi: 10.2139/ssrn.4152378.
- [21] R. H. P. Sejati and R. Mardhiyyah, "Deteksi Wajah Berbasis Facial Landmark Menggunakan OpenCV dan DLib," *JurTI (Jurnal Teknol. Informasi)*, vol. 5, no. 2, pp. 144–148, 2021, doi: <https://doi.org/10.36294/jurti.v5i2.2220>.
- [22] W. I. Rahayu and M. R. Shafina, "Aplikasi Analisis Kelayakan Sistem Untuk Pengukuran Usability Dengan Menerapkan Metode Use Questionnaire," *J. Tek. Inform.*, vol. 14, no. 3, pp. 152–160, 2022.
- [23] Firdaus, E. A., Maulani, S. (2023). Perencanaan Kerangka Kerja Menggunakan The Open Group Architecture Framework-Architecture Development Method (TOGAF-ADM) pada Puskesmas Sukatani. *Jurnal Sistem Informasi Galuh*, 32-37.