

IMPLEMENTASI ALGORITMA HSV UNTUK AKSEBILITAS PENDERITA BUTA WARNA PARSIAL

Muhamad Naufal dan Muhammad Subali

Universitas Cendekia Abditama
Jl Islamic Raya, Kelapa. Dua Tangerang, Kab. Tangerang, Banten 15811
naufal@cendekia.ac.id, subali@uca.ac.id

ABSTRAK

Tujuan aplikasi ini adalah untuk membuat alat bantu untuk orang dengan buta warna parsial yang menggunakan teknologi pengolahan gambar berbasis desktop. Algoritma HSV (Hue, Saturation, Value) digunakan untuk mengidentifikasi kasi warna pada gambar yang diambil secara real-time oleh kamera. Dengan algoritma ini, pengguna dapat mengidentifikasi kasi warna utama seperti merah, hijau, biru, kuning, dan ungu, yang sering sulit dilihat oleh orang yang buta warna. Untuk meningkatkan aksesibilitas informasi warna, aplikasi juga memiliki tur pemuatan suara berdasarkan warna yang terdeteksi. Evaluasi aplikasi menunjukkan respons yang cepat dan akurat dalam deteksi warna, yang memenuhi kebutuhan sehari-hari orang yang buta warna parsial.

Kata Kunci: Aksesibilitas, HSV, Buta warna parsial, Deteksi warna.

PENDAHULUAN

Di era digital saat ini, teknologi semakin berkembang, terutama dalam bidang kecerdasan buatan (AI). Artificial Intelligence (AI), atau dalam bahasa Indonesia dikenal sebagai Kecerdasan Buatan, adalah bidang ilmu komputer yang berfokus pada pengembangan sistem dan mesin yang memiliki kemampuan untuk melakukan tugas yang biasanya memerlukan kecerdasan manusia untuk melakukannya [1]. AI telah menjadi bagian penting dari kehidupan sehari-hari, membantu manusia dengan banyak hal seperti pengenalan wajah, kendaraan otonom, asisten virtual, dan rekomendasi konten. Banyak aspek kehidupan menjadi lebih otomatis dan efisien berkat kemajuan AI yang cepat. Perkembangan ini tidak dapat dihindari, dan manusia harus beradaptasi untuk memanfaatkan potensi AI untuk kemajuan bersama.

Machine learning adalah komponen kecerdasan buatan yang banyak digunakan untuk memecahkan berbagai masalah [2]. Ini terutama digunakan untuk membantu orang buta warna parsial dalam mendeteksi warna. Ketidakmampuan untuk membedakan warna tertentu disebut buta warna [3, 4]. Solusi lain bagi orang buta warna parsial adalah terapi penggunaan kacamata atau lensa. Model warna HSV

(Hue, Saturation, Value) adalah salah satu model warna alternatif selain model warna RGB. AI dapat mengidentifikasi kasi dan membedakan warna dengan lebih akurat dengan menggunakan algoritma ini. Hue menunjukkan warna asli, seperti merah, ungu, dan kuning. Ini juga digunakan untuk membedakan antara warna dan menentukan apakah cahaya kemerahan, kehijauan, atau warna lainnya [5]. Algoritma HSV dapat memproses warna dengan cara yang lebih sesuai dengan persepsi manusia, sehingga dapat memberikan informasi warna yang lebih jelas kepada pasien yang memiliki kesulitan membedakan warna. Teknologi ini tidak hanya membantu dalam kehidupan sehari-hari, tetapi juga meningkatkan kualitas hidup bagi orang dengan buta warna parsial, membantu mereka memahami dan menikmati dunia visual dengan lebih baik. Teknologi ini juga menawarkan solusi bagi mereka yang belum menggunakan lensa atau kacamata, yang mahal dan tidak dapat diakses oleh semua orang.

Oleh karena itu, aplikasi pendeteksi warna berbasis desktop yang memanfaatkan teknologi kamera secara real-time menjadi solusi praktis untuk pengolahan gambar digital. Aplikasi desktop adalah aplikasi yang dapat berjalan secara independen atau secara bersamaan dalam sistem desktop atau laptop dan memiliki kemampuan untuk

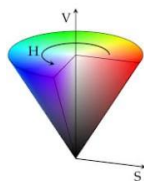
melakukan berbagai tugas yang telah diatur oleh pengguna[6]. Untuk melakukan penangkapan gambar secara langsung, kamera adalah bagian yang paling penting[7]. Aplikasi ini dapat menangkap dan menganalisis warna dari objek secara langsung, memberikan hasil deteksi yang cepat dan akurat. Pengguna hanya perlu mengarahkan kamera ke objek atau area yang ingin diperiksa, dan informasi tentang warna akan ditampilkan segera setelah itu. Teknik ini sangat bermanfaat bagi desainer, seniman, dan orang yang mengalami gangguan penglihatan warna karena membantu mereka lebih mudah memahami dan bekerja dengan warna. Selain itu, aplikasi ini mencakup berbagai domain, seperti pendidikan, bisnis, dan penelitian, di mana analisis warna real-time dapat meningkatkan produktivitas dan akurasi serta memenuhi kebutuhan khusus dari setiap domain.

Aksesibilitas

Aplikasi pendeteksi warna ini dikembangkan dengan tujuan utama untuk membuatnya lebih mudah digunakan bagi orang dengan kebutuhan khusus, seperti orang yang buta warna. Aplikasi ini memiliki antarmuka yang sederhana dan mudah digunakan yang memungkinkan pengguna dengan cepat mendeteksi dan mengidentifikasi warna objek. Selain itu, aplikasi ini memiliki tur output suara yang dirancang khusus untuk mempermudah akses.

HSV

Model Hue Saturation Value (HSV) berasal dari model warna RGB, dan HSV harus terlebih dahulu mengubah warna RGB menjadi HSV. Model HSV juga lebih umum daripada pemodelan RGB.



Gambar 1: Ilustrasi Ruang Warna HSV

Nilai pada Hue mencapai hingga 360 pada ruang warna HSV, sedangkan Saturation diwakili oleh garis lurus yang berpusat di tengah lingkaran dengan warna putih, yang memiliki nilai dari 0 hingga 100%, dengan nilai yang lebih tinggi semakin jelas warna yang terlihat, dan sebaliknya nilai yang lebih rendah semakin memutih warnanya. Untuk mengatur warna kegelapan, nilai pada Value diwakili dengan garis tegak lurus yang berkisar dari 0 hingga 100%. Nilai yang lebih tinggi menunjukkan warna yang lebih sempurna, dan nilai yang lebih rendah menunjukkan warna yang lebih gelap[8]. Aplikasi yang ingin dibuat untuk menerapkan HSV ke Program membutuhkan bantuan teknologi untuk memastikan bahwa aplikasi berjalan dengan baik dan lancar. Selain itu, untuk mendapatkan nilai hue yang diinginkan untuk warna yang diinginkan, algoritma HSV harus dirumuskan Sebagai berikut :

Standarisasi nilai RGB

Pertama, nilai RGB harus dinormalisasi ke dalam rentang 0 hingga 1. Misalnya, nilai awal RGB berada dalam rentang 0 hingga 255, maka masing-masing nilai dibagi dengan 255 untuk normalisasi :

$$\text{Red} = 255R, \text{Green} = 255G, \text{Blue} = 255B \quad (1)$$

Cari Nilai Minimum dan Maksimum.

Selanjutnya, nilai minimum dan maksimum dari komponen warna yang telah dinormalisasi dihitung:

$$\begin{aligned} C_{max} &= \max(R, G, B), \\ C_{min} &= \min(R, G, B) \end{aligned} \quad (2)$$

Nilai C_{max} di antara merah, hijau, dan biru adalah nilai tertinggi, sedangkan C_{min} adalah nilai terkecil.

Hitung Mencari Delta

Selisih Nilai maksimum dan minimum dapat dihitung dengan cara berikut :

$$\Delta = C_{max} - C_{min} \quad (3)$$

Menghitung Nilai Hue

Jika $\Delta = 0$ ini berarti tidak ada perbedaan antara nilai maksimum dan minimum, sehingga warna adalah warna netral (seperti hitam, putih, atau abu-abu). dan Hue didefinisikan sebagai 0. Jika $C_{max} = R$ (Jika komponen warna Merah adalah yang paling dominan), Maka Hue Di Hitung Sebagai berikut :

$$H = 60 \times (\Delta G - B) \quad (4)$$

Jika $C_{max} = G$ (Jika komponen warna Hijau adalah yang paling dominan), Maka Hue Di Hitung Sebagai berikut :

$$H = 60 \times (\Delta B - R + 2) \quad (5)$$

Jika $C_{max} = B$ (Jika komponen warna Biru adalah yang paling dominan), Maka Hue Di Hitung Sebagai berikut :

$$H = 60 \times (\Delta R - G + 4) \quad (6)$$

Hasil Akhirnya Jika nilai H yang dihitung negatif, tambahkan 360 untuk memastikan bahwa Hue berada dalam rentang $[0, 360]$.

Buta Warna

Buta Warna Sendiri adalah ketidakmampuan untuk membedakan warna tertentu [3, 4]. Buta warna sendiri terbagi menjadi dua kategori: buta warna Parsial dan Total [4]. Apa itu buta warna parsial? Dalam penelitian ini, itu adalah jenis buta warna yang paling umum disebabkan oleh kehilangan atau penurunan kemampuan protan (yang mengatur sensitivitas warna merah) atau deutan (yang mengatur sensitivitas warna hijau). Kasus seperti ini disebut sebagai buta warna merahhijau. Buta biru-kuning lebih langka daripada buta merah-hijau. Buta Warna Juga Dapat Di klasi kasikan melalui tingkatan buta warnanya, diklasifikasi menjadi 3 yaitu Anomali Trikomat dan Dikhromat [4].

Deteksi Warna

Studi ini menyelidiki pengembangan aplikasi pendeteksi warna yang menggunakan algoritma HSV. Aplikasi ini dimaksudkan untuk meningkatkan aksesibilitas orang dengan gangguan

penglihatan seperti buta warna parsial. Aplikasi ini mendeteksi warna melalui analisis komponen warna merah, hijau, dan biru (RGB) yang diambil oleh kamera dalam waktu nyata. Proses ini melibatkan normalisasi nilai RGB serta perhitungan nilai Hue (H), Saturation (S), dan Value (V). Nilai-nilai ini kemudian diinterpretasikan untuk mengidentifikasi warna objek secara akurat. Aplikasi ini tidak hanya memungkinkan pengguna melihat warna objek secara visual, tetapi juga mengeluarkan suara yang membantu pengguna dengan keterbatasan visual memahaminya. Ini membuatnya alat bantu yang berguna dan inklusif. Penelitian ini menggunakan teknologi pengolahan citra untuk mendapatkan data yang diperlukan. Termasuk peningkatan gambar, segmentasi, pengenalan pola, dan deteksi tepi. Berbagai industri, seperti medis, keamanan, fotogra, dan pengenalan wajah, menggunakan teknologi pengolahan gambar. Pengolahan gambar biasanya melibatkan manipulasi piksel, menggunakan filter, dan mengubah gambar dari format RGB ke ruang warna lain seperti HSV. Studi ini menggunakan algoritma pengolahan citra untuk mendeteksi warna objek yang diambil oleh kamera secara real-time. Algoritma ini memproses gambar yang diambil oleh kamera untuk mengidentifikasi kasi warna-warna tertentu, seperti merah, hijau, biru, kuning, dan ungu, berdasarkan nilai Hue, Saturation, dan Value HSV. Aplikasi ini dapat mengubah informasi visual menjadi informasi yang dapat dikenali oleh pengguna dengan menggunakan teknik pengolahan citra. Adapun Teknologi pendukung lainnya sebagai berikut:

1. Visual Studio Code

Visual Studio Code adalah editor kode gratis yang membantu Anda memulai membuat kode dengan cepat. Gunakan untuk membuat kode dalam bahasa pemrograman apa pun, tanpa harus beralih ke editor lain. Visual Studio Code mendukung banyak bahasa, termasuk Python, Java, C++, JavaScript, dan banyak lagi [9]. Untuk membuat aplikasi deteksi warna berbasis desktop, Visual Studio Code

(VS Code) adalah lingkungan pengembangan yang sangat penting. Dengan menggunakan VS Code, pengembang dapat dengan mudah menulis, mengedit, dan mengelola kode Python yang kompleks untuk memproses gambar secara real-time menggunakan OpenCV dan mengintegrasikannya dengan antarmuka pengguna yang dikembangkan dengan PyQt5. Integrasi yang kuat dengan Extension Python memungkinkan penggunaan tur seperti IntelliSense untuk pemrograman yang lebih produktif, serta kemampuan debugging yang mendalam untuk mengoptimalkan dan memperbaiki kode. Selain itu, VS Code membantu manajemen proyek, memungkinkan struktur proyek yang baik dan manajemen dependensi yang diperlukan untuk aplikasi kompleks seperti ini. Penggunaan ekstensi dan plugin tambahan juga meningkatkan kemampuan VS Code. Hal ini termasuk penggunaan Git untuk kontrol versi dan kerja kolaboratif, serta alat bantu pengembangan yang dapat disesuaikan untuk memenuhi persyaratan pembuatan aplikasi deteksi warna berbasis desktop. Akibatnya, VS Code memudahkan pengembangan aplikasi dan membantu pengembang membuat solusi yang tepat dan dapat diandalkan untuk pengguna yang mengalami buta warna parsial.

2. Python

Python merupakan salah satu bahasa pemrograman yang banyak digunakan oleh perusahaan besar maupun para Developer untuk mengembangkan berbagai macam aplikasi berbasis Desktop, Web dan Mobile. Selain Perusahaan besar dan Developer, Python juga banyak dipelajari di kalangan mahasiswa dan kampus, terutama kampus berbasis IT. Pembelajaran pemrograman tidak mudah, namun tidak juga sulit, tergantung bagaimana cara belajarnya, dan seberapa cepat dalam mempelajarinya. Dalam penelitian ini, Python memiliki peran yang sangat penting dalam mendukung pengembangan aplikasi pendeteksi warna berbasis Desktop. Pertama, Python digunakan untuk pengolahan citra melalui pustaka OpenCV. OpenCV memungkinkan aplikasi untuk

mengakses dan menangkap Video dari kamera secara real-time, mengkonversi gambar dari format BGR ke HSV untuk mendeteksi warna, serta mengolah frame video untuk mendeteksi warna tertentu pada titik tengah gambar. Selain itu, OpenCV juga berfungsi untuk menggambar elemen visual pada frame, seperti titik tengah, yang membantu dalam proses deteksi warna[10]. Adapun Library Yang Mendukung kinerja kinerja dari Program, Penulis melakukan implementasi penggunaan library python untuk mengolah data yang telah didapatkan[11]. dan untuk aplikasi akseibilitas buta warna, library berikut dapat mendukung proses deteksi warna:

OpenCV Python

Python OpenCV (Open Source Computer Vision Library) adalah sebuah library yang digunakan untuk mengolah gambar dan video sehingga pengguna dapat meng-ekstrak informasi dari sebuah citra. OpenCV pada program ini memainkan peran penting dalam menangkap dan mengolah gambar secara realtime. Pertama, OpenCV digunakan untuk mengakses kamera perangkat dan mengambil frame gambar secara langsung. Gambar yang ditangkap kemudian dikonversi dari format BGR ke format HSV, yang lebih ideal untuk deteksi warna. OpenCV memeriksa piksel pada titik tertentu untuk mendeteksi warna berdasarkan rentang nilai HSV yang telah ditentukan. Selain itu, OpenCV menambahkan elemen visual seperti titik pusat pada gambar untuk membantu proses deteksi warna. Setelah warna terdeteksi, program menggunakan OpenCV untuk mengubah citra tersebut menjadi format yang bisa ditampilkan pada antarmuka pengguna dengan bantuan PyQt5. OpenCV juga memungkinkan deteksi warna secara akurat dan cepat, sehingga sistem dapat memberikan umpan balik suara yang sesuai berdasarkan warna yang terdeteksi[12].

PyQt5

PyQT5 adalah sebuah toolkit widget Graphic User Interface (GUI), yaitu sebagai antarmuka gra k antara bahasa pemrograman Python dan QT Digunakan untuk membuat

antarmuka pengguna (GUI) yang interaktif[13]. dapat membuat tampilan aplikasi yang mudah digunakan dan ramah pengguna, seperti tombol, label, dan jendela utama. Dengan memanfaatkan Rancangan menggunakan toolkit PyQt5 yang menjadi perantara antara bahasa pemrograman python dan Graphic User Interface (GUI) dari QT [13].Maka Aplikasi Akan Memiliki Tampilan yang dapat membantu user memahami penggunaan aplikasi yang baik dan benar.

NumPy

NumPy digunakan untuk mengkonversi nilai warna dari format BGR ke HSV, memungkinkan deteksi warna yang akurat dalam citra. NumPy juga digunakan untuk melakukan operasi matematika dan logika pada array untuk menentukan apakah nilai HSV dari piksel tertentu berada dalam rentang warna yang telah ditentukan. Selain itu, NumPy meningkatkan efisiensi proses dengan memungkinkan operasi matriks dan array yang cepat dan efektif, yang sangat penting untuk aplikasi real-time seperti deteksi warna. dan numpy sendiri adalah library pemrograman array utama untuk bahasa Python[14].

Playsound

Library ini digunakan untuk memutar le suara. Aplikasi menggunakan Playsound untuk memberikan umpan balik audio kepada pengguna berdasarkan warna yang terdeteksi. Modul playsound hanya memiliki satu fungsi yaitu fungsi (yang juga bernama) playsound dan Fungsi ini membutuhkan satu argumen yaitu path ke le suara yang ingin Anda putar. File ini bisa berupa le lokal atau URL[15].

Date Time

Modul DateTime pada aplikasi berfungsi untuk mengelola dan memanipulasi informasi tanggal dan waktu dengan efisien. Ini mencakup kemampuan untuk mendapatkan waktu saat ini, memformat tanggal dan waktu agar mudah dibaca, melakukan perhitungan waktu seperti menghitung selisih antara dua waktu, serta menyimpan data waktu untuk penggunaan di masa mendatang. Dengan

demikian, modul DateTime sangat penting untuk memastikan aplikasi dapat menangani aspek waktu dengan akurat dan efektif[16].

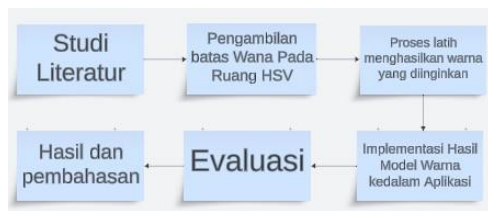
3. PIP

Dalam penelitian ini, PIP digunakan sebagai manajer paket Python yang krusial untuk mengelola instalasi, pembaruan, dan penghapusan modul-modul seperti PyQt5 untuk pengembangan antarmuka pengguna gras (GUI) dan OpenCV untuk pengolahan gambar. Peran PIP sangat penting dalam memastikan integrasi yang lancar dari teknologi-teknologi ini ke dalam aplikasi, sehingga memungkinkan fokus utama penelitian pada pengembangan turtur aplikasi serta eksperimen dalam deteksi dan manipulasi warna secara efektif. Selain itu, PIP juga mendukung manajemen dependensi yang stabil dan efisien, memastikan aplikasi dapat beroperasi dengan konsistensi di berbagai lingkungan pengembangan.

4. QT Designer

Qt Designer adalah antarmuka gras yang diisi dengan widget Qt dan alat lain yang digunakan untuk membangun GUI. Menggunakan drag aplikasi Qt Designer dan drop interface, Anda dapat membuat dan menyesuaikan dialog Anda sendiri, jendela, dan Widget. Dalam Aplikasi, QtDesigner berfungsi sebagai alat pengembangan visual yang memungkinkan desain antarmuka pengguna (GUI) secara gra s untuk aplikasi PyQt5 [17]. QtDesigner memungkinkan Anda untuk merancang dan menyesuaikan tata letak GUI dengan menarik dan menjatuhkan komponen-komponen UI seperti tombol, label, widget input teks, dan lainnya, tanpa harus menulis kode secara manual. Dengan menggunakan QtDesigner, Anda dapat menciptakan antarmuka pengguna yang intuitif dan estetik secara lebih efisien, yang kemudian dapat diintegrasikan langsung ke dalam aplikasi Python Anda menggunakan PyQt5

METODE PENELITIAN



Gambar 2: Alur Penelitian

Alur dari tahapan penelitian ini menggunakan metode Kualitatif, yang bertujuan untuk memahami secara mendalam proses yang saling berhubungan dan mengidentifikasi permasalahan yang sedang terjadi secara langsung. Penelitian Kualitatif menitikberatkan pada analisis yang komprehensif tentang berbagai aspek kejadian, termasuk apa yang terjadi, bagaimana proses tersebut berlangsung, kapan dan di mana peristiwa tersebut terjadi. Metode ini bertujuan untuk mengungkap arti yang lebih dalam, mendefinisikan konsep yang relevan, memahami simbol-simbol yang muncul, dan memberikan deskripsi rinci mengenai fenomena yang diteliti.

Berikut ini adalah penjelasan lebih rinci mengenai gambar Gambar 2, yang akan memperlihatkan tahapan-tahapan penelitian yang telah dilakukan, termasuk langkah-langkah spesifik yang diambil untuk mengumpulkan dan menganalisis Data, serta hasil yang diperoleh dari setiap tahapan tersebut.

Tahap pertama dalam penelitian ini adalah melakukan studi literatur yang menyeluruh dan mendalam, yang melibatkan peninjauan berbagai sumber ilmiah, jurnal, buku, dan artikel yang relevan dengan deteksi warna, algoritma HSV, serta aplikasi aksesibilitas untuk penderita buta warna parsial. Tujuan utama dari tahap ini adalah mengumpulkan informasi dan wawasan yang relevan untuk membangun dasar teoritis yang kuat bagi penelitian ini. Studi literatur dilakukan dengan teliti mengumpulkan data dari berbagai literatur terpercaya, termasuk jurnal ilmiah, tulisan ilmiah, buku akademis, sumber-sumber internet, dan bahan pustaka lainnya, baik daring maupun luring. Dalam proses ini, peneliti fokus pada berbagai aspek yang terkait dengan algoritma HSV, bagaimana

algoritma ini dapat diterapkan dalam aplikasi pendeteksi warna, dan bagaimana aplikasi ini dapat memberikan manfaat bagi penderita buta warna parsial. Pendekatan ini membantu peneliti untuk memahami kerangka teoritis secara komprehensif, mendapatkan referensi yang berhubungan langsung dengan topik penelitian, serta mendukung tujuan utama penelitian. Melalui studi literatur yang mendalam ini, peneliti dapat mengidentifikasi celah-celah dalam penelitian sebelumnya, menemukan metode yang paling efektif, dan membangun fondasi yang kokoh untuk tahap-tahap penelitian selanjutnya.

Langkah berikutnya adalah pengumpulan data yang relevan dengan buta warna parsial. Data ini dikumpulkan berdasarkan penelitian yang telah ada mengenai spektrum warna yang sulit dikenali oleh penderita buta warna parsial. Informasi ini akan digunakan sebagai acuan untuk menentukan ambang batas warna dalam ruang warna HSV. Data dikumpulkan dari berbagai jurnal ilmiah dan artikel yang mengkaji warna-warna yang sulit dikenali oleh penderita buta warna parsial, seperti merah dan hijau. Selain itu, data mengenai spektrum warna ini dianalisis untuk memastikan bahwa ambang batas warna yang diterapkan dalam aplikasi dapat mendeteksi warna dengan akurat dan membantu penderita buta warna parsial dalam membedakan warna yang sulit mereka identifikasi. Setelah data terkumpul, langkah selanjutnya adalah memproses data tersebut menjadi data warna yang akan digunakan sebagai ambang batas warna dalam ruang warna HSV. Proses ini melibatkan pengolahan dan pengorganisasian data agar siap digunakan dalam aplikasi. Penelitian ini menekankan pentingnya data yang akurat dan representatif untuk memastikan bahwa aplikasi yang dikembangkan dapat memberikan manfaat maksimal bagi penderita buta warna parsial.

Pada tahap ketiga ini, data warna yang telah dikumpulkan digunakan untuk melatih model deteksi warna menggunakan algoritma HSV. Proses pelatihan ini mencakup beberapa langkah penting seperti preprocessing data, augmentasi data, dan

pembagian dataset menjadi set pelatihan dan validasi. Preprocessing data melibatkan normalisasi nilai warna untuk memastikan bahwa data konsisten dan siap untuk digunakan dalam pelatihan model. Augmentasi data dilakukan dengan menambahkan variasi pada data warna, seperti perubahan intensitas dan saturasi, untuk meningkatkan kemampuan model dalam mengenali berbagai kondisi warna. Dataset kemudian dibagi menjadi set pelatihan dan validasi. Set pelatihan digunakan untuk melatih model dalam mengenali dan mengklasifikasi warna-warna yang sulit dikenali oleh penderita buta warna parsial. Proses ini juga mencakup pembuatan pemetaan dari label warna ke indeks numerik untuk memudahkan proses klasifikasi data. Tujuan dari tahap ini adalah untuk mengembangkan model yang dapat mendeteksi dan mengklasifikasi warna dengan akurat, membantu penderita buta warna parsial dalam mengenali warna yang sulit mereka bedakan. Proses pelatihan ini merupakan langkah krusial dalam pengembangan aplikasi pendeteksi warna yang efektif dan bermanfaat.

Setelah model deteksi warna menggunakan algoritma HSV dilatih, langkah berikutnya adalah mengimplementasikan model tersebut ke dalam aplikasi pendeteksi warna. Aplikasi ini akan menggunakan model yang telah dilatih untuk mendeteksi dan mengenali warna secara otomatis, membantu penderita buta warna parsial dalam mengenali warna yang sulit mereka bedakan. Setelah model deteksi warna menggunakan algoritma HSV dilatih, langkah berikutnya adalah mengimplementasikan model tersebut ke dalam aplikasi pendeteksi warna. Aplikasi ini akan menggunakan model yang telah dilatih untuk mendeteksi dan mengenali warna secara otomatis, membantu penderita buta warna parsial dalam mengenali warna yang sulit mereka bedakan. Proses implementasi ini melibatkan integrasi model deteksi warna ke dalam antarmuka pengguna yang dikembangkan menggunakan PyQt5. Model deteksi warna akan diintegrasikan dengan kamera atau sumber gambar lainnya untuk mendeteksi warna secara real-time. Aplikasi akan

memberikan umpan balik audio menggunakan Playsound untuk memberi tahu pengguna tentang warna yang terdeteksi, sesuai dengan kebutuhan aksesibilitas Buta warna. Tahap analisis dilakukan terhadap hasil yang diperoleh dari implementasi aplikasi pendeteksi warna. Data yang dikumpulkan selama penggunaan aplikasi dianalisis untuk mengevaluasi akurasi, efisiensi, dan keandalan sistem dalam mendeteksi warna yang sulit dikenali oleh penderita buta warna parsial. Metode analisis yang digunakan mencakup analisis statistik dan evaluasi kualitatif berdasarkan umpan balik pengguna. Tujuan dari tahap ini adalah untuk menilai kinerja aplikasi, mengidentifikasi area yang perlu diperbaiki, serta memastikan aplikasi memberikan manfaat nyata dan efektif dalam membantu penderita buta warna parsial mengenali warna dengan lebih baik.

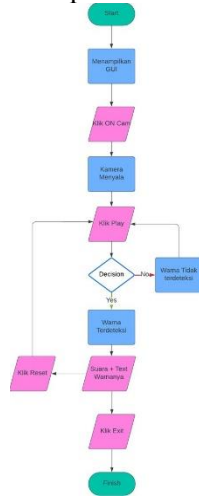
Pada tahap akhir ini, hasil dari analisis disusun dan dibahas secara mendetail. Pembahasan mencakup interpretasi hasil deteksi warna, kesimpulan yang dapat diambil, serta implikasi dari temuan penelitian. Selain itu, batasan penelitian dan rekomendasi untuk penelitian selanjutnya juga dibahas untuk menyajikan hasil penelitian secara komprehensif. Pembahasan ini bertujuan untuk memberikan wawasan mendalam tentang efektivitas aplikasi pendeteksi warna dalam membantu penderita buta warna parsial, serta memberikan rekomendasi untuk pengembangan lebih lanjut agar aplikasi ini dapat lebih bermanfaat di masa depan.

HASIL DAN PEMBAHASAN

Analisis Perancangan Sistem

Sebagai model desain perancangan sistem menggunakan Unified Modeling Language (UML), berikut adalah beberapa diagram UML yang dapat digunakan dalam perancangan program pendeteksi warna untuk membantu individu dengan buta warna parsial. Dalam tahap perancangan sistem, langkah-langkahnya mencakup berbagai diagram dan rancangan yang penting untuk membangun program pendeteksi warna.

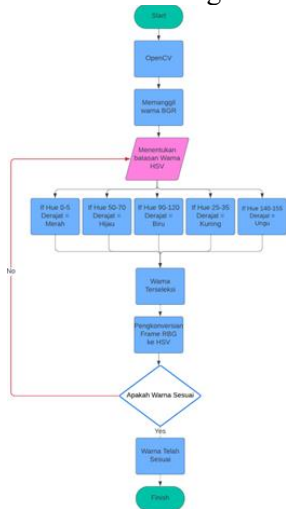
1. Flowchart Aplikasi



Gambar 3: Flowchart Aplikasi

Gambar 3 menjelaskan bahwa pengguna harus mengklik tombol "On Cam" untuk mengaktifkan kamera dan memulai penangkapan gambar secara real-time, menggunakan teknologi OpenCV. Setelah objek diletakkan di depan kamera, pengguna dapat menekan tombol "Start" untuk mendeteksi warna objek tersebut dengan algoritma HSV. Hasil deteksi ditampilkan sebagai teks dan suara, membantu pengguna dengan keterbatasan penglihatan. Tombol "Reset" memungkinkan pengguna untuk mengulangi proses, sementara tombol "Exit" digunakan untuk menutup aplikasi dengan aman, melepaskan semua sumber daya yang digunakan.

2. Flowchart Sistem Algoritma HSV



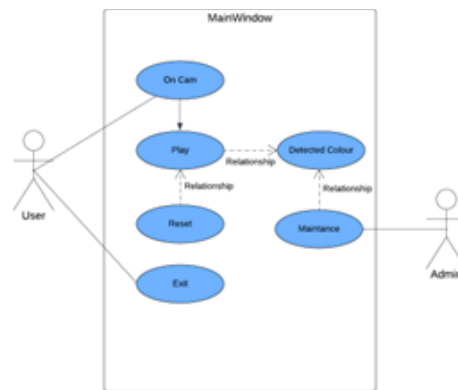
Gambar 4: Flowchart Sistem Algoritma HSV

Pada Gambar 4 dijelaskan proses pembuatan frame dalam aplikasi pendeteksi warna. Pertama, aplikasi memanggil OpenCV untuk mengakses kamera dan menangkap gambar dalam format BGR. Gambar ini kemudian dikonversi ke format HSV untuk memudahkan deteksi warna. Aplikasi menerima batasan warna dalam format HSV dan memprosesnya untuk mengidentifikasi warna sesuai rentang hue yang telah ditentukan:

1. Merah: 0-5 derajat,
2. Hijau: 50-70 derajat,
3. Biru: 90-120 derajat,
4. Kuning: 25-35 derajat,
5. Ungu: 140-155 derajat.

Setelah warna terdeteksi dan sesuai preferensi pengguna, aplikasi memberikan informasi yang akurat mengenai warna objek yang diamati.

3. Use Case Diagram



Gambar 5: Usecase Diagram

Gambar 5 menggambarkan Use Case Diagram yang menunjukkan interaksi pengguna dengan aplikasi pendeteksi warna melalui empat tombol utama: "On Cam" untuk mengaktifkan kamera, "Start" untuk memulai pendeteksian warna menggunakan algoritma HSV, "Reset" untuk mengatur ulang aplikasi, dan "Exit" untuk menutup aplikasi dengan aman. Diagram ini juga mencakup peran admin yang bertanggung jawab melakukan pemeliharaan aplikasi, memastikan aplikasi berfungsi dengan baik untuk pengguna.

4. Wireframe Aplikasi



Gambar 6: Usecase Diagram

Gambar 6 menampilkan wireframe aplikasi yang menggambarkan tampilan utama, termasuk toolbar "Help" dengan opsi "About" yang menampilkan informasi pembuat aplikasi. Tampilan utama memiliki GridView yang menampilkan video real-time dari kamera menggunakan OpenCV untuk mendeteksi warna. GridView memungkinkan pengguna menyesuaikan gambar agar titik hijau pendeteksi terlihat. Label "colour detected" berfungsi sebagai indikator saat warna terdeteksi. Aplikasi juga menampilkan logo UCA sebagai penanda hak cipta dan widget waktu untuk menampilkan jam saat aplikasi digunakan. Urutan penggunaan aplikasi melibatkan tombol On Cam, Start, Reset (opsional), dan Exit (opsional). Tombol Reset dan Exit bersifat opsional, tergantung kebutuhan pengguna.

Implementasi dan Pembahasan

Program deteksi warna ini dirancang untuk membantu individu dengan kesulitan melihat warna, dengan mengenali warna objek dan menerjemahkannya ke dalam teks dan suara. Menggunakan metode pengolahan citra dengan OpenCV dan algoritma HSV, aplikasi mendeteksi warna secara real-time melalui kamera. Warna yang terdeteksi diidentifikasi berdasarkan nilai hue spesifik, dan hasilnya ditampilkan sebagai teks serta suara untuk membantu pengguna memahami warna. Aplikasi ini dilengkapi tombol kontrol intuitif, memungkinkan pengguna mengoperasikan dengan mudah. Evaluasi kinerja memastikan akurasi deteksi dan efektivitas aplikasi dalam memberikan informasi warna yang diperlukan.

1. Analisa Sistem

Sistem deteksi warna ini dibangun dengan Python menggunakan pustaka OpenCV untuk pengolahan gambar, Playsound untuk output suara, dan PyQt5 untuk antarmuka pengguna.

- **Antarmuka Pengguna:** Dirancang dengan PyQt5, antarmuka memudahkan pengguna menyalakan kamera, mendeteksi warna, mereset aplikasi, dan keluar, dengan akses yang efisien dan intuitif.
- **Deteksi Kamera:** Menggunakan cv2.VideoCapture, sistem menghubungkan dan mendeteksi kamera untuk mengambil data warna secara real-time dari objek.
- **Pengumpulan Data:** Data warna dikumpulkan langsung dari kamera dan ditampilkan sebagai teks dan suara tanpa anotasi manual.
- **Deteksi Warna:** Sistem mengonversi warna dari BGR ke HSV dan menganalisis piksel untuk mendeteksi warna dengan akurasi tinggi.

Penerjemahan Warna: Hasil deteksi warna diterjemahkan ke teks dan suara, membantu pengguna, terutama yang memiliki kesulitan melihat warna, memahami warna objek dengan lebih baik.

2. Pengumpulan Data

Dalam penerapan algoritma HSV untuk deteksi warna pada OpenCV, pengumpulan data menjadi langkah penting untuk menetapkan batasan-batasan yang digunakan dalam pengenalan warna oleh kamera. Proses pengumpulan data ini merupakan bagian krusial dari penelitian, karena menentukan keakuratan sistem dalam mendeteksi dan mengenali warna objek. Pengumpulan data tersebut mencakup identifikasi dan analisis warnawarna tertentu yang menjadi fokus utama, terutama mengingat bahwa individu dengan buta warna parsial sering kali mengalami kesulitan dalam membedakan lima warna spesifik. Warna-warna ini memerlukan perhatian khusus dalam penelitian untuk memastikan bahwa sistem dapat memberikan hasil deteksi yang akurat.

dan sesuai dengan kebutuhan pengguna yang mengalami kelemahan penglihatan pada warna-warna sebagai berikut :

Merah



Gambar 7: Merah Dalam Ruang HSV Pada OpenCV

Gambar 7 menunjukkan warna merah dalam ruang HSV dengan koordinat hue antara 0 5 . Rentang 5 ini adalah batas warna merah sebelum tercampur dengan warna lain, menggambarkan bagaimana warna merah tampak pada hue 0 dan 5 di ruang warna HSV dalam OpenCV.

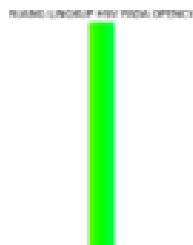
Kuning



Gambar 8: Kuning Dalam Ruang HSV Pada OpenCV

Gambar 8 menampilkan warna kuning dalam ruang HSV dengan hue antara 25 hingga 35 . Rentang ini menetapkan batas atas untuk warna kuning sebelum bercampur dengan warna lain, memastikan representasi yang akurat dalam aplikasi deteksi warna menggunakan OpenCV.

Hijau



Gambar 9: Hijau Dalam Ruang HSV Pada OpenCV

Gambar 9 menunjukkan warna hijau dalam ruang HSV dengan hue antara 50 hingga 70. Rentang ini menentukan batas atas hue untuk warna hijau sebelum bercampur dengan warna lain, penting untuk akurasi deteksi warna dalam aplikasi menggunakan OpenCV.

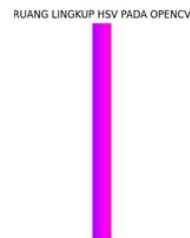
Biru



Gambar 10: Biru Dalam Ruang HSV Pada OpenCV

Gambar 10 menunjukkan warna biru dalam ruang HSV dengan hue antara 90 hingga 120. Rentang ini menetapkan batas atas hue untuk warna biru sebelum bercampur dengan warna lain, yang penting untuk akurasi deteksi warna dalam aplikasi menggunakan OpenCV.

Ungu



Gambar 11: Ungu Dalam Ruang HSV Pada OpenCV

Gambar 11 menunjukkan warna ungu dalam ruang HSV dengan hue antara 140 hingga 155 . Rentang ini menetapkan batas atas hue untuk warna ungu sebelum bercampur dengan warna lain, penting untuk mendefinisikan rentang warna ungu secara akurat dalam aplikasi deteksi warna menggunakan OpenCV.

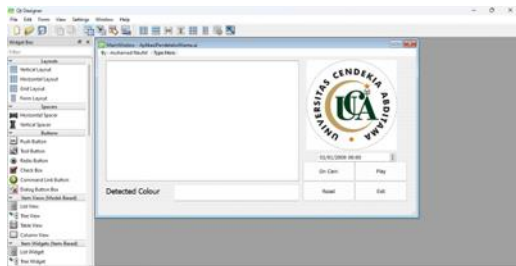
3. Perancangan GUI

Dalam Pembuatan UI aplikasi di perlukan pencocokan tur , dan dalam

penelitian ini ada 11 bagian penting yang menjadi Bagian Dari aplikasi ini , 11 bagian Tersebut Mencakup :

- a. Button On Cam
- b. Button Play
- c. Button Reset
- b. Button Exit
- c. Mainwindow
- d. Toolbar Help
- e. Gridview
- f. Label bertuliskan Detected Colour
- g. Text Edit Untuk menampilkan Output
- h. Widget jam
- i. Logo Instansi

Semua Itu Dibutuhkan dalam aplikasi ini sehingga terciptalah tampilan aplikasi sebagai berikut :



Gambar 12: GUI Aplikasi Pendeteksi Warna

4. Pengujian

Dalam Pengujian Terdapat dua kali pengujian dimana yang pertama adalah pengujian sistem dan yang kedua adalah pengujian aplikasi , Pengujian Blackbox Sistem tersebut di cantumkan sebagai berikut :

Tabel 4.1: Blackbox Sistem

NO	Input	Output	Hasil	
			Suara	Text
1	[Red Square]	[Camera Icon]	✓	✓
2	[Red Square]	[Camera Icon]	✓	✓
3	[Blue Square]	[Camera Icon]	✓	✓
4	[Blue Square]	[Camera Icon]	✓	✓
5	[Green Square]	[Camera Icon]	✓	✓
6	[Green Square]	[Camera Icon]	✓	✓
7	[Yellow Square]	[Camera Icon]	✓	✓
8	[Yellow Square]	[Camera Icon]	✓	✓
9	[Purple Square]	[Camera Icon]	✓	✓
10	[Purple Square]	[Camera Icon]	✓	✓

Dan Pengujian Blackbox Aplikasi Sebagai Berikut :

Tabel 4.2: Blackbox Aplikasi

No	Input	Output	Hasil
1	On Cam	Menyalakan kamera	✓
2	Play	Mulai mendeteksi Warna	✓
3	Reset	Mengulang Kembali proses seperti saat awal menyalakan kamera.	✓
4	Exit	Keluar Dari aplikasi.	✓
5	Text Edit	Mengeluarkan Output Text.	✓
6	18/06/2024 22:05	Memberitahu Waktu yang ada pada Device.	✓
7	Help About	Mengeluarkan msg box tentang pembuat aplikasi.	✓

Setelah itu adapun Mendukung Tujuan Dari penelitian maka di butuhkan sebuah pengujian langsung terhadap pengidap buta warna parcial dengan memiliki 50 Orang maka Peneliti akan menjabarkan hasil pengujian dari 50 orang tersebut sebagai berikut :



Gambar 13: Pengujian Terhadap Pasien Buta Warna Parcial

Pada Gambar 4.26 di atas, dari 50 orang yang merespon, 90% (45 orang) menganggap bahwa aplikasi berjalan dengan baik dalam mendeteksi warna pada objek, sementara 10% (5 orang) menganggap bahwa aplikasi tidak berjalan dengan baik. Hal ini menunjukkan bahwa masih terdapat kesalahan dalam mendeteksi warna pada objek, kemungkinan karena objek yang dideteksi tidak memenuhi syarat dalam ruang lingkup HSV dan batasan warnanya tidak sesuai dengan warna yang dituju.

PENUTUP

Kesimpulan yang dapat diambil adalah bahwa algoritma HSV dapat diimplementasikan ke dalam aplikasi pendeteksi warna pada objek. Namun, algoritma ini memiliki beberapa kekurangan, seperti sensitivitas terhadap cahaya. Jika cahaya yang menyelimuti objek tidak cukup, OpenCV mungkin mendeteksi warna lain, bukan warna yang sebenarnya ingin dideteksi. Di balik kekurangannya, algoritma HSV memiliki kelebihan, yaitu sangat cocok untuk aplikasi yang menggunakan kamera secara real-time, sehingga tidak memerlukan waktu lama dalam mendapatkan output warna dari objek tersebut.

Aplikasi ini juga dapat digunakan dalam kehidupan sehari-hari dengan menggunakan perangkat umum. Aplikasi ini mudah digunakan, sehingga dapat membantu individu dengan buta warna parsial dalam kehidupan sehari-hari, memungkinkan mereka menjalani kehidupan dengan lebih percaya diri dan tidak bingung lagi dengan warna pada objek yang mereka miliki. Berdasarkan hasil survei pada Gambar 13 di atas, 90% dari 50 responden menganggap bahwa aplikasi ini dapat membantu meningkatkan kepercayaan diri dalam kehidupan sehari-hari. Mereka merasa lebih percaya diri dan semangat menjalani hidup karena aplikasi ini memberikan informasi warna yang akurat. Namun, 10% responden lainnya merasa bahwa aplikasi ini belum sepenuhnya membantu meningkatkan kepercayaan diri mereka. Oleh karena itu, pengembangan lebih lanjut diperlukan untuk memastikan

aplikasi ini dapat memberikan manfaat yang optimal bagi semua pengguna. Dalam pengembangan aplikasi pendeteksi warna untuk aksesibilitas buta warna parsial, terdapat beberapa area yang dapat ditingkatkan dan dieksplorasi lebih lanjut guna memperbaiki kinerja dan kegunaan aplikasi. Berikut adalah beberapa saran untuk pengembangan lebih lanjut:

1. Peningkatan Algoritma Deteksi Warna:

Algoritma HSV yang digunakan saat ini memiliki keterbatasan dalam kondisi pencahayaan rendah. Menggabungkan metode lain, seperti algoritma pembelajaran mesin (machine learning) atau deep learning, dapat meningkatkan akurasi deteksi warna dalam berbagai kondisi pencahayaan. Selain itu, kalibrasi otomatis berdasarkan kondisi pencahayaan dapat diimplementasikan untuk menyesuaikan parameter deteksi warna secara dinamis.

2. Penggunaan Kamera dengan Resolusi Lebih Tinggi:

Menggunakan kamera dengan resolusi lebih tinggi dapat meningkatkan kualitas video stream dan memungkinkan deteksi warna yang lebih akurat, terutama untuk objek yang berukuran kecil atau memiliki detail warna yang halus.

3. Pengembangan Antarmuka Pengguna Lebih Intuitif:

Meskipun antarmuka saat ini sudah cukup interaktif, penambahan tur seperti mode malam (dark mode) atau tampilan high contrast dapat membantu pengguna dengan gangguan penglihatan. Selain itu, implementasi tur voice command untuk mengontrol aplikasi dapat meningkatkan aksesibilitas bagi pengguna dengan disabilitas sik.

4. Integrasi dengan Teknologi Augmented Reality (AR):

Integrasi aplikasi dengan teknologi AR dapat memberikan pengalaman pengguna yang lebih kaya, misalnya dengan menampilkan informasi warna secara langsung di objek yang

- terlihat melalui kamera. Ini dapat membantu pengguna memahami konteks warna dalam situasi nyata dengan lebih baik.
5. Penambahan Fitur Pembelajaran Warna:
Menambahkan tur di mana pengguna dapat belajar dan menguji pengetahuan mereka tentang warna. Ini bisa berupa modul edukasi interaktif yang memberikan informasi tentang berbagai warna dan bagaimana mereka digunakan dalam kehidupan sehari-hari.
 6. Ekspansi ke Platform Mobile:
Mengembangkan versi aplikasi untuk perangkat mobile (Android dan iOS) dapat meningkatkan jangkauan pengguna, mengingat banyak orang yang lebih sering menggunakan ponsel pintar dalam kehidupan sehari-hari.
 7. Penggunaan Data Historis:
Mencatat dan menganalisis data historis deteksi warna dapat memberikan wawasan lebih lanjut tentang pola penggunaan dan kebutuhan pengguna. Fitur ini dapat digunakan untuk meningkatkan pengalaman pengguna secara keseluruhan, misalnya dengan memberikan rekomendasi berdasarkan riwayat penggunaan.
 8. Kolaborasi dengan Ahli Kesehatan dan Peneliti:
Berkolaborasi dengan ahli kesehatan mata dan peneliti dalam bidang aksesibilitas dapat memberikan masukan berharga untuk memperbaiki dan mengoptimalkan aplikasi sesuai dengan kebutuhan pengguna buta warna parsial.
 9. Uji Coba dan Pengujian Lebih Luas:
Melakukan uji coba aplikasi dengan kelompok pengguna yang lebih besar dan beragam untuk mendapatkan umpan balik yang lebih komprehensif. Hal ini dapat membantu mengidentifikasi area yang masih memerlukan perbaikan dan memastikan aplikasi dapat digunakan secara efektif oleh berbagai jenis pengguna.

Dengan mengadopsi saran-saran di atas, diharapkan aplikasi pendeteksi warna ini dapat berkembang menjadi alat yang lebih canggih dan berguna, memberikan manfaat yang lebih besar bagi individu dengan buta warna parsial dalam kehidupan sehari-harinya.

DAFTAR PUSTAKA

- [1] EMI SITA ERIANA and Afrizal Zein . Arti - cial intelligence (AI). EUREKA MEDIA AK- SARA, 24 November 2023.
- [2] Ahmad Roihan, Abas Sunarya, And Ageng Setiani Ra ka. Pemanfaatan Machine Learn- ing Dalam Berbagai Bidang:Review Paper. IJCIT(Indonesian Journal On Computer And Information Technology), 22 April 2022.
- [3] Fadhli Rizal Makarim. Buta Warna. Halodoc, 2024.
<https://www.halodoc.com/kesehatan/buta-warna>.
- [4] Romadhon, Syahru, Irma, Khairuna, So- fyana, Fitri, Putri, And Helmalia. Hubungan Penyuluhan Kesehatan Mata Dengan Tingkat Pengetahuan Siswa/I Tentang Buta Warna Di Smp Perguruan Karya Bunda. Jurnal Darma Agung, 29(2):332, August 2021.
- [5] Goenawan, Aaqila Dhiyaanisafa, Rachman, M. Bakhara Alief, Pulungan, And Mutiara Per- sada. Identi kasi Warna Pada Objek Citra Digital Secara Real Time Menggunakan Pengo- lahan Model Warna Hsv. Jurnal Teknik Infor- matika Dan Elektro, 4(1):68 74, January 2022.
- [6] Mohammad Syamsul Azis, Lukmanul Hakim, And Walim. Perancangan Aplikasi Berbasis Desktop Denganmicrosoft Visual Basic (Studi Kasus: Aplikasiabsensi Anak Magang 1.0). JU- RNAL RESPONSIF, 20 Februari 2020.
- [7] M.Zikri Andrekha And Yasdinul Huda. De- teksi Warna Manggis Menggunakan Pengola- han Citra Dengan Opencv Python. Jurnal Vo-

- cational Teknik Elektronika Dan Informatika, 4 Desember 2021.
- [8] Panggabean, Adelia Kartika, Syahfaridzah, Al- dina, Ardiningih, And Nabila Ayu. Mende- teksi Objek Berdasarkan Warna Dengan Segmentasi Warna Hsv Menggunakan Aplikasi Matlab. METHOMIKA Jurnal Manajemen Informatika Dan Komputerisasi Akuntansi, 4(2):94 97, October 2021.
- [9] Microsoft. Visual studio code. Microsoft (Windows, MacOS and linux), 2024. <https://code.visualstudio.com/learn>.
- [10] Muhammad Romzi And Budi Kurniawan. Implementasi Pemrograman Python Menggu- nakan Visual Studio Code. Jik, Desember 2020.
- [11] Kurnia, Rahmi Putri, Atma, And Yori Adi. Analisis Rekomendasi Film Dari Data Imdb Menggunakan Python. Journal Of Information System, Computer Science And Information Technology, 3(2):23 28, 2022.
- [12] M. Zikri Andrekha and Yasdinul Huda. De- teksi warna manggis menggunakan pengolahan citra dengan opencv python. Jurnal Vocational Teknik Elektronika dan Informatika, 4 Desem- ber 2021.
- [13] Mohamad Amin, Joko Triyanto, And Istofa. Rancangan Perangkat Lunak Akuisisi Data Modul Detektorgamma Rosrao Berbasis Modbus Over Tcp/Ip Menggunakan Pyqt5. PRIMA, 2020.
- [14] Harris, Charles R., Millman, K. Jarrod, Van Der Walt, Stacfan J., Gommers,Ralf, Virtanen Pauli, Cournapeau David, Wieser Eric, Taylor Julian, Berg Sebastian, Smith Nathaniel J., Kern Robert, Picus Matti, Hoyer Stephan, Van Kerkwijk Marten H., Brett Matthew, Haldane Allan, Del Rao Jaime Ferna Ndez, Wiebe Mark, Peterson Pearu, Pierre, Garard Marchant,Sheppard Kevin, Reddy Tyler, Weckesser Warren, Abbasi Hameer, Gohlke Christoph, And Oliphant Travis E. Array Programming With Numpy. Nature, 585(7825):357 362, 2020.
- [15] Taylor. Playsound Library In Python. Python Software Foundation, 2021. <https://Pypi.Org/Project/Playsound/>
- [16] Python Software Foundation. Date- time Python. Datetime Python, 2021. <https://Docs.Python.Org/3/Library/Datetime.Html>.
- [17] Dias Ayu Budi Utami. Perancangan Sistem Lo- gin Pada Aplikasi Berbasis Guimenggunakan Qtdesigner Python. Jurnal SIMADA, 2021.