ELKOMIKA: : Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika

ISSN(p): 2338-8323 | ISSN(e): 2459-9638 | Vol. 9 | No. 2 | Halaman 359 - 373
DOI : http://dx.doi.org/10.26760/elkomika.v9i2.359 April 2021

Pendeteksi Golongan Darah Manusia Berbasis Tensorflow menggunakan ESP32-CAM

HUMMAM GHASSAN GHIFARI, DENNY DARLIS, ARIS HARTAMAN

Teknologi Telekomunikasi Universitas Telkom, Indonesia
Email: hummamghassan@student.telkomuniversity.ac.id

Received 1 Oktober 2020 **|** *Revised* 20 Oktober 2020 **|** *Accepted* 25 Desember 2020

ABSTRAK

Pendeteksian golongan darah dilakukan untuk mengetahui golongan darah yang dimiliki. Hingga saat ini pendeteksian golongan darah masih dilakukan oleh petugas analis kesehatan menggunakan kemampuan mata manusia. Pada penelitian ini dilakukan perancangan alat pendeteksi golongan darah menggunakan ESP32-CAM. Alat ini menggunakan kamera OV2640 untuk menangkap citra, yang diproses menggunakan Tensorflow Object Detection API sebagai framework untuk melatih serta mengolah citra darah. Model latih akan digunakan pada kondisi pendeteksian langsung dan ditampilkan dalam bentuk jendela program golongan darah beserta tingkat akurasinya. Dalam penelitian ini pengujian dilakukan menggunakan 20 dataset dengan jarak pengukuran antara ESP32-CAM dengan citra golongan darah yaitu sejauh 20 cm. Hasil yang didapat selama pengujian mayoritas golongan darah yang dapat terdeteksi adalah golongan darah AB.

***Kata kunci**: ESP32-CAM, Tensorflow, Python, Golongan Darah, Pengolahan Citra*

ABSTRACT

Blood group detection is performed to determine the blood group. Currently, in detecting blood type, it still relies on the ability of the human eyeThis paper presents a human blood group detection device using ESP32-CAM. This tool uses ESP32-CAM to capture images, and the Tensorflow Object Detection API as a framework used to train and process an image. The way this tool works is that the ESP32-CAM will capture an image of the blood sample and then send it via the IP address. Through the IP Address, the python program will access the image, then the image will be processed based on a model that has been previously trained. The results of this processing will be displayed in the form of a window program along with the blood type and level of accuracy. In this study, testing was carried out based on the number of image samples, the number of datasets, and the measurement distance. The ideal measurement distance between the ESP32-CAM and the blood group image is 20 cm long. The results obtained during the testing of the majority of blood groups that can be detected are AB blood group.

***Keywords**: ESP32-CAM, Tensorflow, Python, Blood Type, Image Processing*

1. PENDAHULUAN

Penyesuaian golongan darah manusia menjadi syarat penting agar dapat mendonorkan darah atau menerima transfusi darah, hal ini disebabkan oleh ketidakcockan semua golongan darah satu sama lain. Mendonorkan ataupun menerima darah yang tidak sesuai dengan golongan darah yang dimiliki dapat memicu penggumpalan darah serta mengakibatkan komplikasi yang fatal bagi tubuh manusia. Pengecekan golongan darah manusia terbagi menjadi 2 jenis, yaitu sistem ABO dan sistem Rhesus (Rh). Biasanya pendeteksian golongan darah yang sering digunakan yaitu dengan menggunakan metode ABO **(Lering, 2013)**. Pengecekan dengan melakukan reaksi antara sampel darah manusia dengan cairan serum anti-A dan anti-B, merupakan pengujian yang umum dilakukan untuk menentukan suatu golongan darah manusia. Perubahan yang terjadi dapat dibedakan berdasarkan menggumpal atau tidak menggumpalnya darah, dari perubahan tersebut akan menentukan hasil tipe golongan darah.

Saat ini dalam menentukan golongan darah manusia masih mengandalkan kemampuan visual, sehingga hasil keakuratannya bergantung pada mata penguji **(Retyanto, dkk, 2018)**. Kesalahan dalam penentuan golongan darah bisa saja terjadi apabila pengujian sampel dalam jumlah yang banyak. Hilangnya konsentrasi serta kelelahan menjadi faktor kesalahan dalam pendeteksian golongan darah. Solusi untuk mengatasi masalah tersebut yaitu dengan melakukan pendeteksian golongan darah dengan berbasiskan *Convolutional Neural Network*. Berdasarkan masalah tersebut, penulis membuat alat yang bertujuan untuk mempermudah pendeteksian golongan darah manusia dengan menggunakan ESP32-CAM.

Pada tahun 2018, **(Hendri, 2018)** menggunakan algoritma *Convolutional Neural Network* dan *framework* tensorflow untuk mendeteksi asap dan api. Akan tetapi pada penelitian tersebut sistem pendeteksian masih menggunakan citra sebagai *input* beserta *output*. Sedangkan pada penelitian untuk mendeteksi meja dan kursi, **(Dewi, 2018)** menggunakan video sebagai input untuk mendeteksi objek. Oleh karena itu, diperlukan pengembangan sistem pendeteksian dengan menggunakan *real time video processing.* Pada penelitian ini digunakan *framework* Tensorflow *Object Detection* API. Tensorflow merupakan *interface* untuk mengeksekusi perintah dengan menggunakan informasi yang dimiliki mengenai objek yang dikenali serta dapat membedakan antara objek satu dengan objek lainnya **(Nurfita & Ariyanto, 2018)**. *Framework* ini mendukung arsitektur *Neural Networks* (NN) yang umum digunakan seperti *Recurrent Neural Networks*, *Convolutional Neural Networks*, dan *Deep Belief Networks*. Tensorflow memiliki fitur untuk menjalankan proses pelatihan suatu *model* dengan menggunakan *Central Processing Unit* (CPU) ataupun *Graphic Processing Unit* (GPU).

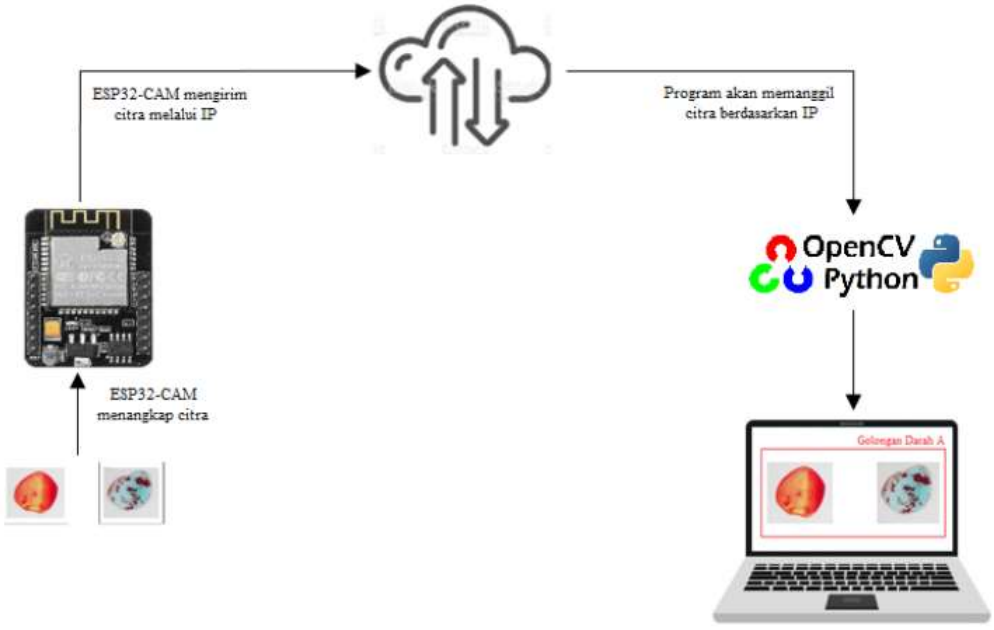
Deep Learning merupakan cabang dari *Machine Learning* yang terinspirasi dari kortex manusia dengan menerapkan jaringan syaraf buatan yang memiliki banyak *hidden layer* **(Santoso & Ariyanto, 2018)**. Metode *Deep Learning* yang akan digunakan pada penelitian ini yaitu *Convolutional Neural Network* (CNN). Metode ini memiliki hasil yang lebih signifikan dalam pengenalan citra **(Suartika, dkk, 2016)**. CNN terdiri dari dua metode, yaitu tahap klasifikasi dengan menggunakan *feedward* serta tahap pembelajaran menggunakan *backpropagation* **(Salawazo, dkk, 2019)**. Jaringan ini menggunakan masukan berupa citra, kemudian akan melalui lapisan konvolusi dan diolah berdasarkan filter yang telah ditentukan, di tiap lapisan tersebut akan menghasilkan pola dari beberapa bagian citra yang memudahkan proses klasifikasi **(Danukusumo, 2017)**. CNN lebih efektif dalam menyelesaikan masalah yang berhubungan dengan citra, sehingga hampir seluruh sistem *Machine Learning* yang berhubungan dengan citra saat ini berbasiskan CNN **(Harjoseputro, 2018).**

Algoritma CNN merupakan perkembangan dari *Multi Layer Perceptron* (MLP). CNN dirancang untuk mengolah data dalam bentuk grid **(Kusumaningrum, 2018)**. Pada CNN tiap *neuron* direpresentasikan dalam bentuk dua dimensi contohnya seperti citra atau suara, tidak seperti MLP yang tiap neuron hanya berukuran satu dimensi **(Rena, 2019)**. Pada kasus klasifikasi citra, MLP kurang sesuai untuk digunakan, karena tidak menyimpan informasi spasial dari data citra serta menganggap setiap piksel merupakan fitur yang independen sehingga hasil yang didapatkan kurang baik **(Pujoseno, 2018)**. CNN banyak digunakan dalam penelitian dengan metode *image processing* (pengolahan citra). Algoritma ini memiliki tingkat akurasi yang tinggi dalam mendeteksi objek maupun klasifikasi citra. CNN dapat mendukung layanan seperti pencarian citra, pengenalan suara, juga pemrosesan bahasa alami (NLP) **(Ramadhan, 2020)**.

2. METODOLOGI PENELITIAN

2.1 Rancangan Sistem

Alat ini dirancang menggunakan ESP32-CAM sebagai alat untuk menangkap citra, serta Tensorflow *Object Detection* API sebagai *framework* yang digunakan untuk melatih serta mengolah suatu citra. ESP32-CAM akan menangkap citra sampel darah lalu hasil dari citra tersebut akan dikirimkan melalui IP *Address*. Setelah itu, program python akan mengakses citra secara *real time* melalui IP *Address*. Pada program tersebut citra akan diproses berdasarkan *model* yang telah dilatih sebelumnya. Hasil dari pendeteksian tersebut akan muncul berupa *model* beserta kelas dan tingkat akurasinya. Berikut merupakan rancangan sistem yang akan ditunjukkan pada Gambar 1.

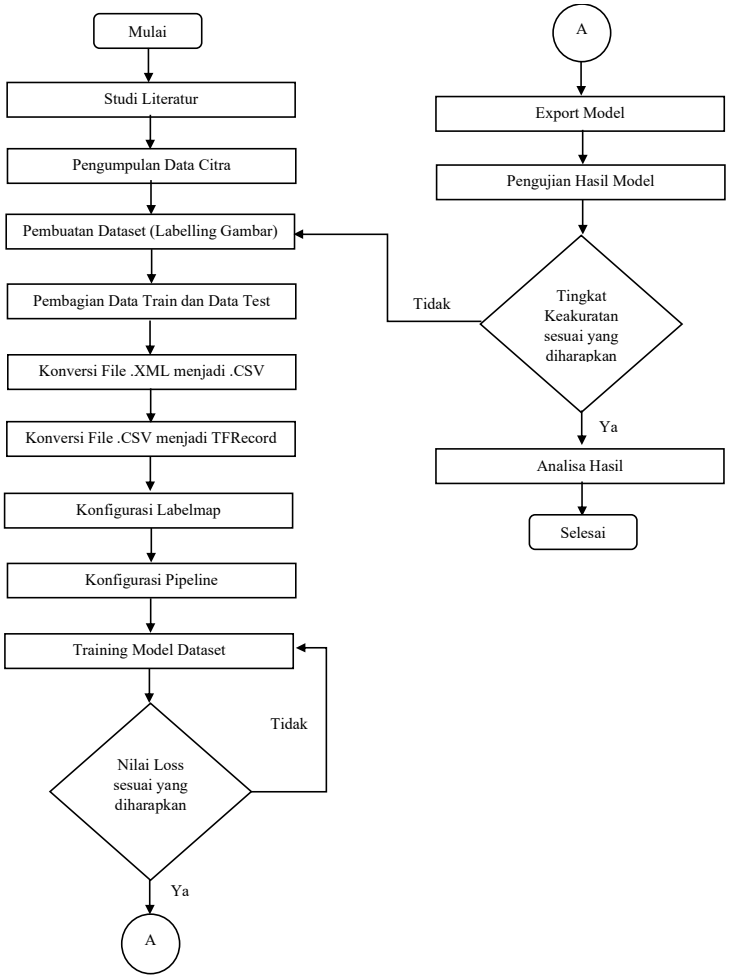


Gambar 1. Rancangan Sistem

2.2 Diagram Alir Tahapan Perancangan

Pada perancangan alat ini, terdapat beberapa tahapan perancangan. Dimulai dari melakukan studi literatur untuk merumuskan masalah, setelah itu melakukan pembuatan *dataset*, lalu tahap selanjutnya mengkonfigurasi dan melakukan *training* agar citra dapat terdeteksi sesuai

dataset, kemudian melakukan perancangan alat dan melakukan pengujian. Tahapan perancangan tersebut akan dijelaskan dalam bentuk diagram alir seperti pada Gambar 2.



Gambar 2. Diagram Alir Tahapan Perancangan Alat

2.3 Perancangan Penelitian

2.3.1 Perancangan ESP32-CAM

Untuk menghubungkan antara ESP32-CAM dengan USB to TTL digunakan *jumper* tipe *female to female* dengan pin RX ke U0T dan TX ke U0R pada ESP32-CAM, untuk catu dayanya sendiri menghubungkan antara pin GND ke GND dan 5V ke 5V, agar dapat mengunggah pemrograman pin IO0 harus terhubung ke GND pada ESP32-CAM.

2.3.2 Pemrograman ESP32-CAM

Agar dapat memprogram ESP32-CAM, digunakan *software* Arduino IDE beserta *board library* ESP32. Sebelum mengunggah pemrograman pastikan pin GPIO pada ESP32-CAM dihubungkan dengan pin GND. Jika pin tersebut telah saling tehubung, maka ESP32 akan masuk ke dalam mode *flashing* sehingga dapat mengunggah pemrograman. Apabila pemrograman telah selesai diunggah, pin tersebut harus dilepas terlebih dahulu agar ESP32-CAM dapat menjalankan pemrograman.

2.3.3 Pembuatan *Dataset*

Dalam tahap ini, objek pada citra *dataset* akan diberikan label yang bertujuan untuk menentukan letak objek yang ingin dideteksi sebagai golongan darah serta menyimpan informasi citra dalam berkas file berekstensi .XML dengan format berupa PASCAL VOC. Setelah dilakukan pemberian label, file dengan format .XML yang telah dibuat perlu di konversi menjadi .CSV dengan tujuan untuk mengambil data citra dari label yang telah di berikan, yang nantinya akan digunakan untuk menghasilkan berkas TFRecord.

Setelah proses konversi file .XML menjadi file .CSV, diperlukan proses konversi menjadi Tensorflow *Record* file agar *dataset* dapat dibaca oleh tensorflow. Konversi CSV ke TFRecord perlu dilakukan karena pada saat proses *training*, tensorflow akan membaca data *input* (proses *feeding data*) yang diambil dari *dataset* dalam format TFRecord **(Novyantika, 2018).** Labelmap digunakan untuk memberikan pelabelan pada objek yang akan dideteksi. File labelmap disimpan pada berkas dengan format file ".pbtxt" yang dibutuhkan untuk konfigurasi *pipeline.* Konfigurasi *pipeline* berguna untuk mengatur algoritma dari *Convolutional Neural Network* (CNN) serta mengkonfigurasi proses pelatihan (*training*) dan evaluasi (*testing*) dengan menggunakan ProtoBuf, sehingga diperlukan konfigurasi *pipeline* terlebih dahulu. *Pre-trained model* yang digunakan dalam pembuatan *pipeline* ini yaitu *Faster* RCNN *Inception* v2.

2.3.4 *Training Model*

Tahap *training* merupakan tahap utama dimana sebuah *Neural Network* dilatih untuk mempelajari pola dari suatu objek, sehingga dapat menghasilkan pengenalan untuk mendeteksi objek dengan tingkat akurasi yang tinggi. Tahap awal dalam proses *training* yaitu memasukkan data *training* ke dalam *framework* tensorflow. Saat proses *training model*, tensorflow akan menghasilkan *checkpoint* secara otomatis yang berbentuk *graph tensor*, *checkpoint* ini berguna untuk menyimpan informasi pada saat proses *training model*. Setelah proses *training* selesai, hasil dari *checkpoint* terakhir akan dilakukan konversi menjadi file berupa *frozen inference graph* sehingga dapat digunakan untuk memprediksi sebuah citra yang disediakan.

2.4 Pengujian Proses Latih

Model dataset dibuat dengan menggunakan sampel citra sebanyak 10 citra tiap golongan darah. Hasil dari pengujiannya sendiri seluruh sampel dapat terdeteksi sesuai dengan golongan darahnya. Sampel yang digunakan untuk pengujian dilakukan dengan menggunakan sampel *dataset* untuk proses latih. Pada Tabel 1 ditunjukan hasil dari pengujian keakuratan yang dilakukan pada 10 sampel pengujian.

Tabel 1. Tabel Kalibrasi Tingkat Keakuratan Alat

Pengujian ke	Sampel Golongan Darah yang Terdeteksi			
	A	AB	B	O
1	96%	100%	97%	99%
2	97%	100%	85%	100%
3	78%	98%	93%	100%
4	77%	99%	81%	100%
5	89%	91%	82%	99%
6	87%	92%	94%	99%
7	96%	94%	91%	100%
8	92%	100%	97%	99%
9	91%	100%	73%	99%
10	93%	100%	95%	99%

3. HASIL DAN PEMBAHASAN

3.1 Realisasi Alat

Dari hasil perancangan, maka direalisasikan alat pendeteksi golongan darah manusia menggunakan ESP32-CAM. Tampilan dari alat pendeteksi ini dapat ditunjukkan pada Gambar 3 berikut ini.

Gambar 3. Tampilan Akhir Alat Pendeteksi Golongan Darah

Untuk proses pengujian alat pendeteksi golongan darah dapat ditunjukkan pada Gambar 4.

Gambar 4. Pengujian Pendeteksi Golongan Darah

3.2 Uji Coba dan Evaluasi Hasil

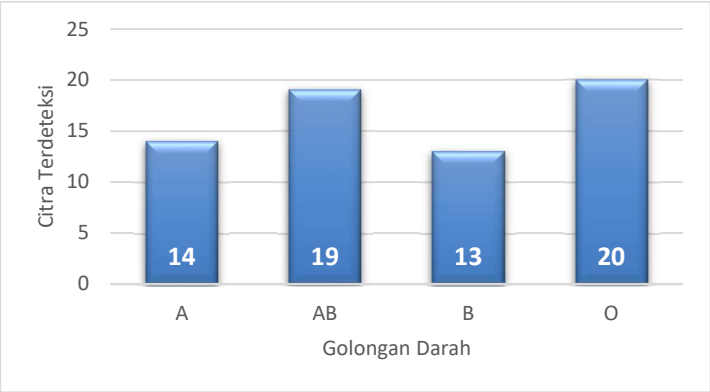
Pada uji coba dan evaluasi ini, pengujian difokuskan untuk mencapai nilai keakuratan tertinggi dari sampel golongan darah yang terdeteksi. Pengujian ini dilakukan untuk mendapatkan hasil seakurat mungkin berdasarkan jumlah sampel citra, jumlah dataset serta jarak pengukuran. Dalam pengujian ini ESP32-CAM akan mendeteksi citra golongan darah sesuai dengan jarak pengukuran antara ESP32-CAM dengan citra, hasil dari pendeteksian tersebut akan muncul berupa *model* beserta tipe golongan darah dan tingkat keakuratan dari golongan darah tersebut.

3.2.1 Percobaan Pertama

Pada percobaan pertama dilakukan pengujian 20 sampel darah A, B, AB dan O pada jarak pendeteksian sejauh 20 cm dan 30 cm seperti ditunjukkan pada Tabel 2. Jumlah dataset yang digunakan adalah 90 citra. Untuk jarak deteksi 20cm hasilnya ditunjukkan pada Gambar 5, dimana golongan darah A terdeteksi sebanyak 14 sampel, golongan darah AB sebanyak 19 sampel, golongan darah B terdeteksi sebanyak 13 sampel dan untuk golongan darah O dapat seluruhnya terdeteksi.

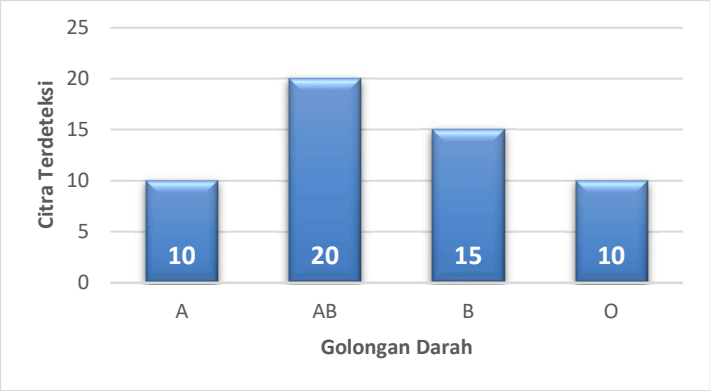
Tabel 2. Percobaan Pertama dengan 20 Sampel dan 90 Citra

Sampel Digunakan	Jarak Pengukuran	Dataset Digunakan	Golongan Darah yang Terdeteksi			
			Darah A	Darah AB	Darah B	Darah O
20 sampel	20 cm	90 citra	14	19	13	20
20 sampel	30 cm	90 citra	10	20	15	10



Gambar 5. Grafik Pengujian I pada jarak 20cm

Pada pengujian pertama ini seluruh golongan darah dapat terdeteksi, hanya saja untuk golongan darah dan A dan B belum dapat terdeteksi secara menyeluruh. Untuk nilai akurasinya sendiri seluruh golongan darah dapat terdeteksi dengan baik, terutama untuk golongan darah AB yang nilai akurasinya mencapai 100% per sampel. Pengujian pada jarak 30cm didapatkan golongan darah A dan O masing - masing terdeteksi sebanyak 10 sampel, golongan darah AB dapat terdeteksi seluruhnya, sedangkan untuk golongan darah B terdeteksi sebanyak 15 sampel. Hasil dari pengujian tersebut ditunjukkan pada Gambar 6.



Gambar 6. Grafik Pengujian II pada jarak 30cm

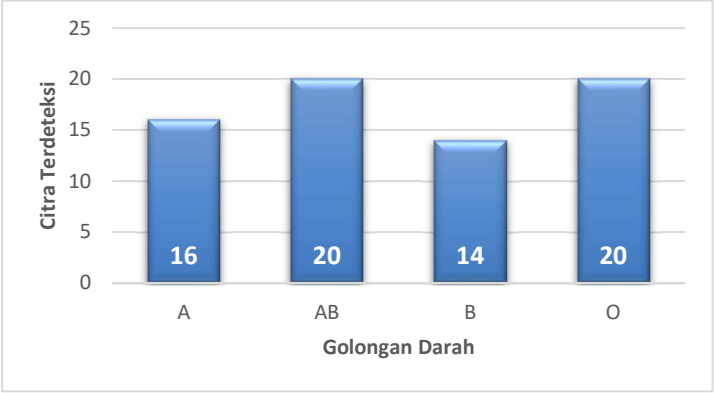
Untuk pengujian ini dapat dilihat bahwa golongan darah A dan O tidak terlalu akurat dalam mendeteksi sampel, sedangkan untuk golongan darah AB dapat terdeteksi dengan sangat akurat dengan nilai 100% tiap sampel.

3.2.2 Percobaan Kedua

Pada percobaan kedua yang ditunjukkan pada Gambar 7 hasil dari pengujiannya menggunakan data pada Tabel 3. Pada pengujian ini hasil yang didapat lebih baik dibandingkan pengujian sebelumnya yang dilakukan pada percobaan pertama. Berdasarkan Gambar 7, golongan darah A terdeteksi 16 sampel, sedangkan golongan darah B terdeteksi 14 sampel, dan untuk golongan darah B dan O seluruh sampel dapat terdeteksi.

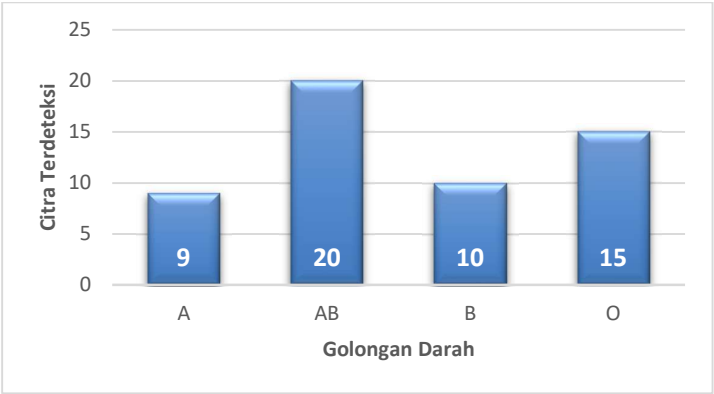
Tabel 3. Percobaan Kedua dengan 20 Sampel dan 90 Citra

Sampel Digunakan	Jarak Pengukuran	Dataset Digunakan	Golongan Darah yang Terdeteksi			
			Darah A	Darah AB	Darah B	Darah O
20 sampel	20 cm	90 citra	16	20	14	20
20 sampel	30 cm	90 citra	9	20	10	15



Gambar 7. Grafik Pengujian I pada jarak 20cm

Pada pengujian berikutnya, hasil dari golongan darah A hanya terdeteksi 9 citra dari 20 sampel, golongan darah AB terdeteksi 20 sampel, sedangkan untuk golongan darah B terdeteksi 10 sampel dan golongan darah O terdeteksi 15 sampel. Pada Gambar 8 merupakan hasil dari pengujian kedua.



Gambar 8. Grafik Pengujian II pada jarak 30cm

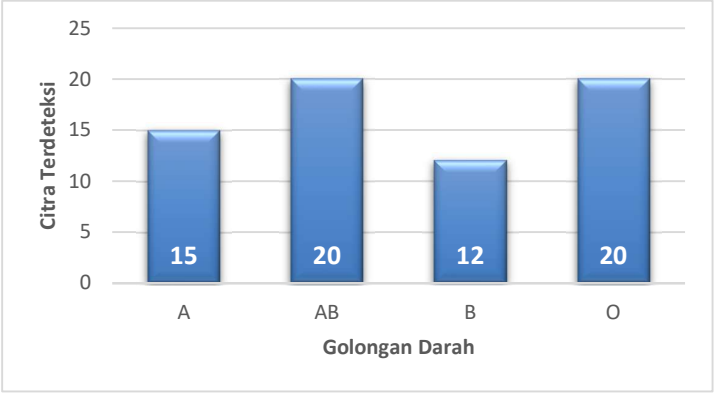
Dibandingkan dengan pengujian kedua pada percobaan pertama, pengujian ini dapat mendeteksi seluruh sampel dengan benar walaupun nilai akurasinya tidak terlalu baik.

3.2.3 Percobaan Ketiga

Pada percobaan ketiga, hasil yang didapat hasil tidak lebih baik dari sebelumnya. Hal ini dikarenakan terdapat beberapa golongan darah yang terdeteksi sebagai golongan darah lain, contohnya seperti golongan darah A yang mampu mendeteksi seluruh golongan darah tetapi 5 citra dari 20 sampel terdeteksi golongan darah AB. Untuk hasilnya sendiri, golongan darah A dapat dideteksi sebanyak 15 sampel, golongan darah B sebanyak 12 sampel, serta untuk golongan darah AB dan O seluruh sampel dapat terdeteksi. Hasil dari pengujian ketiga ditunjukkan pada Gambar 9.

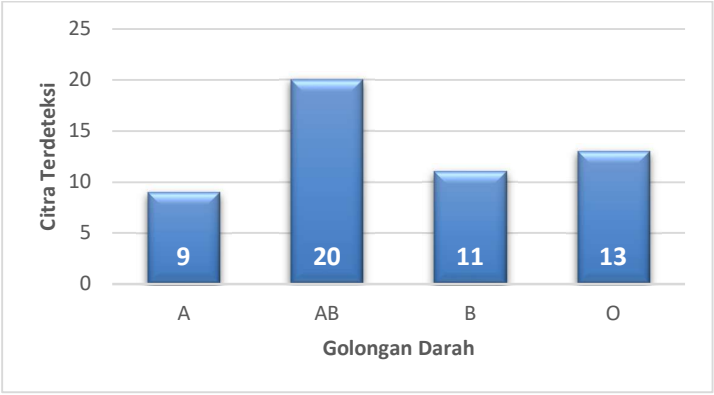
Tabel 4. Percobaan Ketiga dengan 20 Sampel dan 90 Citra

Sampel Digunakan	Jarak Pengukuran	Dataset Digunakan	Golongan Darah yang Terdeteksi			
			Darah A	Darah AB	Darah B	Darah O
20 sampel	20 cm	90 citra	15	20	12	20
20 sampel	30 cm	90 citra	9	20	11	13



Gambar 9. Grafik Pengujian I pada jarak 20cm

Pada jarak 30cm, didapatkan hasil yang lebih buruk dibandingkan kedua percobaan sebelumnya. Golongan darah A hanya 9 sampel yang terdeteksi, golongan darah B sebanyak 11 sampel, dan untuk golongan darah O 13 sampel yang terdeteksi. Hasil pengujian tersebut ditunjukkan pada Gambar 10.



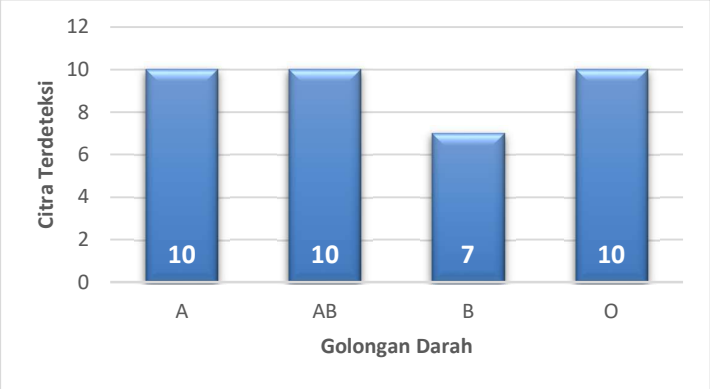
Gambar 10. Grafik Pengujian II pada jarak 30cm

3.2.4 Percobaan Keempat

Pada pengujian keempat menggunakan 10 sampel citra, golongan darah A, AB, dan O dapat terdeteksi seluruhnya, sedangkan untuk golongan darah B hanya 7 dari 10 sampel citra yang terdeteksi. Pada Gambar 11 merupakan hasil dari pengujian pertama.

Tabel 5. Percobaan Keempat dengan 10 Sampel dan 10 Citra

Sampel Digunakan	Jarak Pengukuran	Dataset Digunakan	Golongan Darah yang Terdeteksi			
			Darah A	Darah AB	Darah B	Darah O
10 sampel	20 cm	10 citra	10	10	7	10
10 sampel	30 cm	10 citra	4	10	8	3



Gambar 11. Grafik Pengujian I pada jarak 20cm

Hasil dari pengujian ini, pada golongan darah B mengalami kesalahan pendeteksian sebanyak 3 sampel citra yang dimana mendeteksi golongan darah O.

Pada pengujian ini hasil dari golongan darah A terdeteksi 4 citra dari 10 sampel, golongan darah AB dapat terdeteksi seluruh sampel, golongan darah B terdeteksi 8 sampel, dan untuk golongan darah O hanya terdeteksi 3 sampel seperti ditunjukkan pada Gambar 12.

Gambar 12. Grafik Pengujian II pada jarak 30cm

Hasil dari pengujian ini golongan darah A, B, dan O mengalami kesalahan pendeteksian. Pada golongan darah A mengalami kesalahan deteksi sebanyak 6 sampel dengan mendeteksi golongan darah AB, golongan darah B mendeteksi golongan darah AB sebanyak 2 sampel, sedangkan golongan darah O mendeteksi golongan darah B sebanyak 7 sampel.

3.2.5 Percobaan Kelima

Pada pengujian ini hanya golongan darah AB dan O yang dapat terdeteksi seluruhnya, sedangkan untuk golongan darah A dan B masing-masing 8 citra dari 10 sampel citra yang terdeteksi seperti ditunjukkan pada Gambar 13.

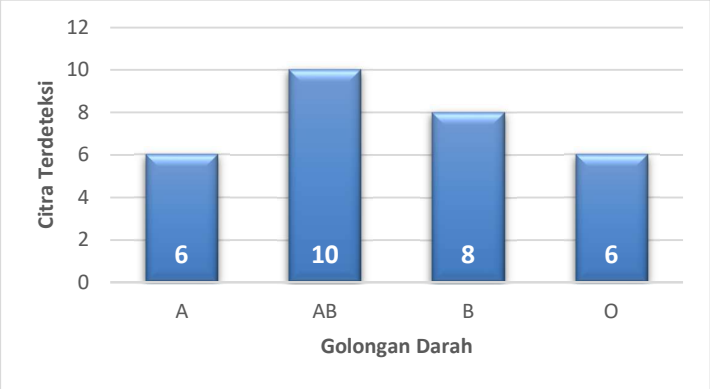
Tabel 6. Percobaan Kelima dengan 10 Sampel dan 10 Citra

Sampel Digunakan	Jarak Pengukuran	Dataset Digunakan	Golongan Darah yang Terdeteksi			
			Darah A	Darah AB	Darah B	Darah O
10 sampel	20 cm	10 citra	8	10	8	10
10 sampel	30 cm	10 citra	6	10	8	6

Gambar 13. Grafik Pengujian I pada jarak 20cm

Golongan darah A dan B masing-masing mengalami kesalahan pendeteksian sebanyak 2 sampel. Hasil dari pengujian ini tingkat akurasi yang didapat dari golongan darah B nilainya tidak terlalu bagus.

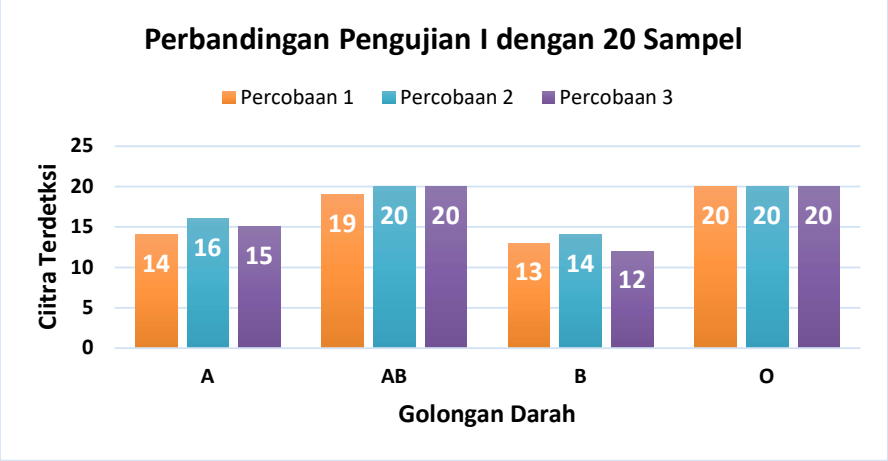
Pada pengujian ini hasil yang didapatkan lebih baik dibandingkan dengan pengujian sebelumnya pada percobaan keempat, golongan darah A dan O terdeteksi 6 sampel, golongan darah AB dapat terdeteksi seluruhnya, dan untuk golongan darah B terdeteksi 8 sampel seperti ditunjukkan Gambar 14.



Gambar 14. Grafik Pengujian II pada jarak 30cm

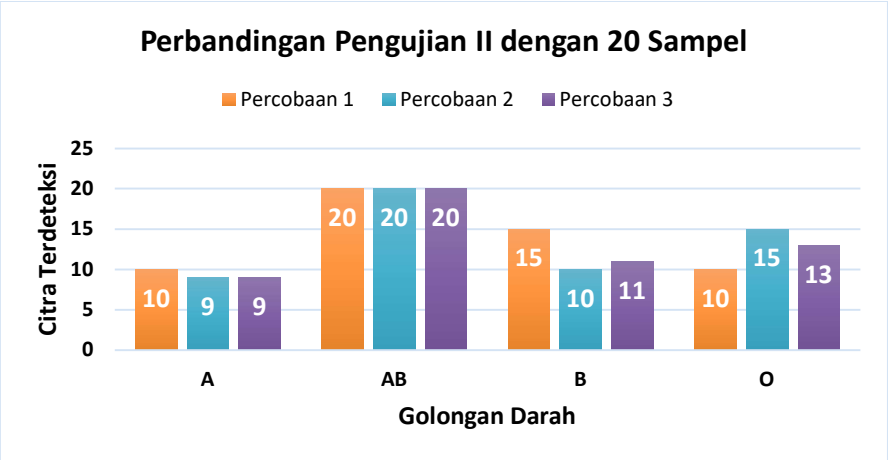
3.6 Analisis Perbandingan

Berdasarkan Gambar 15, pada pengujian pertama didapatkan hasil dari perbandingan antara percobaan pertama hingga percobaan ketiga. Hasil dari pengujian ini, didapatkan golongan darah dengan hasil tingkat keakuratan sebesar 100% untuk golongan darah AB pada percobaan kedua dan percobaan ketiga. Sedangkan untuk golongan darah O tingkat keakuratan didapatkan sebesar 100% pada seluruh percobaan. Hal ini dibuktikan dengan hampir seluruh sampel citra darah dapat terdeteksi dengan baik, sedangkan pada golongan darah A dan B didapat hasil yang cukup buruk. Kondisi ini dikarenakan kurangnya *dataset* yang mendukung agar golongan darah A dan B dapat terdeteksi oleh program.



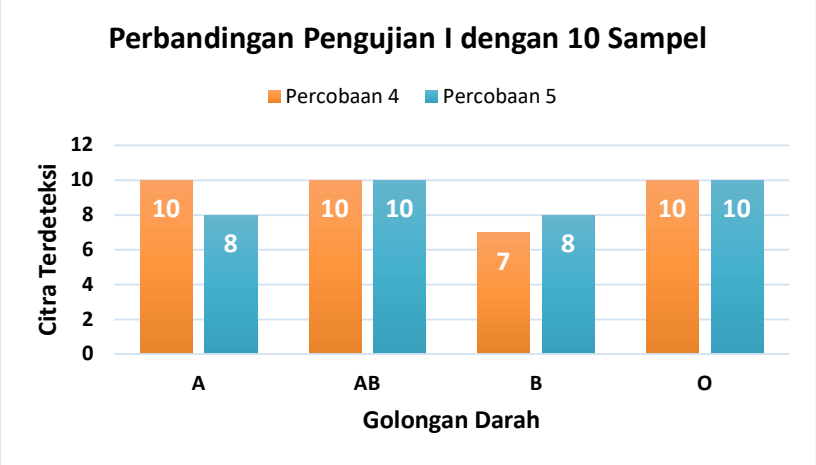
Gambar 15. Perbandingan Pengujian I pada Percobaan 1, Percobaan 2, dan Percobaan 3 (Jarak 20 cm dengan 90 Dataset, dan 20 Sampel)

Dari Gambar 16, dapat dilihat bahwa sampel citra yang seluruhnya terdeteksi hanya pada golongan darah AB dengan tingkat keakuratan sebesar 100% sedangkan untuk golongan darah A, B dan O hasil yang didapat cukup buruk. Berdasarkan grafik tersebut dapat disimpulkan bahwa semakin jauh jarak deteksi antara kamera dengan sampel citra maka semakin buruk hasil yang didapat, terkecuali untuk golongan darah AB yang mendeteksi seluruh sampel dari ketiga percobaan.

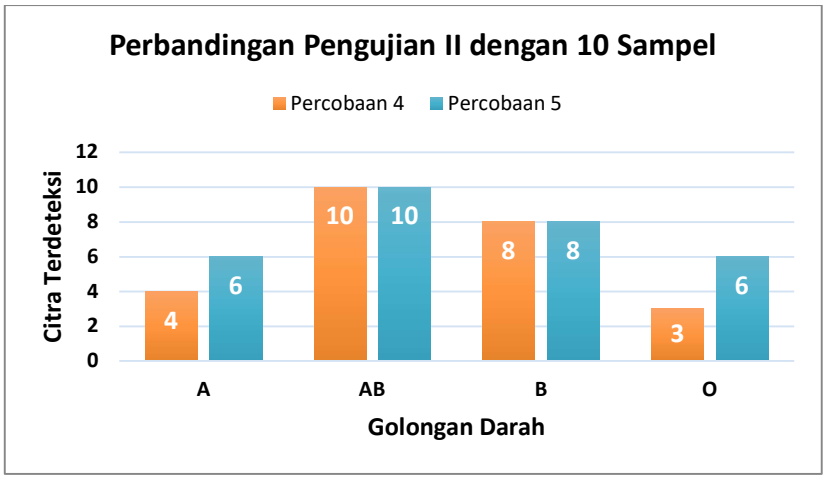


Gambar 16. Perbandingan Pengujian II pada Percobaan 1, Percobaan 2, dan Percobaan 3 (Jarak 30 cm dengan 90 Dataset dan 20 Sampel)

Pada Gambar 17, dapat dilihat hasil dari pengujian pertama antara percobaan keempat dengan percobaan kelima. Hasil dari pengujian ini didapatkan tingkat keakuratan golongan darah sebesar 100% untuk golongan darah A pada percobaan pertama. Sedangkan untuk golongan darah AB dan O tingkat keakuratan didapatkan sebesar 100% pada seluruh percobaan. Dapat disimpulkan bahwa pada pengujian ini golongan darah AB dan O dapat terdeteksi dengan sangat baik.



Gambar 17. Perbandingan Pengujian I pada Percobaan 4 dan Percobaan 5 (Jarak 20 cm dengan 10 Dataset dan 10 Sampel)



Gambar 18. Perbandingan Pengujian II pada Percobaan 4 dan Percobaan 5 (Jarak 30 cm dengan 10 Dataset dan 10 Sampel)

Berdasarkan Gambar 18, didapatkan hasil perbandingan antara percobaan keempat dan percobaan kelima dengan jarak sepanjang 30 sentimeter. Hasil yang didapat dari kedua percobaan tersebut, tingkat keakuratan golongan darah AB dapat terdeteksi sebesar 100% pada kedua percobaan. Untuk golongan darah A, B, dan O hasil yang didapat lebih buruk dibandingkan dengan pengujian sebelumnya. Hal ini dapat dibuktikan dengan jumlah hasil sampel citra yang terdeteksi jauh lebih sedikit. Dari pengujian tersebut disimpulkan bahwa jarak mempengaruhi tingkat akurasi yang didapat.

4. KESIMPULAN

Berdasarkan analisis pengujian yang telah dilakukan maka didapatkan beberapa kesimpulan, yaitu berdasarkan percobaan yang dilakukan dengan validasi jarak, jumlah sampel dan jumlah dataset yang dilakukan, mayoritas golongan darah yang dapat terdeteksi adalah golongan AB. Berdasarkan hasil tingkat keakuratan, pada pengujian dengan menggunakan 90 dataset dan 20 sampel, pengujian pertama didapatkan golongan darah dengan hasil tingkat keakuratan sebesar 100% untuk golongan darah O pada ketiga percobaan. Sedangkan untuk pengujian kedua, sampel citra golongan darah B terdeteksi sebesar 100% pada seluruh percobaan. Untuk pengujian dengan menggunakan 10 dataset dan 10 sampel, pada pengujian pertama golongan darah AB dan O dapat terdeteksi sebesar 100% di kedua percobaan. Pada pengujian kedua, hanya golongan darah AB yang terdeteksi sebesar 100%. Berdasarkan pengujian dan analisa yang dilakukan, disimpulkan bahwa semakin banyak *dataset* yang digunakan maka semakin tinggi nilai akurasi yang didapat. Jarak yang ideal antara objek dengan kamera ESP32-CAM untuk menangkap citra yaitu sepanjang 20 sentimeter.

DAFTAR RUJUKAN

Danukusumo, K. P. (2017). *Implementasi Deep Learning Menggunakan Convolutional Neural Network Untuk Klasifikasi Citra Candi Berbasis GPU*. Yogyakarta: Universitas Atma Jaya Yogyakarta.

Dewi, S. R. (2018). *Deep Learning Object Detection Pada Video Menggunakan Tensorflow Dan Convolutional Neural Network*. Yogyakarta: Universitas Islam Indonesia.

Harjoseputro, Y. (2018). Convolutional Neural Network (CNN) Untuk Pengklasifikasian Aksara Jawa. Yogyakarta: Universitas Atma Jaya Yogyakarta.

Hendri, M. (2018). *Perancangan Sistem Deteksi Asap dan Api Menggunakan Pemrosesan Citra*. Yogyakarta: Universitas Islam Indonesia.

Kusumaningrum, T. F. (2018). *Implementasi Convolution Neural Network (CNN) Untuk Klasifikasi Jamur Konsumsi Di Indonesia Menggunakan Keras*. Yogyakarta: Universitas Islam Indonesia.

Lering, S. T. (2013). *Penentuan Jenis Golongan Darah Manusia Berbasis Mikrokontroler AT-Mega 8535*. Yogyakarta: Universitas Sanata Dharma.

Novyantika, R. D. (2018). *Deteksi Tanda Nomor Kendaraan Bermotor Pada Media Streaming dengan Algoritma Convolutional Neural Network Menggunakan Tensorflow*. Yogyakarta: Universitas Islam Indonesia.

Nurfita, R. D., & Ariyanto, G. (2018). Implementasi Deep Learning Berbasis Tensorflow Untuk Pengenalan Sidik Jari. *Emitor: Jurnal Teknik Elektro*, *18*(1), 1–6.

Pujoseno, J. (2018). *Implementasi Deep Learning Menggunakan Convolutional Neural Network Untuk Klasifikasi Alat Tulis (Studi Kasus : Gambar Alat Tulis (Ballpoint, Penghapus, dan Penggaris))*. Yogyakarta: Universitas Islam Indonesia.

Ramadhan, F. E. (2020). *Penerapan Image Classification Dengan Pre-Trained Model Mobilenet Dalam Client-Side Machine Learning*. Jakarta: UIN Syarif Hidayatullah.

Rena, P. N. (2019). *Penerapan Metode Convolutional Neural Network pada Sistem Pendeteksi Notasi Balok*. Jakarta: UIN Syarif Hidayatullah.

Retyanto, B. D., Maghfiroh, D. I., & Hidayah, I. (2018). Rancang Bangun Prototipe Alat Ukur Golongan Darah Manusiaberbasis Arduino UNO. *SPEKTRA : Jurnal Kajian Pendidikan Sains*, *4*(2), 1–11.

Salawazo, V. M. P., Gea, D. P. J., Gea, R. F., & Azmi, F. (2019). Implementasi Metode Convolutional Neural Network (CNN) Pada Peneganalan Objek Video CCTV. *Jurnal Mantik Penusa*, *3*(1), 1–6.

Santoso, A., & Ariyanto, G. (2018). Implementasi Deep Learning Berbasis Keras Untuk Pengenalan Wajah. *Emitor: Jurnal Teknik Elektro*, *18*(01), 1–7.

Suartika, I. W., Wijaya, A. Y., & Soelaiman, R. (2016). Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101. *Jurnal Teknik ITS*, *5*(1), 1–5.