

SISTEM KLASIFIKASI CITRA DAUN UNTUK IDENTIFIKASI PENYAKIT TANAMAN SEMANGKA BERBASIS KECERDASAN BUATAN

Marsell¹, Manorang Gultom², Riyadi J. Iskandar³

Informatika, Fakultas Teknologi Informasi Universitas Widya Dharma Pontianak

¹21421466_marsell@widyadharma.ac.id, ²manorang_gultom@widyadharma.ac.id, ³riyadi@widyadharma.ac.id

Abstract

As a widely cultivated horticultural crop, watermelon plays an important role in supporting the national economy due to its high demand in international markets. However, leaf diseases that are not promptly identified often lead to reduced crop yields. To solve this problem, technology plays a crucial role in supporting the agricultural industry. The purpose of this study is to design a system for identifying watermelon leaf diseases using the Convolutional Neural Network (CNN) method. A literature review was conducted to explore relevant methods in plant disease identification. Images of watermelon leaves were collected and processed through several stages, including preprocessing, CNN model training, and performance evaluation. The model was trained on a three-class dataset with proportions of 80 percent for training, 10 percent for validation, and 10 percent for testing. Evaluation was performed using metrics such as accuracy, precision, recall, and confusion matrix to assess the model's classification performance. The training results showed a gradual improvement in performance over 50 epochs, achieving a final training accuracy of 98.75 percent and a validation accuracy of 97.98 percent. This system demonstrates good performance in recognizing new images and has the potential to serve as an effective tool for identifying diseases in watermelon leaves.

Keywords: Artificial Intelligence, Convolutional Neural Network, Plant Disease Classification, Watermelon Plant

Abstrak

Sebagai tanaman hortikultura yang banyak dibudidayakan, semangka memegang peranan penting dalam mendukung perekonomian nasional, dikarenakan semangka merupakan buah yang banyak diminati di pasar internasional. Meski demikian, serangan penyakit daun yang tidak segera dikenali kerap menimbulkan penurunan hasil panen. Untuk mengatasi masalah ini, teknologi memegang peranan penting dalam mendukung industri pertanian. Penelitian ini bertujuan merancang sistem identifikasi penyakit daun semangka menggunakan metode *Convolutional Neural Network* (CNN). Studi literatur dilakukan untuk memahami metode yang relevan dalam identifikasi penyakit tanaman. Data citra daun semangka dikumpulkan dan diproses melalui beberapa tahap, yaitu pra-pemrosesan, pelatihan model CNN, serta evaluasi performa. Model dilatih menggunakan dataset tiga kelas dengan proporsi data latih 80 persen, validasi 10 persen, dan uji 10 persen. Evaluasi dilakukan menggunakan metrik akurasi, *precision*, *recall*, dan *confusion matrix* untuk mengukur kemampuan klasifikasi model. Hasil pelatihan menunjukkan peningkatan performa secara bertahap selama 50 *epoch* dengan akurasi pelatihan mencapai 98,75 persen dan pada akurasi validasi 97,98 persen. Sistem ini menunjukkan performa yang baik dalam mengenali citra baru dan memiliki potensi sebagai alat bantu yang efektif untuk identifikasi penyakit pada daun semangka.

Kata kunci: Kecerdasan Buatan, *Convolutional Neural Network*, Klasifikasi Penyakit Tanaman, Tanaman Semangka

1. PENDAHULUAN

Semangka merupakan salah satu komoditas pertanian penting di Indonesia, baik untuk konsumsi domestik maupun ekspor. Provinsi penghasil semangka terutama seperti Jawa Timur, Sumatera Utara, dan Kalimantan Barat menunjukkan kontribusi signifikan terhadap produksi nasional. Namun, penurunan produktivitas kadang terjadi dikarenakan beberapa faktor. Contohnya terdapat pada Kabupaten Sanggau, dilansir dari (sanggaukab.bps.go.id) produksi semangka di daerah Kabupaten Sanggau sebagai salah satu daerah penghasil semangka di Indonesia menurun signifikan pada tahun 2024.

Meskipun masih belum diketahui pasti faktor apa yang menjadi penyebab menurunnya produksi semangka pada daerah tersebut, namun salah satu faktor yang dapat menjadi penyebab menurunnya produksi semangka adalah karena adanya serangan penyakit pada daun tanaman semangka, contohnya seperti penyakit bulai (*downy mildew*), dan penyakit mosaik virus (*virus mosaic*) pada daun tanaman semangka. Penyakit bulai dan virus mosaik merupakan penyakit yang umum menyerang daun tanaman semangka dan dapat membuat kegagalan panen jika tidak ditangani dengan benar.

Terbatasnya pengetahuan dan kurangnya pemahaman mengenai cara penanganan secara tepat mengidentifikasi penyakit tanaman semangka sering mengakibatkan pertumbuhan tanaman semangka kurang

maksimal, kondisi ini merupakan persoalan yang sering dialami masyarakat, tidak hanya para petani tanaman semangka tetapi untuk siapa saja yang ingin merawat tanaman semangka^[1].

Untuk mengatasi masalah ini, teknologi memegang peranan penting dalam mendukung industri pertanian. Salah satu alat yang efektif adalah kecerdasan buatan, khususnya melalui penerapan *deep learning* dengan metode *Convolution Neural Networks*, yang membantu petani mengidentifikasi penyakit yang menyerang tanaman semangka melalui daunnya. *Deep learning* memanfaatkan jaringan saraf buatan dengan beberapa lapisan dan belajar dari data yang ada, yang memungkinkan teknologi ini mengenali dan mengidentifikasi gambar yang diberikan^[2].

Perancangan sistem identifikasi penyakit tanaman semangka melalui daunnya dengan berbasis web dapat menjadi solusi yang tepat dalam mengatasi kebingungan dari para petani, apalagi para petani kadang sulit untuk mengenali penyakit yang menyerang tanaman semangka mereka. Sistem ini bertujuan memudahkan petani dan masyarakat dalam mengidentifikasi hama dan penyakit pada tanaman semangka tanpa harus menunggu atau bergantung pada pengetahuan ahli, maka dari itu sistem ini dibuat agar petani dapat mengenali penyakit yang menyerang tanaman semangka mereka dengan lebih baik dan tepat.

2. METODE PENELITIAN

2.1 Metode penelitian

Penelitian diawali dengan studi literatur dilakukan dengan mengumpulkan berbagai sumber ilmiah yang relevan untuk mendukung pemahaman terhadap topik penelitian, yaitu sistem identifikasi penyakit pada tanaman semangka melalui citra daun menggunakan metode *Convolutional Neural Network* (CNN).

Teknik perencanaan sistem dalam penelitian ini dilakukan dengan pendekatan berorientasi objek (*object-oriented*), yang diformulasikan melalui pemodelan menggunakan *Unified Modeling Language* (UML). UML digunakan untuk menggambarkan struktur dan perilaku sistem secara visual dan sistematis, meliputi berbagai diagram seperti *use case diagram*, *activity diagram*, dan *sequence diagram*^[3]. Pendekatan ini bertujuan untuk memastikan bahwa perencanaan sistem memiliki representasi yang jelas, terstruktur, serta dapat digunakan sebagai acuan dalam proses implementasi dan pengujian sistem. Berikut langkah-langkah dalam pengembangan model:

a. Pengumpulan Dataset

Dataset yang digunakan dalam penelitian ini diperoleh melalui situs Kaggle, yaitu sebuah *platform* berbagi *dataset* dan kompetisi *data science* yang umum digunakan oleh para peneliti dan pengembang kecerdasan buatan.

b. Pra-Pemrosesan Dataset

Pada tahapan ini di rencanakan akan dilakukan pengelompokan data citra ke dalam tiga kategori kelas, yaitu semangka sehat, terinfeksi bulai, dan terinfeksi virus mosaik. Setiap citra dari masing-masing kelas kemudian dibagi menjadi tiga *subset*, yaitu data *test*, data *val*, dan data *train*.

c. Perancangan Model

Setelah melewati perencanaan pra-pemrosesan, pada tahap ini dilakukan perencanaan perancangan model *deep learning* yang digunakan untuk mengklasifikasikan citra daun tanaman semangka berdasarkan jenis penyakitnya. Model yang rencananya digunakan dalam penelitian ini adalah MobileNet V1, yaitu arsitektur *Convolutional Neural Network* (CNN) yang dirancang untuk sistem dengan komputasi ringan. Model rencananya akan disesuaikan dengan kebutuhan klasifikasi terhadap tiga kelas yang ditentukan.

d. Pelatihan Model

Proses pelatihan dirancang dengan menggunakan data yang telah diaugmentasi untuk meningkatkan keragaman citra latih. Selain itu, direncanakan pula penggunaan strategi pelatihan yang mencakup pemantauan akurasi dan *loss* pada data validasi, penyimpanan model terbaik secara otomatis, penurunan *learning rate* saat diperlukan, serta penghentian dini jika performa tidak menunjukkan peningkatan. Seluruh perencanaan ini bertujuan untuk menghasilkan model yang mampu mengenali penyakit pada daun semangka dengan akurasi tinggi dan generalisasi yang baik terhadap data baru.

e. Evaluasi Model

Rencana evaluasi model MobileNet bertujuan mengukur kinerjanya dalam mengklasifikasikan citra daun semangka menggunakan data uji yang terpisah (tidak untuk pelatihan). Hal ini penting untuk menilai kemampuan generalisasi model dan mencegah overfitting. Kinerja model akan diukur menggunakan metrik standar seperti akurasi, presisi, *recall*, dan *F1-score*.

f. Implementasi

Implementasi sistem identifikasi penyakit tanaman semangka ini dilakukan dengan menerapkan model yang telah dilatih ke dalam *web*. Aplikasi ini dirancang agar dapat diakses secara mudah oleh pengguna, baik tanpa harus menginstall aplikasi khusus.

2.2 Landasan Teori

a. Artificial Intelligence

Artificial Intelligence (AI) adalah cabang ilmu komputer yang bertujuan untuk mengembangkan sistem komputer yang mampu menjalankan tugas-tugas yang biasanya memerlukan kecerdasan manusia. Tugas-tugas tersebut mencakup kemampuan berpikir, belajar, memahami, serta mengambil keputusan. AI merupakan bidang yang terus berkembang pesat karena potensinya dalam menciptakan sistem yang dapat meniru dan bahkan melampaui kemampuan kognitif manusia dalam berbagai aspek kehidupan.^{[4][5]}

b. Convolutional Neural Network (CNN).

Convolutional Neural Network (CNN) merupakan pengembangan dari *Artificial Neural Network* (ANN) konvensional atau *Multi Layer Perceptron* (MLP), yang secara khusus dirancang untuk mengolah data dua dimensi, seperti citra. CNN termasuk dalam kategori *deep learning* karena memiliki arsitektur jaringan yang dalam, terdiri dari banyak lapisan dan mampu memproses data dalam jumlah besar. Melalui lapisan-lapisan tersebut, CNN secara bertahap mengekstraksi fitur dari citra dan menghasilkan output dalam bentuk klasifikasi ke dalam kelas tertentu.^{[6][7]}

c. Deep Learning

Deep learning adalah cabang dari *machine learning* yang memanfaatkan jaringan saraf tiruan (*neural network*) dengan banyak lapisan untuk mempelajari representasi data yang kompleks secara hierarkis. Dengan struktur lapisan yang dalam, *deep learning* mampu mengenali pola-pola abstrak dari data dan digunakan dalam berbagai aplikasi seperti pengenalan gambar, pemrosesan bahasa alami, dan sistem rekomendasi.^{[8][9]}

d. Klasifikasi

klasifikasi merupakan teknik dalam *data mining* atau *machine learning* yang digunakan untuk mengelompokkan data berdasarkan label atau target tertentu. Klasifikasi berfungsi untuk menganalisis dan mengekstraksi pola atau model dari data yang ada, sehingga dapat digunakan untuk memprediksi kelas atau kategori dari data baru. Kategori yang diprediksi bersifat diskrit dan tidak terurut, menjadikan klasifikasi sebagai metode penting dalam berbagai sistem seperti pengenalan pola dan diagnosis penyakit.^{[10][11]}

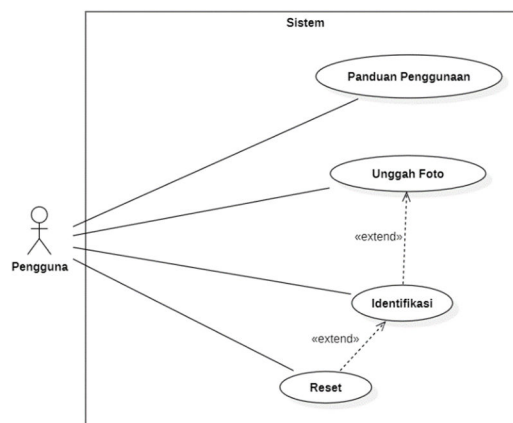
e. Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah pemodelan yang digunakan untuk menspesifikasikan, memvisualisasikan, membangun, dan mendokumentasikan artefak yaitu bagian informasi yang dihasilkan selama proses pembuatan perangkat lunak maupun sistem bisnis dan non-perangkat lunak lainnya. UML berfungsi sebagai alat bantu penting dalam pemodelan sistem secara menyeluruh, baik dari sisi teknis maupun bisnis, sehingga memudahkan komunikasi dan dokumentasi dalam pengembangan sistem.^{[12][13]}

2.3 Platform Pengembangan Sistem

Untuk menggambarkan alur perencanaan dan cara kerja sistem secara sistematis, digunakan diagram *use case* yang berfungsi untuk memodelkan interaksi antara pengguna (aktor) dan sistem. Diagram ini membantu dalam mengidentifikasi fungsi-fungsi utama yang dibutuhkan oleh pengguna serta bagaimana sistem merespons setiap aksi yang dilakukan, sehingga mempermudah proses perancangan dan pengembangan sistem secara lebih terstruktur dan terarah.

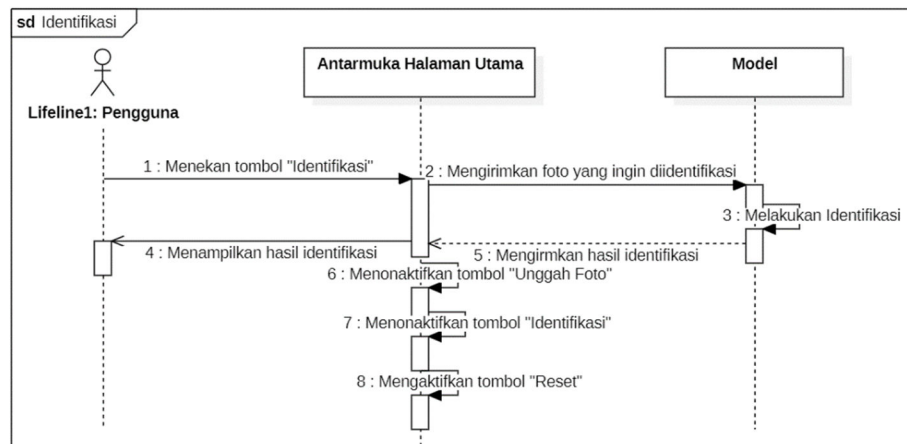
Use case diagram merupakan pemodelan yang digunakan untuk menggambarkan interaksi antara pengguna dengan sistem. Diagram ini menunjukkan fungsionalitas sistem dari perspektif pengguna dan membantu dalam memahami kebutuhan fungsional sistem yang akan dikembangkan^[14].



Gambar 1. Use Case Diagram Sistem Identifikasi

Use case diagram pada Gambar 1. memberikan gambaran interaksi antara pengguna dengan sistem identifikasi penyakit tanaman. Diagram ini berfungsi untuk memberikan gambaran umum tentang bagaimana pengguna dapat berinteraksi dengan sistem, serta bagaimana masing-masing fungsi saling berkaitan. Dalam diagram ini, terdapat empat *use case* utama yang dapat diakses oleh pengguna, yaitu “Panduan Penggunaan”, “Unggah Foto”, “Identifikasi”, dan “Reset”. Setiap *use case* dihubungkan langsung ke aktor Pengguna dengan garis lurus yang menunjukkan bahwa pengguna dapat menggunakan fitur-fitur tersebut.

Activity diagram merupakan pemodelan yang digunakan untuk memodelkan alur kerja (*workflow*) dari suatu sistem. Diagram ini menggambarkan aktivitas-aktivitas yang dilakukan secara berurutan, termasuk kondisi percabangan, pengulangan, dan aktivitas yang berjalan secara paralel. Activity diagram digunakan untuk memvisualisasikan secara garis besar alur kerja sistem^[15].



Gambar 2. Diagram Sequence Unggah Foto

Diagram Sequence pada Gambar 2 menggambarkan interaksi antara pengguna dan antarmuka halaman sistem dalam menggunakan tombol “Identifikasi”. Dimulai ketika pengguna menekan tombol "Identifikasi", kemudian sistem akan mengirim foto yang ingin diidentifikasi pengguna ke model. Setelah itu foto diterima oleh model, lalu model akan mengidentifikasinya, kemudian hasil identifikasi model dikirimkan kembali ke antarmuka halaman sistem, dan sistem akan menampilkan hasil identifikasi model ke halaman antarmuka dan menonaktifkan tombol “Unggah Foto” serta menonaktifkan tombol “Identifikasi” dan mengaktifkan tombol “Reset”.

Diagram ini menjadi penting karena menunjukkan alur identifikasi foto merupakan inti dari fungsionalitas aplikasi. Dengan menyediakan mekanisme yang sederhana dan mudah dipahami,

2.4 Rangkaian Dataset

Dataset yang digunakan dalam penelitian ini diperoleh melalui situs Kaggle, yaitu sebuah *platform* berbagi dataset dan kompetisi *data science* yang umum digunakan oleh para peneliti dan pengembang kecerdasan buatan.

Pemilihan dataset ini didasarkan pada kelengkapan label kelas, kualitas gambar yang memadai, serta relevansi langsung dengan tujuan dari penelitian, yaitu mengembangkan sistem identifikasi penyakit tanaman semangka berbasis citra dengan kategori citra daun tanaman semangka yang terbagi dalam tiga kelas, yaitu daun sehat (*healthy*) sebanyak 205 buah *dataset*, daun terinfeksi penyakit bulai (*downy mildew*) 380 buah *dataset*, dan daun yang terinfeksi virus mosaik (*mosaic virus*) 415 buah *dataset*, dengan total dari masing-masing *dataset* berjumlah 1000 buah *dataset* serta memiliki resolusi 1080×1080 pixels (setelah *resize*) dan memiliki format jpg.



Gambar 3. Contoh Dataset Daun Sehat (Healthy)

Gambar 3 menunjukkan 3 sampel *dataset* citra daun semangka dalam kondisi sehat, yaitu gambar-gambar daun yang tidak menunjukkan adanya gejala penyakit seperti bercak, perubahan warna, pelapukan, atau kerusakan

fisik lainnya. Dataset ini disusun untuk merepresentasikan kondisi normal tanaman semangka, yang menjadi acuan utama bagi model klasifikasi dalam membedakan antara daun yang sehat dan yang mengalami gangguan. Citra-citra dalam kelas ini penting karena berfungsi sebagai *baseline* atau pembanding saat model dilatih untuk mengenali pola visual pada daun yang terindikasi penyakit. Dengan menyertakan data daun sehat secara proporsional, dalam hal ini model MobileNet diharapkan dapat belajar dengan akurat mengenai tekstur, bentuk, dan warna daun semangka yang dianggap normal.



Gambar 4. Contoh Dataset Daun yang Terkena Penyakit Bulai (Downy Mildew)

Gambar 4 memperlihatkan citra daun tanaman semangka yang terinfeksi penyakit bulai, yaitu salah satu penyakit yang disebabkan oleh infeksi jamur. Penyakit ini umumnya ditandai dengan munculnya bintik-bintik berwarna kuning hingga putih pada permukaan daun, terutama pada bagian atas daun, serta sering kali disertai dengan bercak berjamur atau lapisan seperti embun pada bagian bawah daun. Ciri khas ini membuat penyakit bulai berbulu cukup mudah dikenali secara visual, khususnya pada tahap lanjut infeksi. Namun, pada tahap awal, gejala tersebut bisa tampak samar dan menyerupai perubahan warna akibat faktor lingkungan, sehingga dibutuhkan model klasifikasi yang cermat dan akurat.



Gambar 5. Contoh Dataset Daun yang Terkena Virus Mosaik (Mosaic Virus)

Gambar 5 menampilkan contoh citra daun tanaman semangka yang terinfeksi penyakit virus mosaik, yaitu penyakit yang disebabkan oleh infeksi virus dan termasuk salah satu jenis penyakit yang umum menyerang tanaman hortikultura. Gejala utama dari virus mosaik adalah munculnya pola belang atau bercak tidak merata pada daun, yang ditandai dengan kombinasi warna hijau muda dan hijau tua yang tidak seragam. Selain perubahan warna, infeksi virus mosaik juga sering mengakibatkan deformasi pada bentuk daun, seperti menggulung, keriting, atau tidak simetris, tergantung tingkat keparahan infeksi dan jenis virus yang menyerang. Ciri-ciri visual ini menjadi indikator penting dalam klasifikasi berbasis citra.

2.5 Pengembangan Model

Model klasifikasi citra penyakit tanaman dikembangkan menggunakan bahasa pemrograman Python dengan memanfaatkan pustaka TensorFlow dan Keras sebagai framework utama dalam implementasi *deep learning*. Model yang digunakan adalah MobileNet, yaitu model berbasis *Convolutional Neural Network* (CNN) yang ringan dan efisien, serta telah dipra-latih pada *dataset* ImageNet untuk memaksimalkan kemampuan ekstraksi fitur. Pada proses pelatihan, model ditambahkan beberapa lapisan tambahan berupa *GlobalAveragePooling2D*, *Batch Normalization*, *Dense*, dan *Dropout* untuk mengurangi risiko *overfitting* dan meningkatkan kemampuan generalisasi.

Data pelatihan dan validasi diproses menggunakan teknik augmentasi citra melalui ImageDataGenerator, yang mencakup transformasi seperti rotasi, pergeseran horizontal, pergeseran vertikal dan *zoom*. Seluruh data citra diskalakan ulang ke ukuran 224x224 piksel dan dinormalisasi dalam rentang [0, 1]. Untuk mengoptimalkan proses pelatihan, digunakan strategi *fine-tuning*, di mana sebagian besar lapisan awal MobileNet dibekukan dan hanya lapisan akhir hanya mengaktifkan 30 *layer* terakhir untuk dilatih ulang karena menggunakan model yang sudah pernah dilatih sebelumnya dan menggunakan fungsi *compute class weight* untuk menyeimbangkan bobot kelas.

Selama proses pelatihan, diterapkan callback seperti *EarlyStopping*, *ModelCheckpoint*, dan *ReduceLROnPlateau* untuk menghentikan pelatihan secara dini ketika performa validasi tidak mengalami peningkatan, menyimpan model terbaik berdasarkan akurasi validasi tertinggi, serta menyesuaikan laju pembelajaran jika diperlukan. Setelah pelatihan selesai, model dievaluasi menggunakan data uji dengan menghitung metrik seperti akurasi, *precision*, *recall*, *F1-score*, serta memvisualisasikan hasil melalui *confusion*

matrix. Selain itu, hasil pelatihan juga divisualisasikan dalam bentuk grafik performa akurasi, *loss*, dan laju pembelajaran terhadap jumlah *epoch* pelatihan.

3. HASIL DAN PEMBAHASAN

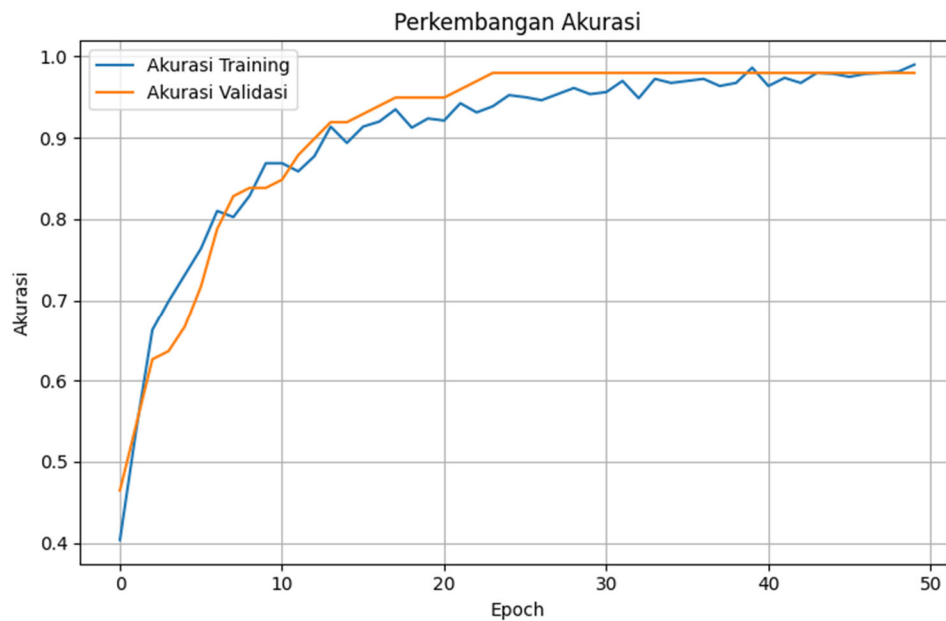
Tabel 1. Hasil Pelatihan Model

Epoch	Akurasi Training	Loss Training	Akurasi Validasi	Loss Validasi
1	39,96%	1,5674	46,46%	1,0931
2	52,13%	1,1596	54,55%	0,9943
3	66,09%	0,8389	62,63%	0,8936
4	69,80%	0,7411	63,64%	0,7858
5	73,96%	0,6441	66,67%	0,6858
6	75,90%	0,5405	71,72%	0,6084
7	79,28%	0,4872	78,79%	0,5389
8	81,24%	0,4610	82,83%	0,4848
9	82,92%	0,4155	83,84%	0,4356
10	86,46%	0,3709	83,84%	0,3918
11	85,68%	0,3553	84,85%	0,3613
12	85,90%	0,3416	87,88%	0,3283
13	87,71%	0,3251	89,90%	0,3063
14	90,50%	0,2902	91,92%	0,2855
15	90,28%	0,2984	91,92%	0,2716
16	90,91%	0,2601	92,93%	0,2528
17	91,55%	0,2534	93,94%	0,2388
18	94,02%	0,2413	94,95%	0,2280
19	91,46%	0,2739	94,95%	0,2182
20	92,71%	0,2370	94,95%	0,2036
21	92,03%	0,2368	94,95%	0,1856
22	94,08%	0,1990	95,96%	0,1718
23	92,77%	0,2012	96,97%	0,1592
24	94,85%	0,1777	97,98%	0,1550
25	94,50%	0,1787	97,98%	0,1542
26	94,35%	0,1907	97,98%	0,1481
27	94,44%	0,1985	97,98%	0,1425
28	95,80%	0,1616	97,98%	0,1397
29	95,32%	0,1624	97,98%	0,1365
30	95,07%	0,1853	97,98%	0,1339
31	96,59%	0,1482	97,98%	0,1302
32	96,75%	0,1535	97,98%	0,1230
33	95,57%	0,1774	97,98%	0,1188
34	97,74%	0,1354	97,98%	0,1148
35	96,03%	0,1631	97,98%	0,1163
36	97,67%	0,1264	97,98%	0,1151
37	98,16%	0,1256	97,98%	0,1126
38	95,67%	0,1458	97,98%	0,1072
39	97,23%	0,1229	97,98%	0,1067
40	98,26%	0,1185	97,98%	0,1059
41	96,42%	0,1566	97,98%	0,1043
42	97,95%	0,1238	97,98%	0,1020
43	97,02%	0,1243	97,98%	0,0958
44	98,48%	0,1186	97,98%	0,0918
45	98,10%	0,1006	97,98%	0,0894
46	97,27%	0,1294	97,98%	0,0876
47	98,15%	0,1166	97,98%	0,0846
48	98,68%	0,1070	97,98%	0,0822
49	98,02%	0,1055	97,98%	0,0765
50	98,75%	0,0915	97,98%	0,0748

Tabel 2. Rekap Hasil Akurasi dan Loss

Akurasi Pelatihan	Loss Pelatihan	Akurasi Validasi	Loss Validasi
89.47%	0.3014	91.91%	0.2792

merupakan rekap hasil akurasi dan *loss* pelatihan dan validasi, Secara keseluruhan, hasil ini menunjukkan sebuah model yang sudah baik dalam generalisasi. Model ini tidak hanya belajar dengan baik, tetapi juga mampu menerapkan pengetahuannya pada data baru dengan baik. Terlihat dari akurasi validasi 91,98 persen dan akurasi pelatihan yang hanya 89,47 persen, dan *loss* validasi 0,2792 yang lebih rendah daripada *loss* pelatihan 0,3015. Ini menunjukkan bahwa model walaupun sedikit menghafal data pelatihan tetapi model berhasil mempelajari pola dan fitur yang mendasari data, sehingga model bisa menerapkannya dengan baik pada data baru yang belum pernah dilihat sebelumnya.



Gambar 6. Grafik Perbandingan Akurasi Pelatihan dan Validasi per Epoch

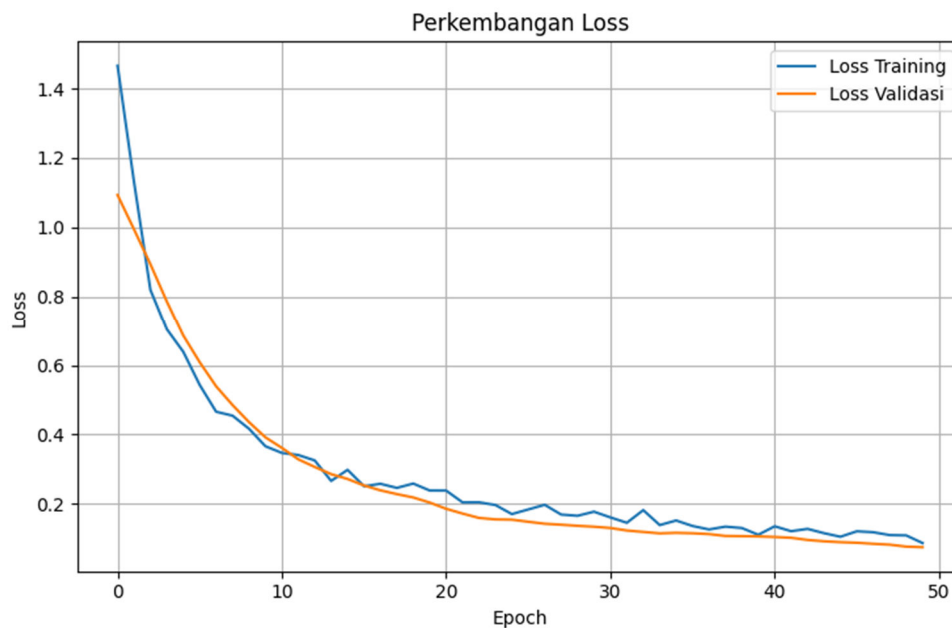
Gambar 6 merupakan grafik perkembangan akurasi pelatihan (*training*) dan validasi per *epoch* selama 50 *epoch*, yang menjelaskan garis besar perkembangan pada pelatihan dan validasi, pada grafik ini, terdapat dua kurva utama yaitu kurva biru yang menunjukkan akurasi pada data pelatihan (*training*), dan kurva jingga menunjukkan akurasi pada data validasi. Pada *epoch* 1 hingga 10, pada periode ini, kedua kurva akurasi pelatihan dan validasi mengalami peningkatan yang lumayan tinggi. Akurasi pelatihan naik dari sekitar 0,4 pada *epoch* 1 menjadi sekitar 0,86 pada *epoch* 10, sementara akurasi validasi naik dari sekitar 0,47 pada *epoch* 1 menjadi 0,85 pada *epoch* 10. Ini merupakan fase awal model belajar (*initial learning phase*). Pada mulanya, bobot dalam model diinisialisasi secara acak dan model masih menebak-nebak, sehingga performanya sangat buruk. Seiring berjalannya *epoch*, model dengan cepat mempelajari pola-pola dan fitur-fitur yang paling mendasar dan penting dari data pelatihan. Kenaikan pada akurasi validasi menunjukkan bahwa pengetahuan yang dipelajari model dapat diterapkan pada data baru.

Pada *epoch* 11 hingga 20, Akurasi pelatihan dan validasi tetap meningkat, namun dengan kecepatan yang lebih lambat. Akurasi validasi mulai menyamai akurasi pelatihan. Terlihat pada *epoch* 11, kurva akurasi validasi mulai menyalip kurva akurasi pelatihan. Ini adalah titik yang menunjukkan model mulai mempelajari data validasi. Akurasi pelatihan terus naik walaupun dengan lebih banyak fluktuasi, yaitu dari 0,86 pada *epoch* 11 hingga sekitar 0,93 pada *epoch* 20 dan pada akurasi validasi juga naik dengan kurva yang lebih baik daripada akurasi pelatihan hingga mencapai puncaknya di sekitar 0,95 pada *epoch* 20. Ini dikarenakan model memasuki fase penyempurnaan (*fine-tuning phase*). Sebagian besar pola umum yang telah dipelajari pada 10 *epoch* pertama. Sekarang, model sedang menyesuaikan bobotnya untuk mempelajari pola-pola yang lebih detail dan kompleks.

Pada *epoch* 21 hingga 30, kurva akurasi pelatihan terus mengalami fluktuasi, dari *epoch* 21 sekitar 0,92 dan mencapai sekitar 0,96 di akhir *epoch* 30. Sebaliknya, kurva akurasi validasi cenderung stagnan di rentang 0,97 dari *epoch* 24 hingga *epoch* 30. Fluktuasi yang terjadi pada kurva pelatihan dapat dikarenakan beberapa faktor diantaranya karena regulasi yang digunakan ataupun beban dari augmentasi. Namun jika dilihat pada akurasi validasi yang stagnan, ini dapat menandakan bahwa model telah mencapai titik optimal pada *dataset* saat ini.

Pada *epoch* 31 hingga 40, kurva pada akurasi pelatihan masih tetap mengalami fluktuasi, pada sekitar *epoch* 38 akurasi sedikit mengalami kenaikan hingga mencapai 0,97 yang sedikit melebihi kurva akurasi validasi, ini dapat menjadi kekhawatiran. Namun akurasi validasi yang tetap tinggi dan stabil dan gap antara validasi dan pelatihan yang kecil menunjukkan model sudah baik dan mampu mengenali data baru dengan baik.

Pada *epoch* 41 hingga 50, Model mulai memperkecil gap antara akurasi validasi dan pelatihan dapat dilihat pada sekitar *epoch* 43 kurva pada akurasi pelatihan dan validasi mendekat, Ini dapat menjadi tanda bahwa model telah optimal dan model meminimalkan kesalahan sekecil mungkin karena akurasi validasi tidak menurun dan *loss* pada validasi terus menurun, walaupun pada akhir *epoch* akurasi pada pelatihan sedikit melampaui akurasi pelatihan. Namun, jika dilihat dari kurva pada gambar di atas kurva pada akurasi validasi stagnan yang menandakan jika model telah mencapai titik optimalnya dan model sudah baik serta dapat digunakan untuk tugas klasifikasi.



Gambar 7. Grafik Perbandingan Loss Pelatihan dan Validasi per Epoch

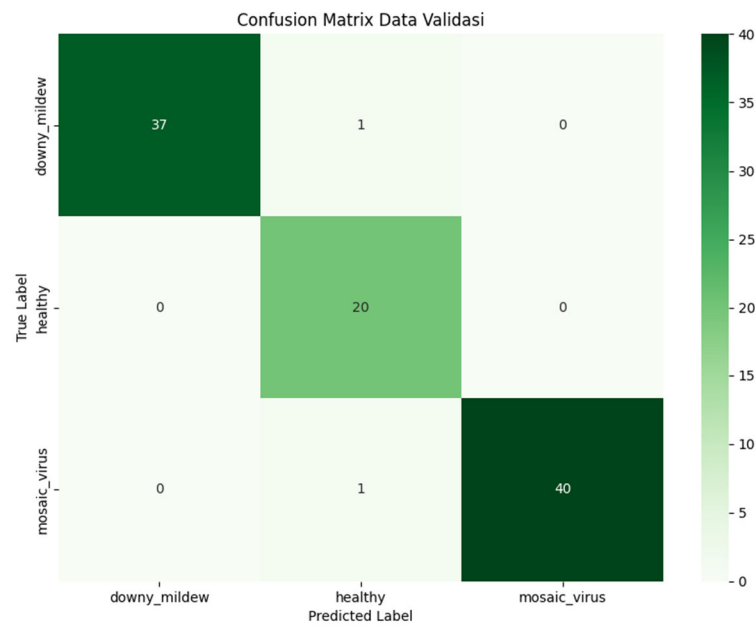
Gambar 7 merupakan grafik perkembangan *loss* pelatihan (*training*) dan validasi per *epoch* selama 50 *epoch*, yang menjelaskan garis besar perkembangan *loss* pada pelatihan dan validasi. pada grafik ini, terdapat dua kurva utama yaitu kurva biru yang menunjukkan *loss* pada data pelatihan, dan kurva jingga menunjukkan *loss* pada data validasi. Pada *epoch* 1 hingga 10, pada periode ini, model terlihat mengalami penurunan nilai *loss* yang drastis. *Loss* pelatihan turun dari sekitar 1,45 pada *epoch* 1 menjadi di bawah 0,4 pada *epoch* 10. *Loss* validasi juga turun dari sekitar 1,15 pada *epoch* 1 hingga sekitar dibawah 0,4 juga pada *epoch* 10. Ini adalah fase pembelajaran. Pada awalnya, model tidak tahu apa-apa, sehingga tingkat kesalahannya (*loss*) sangat tinggi. Seiring berjalannya *epoch*, algoritma optimisasi pada model bekerja menyesuaikan parameter untuk meminimalkan kesalahan. Penurunan *loss* yang tajam menandakan bahwa model berhasil meminimalkan kesalahan prediksi secara efektif, baik pada data pelatihan maupun data validasi

Pada *epoch* 11 hingga 20, kecepatan penurunan *loss* mulai melambat secara signifikan. Kedua kurva masih terus menurun, tetapi jauh lebih sedikit demi sedikit dibandingkan 10 *epoch* pertama. Kesenjangan antara *loss* pelatihan dan *loss* validasi tetap ada, namun relatif kecil. *Loss* pelatihan turun dari sekitar 0,38 pada *epoch* 11 ke sekitar 0,21 pada *epoch* 20 dan *loss* pada validasi turun dari 0,38 pada *epoch* 11 ke sekitar 0,19 pada *epoch* 20. Ini dikarenakan model memasuki fase penyempurnaan (*fine-tuning phase*). Pola-pola kasar sudah dipelajari. Sekarang, model melakukan penyesuaian yang lebih kecil dan lebih halus untuk menangkap detail yang lebih kompleks dari data. Karena model tidak lagi menemukan pola “mudah”, yang membuat penurunan *loss* perlahan-lahan lebih sedikit.

Pada *epoch* 21 hingga 30, kurva *loss* pada validasi cenderung lebih stabil. Nilai *loss* pada validasi sudah sangat rendah yaitu di 0,18 dan tidak ada penurunan yang signifikan dari satu *epoch* ke *epoch* berikutnya. *Loss* pelatihan masih mengalami fluktuasi dan masih lebih tinggi dari *loss* validasi, *loss* pelatihan yang mengalami fluktuasi kemungkinan dikarenakan regulasi dan *dropout* yang dilakukan pada pelatihan.

Pada *epoch* 31 hingga 40, kurva *loss* pada pelatihan masih mengalami fluktuasi dengan perbedaan yang sangat kecil antara pelatihan dan validasi. Walaupun begitu, kurva pada *loss* validasi semakin menurun dan tetap stabil. Ini mengindikasikan bahwa model semakin baik dalam memprediksi data baru.

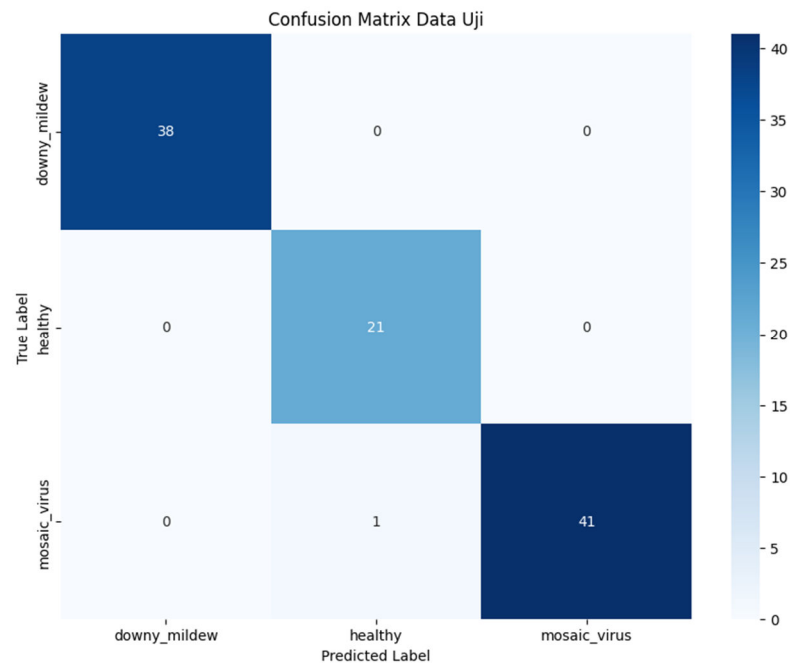
Pada *epoch* 41 hingga 50, fluktuasi pada kurva *loss* pelatihan sudah menjadi lebih sedikit dan *loss* pada validasi tetap menurun. Ini mengindikasikan model sudah matang dan semakin baik dalam memprediksi data baru, sehingga dapat digunakan untuk tugas klasifikasi.



Gambar 9. Confusion Matrix Terhadap Data Validasi

Tabel 3. Laporan Klasifikasi Hasil Terhadap Data Validasi

Kelas	Precision	Recall	F1-Score	Dataset
downy_mildew	1,00	0,97	0,99	38
healthy	0,91	1,00	0,95	20
mosaic_virus	1,00	0,98	0,99	41



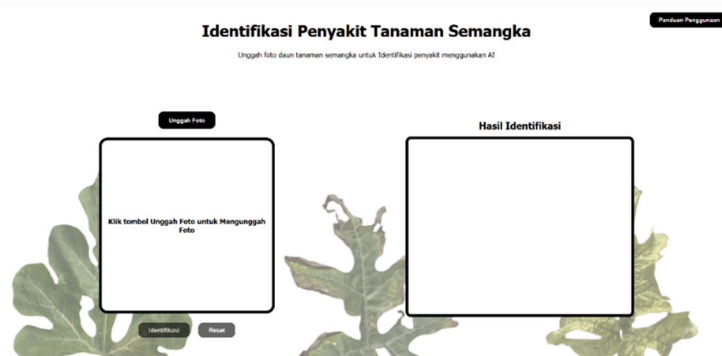
Gambar 10. Confusion Matrix Terhadap Data Uji

Tabel 4. Laporan Klasifikasi Hasil Terhadap Data Uji

Kelas	Precision	Recall	F1-Score	Dataset
downy_mildew	1,00	1,00	1,00	38
healthy	0,95	1,00	0,98	20
mosaic_virus	1,00	0,98	0,99	41

3.1 Implementasi

Implementasi sistem identifikasi citra ke dalam platform web dilakukan agar pengguna dapat dengan mudah mengaksesnya melalui browser. Model yang telah dilatih dan disimpan dalam format .keras dimuat kembali menggunakan framework Flask, yang berfungsi sebagai backend web. Digunakan juga metode threshold pada sistem, yaitu metode pengambilan keputusan dengan membandingkan nilai keluaran sistem terhadap batas tertentu, dalam sistem ini digunakan threshold 50 persen. Metode ini digunakan untuk mengurangi kesalahan identifikasi pada citra yang tidak relevan..



Gambar 11. Tampilan Halaman Sistem Identifikasi

4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan:

- Model *Convolutional Neural Network* (CNN) yang diterapkan pada penelitian ini memperlihatkan kemampuan yang baik dalam memproses dan mengidentifikasi citra daun tanaman semangka untuk keperluan identifikasi penyakit. Model arsitektur yang digunakan adalah MobileNet V1 menunjukkan performa yang baik, dengan hasil akurasi pelatihan mencapai 98,75 persen dan pada akurasi validasi 97,98 persen.
- Sistem tidak memiliki kelas anonim atau kelas tidak diketahui yang berisikan citra-citra tidak diketahui atau tidak relevan, yang menyebabkan model masih dapat mengidentifikasi citra yang tidak diketahui atau tidak relevan sebagai penyakit tanaman semangka.
- Sistem yang telah diimplementasikan ke web dan dihosting dengan lokal hosting, berhasil menyediakan antarmuka yang mudah digunakan dan dapat diakses melalui browser. Implementasi web memungkinkan untuk melakukan identifikasi penyakit tanaman secara efisien tanpa memerlukan instalasi aplikasi khusus.

5. SARAN

- Untuk meningkatkan performa sistem identifikasi penyakit tanaman semangka, disarankan untuk membuat 1 kelas baru yaitu kelas anonim atau tidak diketahui agar mencegah model untuk mengidentifikasi citra yang tidak diketahui atau tidak relevan sebagai kelas penyakit.
- Untuk meningkatkan kegunaan sistem dalam dunia nyata, disarankan untuk menambah variasi kelas penyakit lain yang umum menyerang tanaman semangka, seperti penyakit embun tepung (powdery mildew) atau penyakit layu fusarium (fusarium wilt), agar sistem dapat menangani kasus lebih luas.
- Untuk meningkatkan fungsionalitas dan kenyamanan pengguna, disarankan agar sistem dilengkapi dengan beberapa fitur tambahan. Salah satunya adalah fitur riwayat klasifikasi, yang memungkinkan pengguna melihat kembali hasil identifikasi sebelumnya dan dapat mengunduh riwayat klasifikasinya.

UCAPAN TERIMA KASIH

Peneliti mengucapkan terima kasih yang sebesar-besarnya kepada Tuhan Yang Maha Esa atas rahmat dan karunia-Nya sehingga penelitian ini dapat diselesaikan dengan baik. Ucapan terima kasih juga disampaikan kepada dosen pembimbing dan para staff Fakultas Teknologi Informasi Universitas Widya Dharma Pontianak yang telah memberikan bimbingan, arahan, dan masukan yang sangat berarti selama proses penyusunan penelitian ini. Tak lupa, penulis mengucapkan terima kasih kepada seluruh pihak yang telah membantu, baik secara langsung maupun tidak langsung, dalam proses penelitian dan penyusunan penelitian ini, termasuk keluarga, teman-teman, serta semua pihak yang turut mendukung terlaksananya penelitian ini. Penulis menyadari bahwa penelitian ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan demi perbaikan ke depannya.

DAFTAR PUSTAKA

- [1] Betrisandi. (2019). “Penerapan Case Based Reasoning Untuk Penyakit Tanaman Semangka dengan persoalan penyakit . Terbatasnya pengetahuan dan kurangnya,” *Penerapan Case Based Reason. Untuk Penyakit Tanam. Semangka*, vol. Vol.3 No.2.
- [2] Jinan, Abdul, B. Herawan Hayadi, dan Universitas Potensi Utama. (2022). “Klasifikasi Penyakit Tanaman Padi Menggunakan Metode Convolutional Neural Network Melalui Citra Daun (Multilayer Perceptron),” *J. Comput. Eng. Sci.*, vol. 1, no. 2, pp. 37–44.
- [3] Ahmad, Nazaruddin Erly Krisnanik, Frits Gerit John Rupilele, Anita Muliawati, Nur Syamsiyah, Kraugusteeliana, Bagus Dwi Cahyono, Yesi Sriyeni, Titus Kristanto, Irwanto, Guntoro. (2022). *Analisa & Perancangan Sistem Informasi Berorientasi Objek*. Penerbit Widina. Bandung.
- [4] Hatta, Heliza Rahmania, Deyidi Mokoginta, Zen Munawar, Nurul Chafid, Subhan Nooriansyah, Ade Suparman, Farid Fitriyadi, Lili Tanti, Adi Kurniawan Saputro, dan Lies Hendrawan K. (2024). *KECERDASAN BUATAN*. Cendikia Mulia Mandiri. Batam.
- [5] Manalu, Sovantro Derisjon, Adinda Naya Laka Chilfi, Ega Arif Surya Baskara, Liyanti, Indah Puspika Widarma, Celena Helga Izdiyar, Daniel Siburian, Muhammad Ferlian Adhitama, Mika Apriela Situmorang, Tamaro Simamora, Risydi Liana Balqis, Azizah, Habib Dermawan, Muhammad Alief Nurrahman, Muhammad Rifki Januarta, Monasari, Dzhinul Abdi Wicaksono, Rikhesya Anjani Gumanti, Fanny Tamba, Indah Siburian, Gustina Triayu Hana Nazwa, Tiur Pane. (2024). *AI: revolusi pembelajaran menerobos batasan melalui pemanfaatan kecerdasan buatan dalam pendidikan*. CV Brimedia Global. Bengkulu.
- [6] Batubara, Nur Arkhamia, dan Rolly Maulana Awangga. (2020). *Tutorial Object Detection Plate Number With Convolution Neural Network (CNN)*. in Computer Vision. Kreatif. Bandung.
- [7] Wahyudi. Setiawan. (2021). *Deep Learning menggunakan Convolutional Neural Network: Teori dan Aplikasi*. Media Nusa Creative (MNC Publishing). Malang.
- [8] Darnila, Eva, Zelvi Gustiana, Prayoga, Cut Amalia Saffiera, Muhammad Eka, Zulham, Cut Fadhillah. (2024). *Data Science*. Serasi Media Teknologi. Medan.
- [9] Kharisma, Lalu Puji Indra, Sitti Rachmawati Yahya, Sepriano, Rahmadya Trias Handayanto, Herlawati, I Made Agus Oka Gunawan, I Putu Susila Handika, Heliza Rahmania Hatta, Ahmad Syami. (2023). *Metode SPK Favorit di Masa Depan : Teori dan Contoh*. PT. Sonpedia Publishing Indonesia. Jambi.
- [10] Sadikin, Mujiono. (2022). *Teknik Machine Learning Untuk Menangani Permasalahan Data Bias Pada Transaksi Aplikasi Pos*. in Machine Learning. Zahira Media Publisher. Purwokerto.
- [11] Jollyta, Deny, Alyauma Hajjah, Elin Haerani, dan Muhammad Siddik. (2023). *Algoritma Klasifikasi untuk Pemula Solusi Python dan RapidMiner*. Deepublish. Yogyakarta.
- [12] Rusli, Muhammad, dan Evi Triandini. (2022). *Memodelkan Sistem Informasi Berorientasi Objek: Konsep Dasar, Prosedur, dan Implementasi*. Penerbit Andi. Yogyakarta.
- [13] Destriana, Rachmat, Syepri Maulana Husain, Nurdiana Handayani, dan Aditya Tegar Prahara Siswanto. (2022). *Diagram UML Dalam Membuat Aplikasi Android Firebase “Studi Kasus Aplikasi Bank Sampah.”* Deepublish. Yogyakarta.
- [14] Agustino, Wahyudi, Yudha Dwi Putra N, Doni Abdul Fatah. (2022). *Analisa dan Desain Sistem Informasi*. Media Nusa Creative (MNC Publishing). Malang.
- [15] Rohi, Habibi, Fahira, Zian Asti Dwiyaniti. (2023). *Prediksi Diagnosa Kehamilan Menggunakan Algoritma C4.5 dan Random Forest*. Bandung.