

**KLASIFIKASI PENYAKIT DAUN TOMAT MENGGUNAKAN
CONVOLUTIONAL NEURAL NETWORK (CNN) DENGAN MODEL
DENSENET121**

Disusun untuk Memenuhi Tugas Mata Kuliah TK 1 (Citra Digital)

Dosen Pembimbing:
Prof. Drs. Dr. Zuherman Rustam, D.E.A.



Disusun oleh:
Reizka Fathia (2206052755)

**DEPARTEMEN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS INDONESIA**

2024

DAFTAR ISI

| | |
|--|-----------|
| DAFTAR ISI..... | 1 |
| BAB I..... | 3 |
| PENDAHULUAN..... | 3 |
| 1. Pendahuluan..... | 3 |
| 1.1. Latar Belakang..... | 3 |
| 1.2. Rumusan Masalah..... | 4 |
| 1.3. Tujuan..... | 4 |
| 1.4. Manfaat..... | 4 |
| BAB II..... | 5 |
| METODOLOGI PENELITIAN..... | 5 |
| 2. Metodologi Penelitian..... | 5 |
| 2.1. Dataset..... | 5 |
| 2.2. Metode Pengolahan Citra..... | 5 |
| 2.3. Arsitektur Model..... | 6 |
| 2.4. Strategi Pelatihan..... | 6 |
| 2.5. Metode Evaluasi..... | 7 |
| BAB III..... | 8 |
| IMPLEMENTASI..... | 8 |
| 3. Implementasi..... | 8 |
| 3.1. Persiapan Data..... | 8 |
| 3.1.1. Pengunduhan dan Eksplorasi Dataset..... | 8 |
| 3.1.2. Pemeriksaan Duplikasi Data..... | 9 |
| 3.1.3. Pembagian Dataset..... | 9 |
| 3.2. Praproses Data..... | 9 |
| 3.2.1. Penyeragaman Ukuran Gambar..... | 9 |
| 3.2.2. Preprocessing DenseNet121..... | 9 |
| 3.3. Arsitektur Model..... | 10 |
| 3.3.1. DenseNet121..... | 10 |
| 3.3.2. Transfer Learning dan Modifikasi Model..... | 10 |
| 3.3.3. Ringkasan Model..... | 11 |
| 3.4. Training Model..... | 11 |
| 3.4.1. Tahap Pelatihan Awal..... | 11 |
| 3.4.2. Tahap Fine-tuning..... | 12 |
| 3.4.3. Evaluasi Model Akhir..... | 13 |
| 3.5. Pengujian Model dengan Sampel Gambar..... | 13 |
| 3.6. Analisis dan Diskusi..... | 14 |
| BAB VI..... | 16 |
| KESIMPULAN..... | 16 |

| | |
|----------------------------|-----------|
| 4. Kesimpulan..... | 16 |
| DAFTAR PUSTAKA..... | 17 |
| LAMPIRAN..... | 18 |

BAB I

PENDAHULUAN

1. Pendahuluan

1.1. Latar Belakang

Tomat merupakan salah satu tanaman budidaya yang memiliki nilai ekonomi tinggi di Indonesia. Namun, produktivitasnya sering terhambat oleh berbagai penyakit yang menyerang bagian daun tanaman. Serangan penyakit pada daun dapat menurunkan hasil panen secara signifikan, bahkan dalam beberapa kasus dapat menyebabkan kegagalan panen dalam waktu yang relatif singkat.

Beberapa jenis penyakit daun tomat yang umum terjadi di Indonesia antara lain:

1. **Early Blight**: ditandai dengan munculnya bercak coklat bulat di daun
2. **Late Blight**: bercak basah yang menyebar cepat ke seluruh daun dan batang
3. **Leaf Mold**: bercak kuning di bagian atas daun yang kemudian berubah menjadi coklat dari bagian bawah
4. **Septoria Leaf Spot**, dan berbagai jenis lainnya

Permasalahan utama yang dihadapi petani tomat adalah kesulitan dalam mengidentifikasi jenis penyakit ini dengan pengamatan kasat mata.

Ketidakmampuan mengidentifikasi penyakit dengan tepat menyebabkan:

- Penanganan yang keliru
- Penggunaan pestisida berlebihan yang berdampak pada biaya produksi dan lingkungan
- Kerugian hasil panen yang signifikan

Dampak dari permasalahan ini sangat terasa langsung bagi petani:

- Produktivitas dapat menurun drastis dari 1 ton menjadi hanya 400 kg akibat serangan penyakit
- Kerugian finansial mencapai jutaan rupiah dalam satu musim tanam
- Kualitas buah tidak memenuhi standar pasar karena kerusakan sebelum panen
- Ekonomi keluarga petani yang bergantung pada hasil pertanian ikut terdampak

1.2. Rumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah dalam penelitian ini adalah:

1. Bagaimana cara mengklasifikasikan jenis penyakit pada daun tomat secara akurat menggunakan teknologi computer vision?
2. Bagaimana kinerja model Convolutional Neural Network (CNN) dengan arsitektur DenseNet121 dalam mengklasifikasikan penyakit daun tomat?
3. Apakah model yang dikembangkan dapat menjadi alat bantu yang efektif bagi petani untuk identifikasi awal penyakit daun tomat?

1.3. Tujuan

Tujuan dari penelitian ini adalah:

1. Mengembangkan model klasifikasi penyakit daun tomat menggunakan Convolutional Neural Network (CNN) dengan arsitektur DenseNet121
2. Menganalisis kinerja model dalam mengklasifikasikan berbagai jenis penyakit daun tomat
3. Menyediakan solusi teknologi yang dapat digunakan sebagai alat bantu untuk petani dalam mengidentifikasi penyakit daun tomat secara dini

1.4. Manfaat

Manfaat yang diharapkan dari penelitian ini adalah:

1. **Manfaat Akademis:**
 - Memberikan kontribusi pada pengembangan metode klasifikasi penyakit tanaman menggunakan deep learning
 - Menjadi referensi untuk penelitian selanjutnya terkait penerapan CNN pada klasifikasi penyakit tanaman
2. **Manfaat Praktis:**
 - Membantu petani mengidentifikasi penyakit daun tomat dengan lebih cepat dan akurat
 - Mendukung penanganan penyakit yang lebih tepat sasaran
 - Mengurangi penggunaan pestisida berlebihan
 - Meningkatkan produktivitas dan kualitas hasil panen tomat

BAB II

METODOLOGI PENELITIAN

2. Metodologi Penelitian

2.1. Dataset

Penelitian ini menggunakan dataset "Tomato Leaf Disease" dari PlantVillage yang tersedia di platform Kaggle. Dataset ini terdiri dari gambar daun tomat dengan berbagai kondisi penyakit serta daun sehat. Gambar-gambar ini telah dikategorikan ke dalam beberapa kelas berdasarkan jenis penyakit yang menyerang daun tomat.

Dataset ini mencakup beberapa kategori penyakit daun tomat, antara lain:

1. Bacterial Spot
2. Early Blight
3. Late Blight
4. Leaf Mold
5. Septoria Leaf Spot
6. Spider Mites
7. Target Spot
8. Yellow Leaf Curl Virus
9. Mosaic Virus
10. Healthy (daun sehat)

Jumlah total gambar dan distribusinya di setiap kategori akan dianalisis pada tahap implementasi.

2.2. Metode Pengolahan Citra

Dalam penelitian ini, dilakukan beberapa tahap pengolahan citra, antara lain:

1. Pra-pemrosesan Data:

- Perubahan ukuran gambar menjadi 224×224 piksel (standar untuk model DenseNet121)
- Normalisasi data untuk menyesuaikan dengan format input yang dibutuhkan oleh model DenseNet121
- Pembagian dataset menjadi data latih dan data validasi dengan proporsi 80:20

2. Augmentasi Data:

- Pengolahan citra menggunakan preprocessing function khusus dari DenseNet121
- Proses ini membantu model dalam generalisasi dan mengurangi overfitting

2.3. Arsitektur Model

Penelitian ini menggunakan model Deep Learning berupa Convolutional Neural Network (CNN) dengan arsitektur DenseNet121. DenseNet (Densely Connected Convolutional Networks) adalah salah satu arsitektur CNN yang diperkenalkan oleh Huang et al. pada tahun 2017. Keunggulan arsitektur ini adalah adanya dense connections (koneksi padat) di mana setiap layer terhubung dengan semua layer berikutnya secara langsung.

Beberapa keunggulan DenseNet121:

1. **Pengurangan Vanishing Gradient:** Koneksi padat memungkinkan gradient mengalir lebih lancar selama proses backpropagation
2. **Feature Reuse:** Layer-layer dapat mengakses fitur dari semua layer sebelumnya
3. **Jumlah Parameter yang Lebih Sedikit:** Meskipun memiliki koneksi yang padat, jumlah parameter lebih sedikit karena tidak perlu "relearning" feature maps
4. **Performa Tinggi:** Terbukti memiliki performa yang tinggi dalam berbagai tugas klasifikasi citra

Berikut adalah struktur model yang digunakan:

1. **Base Model:** DenseNet121 dengan bobot pre-trained dari ImageNet
2. **Layer Tambahan:**
 - Global Average Pooling 2D untuk mereduksi dimensi feature maps
 - Dropout layer dengan rate 0.3 untuk mengurangi overfitting
 - Dense layer dengan fungsi aktivasi softmax untuk klasifikasi multi-kelas

2.4. Strategi Pelatihan

Pelatihan model dilakukan dalam dua tahap:

1. Transfer Learning (Freeze Base Model):
 - Base model (DenseNet121) di-freeze untuk mempertahankan bobot pre-trained
 - Hanya layer tambahan yang dilatih
 - Learning rate: $1e-4$
 - Optimizer: Adam
 - Loss function: Categorical Cross Entropy

- Metrics: Accuracy
 - Epochs: 10
2. Fine Tuning:
- Unfreeze sebagian layer dari base model (layer setelah indeks 300)
 - Learning rate yang lebih kecil: $1e-5$
 - Epochs tambahan: 5

Strategi ini memanfaatkan transfer learning untuk mempercepat konvergensi dan meningkatkan performa model, khususnya dalam kasus dengan jumlah data terbatas.

2.5. Metode Evaluasi

Model yang dibangun dievaluasi menggunakan beberapa metrik:

1. **Akurasi:** Persentase prediksi yang benar dari total prediksi
2. **Loss:** Mengukur seberapa jauh prediksi model dari label sebenarnya
3. **Kurva Pembelajaran:** Visualisasi performa model selama proses pelatihan, baik pada data training maupun validasi

BAB III

IMPLEMENTASI

3. Implementasi

3.1. Persiapan Data

Dataset yang digunakan dalam penelitian ini adalah PlantVillage Dataset yang dikhususkan untuk daun tomat. Dataset ini berisi gambar-gambar daun tomat dengan berbagai kondisi kesehatan, termasuk daun sehat dan daun yang terinfeksi oleh berbagai penyakit. Berikut adalah proses persiapan data:

3.1.1. Pengunduhan dan Eksplorasi Dataset

Dataset diunduh dari Kaggle menggunakan API Kaggle. Dataset terdiri dari 10 kelas dengan total 14.529 gambar, yang terbagi sebagai berikut:

| Label | Jumlah Gambar |
|---|---------------|
| Tomato___Bacterial_spot | 1702 |
| Tomato___Early_blight | 800 |
| Tomato___Late_blight | 1527 |
| Tomato___Leaf_Mold | 761 |
| Tomato___Septoria_leaf_spot | 1417 |
| Tomato___Spider_mites Two-spotted_spider_mite | 1341 |
| Tomato___Target_Spot | 1123 |
| Tomato___Tomato_Yellow_Leaf_Curl_Virus | 4286 |
| Tomato___Tomato_mosaic_virus | 299 |
| Tomato___healthy | 1273 |

Tabel 1. Dataset

Kelas-kelas ini merepresentasikan berbagai kondisi kesehatan daun tomat, mulai dari kondisi sehat hingga berbagai jenis penyakit yang umum menyerang tanaman tomat.

3.1.2. Pemeriksaan Duplikasi Data

Langkah penting dalam proses persiapan data adalah pemeriksaan data duplikat untuk memastikan kualitas dataset. Berdasarkan pemeriksaan, tidak ditemukan duplikasi dalam dataset.

3.1.3. Pembagian Dataset

Dataset dibagi menjadi data pelatihan (training) dan validasi dengan proporsi 80:20, menggunakan metode stratified sampling untuk memastikan distribusi kelas yang seimbang pada kedua subset:

- Jumlah train: 11623
- Jumlah validasi: 2906

Pembagian ini memastikan bahwa model memiliki cukup data untuk pelatihan sambil tetap menyediakan subset yang memadai untuk validasi.

3.2. Praproses Data

Praproses data merupakan tahap penting untuk mempersiapkan gambar agar dapat diproses dengan baik oleh model deep learning. Berikut adalah langkah-langkah praproses yang dilakukan:

3.2.1. Penyeragaman Ukuran Gambar

Semua gambar diseragamkan ke ukuran 224×224 piksel, yang merupakan ukuran input standar untuk banyak arsitektur CNN modern, termasuk DenseNet121:

```
image_size = (224, 224)
```

3.2.2. Preprocessing DenseNet121

Data gambar diproses menggunakan fungsi preprocessing khusus untuk DenseNet121:

```
train_datagen = ImageDataGenerator(preprocessing_function=preprocess_input)
val_datagen = ImageDataGenerator(preprocessing_function=preprocess_input)
```

Fungsi preprocess_input melakukan normalisasi gambar sesuai dengan format yang diharapkan oleh model DenseNet121, yang meliputi:

- Konversi RGB ke BGR
- Penskalaan nilai piksel
- Penerapan normalisasi mean subtraction

3.3. Arsitektur Model

Penelitian ini menggunakan arsitektur DenseNet121 sebagai model dasar dengan pendekatan transfer learning.

3.3.1. DenseNet121

DenseNet (Densely Connected Convolutional Networks) merupakan arsitektur CNN yang memiliki keunggulan dalam hal efisiensi parameter dan performa. Keunikan DenseNet terletak pada struktur Dense Block yang menghubungkan setiap layer ke semua layer berikutnya dalam blok yang sama. Fitur ini memungkinkan aliran informasi dan gradien yang lebih baik selama pelatihan.

Arsitektur DenseNet121 terdiri dari:

- Lapisan konvolusi awal
- 4 Dense Block dengan masing-masing 6, 12, 24, dan 16 lapisan
- Transition Layer di antara Dense Block
- Global Average Pooling dan fully connected layer di bagian akhir

3.3.2. Transfer Learning dan Modifikasi Model

Model DenseNet121 yang telah dilatih sebelumnya pada dataset ImageNet digunakan sebagai basis, dengan beberapa modifikasi untuk menyesuaikan dengan tugas klasifikasi daun tomat:

```
base_model = DenseNet121(weights='imagenet', include_top=False,
input_shape=(224,224,3))
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dropout(0.3)(x)
output = Dense(len(labels), activation='softmax')(x)
model = Model(inputs=base_model.input, outputs=output)
```

Modifikasi utama meliputi:

1. Menghapus lapisan fully-connected pada bagian atas model (parameter `include_top=False`)
2. Menambahkan lapisan Global Average Pooling untuk mengurangi dimensi feature map
3. Menambahkan Dropout (30%) untuk mencegah overfitting

4. Menambahkan lapisan Dense dengan aktivasi softmax untuk output 10 kelas (sesuai jumlah kondisi daun tomat)

3.3.3. Ringkasan Model

Berikut adalah ringkasan model yang digunakan:

- Total params: 7,047,754 (26.89 MB)
- Trainable params: 10,250 (40.04 KB)
- Non-trainable params: 7,037,504 (26.85 MB)

Pada tahap awal, hanya lapisan teratas yang dapat dilatih (trainable) sementara bagian dasar model (base model) dibekukan untuk mempertahankan fitur yang telah dipelajari dari ImageNet.

3.4. Training Model

Pelatihan model dilakukan dalam dua tahap: pelatihan awal dan fine-tuning.

3.4.1. Tahap Pelatihan Awal

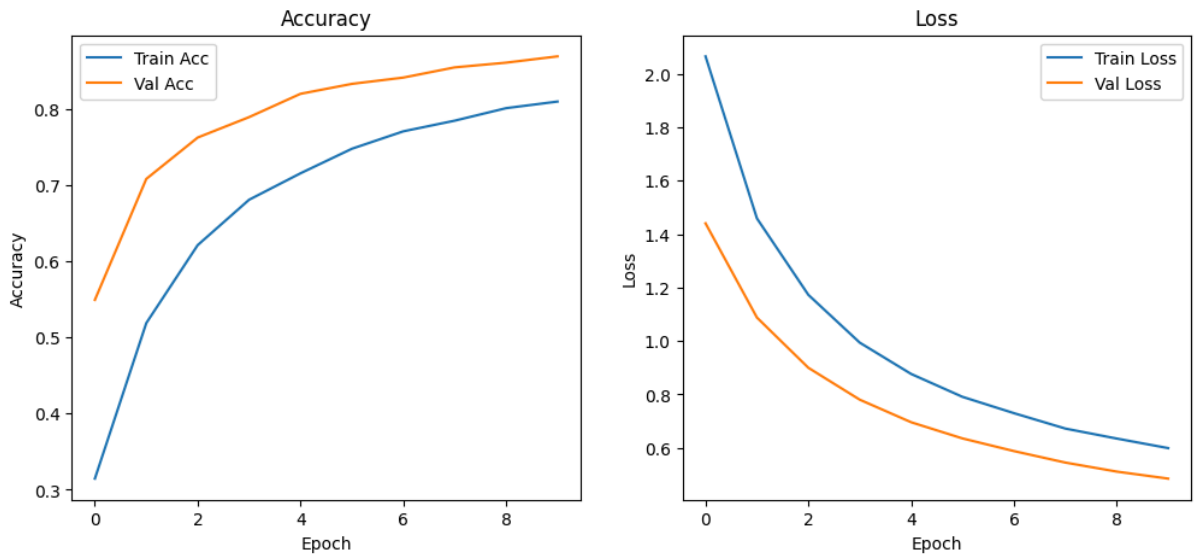
Pada tahap ini, base model dibekukan dan hanya lapisan yang baru ditambahkan yang dilatih:

```
base_model.trainable = False
model.compile(optimizer=Adam(learning_rate=1e-4),
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

Konfigurasi pelatihan awal:

- Optimizer: Adam dengan learning rate 1e-4
- Loss function: Categorical Crossentropy (sesuai untuk klasifikasi multi-kelas)
- Metrics: Accuracy
- Epochs: 10

Hasil pelatihan awal menunjukkan peningkatan akurasi yang signifikan dari epoch 1 hingga epoch 10:



Gambar 1. Hasil pelatihan awal

3.4.2. Tahap Fine-tuning

Setelah lapisan teratas terlatih dengan baik, dilakukan fine-tuning dengan membuka kunci sebagian lapisan konvolusional dari base model:

```
base_model.trainable = True
fine_tune_at = 300
for layer in base_model.layers[:fine_tune_at]:
    layer.trainable = False
model.compile(optimizer=Adam(learning_rate=1e-5),
loss='categorical_crossentropy',
metrics=['accuracy'])
```

Konfigurasi fine-tuning:

- Membuka kunci lapisan setelah indeks 300
- Penurunan learning rate menjadi 1e-5 untuk menghindari overfitting
- Epochs tambahan: 5

Hasil fine-tuning menunjukkan peningkatan performa yang signifikan:

Epoch 1/5

364/364 ————— **141s** 212ms/step - accuracy: 0.5820 -
loss: 1.4655 - val_accuracy: 0.8950 - val_loss: 0.3370

Epoch 2/5

364/364 ————— **40s** 109ms/step - accuracy: 0.8782 -
loss: 0.3816 - val_accuracy: 0.9319 - val_loss: 0.2221

Epoch 3/5

```

364/364 ————— 36s 99ms/step - accuracy: 0.9248 -
loss: 0.2533 - val_accuracy: 0.9491 - val_loss: 0.1691
Epoch 4/5
364/364 ————— 36s 99ms/step - accuracy: 0.9316 -
loss: 0.2076 - val_accuracy: 0.9549 - val_loss: 0.1406
Epoch 5/5
364/364 ————— 36s 98ms/step - accuracy: 0.9511 -
loss: 0.1575 - val_accuracy: 0.9618 - val_loss: 0.1172

```

Gambar 2. Hasil fine-tuning

3.4.3. Evaluasi Model Akhir

Setelah proses pelatihan dan fine-tuning selesai, model dievaluasi menggunakan dataset validasi:

- Validation Accuracy: 0.9618
- Validation Loss: 0.1172



Model mencapai akurasi validasi 96.18%, yang menunjukkan kemampuan model untuk mengklasifikasikan kondisi daun tomat dengan sangat baik.

3.5. Pengujian Model dengan Sampel Gambar

Untuk menguji keefektifan model dalam situasi nyata, model diuji menggunakan gambar baru yang dimasukkan ke dalam folder /kaggle/input/images. Setiap gambar diproses melalui:

- Resize ke 224x224
- Preprocessing dengan preprocess_input
- Prediksi menggunakan model
- Output berupa label kelas prediksi dan tingkat kepercayaan (confidence score)

Berikut adalah hasil pengujian:

| Gambar | Prediksi | Tingkat Kepercayaan |
|--|----------------------|---------------------|
|  late-blight.jpeg | Tomato__Late_blight | 99.44% |
|  early-blight.jpg | Tomato__Early_blight | 96.93% |

Tabel 2. Hasil pengujian

Hasil pengujian menunjukkan bahwa model dapat mengidentifikasi jenis penyakit pada daun tomat dengan tingkat kepercayaan yang sangat tinggi, sesuai dengan label yang diharapkan.

3.6. Analisis dan Diskusi

Model DenseNet121 menunjukkan performa yang baik dalam mengenali berbagai jenis penyakit daun tomat. Dengan metode transfer learning dan fine-tuning, model dapat memanfaatkan pengetahuan dari dataset umum (ImageNet) dan mengadaptasikannya untuk domain khusus (penyakit daun tomat).

Kelebihan pendekatan ini:

- **Akurasi tinggi** tanpa perlu melatih model dari awal
- **Waktu pelatihan lebih singkat**
- **Dapat diterapkan di dunia nyata**, misalnya sebagai fitur aplikasi deteksi penyakit tanaman bagi petani

Namun, keterbatasannya termasuk:

- Membutuhkan perangkat keras yang cukup (terutama GPU) untuk pelatihan
- Akurasi bisa menurun jika gambar diambil dari kondisi yang berbeda dengan dataset (misalnya pencahayaan buruk)

BAB VI

KESIMPULAN

4. Kesimpulan

Berdasarkan hasil pelatihan model CNN pretrained DenseNet121 pada dataset penyakit daun tomat yang terdiri dari 14.529 gambar dalam 10 kelas, diperoleh hasil yang sangat memuaskan dengan akurasi validasi mencapai 96,18% dan loss validasi 0,1172 setelah proses fine-tuning. Model ini menunjukkan performa yang stabil dan konsisten sejak awal pelatihan hingga akhir, dengan peningkatan bertahap dari akurasi awal sebesar 23,29% hingga 81,38% sebelum fine-tuning. Evaluasi akhir menghasilkan akurasi sebesar 96,26%, dan pengujian pada gambar uji seperti *late-blight.jpeg* dan *early-blight.jpg* menunjukkan prediksi yang sangat akurat dengan tingkat keyakinan di atas 96%. Dengan demikian, model ini terbukti efektif dalam melakukan klasifikasi otomatis penyakit daun tomat dan berpotensi untuk diimplementasikan dalam sistem deteksi penyakit tanaman berbasis citra.

DAFTAR PUSTAKA

- Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).
- Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Frontiers in plant science*, 7, 1419.
- Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, 145, 311-318.
- Chollet, F. (2017). *Deep Learning with Python*. Manning Publications Co.
- TensorFlow Documentation. (2021). Transfer learning and fine-tuning. Retrieved from https://www.tensorflow.org/tutorials/images/transfer_learning
- Kaggle. (2021). PlantVillage Tomato Leaf Dataset. Retrieved from <https://www.kaggle.com/datasets/charuchaudhry/plantvillage-tomato-leaf-dataset>.

LAMPIRAN

1. [Dataset](#)
2. [Kode Program \(Kaggle Notebook\)](#)
3. [Manual Program](#)