



Identifikasi Penyakit Daun Cabai Menggunakan Arsitektur DenseNet169

Putri Rizka Setiari¹, Barry Ceasar Oktariadi², Alda Cendekia Siregar³

Email: ¹211220061@unmuhpnk.ac.id, ²barry.ceasaro@unmuhpnk.ac.id, ³alda.siregar@unmuhpnk.ac.id

¹Program Studi Teknik Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Muhammadiyah Pontianak

²Program Studi Teknik Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Muhammadiyah Pontianak

³Program Studi Teknik Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Muhammadiyah Pontianak

Diterima: 04 Desember 2025 | Direvisi: - | Disetujui: 12 Desember 2025

©2020 Program Studi Teknik Informatika Fakultas Ilmu Komputer,
Universitas Muhammadiyah Riau, Indonesia

Abstrak

Cabai merupakan komoditas pertanian bernilai ekonomi tinggi di Indonesia, namun produksinya sering terganggu oleh serangan penyakit pada daun, seperti bercak, keriting, dan kuning. Identifikasi dini penyakit ini sangat penting untuk mencegah kerugian hasil panen. Penelitian ini bertujuan mengembangkan sistem identifikasi penyakit daun cabai secara otomatis menggunakan arsitektur *Deep Learning* DenseNet169 berbasis *website*. Metode yang digunakan meliputi pengumpulan dataset dari Roboflow.com sebanyak 3.610 citra daun cabai dengan empat kelas (bercak, keriting, kuning, dan sehat), pra-pemrosesan data, augmentasi, serta pelatihan dan evaluasi model. Hasil penelitian menunjukkan bahwa model DenseNet169 mencapai akurasi sebesar 98% dengan nilai *precision*, *recall*, dan F1-score yang konsisten pada setiap kelas. Model ini kemudian diintegrasikan ke dalam aplikasi *website* menggunakan Flask, memungkinkan pengguna mengunggah gambar daun cabai untuk mendapatkan prediksi penyakit beserta rekomendasi penanganannya. Sistem ini diharapkan dapat membantu petani dalam mendeteksi penyakit secara dini, sehingga meningkatkan efisiensi budidaya cabai dan mengurangi risiko kegagalan panen.

Kata kunci: Cabai, Penyakit Daun, DenseNet169, Deep Learning, Identifikasi Penyakit.

Identification of Chili Leaf Diseases Using DenseNet169 Architecture

Abstract

*Chili is a high-value agricultural commodity in Indonesia, but its production is often hindered by leaf diseases such as spots, curling, and yellowing. Early identification of these diseases is crucial to prevent significant yield losses. This study aims to develop an automated system for identifying chili leaf diseases using the DenseNet169 Deep Learning architecture, implemented via a web-based platform. The methodology includes data collection from Roboflow.com (3,610 images of chili leaves across four classes: spots, curling, yellowing, and healthy), data preprocessing, augmentation, model training, and evaluation. The results demonstrate that the DenseNet169 model achieves an accuracy of 98%, with consistent precision, recall, and *F1-score* values for each class. The model is integrated into a Flask-based web application, allowing users to upload images of chili leaves for disease prediction and treatment recommendations. This system is expected to assist farmers in early disease detection, thereby improving cultivation efficiency and reducing crop failure risks.*

Keywords: Chili, Leaf Disease, DenseNet169, Deep Learning, Disease Identification

1. PENDAHULUAN

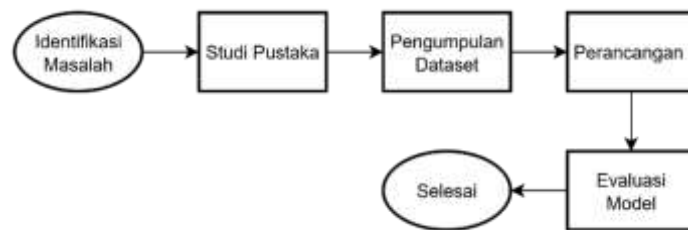
Cabai merupakan komoditas hortikultura bernilai ekonomi tinggi di Indonesia[1]. Berdasarkan Badan Pusat Statistik Kota Pontianak, produksi cabai rawit di Pontianak tahun 2023 mencapai 1.567 kuintal[2]. Oleh karena itu pemanfaatan teknologi dapat menjadi solusi untuk memudahkan mendeteksi penyakit secara otomatis menggunakan komputer secara cepat dan akurat dengan membangun sistem yang dapat mengenali penyakit. Salah satunya dengan pengolahan gambar berbasis *Deep Learning*, khususnya *Convolutional Neural Network* (CNN). CNN adalah salah satu jaringan *Deep Learning* yang paling

populer digunakan. Keuntungan utama dari CNN dibandingkan dengan pendahulunya adalah secara otomatis mendeteksi fitur-fitur penting tanpa pengawasan manusia[3]. DenseNet169 adalah salah satu arsitektur CNN dan perpanjangan dari arsitektur DenseNet, model ini memiliki 169 lapisan dan pilihan populer untuk aplikasi klasifikasi *deep learning* karena strukturnya yang berlapis, dibandingkan dengan arsitektur DenseNet lainnya ia memiliki jumlah parameter yang dapat dilatih secara signifikan lebih kecil dan memiliki kemampuan sangat andal dalam menghindari masalah gradien yang menghilang, memiliki taktik propagasi fitur yang efisien, mengurangi jumlah parameter yang dapat dilatih dan mempromosikan penggunaan kembali fitur[4]. Beberapa penelitian mengenai identifikasi penyakit tanaman telah dilakukan. Penelitian *A Novel Deep Learning Design of Plant Disease Recognition and Detection using VGG19, ResNet50, and DenseNet169* mengembangkan sistem deteksi penyakit tanaman menggunakan VGG19, ResNet50, dan DenseNet169 dengan menerapkan algoritma CLAHE (*Contrast Limited Adaptive Histogram Equalization*) menunjukkan model DenseNet lebih akurat dengan akurasi tertinggi 99.67 dan *loss* 01.12 terendah diikuti oleh ResNet50 dan VGG19[5]. Kemudian topik penelitian *Automated Paddy Leaf Disease Identification using Visual Leaf Images based on Nine Pre-trained Models Approach* mengaplikasikan 9 model salah satunya DenseNet169, model dilatih menggunakan TensorFlow dengan *optimizer* Adam, *batch size* 32, DenseNet169 mencapai akurasi sebesar 97.3%[6]. Selanjutnya penelitian *Research on Plant Disease Identification Based on CNN* membandingkan model ResNet50, DenseNet169, dan EfficientNet dengan proses pelatihan menggunakan *optimizer* Adam *learning rate* 0.001, *batch size* 16 dan 15 *epoch*, DenseNet169 menghasilkan akurasi 99% dengan nilai *loss* 0.082[7].

Meskipun DenseNet169 terbukti unggul, penelitian khusus untuk identifikasi penyakit daun cabai masih terbatas. Sebagian besar penelitian hanya fokus pengembangan model tanpa mempertimbangkan kemudahan implementasi bagi pengguna, terutama petani. Penelitian ini bertujuan mengembangkan sistem identifikasi penyakit daun cabai menggunakan arsitektur DenseNet169 yang diimplementasikan dalam bentuk *website* untuk memudahkan akses petani melalui perangkat *mobile* maupun *desktop* dalam mendeteksi penyakit daun cabai secara dini.

2. METODE PENELITIAN

Metode penelitian ini ditunjukkan pada diagram alir yang digunakan pada Gambar 1.



Gambar 2. 1 Alur Penelitian

2.1. Identifikasi Masalah

Identifikasi masalah dilakukan untuk menganalisis dan merumuskan permasalahan yang terkait dengan deteksi penyakit daun cabai secara manual, kebutuhan akan sistem identifikasi otomatis yang akurat dan cepat, serta evaluasi efektivitas penerapan arsitektur DenseNet169 dalam klasifikasi penyakit daun cabai.

2.2. Studi Pustaka

Studi pustaka merupakan menelusuri data dari berbagai referensi dan temuan penelitian sebelumnya yang mendukung sumber penelitian untuk memperoleh landasan teori yang akan dibahas khususnya literatur yang berkaitan dengan pengolahan citra, penyakit daun cabai, serta arsitektur DenseNet169 dalam CNN.

2.3. Pengumpulan Dataset

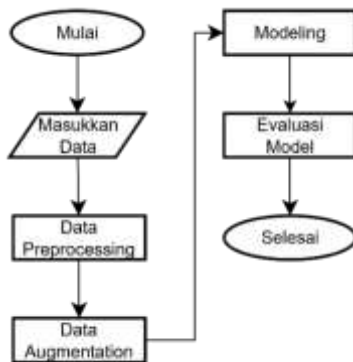
Pada tahap ini data yang dibutuhkan adalah citra daun cabai yang terkena penyakit yang diperoleh dari website [Roboflow.com](https://www.kaggle.com/datasets/roboflow/paddy-leaf-disease-detection). Citra terdiri dari 4 jenis penyakit yaitu bercak, keriting, kuning, dan sehat dengan total data yang diperoleh 3611 data citra dalam format *Joint Photographic Group* (JPG) yang terbagi menjadi dua bagian yaitu data *train* dan *valid*. Dari total 3611 citra, 3328 digunakan sebagai set pelatihan, 283 citra sebagai set validasi. Data validasi berfungsi untuk mengawasi efektivitas model selama proses pelatihan dan menghindari *overfitting*.

2.4. Perancangan

Tahap perancangan terbagi menjadi dua bagian utama, yaitu perancangan model dan perancangan sistem.

2.4.1 Perancangan Model

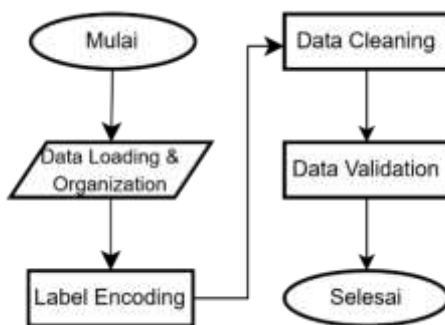
Pada tahap perancangan model, dilakukan perancangan arsitektur model untuk mengklasifikasikan penyakit pada daun cabai dengan menggabungkan seluruh proses mulai dari pemasukan data hingga pengevaluasian model. Tahapan perancangan model meliputi beberapa proses utama yaitu *data preprocessing*, *data augmentation*, *modeling*, dan evaluasi model.



Gambar 2. 2 Diagram Alur Proses Perancangan Model

a) Data Preprocessing

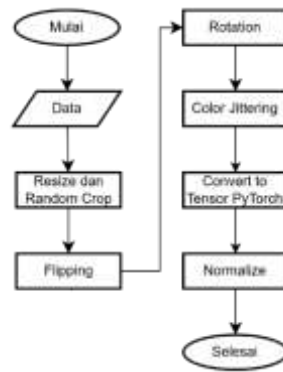
Data Preprocessing bertujuan untuk mengubah data mentah menjadi data yang berkualitas sehingga data layak untuk diolah pada tahapan selanjutnya[8]. Tahap *preprocessing* data dalam penelitian ini meliputi empat langkah utama yang ditunjukkan pada Gambar 2.3. Pertama, *data loading & organization* dilakukan untuk mengatur bagaimana data dimuat dan diorganisir dari berbagai sumber penyimpanan agar dapat diakses secara sistematis. Kedua, *label encoding* diterapkan untuk mengubah label kategorikal berupa teks menjadi format numerik yang dapat diproses oleh model *machine learning*. Ketiga, *data cleaning* dilaksanakan untuk menangani data yang rusak, *corrupt*, atau tidak sesuai dengan format yang diharapkan sehingga kualitas dataset terjaga. Keempat, *data validation* diimplementasikan untuk memverifikasi bahwa semua data telah memenuhi kriteria yang diperlukan sebelum dimasukkan ke dalam dataset akhir yang akan digunakan untuk pelatihan model.



Gambar 2. 3 Diagram Proses Preprocessing Data

b) Data Augmentation

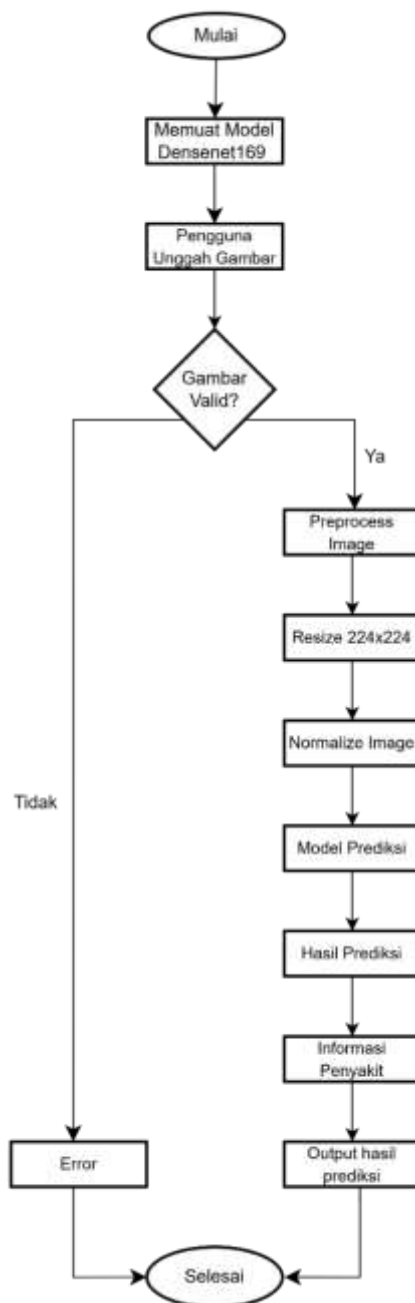
Data augmentation bertujuan untuk meningkatkan volume, kualitas, dan keragaman data pelatihan[9]. Tahap *data augmentation* dalam penelitian ini meliputi enam langkah transformasi yang diterapkan secara berurutan untuk meningkatkan variasi dataset pelatihan dapat ditunjukkan pada Gambar 2.4. Pertama, dilakukan *resize* untuk mengubah dimensi gambar menjadi 256×256 piksel agar sesuai dengan kebutuhan model, kemudian diikuti *random crop* yang memotong gambar secara acak menjadi 224×224 piksel pada setiap iterasi training. Kedua, *flipping* diterapkan untuk membalik gambar secara horizontal dengan probabilitas 50% dan vertikal dengan probabilitas 30% guna menciptakan variasi orientasi. Ketiga, *random rotation* digunakan untuk merotasi gambar secara acak hingga 15 derajat searah atau berlawanan arah jarum jam. Keempat, *color jittering* diterapkan untuk mengubah *brightness*, *contrast*, *saturation*, dan *hue* secara acak sehingga model menjadi lebih robust terhadap perubahan kondisi pencahayaan. Kelima, gambar dikonversi ke format tensor PyTorch untuk kompatibilitas dengan framework yang digunakan. Terakhir, dilakukan normalisasi menggunakan nilai mean dan *standard deviation* ImageNet untuk menstandarisasi distribusi pixel gambar sebelum diproses oleh model.



Gambar 2. 4 Diagram Proses Data Augmentation

2.4.2 Perancangan Sistem

Pada tahap perancangan sistem, dilakukan perancangan sistem aplikasi web dengan memanfaatkan *framework* Flask yang berperan sebagai antarmuka untuk memprediksi penyakit pada daun cabai, dimulai dari proses *upload* gambar hingga menghasilkan *output* berupa klasifikasi penyakit serta informasi *treatment* sesuai pada gambar 2.5. Proses deteksi penyakit daun cabai dalam sistem ini dimulai dengan memuat model DenseNet169 yang telah dimodifikasi dengan 4 kelas *output* dari *file* 'modified_densenet169_model.pth' beserta *dictionary* informasi penyakit, kemudian pengguna mengunggah gambar daun cabai melalui *web interface* yang akan melalui tahap validasi untuk memeriksa keberadaan *file*, format, dan validitas gambar. Jika validasi berhasil, gambar disimpan dengan nama unik menggunakan UUID dan dikonversi ke format RGB, selanjutnya diresize ke dimensi 224×224 piksel sesuai *requirement* model DenseNet169. Gambar kemudian dinormalisasi menggunakan nilai mean=[0.485, 0.456, 0.406] dan std=[0.229, 0.224, 0.225] standar ImageNet, dikonversi ke tensor PyTorch dengan penambahan *batch dimension* sebelum dimasukkan ke model untuk proses inferensi. *Output* model dikonversi menggunakan softmax untuk mendapatkan probabilitas setiap kelas, sistem menentukan kelas dengan probabilitas tertinggi dan menghitung confidence score, kemudian mengambil informasi detail penyakit dari *dictionary* DISEASE_INFO yang mencakup nama, deskripsi, gejala, dan rekomendasi penanganan. Akhirnya, sistem menghapus *file* sementara dan mengembalikan respon JSON berisi status sukses, hasil prediksi dengan *confidence score*, probabilitas semua kelas, dan informasi lengkap penyakit yang terdeteksi kepada *user* melalui *web interface*, atau mengembalikan pesan *error* dengan status kode yang sesuai jika validasi gagal.



Gambar 2. 5 Diagram Alur Data Augmentation

2.4.3 Evaluasi Model

Evaluasi model pada penelitian ini menggunakan *confusion matrix* kemudian dihitung *performance metric* mencakup *precision*, *recall*, *accuracy*, dan *F1-Score* yang dapat dihitung dengan rumus di bawah ini.

a) *Precision*

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

b) *Recall*

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

c) Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

d) F1-Score

$$F1_{score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

Confusion matrices mewakili hitungan dari nilai prediksi dan aktual. Output "TN" adalah singkatan dari *True Negative* yang menunjukkan jumlah contoh negatif yang diklasifikasikan secara akurat, "TP" adalah singkatan dari *True Positive* yang menunjukkan jumlah contoh positif yang diklasifikasikan secara akurat, "FP" menunjukkan nilai *False Positive*, yaitu jumlah contoh negatif aktual yang diklasifikasikan sebagai positif dan "FN" berarti nilai *False Negative* yang merupakan jumlah contoh positif aktual yang diklasifikasikan sebagai negatif [10].

3. HASIL DAN PEMBAHASAN

Bagian ini berisi tentang hasil sistem identifikasi penyakit daun cabai menggunakan arsitektur DenseNet169.

3.1. Hasil Evaluasi Model

Evaluasi model merupakan tahap penting untuk memastikan bahwa model dapat memberikan prediksi yang akurat dan *reliable*. Berikut kode program yang akan digunakan:

```
print(classification_report(all_labels, all_preds, target_names=class_names))
```

Kode program di atas untuk menampilkan *precision*, *recall*, *F1-score* untuk setiap kelas. Berikut hasil evaluasi model menggunakan *confusion matrix* ditunjukkan pada Gambar 3.1.

Classification Report:				
	precision	recall	f1-score	support
bercak daun	0.97	0.97	0.97	78
keriting	1.00	0.98	0.99	64
sehat	0.96	0.97	0.97	74
virus kuning	1.00	1.00	1.00	64
accuracy			0.98	280
macro avg	0.98	0.98	0.98	280
weighted avg	0.98	0.98	0.98	280

Gambar 3.1 Hasil Evaluasi Model

Kode program di atas merupakan hasil evaluasi model dengan *accuracy f1-score* 98%, *precision* 98%, dan *recall* 98%.

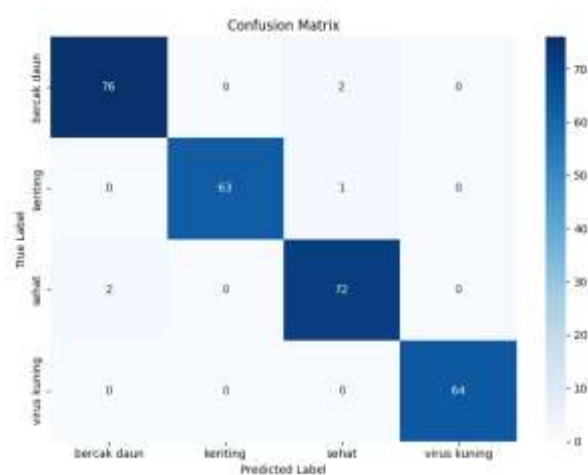
```
cm = confusion_matrix(all_labels, all_preds)

plt.figure(figsize=(8, 6))

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',

            xticklabels=class_names, yticklabels=class_names)
```

Kode di atas berfungsi visualisasi *confusion matrix* untuk melihat kesalahan klasifikasi. Gambar 3.2 menunjukkan hasil *confusion matrix* yang diperoleh:



Gambar 3. 2 Hasil Confusion Matrix

4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan mengenai identifikasi penyakit daun cabai menggunakan arsitektur DenseNet169, dapat disimpulkan bahwa model ini menunjukkan kinerja yang sangat baik dengan akurasi mencapai 98% dan nilai precision, recall, serta F1-score untuk setiap kelas penyakit (bercak, keriting, kuning, dan sehat) berada di atas 96%, menunjukkan kemampuan model dalam mengenali pola penyakit dengan sangat baik. Model DenseNet169 berhasil diintegrasikan ke dalam aplikasi berbasis web menggunakan Flask yang memungkinkan pengguna mengunggah gambar daun cabai dan mendapatkan hasil identifikasi penyakit beserta rekomendasi penanganan secara *real-time*, sehingga dapat membantu petani atau pelaku budidaya cabai dalam mendeteksi penyakit secara dini dan mengambil tindakan pencegahan yang tepat untuk mengurangi risiko kegagalan panen. Penggunaan DenseNet169 dengan teknik *transfer learning* terbukti efektif dalam menangani dataset yang relatif kecil namun menghasilkan akurasi yang tinggi, hal ini didukung oleh kemampuan DenseNet169 dalam mengekstrak fitur-fitur penting dari gambar daun cabai yang menjadikannya sebagai solusi yang tepat untuk implementasi sistem deteksi penyakit tanaman berbasis *computer vision*.

DAFTAR PUSTAKA

- [1] A. Astining, R. Herawaty, and B. Bangun, “KARAKTERISTIK PETANI DAN KELAYAKAN USAHATANI CABAI BESAR (*Capsicum Annuum* L) DAN CABAI RAWIT (*Capsicum Frutescens* L) DI SUMATERA UTARA,” vol. 5, no. 1, 2020.
- [2] “Produksi Tanaman Sayuran dan Buah–Buahan Semusim Menurut Jenis Tanaman di Kota Pontianak, 2023,” Badan Pusat Statistik Kota Pontianak.
- [3] L. Alzubaidi *et al.*, “Review of deep learning: concepts, CNN architectures, challenges, applications, future directions,” *J Big Data*, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40537-021-00444-8.
- [4] Aishwarya M.P and A. Padmanabha Reddy, “Dataset of groundnut plant leaf images for classification and detection,” *Data Brief*, 2023, doi: 10.1016/j.dib.2023.109185.
- [5] V. Goyal and V. Sejwar, “A Novel Deep Learning Design of Plant Disease Recognition and Detection using VGG19, ResNet50, and DenseNet169.”
- [6] A. Petchiammal and D. Murugan, “Automated Paddy Leaf Disease Identification using Visual Leaf Images based on Nine Pre-trained Models Approach,” in *Procedia Computer Science*, Elsevier B.V., 2025, pp. 118–126. doi: 10.1016/j.procs.2024.12.013.
- [7] X. Sun, G. Li, P. Qu, X. Xie, X. Pan, and W. Zhang, “Research on plant disease identification based on CNN,” *Cognitive Robotics*, vol. 2, pp. 155–163, Jan. 2022, doi: 10.1016/j.cogr.2022.07.001.
- [8] F. Alghifari and D. Juardi, “Fauzan Alghifari Penerapan Data Mining Pada Penerapan Data Mining Pada Penjualan Makanan Dan Minuman Menggunakan Metode Algoritma Naïve Bayes.”
- [9] A. Mumuni and F. Mumuni, “Data augmentation: A comprehensive survey of modern approaches,” Dec. 01, 2022, *Elsevier B.V.* doi: 10.1016/j.array.2022.100258.
- [10] A. Kulkarni, D. Chong, and F. A. Batarseh, “Foundations of data imbalance and solutions for a data democracy,” in *Data Democracy: At the Nexus of Artificial Intelligence, Software Development, and Knowledge Engineering*, Elsevier, 2020, pp. 83–106. doi: 10.1016/B978-0-12-818366-3.00005-8.