

**SKRIPSI**

**KLASIFIKASI TANAMAN OBAT MENGGUNAKAN  
MULTICLASS SUPPORT VECTOR MACHINE BERBASIS  
ANDROID**

**Disusun dan diajukan oleh:**

**RIEKA ZALZABILLAH PUTRI  
D121 17 1001**



**PROGRAM STUDI SARJANA TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS HASANUDDIN  
GOWA  
2023**

## LEMBAR PENGESAHAN SKRIPSI

### KLASIFIKASI TANAMAN OBAT MENGGUNAKAN *MULTICLASS SUPPORT VECTOR MACHINE* BERBASIS ANDROID

Disusun dan diajukan oleh

**Rieka Zalzabillah Putri**  
**D121 17 1001**

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana Program Studi Teknik Informatika Fakultas Teknik Universitas Hasanuddin Pada tanggal 10 Maret 2023 dan dinyatakan telah memenuhi syarat kelulusan

Menyetujui,

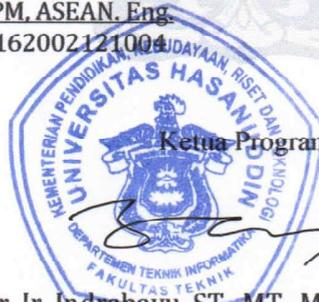
Pembimbing Utama,

Pembimbing Pendamping,



Prof. Dr. Ir. Indrabayu, ST., MT.,  
M.Bus.Sys., IPM, ASEAN. Eng.  
NIP 197507162002121004

A. Ais Prayogi Alimuddin, S.T., M.Eng  
NIP 198305102014041001



Ketua Program Studi,

Prof. Dr. Ir. Indrabayu, ST., MT., M.Bus.Sys., IPM, ASEAN. Eng.  
NIP 197507162002121004

## PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini ;

Nama : Rieka Zalzabillah Putri  
NIM : D121171001  
Program Studi : Teknik Informatika  
Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

Klasifikasi Tanaman Obat menggunakan *Multiclass Support Vector Machine*  
Berbasis Android

Adalah karya tulisan saya sendiri dan bukan merupakan pengambilan alihan tulisan orang lain dan bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Semua informasi yang ditulis dalam skripsi yang berasal dari penulis lain telah diberi penghargaan, yakni dengan mengutip sumber dan tahun penerbitannya. Oleh karena itu semua tulisan dalam skripsi ini sepenuhnya menjadi tanggung jawab penulis. Apabila ada pihak manapun yang merasa ada kesamaan judul dan atau hasil temuan dalam skripsi ini, maka penulis siap untuk diklarifikasi dan mempertanggungjawabkan segala resiko.

Segala data dan informasi yang diperoleh selama proses pembuatan skripsi, yang akan dipublikasi oleh Penulis di masa depan harus mendapat persetujuan dari Dosen Pembimbing.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan isi skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Gowa, 10 Maret 2023

Yang Menyatakan,



Rieka Zalzabillah Putri

## ABSTRAK

**RIEKA ZALZABILLAH PUTRI.** *Klasifikasi Tanaman Obat menggunakan Multiclass Support Vector Machine Berbasis Android* (dibimbing oleh Prof. Dr. Indrabayu, ST., MT., M.Bus.Sys., IPM., ASEAN Eng. dan A. Ais Prayogi Alimuddin, S.T., M.Eng.)

Pemanfaatan tanaman sebagai obat sejak dulu diminati oleh masyarakat desa, hal itu ditandai dengan banyaknya tempat pengobatan tradisional serta banyak beredar produk obat tradisional di tengah-tengah masyarakat, yang biasa disebut herbal. Karena banyaknya jumlah dan jenis tanaman obat, sehingga pengenalan otomatis tanaman obat akan membantu masyarakat untuk meningkatkan pengetahuan tentang tanaman obat. Oleh karena itu, tujuan dari penelitian ini adalah merancang sistem klasifikasi tanaman obat menggunakan *Multiclass Support Vector Machine* berbasis android. Penelitian ini menggunakan 10 jenis tanaman obat yang sering dijumpai dalam kehidupan masyarakat, dengan data *input* berupa citra daun tanaman obat yang diambil langsung dilapangan dengan kamera *smartphone*. Sebanyak 1800 data dibagi menjadi 1440 untuk proses latih dan 360 untuk proses uji. Ekstraksi fitur yang digunakan adalah fitur bentuk, fitur warna, dan fitur tekstur. Adapun hasil dari penelitian ini adalah klasifikasi tanaman obat dengan menggunakan metode *multiclass* SVM dengan akurasi yang baik yakni 97%. Parameter yang digunakan yakni kernel RBF, *Cost* (C) = 1000 dan *gamma* ( $\gamma$ ) = 0.01 yang diperoleh dari *Grid Search*, dan metode multi kelas OAA (*One Against All*). Kemudian model yang dihasilkan diimplementasi kedalam sistem berbasis android dengan hasil akurasi yang didapatkan sebesar 92%. Namun agar sistem dapat memperbaiki performa dalam mengklasifikasi tanaman obat dan mengklasifikasi yang bukan tanaman obat pada sistem ditetapkan threshold 55%, sehingga didapatkan akurasi sistem yakni 94%.

Kata Kunci: tanaman obat, ekstraksi fitur, svm, multikelas, android

## ABSTRACT

**RIEKA ZALZABILLAH PUTRI.** *Classification of Medicinal Plants using Multiclass Support Vector Machine based on Android.* (supervised by Prof. Dr. Indrabayu, ST., MT., M.Bus.Sys., IPM., ASEAN Eng. and A. Ais Prayogi Alimuddin, S.T., M.Eng.)

*The use of plants as medicine has been of interest to rural communities since ancient times, as evidenced by the numerous traditional healing centers and the widespread use of herbal medicines among the people. Due to the large number and variety of medicinal plants, automatic plant recognition systems can help to increase public knowledge about them. Therefore, the aim of this research was to design a plant classification system using Multiclass Support Vector Machine based on Android. This study utilized 10 commonly found medicinal plants in society, with input data in the form of images of plant leaves captured directly in the field using a smartphone camera. A total of 1800 data were divided into 1440 for training and 360 for testing. The feature extraction used was shape, color, and texture. The result of this research was the classification of medicinal plants using the multiclass SVM method with a good accuracy of 97%. The parameters used were the RBF kernel, Cost (C) = 1000, and gamma ( $\gamma$ ) = 0.01 obtained from Grid Search, and the One Against All (OAA) multiclass method. The resulting model was then implemented in an Android-based system with an accuracy of 92%. To improve the system's performance in classifying medicinal plants and non-medicinal plants, a threshold of 55% was set, resulting in a system accuracy of 94%.*

*Keywords: herbal plant, feature extraction, svm, multiclass, android*

## DAFTAR ISI

LEMBAR PENGESAHAN SKRIPSI.....	i
PERNYATAAN KEASLIAN.....	ii
ABSTRAK .....	iii
ABSTRACT .....	iv
DAFTAR ISI.....	v
DAFTAR GAMBAR .....	vii
DAFTAR TABEL.....	ix
DAFTAR LAMPIRAN.....	x
DAFTAR SINGKATAN DAN ARTI SIMBOL .....	xi
KATA PENGANTAR .....	xii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan Penelitian.....	3
1.4 Manfaat Penelitian.....	3
1.5 Ruang Lingkup .....	4
1.6 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA.....	6
2.1 Tanaman Obat.....	6
2.2 Daun.....	6
2.3 Visi Komputer .....	8
2.4 Ekstraksi Fitur.....	9
2.4.1 Fitur Bentuk.....	9
2.4.2 Fitur Warna.....	11
2.4.3 Fitur Tekstur .....	12
2.5 Klasifikasi dengan <i>Support Vector Machine</i> .....	15
2.5.1 <i>Hard – Margin SVM / Linear SVM</i> .....	16
2.5.2 <i>Soft Margin SVM</i> .....	17
2.5.3 Kernel SVM.....	18
2.6 Multiclass Support Vector Machine .....	19
2.6.1 <i>One-Agains-All</i> .....	20
2.6.2 <i>One-Agains-One</i> .....	20
2.7 <i>Confusion Matrix</i> .....	21
2.8 <i>GrabCut</i> .....	23
2.9 <i>Pickle</i> .....	24
2.10 <i>FastAPI</i> .....	25
2.11 Android.....	25
2.12 Penelitian Terkait.....	26
BAB III METODE PENELITIAN.....	29
3.1 Tahapan Penelitian .....	29
3.2 Waktu dan Lokasi Penelitian.....	30
3.3 Instrumen Penelitian .....	31
3.4 Teknik Pengambilan Data .....	31
3.5 Perancangan Implementasi Sistem .....	32
3.5.1 Input Citra.....	33

3.5.2	<i>Data Splitting</i> .....	36
3.5.3	<i>Preprocessing</i> .....	36
3.5.4	Ekstraksi Fitur.....	42
3.5.5	Seleksi Fitur.....	48
3.5.6	Klasifikasi dengan Algoritma <i>Multiclass SVM</i> .....	50
3.5.7	Evaluasi Model.....	52
3.5.8	Integrasi Model kedalam Sistem Berbasis Android .....	53
3.6	Analisis Kerja Sistem .....	55
BAB IV HASIL DAN PEMBAHASAN .....		57
4.1	Hasil.....	57
4.1.1	Klasifikasi Algoritma Multiclass Support Vector Machine .....	57
4.1.2	Sistem Android.....	59
4.1.3	Pengujian Model <i>Multiclass Support Vector Machine</i> .....	65
4.1.4	Pengujian Model pada Sistem Aplikasi Android .....	67
4.2	Pembahasan .....	72
BAB V KESIMPULAN & SARAN .....		85
5.1	Kesimpulan.....	85
5.2	Saran .....	86
DAFTAR PUSTAKA .....		87
LAMPIRAN .....		91
Lampiran 1. Beberapa Contoh Dataset Primer.....		91
Lampiran 2. Source Code Preprocessing .....		101
Lampiran 3. Source Code Ekstraksi Fitur .....		102
Lampiran 4. Source Code Klasifikasi M-SVM.....		105
Lampiran 5. Source Code Android.....		107
Lampiran 6. Contoh Dataset Hasil Ekstraksi Fitur pada Kelas 1 .....		110
Lampiran 7. Contoh Dataset Hasil Seleksi Fitur pada 10 Kelas .....		111
Lampiran 8. Lembar Perbaikan Skripsi.....		119

## DAFTAR GAMBAR

Gambar 1 Tanaman Obat .....	6
Gambar 2 Bentuk Daun.....	7
Gambar 3 Lingkaran Elemen Hue.....	11
Gambar 4 Sudut dalam GLCM .....	13
Gambar 5 Hard Margin SVM .....	16
Gambar 6 Hyperplane terbaik yang memisahkan antar dua kelas positif (+1) dan negatif (-1).....	16
Gambar 7 Beberapa misklasifikasi pada Soft Margin SVM.....	17
Gambar 8 Kernel SVM untuk memisahkan Data Secara Linear .....	18
Gambar 9 Multi Class Confusion Matrix.....	22
Gambar 10 Ilustrasi Kerja <i>Grabcut</i> .....	24
Gambar 11 Tahapan Penelitian .....	29
Gambar 12 Skenario pengambilan gambar .....	32
Gambar 13 <i>Flowchart</i> Perancangan Sistem.....	33
Gambar 14 <i>Split Dataset</i> .....	36
Gambar 15 <i>Preprocessing</i> .....	36
Gambar 16 Proses augmentasi pada citra.....	37
Gambar 17 Proses resize pada citra .....	37
Gambar 18 ROI yang tentukan .....	38
Gambar 19 Metode <i>Grabcut</i> .....	38
Gambar 20 Implementasi <i>Grabcut</i> .....	39
Gambar 21 Inisiasi mask.....	40
Gambar 22 Pemisahan <i>background</i> dan <i>foreground</i> citra.....	40
Gambar 23 Hasil segmentasi pada citra .....	41
Gambar 24 Citra keabuan.....	41
Gambar 25 Citra hasil <i>thresholding</i> .....	41
Gambar 26 Citra hasil <i>cropping</i> .....	42
Gambar 27 Konversi Citra HSV .....	43
Gambar 28 Nilai HSV .....	44
Gambar 29 Nilai korelasi antar fitur .....	49
Gambar 30 Fitur yang tidak digunakan.....	49
Gambar 31 Fitur yang memiliki nilai optimal.....	50
Gambar 32 Diagram alir SVM.....	51
Gambar 33 Proses pengembangan model kedalam android .....	53
Gambar 34 Proses klasifikasi tanaman obat dengan aplikasi android .....	54
Gambar 35 Hasil penentuan parameter dari metode <i>Grid Search</i> .....	57
Gambar 36 Tampilan halaman utama .....	60
Gambar 37 Halaman deteksi tanaman obat.....	61
Gambar 38 Halaman Deteksi dengan kamera.....	62
Gambar 39 Tampilan sistem setelah <i>user</i> memasukkan gambar .....	62
Gambar 40 Tampilan ketika sistem telah melakukan proses klasifikasi.....	63
Gambar 41 Tampilan hasil prediksi dengan probabilitas.....	64
Gambar 42 Tampilan hasil klasifikasi tanaman obat .....	64
Gambar 43 Tampilan halaman galeri hasil .....	65
Gambar 44 Confusion Matrix uji performa model .....	66

Gambar 45 Tampilan sistem setelah <i>user</i> memasukkan gambar untuk data uji ...	68
Gambar 46 Hasil Preprocessing Data Uji .....	68
Gambar 47 Hasil Nilai Fitur Data Uji .....	69
Gambar 48 Hasil Probabilitas Tiap Kelas Tanaman pada Data Uji.....	70
Gambar 49 Hasil Klasifikasi pada Data Uji.....	70
Gambar 50 Tampilan nilai probabilitas.....	71
Gambar 51 Halaman informasi tanaman obat hasil klasifikasi.....	71
Gambar 52 Data uji yang tidak masuk dalam data training .....	76
Gambar 53 Data uji bayam duri yang mengalami kesalahan klasifikasi .....	82
Gambar 54 Tanaman obat anting-anting pada data <i>training</i> yang memiliki kesamaan dengan data uji tanaman obat bayam duri .....	82
Gambar 55 Data uji sidaguri yang mengalami kesalahan klasifikasi .....	83
Gambar 56 Tanaman obat kirinyuh pada data <i>training</i> yang memiliki kesamaan dengan data uji tanaman obat sidaguri .....	83
Gambar 57 Gambar Kesalahan prediksi bukan tanaman obat .....	84
Gambar 58 Hasil segmentasi.....	84

## DAFTAR TABEL

Tabel 1. Garis Besar Human Vision dan Computer Vision .....	8
Tabel 2. Jenis Fungsi Kernel.....	19
Tabel 3. Contoh 4 SVM <i>biner</i> dengan metode <i>One-Against-All</i> .....	20
Tabel 4. Contoh 6 SVM <i>biner</i> dengan metode <i>One-Against-One</i> .....	21
Tabel 5. Jumlah data per kelas .....	34
Tabel 6. Citra <i>grayscale</i> .....	45
Tabel 7. Komposisi piksel.....	45
Tabel 8. Jumlah pasangan piksel dengan dengan sudut $90^0$ .....	45
Tabel 9. Matriks GLCM $90^0$ sudut yang telah dinormalisasi.....	46
Tabel 10. Total data per-kelas.....	52
Tabel 11. <i>Confusion Matrix</i> .....	55
Tabel 12. Contoh <i>confusion matrix</i> dimensi 10x10 tanaman obat untuk kondisi prediksi tanaman pada kelas 4 .....	55
Tabel 13. Perbandingan hasil OvO dengan OvA .....	58
Tabel 14 Hasil uji performa model .....	66
Tabel 15. Hasil Prediksi 100 Data Uji dengan Probabilitas.....	72
Tabel 16. Hasil kesalahan klasifikasi data uji .....	73
Tabel 17. Confusion Matriks Data Uji pada Aplikasi Android .....	74
Tabel 18. Hasil uji performa pada Aplikasi Android.....	74
Tabel 19. Contoh kesalahan prediksi bukan tanaman obat .....	77
Tabel 20. Evaluasi Confusion Matrix Berdasarkan <i>Threshold</i> .....	77
Tabel 21. Performa Sistem Berdasarkan <i>Threshold</i> .....	78
Tabel 22. Performa Sistem Sebelum dan Setelah dengan <i>Threshold</i> .....	79
Tabel 23. Hasil Klasifikasi Data Uji Bukan Tanaman Obat dengan <i>Threshold</i> ...	80
Tabel 24. Performa Sistem Keseluruhan Data Uji dengan <i>Threshold</i> .....	81
Tabel 25. Fitur bentuk data uji bayam duri salah klasifikasi .....	82
Tabel 26. Fitur bentuk salah satu bayam duri pada data training.....	82
Tabel 27. Fitur warna data uji sidaguri salah klasifikasi.....	83
Tabel 28. Fitur warna salah satu kirinyuh pada data training .....	83

## DAFTAR LAMPIRAN

Lampiran 1. Beberapa Contoh Dataset Primer .....	91
Lampiran 2. Source Code Preprocessing .....	101
Lampiran 3. Source Code Ekstraksi Fitur .....	102
Lampiran 4. Source Code Klasifikasi M-SVM.....	105
Lampiran 5. Source Code Android .....	107
Lampiran 6. Contoh Dataset Hasil Ekstraksi Fitur pada Kelas 1.....	110
Lampiran 7. Contoh Dataset Hasil Seleksi Fitur pada 10 Kelas .....	111
Lampiran 8. Lembar Perbaikan Skripsi .....	119

## DAFTAR SINGKATAN DAN ARTI SIMBOL

---

Lambang/Singkatan	Arti dan Keterangan
SVM	<i>Support Vector Machine</i>
HSV	<i>Hue, Saturation, Value</i>
RGB	<i>Red, Green, Blue</i>
GLCM	<i>Gray Level Co-occurrence</i>
ASM	<i>Angular Second Moment</i>
OvO	<i>One-Agains-One</i>
OvA	<i>One-Agains-All</i>
RBF	<i>Radial Basis Function</i>
ROI	<i>Region Of Interes</i>
TP	<i>True Positive</i>
FP	<i>False Positive</i>
FN	<i>False Negative</i>
TN	<i>True Negative</i>
API	<i>Application Programming Interface</i>

## KATA PENGANTAR

Puji syukur penulis panjatkan atas kehadiran Allah SWT karena berkat Rahmat dan Karunia-Nya sehingga Tugas Akhir yang berjudul **“KLASIFIKASI TANAMAN OBAT MENGGUNAKAN *MULTICLASS SUPPORT VECTOR MACHINE* BERBASIS ANDROID”** ini dapat terselesaikan dengan baik sebagai salah satu syarat dalam menyelesaikan jenjang Strata-1 pada Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

Penulis menyadari bahwa banyak kendala yang dihadapi dalam menyelesaikan skripsi ini, namun berkat dorongan, dukungan, bimbingan serta motivasi yang diberikan oleh berbagai pihak, sehingga penyusunan skripsi ini dapat terselesaikan dengan baik. Ucapan terima kasih serta penghargaan yang setinggi-tingginya penulis sampaikan kepada:

1. Orang tua penulis, Mama Rahmawati Pawelloi, Bapak Achmad Amiruddin, Ummi Nurhidayani Pawelloi, dan Abi Syamsuir Sawedi yang selalu memberikan doa, dukungan semangat, dan motivasi, serta selalu sabar dalam mendidik penulis sejak kecil.
2. Bapak Prof. Dr. Indrabayu, ST., MT., M.Bus.Sys., IPM., ASEAN Eng. selaku selaku Ketua Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin sekaligus selaku pembimbing I, dan Bapak A. Ais Prayogi Alimuddin, S.T., M.Eng., selaku pembimbing II, yang senantiasa menyediakan waktu, tenaga, pikiran, perhatian serta memberikan saran dan arahan dalam menyelesaikan tugas akhir.
3. Segenap Dosen dan Staf Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang telah banyak membantu penulis selama masa perkuliahan.
4. Adik-adik penulis, Alim, Amira, Aqila, dan Aghnia yang telah memberikan dukungan kepada penulis untuk tetap semangat dalam penyusunan tugas akhir, terkhusus adik Amira yang turut membantu penulis selama proses pengambilan data lapangan.
5. Muhammad Abdillah Rahmat yang selalu menemani, membantu dan memberikan supportnya kepada penulis. Sahabat penulis Priska, Ilmi, Jum, Suci dan Rini yang selalu kebersamai disetiap proses dan moment selama masa perkuliahan di Teknik Informatika. Serta Yusril, Fadhil, Nublan, Irfan, dan Taslinda yang juga selalu membantu penulis dalam pengerjaan tugas akhir ini.
6. Teman-teman dan kakak-kakak AIMP *Research Group* yang telah banyak meluangkan waktu untuk mendengarkan keluh kesah penulis, juga saran-saran dan bantuan dalam proses penyelesaian tugas akhir ini.
7. Teman-teman RECOGNIZER atas dukungan, bantuan, dan semangat yang diberikan selama ini.
8. Serta berbagai pihak atas segala dukungan dan bantuannya yang tidak dapat penulis tuliskan satu persatu.
9. *Last but not least, I wanna thank me. I wanna thank me for believing in me, I wanna thank me for doing all this hard work, I wanna thank me for having no days off. I wanna thank me for never quitting, I wanna thank me for always being a giver and tryna give more than I receive. I wanna thank me*

*for tryna do more right than wrong, I wanna thank me for just being me at all times.*

Akhir kata, penulis berharap semoga Tuhan Yang Maha Esa berkenan membalas segala kebaikan dari semua pihak yang telah banyak membantu. penulis menyadari masih terdapat kekurangan dalam penyusunan Laporan Tugas Akhir ini baik isi maupun cara penyajian. Oleh dikarenakan itu penulis mengharapkan adanya bentuk saran serta masukan yang membangun demi kesempurnaan laporan ini. Penyusun berharap semoga Laporan Tugas Akhir ini dapat memberikan manfaat bagi pembaca pada umumnya dan manfaat bagi penulis pada khususnya. Aamiin.

Gowa, Maret 2023

Penulis,  
Rieka Zalzabillah Putri

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Tanaman merupakan tumbuhan yang hidup dimana saja baik itu di lingkungan rumah, kebun, maupun hutan. Pada dasarnya, tanaman dapat dimanfaatkan sebagai sumber pangan, sandang, dan juga sebagai obat. Dalam kehidupan masyarakat tanaman dimanfaatkan sebagai obat untuk pengobatan segala jenis penyakit. Pemanfaatan tanaman sebagai obat sejak dulu diminati oleh masyarakat desa, hal itu ditandai dengan banyaknya tempat pengobatan tradisional serta banyak beredar produk obat tradisional di tengah-tengah masyarakat, yang biasa disebut herbal. Herbal merupakan tumbuhan atau tanaman yang memiliki khasiat sebagai obat. Herbal merupakan kekayaan alam Indonesia yang tidak ternilai harganya (Harefa et al., 2020). Ada beberapa tanaman obat yang dikenal memiliki efek samping yang rendah dibandingkan obat kimia, sehingga dewasa ini banyak orang yang menggunakan tanaman obat untuk mengobati penyakit, serta mengurangi penggunaan obat kimia (Darnita et al., 2020).

Masyarakat sering salah dalam menentukan bahan baku dalam pembuatan obat tradisional dan tidak mengerti cara untuk mengolah bahan tersebut. Hal ini dapat menyebabkan efek samping yang berbeda bagi tiap orang jika dosis obat diberikan secara berlebihan (Renny et al., 2017). Karena banyaknya jumlah dan jenis tanaman obat, tak jarang masyarakat sulit membedakan dan mengenali tanaman obat karena tanaman yang memiliki kesamaan atau tidak memiliki perbedaan yang cukup signifikan. Sehingga pengenalan otomatis tanaman obat akan membantu masyarakat untuk meningkatkan pengetahuan tentang tanaman obat.

Kecerdasan Buatan atau *Artificial Intelligence* (AI) adalah teknik yang digunakan untuk meniru kecerdasan yang dimiliki oleh makhluk hidup maupun benda mati untuk menyelesaikan sebuah persoalan. *Machine Learning* (ML) atau pembelajaran mesin merupakan pendekatan dalam AI yang banyak digunakan

untuk menggantikan atau menirukan perilaku manusia untuk menyelesaikan masalah atau melakukan otomatisasi. Setidaknya ada dua aplikasi utama dalam ML yaitu, klasifikasi dan prediksi. Ciri khas dari ML adalah adanya proses pelatihan, pembelajaran, atau *training*. Oleh karena itu, ML membutuhkan data untuk dipelajari yang disebut sebagai data *training*. Klasifikasi adalah metode dalam ML yang digunakan oleh mesin untuk memilih atau mengklasifikasikan objek berdasarkan ciri tertentu sebagaimana manusia mencoba membedakan benda satu dengan yang lain (Ahmad, n.d.). Seiring berkembangnya teknologi klasifikasi dapat dilakukan secara tidak langsung dengan melakukan klasifikasi citra objek tersebut berdasarkan pada beberapa fitur. Fitur-fitur yang dapat diekstrak dalam sebuah citra dapat berupa warna, bentuk, dan tekstur (Budi Prawira & Widiastiwi, 2021). Salah satu algoritma klasifikasi yang dimiliki machine learning yaitu *Support Vector Machine (SVM)*. *Support Vector Machine (SVM)* merupakan salah satu teknik dalam proses klasifikasi yang relatif baru, teknik ini sudah biasa dipakai para peneliti untuk menuntaskan masalah yang berkaitan dengan komputasi. Konsep dasar dari SVM adalah melalui pembentukan *hyperplane (maximal margin hyperplane)* (Budi Prawira & Widiastiwi, 2021).

Pada penelitian ini, klasifikasi tanaman obat akan dilakukan berdasarkan citra daun karena daun merupakan bagian tanaman yang mudah ditemui dan selalu ada di semua musim, namun tak jarang daun memiliki kemiripan antara satu sama lain. Daun merupakan salah satu bagian tanaman yang memiliki fitur yang khas sehingga dapat diterapkan dalam proses identifikasi spesies citra daun. Proses klasifikasi citra daun dapat dilakukan dengan cara mengidentifikasi citra bentuk daun yang dapat dilakukan dengan cara pengenalan pola daun dengan mengenali karakteristik struktural daun seperti bentuk dan tekstur daun tersebut (Hidayat & Makhsun, 2018). Berdasarkan latar belakang tersebut penulis mengusulkan judul **“Klasifikasi Tanaman Obat menggunakan *Multiclass Support Vector Machine* Berbasis Android”**. Sistem yang diusulkan berbasis android agar dapat mudah digunakan masyarakat dalam mengenali tanaman obat yang biasa dijumpai pada kehidupan sehari-hari. Selain melakukan klasifikasi untuk pengenalan pada

tanaman obat, sistem ini memberikan kepastian yang akan membantu masyarakat dalam mendapatkan informasi yang dibutuhkan terkait tanaman obat.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang dijelaskan, maka rumusan masalah pada tugas akhir ini adalah:

- a. Bagaimana membangun suatu sistem klasifikasi tanaman obat menggunakan *Multiclass Support Vector Machine* berbasis android?
- b. Bagaimana tingkat akurasi sistem klasifikasi tanaman obat dengan menggunakan *Multiclass Support Vector Machine*?

## 1.3 Tujuan Penelitian

Untuk bagian tujuan masalah, penulis membagi menjadi beberapa poin:

- a. Untuk merancang sistem klasifikasi tanaman obat menggunakan *Multiclass Support Vector Machine* berbasis android.
- b. Untuk mengetahui tingkat akurasi sistem klasifikasi tanaman obat dengan menggunakan *Multiclass Support Vector Machine*.

## 1.4 Manfaat Penelitian

Untuk manfaat yang dapat diberikan dari penelitian ini adalah:

- a. Bagi masyarakat, penelitian ini dapat digunakan sebagai alat bantu untuk mengklasifikasikan tanaman obat agar dapat mengurangi kesalahan dalam mengidentifikasi jenis tanaman obat dan membantu dalam memperoleh informasi terkait tanaman obat tersebut.
- b. Bagi peneliti, penelitian ini dapat dijadikan referensi dalam menggunakan metode *Multiclass Support Vector Machine* dalam mengklasifikasi tanaman dan menjadi referensi dalam menggunakan sistem berbasis android untuk implementasi.
- c. Bagi institusi, penelitian ini dapat digunakan sebagai referensi ilmiah untuk penelitian-penelitian selanjutnya.

## 1.5 Ruang Lingkup

Untuk bagian ruang lingkup pada penelitian ini, penulis membagi menjadi beberapa poin:

- a. Data yang digunakan adalah data primer yang merupakan 10 jenis tanaman obat yang biasa dijumpai masyarakat disekitar. Data yang dikumpulkan dari wilayah Maros dan Gowa Sulawesi Selatan.
- b. Bagian tanaman obat yang digunakan dalam penelitian ini adalah daun tanaman obat.
- c. Data yang digunakan merupakan gambar / foto yang diambil menggunakan kamera *smartphone* dengan rasio 1:1. Data diambil pada siang hari atau dengan kondisi cuaca cerah.
- d. Pembangunan model klasifikasi menggunakan *Multiclass Support Vector Machine* dan diimplementasi berbasis android.
- e. Evaluasi keakuratan sistem menggunakan metode *Confusion Matrix*.

## 1.6 Sistematika Penulisan

Pada bagian ini memberikan gambaran singkat mengenai isi tulisan secara keseluruhan pada tugas akhir ini, maka akan diuraikan beberapa tahapan dari penulisan secara sistematis, yaitu:

### **BAB 1 PENDAHULUAN**

Pada bab ini diuraikan mengenai latar belakang masalah dari judul penelitian Klasifikasi Tanaman Obat menggunakan *Multiclass Support Vector Machine* berbasis Android, disertai dengan rumusan masalah, tujuan penelitian, manfaat penelitian, ruang lingkup dan sistematika penulisan.

### **BAB II TINJAUAN PUSTAKA**

Dalam bab ini berisi teori-teori yang berhubungan dengan proses penelitian yang dilakukan untuk menganalisis masalah yang akan diteliti serta hal-hal lain yang berhubungan dengan variabel-variabel data yang akan digunakan, ekstraksi fitur, dan *Multiclass Support Vector Machine*.

### **BAB III METODOLOGI PENELITIAN**

Pada bab ini berisi tentang apa saja yang akan dilakukan pada saat penelitian, yang meliputi tahapan penelitian, waktu dan lokasi, instrumen penelitian, perancangan sistem dan analisis kerja sistem.

### **BAB IV HASIL DAN PEMBAHASAN**

Dalam bab ini berisi tentang hasil pengolahan data serta pembahasan yang disertai tabel hasil penelitian.

### **BAB V PENUTUP**

Dalam bab ini berisi tentang kesimpulan yang didapatkan berdasarkan hasil penelitian yang dilakukan, serta memberikan saran-saran untuk penelitian/pengembangan selanjutnya.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Tanaman Obat**

Dalam kehidupan masyarakat tanaman dimanfaatkan sebagai obat untuk pengobatan segala jenis penyakit. Pemanfaatan tanaman sebagai obat sejak dulu diminati oleh masyarakat desa, hal itu ditandai dengan banyaknya tempat pengobatan tradisional serta banyak beredar produk obat tradisional di tengah-tengah masyarakat, yang biasa disebut herbal. Herbal merupakan tumbuhan atau tanaman yang memiliki khasiat sebagai obat. Herbal merupakan kekayaan alam Indonesia yang tidak ternilai harganya. Herbal dalam kehidupan sehari-hari sering kita kenal sebagai rempah-rempah. Herbal dapat berperan dalam pertolongan pertama pada gangguan kesehatan dan terbukti ampuh menyembuhkan berbagai macam penyakit bahkan penyakit yang hampir tidak dapat disembuhkan sekalipun masih berpeluang dapat disembuhkan menggunakan ramuan herbal. Hal tersebut karena pengobatan dengan herbal lebih mudah diterima tubuh sehingga kesembuhannya pun dapat lebih cepat (Darmawan, 2020).



Gambar 1 Tanaman Obat

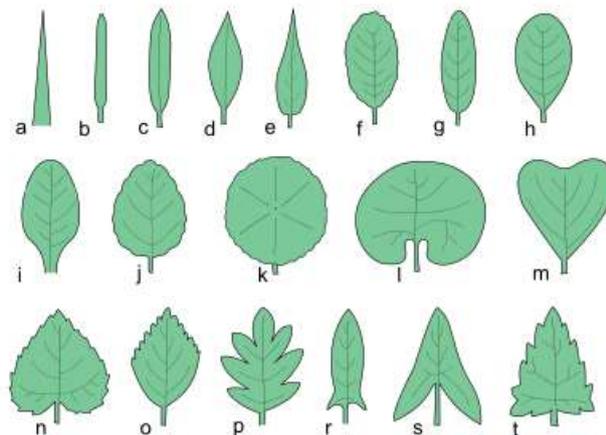
#### **2.2 Daun**

Daun merupakan salah satu organ tumbuhan yang tumbuh dari batang, umumnya berwarna hijau (mengandung klorofil) dan terutama berfungsi sebagai penangkap energi dari cahaya matahari melalui fotosintesis. Daun merupakan organ terpenting bagi tumbuhan dalam melangsungkan hidupnya karena tumbuhan adalah

organisme autotrof obligat, ia harus memasok kebutuhannya sendiri melalui konversi energi cahaya menjadi energi kimia (Latifa, 2015).

Daun tumbuhan memiliki bentuk dan ukuran yang bervariasi, mulai dari yang berbentuk duri kecil pada kaktus hingga yang berbentuk lebar pada palm. Sekalipun bentuk dan ukuran daun tampak bervariasi, pada dasarnya daun terdiri dari tiga bagian, yaitu bagian basal yang berkembang menjadi pelepah (vagina), tangkai daun (petiolus) dan helaian daun (lamina). Daun yang memiliki ketiga bagian tersebut dinamakan daun lengkap. Pada sebagian besar tumbuhan, daun hanya terdiri dari satu atau dua bagian saja, yakni helaian daun saja, tangkai dan helaian daun, pelepah dan helaian daun, atau tangkai daun saja. Daun-daun yang demikian dinamakan sebagai daun tak lengkap (Latifa, 2015).

Daun dibagi menjadi daun tunggal dan daun majemuk, pada daun majemuk terdapat sejumlah anak daun yang melekat pada tangkai daun atau perpanjangannya pada sumbu (Rachis) yang sama. Bentuk-bentuk daun dapat dilihat pada Gambar 2 yaitu : a) pedang/belati, b) jarum, c) linear, d) lanset, e) lanset oval, f) bulat telur, g) telur pipih, h) oval meruncing, i) sudip, j) bulat telur, k) lingkaran, l) ginjal, m) jantung terbalik, n) jantung, o) belah ketupat, p) berbagi menyirip, r) tombak, s) anak panah, dan t) segitiga (Isman, Ahmad, and Latief 2021).



Gambar 2 Bentuk Daun

## 2.3 Visi Komputer

*Computer vision* adalah ilmu dan teknologi mesin yang melihat dimana mesin mampu mengekstrak informasi dari gambar yang diperlukan untuk menyelesaikan tugas tertentu. Pada hakikatnya, *computer vision* mencoba meniru cara kerja visual manusia. Dalam proses penglihatan manusia, manusia melihat objek menggunakan indera penglihatan yaitu mata, lalu citra objek diteruskan ke otak untuk diinterpretasikan sehingga manusia mengerti objek apa yang tampak. Hasil interpretasi ini kemudian digunakan untuk pengambilan keputusan (Munir & Yuliani, 2019).

Tabel 1. Garis Besar Human Vision dan Computer Vision

<i>Human Vision</i>	<i>Computer Vision</i>
Menggunakan mata dan visual <i>cortex</i> di dalam otak.	Menggunakan kamera-kamera yang terhubung pada sistem komputer
Menemukan dari gambar objek apa yang ada dalam penglihatan, dimana posisinya, bagaimana mereka bergerak, dan apa bentuknya.	Secara otomatis menginterpretasi gambar-gambar dan mencoba untuk mengerti isinya seperti pada <i>human vision</i>

Tabel 1 menunjukkan gambaran umum antara *human vision* dan *computer vision*. Untuk itu *computer vision* mampu mengenali objek yang diamati. Kemampuan untuk mengenali ini merupakan kombinasi dari pengolahan citra dan pengenalan pola. Pengolahan citra adalah tahap awal dalam *computer vision* untuk menghasilkan citra yang lebih baik atau lebih mudah diinterpretasikan, sedangkan pengenalan pola adalah proses identifikasi objek pada citra. Proses-proses dalam *computer vision* secara garis besar dapat dibagi menjadi (Basuki & Yuliani, 2019):

- a. Proses mengakuisisi citra digital (*image acquisition*)
- b. Proses pengolahan citra (*image processing*)
- c. Proses analisis data citra (*image analysis*)
- d. Proses pemahaman data citra (*image understanding*)

Sebagai suatu disiplin ilmu, visi computer berkaitan dengan teori-teori di balik sistem buatan yang mengekstrak informasi dari gambar. Informasi yang diekstrak dari gambar dapat berupa data-data yang berbeda-beda, seperti urutan jalannya video, intensitas cahaya atau perspektif dari sudut gambar yang berbeda-beda. Visi komputer sekarang ini sering digunakan untuk mendeteksi wajah pada gambar (*face detection*), mengenali ekspresi wajah (*facial expression recognition*) dan dalam prakteknya sering digunakan Bersama dengan jaringan syaraf tiruan (*artificial neural network*).

## 2.4 Ekstraksi Fitur

Fitur merupakan karakteristik unik dari suatu objek. Fitur dibedakan menjadi dua yaitu fitur alami dan buatan. Fitur alami adalah bagian dari gambar, misalnya kecerahan dan tepi objek. Sedangkan fitur buatan adalah fitur yang diperoleh dengan operasi tertentu pada gambar, misalnya histogram tingkat keabuan. Sehingga ekstraksi fitur adalah proses untuk mendapatkan ciri-ciri pembeda yang membedakan suatu objek dari objek yang lain (Admin, 2018).

Menurut (Sari dkk, 2017) ekstraksi fitur merupakan suatu pengambilan ciri atau *feature* dari suatu bentuk yang nantinya nilai yang didapatkan akan dianalisis untuk proses selanjutnya. Ekstraksi fitur dilakukan dengan cara menghitung jumlah titik atau piksel yang ditemui dalam setiap pengecekan. Pengecekan dilakukan dalam berbagai arah koordinat kartesius dari citra digital yang dianalisis, yaitu vertikal, horizontal, diagonal kanan, dan diagonal kiri. Fitur yang didapat dari sebuah citra merupakan ciri khas pembeda dengan citra-citra yang lain. Pada ekstraksi fitur warna, gambar dihitung menggunakan model warna RGB.

### 2.4.1 Fitur Bentuk

Bentuk dari suatu objek adalah karakter konfigurasi permukaan yang diwakili oleh garis dan kontur. Fitur bentuk dikategorikan bergantung pada teknik yang digunakan. Kategori tersebut adalah berdasarkan batas (*boundary-based*) dan berdasarkan daerah (*region-based*). Teknik berdasarkan batas (*boundary-based*) menggambarkan bentuk daerah

dengan menggunakan karakteristik eksternal, contohnya adalah piksel sepanjang batas objek. Sedangkan teknik berdasarkan daerah (*region-based*) menggambarkan bentuk wilayah dengan menggunakan karakteristik internal, contohnya adalah piksel yang berada dalam suatu wilayah. Fitur bentuk yang biasa digunakan adalah:

- a. *Area*, yang merupakan jumlah piksel dalam wilayah yang digambarkan oleh bentuk (*foreground*).
- b. *Perimeter* atau lingkaran adalah jumlah dari piksel yang berada pada batas dari bentuk, perimeter didapatkan dari hasil deteksi tepi.
- c. *Physiological length* ( $L_p$ ), merupakan jarak antara ujung dan pangkal daun (Panjang tulang daun primer).
- d. *Physiological width* ( $L_w$ ), merupakan jarak terpanjang dari garis yang memotong tegak lurus *physiological length* yang dibatasi tepi daun.
- e. *Aspect ratio*, merupakan rasio antara *physiological length* dan *physiological width*.

$$\text{Aspect ratio} = \frac{\text{physiological length } (L_p)}{\text{physiological width } (W_p)} \quad (1)$$

- f. *Rectangularity*, digunakan untuk mendiskripsikan kemiripan antara daun dan empat persegi panjang.

$$RI = \frac{\text{physiological length}(L_p) \cdot \text{physiological width}(W_p)}{\text{Area}} \quad (2)$$

- g. *Circularity* merupakan salah satu fitur penting yang umum digunakan. *Circularity* biasanya didefinisikan dalam persamaan:

$$\text{Circularity} = 4\pi * \frac{\text{area}}{\text{perimeter}^2} \quad (3)$$

- h. *Eccentricity*, merupakan rasio jarak antar fokus *elips* dengan ujung sumbu utama (*mayor axis*) sebuah objek. *Eccentricity* bernilai antara 0-1. *Eccentricity* merupakan Teknik untuk menggambarkan sebuah objek dengan bentuk elips.

$$\text{Eccentricity} = \frac{\sqrt{(\text{mayor axis}^2 - \text{minor axis}^2)}}{\text{mayor axis}} \quad (4)$$

- i. *Metric* merupakan nilai perbandingan antara luas dan keliling objek. *Metric* memiliki rentang nilai antara 0 hingga 1. Objek berbentuk

memanjang/mendekati bentuk garis lurus, nilai metricnya mendekati angka 0, sedangkan objek yang berbentuk bulat/lingkaran, nilai metricnya mendekati angka 1.

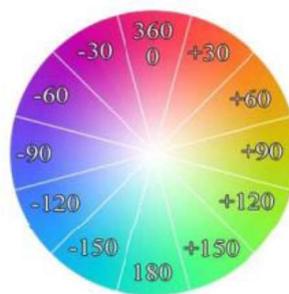
$$Metric = \frac{4\pi \times area}{Circumference^2} \quad (5)$$

## 2.4.2 Fitur Warna

Pada ekstraksi fitur warna, ciri pembeda adalah warna. Biasanya ekstraksi fitur ini digunakan pada citra berwarna yang memiliki komposisi warna RGB (*red, green, blue*). Fitur warna adalah fitur penting dalam citra sebab warna dapat dilihat oleh mata manusia. Untuk membedakan suatu objek dengan warna tertentu dapat menggunakan nilai *hue* yang merupakan representasi dari cahaya tampak (merah, jingga, kuning, hijau, biru, ungu). Nilai *hue* dapat dikombinasikan dengan nilai *saturation* dan *value* yang merupakan tingkat kecerahan suatu warna. Untuk mendapatkan ketiga nilai tersebut, perlu dilakukan konversi ruang warna citra yang semula RGB menjadi HSV (*Hue, Saturation, Value*).

### - *Hue*

*Hue* merupakan salah satu elemen dalam ruang warna HSV yang mewakili nilai warna sehingga nilai toleransi *hue* juga akan mempengaruhi nilai warna yang terseleksi dalam proses segmentasi. Nilai *hue* direpresentasikan dalam bentuk lingkaran dan memiliki rentang berupa sudut antara  $0^\circ$  -  $360^\circ$  (Budi et al., 2010), penggambaran elemen warna *hue* dapat dilihat pada Gambar 3:



Gambar 3 Lingkaran Elemen *Hue*

Oleh karena elemen warna *hue* berupa lingkaran dan dituliskan dalam sudut maka setiap operasi yang berkaitan dengan elemen warna *hue* (penambahan/ pengurangan, perhitungan toleransi, filter warna) merupakan operasi sudut.

- *Saturation*

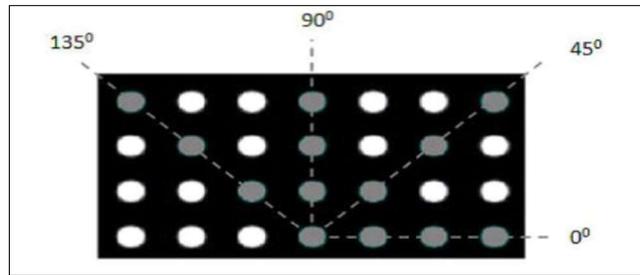
*Saturation* merupakan salah satu elemen warna HSV yang mewakili tingkat intensitas warna. Pada nilai tingkat kecerahan (*value*) yang sama nilai *saturation* akan menggambarkan kedekatan suatu warna pada warna abu-abu. Pada sistem nilai saturasi memiliki rentang antara 0 (minimum) dan 1 atau 100% (maksimum)

- *Value*

Dalam ruang warna HSV, untuk mempresentasikan tingkat kecerahan warna digunakan elemen *value*. Pada nilai *value* maksimum warna yang dihasilkan adalah warna dengan tingkat kecerahan minimum (warna hitam). Berapapun nilai *hue* dan *saturation* warna, jika nilai *value*, yang dimiliki adalah 0 (minimum) maka warna yang dihasilkan adalah warna hitam. Nilai *value* maksimum adalah 1 (100%), dimana warna yang dihasilkan akan memiliki tingkat kecerahan maksimum.

### 2.4.3 Fitur Tekstur

Pada ekstraksi fitur ini, fitur pembeda adalah tekstur yang merupakan karakteristik penentu pada citra. Ciri tekstur merupakan ciri penting dalam sebuah gambar yang merupakan informasi berupa susunan struktur permukaan suatu gambar. Teknik statistik yang terkenal untuk ekstraksi fitur adalah matriks *Gray Level Co-occurrence* (GLCM). GLCM merupakan suatu matriks yang menggambarkan frekuensi munculnya pasangan dua piksel dengan intensitas tertentu dalam jarak dan arah tertentu dalam citra (Praseptiyana, 2019).



Gambar 4 Sudut dalam GLCM

Koordinat pasangan piksel memiliki jarak  $d$  dan orientasi sudut  $\theta$ . Jarak direpresentasikan dalam derajat. Orientasi sudut terbentuk berdasarkan empat arah sudut yaitu  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ , dan  $135^\circ$ , dan jarak antar piksel sebesar 1 piksel (Surya et al., 2017). Tahapan yang dilakukan pada perhitungan GLCM adalah sebagai berikut:

1. Pembentukan matriks awal GLCM dari pasangan dua piksel yang berjajar sesuai dengan arah  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ , dan  $135^\circ$ .
2. Membentuk matriks yang simetris dengan menjumlahkan matriks awal GLCM dengan nilai transposnya.
3. Menormalisasikan matriks GLCM dengan membagi setiap elemen matriks dengan jumlah pasangan piksel.
4. Ekstraksi ciri yaitu:

a. *Homogeneity*

*Homogeneity* menunjukkan kehomogenan variasi intensitas dalam citra. Persamaan *homogeneity* sebagai berikut (Praseptiyana, 2019):

$$\text{Homogeneity} = \sum_i^m \sum_j^n p \frac{p(i,j)}{1+(i-j)^2} \quad (6)$$

b. *ASM (Angular Second Moment)*

*ASM* bisa juga disebut *uniformity* atau *energy*. *ASM* adalah penjumlahan pangkat dari elemen matriks GLCM. *ASM* memiliki nilai yang tinggi Ketika citra memiliki *homogenitas* yang baik atau nilai piksel yang hampir serupa. Adapun persamaan *ASM* sebagai berikut (Alvansga, 2019):

$$\text{ASM} = \sum_{i=1}^m \sum_{j=1}^n (p(i,j))^2 \quad (7)$$

c. *Energy*

*Energy* merupakan fitur GLCM yang digunakan untuk mengukur konsentrasi pasangan intensitas pada matriks GLCM (Praseptiyana, 2019), dan didefinisikan dengan persamaan sebagai berikut:

$$Energy = \sqrt{ASM} \quad (8)$$

d. *Contrast*

*Contrast* adalah perhitungan perbedaan intensitas antara piksel satu dan piksel yang berdekatan diseluruh gambar. Kontras bernilai nol untuk gambar yang konstan (Praseptiyana, 2019). Persamaan *contrast*:

$$Contrast = -\sum_{i=0}^m \sum_{j=0}^n p_{(i,j)} (i - j)^2 \quad (9)$$

e. *Correlation*

*Correlation* menyatakan ukuran hubungan linier dari nilai gray level piksel ketetanggaan. Nilai *correlation* dapat dicari menggunakan persamaan (Septiarini et al., 2015):

$$Correlation = \sum_{i,j} \frac{(1-\mu_i)(j-\mu_j)p(i,j)}{\sigma_i \sigma_j} \quad (10)$$

f. *Dissimilarity*

*Dissimilarity* mengukur ketidakmiripan suatu tekstur, yang akan bernilai besar bila acak dan sebaliknya akan bernilai kecil bila seragam (Yunus, 2020). Berikut persamaannya:

$$Dissimilarity = \sum_i \sum_j |i - j| p(i,j) \quad (11)$$

Dimana:

p = matriks GLCM normalisasi

i = indeks baris matriks p

j = indeks kolom matriks p

$\mu$  = korelasi

$\sigma$  = sudut

g. *Entropy*

*Entropy* digunakan untuk mengukur keteracakan dari distribusi intensitas (Praseptiyana, 2019). Persamaan *entropy*:

$$Entropy = -\sum_i^m \sum_j^n p(i,j) \log \{p(i,j)\} \quad (12)$$

## 2.5 Klasifikasi dengan *Support Vector Machine*

*Support Vector Machine* (SVM) merupakan salah satu teknik dalam proses klasifikasi yang relatif baru, teknik ini sudah biasa dipakai para peneliti untuk menuntaskan masalah yang berkaitan dengan komputasi. Konsep dasar dari SVM adalah melalui pembentukan *hyperplane* (*maximal margin hyperplane*) (Prawira, 2021). SVM merupakan sistem pembelajaran menggunakan ruang berupa fungsi-fungsi linear dalam sebuah ruang fitur yang berdimensi tinggi yang dilatih menggunakan algoritma pembelajaran berdasarkan pada teori optimasi dengan mengimplementasikan *learning bias* (Santosa & Tuti, 2022). Rumus perhitungan *hyperplane* algoritma SVM:

$$f: w \cdot x + b = 0 \quad (13)$$

Keterangan:

$f$  = fungsi *hyperplane*

$w$  = parameter *hyperplane* (garis tegak lurus antara garis *hyperplane* dan titik *support vector*)

$x$  = data *input* SVM ( $x_1$  = index kata,  $x_2$  = bobot kata)

$b$  = parameter *hyperplane* nilai bias

Penggunaan algoritma SVM ini mempunyai beberapa manfaat seperti kondisi model yang dibuat sangat bergantung secara eksplisit pada subset titik data, serta dalam interpretasi model tersebut mendapatkan bantuan dari *support vector*. Selain itu, SVM memakai teknik *kernel* namun hanya data yang telah terseleksi yang dapat berkontribusi dalam pembuatan suatu model klasifikasi. Hal inilah yang membuat SVM unggul, karena tidak perlu melibatkan semua data latih pada setiap iterasi pelatihan.

Pembangunan *kernel* dapat dibuat melalui urutan data dan grafik pada berbagai objek data, baik itu data diskrit maupun kontinu. Konsep substitusi kernel berlaku bagi metode lain dalam analisis data. Berikut merupakan rumus dasar dari fungsi *kernel*:

$$f(x) = \sum_{i=1}^m \alpha_i y_i K(x, x_i) + b \quad (14)$$

Keterangan;

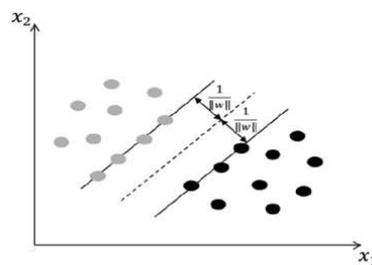
$m$  = jumlah *support vector*

$a_i$  = nilai bobot setiap titik data

$K(x, x_i)$  = fungsi *kernel*

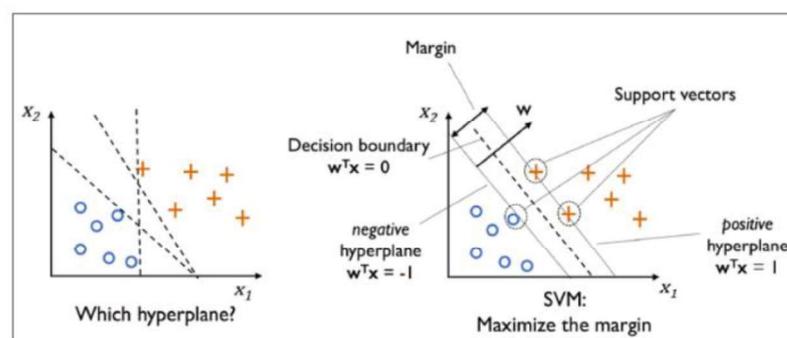
### 2.5.1 *Hard – Margin SVM / Linear SVM*

Teknik SVM merupakan *classifier* yang menemukan *hyperplane* dengan kasus data yang digunakan merupakan data dengan dua kelas yang sudah terpisah secara linear seperti pada gambar berikut:



Gambar 5 *Hard Margin SVM*

Pada Gambar 5 di atas, dapat dilihat bahwa antara kelas positif dan kelas negatif sudah terpisah secara total terlihat dari lingkaran dari lingkaran abu-abu yang berada dekat dengan garis  $x_2$  sedangkan untuk lingkaran hitam terletak dekat dengan garis  $x_1$  (Awad & Khanna, 2015).



Gambar 6 *Hyperplane* terbaik yang memisahkan antar dua kelas positif (+1) dan negatif (-1)

*Hyperplane* yang ditemukan SVM diilustrasikan seperti pada Gambar 6 di atas, posisinya berada di tengah-tengah antara dua kelas, artinya jarak antara *hyperplane* dengan objek-objek data berbeda dengan kelas yang

berdekatan (terluar) yang paling dekat dengan *hyperplane* disebut *support vector*. Objek yang disebut *support vector* paling sulit diklasifikasikan dikarenakan posisi yang hampir tumpang tindih (*overlap*) dengan kelas lain. Mengingat sifatnya yang kritis, hanya *support vector* inilah yang diperhitungkan untuk menemukan *hyperplane* yang paling optimal oleh SVM (Samsudiney, 2019).

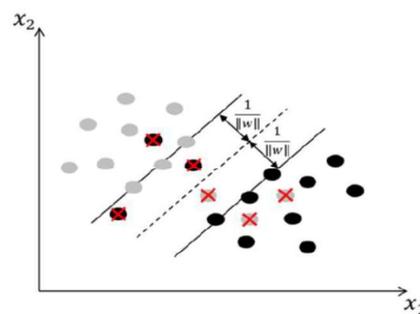
Pencarian lokasi *hyperplane* optimal merupakan inti dari metode SVM. Diasumsikan bahwa terdapat data *learning* dengan *data points*  $x_i$  ( $i=1,2,\dots,m$ ) memiliki dua kelas  $y_i = \pm 1$  yaitu kelas positif (+1) dan kelas negatif (-1) sehingga akan diperoleh *decision function* berikut:

$$f(x) = \text{sign}(w \cdot x + b) \quad (15)$$

Dimana  $(\cdot)$  merupakan scalar sehingga  $w \cdot x \equiv w^T x$

### 2.5.2 *Soft Margin SVM*

Ketika data yang digunakan tidak sepenuhnya dapat dipisahkan, *slack variables*  $\xi_i$  diperkenalkan ke dalam fungsi objektif SVM untuk memungkinkan kesalahan dalam misklasifikasi. Dalam hal ini, SVM bukan lagi *hard margin classifier* yang akan mengklasifikasi semua data dengan sempurna melainkan sebaliknya yaitu SVM *soft margin classifier* dengan mengklasifikasikan sebagian besar data dengan benar, sementara memungkinkan model untuk membuat misklasifikasi beberapa titik di sekitar batas pemisah. Berikut merupakan gambar ketika data termasuk ke dalam *soft margin SVM* (Awad & Khanna, 2015).

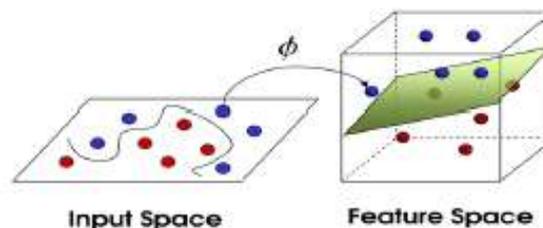


Gambar 7 Beberapa misklasifikasi pada Soft Margin SVM

Pada Gambar 7 diatas, terlihat bahwa data pada kedua kelas tidak terpisah secara sempurna dapat dilihat dari beberapa lingkaran abu-abu yang persebarannya berada di sekitar area lingkaran hitam serta sebaliknya terdapat beberapa lingkaran hitang yang persebarannya berada di sekitar lingkaran abu-abu.

### 2.5.3 Kernel SVM

Ketika terdapat permasalahan data yang tidak terpisah secara linear dalam ruang *input*, *soft margin svm* tidak dapat menemukan *hyperplane* pemisah yang kuat yang meminimalkan misklasifikasi dari *data points* serta menggeneralisasi dengan baik. Untuk itu, kernel dapat digunakan untuk mentransformasi data ke ruang berdimensi lebih tinggi yang disebut sebagai kernel, dimana akan menjadikan data terpisah secara linear (Awad & Khanna, 2015).



Gambar 8 Kernel SVM untuk memisahkan Data Secara Linear

Data disimpan dalam bentuk kernel yang mengukur kesamaan atau ketidaksamaan objek data. Kernel dapat dibangun untuk berbagai objek data mulai dari data kontinu dan data diskrit melalui urutan data dan grafik. Konsep substitusi kernel berlaku bagi metode lain dalam analisis data, tetapi SVM merupakan yang paling terkenal dari metode dengan jangkauan kelas luas yang menggunakan kernel untuk merepresentasikan data dan dapat disebut sebagai metode berbasis kernel (Cambell & Ying, 2011).

Ada 4 fungsi kernel yang cukup banyak digunakan pada masalah klasifikasi yaitu fungsi kernel *linier*, *sigmoid*, *Radial Basis Function* (RBF), dan

*polinomial* (Feta & Ginanjar, 2019). Bentuk formula pemetaan oleh untuk setiap fungsi kernel dapat dilihat pada Tabel 2

Tabel 2. Jenis Fungsi Kernel

Jenis Kernel	Definisi
<i>Linear</i>	$H(x,x') = (x,x')$
<i>Polynomial</i>	$H(x,x') = (.(x,x')+c)^d$
<i>RBF</i>	$H(x,x') = \exp (-.  x,x'   ^2)$
<i>Sigmoid</i>	$H(x,x') = \tan (.(x,x')+c)$

Dimana  $C$  merupakan parameter,  $Cost$  merupakan parameter  $gamma$ ,  $c$  merupakan *coeffisient* dan  $d$  merupakan pangkat atau derajat *polinomial*. Berdasarkan pada Tabel 2, parameter  $gamma$  menyatakan *slope* yaitu perubahan nilai *output* sebagai akibat dari perubahan nilai variabel *input*, dan parameter *coeffisient* menyatakan *intercept* dari model, sedangkan parameter pangkat atau derajat menyatakan pangkat atau derajat dari fungsi *polinomial* (Feta & Ginanjar, 2019).

Penggunaan kernel dilakukan pada percobaan untuk menentukan parameter kernel dan menghasilkan keakuratan yang terbaik dalam proses klasifikasi. Kernel linear digunakan pada saat data yang diklasifikasikan dapat dengan mudah dipisahkan dengan sebuah garis atau *hyperplane*, sementara untuk kernel *nonlinear* digunakan pada saat data yang digunakan dipisah dengan menggunakan garis lengkung atau sebuah bidang pada ruang yang mempunyai dimensi tinggi (Puspitasari et al., 2018).

## 2.6 Multiclass Support Vector Machine

SVM pertama kali diperkenalkan oleh Vapnik pada tahun 1992. SVM pertama kali hanya dapat mengkasifikasikan data kedalam dua kelas. Seiring berjalannya waktu, SVM berkembang dan mampu mengklafisikasikan data kedalam banyak kelas (*Multiclass*). Terdapat dua pendekatan untuk mengimplementasikan SVM multikelas. Pendekatan pertama, menggabungkan beberapa SVM *biner*. Pendekatan kedua, menggabungkan semua data dari semua kelas ke dalam sebuah bentuk permasalahan optimasi (Suyanto, 2019). Pendekatan

yang umum digunakan adalah pendekatan pertama karena pendekatan kedua jauh lebih rumit dalam menyelesaikan masalah optimasi. Metode yang menggunakan pendekatan pertama yaitu:

### 2.6.1 *One-Agains-All*

Sesuai dengan namanya, satu lawan semua, metode ini membandingkan satu kelas dengan semua kelas. Jika akan mengklasifikasikan data ke dalam k kelas, maka harus membangun sejumlah k model SVM *biner*. Setiap model SVM *biner* ke-i dilatih menggunakan keseluruhan data, untuk mencari jawaban apakah sebuah data diklasifikasikan sebagai kelas ke-i atau tidak. Contohnya, ketika melakukan klasifikasi kedalam 4 kelas, maka harus membangun 4 buah SVM *biner*. Dimana SVM *biner* pertama dilatih menggunakan semua data latih untuk mengklasifikasikan data kedalam kelas C1 atau bukan. SVM *biner* kedua dilatih menggunakan semua data latih untuk mengklasifikasikan data kedalam C2 atau bukan, dan begitu seterusnya (Suyanto, 2019).

Contohnya, terdapat permasalahan klasifikasi dengan 4 buah kelas. Untuk pelatihan digunakan 4 buah SVM *biner* seperti pada Tabel 3 (Pricila, 2016).

Tabel 3. Contoh 4 SVM *biner* dengan metode *One-Against-All*

<b><math>y_i = 1</math></b>	<b><math>y_i = -1</math></b>	<b>Hipotesis</b>
Kelas 1	Bukan Kelas 1	$f^1(x) = (w^1)x + b^1$
Kelas 2	Bukan Kelas 2	$f^2(x) = (w^2)x + b^2$
Kelas 3	Bukan Kelas 3	$f^3(x) = (w^3)x + b^3$
Kelas 4	Bukan Kelas 4	$f^4(x) = (w^4)x + b^4$

### 2.6.2 *One-Agains-One*

Metode *One Agains One* atau satu lawan satu prosesnya adalah membangun sejumlah model SVM *biner* yang membandingkan satu kelas dengan kelas yang lainnya. Ketika akan melakukan klasifikasi data kedalam k kelas, maka harus membangun sejumlah model dengan rumus 16 (Suyanto, 2019) :

$$\frac{k(k-1)}{2} \quad (16)$$

Keterangan :

k = jumlah kelas

SVM *biner* pertama dilatih menggunakan data latih dari kelas pertama dan kelas kedua untuk melakukan klasifikasi data kedalam kelas C1 atau C2.

SVM *biner* kedua dilatih menggunakan data latih dari kelas kedua dan kelas ketiga untuk klasifikasi kedalam kelas C2 atau C3 dan seterusnya. Sehingga, setiap kelas harus dibandingkan dengan tiga kelas lainnya. Kelas yang paling sering menang adalah kelas keputusan (Suyanto, 2019).

Contohnya, terdapat permasalahan klasifikasi dengan 4 buah kelas. Sehingga digunakan 6 buah SVM *biner* seperti pada Tabel 4

Tabel 4. Contoh 6 SVM *biner* dengan metode *One-Against-One*

<b>yi = 1</b>	<b>yi = -1</b>	<b>Hipotesis</b>
Kelas 1	Kelas 2	$f^{12}(x) = (w^{12})x + b^{12}$
Kelas 1	Kelas 3	$f^{13}(x) = (w^{13})x + b^{13}$
Kelas 1	Kelas 4	$f^{14}(x) = (w^{14})x + b^{14}$
Kelas 2	Kelas 3	$f^{23}(x) = (w^{23})x + b^{23}$
Kelas 2	Kelas 4	$f^{24}(x) = (w^{24})x + b^{24}$
Kelas 3	Kelas 4	$f^{34}(x) = (w^{34})x + b^{34}$

## 2.7 Confusion Matrix

Berdasarkan jumlah keluaran kelasnya, sistem klasifikasi dibagi menjadi 4 jenis, yaitu klasifikasi *binary*, *multi-class*, *multi-label* dan *hierarchical*. Secara umum penentuan baik tidaknya performa suatu model klasifikasi dapat dilihat dari parameter pengukuran performanya, yaitu *recall*, *precision*, *F1-Score*, dan *accuracy*. Untuk menghitung faktor-faktor tersebut diperlukan sebuah matriks yang biasa disebut dengan *confusion matrix* (Taslinda, 2022).

Terdapat empat kondisi untuk mempresentasikan hasil proses klasifikasi, yaitu (Mohajon & Tedy, 2021):

- a. *True Positive* (TP), keadaan dimana *classifier* memprediksi dengan benar kelas positif sebagai positif.
- b. *True Negative* (TN), keadaan dimana *classifier* memprediksi dengan benar kelas negatif sebagai negatif.
- c. *False Positive* (FP), keadaan dimana *classifier* salah, yaitu memprediksi kelas negatif sebagai positif.
- d. *False Negative* (FN), keadaan dimana *classifier* salah, yaitu memprediksi kelas positif sebagai negatif.

Pada Gambar 9 merupakan contoh ilustrasi *multi-class confusion matrix* ditinjau pada performa sistem dalam mengenali kelas B.

		PREDICTED			
		Classes	A	B	C
ACTUAL	A	TN	FP	TN	TN
	B	FN	TP	FN	FN
	C	TN	FP	TN	TN
	D	TN	FP	TN	TN

Gambar 9 *Multi Class Confusion Matrix*

Dalam *multi-class classification*, kelas-kelas dicantumkan dalam urutan yang sama dalam baris. Kelas-kelas yang diklasifikasikan dengan benar terletak pada diagonal utama dari kiri atas ke kanan bawah dan merupakan titik pertemuan antara prediksi dan aktual (Grandini et al., 2020). Metrik kinerja dari sebuah algoritma adalah *recall*, *precision*, *F1-Score*, dan *accuracy*, yang dihitung berdasarkan nilai dari TP, TN, FP, dan FN (Singh et al., 2021).

- a. *Recall*

*Recall* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang benar positif dan salah negatif. Data positif yang diklasifikasikan dengan benar dibagi dengan total data positif.

$$Recall = \frac{(TP)}{(TP+FN)} \times 100\% \quad (16)$$

b. *Precision*

*Precision* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif. Data positif yang diklasifikasikan dengan benar dibagi dengan total data yang diklasifikasikan sebagai data positif.

$$Precision = \frac{(TP)}{(TP+FP)} \times 100\% \quad (17)$$

c. *F1-Score*

*F1-Score* adalah nilai yang di dapat untuk menunjukkan performa algoritma yang diterapkan. *F1-Score* menunjukkan keseimbangan antara *recall* dan *precision*.

$$F1 - Score = 2X \frac{(Precision \times Recall)}{(Precision+Recall)} \times 100\% \quad (18)$$

d. *Accuracy*

*Accuracy* adalah rasio prediksi benar (positif) dengan keseluruhan data.

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \times 100\% \quad (19)$$

## 2.8 *GrabCut*

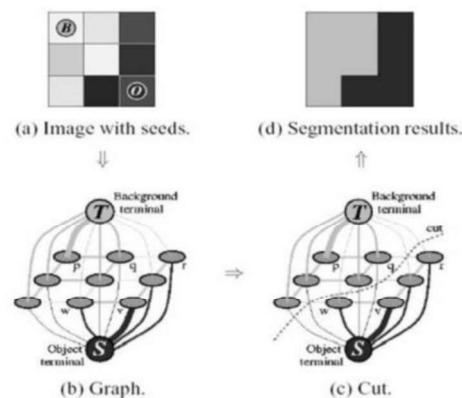
*Grabcut* yang dikembangkan oleh rother (2004) merupakan salah satu metode segmentasi citra yang didasarkan pada algoritma *graphcut*. *Grabcut* adalah suatu algoritma iteratif yang menggabungkan metode statistika dan algoritma *graphcu* dalam rangka untuk mencapai segmentasi 2D rinci dengan *input* yang terbatas. Metode ini awalnya dikembangkan *Microsoft Research* yang bertempat di Cambridge.

*Grabcut* menyempurnakan algoritma *graphcut*, *Grabcut* memungkinkan segmentasi dilakukan langsung pada citra berwarna. berbeda dengan *graphcut* yang didasarkan pada citra *grayscale*. informasi yang bertindak sebagai *input* metode *grabcut* adalah wilayah gambar persegi panjang yang dibuat oleh *user*, yang selanjutnya informasi ini akan dimasukkan pada trimap yang sesuai. Dengan menggunakan *Grabcut* dimungkinkan interaksi *user* menjadi minimal. *Grabcut* merupakan metode yang sangat efektif untuk teknik *foreground extraction*. secara

singkat tahapan metode *Grabcut* adalah sebagai berikut (*Open Computer Vision Module 2021*) :

1. *User* menciptakan trimap awal dengan membuat persegi panjang pada sekitar objek yang ditentukan sebagai *foreground* dari gambar. Pixel dalam trimap ditandai sebagai kelompok *trimap unknown* (tidak diketahui). Pixel diluar trimap ditandai sebagai pixel dalam kelompok trimap *background*.
2. Segmentasi awal dilaksanakan, *pixel unknown* ditempatkan pada kelas *foreground* sedangkan pixel *background* dikelompokkan pada kelas *background*.
3. GMM (*Gaussian mixture model*) dibuat untuk masing masing kelas (*foreground* dan *background*).
4. *Graph* dari antar pixel dibentuk dan dijalankan operasi *Graphcut* untuk mendapatkan hasil segmentasi antara *foreground* dan *background*.
5. Langkah 3-5 diulang hingga hasil segmentasi konvergen atau hingga didapat hasil yang diharapkan.

Hasil ilustrasi kerja dari *Grabcut* dapat dilihat pada Gambar 10



Gambar 10 Ilustrasi Kerja *Grabcut*

## 2.9 Pickle

Modul *Pickle* merupakan *library Python* yang mengimplementasikan protokol *biner* untuk membuat serialisasi dan de-serialisasi struktur objek. “*Pickling*” adalah proses di mana hierarki objek *Python* diubah menjadi aliran *byte*,

dan “*unpickling*” adalah operasi terbalik, di mana aliran *byte* (dari file *biner* atau objek seperti *byte*) diubah kembali menjadi hierarki objek. *Pickling* (dan *unpickling*) juga dikenal sebagai serialisasi, *marshalling*, atau *flattening*. Namun, untuk menghindari kebingungan, istilah yang digunakan di sini adalah “*pickling*” dan “*unpickling*”.

Format data yang digunakan oleh *Pickle* adalah format khusus *Python*. Hal ini memiliki keuntungan bahwa tidak ada batasan yang diberlakukan oleh standar eksternal seperti JSON atau XDR (yang tidak dapat mewakili berbagai *s*). Namun, hal ini berarti bahwa program *non-Python* mungkin tidak dapat merekonstruksi objek *Python* menggunakan *Pickle*. Secara default, format data *Pickle* menggunakan representasi *biner* (Chandra 2022).

## 2.10 *FastAPI*

*FastAPI* merupakan sebuah *web framework* modern yang memiliki performa tinggi (Gupta, 2021). Setelah dirilis pada tahun 2018, *fastAPI* memiliki performa yang tinggi mengalahkan *flask*. *FastAPI* dan *flask* memiliki fungsi untuk membangun sebuah *Application Programming Interface* (API). Pada dasarnya, *fastAPI* dibangun pada *Asynchronous Server Gateway Interface* (ASGI), sedangkan *flask* dibangun pada *Web Server Gateway Interface* (WSGI). Oleh karena itu, ASGI merupakan *server* yang digunakan sebagai penerus WSGI dikarenakan ASGI mampu bekerja hingga *throughput* tinggi yang tidak dapat ditangani oleh WSGI (Shin and Han 2021).

## 2.11 **Android**

Android adalah sistem operasi yang dibuat oleh Google yang awalnya ditujukan untuk perangkat telepon pintar dan tablet. Android bersifat *open source*, yang artinya adalah kode sumbernya terbuka dan dapat dikembangkan oleh siapapun. Google merilis kode sumber Android dibawah Lisensi *Apache*. Didukung oleh sifat *open source*-nya, Android berkembang sangat pesat dan dalam perkembangannya memiliki banyak komunitas pengembang aplikasi (Hasan Robbani et al., 2016).

Android adalah platform *open source* yang komprehensif dan dirancang untuk mobile devices. Dikatakan komprehensif karena Android menyediakan semua *tools* dan *frameworks* yang lengkap untuk pengembangan aplikasi pada suatu mobile *device*. Sistem Android menggunakan database untuk menyimpan informasi penting yang diperlukan agar tetap tersimpan meskipun device dimatikan. Untuk melakukan penyimpanan data pada *database*, sistem Android menggunakan SQLite yang merupakan suatu *open source database* yang cukup stabil dan banyak digunakan pada banyak *device* berukuran kecil (Silvia et al., 2014).

Android adalah sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, TMobile, dan Nvidia (Darnita et al., 2020)

## 2.12 Penelitian Terkait

Berikut ini merupakan beberapa penelitian terkait dengan penelitian yang dilakukan:

### 1. **Aplikasi Mobile untuk Deteksi dan Klasifikasi Daun Secara Real Time**

Penelitian ini menggunakan metode *hear like fitur* dan LBP untuk ekstraksi fitur dan SVM untuk klasifikasi. Dimana hasil yang didapatkan dalam proses deteksi dan pengenalan dengan menggunakan *mobile phone* berbasis android ialah mencapai akurasi 66.91%.

### 2. **Automatic Recognition of Medicinal Plants using Machine Learning Techniques**

Pada penelitian ini menggunakan metode *Random Forest*, *Multilayer Perceptron Neural Network*, *Support Vector Machine*, *Naïve Bayes*, dan *K-Nearest Neighbour*. Dimana pada penelitian ini menggunakan beberapa algoritma *machine learning* untuk mengklasifikasikan 24 spesies tanaman. Dan hasilnya, *Random Forest* memiliki tingkat akurasi paling tinggi yaitu 90.1% dibandingkan dengan *Multilayer Perceptron Neural Network*, *Support Vector Machine*, *Naïve Bayes*, dan *K-Nearest Neighbour*.

### 3. ***Pattern Recognition and Classification of Indian Herbal Leaves***

Kashif Akhtar dkk melakukan penelitian menggunakan metode SIFT untuk ekstraksi fitur dan *Multi Class SVM* dan *Probabilistic Neural Network* untuk klasifikasi. Dimana hasil yang didapatkan klasifikasi *Multi SVM* memiliki akurasi tertinggi 94% daripada PNN, dan sistem ini bekerja secara *real time* dengan baik.

### 4. **Identifikasi Tumbuhan Obat Herbal Berdasarkan Citra Daun Menggunakan Algoritma *Gray Level Co-occurrence Matrix* dan *K-Nearest Neighbor***

Penelitian ini menggunakan metode *Gray Level Cooccurence Matrix* untuk ekstraksi fitur dan metode *K-Nearest Neighbor* untuk klasifikasi. Hasil yang didapatkan ialah ada 10 spesies tumbuhan obat herbal yang digunakan dalam penelitian ini. Analisis tekstur yang digunakan adalah GLCM dengan mengekstrak nilai kontras, korelasi, energi dan homogenitas. Klasifikasi dilakukan dengan KNN. Hasil percobaan menunjukkan akurasi identifikasi menggunakan metode 9-fold cross *validation* mencapai 83.33% dengan menggunakan 9 *subset*.

### 5. **Implementasi *Deep Learning* untuk Klasifikasi Tanaman Toga Berdasarkan Citra Daun Berbasis Android**

Pada penelitian ini menggunakan metode *Convolutional Neural Network* (CNN). Dimana hasil yang diperoleh pada penelitian ini menggunakan 10 kelas jenis tanaman toga yaitu teh hijau, tapak dewa, sirsak, semanggi, mengkudu, mahoni, kumis kucing, jambu biji,

belimbing wuluh, bayam merah, Pengujian terhadap data pelatihan menghasilkan akurasi 75% dan data pengujian menghasilkan akurasi 80%.

#### **6. Klasifikasi Tanaman Obat-Obatan Berdasarkan Citra Daun dengan Menggunakan Jaringan Syaraf Tiruan**

Penelitian ini menggunakan *Gray Level Co-occurrence Matrix* (GLCM) untuk ekstraksi fitur dan Jaringan Syaraf Tiruan *Backpropagation* untuk klasifikasinya. Berdasarkan pengujian pengenalan pada keseluruhan citra, citra dapat dikenali jenisnya dan menghasilkan tingkat akurasi pengenalan sebesar 83,5% dengan hasil akurasi tertinggi dihasilkan oleh daun jarak dan mengkudu sebesar 100%, serta tingkat pengenalan terendah dihasilkan oleh daun sirih sebesar 60%.

#### **7. Perancangan Sistem Pengenalan Jenis Tanaman Obat dengan Kamera Berbasis Android**

Pada penelitian ini menggunakan metode *Convolutional Neural Network* (CNN). Hasil dari metode tersebut diperoleh 100% akurasi terhadap 6 jenis daun herbal yang telah dipotret diatas *background* putih.

#### **8. *Comparison of Machine Learning Algorithms for Detection of Medicinal Plants***

Penelitian ini menggunakan metode GLCM untuk ekstraksi fitur dan *K-Nearest Neighbours* dan *Support Vector Machine* untuk klasifikasinya. Penelitian ini membandingkan algoritma klasifikasi KNN dan SVM dengan menggunakan metode GLCM untuk fitur ekstraksi. Menghasilkan tingkat akurasi 100% pada KNN dan 93,23% pada SVM.