



**PENGENALAN TIPE PERMUKAAN JALAN
BERDASARKAN INFORMASI WARNA DAN TEKSTUR
MENGUNAKAN METODE KLASIFIKASI BERTINGKAT**

TESIS

Untuk memenuhi sebagian persyaratan
memperoleh gelar Magister Komputer

Disusun oleh:
Susi Marianingsih
NIM: 176150100111039



**PROGRAM STUDI MAGISTER ILMU KOMPUTER
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG**



DAFTAR ISI

PENGESAHAN	2
PERNYATAAN ORISINALITAS	Error! Bookmark not defined.
KATA PENGANTAR	Error! Bookmark not defined.
ABSTRAK	Error! Bookmark not defined.
DAFTAR ISI	2
DAFTAR GAMBAR	2
DAFTAR TABEL	9
BAB I PENDAHULUAN	Error! Bookmark not defined.
1.1 Latar belakang	Error! Bookmark not defined.
1.2 Rumusan masalah	Error! Bookmark not defined.
1.3 Tujuan penelitian	Error! Bookmark not defined.
1.4 Manfaat Penelitian	Error! Bookmark not defined.
1.5 Batasan Masalah	Error! Bookmark not defined.
1.6 Sistematika Pembahasan	Error! Bookmark not defined.
BAB II LANDASAN KEPUSTAKAAN	Error! Bookmark not defined.
2.1 <i>Review</i> penelitian sebelumnya	Error! Bookmark not defined.
2.1.1 Klasifikasi tipe permukaan jalan	Error! Bookmark not defined.
2.1.2 Kombinasi <i>KNN-NB</i> dan <i>KNN-SVM</i>	Error! Bookmark not defined.
2.1.3 Ekstraksi fitur warna dan tekstur	Error! Bookmark not defined.
2.2 Pengenalan Citra	Error! Bookmark not defined.
2.2.1 Citra analog	Error! Bookmark not defined.
2.2.2 Citra digital	Error! Bookmark not defined.
2.3 Warna	Error! Bookmark not defined.
2.3.1 Ruang warna	Error! Bookmark not defined.
2.3.1.1 Ruang warna <i>RGB</i>	Error! Bookmark not defined.
2.3.1.2 Ruang warna <i>HSV</i>	Error! Bookmark not defined.
2.3.2 Konversi ruang warna <i>RGB</i> ke <i>HSV</i>	Error! Bookmark not defined.
2.3.3 Ekstraksi fitur warna	Error! Bookmark not defined.
2.3 Tekstur	Error! Bookmark not defined.
2.3.1 Metode ekstraksi fitur tekstur	Error! Bookmark not defined.
2.3.2 Ekstraksi fitur tekstur menggunakan <i>GLCM</i>	Error! Bookmark not defined.
2.4 Metode Klasifikasi	Error! Bookmark not defined.
2.4.1 <i>K-Nearest Neighbors (KNN)</i>	Error! Bookmark not defined.
2.4.2 <i>Naïve Bayes (NB)</i>	Error! Bookmark not defined.
2.4.3 <i>Support Vector Machine (SVM)</i>	Error! Bookmark not defined.
2.5 Pengukuran Kinerja Sistem	Error! Bookmark not defined.
BAB III METODOLOGI	Error! Bookmark not defined.
3.1 Studi pustaka	Error! Bookmark not defined.
3.2 Pengumpulan data penelitian	Error! Bookmark not defined.
3.2.1 Pengambilan data permukaan jalan dari <i>Instant Google Street View</i>	Error! Bookmark not defined.



3.2.2 Pengambilan data permukaan jalan di wilayah kota Malang.....**Error!**
Bookmark not defined.

3.3 Metode ekstraksi fitur.....**Error! Bookmark not defined.**

3.3.1 Ekstraksi fitur warna.....**Error! Bookmark not defined.**

3.3.2 Ekstraksi fitur tekstur.....**Error! Bookmark not defined.**

3.4 Metode klasifikasi.....**Error! Bookmark not defined.**

3.4.1 Metode *Nearest Neighbors (NN)*.....**Error! Bookmark not defined.**

3.4.2 *K-Nearest Neighbors (KNN)*.....**Error! Bookmark not defined.**

3.4.3 *Naïve Bayes (NB)*.....**Error! Bookmark not defined.**

3.4.5 *Support Vector Machine (SVM)*.....**Error! Bookmark not defined.**

3.4.6 Kombinasi metode klasifikasi, nilai k , d , dan θ yang menghasilkan akurasi terbaik pada tiap tipe jalan.....**Error! Bookmark not defined.**

3.4.7 Metode klasifikasi bertingkat.....**Error! Bookmark not defined.**

3.5 Rencana pengujian.....**Error! Bookmark not defined.**

3.5.1 Pengujian sistem klasifikasi secara *offtime***Error! Bookmark not defined.**

3.5.2 Pengujian sistem klasifikasi secara *realtime***Error! Bookmark not defined.**

3.5.3 Pengujian metode perbandingan.....**Error! Bookmark not defined.**

3.5.3.1 Pengujian metode *NB-Histogram-GLCM***Error! Bookmark not defined.**

3.5.3.2 Pengujian *SVM-Histogram-DCT-GLCM***Error! Bookmark not defined.**

3.6 Hasil dan pembahasan.....**Error! Bookmark not defined.**

BAB IV HASIL DAN PEMBAHASAN.....**Error! Bookmark not defined.**

4.1 Hasil dan pembahasan ekstraksi fitur.....**Error! Bookmark not defined.**

4.1.1 Fitur Warna.....**Error! Bookmark not defined.**

4.1.2 Fitur Tekstur.....**Error! Bookmark not defined.**

4.2 Hasil dan pembahasan pengujian metode klasifikasi**Error! Bookmark not defined.**

4.2.1 Hasil pengujian nilai parameter C dan γ pada metode klasifikasi *SVM*.....**Error! Bookmark not defined.**

4.2.2 Hasil pengujian nilai k pada metode klasifikasi *KNB* dan *KSVM*.....**Error! Bookmark not defined.**

4.2.3 Pembahasan hasil pengujian nilai k pada metode klasifikasi *KNB* dan *KSVM*.....**Error! Bookmark not defined.**

4.2.3 Hasil dan pembahasan pengujian klasifikasi bertingkat secara *offtime*.....**Error! Bookmark not defined.**

4.2.4 Hasil dan pembahasan pengujian metode klasifikasi bertingkat secara *realtime*.....**Error! Bookmark not defined.**

4.3. Hasil dan pembahasan metode perbandingan**Error! Bookmark not defined.**

4.3.1 Perbandingan hasil pengujian klasifikasi bertingkat dengan metode perbandingan.....**Error! Bookmark not defined.**

BAB V PENUTUP.....**Error! Bookmark not defined.**

5.1 Kesimpulan.....**Error! Bookmark not defined.**



DAFTAR GAMBAR

Gambar 2.1 Contoh citra analog	Error! Bookmark not defined.
Gambar 2.2 Contoh citra biner (a), citra <i>grayscale</i> (b), dan citra warna (c)	Error! Bookmark not defined.
Gambar 2.3 Ruang warna RGB dalam bentuk koordinat (a) dan 24-bit (b)	Error! Bookmark not defined.
Gambar 2.4 Ruang warna <i>HSV</i>	Error! Bookmark not defined.
Gambar 2.5 Arah sudut dan jarak dari pasangan piksel	Error! Bookmark not defined.
Gambar 2.6 Proses pembentukan matriks <i>GLCM</i>	Error! Bookmark not defined.
Gambar 2.7 Garis hitam, merah dan biru adalah beberapa alternatif garis pemisah (<i>hyperplane</i>)	Error! Bookmark not defined.
Gambar 2.8 Garis hitam adalah <i>hyperplane</i> terbaik dan titik-titik data yang dilingkari adalah <i>support vector</i>	Error! Bookmark not defined.
Gambar 2.9 Memetakan data input ke ruang fitur menggunakan fungsi kernel	Error! Bookmark not defined.
Gambar 3.1 Blok diagram tahapan proses penelitian	Error! Bookmark not defined.
Gambar 3.2 Contoh gambar jalan dari Instant Google Street View	Error! Bookmark not defined.
Gambar 3.3 Pengambilan potongan-potongan gambar dari area jalan	Error! Bookmark not defined.
Gambar 3.4 Contoh <i>data set</i> dari tiap tipe permukaan jalan	Error! Bookmark not defined.
Gambar 3.5 <i>Flowchart</i> pengambilan data area permukaan jalan	Error! Bookmark not defined.
Gambar 3.6 Perangkat untuk pengambilan data permukaan jalan	Error! Bookmark not defined.
Gambar 3.7 Contoh frame ukuran 960x540 dari video jalan aspal	Error! Bookmark not defined.
Gambar 3.8 <i>Flowchart</i> ekstraksi fitur warna	Error! Bookmark not defined.
Gambar 3.9 <i>Flowchart</i> ekstraksi fitur tekstur	Error! Bookmark not defined.
Gambar 3.10 <i>Flowchart</i> ekstraksi fitur tekstur <i>data set</i>	Error! Bookmark not defined.
Gambar 3.11 <i>Flowchart</i> metode NN	Error! Bookmark not defined.
Gambar 3.12 <i>Flowchart</i> metode KNN	Error! Bookmark not defined.
Gambar 3.13 <i>Flowchart</i> metode klasifikasi NB	Error! Bookmark not defined.
Gambar 3.14 <i>Flowchart</i> untuk mencari kombinasi metode klasifikasi, nilai <i>k</i>, <i>d</i>, dan θ dengan akurasi terbaik pada tiap tipe jalan	Error! Bookmark not defined.



[Gambar 3.15 Flowchart dari metode klasifikasi bertingkat](#) **Error! Bookmark not defined.**

[Gambar 3.16 Cropping pada area permukaan jalan](#) **Error! Bookmark not defined.**

[Gambar 4.1 Perubahan nilai fitur contrast tipe aspal berdasarkan nilai \$d\$](#) **Error! Bookmark not defined.**

[Gambar 4.2 Perubahan nilai fitur *contrast* tipe aspal berdasarkan nilai \$\theta\$](#) **Error! Bookmark not defined.**

[Gambar 4.3 Perubahan nilai fitur *dissimilarity* tipe aspal berdasarkan nilai \$d\$](#) **Error! Bookmark not defined.**

[Gambar 4.4 Perubahan nilai fitur *dissimilarity* tipe aspal berdasarkan nilai \$\theta\$](#) **Error! Bookmark not defined.**

[Gambar 4.5 Perubahan nilai fitur *homogeneity* tipe aspal berdasarkan nilai \$d\$](#) **Error! Bookmark not defined.**

[Gambar 4.6 Perubahan nilai fitur *homogeneity* tipe aspal berdasarkan nilai \$\theta\$](#) **Error! Bookmark not defined.**

[Gambar 4.7 Perubahan nilai fitur ASM tipe aspal berdasarkan nilai \$d\$](#) **Error! Bookmark not defined.**

[Gambar 4.8 Perubahan nilai fitur ASM tipe aspal berdasarkan nilai \$\theta\$](#) **Error! Bookmark not defined.**

[Gambar 4.9 Perubahan nilai fitur *energy* tipe aspal berdasarkan nilai \$d\$](#) **Error! Bookmark not defined.**

[Gambar 4.10 Perubahan nilai fitur *energy* tipe aspal berdasarkan nilai \$\theta\$](#) **Error! Bookmark not defined.**

[Gambar 4.11 Perubahan nilai fitur *correlation* tipe aspal berdasarkan nilai \$d\$](#) **Error! Bookmark not defined.**

[Gambar 4.12 Perubahan nilai fitur *correlation* tipe aspal berdasarkan nilai \$\theta\$](#) **Error! Bookmark not defined.**

[Gambar 4.13 Perubahan nilai fitur *contrast* tipe aspal rusak berdasarkan nilai \$d\$](#) **Error! Bookmark not defined.**

[Gambar 4.14 Perubahan nilai fitur *contrast* tipe aspal rusak berdasarkan nilai \$\theta\$](#) **Error! Bookmark not defined.**

[Gambar 4.15 Perubahan nilai fitur *dissimilarity* tipe aspal rusak berdasarkan nilai \$d\$](#) **Error! Bookmark not defined.**

[Gambar 4.16 Perubahan nilai fitur *dissimilarity* tipe aspal rusak berdasarkan nilai \$\theta\$](#) **Error! Bookmark not defined.**

[Gambar 4.17 Perubahan nilai fitur *homogeneity* tipe aspal rusak berdasarkan nilai \$d\$](#) **Error! Bookmark not defined.**

[Gambar 4.18 Perubahan nilai fitur *homogeneity* tipe aspal rusak berdasarkan nilai \$\theta\$](#) **Error! Bookmark not defined.**

[Gambar 4.19 Perubahan nilai fitur ASM tipe aspal rusak berdasarkan nilai \$d\$](#) **Error! Bookmark not defined.**



[Gambar 4.20 Perubahan nilai fitur *ASM* tipe aspal rusak berdasarkan nilai \$\theta\$](#) **Error!**

Bookmark not defined.

[Gambar 4.21 Perubahan nilai fitur *energy* tipe aspal rusak berdasarkan nilai \$d\$](#)

Error! Bookmark not defined.

[Gambar 4.22 Perubahan nilai fitur *energy* tipe aspal rusak berdasarkan nilai \$\theta\$](#)

Error! Bookmark not defined.

[Gambar 4.23 Perubahan nilai fitur *correlation* tipe aspal rusak berdasarkan nilai \$d\$](#)

Error! Bookmark not defined.

[Gambar 4.24 Perubahan nilai fitur *correlation* tipe aspal rusak berdasarkan nilai \$\theta\$](#)

Error! Bookmark not defined.

[Gambar 4.25 Perubahan nilai fitur *contrast* tipe tanah berdasarkan nilai \$d\$...](#) **Error!**

Bookmark not defined.

[Gambar 4.26 Perubahan nilai fitur *contrast* tipe tanah berdasarkan nilai \$\theta\$...](#) **Error!**

Bookmark not defined.

[Gambar 4.27 Perubahan nilai fitur *dissimilarity* tipe tanah berdasarkan nilai \$d\$](#)

Error! Bookmark not defined.

[Gambar 4.28 Perubahan nilai fitur *dissimilarity* tipe tanah berdasarkan nilai \$\theta\$](#)

Error! Bookmark not defined.

[Gambar 4.29 Perubahan nilai fitur *homogeneity* tipe tanah berdasarkan nilai \$d\$](#)

Error! Bookmark not defined.

[Gambar 4.30 Perubahan nilai fitur *homogeneity* tipe tanah berdasarkan nilai \$\theta\$](#)

Error! Bookmark not defined.

[Gambar 4.31 Perubahan nilai fitur *ASM* tipe tanah berdasarkan nilai \$d\$](#) **Error!**

Bookmark not defined.

[Gambar 4.32 Perubahan nilai fitur *ASM* tipe tanah berdasarkan nilai \$\theta\$](#) **Error!**

Bookmark not defined.

[Gambar 4.33 Perubahan nilai fitur *energy* tipe tanah berdasarkan nilai \$d\$](#) **Error!**

Bookmark not defined.

[Gambar 4.34 Perubahan nilai fitur *energy* tipe tanah berdasarkan nilai \$\theta\$](#) **Error!**

Bookmark not defined.

[Gambar 4.35 Perubahan nilai fitur *correlation* tipe tanah berdasarkan nilai \$d\$](#)

Error! Bookmark not defined.

[Gambar 4.36 Perubahan nilai fitur *correlation* tipe tanah berdasarkan nilai \$\theta\$](#)

Error! Bookmark not defined.

[Gambar 4.37 Perubahan nilai fitur *contrast* tipe paving berdasarkan nilai \$d\$..](#) **Error!**

Bookmark not defined.

[Gambar 4.38 Perubahan nilai fitur *contrast* tipe paving berdasarkan nilai \$\theta\$..](#) **Error!**

Bookmark not defined.

[Gambar 4.39 Perubahan nilai fitur *dissimilarity* tipe paving berdasarkan nilai \$d\$](#)

Error! Bookmark not defined.

[Gambar 4.40 Perubahan nilai fitur *dissimilarity* tipe paving berdasarkan nilai \$\theta\$](#)

Error! Bookmark not defined.



[Gambar 4.41 Perubahan nilai fitur *homogeneity* tipe paving berdasarkan nilai \$d\$](#)

..... **Error! Bookmark not defined.**

[Gambar 4.42 Perubahan nilai fitur *homogeneity* tipe paving berdasarkan nilai \$\theta\$](#)

..... **Error! Bookmark not defined.**

[Gambar 4.43 Perubahan nilai fitur *ASM* tipe paving berdasarkan nilai \$d\$ ](#) **Error!**

Bookmark not defined.

[Gambar 4.44 Perubahan nilai fitur *ASM* tipe paving berdasarkan nilai \$\theta\$ ](#) **Error!**

Bookmark not defined.

[Gambar 4.45 Perubahan nilai fitur *energy* tipe paving berdasarkan nilai \$d\$ ](#) **Error!**

Bookmark not defined.

[Gambar 4.46 Perubahan nilai fitur *energy* tipe paving berdasarkan nilai \$\theta\$ ](#) **Error!**

Bookmark not defined.

[Gambar 4.47 Perubahan nilai fitur *correlation* tipe paving berdasarkan nilai \$d\$](#)

..... **Error! Bookmark not defined.**

[Gambar 4.48 Perubahan nilai fitur *correlation* tipe paving berdasarkan nilai \$\theta\$](#)

..... **Error! Bookmark not defined.**

[Gambar 4.49 Flowchart dari metode klasifikasi bertingkat](#) **Error! Bookmark not defined.**

[Gambar 4.50 Akurasi pengujian realtime metode klasifikasi bertingkat pada ukuran frame 1920x1080](#)..... **Error! Bookmark not defined.**

[Gambar 4.51 Akurasi pengujian realtime metode klasifikasi bertingkat di berbagai ukuran frame uji](#)..... **Error! Bookmark not defined.**

[Gambar 4.52 Perbandingan akurasi metode klasifikasi bertingkat dengan metode pembandingan](#)..... **Error! Bookmark not defined.**

[Gambar 4.53 Perbandingan waktu komputasi rata-rata per frame metode klasifikasi bertingkat dengan metode pembandingan](#).. **Error! Bookmark not defined.**



DAFTAR TABEL

Tabel 2.1 Confusion matrix	Error! Bookmark not defined.
Tabel 4.1 Contoh fitur warna 3 channel HSV untuk tiap kelas	Error! Bookmark not defined.
Tabel 4.2 Contoh fitur tekstur tipe aspal	Error! Bookmark not defined.
Tabel 4.3 Nilai fitur <i>contrast</i> tipe aspal	Error! Bookmark not defined.
Tabel 4.4 Nilai fitur <i>dissimilarity</i> tipe aspal	Error! Bookmark not defined.
Tabel 4.5 Nilai fitur <i>homogeneity</i> tipe aspal	Error! Bookmark not defined.
Tabel 4.6 Nilai fitur ASM tipe aspal	Error! Bookmark not defined.
Tabel 4.7 Nilai fitur <i>energy</i> tipe aspal	Error! Bookmark not defined.
Tabel 4.8 Nilai fitur <i>correlation</i> tipe aspal	Error! Bookmark not defined.
Tabel 4.9 Contoh fitur tekstur tipe aspal rusak	Error! Bookmark not defined.
Tabel 4.10 Nilai fitur <i>contrast</i> tipe aspal rusak	Error! Bookmark not defined.
Tabel 4.11 Nilai fitur <i>dissimilarity</i> tipe aspal rusak	Error! Bookmark not defined.
Tabel 4.12 Nilai fitur <i>homogeneity</i> tipe aspal rusak	Error! Bookmark not defined.
Tabel 4.13 Nilai fitur ASM tipe aspal rusak	Error! Bookmark not defined.
Tabel 4.14 Nilai fitur <i>energy</i> tipe aspal rusak	Error! Bookmark not defined.
Tabel 4.15 Nilai fitur <i>correlation</i> tipe aspal rusak	Error! Bookmark not defined.
Tabel 4.16 Contoh fitur tekstur tipe tanah	Error! Bookmark not defined.
Tabel 4.17 Nilai fitur <i>contrast</i> tipe tanah	Error! Bookmark not defined.
Tabel 4.18 Nilai fitur <i>dissimilarity</i> tipe tanah	Error! Bookmark not defined.
Tabel 4.19 Nilai fitur <i>homogeneity</i> tipe tanah	Error! Bookmark not defined.
Tabel 4.20 Nilai fitur ASM tipe tanah	Error! Bookmark not defined.
Tabel 4.21 Nilai fitur <i>energy</i> tipe tanah	Error! Bookmark not defined.
Tabel 4.22 Nilai fitur <i>correlation</i> tipe tanah	Error! Bookmark not defined.
Tabel 4.23 Contoh fitur tekstur tipe paving	Error! Bookmark not defined.
Tabel 4.24 Nilai fitur <i>contrast</i> tipe paving	Error! Bookmark not defined.
Tabel 4.25 Nilai fitur <i>dissimilarity</i> tipe paving	Error! Bookmark not defined.
Tabel 4.26 Nilai fitur <i>homogeneity</i> tipe paving	Error! Bookmark not defined.
Tabel 4.27 Nilai fitur ASM tipe paving	Error! Bookmark not defined.
Tabel 4.28 Nilai fitur <i>energy</i> tipe paving	Error! Bookmark not defined.
Tabel 4.29 Nilai fitur <i>correlation</i> tipe paving	Error! Bookmark not defined.
Tabel 4.30 Akurasi (%) metode SVM pada pengujian C dan y	Error! Bookmark not defined.
Tabel 4.31 Akurasi metode klasifikasi KNB dengan $k=300$	Error! Bookmark not defined.
Tabel 4.32 Hasil pengujian metode klasifikasi KNB dengan $k=500$	Error! Bookmark not defined.
Tabel 4.33 Hasil pengujian klasifikasi KNB dengan $k=750$	Error! Bookmark not defined.
Tabel 4.34 Hasil pengujian klasifikasi KNB dengan $k=1000$	Error! Bookmark not defined.



[Tabel 4.35 Hasil pengujian klasifikasi KSVM dengan \$k=300\$](#) **Error! Bookmark not defined.**

[Tabel 4.36 Hasil pengujian klasifikasi KSVM dengan \$k=500\$](#) **Error! Bookmark not defined.**

[Tabel 4.37 Hasil pengujian klasifikasi KSVM dengan \$k=750\$](#) **Error! Bookmark not defined.**

[Tabel 4.38 Hasil pengujian klasifikasi KSVM dengan \$k=1000\$](#) **Error! Bookmark not defined.**

[Tabel 4.39 Akurasi metode klasifikasi dengan kombinasi \$k, d,\$ dan \$\theta\$](#)**Error! Bookmark not defined.**

[Tabel 4.40 Kombinasi \$k, d, \theta,\$ dan metode klasifikasi yang menghasilkan akurasi terbaik untuk tiap kelas](#).....**Error! Bookmark not defined.**

[Tabel 4.41 Confusion matrix klasifikasi bertingkat secara offtime](#)**Error! Bookmark not defined.**

[Tabel 4.42 Perbandingan nilai fitur GLCM tipe aspal rusak, tanah, dan paving](#).....**Error! Bookmark not defined.**

[Tabel 4.43 Hasil pengujian \(% akurasi\) realtime klasifikasi bertingkat](#).....**Error! Bookmark not defined.**

[Tabel 4.44 Pembahasan hasil pengujian realtime tipe aspal](#)**Error! Bookmark not defined.**

[Tabel 4.45 Pembahasan hasil pengujian realtime tipe paving](#)**Error! Bookmark not defined.**

[Tabel 4.46 Pembahasan hasil pengujian realtime tipe paving](#)**Error! Bookmark not defined.**

[Tabel 4.47 Perbandingan metode klasifikasi bertingkat dengan metode pembandingan](#).....**Error! Bookmark not defined.**

[Tabel 4.48 Perbandingan hasil pengujian realtime dengan metode pembandingan](#).....**Error! Bookmark not defined.**



BAB I PENDAHULUAN

1.1 Latar belakang

Global Status Report on Safety (WHO, 2015) melaporkan bahwa di seluruh dunia terdapat lebih dari 1,25 juta orang meninggal akibat kecelakaan lalu lintas dan 50 juta orang luka berat. Lebih dari 90% kecelakaan lalu lintas terjadi di negara berkembang yang jumlah kendaraannya hanya 54% dari jumlah kendaraan yang terdaftar di dunia. Hal ini membuat keselamatan berkendara menjadi perhatian dunia. Pada tahun 2010 PBB mendeklarasikan *Decade of Action (DoA) for Road Safety 2011–2020* yang bertujuan mengendalikan tingkat fatalitas kecelakaan lalu lintas. Indonesia adalah salah satu negara berkembang, dan kecelakaan lalu lintas di jalan raya pada tahun 2012–2016 sangat memprihatinkan, di mana 70% terjadi karena faktor pengemudi dan faktor kendaraan, sedangkan 30% disebabkan karena kondisi permukaan jalan yang sudah rusak (Statistik Perhubungan Buku 1, 2016). Sejalan dengan DoA yang dideklarasikan PBB, maka melalui Bappenas, Pemerintah Indonesia merumuskan sebuah program yaitu Rencana Umum Nasional Keselamatan (RUNK) Jalan 2011–2035 yang merupakan panduan bagi berbagai kementerian, Kepolisian Republik Indonesia bersama dengan masyarakat dan dunia usaha, untuk memastikan bahwa program-program kerjanya mengutamakan keselamatan dan mensinergikan semua potensi yang ada. Salah satu pilar penting dalam program RUNK Jalan 2011–2035 adalah meningkatkan riset-riset tentang kendaraan berkeselamatan.

Review singkat di atas memperlihatkan bahwa keselamatan di jalan raya menjadi perhatian serius. Faktor-faktor pendukung keselamatan di jalan raya seperti pengemudi, kelengkapan kendaraan, serta kondisi dan tipe jalan memiliki peran yang sangat penting. Peningkatan jumlah produksi kendaraan di seluruh dunia (*World Production, 2018*) tidak didukung oleh ketersediaan perangkat kendaraan bermotor untuk mendeteksi kondisi dan tipe permukaan jalan dalam mengantisipasi dan meminimalisir kecelakaan (Jeong dkk., 2017). Oleh karena itu diperlukan sistem dan perangkat pada kendaraan yang dapat mendeteksi dan memberikan informasi kepada pengemudi tentang tipe permukaan jalan yang akan dilalui sehingga kecepatan kendaraan dapat disesuaikan.

Salah satu perangkat yang telah dipakai oleh beberapa peneliti adalah penggunaan sensor deteksi permukaan jalan. Gailius dan Jacenas (2007) menggunakan *transducers, amplifier* dan *Analog/Digital (A/D) Converter* untuk menganalisis permukaan jalan. Alonso dkk. (2014) menggunakan mikrofon, *signal conditioning* dan *A/D converter* untuk merekam kebisingan antara ban kendaraan dan jalan yang dilalui. Wang dkk. (2008) menggunakan *roadside sensor technology* dan *floating car technology* untuk mengumpulkan berbagai informasi tentang kondisi jalan dan cuaca. Namun sayangnya, penggunaan perangkat-perangkat tersebut di atas hanya dapat memperoleh informasi dari suatu bagian jalan tertentu, yaitu bagian jalan yang telah dilewati kendaraan dan pemeliharaan dari sensor sangat mahal.

Beberapa peneliti menggunakan data yang diambil dengan menggunakan kamera dan memanfaatkan teknologi pengolahan citra untuk mengolah data gambar sehingga tipe permukaan jalan dapat dikenali. Popescu dkk. (2008) menggunakan fitur tekstur untuk mengklasifikasi permukaan jalan sedangkan Sun dan Jia (2013) dan Hadjidemetriou (2016) menggunakan informasi warna dan tekstur. Fitur warna, tekstur dan tepi digunakan oleh Tang dkk. (2011) untuk mengenali area jalan dan bukan jalan. Slimane dkk. (2008) menganalisis spektrum gambar refleksi permukaan jalan dan menggambarkan



korelasi antara tekstur permukaan dan koefisien gesekan, untuk menentukan kekasaran permukaan jalan. Yamamoto dkk. (2005) menerapkan metode interaksi manusia-komputer untuk mengekstrak nilai *grayscale* dan parameter karakteristik suhu permukaan jalan untuk memprediksi kondisi permukaan jalan. Muneo dkk. (2005) menggunakan kamera kendaraan untuk mengumpulkan informasi dari permukaan jalan dan menggunakan parameter karakteristik polarisasi citra permukaan jalan untuk membentuk model evaluasi kondisi jalan yang licin. Zhao dkk. (2017) menggunakan metode *SVM-PSO* untuk mengklasifikasi kondisi permukaan jalan (basah, kering, salju, es, dan tergenang air) berdasarkan fitur warna dan tekstur. Namun sayangnya, penelitian-penelitian ini hanya fokus pada pengaruh kondisi dan tekstur jalan yang dipengaruhi oleh cuaca (kering, basah, berair, bersalju dan jalan yang dipenuhi es) dengan objek jalan yang baik. Penelitian ini diajukan untuk mengenali tipe permukaan jalan itu sendiri tanpa dipengaruhi oleh kondisi cuaca.

Data permukaan jalan yang digunakan bersumber dari kamera, oleh karena itu akan menggunakan fitur warna dan tekstur untuk mendeskripsikan tipe jalan. Metode untuk mendeskripsikan fitur warna antara lain *histogram* warna (Liu dan Yang, 2013), *color correlogram* (Huang dkk., 1997), *color moment* (You dan Zhang, 2010), dan *color coherence vector* (Pass dkk., 1996). *Color moment* (*mean* dan *standard deviation*) mudah untuk diimplementasikan (You dan Zhang, 2010) dan memiliki sifat *invariant* terhadap translasi dan rotasi (Jeong dkk., 2004), namun *color moment* masih memiliki kelemahan yaitu ketika terdapat 2 gambar berbeda maka menghasilkan nilai yang sama jika objek memiliki distribusi warna yang sama. Kajian tentang ekstraksi fitur tekstur dengan menggunakan berbagai metode juga telah dilakukan oleh banyak peneliti. *Metode statistik seperti Gray-Level Co-occurrence Matrix (GLCM)*, *metode model-based seperti model Fractal, dan Markov (Mihran Tuceryan dan Anil K. Jain, 1998; Zhang dan Tan, 2002)*, metode pendekatan spectral seperti: filter *Gabor* (Younesi dan Amirani, 2017) dan *wavelet transform* (Vimala dan Priya, 2019) adalah metode-metode ekstraksi fitur yang banyak dikembangkan oleh peneliti. Metode *Gray-Level Co-occurrence Matrix (GLCM)* digunakan oleh Öztürk and Akdemir (2018), Karimah dan Harjoko (2017), Palm (2004) dan Alsadegh dan Joan (2015), yang membuktikan bahwa jarak (d) dan sudut (θ) pasangan piksel pada matriks GLCM mempengaruhi nilai fitur tekstur. Penelitian ini akan menggunakan *standard deviation* tiap *channel* pada ruang warna *HSV* untuk mendeskripsikan warna dan metode *GLCM* untuk mendeskripsikan fitur tekstur dan meneliti pengaruh jarak (d) dan sudut (θ) antara pasangan piksel terhadap nilai fitur yang dihasilkan.

Kombinasi *K-Nearest Neighbor (KNN)* dan metode klasifikasi yang lain (Naive Bayes dan Support Vector Machine) banyak digunakan peneliti dengan tujuan meningkatkan akurasi. Gabungan *KNN* dan *SVM* untuk mengenali nomor polisi negara Iran diusulkan oleh Tabrizi dan Cavus (2016) menghasilkan akurasi 97,03%, lebih baik dari pada metode *SVM* (94%). Hibrid dari *SVM-KNN* juga digunakan Machhale dkk. (2015) untuk mengklasifikasi data *MRI* otak, mampu menghasilkan akurasi (98%) yang lebih baik dari metode *SVM* (akurasi 96%). Calle-Alonso dkk. (2013) menggunakan gabungan *Bayesian Regresion* dan *KNN* untuk mengklasifikasi data biomedik dan menghasilkan rata-rata akurasi 77%, lebih baik dari metode Naive Bayes (rata-rata akurasi 62%). *Local Naive Bayes Nearest Neighbors* (Sancho McCann dan David G. Lowe, 2012) digunakan untuk mengklasifikasi gambar dan menghasilkan akurasi yang lebih baik dari pada *Naive Bayes Nearest Neighbors*. Ferdousy dkk. (2013) menggabungkan algoritme berbasis jarak *KNN* untuk mengklasifikasi atribut numerik, dan *Naive Bayes* untuk mengklasifikasi atribut kategori. Selain itu, Lee (2015) menggunakan hibrida *KNN* dan *NB* untuk pengenalan emosi bicara, menunjukkan bahwa gabungan *KNN* dan *NB* menghasilkan akurasi yang lebih baik dari metode tunggalnya (seperti *NB* atau *KNN*). Dari studi ini, menunjukkan bahwa gabungan *KNN* dan *NB* atau *SVM* cukup akurat dalam



memecahkan masalah klasifikasi. Penelitian ini mengusulkan metode klasifikasi tipe permukaan jalan menggunakan gabungan metode *KNN* dan *NB* atau *SVM* dengan menggunakan pendekatan bertingkat.

Metode Histogram-GLCM-NB (Sun and Jia, 2013) dan Histogram-GLCM-DCT-SVM (Hadjidemetriou et al., 2016) akan digunakan sebagai metode pembandingan. Metode pembandingan tersebut akan dilatih dan diuji dengan menggunakan data set dan perangkat pengujian dari penelitian yang diusulkan.

1.2 Rumusan masalah

Proses klasifikasi objek sebuah gambar dapat ditentukan berdasarkan fitur warna, akan tetapi keakuratan hasil klasifikasinya masih kurang sehingga diperlukan fitur tekstur. Fitur tekstur dengan menggunakan metode *GLCM* dipengaruhi oleh dua parameter yaitu jarak (d) dan sudut (θ) pasangan piksel dalam sebuah gambar. Penggunaan hanya satu metode klasifikasi menghasilkan akurasi yang kurang signifikan. Gabungan metode *KNN* dan metode *NB* atau *SVM* adalah salah satu usulan solusi untuk meningkatkan akurasi klasifikasi. Selain itu akurasi klasifikasi akan meningkat juga dengan melakukan klasifikasi bertingkat terhadap warna dan tekstur. Solusi lain yang akan dilakukan adalah dengan memasukkan variasi nilai k (jumlah data terdekat dengan data uji) pada metode *KNN-NB* (*KNB*) atau *KNN-SVM* (*KSVM*). Oleh karena itu, melalui penelitian ini terdapat beberapa permasalahan yang perlu dikaji dan diungkapkan, antara lain:

1. Bagaimana pengaruh variasi nilai jarak (d) dan sudut (θ) dari pasangan piksel terhadap nilai fitur tekstur *GLCM* (*contrast*, *dissimilarity*, *homogeneity*, *ASM*, *energy*, dan *correlation*) pada tipe jalan aspal, aspal rusak, tanah, dan paving?
2. Bagaimana kombinasi nilai k , jarak (d), dan sudut (θ) yang menghasilkan akurasi terbaik pada klasifikasi tipe permukaan jalan menggunakan metode *KNB* dan *KSVM*?
3. Bagaimana nilai akurasi dari metode klasifikasi bertingkat dalam mengenali tipe permukaan jalan berdasarkan informasi warna dan tekstur dalam pengujian secara *offtime*?
4. Bagaimana nilai akurasi dan waktu komputasi metode klasifikasi bertingkat pada pengujian *realtime* dengan ukuran *frame* uji yang bervariasi?
5. Bagaimana perbandingan akurasi dan waktu komputasi dari metode klasifikasi bertingkat dan metode pembandingan (metode *histogram-GLCM-Naïve Bayes* (Sun dan Jia, 2013) dan metode *histogram-DCT-GLCM-SVM* (Hadjidemetriou dkk., 2016)?)

1.3 Tujuan penelitian

Tujuan yang ingin dicapai dalam penelitian ini adalah:

1. Untuk mengetahui pengaruh variasi nilai jarak (d) dan sudut (θ) dari pasangan piksel terhadap nilai fitur tekstur menggunakan *GLCM*.
2. Untuk menghasilkan kombinasi nilai k , jarak (d), dan sudut (θ) yang menghasilkan akurasi terbaik pada klasifikasi tipe permukaan jalan menggunakan metode *KNB* dan metode *KSVM*.
3. Untuk mengetahui nilai akurasi dari metode klasifikasi bertingkat dalam mengenali tipe permukaan jalan berdasarkan informasi warna dan tekstur pada pengujian *offtime*.



4. Untuk mengetahui nilai akurasi dan waktu komputasi metode klasifikasi bertingkat pada pengujian *realtime* dengan ukuran *frame* uji yang bervariasi.
5. Untuk mengetahui perbandingan akurasi dan waktu komputasi dari metode klasifikasi bertingkat dengan metode pembandingan.

1.4 Manfaat Penelitian

Penelitian ini diharapkan memiliki manfaat sebagai berikut:

1. Memberikan pengetahuan bahwa nilai fitur tekstur GLCM dipengaruhi oleh nilai jarak (d) dan sudut (θ) dari pasangan piksel pada gambar.
2. Memberikan pengetahuan kombinasi nilai k , jarak (d) dan sudut (θ) yang menghasilkan akurasi terbaik pada klasifikasi tipe permukaan jalan menggunakan metode *KNB* dan metode *K SVM*.
3. Memberikan pengetahuan tentang nilai akurasi metode klasifikasi bertingkat pada pengujian *offtime*.
4. Memberikan pengetahuan tentang nilai akurasi dan waktu komputasi metode klasifikasi bertingkat pada pengujian *realtime*.
5. Mengetahui perbandingan akurasi dan waktu komputasi dari metode klasifikasi bertingkat dengan metode *histogram-GLCM-Naïve Bayes* dan *histogram-DCT-GLCM-SVM*.
6. Memberikan kontribusi terhadap sistem pengenalan tipe permukaan jalan dengan menggunakan metode klasifikasi bertingkat.

1.5 Batasan Masalah

Agar arah penelitian ini tetap fokus, maka pembahasan penelitian dibatasi pada:

1. Penelitian ini fokus pada klasifikasi dan pengenalan tipe permukaan jalan.
2. Data yang digunakan pada penelitian ini terdiri dari 4 tipe permukaan jalan (aspal, aspal rusak, tanah, dan berpaving).
3. Data gambar diambil dari *Instant Google Street View* ("<https://www.instantstreetview.com/>," 2018) dan jalan-jalan di wilayah sekitar Kota Malang yang diambil dengan menggunakan kamera *HP Samsung J4+* dan *stabilizer HP* dengan kondisi pencahayaan baik (waktu siang hari).
4. Pada proses pelatihan variasi nilai d adalah (1, 2, 3, 4), θ adalah (0° , 45° , 90° , 135°) dan k adalah (300, 500, 750, 1000).
5. Menggunakan *data set* yang terdiri dari 2000 gambar permukaan jalan (500 gambar untuk tiap kelas). Proses penyusunan *data set* dijelaskan pada sub bab 3.2.
6. Menggunakan fitur warna pada ruang warna *HSV* dan 6 fitur tekstur GLCM (*contrast*, *dissimilarity*, *homogeneity*, *ASM*, *energy*, dan *correlation*).
7. Metode pembandingan ada 2: metode *histogram-GLCM-Naïve Bayes* (Sun dan Jia, 2013) dan metode *histogram-DCT-GLCM-SVM* (Hadjidemetriou dkk., 2016)



1.6 Sistematika Pembahasan

Sistematika pembahasan penelitian ini disusun sebagai berikut:

BAB I Pendahuluan, berisi gambaran umum dari penelitian yang meliputi: latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika pembahasan.

BAB II Landasan Kepustakaan, menguraikan tentang dasar teori dan kajian pustaka yang berkaitan dengan domain permasalahan, penelitian-penelitian terkait sebelumnya, metode ekstraksi fitur, metode klasifikasi tipe permukaan jalan dan metode klasifikasi *KNB* dan *K SVM* dengan pendekatan bertingkat.

BAB III Metodologi, berisi tentang metode yang digunakan pada penelitian ini, yang meliputi: pengambilan data permukaan jalan, metode ekstraksi fitur, metode klasifikasi yang diusulkan, dan rencana pengujian metode.

BAB IV Hasil dan Pembahasan, berisi hasil pengujian dari metode yang diusulkan dan metode perbandingan dan pembahasan hasil pengujian sistem.

BAB V Penutup, berisi kesimpulan dari hasil dan pembahasan pengujian sistem dan potensi penelitian berikutnya.



BAB II LANDASAN KEPUSTAKAAN

2.1 Review penelitian sebelumnya

2.1.1 Klasifikasi tipe permukaan jalan

Penggunaan sensor deteksi tipe permukaan jalan digunakan di beberapa penelitian. Tipe permukaan jalan dapat dikategorikan dalam kondisi kering, basah, air, salju, dan es. Gailius dan Jacenas (2007) menggunakan *transducers*, *amplifier* dan *A/D converter* untuk mengukur kebisingan gesekan antara ban dan permukaan jalan dan memperoleh karakteristik permukaan jalan berdasarkan metode analisis spektrum kebisingan. Perangkat dipasang pada kendaraan uji untuk mendapatkan data kebisingan yang berupa sinyal *power spectral densities (PSD)* pada kecepatan kendaraan 20 km/jam, 40km/jam, 60 km/jam dan 80 km/jam dan diujikan pada permukaan jalan es dan tanpa es. Meskipun tidak ditemukan komponen frekuensi tertentu atau rentang spektrum yang tampaknya berkorelasi dengan es di jalan, hasil percobaan menunjukkan adanya pergeseran amplitudo yang terukur dalam rentang spektrum yang luas dari kebisingan akustik.

Alonso dkk. (2014) menggunakan mikrofon, *signal conditioning* dan *A/D converter* untuk merekam kebisingan antara ban kendaraan dan permukaan jalan yang dilalui. Mikrofon dipasang pada ban kendaraan dengan posisi jauh dari mesin. Selama mengemudi, kebisingan yang dihasilkan karena interaksi ban dan jalan ditangkap dengan mikrofon. Sinyal yang dihasilkan oleh mikrofon disesuaikan dan diperkuat oleh *signal conditioning*. *A/D converter* mengubah sinyal analog menjadi sinyal digital. Ekstraksi fitur dilakukan dengan cara membagi blok sinyal digital yang dihasilkan menjadi blok-blok sinyal selama 125 ms. $1/3$ *octave bands* dari tiap blok dihitung dan dinormalisasi. Jadi fitur yang dihasilkan adalah sebuah vektor yang berisi komponen spektral dari blok-blok sinyal 125 ms. Seleksi fitur dilakukan untuk menghemat memori dan meningkatkan *performance* sistem. Seleksi fitur menggunakan *Recursive Feature Elimination (RFE)* dan *zero-norm minimization (L0)*. Kemudian dijalankan *SVM* mengklasifikasi kondisi permukaan jalan dalam dua kategori: jalan kering dan jalan basah.

Wang dkk. (2008) menggunakan *roadside sensor technology* dan *floating car technology* untuk mengumpulkan berbagai informasi tentang kondisi jalan dan cuaca. Informasi yang dikumpulkan antara lain termasuk gesekan ban-jalan, suhu atmosfer, kelembaban, tekanan, kecepatan angin, arah angin, dan visibilitas. Kemudian struktur dan prinsip dari kedua peralatan teknis ini dianalisis. Teori *D-S evidence* dan metode jaringan saraf tiruan digunakan untuk memprediksi tingkat kondisi jalan di bawah kondisi cuaca buruk.

Namun, penggunaan sensor untuk mengenali tipe permukaan jalan hanya dapat memperoleh informasi dari suatu bagian jalan tertentu dan pemeliharaan dari sensor sangat mahal. Oleh karena itu seiring dengan semakin meningkatnya penggunaan kamera maka semakin



banyak peneliti yang menggunakan teknologi pengolahan citra untuk mengenali tipe permukaan jalan.

Yamamoto dkk. (2005) menggunakan rekaman CCTV untuk mengumpulkan informasi tentang kondisi jalan dan menerapkan metode interaksi manusia-komputer untuk mengekstrak nilai *grayscale* dan parameter karakteristik suhu permukaan jalan untuk memprediksi kondisi permukaan jalan. Penggunaan CCTV dan metode interaksi manusia dan komputer menghasilkan akurasi sampai 80%.

Muneo dkk. (2005) menggunakan kamera kendaraan untuk mengumpulkan informasi dari permukaan jalan dan menggunakan parameter karakteristik polarisasi citra permukaan jalan untuk membentuk model evaluasi kondisi jalan yang licin.

Popescu dkk. (2008) menggunakan fitur *GLCM (contrast, energy, entropy, homogeneity, dan variance)* dan fitur kepadatan tepi untuk mengklasifikasi permukaan jalan. Setiap data gambar dibagi ke dalam 16 blok ukuran 128x128 piksel. Tiap blok akan diekstraksi fitur teksturnya menggunakan *GLCM*. Data kemudian dibagi menjadi data latih dan data uji. Klasifikasi menggunakan *euclidean distance*. Tipe permukaan jalan dibedakan atas area jalan dan bukan area jalan (seperti rumput dan kerikil). Hasil penelitian menunjukkan bahwa nilai *contrast* dan *energy* memiliki pengaruh besar terhadap kemiripan tekstur permukaan jalan.

Tang dkk. (2011) menggunakan fitur warna, tekstur dan tepi jalan untuk mengklasifikasi area dari suatu pemandangan (area jalan, bukan jalan dan tepi jalan). Fitur warna menggunakan kombinasi dari ruang warna *RGB, YcbCr, dan HSV*. Fitur tekstur menggunakan fitur *GLCM (entropy, energy, contrast, correlation, homogeneity)* dan fitur tepi menggunakan gabungan dari *Hough-transform, countour tracking* dan deteksi tepi *Canny*. Fitur yang dihasilkan adalah sebuah vektor berdimensi 163. Klasifikasi menggunakan *K-Nearest Neighbors (KNN)* dan *Artificial Neural Network (ANN)* dengan 2 layer dan *H hidden node*. Hasil klasifikasi *KNN* menunjukkan bahwa nilai *k* menghasilkan akurasi yang berbeda-beda dan akurasi terbaik mencapai 85%. Hasil klasifikasi *ANN* menunjukkan bahwa jumlah *hidden node* dan jumlah iterasi pelatihan mempengaruhi nilai akurasi, dan akurasi terbaik mencapai 93%.

Sun dan Jia (2013) menggunakan fitur warna dari *histogram level* keabuan, fitur tekstur dari *GLCM* dan metode *Naïve Bayes* untuk mengklasifikasikan jenis permukaan jalan (kering, bersalju tipis dan bersalju tebal). Klasifikasi dengan metode NB menghasilkan akurasi 96,3%.

Hadjidemetriou (2016) menggunakan fitur warna dari *histogram* dan fitur tekstur dari *DCT (Discrete Cosinus Transform)* dan *GLCM* dan metode *SVM* untuk mendeteksi kerusakan pada permukaan jalan aspal. Proses klasifikasi dengan metode *SVM* menghasilkan akurasi 81,97%.

Zhao dkk. (2017) menggunakan metode *SVM* untuk mengklasifikasi kondisi permukaan jalan (basah, kering, salju, es, dan tergenang air). Algoritme *grid search* dan *Particle Swarm Optimization (PSO)* digunakan untuk mengoptimalkan fungsi *kernel* dan faktor pinalti pada *SVM*, sehingga akurasi meningkat. Fitur warna dan tekstur diekstraksi menggunakan *eigen vector*. Hasil klasifikasi mencapai 85%.



Kajian penelitian di atas membuktikan bahwa teknologi pengolahan citra cukup efektif untuk mengklasifikasi dan mengenali tipe permukaan jalan yang dipengaruhi kondisi cuaca dengan kondisi jalan yang baik. Oleh karena itu masih diperlukan penelitian lanjutan untuk mengenali tipe jalan berdasarkan kondisi jalan itu sendiri tanpa pengaruh dari keadaan cuaca.

2.1.2 Kombinasi *KNN-NB* dan *KNN-SVM*

Tabrizi dan Cavus (2016) menggunakan gabungan metode *KNN* dan *SVM* untuk mengklasifikasi nomor polisi negara Iran. Prosesnya diawali dengan mencari posisi nomor polisi pada gambar mobil, kemudian dilakukan segmentasi untuk mendapatkan gambar tiap karakter pada nomor polisi (32x30 piksel). Ekstraksi fitur menggunakan *structural feature*, fitur *histogram* horisontal dan vertikal dan fitur *zoning*. *Structural feature* menghasilkan vektor berdimensi 16, fitur *histogram* horisontal dan vertikal menghasilkan vektor berdimensi 10 dan fitur *zoning* menghasilkan vektor berdimensi 19. Metode *KNN* dengan fitur struktural dan *histogram* digunakan pertama kali untuk mengklasifikasi tiap karakter, jika klasifikasi menghasilkan beberapa karakter yang mirip maka dilakukan proses klasifikasi menggunakan metode *SVM* dengan fitur *zoning*. Gabungan *KNN-SVM* menghasilkan akurasi 97%, lebih baik dari metode tunggalnya (*SVM* dengan akurasi 94%).

Machhale dkk. (2015) mengklasifikasi gambar *Magnetic Resonance Imaging (MRI)* otak ke dalam kelas normal dan abnormal menggunakan metode *SVM-KNN*. Sistem *preprocessing* meliputi konversi gambar *RGB* menjadi gambar *grayscale*, *filtering* menggunakan *median filter* untuk menghapus *noise* gambar, dan proses *skull masking*, yaitu menghapus jaringan yang bukan jaringan otak. *Preprocessing* diperlukan untuk meningkatkan akurasi klasifikasi. Ekstraksi fitur gambar hasil *preprocessing* menggunakan fitur *grayscale* (*mean*, *variance*, *standard deviation*, *skewness*, dan *kurtosis*), fitur tekstur *GLCM* (*entropy*, *dissimilarity*, *inverse*, *energi*, *contrast*, dan *IDM*), dan fitur simetris. Pada tahap pelatihan menggunakan metode *SVM* kernel *RBF* dengan parameter $C=2$ dan $\gamma=2e-1$. Tahap pengujian menggunakan metode *KNN* untuk menghitung jarak terdekat data uji ke titik *support vector*. Metode *SVM-KNN* menghasilkan akurasi 98% sedangkan pengujian pada metode *SVM* saja menghasilkan akurasi 96%.

Calle-Alonso dkk. (2013) mengklasifikasi data biomedis ke dalam 6 kelas (*Carcinoma*, *Fibroadenoma*, *Mastopathy*, *Glandular*, *Connective*, dan *Adipose*) berdasarkan kemiripan kriteria menggunakan *Bayesian Regression* dan *KNN*. Data biomedis yang dipakai adalah data jaringan payudara hasil dari tes *Electrical Impedance Spectroscopy (EIS)*. Fitur diekstraksi menggunakan Argan plot, fitur yang digunakan adalah impedivitas pada frekuensi nol, sudut fase pada 500 kHz, kemiringan frekuensi tinggi sudut fase (pada 250, 500 dan 1000 kHz), jarak impedansi antara ujung spektral, area di bawah spektrum, area dinormalisasi oleh jarak impedansi antara ujung spektral, spektrum maksimum, jarak antara hambatan pada frekuensi nol dan bagian nyata dari titik frekuensi maksimum, dan panjang kurva spektral. Fitur yang dihasilkan adalah sebuah vektor berdimensi 9. Proses klasifikasi diawali dengan membuat sebuah matrik perbandingan dari keseluruhan data latih. Matriks ini berisi perbandingan semua pasangan objek untuk menentukan apakah mereka mirip atau tidak dengan menggunakan fungsi perbedaan tetap. *Bayesian Regression* dijalankan untuk mendapatkan parameter regresi. Parameter regresi dan



metode *KNN* digunakan untuk mengklasifikasikan objek baru. Hasil pengujian menunjukkan bahwa rata-rata akurasi menggunakan metode *Bayesian Regression-KNN* mencapai 77%.

Ferdousy dkk. (2013) menggunakan kombinasi dari *KNN* dan *NB* (*cNK*) untuk mengklasifikasi beberapa *data set* dari *UCI Machine Learning*. Metode *KNN* untuk mengklasifikasi atribut numerik dan metode *NB* untuk mengklasifikasi atribut kategori. Akurasi klasifikasi dengan metode *cNK* lebih baik dibanding dengan metode *NB*.

Kannan dkk. (2011) menggunakan hibrida dari *KNN* dan *SVM* untuk mengklasifikasikan gambar *MRI*. Pertama menjalankan *KNN* untuk mengklasifikasi data uji ke dalam kelas *benign*, *normal*, dan *malignant*. Jika hasil *KNN* ada kesamaan nilai tetangga pada kelas-kelasnya, maka akan dijalankan metode *SVM*. Sistem yang diusulkan menunjukkan akurasi 86,7% dalam klasifikasi gambar *MRI*. Waktu komputasi dan akurasi *KNN-SVM* lebih cepat dibandingkan metode *SVM*.

Lee (2015) menggunakan hibrida *KNN* dan *NB* untuk pengenalan emosi bicara (*data set SAVEE* dan *EmoDB*). *KNN* digunakan untuk memilih *k* data terdekat, jika dalam *k* data terdekat memiliki label kelas yang sama, maka proses klasifikasi dihentikan. Jika dalam *k* data terdekat ada lebih dari satu label kelas, maka dijalankan metode *NB* untuk mendapatkan kelas label dari data uji. Akurasi metode *KNN-NB* untuk *data set EmoDB* mencapai 68,4% dan *data set SAVEE* mencapai 69,2%, sedangkan untuk akurasi metode *SVM* mencapai 65,4% untuk *EmoDB* dan 72% untuk *SAVEE*. Jika dibandingkan dengan metode *SVM*, waktu komputasi *KNN-NB* lebih cepat.

Berdasarkan kajian tersebut diatas terlihat bahwa gabungan metode *KNN* dan metode *SVM* atau *Naive Bayes* cukup efektif untuk meningkatkan akurasi.

2.1.3 Ekstraksi fitur warna dan tekstur

You dan Zhang (2010) menggunakan kombinasi dari fitur warna dan bentuk untuk pengambilan gambar merek dagang. Fitur warna menggunakan *color moment* pada ruang warna *HSV*. Dasar dari momen warna terletak pada asumsi bahwa distribusi warna dalam suatu gambar dapat diartikan sebagai distribusi probabilitas. Distribusi probabilitas dicirikan oleh sejumlah momen unik (contoh: *mean*, *variance* dan *standard deviation*).

GLCM adalah metode ekstraksi fitur tekstur dalam domain spasial yang diusulkan oleh Haralick dkk. (1973). *GLCM* banyak dipakai dalam beberapa aplikasi, antara lain klasifikasi gambar (Karimah dan Harjoko, 2017; Palm, 2004), *CBIR* (*content based image retrieval*) (Chandana dkk., 2017), *biomedical* (Mustafa dkk., 2010), dan lain-lain. *GLCM* memiliki karakter *invariant* terhadap rotasi dan translasi, artinya *GLCM* memiliki nilai yang sama pada gambar *original* maupun gambar hasil rotasi dan translasinya. Dalam beberapa penelitian menunjukkan bahwa *GLCM* memiliki performa yang lebih baik dibandingkan dengan metode ekstraksi fitur lainnya. Weszka dkk. (1976) menggunakan metode ekstraksi fitur tekstur *GLCM* dan *Fourier* pada foto-foto sembilan jenis panorama (danau, rawa *marsh*, kebun buah-buahan, jalan kereta api, semak belukar, pinggiran kota, rawa *swamp*, perkotaan, dan hutan kayu) untuk mengklasifikasi tekstur, hasil penelitian mereka menunjukkan fitur *GLCM* memiliki performa yang lebih baik dari pada *Fourier*. Ohanian dkk. (1992) membandingkan kinerja fitur *Markov*, *Gabor*, *Fractal*, dan *GLCM* dalam mengenali tekstur visual, dan mereka menyimpulkan *GLCM* memiliki kinerja yang lebih baik daripada yang lain. Zhang dkk. (2002) membandingkan fitur dari berbagai algoritme untuk



segmentasi gambar dan hasilnya menunjukkan *GLCM* memiliki kinerja yang lebih baik. Öztürk dan Akdemir (2018) membandingkan fitur *GLCM*, *LBP*, *LBGLCM*, *GLRLM* dan *SFTA* untuk mengklasifikasi gambar hispatologi dan fitur *GLCM* menghasilkan akurasi yang paling tinggi.

Kajian di atas menunjukkan bahwa fitur warna *moment* cukup sederhana untuk diterapkan dalam proses klasifikasi dan fitur tekstur *GLCM* cukup akurat dipakai dalam proses klasifikasi.

2.2 Pengenalan Citra

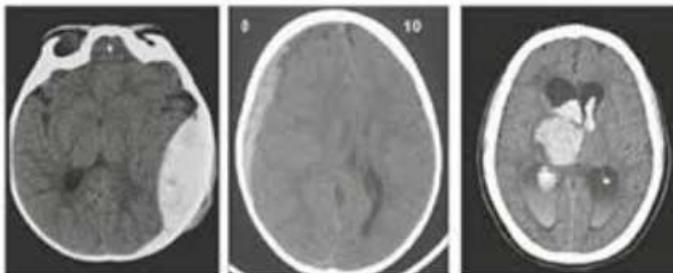
Citra merupakan representasi atau gambaran dari suatu objek. Citra terbagi 2, yaitu citra analog dan citra digital.

2.2.1 Citra analog

Citra analog bersifat *continue*. Citra analog dihasilkan dari alat-alat analog, seperti video kamera analog, *CT scan*, sensor *rontgen*, sensor gelombang pendek pada radar, sensor *ultrasound* pada *USG*, dan lain-lain. Contoh citra analog adalah gambar pada monitor televisi, foto sinar X, foto yang tercetak di kertas foto, lukisan pemandangan alam, hasil *CT scan*, dan gambar-gambar yang terekam pada pita kaset. Gambar 2.1 adalah contoh citra analog.



(a) Foto hasil kamera analog
Sumber: www.flickr.com



(b) Hasil *CT-scan*

Sumber: van Gijn dkk., 2007

Gambar 2.1 Contoh citra analog



2.2.2 Citra digital

Citra digital adalah citra analog yang diubah dalam bentuk diskrit yang diambil menggunakan mesin dengan pendekatan berdasarkan sampling dan kuantisasi. Sampling pada citra menyatakan besar kecilnya ukuran piksel pada citra, sedangkan kuantisasi menyatakan nilai tingkat keabuan suatu piksel (Awcock, 1995).

Dalam pengertian matematis, citra digital merupakan fungsi intensitas cahaya $f(x,y)$, dimana nilai x dan y adalah koordinat spasial dan nilai fungsi tersebut pada setiap titik (x,y) adalah nilai tingkat keabuan pada titik tersebut. Jadi suatu citra digital dapat dianggap sebagai suatu matriks, dimana indeks baris dan kolomnya menyatakan koordinat suatu piksel pada citra dan elemen matriksnya merupakan tingkat keabuan piksel pada citra tersebut.

Besarnya nilai piksel pada kuantisasi citra tergantung pada kedalaman piksel, yaitu banyaknya bit yang digunakan untuk merepresentasikan intensitas warna piksel. Citra digital dengan kedalaman bit n piksel disebut citra n -bit.

Berdasarkan cara penyimpanannya di memori, citra digital dapat dibedakan menjadi: citra biner, citra *grayscale* dan citra warna.

1. Citra biner, terdiri dari 2 warna: hitam dan putih. Dibutuhkan 1 *bit* di memori untuk menyimpan kedua warna ini. Gambar 2.2(a) adalah contoh citra biner.
2. Citra *grayscale*, terdiri dari beberapa warna yang jumlahnya tergantung pada jumlah bit yang disediakan di memori. Citra 2-bit terdiri dari 4 warna keabuan, citra 3-bit terdiri dari 8 warna keabuan, dan seterusnya. Semakin besar jumlah bit warna, maka semakin halus gradasi warna yang terbentuk dan semakin besar memori yang dibutuhkan. Gambar 2.2(b) adalah contoh citra *grayscale*.
3. Citra warna, setiap nilai pikselnya merupakan gabungan dari 3 warna dasar (merah/R, hijau/G, biru/B). Untuk citra 8-bit di memori, setiap warna dasar terdiri dari $2^8 = 256$ warna. Jadi setiap piksel memiliki kombinasi warna $256 \times 256 \times 256 = 16.777.216$ warna (lebih dari 16 juta warna). Jumlah warna yang tercakup cukup besar dan dapat mewakili warna-warna yang ada di alam. Gambar 2.2(c) adalah contoh citra warna.



Gambar 2.2 Contoh citra biner (a), citra *grayscale* (b), dan citra warna (c)



Sumber: www.instantstreetview.com

Elemen-elemen pada citra digital:

1. Kecerahan (*brightness*), adalah intensitas cahaya yang dipantulkan piksel yang dapat ditangkap oleh sistem penglihatan manusia.
2. Kontras (*contrast*), adalah sebaran terang dan gelap pada sebuah citra.
3. Kontur (*contour*), adalah keadaan yang ditimbulkan oleh perubahan intensitas pada piksel-piksel yang bertetangga.
4. Warna (*color*), adalah panjang gelombang cahaya yang dipantulkan oleh objek yang dapat ditangkap sistem visual manusia.
5. Bentuk (*shape*), adalah properti intrinsik dari objek 3 dimensi.
6. Tekstur (*texture*), adalah distribusi spasial dari derajat keabuan dari piksel-piksel yang bertetangga.

2.3 Warna

Warna yang dilihat manusia dari sebuah objek sebenarnya adalah pantulan cahaya yang mengenai objek tersebut. Panjang gelombang cahaya yang dapat ditangkap mata manusia berkisar antara 400-700 nm (Zhou dkk., 2010).

2.3.1 Ruang warna

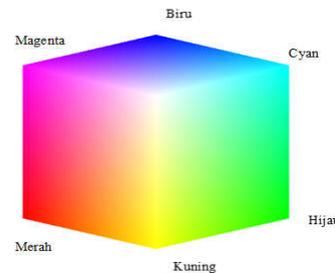
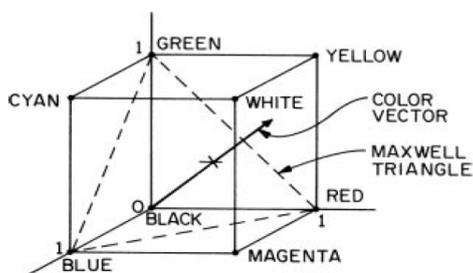
Ruang warna didefinisikan sebagai suatu sistem koordinat dan setiap warna dinyatakan dengan satu titik di dalamnya (Gonzales dan Woods, 2002). Ruang warna dibentuk untuk menstandarkan suatu warna. Ruang warna yang dikenal pada perangkat digital adalah *RGB*, namun kemudian dikembangkan ruang warna yang lain, antara lain *HSV*, *LUV*, *YcbCr*, *CMYK*.

2.3.1.1 Ruang warna *RGB*

Ruang warna *RGB* umumnya dipakai pada grafis komputer. Ruang warna ini terdiri atas 3 *channel* warna: merah (R), hijau (G), dan biru (B). Setiap titik warna (piksel) dibentuk oleh 3 *channel* tersebut. Gambar 2.2 (a) menggambarkan koordinat ruang warna *RGB*, dan Gambar 2.3 (b) adalah kubus ruang warna secara nyata dengan resolusi 24 *bit*.

Gambar 2.3 Ruang warna *RGB* dalam bentuk koordinat (a) dan 24-bit (b)

Sumber: Ghosh dkk. (2018)





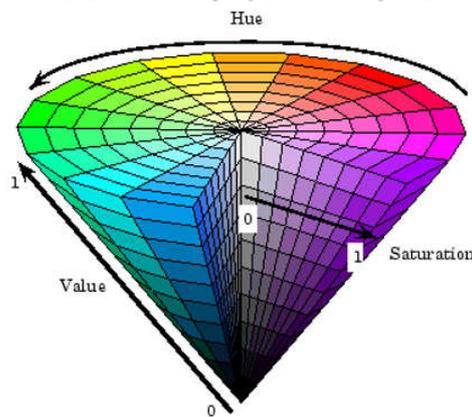
Ruang warna *RGB* biasa digunakan, karena memiliki kemudahan dalam perancangan *hardware*, tetapi tidak ideal untuk beberapa algoritme pemrosesan citra digital, karena ketiga *channel* warna terkorrelasi erat (Crane, 1997). Sebagai contoh, untuk memperoleh warna murni dari biru dengan ruang warna *RGB* menjadi kompleks, karena *channel B* berpasangan dengan *G* dan *R*. Hal ini dapat diatasi jika menggunakan ruang warna *HSV*, karena ruang warna *HSV* mampu memisahkan informasi warna sesuai dengan sistem penglihatan manusia.

2.3.1.2 Ruang warna *HSV*

Ruang warna *HSV* merepresentasikan warna seperti yang dilihat oleh mata manusia. Ruang warna *HSV* memiliki 3 *channel* warna: *hue*, *saturation*, dan *value*.

Hue mengacu properti warna yang bervariasi dari merah ke hijau, *saturation* mengacu properti warna yang bervariasi dari merah ke merah muda, sedangkan *Value* (intensitas cahaya) bervariasi dari hitam ke putih (Kakumanu dkk., 2007). Gambar 2.4 adalah ruang warna *HSV*.

Gambar 2.4 Ruang warna *HSV*



Sumber: An dkk. (2010)

2.3.2 Konversi ruang warna *RGB* ke *HSV*

Proses konversi warna dari *RGB* ke *HSV* menggunakan rumus-rumus dari Acharya dan Ray (2005):

Nilai piksel pada *channel R*, *G*, *B* dinormalisasi ke nilai 0-1 dengan Persamaan (2.1).

$$R = \frac{R}{255}, G = \frac{G}{255}, B = \frac{B}{255} \quad (2.1)$$

Selanjutnya hitung nilai *channel H*, *S*, *V* dengan Persamaan (2.2) – (2.5).

$$V = \max(R, G, B) \quad (2.2)$$

$$S = \begin{cases} 0, & \text{jika } V = 0 \\ 1 - \frac{\min(R, G, B)}{V}, & V \neq 0 \end{cases} \quad (2.3)$$



$$H = \begin{cases} 0, & \text{jika } S = 0 \\ \frac{60 * (G - B)}{(v - \min(R, G, B))}, & \text{jika } V = R \\ 60 * \left[2 + \frac{b * r}{(v - \min(R, G, B))} \right], & \text{jika } V = G \\ 60 * \left[4 + \frac{r - g}{(v - \min(R, G, B))} \right], & \text{jika } V = B \end{cases} \quad (2.4)$$

$$H = H + 360 \text{ jika } H < 0 \quad (2.5)$$

Keluaran dari perhitungan diatas adalah $0 \leq V \leq 1$; $0 \leq S \leq 1$; $0 \leq H \leq 360$

Untuk gambar 8-bit, maka hasil perhitungan H, S, V dikonversi menjadi Persamaan (2.6).

$$V \leftarrow 255V; S \leftarrow 255S; H \leftarrow H/2 \quad (2.6)$$

2.3.3 Ekstraksi fitur warna

Fitur warna dihitung dari *standard deviation channel* H, S, dan V dari ruang warna HSV, Persamaan (2.7). Jadi fitur warna yang dihasilkan adalah sebuah vektor berdimensi 3.

$$S = \sqrt{\frac{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2}{n(n-1)}} \quad (2.7)$$

Dimana:

S = standard deviation

x_i = nilai x ke- i

n = ukuran sampel

2.3 Tekstur

Tekstur adalah pola informasi atau susunan struktur yang ditemukan dalam gambar (Ojala dan Pietik, 1996), dan dapat dicirikan sebagai seperangkat sifat statistik lokal dari intensitas tingkat abu-abu piksel (Semler dan Dettori, 2007). Selain itu tekstur juga dapat didefinisikan sebagai pola struktural berulang dari permukaan objek alam dan buatan, dan secara intuitif dapat dilihat sebagai ukuran sifat permukaan seperti kehalusan, kekasaran, dan keteraturan (Puig dkk., 2010).

2.3.1 Metode ekstraksi fitur tekstur

Terdapat 4 metode berbeda untuk mengekstraksi fitur tekstur yaitu: statistik, struktural, *model-based*, dan *domain transform* (Tuceryan dan Jain, 1998).

1. Metode statistik

Metode statistik merepresentasikan tekstur sebagai distribusi dan hubungan antara tingkat keabuan dari suatu gambar. Teknik ini adalah salah satu metode pertama dalam visi mesin (Tuceryan dan Jain, 1998). Dengan menghitung fitur-fitur lokal pada setiap titik dalam gambar,



dan menurunkan sekumpulan perhitungan statistik dari distribusi fitur-fitur lokal, metode statistik dapat digunakan untuk menganalisis distribusi spasial nilai-nilai keabuan suatu gambar. Berdasarkan jumlah piksel yang menentukan fitur lokal, metode statistik dapat diklasifikasikan menjadi *first-order* (satu piksel), *second-order* (dua piksel) dan *higher-order* (tiga atau lebih piksel). *First-order* memperkirakan sifat-sifat seperti rata-rata dan *varians* dari nilai-nilai piksel individu, dan mengabaikan interaksi spasial antara piksel-piksel dalam suatu gambar. *Second-order* dan *higher-order* memperkirakan sifat-sifat dari hubungan antara dua atau lebih piksel dengan ketentuan tertentu (contohnya berdasarkan jarak dan arah sudut). Fitur statistik *second-order* yang paling populer untuk analisa tekstur didapat dari matriks *co-occurrence* (Kannan dkk., 2011; Materka, 1998; Ojala dan Pietik, 1996). Pendekatan berdasarkan matriks multidimensional *co-occurrence* menggungguli *wavelet packets* (salah satu pendekatan berbasis transformasi) ketika diterapkan pada klasifikasi tekstur (Valkealahti dan Oja, 1998).

2. Metode struktural

Metode struktural merepresentasikan tekstur berdasarkan sifat lokal (tekstur mikro) dan organisasi spasial (tekstur makro) dari sifat lokal. Metode struktural memberikan deskripsi simbolik yang baik dari gambar, dan berguna untuk pembentukan tekstur serta analisis tekstur (Tuceryan dan Jain, 1998). Namun, metode ini tidak cocok untuk tekstur alami karena variabilitas dari tekstur mikro dan tekstur makro, dan tidak ada perbedaan yang jelas antara mereka (Materka, 1998; Ojala dan Pietik, 1996).

3. Metode berbasis model

Metode berbasis model mencoba memodelkan tekstur sebagai operasi matematika. Dengan memberikan parameter model, fitur tekstur dapat dihasilkan. Metode ini didasarkan pada konstruksi model gambar yang dapat digunakan tidak hanya untuk menggambarkan tekstur, tetapi juga untuk mensintesisnya. Parameter model menangkap kualitas tekstur yang dirasakan penting. Contoh metode berbasis model antara lain *Markov random fields* model (Cross dan Jain, 1983) dan *fractals* model (Mandelbrot, 1983).

4. Metode *domain transform*

Metode *domain transform* digunakan untuk mencari informasi domain spektral dari gambar. Metode ini menganalisis konten frekuensi dari gambar. Contoh dari metode *domain transform*, antara lain transformasi *Fourier* (Zhou dkk., 2001), filter *Gabor* (Turner, 1986) dan transformasi *wavelet* (Laine dan Fan, 1993).

2.3.2 Ekstraksi fitur tekstur menggunakan *GLCM*

GLCM merupakan salah satu metode ekstraksi fitur tekstur menggunakan pendekatan statistik. *GLCM* yang diusulkan oleh Haralick dkk. (1973) dengan 14 fitur yang menggambarkan pola spasial gambar abu-abu. *GLCM* didasarkan pada ekstraksi gambar skala keabuan. *GLCM* menyatakan hubungan antara dua piksel yang berdekatan, pixel pertama dikenal sebagai referensi dan yang kedua dikenal sebagai piksel tetangga (Hall-Beyer, 2017). *GLCM* merupakan matriks persegi dengan dimensi L , di mana L sama dengan jumlah tingkat keabuan pada gambar.

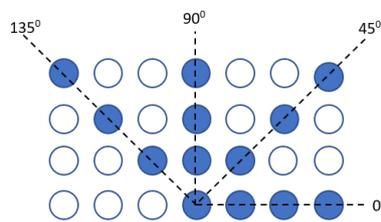


Setiap elemen matriks adalah jumlah kemunculan sepasang piksel dengan nilai i dan piksel dengan nilai j yang berada pada jarak dan sudut tertentu.

Langkah pertama ekstraksi fitur dengan *GLCM* adalah merubah gambar RGB menjadi gambar dalam skala keabuan (*grayscale*), dengan Persamaan (2.3):

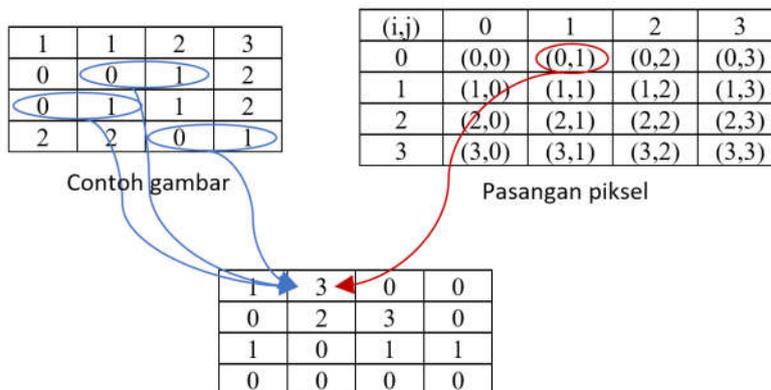
$$G = 0,299 * R + 0,587 * G + 0,114 * B \quad (2.3)$$

Langkah berikutnya adalah membentuk matriks *GLCM* dari gambar skala keabuan. Matriks *GLCM* ($P_{\vec{r}}(i, j)$) adalah jumlah piksel dengan $i \in 1..L$ yang terjadi pada arah \vec{r} terhadap piksel dengan nilai $j \in 1..L$. Gambar 2.5 menunjukkan arah \vec{r} (sudut (θ)) dan jarak (d). Misal gambar original G memiliki tingkat keabuan 4 (0,1,2,3). Gambar 2.6 merupakan contoh proses untuk menghasilkan *framework* matriks *GLCM* untuk $d=1$ dan $\theta=0^\circ$.



Gambar 2.5 Arah sudut dan jarak dari pasangan piksel

Sumber: www.mathworks.com



Matriks *GLCM* dengan $d=1$ dan $\theta=0^\circ$

Gambar 2.6 Proses pembentukan matriks *GLCM*

Sumber: www.mathworks.com

Matriks *GLCM* diubah menjadi matriks *GLCM* yang simetris dengan cara menjumlahkan dengan *transpose* matriksnya, Persamaan (2.8). Untuk menghilangkan ketergantungan pada ukuran citra, maka Matriks *GLCM* yang simetris tersebut dinormalisasi dengan Persamaan (2.9) dan Persamaan (2.10) adalah hasil normalisasi dari matriks di Persamaan (2.9).

$$matriks\ GLCM + transpose\ matriks\ GLCM = matriks\ GLCM\ simetris$$



$$\begin{bmatrix} 1 & 3 & 0 & 0 \\ 0 & 2 & 3 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 1 & 0 \\ 3 & 2 & 0 & 0 \\ 0 & 3 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 1 & 0 \\ 3 & 4 & 3 & 0 \\ 1 & 3 & 2 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (2.8)$$

$$P_{i,j} = \frac{V_{i,j}}{\sum_{i,j=0}^{N-1} V_{i,j}} \quad (2.9)$$

Dimana $V_{i,j}$ nilai elemen matriks dan $P_{i,j}$ adalah nilai elemen matriks hasil normalisasi. Persamaan (2.10) adalah hasil normalisasi dari matriks *GLCM*.

$$\text{Matriks GLCM yang dinormalisasi} = \begin{bmatrix} \frac{1}{12} & \frac{1}{8} & \frac{1}{24} & 0 \\ \frac{1}{8} & \frac{1}{6} & \frac{1}{8} & 0 \\ \frac{1}{24} & \frac{1}{8} & \frac{1}{12} & \frac{1}{24} \\ 0 & 0 & \frac{1}{24} & 0 \end{bmatrix} \quad (2.10)$$

Dari matrik *GLCM* yang dinormalisasi tersebut, kemudian dihitung fitur *GLCM*. Penelitian ini menggunakan 6 fitur *GLCM* sebagai berikut:

1. Contrast

Contrast adalah ukuran intensitas atau variasi tingkat abu-abu antara piksel referensi dan tetangganya (antara pasangan piksel). Nilai fitur *contrast* dirumuskan pada Persamaan (2.11).

$$\text{Contrast} = \sum_{n=0}^{L-1} n^2 \{ \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} P_{\bar{r}}(i,j) \} \quad (2.11)$$

Dimana $|i - j| = n$

2. Dissimilarity

Nilai fitur *dissimilarity* dirumuskan pada Persamaan (2.12).

$$\text{Dissimilarity} = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} P_{\bar{r}}(i,j) |i - j| \quad (2.12)$$

3. Homogeneity

Homogeneity atau disebut juga *Inverse Difference Moment (IDM)* mengukur kesamaan dari *cooccurrence matrix* dan *diagonal matrix*. Nilai fitur *homogeneity* dirumuskan pada Persamaan (2.13).

$$\text{Homogeneity} = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} \frac{P_{\bar{r}}(i,j)}{1+(i-j)^2} \quad (2.13)$$

4. Angular Second Moment (ASM)

ASM mengukur homogenitas suatu gambar. Nilai *ASM* tinggi jika nilai piksel-piksel sangat mirip. Nilai fitur *ASM* dirumuskan pada Persamaan (2.14).

$$\text{ASM} = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} P_{\bar{r}}(i,j)^2 \quad (2.14)$$



Dimana $P_{\bar{r}}(i, j)$ adalah nilai piksel pada matriks *GLCM yang dinormalisasi*, L adalah ukuran matriks *GLCM*.

5. Energy

Nilai fitur *energy* dirumuskan pada Persamaan (2.15).

$$Energy = \sqrt{ASM} \quad (2.15)$$

6. Correlation

Korelasi menghitung ketergantungan linear dari nilai-nilai *gray level* dalam matriks *GLCM*. Korelasi menunjukkan bagaimana piksel referensi terkait dengan tetangganya. Nilai fitur *correlation* dirumuskan pada Persamaan (2.16).

$$Correlation = \frac{\sum_{i=0}^{L-1} \sum_{j=0}^{L-1} P_{\bar{r}}(i, j) \frac{(i-\mu_i)(j-\mu_j)}{\sqrt{\sigma_i^2 \sigma_j^2}}}{\sqrt{\sigma_i^2 \sigma_j^2}} \quad (2.16)$$

Dimana:

$$\mu_i = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} i P_{\bar{r}}(i, j) \quad (2.17)$$

$$\mu_j = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} j P_{\bar{r}}(i, j) \quad (2.18)$$

$$\sigma_i = \sqrt{\sum_{i=0}^{L-1} \sum_{j=0}^{L-1} P_{\bar{r}}(i, j) (i - \mu_i)^2} \quad (2.19)$$

$$\sigma_j = \sqrt{\sum_{i=0}^{L-1} \sum_{j=0}^{L-1} P_{\bar{r}}(i, j) (j - \mu_j)^2} \quad (2.20)$$

2.4 Metode Klasifikasi

2.4.1 K-Nearest Neighbors (KNN)

Nearest Neighbor (NN) merupakan algoritme sederhana yang didasarkan pada pembelajaran yang diawasi (*supervised learning*). Tujuannya adalah menemukan titik yang paling dekat dengan data uji. Jika tujuannya adalah untuk menemukan k data terdekat dari data pelatihan yang ada ketika sampel baru muncul dan mengklasifikasikan sampel yang muncul sesuai dengan kelas yang paling mirip (Mitchell, 1997) maka metode ini dikenal dengan *KNN*. Ukuran kedekatan antar data diukur menggunakan fungsi jarak *euclidean*, yang dijelaskan oleh (Mitchell, 1997) dengan sebuah formula. Misalkan sebuah data x memiliki vektor fitur $\{a_1(x), a_2(x), \dots, a_n(x)\}$, dimana $a_r(x)$ adalah nilai dari atribut ke r dari data x . Maka jarak antara dua data x_i dan x_j adalah $d(x_i, x_j)$, Persamaan (2.21).

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^n (a_r(x_i) - a_r(x_j))^2} \quad (2.21)$$



2.4.2 Naïve Bayes (NB)

Metode NB didasarkan pada teorema Bayes tentang probabilitas (Jang dan Park, 2015), yaitu bahwa prediksi peluang di masa depan berdasarkan pengalaman sebelumnya. Teorema Bayes dikombinasikan dengan Naïve yang mengasumsikan kondisi antar atribut adalah saling bebas. Metode klasifikasi Naïve Bayes mengasumsikan bahwa ada atau tidak ciri tertentu dari sebuah kelas tidak ada hubungannya dengan ciri dari kelas lainnya. Persamaan (2.22) adalah rumus dari teorema Bayes.

$$P(H|X) = P(H) \frac{P(X|H)}{P(X)} \quad (2.22)$$

Dimana: X adalah data dengan kelas yang belum diketahui, H adalah kelas hipotesis dari data X , $P(H|X)$ adalah probabilitas hipotesis H berdasarkan kondisi X (*posterior probability*), $P(H)$ adalah probabilitas hipotesis H (*prior probability*), $P(X|H)$ adalah probabilitas X berdasarkan kondisi pada hipotesis H , dan $P(X)$ adalah probabilitas X . Dalam permasalahan klasifikasi, teorema Bayes disesuaikan menjadi Persamaan (2.23).

$$P(C|x_1, \dots, x_n) = P(C) \frac{P(x_1 \dots x_n|C)}{P(x_1 \dots x_n)} \quad (2.23)$$

Dimana: C adalah kelas, $x_1 \dots x_n$ adalah atribut-atribut yang dibutuhkan untuk melakukan klasifikasi. Maka rumus tersebut menjelaskan bahwa peluang masuknya sampel atribut x_k ke dalam kelas C (*posterior*) adalah peluang munculnya kelas C (*prior*), dikali dengan peluang kemunculan atribut-atribut sampel pada kelas C (*likelihood*), dibagi dengan peluang kemunculan atribut-atribut sampel secara global (*evidence*). Oleh karena itu Persamaan (2.23) dapat ditulis menjadi Persamaan (2.24).

$$\text{Posterior} = \text{Prior} \frac{\text{likelihood}}{\text{evidence}} \quad (2.24)$$

Nilai dari *posterior* tersebut akan dibandingkan dengan nilai-nilai *posterior* kelas lainnya untuk menentukan ke kelas apa suatu sampel akan diklasifikasikan. Rumus Bayes pada Persamaan (2.23) diuraikan lebih lanjut dengan menjabarkan $(C|x_1, \dots, x_n)$ menggunakan aturan perkalian (dijabarkan pada Persamaan 2.25).

$$\begin{aligned} P(C|x_1, \dots, x_n) &= P(C)P(x_1, \dots, x_n|C) \\ &= P(C)P(x_1|C)P(x_2, \dots, x_n|C, x_1) \\ &= P(C)P(x_1|C)P(x_2|C, x_1)P(x_3, \dots, x_n|C, x_1, x_2) \\ &= P(C)P(x_1|C)P(x_2|C, x_1) \dots P(x_n|C, x_1, x_2, \dots, x_{n-1}) \quad (2.25) \end{aligned}$$

Hasil dari penjabaran Persamaan (2.23) menunjukkan bahwa faktor-faktor syarat yang mempengaruhi nilai probabilitas semakin kompleks. Hal ini mengakibatkan perhitungan tersebut sulit untuk dilakukan. Oleh karena itu digunakan asumsi independensi yang sangat tinggi (*naïve*), bahwa tiap atribut (x_1, x_2, \dots, x_n) saling bebas satu sama lain. Dengan asumsi tersebut, maka berlaku Persamaan (2.26).



$$P(x_i|x_j) = \frac{P(x_i \cap x_j)}{P(x_j)} = \frac{P(x_i)P(x_j)}{P(x_j)} = P(x_i) \quad (2.26)$$

Untuk $i \neq j$, maka berlaku Persamaan (2.27).

$$P(x_i|C, x_j) = P(x_i|C) \quad (2.27)$$

Dengan independensi *naïve* tersebut syarat peluang menjadi sederhana, sehingga perhitungan $P(C|x_1, \dots, x_n)$ mudah untuk dilakukan. Maka penjabaran $P(C|x_1, \dots, x_n)$ disederhanakan menjadi Persamaan (2.28).

$$P(C|x_1, \dots, x_n) = P(C)P(x_1|C)P(x_2|C) \dots P(x_n) = P(C) \prod_{i=1}^n P(x_i|C) \quad (2.28)$$

Persamaan di atas merupakan model dari teorema *Naïve Bayes* yang akan digunakan dalam proses klasifikasi. Untuk data kontinyu digunakan rumus distribusi *Gaussian* (Persamaan 2.29).

$$P(X = x_i | C = c_j) = \frac{e^{-\frac{(x_i - \mu_{ij})^2}{2\sigma_{ij}^2}}}{\sqrt{2\pi}\sigma_{ij}} \quad (2.29)$$

Dimana μ_{ij} adalah nilai mean dan σ_{ij} adalah *standard deviation* dari data latih, yang menyatakan varian dari seluruh atribut pada data latih, Persamaan (2.30) dan (2.31).

$$\mu_{ij} = \frac{1}{n} \sum_{k=1}^n f_k \quad (2.30)$$

$$\sigma_{ij} = \sqrt{\frac{1}{(n-1)} \sum_{k=1}^n (f_k - \mu_{ij})^2} \quad (2.31)$$

Probabilitas *posterior* dihitung untuk semua kelas, $C_i \in C$. Kelas dengan nilai probabilitas *posterior* tertinggi merupakan hasil prediksi kelas (\hat{C}_i), Persamaan (2.32).

$$\hat{C}_i = \max P(C_i|X(x_1, x_2, \dots, x_n)) \quad (2.32)$$

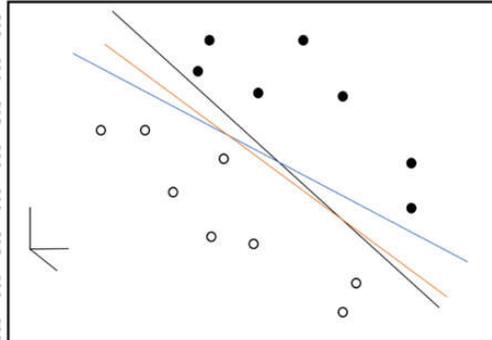
2.4.3 Support Vector Machine (SVM)

Metode SVM adalah salah *state-of-the-art* dalam pengenalan pola (Byun dan Lee, 2003). SVM dikembangkan oleh Vapnik dkk. (1998). Pengembangan SVM pertama digunakan untuk mengklasifikasikan data *linear*, dan dalam perkembangannya dapat memecahkan masalah *non-linear* dengan menggunakan konsep *kernel trick* pada data berdimensi tinggi.

Gambar 2.7 menunjukkan pola yang merupakan anggota dari dua kelas. Pola di kelas -1 dilambangkan dengan pola putih dan pola di kelas 1 dilambangkan dengan warna hitam. Mengklasifikasikan berarti menemukan garis pemisah (*hyperplane*) yang memisahkan pola berdasarkan kelas mereka. Garis hitam, merah dan biru adalah beberapa alternatif *hyperplane*. Garis tebal pada Gambar 2.8 adalah *hyperplane* terbaik. *Margin* adalah jarak antara *hyperplane*

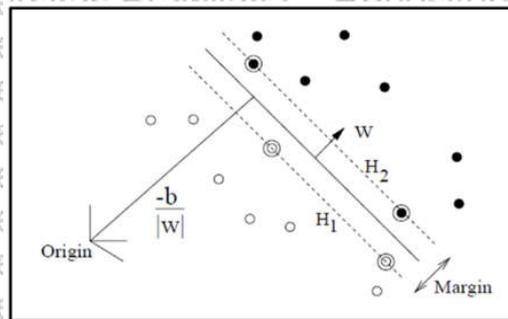


dan pola terdekat dari masing-masing kelas. Pola yang paling dekat dengan *hyperplane* disebut *support vector*.



Gambar 2.7 Garis hitam, merah dan biru adalah beberapa alternatif garis pemisah (*hyperplane*)

Sumber: Osuna dkk. (1997)



Gambar 2.8 Garis hitam adalah *hyperplane* terbaik dan titik-titik data yang dilingkari adalah *support vector*.

Sumber: Osuna dkk. (1997)

Proses untuk menemukan pola *hyperplane* terbaik dari pola-pola ini adalah konsep dasar SVM (Burges, 1998). Data dilambangkan dengan $x_i \in R^d$ dan kelas dilambangkan $y_i \in \{-1, 1\}$ untuk $i = 1, \dots, l$, di mana l menyatakan banyak data. Misalkan kedua kelas dapat dipisahkan oleh sebuah *hyperplane*, yang didefinisikan dalam Persamaan (2.33).

$$w \cdot x + b = 0 \tag{2.33}$$

Data x_i yang termasuk dalam kelas -1 dirumuskan sebagai pola yang memenuhi Pertidaksamaan (2.34), sedangkan data x_i yang termasuk dalam kelas 1 dirumuskan dalam Pertidaksamaan (2.35).

$$w \cdot x_i + b \leq -1 \quad \text{for } y_i = -1 \tag{2.34}$$

$$w \cdot x_i + b \geq 1 \quad \text{for } y_i = 1 \tag{2.35}$$

Pertidaksamaan (2.34) dan (2.35) dapat digabungkan menjadi Pertidaksamaan (2.36).



$$y_i(w \cdot x_i + b) - 1 \geq 0 \quad \forall i \quad (2.36)$$

Margin terbesar ditentukan dengan memaksimalkan nilai jarak dari *hyperplane* dengan titik terdekatnya, yaitu $1/\|w\|$ dan dapat diformulasikan sebagai masalah *Quadratic Programming (QP)*, yang menemukan titik minimum Persamaan (2.37), dengan mempertimbangkan kendala pada Pertidaksamaan (2.36).

$$\min \tau(w) = \frac{1}{2} \|w\|^2 \quad (2.37)$$

Persoalan ini dapat dipecahkan dengan menggunakan teknik *Lagrange Multiplier*, Persamaan (2.38).

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^l \alpha_i y_i ((x_i \cdot w + b) - 1) \quad (2.38)$$

Dimana $i = 1, \dots, l$, l adalah banyaknya data, α_i adalah *Lagrange Multiplier* yang bernilai 0 atau positif ($\alpha_i \geq 0$). Nilai optimal dari Persamaan (2.38) dapat dihitung dengan meminimalkan L terhadap w dan b , dan memaksimalkan L terhadap α_i . Pada titik optimal $L = 0$, Persamaan (2.38) dapat dimodifikasi sebagai masalah maksimalisasi yang hanya mengandung α_i saja, sebagaimana Persamaan 2.39.

$$\sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=0}^l \alpha_i \alpha_j y_i y_j x_i \cdot x_j \quad (2.39)$$

Dengan batasan Persamaan (2.40) dan (2.41).

$$\alpha_i \geq 0, i = 1, \dots, l \quad (2.40)$$

$$\sum_{i=1}^l \alpha_i y_i = 0 \quad (2.41)$$

Data yang berkorelasi dengan α_i yang positif inilah disebut *support vector*.

Jika *hyperplane* tidak dapat memisahkan data dengan sempurna, maka kendala dalam Persamaan (2.36) tidak dapat dipenuhi. Untuk mengatasi masalah ini, *SVM* dirumuskan ulang dengan teknik *soft margin*. Persamaan (2.36) dimodifikasi dengan memasukkan variabel *slack* ξ_i ($\xi_i > 0$), (Persamaan (2.42)).

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad \forall i \quad (2.42)$$

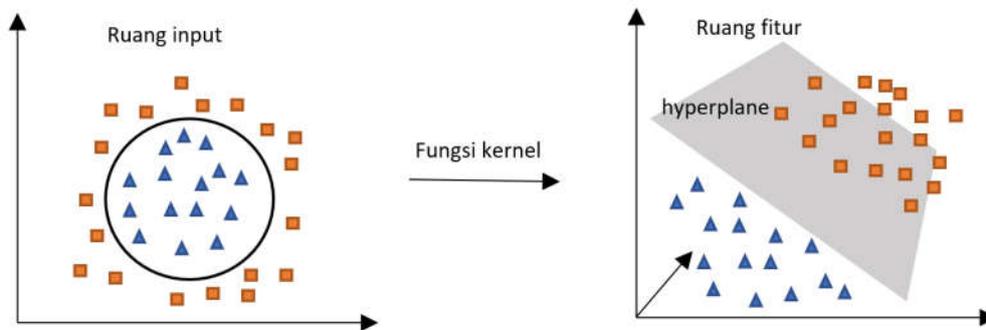
Maka Persamaan (2.37) dimodifikasi menjadi Persamaan (2.43).

$$\min \tau(w, \xi) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i \quad (2.43)$$

Parameter C berfungsi untuk mengontrol *tradeoff* antara *margin* dan kesalahan klasifikasi ξ . Nilai C yang besar berarti akan memberikan *penalty* yang lebih besar untuk kesalahan klasifikasi.



Pada masalah *non-linear*, SVM menggunakan fungsi *kernel*. Data (ruang input/input space) dipetakan oleh fungsi $\Phi(x_i)$ ke ruang vektor dengan dimensi yang lebih tinggi (ruang fitur/feature space), Gambar 2.9. Notasi matematika dari pemetaan ini dalam Persamaan (2.44)



Gambar 2.9 Memetakan data input ke ruang fitur menggunakan fungsi kernel

Sumber: Mahmoodi dkk. (2011)

$$\Phi: R^d \rightarrow R^q, d < q \tag{2.44}$$

Proses pembelajaran dalam SVM dalam mencari titik-titik *support vector* hanya bergantung pada *dot product* dari data di ruang baru dengan dimensi yang lebih tinggi, yaitu $\Phi(x_i) \cdot \Phi(x_j)$. Menurut teori *Merker*, perhitungan produk titik dapat diganti dengan fungsi *kernel* $K(x_i, x_j)$. Hal ini disebut trik *Kernel*, yang diformulasikan dalam Persamaan (2.45).

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \tag{2.45}$$

Hasil klasifikasi dari data x didapatkan dari Persamaan (2.46).

$$f(\Phi(x)) = \sum_{i=1}^l \alpha_i y_i K(x, x_i) + b \tag{2.46}$$

Dimana *SV* adalah data yang dipilih menjadi *support vector*.

Beberapa fungsi kernel yang sering digunakan antara lain:

1. *Kernel Linear*.

Fungsi *kernel linear* merupakan fungsi kernel yang paling sederhana dan fungsi ini merupakan sebuah *inner product* ditambah dengan konstanta c . Persamaan (2.47) adalah fungsi *kernel linear*.

$$K(x_i, x_j) = x_i^T \cdot x_j + c \tag{2.47}$$

2. *Kernel Polynomial* (derajat d).

Kernel Polynomial digunakan untuk data *non-linear* dan sesuai digunakan pada masalah dimana semua data pelatihan sudah dinormalisasi. Persamaan (2.48) adalah fungsi *kernel polynomial*.

$$K(x_i, x_j) = (x_i^T \cdot x_j + c)^d \tag{2.48}$$



c merupakan konstanta dan d adalah derajat dari *polynomial*.

3. Kernel Radial Basis Function (RBF).

Kernel RBF digunakan untuk data *non-linear*. Persamaan (2.49) adalah fungsi *kernel RBF*.

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \quad (2.49)$$

Dimana $\gamma = \frac{1}{2\sigma^2}$, dimana σ adalah nilai varians dari data.

2.5 Pengukuran Kinerja Sistem

Kinerja sistem diukur menggunakan akurasi (Powers, 2011). Akurasi adalah tingkat kedekatan antara hasil prediksi dengan hasil sebenarnya. Akurasi dapat dihitung menggunakan *confusion matrix*, Tabel 2.1.

Tabel 2.1 Confusion matrix

		Prediksi	
		Positif	Negatif
Aktual	Positif	TP	FN
	Negatif	FP	TN

Dimana:

- *TP (True Positive)* bernilai 1 jika kelas aktual bernilai positif/true dan kelas prediksi juga bernilai positif/true.
- *FP (False Positive)* bernilai 1 jika kelas aktual bernilai negatif/false dan kelas prediksi bernilai positif/true, serta melibatkan seluruh tipe kelas data.
- *FN (False Negative)* bernilai 1 jika kelas aktual bernilai positif/true dan kelas prediksi bernilai negatif/false, serta melibatkan satu tipe kelas data.
- *TN (True Negative)* bernilai 1 jika kelas aktual bernilai negatif/false dan kelas prediksi juga bernilai negatif/false.

Berdasarkan *confusion matrix* diatas maka perhitungan *accuracy* ditunjukkan pada Persamaan (2.47).

Accura



BAB V PENUTUP

5.1 Kesimpulan

Berdasarkan penelitian yang sudah dilakukan maka dapat disimpulkan bahwa:

1. Nilai fitur tekstur dengan metode *GLCM* dipengaruhi oleh jarak ketetanggaan (d) dan sudut ketetanggaan (θ) antar piksel. Nilai fitur *contrast* dan *dissimilarity* berbanding lurus dengan d , semakin besar d maka semakin besar nilai fitur yang dihasilkan. Fitur *homogeneity*, *ASM*, *energy*, dan *correlation* berbanding terbalik dengan d , semakin besar nilai d maka semakin kecil nilai fitur yang dihasilkan. Perubahan nilai θ memberikan hasil yang berbeda pada nilai fitur tekstur yang dihasilkan, namun perubahan nilai fiturnya tidak berlaku signifikan. Fitur *contrast* dan *dissimilarity* pada tipe aspal dan aspal rusak berbanding lurus dengan nilai θ . Fitur *homogeneity* dan *ASM* pada kelas paving berbanding terbalik dengan perubahan nilai θ .
2. Perbedaan nilai k , jarak (d) dan sudut (θ) memberikan pengaruh pada akurasi klasifikasi tipe permukaan jalan menggunakan metode klasifikasi *KNB* dan *KSVM* (Tabel 4.31-4.38). Kombinasi dari nilai k , jarak (d), sudut (θ), dan metode klasifikasi dengan akurasi terbaik pada tiap kelas (Tabel 4.40) akan digunakan pada klasifikasi bertingkat.
3. Akurasi pengujian *offtime* klasifikasi bertingkat sebesar 82%. Hasil pengujian menunjukkan bahwa klasifikasi bertingkat memiliki kemampuan pengenalan tipe jalan yang cukup akurat untuk tipe aspal dan aspal rusak ($\pm 90\%$).
4. Pada pengujian *realtime*, data video diambil pada *frame rate* 30 fps dan *frame size* 1920x1080. Pengujian *realtime* pada ukuran *frame* 1920x1080 menghasilkan akurasi yang rendah pada tipe aspal, aspal rusak dan paving. Hal ini dikarenakan adanya perbedaan metode pengambilan data latih dan data uji, sehingga data gambar pada data uji dan data latih berbeda cukup jauh dengan demikian nilai fiturnya juga berbeda jauh. Oleh karena itu perlu dilakukan pengujian pada 6 ukuran *frame* uji yang lain: 480x270, 960x540, 1440x810, 2400x1350, 2880x1620, dan 3360x1890. Pengujian *realtime* pada beberapa ukuran *frame* menunjukkan bahwa tipe aspal rusak memiliki akurasi terbaik pada ukuran *frame* uji 3390x1890, tipe aspal rusak memiliki akurasi terbaik pada ukuran *frame* uji 960x540, tipe paving memiliki akurasi terbaik pada ukuran *frame* uji 960x540, dan tipe tanah memiliki akurasi terbaik pada ukuran *frame* uji 480x270, 960x540, 1440x810, 1960x1080, dan 2400x1350. Waktu komputasi per *frame* pada pengujian *realtime* rata-rata mencapai 0,105 detik.
5. Pada pengujian *offtime*, metode klasifikasi bertingkat memiliki akurasi yang lebih baik dibanding metode pembandingan. Hasil pengenalan di tiap kelas secara umum lebih baik dari metode metode *histogram-GLCM-Naïve Bayes* dan metode *histogram-DCT-GLCM-SVM*, kecuali untuk kelas tanah, hasil pengenalan tertinggi ada pada metode *histogram-DCT-GLCM-SVM*.



Pada pengujian *realtime*, metode *NB-Histogram-GLCM* mencapai hasil pengenalan 90% untuk tipe aspal, 80% untuk tipe aspal rusak, 100% untuk tipe tanah, dan 98% untuk tipe paving. Metode *SVM-Histogram-DCT-GLCM* mencapai hasil pengenalan 100 % untuk tipe tanah, 82% untuk tipe aspal, sedangkan untuk tipe aspal rusak dan paving mencapai hasil 54% dan 60%. Waktu komputasi rata-rata per *frame* dari metode *SVM-Histogram-DCT-GLCM* adalah 0,418 detik dan metode *NB-Histogram-GLCM* adalah 0,036 detik.

5.2 Saran

Beberapa arah pengembangan penelitian ini yang dapat dilakukan untuk memperbaiki kekurangan dan mengembangkan studi ini yaitu: Pertama, menggunakan nilai k , d dan θ dengan variasi yang lebih banyak lagi dan dengan menggunakan metode optimasi untuk memperoleh nilai k , d dan θ yang dapat menghasilkan akurasi yang lebih baik. Kedua, menggunakan metode klasifikasi yang lain juga perlu dicoba agar dapat menghasilkan akurasi dan waktu komputasi yang lebih baik lagi untuk semua tipe jalan dan dalam berbagai ukuran *frame*. Ketiga, pengembangan penelitian ini pada perangkat *mobile* juga sangat memungkinkan, karena penggunaan *data set* yang tidak terlalu besar dan metode klasifikasi yang mudah dan sederhana untuk diterapkan.