

BAB II

KAJIAN PUSTAKA

2.1 Tinjauan Pustaka

Penelitian mengenai penerapan analisis wavelet pada data spektral untuk kategorisasi otot domba. *Wavelet Daubechies* orde kelima ("db5") ditemukan sebagai fungsi *wavelet* terbaik untuk dekomposisi sinyal spektral domba. Ciri koefisien *wavelet* yang diekstraksi dari *wavelet* db5 pada tingkat dekomposisi kelima kemudian digunakan sebagai input dari *Least Squares Support Vector Machine* (LS-SVM) untuk mengembangkan model klasifikasi. *Principal Component Analysis* (PCA) digunakan untuk pengurangan dimensi. Performa klasifikasi dari pengklasifikasi LS-SVM bersamaan dengan transformasi *wavelet* dan PCA dibandingkan dengan model LS-SVM berdasarkan data spektral orisinal, turunan pertama, turunan kedua, smoothing, *Standard Normal Variate* (SNV), dan *Multiplicative Scatter Correction* (MSC). Tujuan dari penelitian ini adalah analisis *wavelet* pada data spektral untuk kategorisasi otot domba. Hasil dari penelitian ini menjelaskan bahwa penerapan analisis *wavelet* memiliki potensi besar untuk kategorisasi otot domba bersamaan dengan analisis multivariat dan pemrosesan citra (Pu dkk., 2015).

Penelitian mengenai kelayakan penggunaan teknologi pencitraan hiperspektral dikombinasikan dengan transformasi *wavelet* dan algoritme *Multiway Partial Least Squares* (N-PLS) untuk memprediksi jumlah layak total (TVC) dari rempah-rempah daging sapi selama penyimpanan diselidiki. Data spektral rata-rata diekstraksi dari citra *hyperspectral* dan selanjutnya didekomposisi dalam sembilan level oleh fungsi *wavelet daubechies8* (db8) menjadi koefisien aproksimasi (A9) dan detail sembilan koefisien (D1–D9). Memilih *wavelet* koefisien untuk menyusun matriks tiga dimensi yang berbeda, selanjutnya mengintegrasikan algoritma N-PLS untuk menetapkan model prediktif untuk mendeteksi TVC pada daging sapi berbumbu. Tujuan dari penelitian ini adalah deteksi nilai *Total Viable Count* (TVC) daging sapi berbumbu menggunakan teknologi pencitraan hiperspektral yang dipadukan dengan transformasi *wavelet* dan *Multiway Partial Least Squares* (N-

PLS) algoritma untuk memprediksi TVC daging sapi berbumbu selama penyimpanan. Hasil percobaan menunjukkan bahwa model N-PLS dengan koefisien D4, D5, D6, D7 (disebut juga $D_{4,5,6,7}$ -N-PLS) menunjukkan kemampuan prediksi yang sangat baik untuk TVC sampel daging sapi berbumbu dengan koefisien determinasi yang lebih tinggi dalam prediksi (R^2_p) sebesar 0,934 dan galat kuadrat rata-rata akar bawah diperkirakan dengan prediksi sebesar 0,755 dibandingkan model N-PLS lainnya, model PLS spektrum mentah, dan model *Unfold*-PLS. Oleh karena itu, model yang dibangun dengan menggunakan *array* data tiga dimensi dan algoritma N-PLS memiliki potensi besar di bidang tersebut (D. Yang dkk., 2018).

Penelitian mengenai DWTLSTM (*Discrete Wavelet Transform and Long Short-Term Memory*) untuk pemrosesan sinyal hidung elektronik dalam pemantauan kualitas daging sapi. Sistem pengemasan cerdas diperlukan untuk terus memantau kualitas daging sapi dan populasi mikroba baik industri daging maupun konsumen akhir. Selain itu, beberapa studi kelayakan hidung elektronik (*enose*) untuk penilaian kualitas daging sapi cepat juga dilakukan dalam beberapa tahun terakhir. Tujuan dari penelitian ini adalah Diskrit *Wavelet* Transform and *Long Short-Term Memory* (DWTLSTM) diusulkan untuk mengatasi sinyal *e-nose* yang terkontaminasi dengan kebisingan dalam pemantauan kualitas daging sapi. Dalam tugas klasifikasi kualitas daging sapi, usulan kami memiliki kinerja yang baik dengan akurasi rata-rata 94,83% dan rata-rata F-measure 85,05%. Hasil ini menunjukkan bahwa DWTLSTM mengungguli metode konvensional seperti *k-Nearest Neighbor* (k-NN), *Linear Discriminant Analysis* (LDA), *Support Vector Machine/Support Vector Regression* (SVM/SVR), *Multilayer Perceptron* (MLP), dan bahkan *Long Short-Term Memory* (LSTM) (Wijaya dkk., 2021).

Penelitian mengenai meningkatkan akurasi klasifikasi deret waktu: menggabungkan informasi global dan lokal dalam kriteria kesamaan. Studi ini, *new Similarity Measure* (SIMscl) berdasarkan informasi global dan local telah diusulkan untuk meningkatkan tingkat presisi dari *One Nearest Neighbor* (1NN). Secara khusus, informasi global mencatat sifat intrinsik deret waktu, dan dicerminkan oleh dua indikator: informasi bentuk dan kompleksitasnya; lingkungan setempat

informasi memperhatikan kecocokan nilai yang tepat dan direalisasikan oleh LB_keogh. Secara bersamaan, metode berdasarkan *Wavelet Haar Diskrit* multi-skala transformasi, ekstraksi titik kunci, dan simbolisasi telah diajukan untuk mengekstrak informasi bentuk. Tujuan dari penelitian ini adalah untuk menguji kemanjuran bentuk yang diusulkan kemiripan SIMshape dan kemiripan hibrid SIMscl, percobaan dilakukan pada dua set data: bintang kurva ringan dan daging sapi. Hasil ini menunjukkan bahwa SIMshape dapat menangani beberapa deret waktu yang salah diklasifikasikan oleh *Euclidean Distance* (ED), LB_keogh, dan *Complexity Invariant Distance* (CID), dan SIMscl memiliki presisi yang lebih tinggi daripada ED, LB_keogh, dan CID dalam klasifikasi time series 1NN (He dkk., 2014).

Penelitian mengenai mencirikan suara dari berbagai sumber di rumah ayam pedaging komersial. Data audio yang dikumpulkan di rumah ayam pedaging komersial adalah suara campuran dari berbagai sumber yang berisi informasi berguna mengenai kondisi kesehatan burung, perilaku burung, dan pengoperasian peralatan. Tujuan dari penelitian ini adalah untuk menentukan rentang frekuensi dari enam suara umum, termasuk vokalisasi burung, kipas, sistem pakan, pemanas, kepak sayap, dan dustbathing, pada burung umur 1 sampai 8 minggu pada broiler Ross 708 komersial. Sebuah mikrofon dipasang di tengah kandang ayam pedaging pada ketinggian 40 cm di atas punggung burung untuk merekam data audio pada frekuensi sampling 44.100 Hz. Data audio didekomposisi menggunakan *Maximum Overlap Discrete Wavelet Transform* (MODWT). Dekomposisi keenam sumber bunyi tersebut kemudian ditransformasikan dengan metode *Fast Fourier Transform* (FFT) untuk menghasilkan spektrum amplitudo satu sisi. Hasil ini menunjukkan bahwa karakterisasi suara dari berbagai sumber suara di kandang broiler komersial memberikan informasi yang berguna untuk analisis akustik lebih lanjut yang dapat membantu manajemen peternakan dalam pemantauan berkelanjutan terhadap kesehatan dan perilaku hewan (X. Yang dkk., 2021).

Penelitian mengenai skema penyembunyian data berdasarkan *U-Net* dan transformasi *wavelet*. Tujuan dari penelitian ini adalah penyembunyian data berdasarkan *U-Net* dan *wavelet*, yang secara inovatif menggabungkan keunggulan

U-Net dalam pemrosesan ciri detail citra dan kemampuan transformasi *wavelet* untuk membagi detail citra. Koefisien *wavelet* dari data rahasia disembunyikan dicitra melalui jaringan tersembunyi, dan citra tersembunyi dengan efek visual yang bagus dapat diperoleh. Itu ciri citra dibagi menjadi empat koefisien *wavelet* dalam jaringan ekstraksi, dan akhirnya citra data rahasia diperoleh melalui transformasi *wavelet* terbalik. Hasil ini menunjukkan bahwa skema yang diusulkan memiliki tembus pandang data tersembunyi yang lebih baik daripada skema tradisional, dan data ekstraksi lebih akurat (Liu dkk., 2021).

Penelitian mengenai sebuah sistem untuk memprediksi marbling dari *dry-cured* ham dari citra irisan ham. Ham kering adalah produk daging tradisional Mediterania yang dikonsumsi di seluruh dunia. Penilaian marbling online sangat menarik bagi industri untuk tujuan klasifikasi. Namun, sampai sekarang penilaian ini secara tradisional dilakukan oleh panel ahli dan metodologi ini tidak bisa akan diterapkan di industri. Tujuan dari penelitian ini adalah mengusulkan sistem yang sepenuhnya otomatis untuk memprediksi tingkat kelerengan pengeringan kering irisan ham, yang menggabungkan: (1) ciri tekstur warna *Regions Of Interest* (ROI) yang diekstraksi secara otomatis untuk setiap otot; dan (2) model *Machine Learning* untuk memprediksi kelerengan. Untuk algoritma ekstraksi ROI, lebih dari 90% piksel ROI jatuh ke otot yang sebenarnya. Setinggi ini akurasi dalam prediksi marbling untuk irisan ham yang diawetkan kering akan memungkinkan kami menerapkan penerapannya di industri daging ham kering. Hasil ini menunjukkan bahwa hasilnya bagus dan kecepatan tinggi dari prediksi marmer untuk irisan ham kering, aplikasi ini dapat digunakan dalam industri ham yang diawetkan kering (Cernadas dkk., 2022).

Penelitian mengenai deteksi *Optimized Electronic Nose System* (OENS) untuk mendeteksi pemalsuan daging babi dalam daging sapi secara lebih akurat. Penyaringan kebisingan dilakukan dengan validasi silang dengan *mother wavelets* yang berbeda, yaitu, *Haar*, *dmey*, *coiflet*, *symlet*, dan *Daubechies*. Array sensor dioptimalkan dengan pengurangan dimensi menggunakan *Principal Component Analysis* (PCA). Tujuan dari penelitian ini adalah menganalisis tujuh kelas daging, yang terdiri dari tujuh campuran daging sapi dan babi yang berbeda. Kelas pertama

dan ketujuh adalah 100% daging sapi dan 100% daging babi, sedangkan kelas kedua, ketiga, keempat, kelima, dan keenam berisi 10%, 25%, 50%, 75%, dan 90% daging sapi dalam sampel masing-masing sebanyak 100 gram.. Hasil ini menunjukkan bahwa hasil uji klasifikasi untuk mendeteksi daging sapi dan babi memiliki akurasi sebesar 98,10% menggunakan *Support Vector Machine* (SVM) yang dioptimalkan. Dengan demikian, OENS memiliki kinerja yang menguntungkan untuk mendeteksi pemalsuan daging babi dalam daging sapi untuk otentikasi halal (Sarno, 2020).

Penelitian mengenai deteksi dan pengenalan residu obat hewan pada penggunaan daging sapi menggunakan transformasi *wavelet* diskrit hiperspektral dan *deep learning*. Tujuan dari penelitian ini adalah algoritma *Discrete Wavelet Transform* (DWT) hiperspektral digabungkan dengan *Convolutional Neural Network* (CNN) untuk membuat algoritma klasifikasi *deep learning*. Klasifikasi dan pengenalan dilakukan pada *hyperspectral* dataset dari lima kelompok sampel daging sapi yang mengandung hewan residu obat. Data hiperspektral mengalami reduksi dimensi menunjukkan peningkatan koefisien OA dan Kappa untuk independent set sampel prediksi. Algoritma DWT berdasarkan basis *wavelet Haar* fungsi mampu menyaring gangguan frekuensi tinggi informasi melalui *filter low-loss multilayer* dan mencapai data pengurangan dimensi. CNN memiliki kemampuan ekstraksi ciri dan pembelajaran yang kuat. OA model berdasarkan DWT dan CNN adalah 91,6%, yang mana melebihi model berdasarkan MLP, SVM, dan RF sebesar 3,0%, 4,0%, dan 2,4%, masing-masing. Hasil ini menunjukkan bahwa DWT dan *deep learning* telah dibuktikan dalam penelitian ini yang mengusulkan metode baru untuk deteksi dan pengenalan *veteriner* yang cepat dan tidak merusak residu obat dalam daging sapi serta lebih unggul dari pada model berdasarkan CARS dan PCA (Jiang, dkk., 2022).

2.2 Landasan Teori

Landasan teori dari *wavelet* dan GLCM (*Gray Level Co-occurrence Matrix*) dapat menjadi dasar yang kuat dalam klasifikasi jenis daging menggunakan citra atau data citra. Penerapan kedua konsep ini dalam klasifikasi jenis daging dalam citra memungkinkan ekstraksi ciri yang kuat dan representasi yang mendalam tentang tekstur atau pola dalam citra daging, yang sangat diperlukan dalam proses klasifikasi yang akurat.

2.2.1 Pengolahan Citra

Penelitian mengenai teknik pemrosesan citra termal untuk penentuan distribusi suhu daging giling dicairkan dengan metode ohmic dan konvensional. Dalam studi ini, efektivitas *Ohmic Thawing* (OT) pada *Temperature Distribution* (TD) dievaluasi dengan menggunakan metode pengukuran tak rusak: *Thermal Imaging* (TI). Teknik yang digunakan untuk memproses citra termal umumnya digunakan dalam industri makanan untuk menentukan titik panas dan/atau titik dingin dalam sistem yang dirancang. Demikian pula, ditentukan bahwa TI adalah metode penginderaan suhu non-invasif dan non-kontak yang berhasil untuk mengevaluasi distribusi suhu untuk proses pencairan dalam studi ini. Penggunaan metode ini dapat memberikan visual dan citra dan data distribusi suhu yang jelas selama ohmik pencairan bahan makanan. Direkomendasikan bahwa kriteria suhu pencairan harus diatur dengan mempertimbangkan kepanasan sampel dan suhu kritis yang dituju. Untuk desain dari sistem OT skala industri, penggunaan suhu permukaan berkorelasi dengan suhu dingin kritis di dalam sampel dapat berguna untuk mengatasi masalah pengukuran titik dingin secara akurat (Döner dkk., 2020).

Penelitian mengenai aplikasi ponsel cerdas berbasis visi mesin yang cerdas untuk evaluasi kualitas daging sapi. Keempukan daging sapi merupakan atribut terpenting yang berkorelasi dengan kualitas daging sapi, kepuasan konsumen, dan keputusan pembelian. Dalam studi ini, aplikasi *Android* baru dikembangkan dan diuji untuk pertama kalinya untuk memprediksi kualitas sampel daging sapi segar secara *real time* berdasarkan citra yang diambil dalam kondisi dunia nyata. Untuk mengatasi efek dari kondisi pencitraan yang tidak standar dan tidak terkontrol,

algoritme pemrosesan citra iluminasi, rotasi, terjemahan, dan invarian skala yang kuat berhasil dikembangkan dan divalidasi. Ciri tekstur citra yang diperoleh melalui algoritma yang dikembangkan digunakan untuk memprediksi nilai kelembutan eksperimental (nilai WBSF) menggunakan model JST cerdas dengan koefisien determinasi sekitar 0,97. Aplikasi yang dikembangkan dengan menjanjikan memprediksi kelembutan 30 sampel daging sapi yang tidak terlihat dengan koefisien determinasi lebih tinggi dari 0,98. Peningkatan "Kualitas Daging" dimasa mendatang harus mencakup penggunaan perangkat pembesar tambahan untuk mendapatkan citra dengan pembesaran tinggi. Teknik analisis tekstur citra telah banyak digunakan dalam industri makanan untuk menunjukkan sifat dan kualitas makanan. Kontribusi utama studi ini terkait dengan analisis tekstur citra sampel daging sapi segar dalam kondisi pencitraan yang tidak terkontrol. Oleh karena itu, aplikasi yang dikembangkan memiliki potensi besar untuk aplikasi pemrosesan dan rekayasa makanan yang melibatkan evaluasi kualitas daging sapi dalam kondisi yang tidak terkendali, sehingga aplikasi tersebut dapat digunakan oleh konsumen umum di pasar atau oleh spesialis di laboratorium atau industri daging untuk evaluasi kualitas daging sapi (Hosseinpour dkk., 2019).

Penelitian mengenai analisis citra sebagai alat untuk penilaian daging sapi. Marbling daging terkait dengan rasa daging, kebasahan dan kelembutan dan berhubungan langsung dengan kandungan lemak intramuskular. Istilah '*Marbled*' mengacu pada adanya garis-garis jaringan adiposa di antaranya kumpulan serabut otot pada otot rangka. Daging dengan nilai marbling tinggi diharapkan memiliki kualitas sensorik yang lebih baik. Oleh karena itu, minat untuk mengembangkan metode dan teknik untuk memperkirakan dan mengukur lemak intramuskular (meat marbling), yaitu untuk mengevaluasi kualitas daging. Digital teknologi pemrosesan citra adalah aset untuk mengukur marbling daging karena bersifat non-invasif, lebih murah dan ramah lingkungan. Studi ini menyajikan metodologi otomatis, berdasarkan teknik pengolahan citra, untuk mengidentifikasi dan melokasikan daging sapi pada citra dan menghitungnya langkah-langkah marmer. Pengujian yang direalisasikan menunjukkan bahwa adalah mungkin untuk menggunakan analisis citra dalam foto berwarna daging sapi untuk secara otomatis mengekstrak

ukuran marbling dan untuk mengevaluasi kualitas daging sapi. Namun, untuk mengembangkan algoritme yang lebih akurat, set pelatihan dan pengujian yang lebih besar harus digunakan (Caridade dkk., 2022).

Penelitian mengenai aplikasi berdasarkan pohon keputusan untuk mengklasifikasikan marbling daging sapi dengan pencitraan hiperspektral. Tujuan dari penelitian ini adalah mengembangkan sistem untuk mengklasifikasikan marbling daging sapi menggunakan teknologi pencitraan hiperspektral. Klasifikasi standar derajat marbling daging sapi Jepang digunakan sebagai referensi dan dua belas standar didigitalkan untuk mendapatkan parameter bentuk dan distribusi spasial marbling dari masing-masing kelas. Sebanyak 35 sampel otot M. longissimus dorsi dipindai oleh sistem pencitraan hiperspektral 400-1000 nm dalam mode pantulan. Algoritma pemrosesan pada citra berdasarkan metode pohon keputusan digunakan pada daerah perhatian mendapatkan kesalahan klasifikasi sebesar 0,08%. Hasil penelitian menunjukkan bahwa teknik yang diusulkan memiliki potensi besar, sebagai teknik yang tidak merusak dan cepat, yang dapat digunakan untuk mengklasifikasikan daging sapi berdasarkan marbling daging sapi (Velásquez dkk., 2017).

Penelitian mengenai sistem pencitraan bangku hiperspektral untuk mengklasifikasikan daging sapi dari sapi *Nellore* berdasarkan kelembutan. Tujuan dari penelitian ini adalah untuk mengevaluasi keakuratan klasifikasi daging sapi Nellore umur 0, 7, 14, atau 21 hari dan klasifikasi berdasarkan kelembutan dan periode penuaan menggunakan *bench-top* sistem pencitraan hiperspektral. Sistem pencitraan hiperspektral ($\lambda = 928-2524$ nm) digunakan untuk mengumpulkan citra *hyperspectral* dari *Longissimus thoracis et lumborum* (penuaan $n = 376$ dan kelembutan $n = 345$) sapi *Nellore*. Enam model diskriminan linier dikembangkan untuk mengklasifikasikan sampel berdasarkan kelembutan dan periode penuaan. Model menggunakan turunan pertama dari spektrum absorbansi parsial (berikan spektrum rentang panjang gelombang) mampu mengklasifikasikan steak berdasarkan kelembutan dengan akurasi keseluruhan 89,8%. Hasil menunjukkan bahwa *Hyperspectral Imaging* (HIS) mungkin merupakan teknologi yang layak

untuk mengklasifikasikan daging sapi berdasarkan keempukan dan periode penuaan (Nubiato dkk., 2018).

2.2.2 Tekstur Daging

Permasalahan aktual yang sering terjadi terkait penjualan daging di pasar konvensional adalah manipulasi daging babi dan sapi. Itu bisa terjadi karena kedua tekstur visual memiliki kemiripan. Tekstur adalah bagian penting dari sebuah objek. Dalam pemrosesan citra, tekstur dapat digunakan untuk klasifikasi, pengenalan, atau prediksi suatu citra. Studi ini menggunakan metode *Minimum Overlap Probability - Neural Network* untuk identifikasi ciri citra digital daging babi dan daging sapi. Metode *Minimum Overlap Probability - Neural Network* digunakan untuk memilih ciri dengan karakteristik terbaik, sedangkan *Neural Network* digunakan untuk pelatihan dan klasifikasi (Khoerul Anwar, 2022).

Sifat warna, marbling, dan tekstur permukaan otot *Longissimus Dorsi* (LD) sapi digunakan di beberapa negara untuk mengklasifikasikan karkas sesuai dengan kualitas makan yang diharapkan. Sistem *Video Image Analysis* (VIA) genggam digunakan untuk meningkatkan penilaian *grader*, namun upaya telah dilakukan untuk mengembangkan sistem citra beresolusi lebih tinggi untuk memberikan prediksi kualitas yang konsisten dan objektif berdasarkan properti ini. Selain itu, rentang ciri yang lebih luas digunakan untuk mengkarakterisasi warna dan marbling dengan lebih baik, dan transformasi *wavelet* digunakan untuk mengkarakterisasi tekstur. Metode statistik klasik *Multi Linear Regression* (MLR) dan *Partial Least Squares Regression* (PLSR) digunakan untuk mengembangkan model prediksi. Visi komputer dapat digunakan untuk mengungkapkan informasi tentang kualitas selain kelembutan seperti rasa dan kebasahan (Jackman dkk., 2009).

Warna dan tekstur adalah salah satunya karakteristik setiap daging. Studi ini digunakan pengenalan daging menggunakan citra untuk ekstraksi tekstur berbasis histogram dan ekstraksi ciri warna *Hue Saturation Intensitas* (HIS). Hasil ekstraksi ciri yang diperoleh kemudian diolah dengan menggunakan metode *Neural Network*, *Support Vector Machine* (SVM), dan *k-Nearest Neighbor* (*k-NN*) untuk dikelompokkan menjadi dua kelas daging yaitu daging babi dan daging sapi.

Penggabungan ciri antara ciri warna HSI dengan ciri tekstur berbasis histogram dapat mengklasifikasikan citra daging dengan 2 label yaitu daging sapi dan daging babi dengan baik yaitu dengan menggunakan klasifikasi *Neural Network* (Chairunnisa dkk., 2018).

Daging sapi merupakan contoh makanan halal, sedangkan daging babi merupakan makanan haram khususnya bagi umat Islam. Studi ini menggunakan metode klasifikasi *Probabilistic Neural Network* (PNN) dan ekstraksi ciri. Beberapa perbedaan mendasar antara daging babi dan sapi berdasarkan warna dan tekstur. Warna diekstraksi dengan model HSV, sebaliknya tekstur diekstraksi dengan 3 metode. Metode tersebut adalah Gabor, *Principal Component Analysis* (PCA) dan *Local Binary Pattern* (LBP). Cara ini dapat digunakan untuk membedakan daging dari makanan yang diperbolehkan dan makanan yang dilarang. Mencampur daging babi dengan daging sapi akan menjadi makanan yang dilarang bagi umat Islam dan orang lain. Klasifikasi menunjukkan bahwa ciri tekstur dan warna yang diekstrak dari citra dapat digunakan secara efektif untuk identifikasi daging sapi, daging babi, dan campuran keduanya. Mencampur dengan daging babi merupakan makanan yang haram, khususnya bagi umat Islam (Handayani dkk., 2017).

Tekstur daging sapi seperti marmer merupakan salah satu indeks kualitas daging sapi yang penting. Karena penilaian *marbling* daging sapi sangat ditentukan oleh pengalaman subyektif para penilai, terdapat ketidak konsistenan dan kesalahan dalam penilaian. Oleh karena itu, bagaimana menemukan ukuran obyektif dan kuantitatif tingkat kelimpahan *marbling* menurut standar grade *marbling* daging sapi telah menjadi salah satu studi baru di bidang ilmu daging dunia. Kita dapat menggunakan alat analisis citra Matlab untuk memroses citra marmer daging sapi. Melalui analisis data, persentase konten citra marmer telah diperoleh, dan model deteksi dapat dibentuk melalui konstruksi *Neural Network* (NN), sehingga dapat meletakkan dasar untuk tingkat prediksi dari jenis daging sapi yang tidak diketahui di masa mendatang. Eksperimen ini membuktikan bahwa menggunakan alat pengolah citra, *Matlab* sebagai pengganti metode manual layak untuk menentukan *marbling* daging sapi (Chang dkk., 2011).

Studi ini adalah untuk menggunakan ciri tekstur citra warna digital untuk memprediksi degradasi *troponin-T* pada daging sapi. Ciri tekstur citra, termasuk 88 ciri tekstur *co-occurrence* tingkat abu-abu, 81 ciri tekstur transformasi *Fourier* cepat dua dimensi, dan 48 ciri tekstur filter *wavelet Gabor*, diekstraksi dari citra warna *steak strip* daging sapi (*longissimus dorsi*, $n = 102$) berumur selama 10 d diperoleh dengan menggunakan kamera digital dan pencahayaan tambahan. Metode analisis statistik (model regresi *STEPWISE*) dan *Artificial Neural Network* (model *Support Vector Machine* (SVM) dirancang untuk mengklasifikasikan degradasi protein. Model *STEPWISE* dan SVM berdasarkan ciri tekstur citra menunjukkan potensi untuk memprediksi degradasi *troponin-T* pada daging. Studi ini menunjukkan bahwa dimungkinkan untuk menggunakan ciri tekstur citra diekstraksi dari citra warna digital dari daging sapi berumur 10 tahun untuk menyiapkan model prediksi untuk degradasi postmortem. Namun, untuk mengembangkan model yang lebih akurat, set pelatihan dan pengujian yang lebih besar harus digunakan (Sun dkk., 2014).

Keempukan merupakan salah satu indeks penting penilaian kualitas daging sapi. Metode penilaian tradisional, seperti metode evaluasi sensorik dan metode gaya geser *Warner-Bratzler*, memiliki kesalahan buatan pada derajat yang berbeda. Dengan model *Texture Profile Analysis* (TPA) ukuran permukaan pengujian setiap sampel adalah $10\text{ mm} \times 10\text{ mm} \times 10\text{ mm}$. Model tahan tekanan dan penjagaan waktu digunakan di seluruh bagian. Korelasi dianalisis antara parameter dan tingkat keempukan daging sapi. Sifat tekstur utama mengalami penurunan seiring dengan meningkatnya tingkat keempukan daging sapi, dan nilai sifat tekstur juga menunjukkan kecenderungan menurun dengan semakin banyaknya waktu mengunyah. Dikombinasikan dengan metode evaluasi sensorik, model jaringan *back propagation* (BP), model jaringan *Radical Basis Function* (RBF) dan model jaringan dan model jaringan kompetisi yang mengatur diri sendiri. Kemudian 3 model jaringan tersebut dibandingkan, dan model jaringan kompetisi mengatur diri sendiri merupakan model yang paling akurat dengan tingkat akurasi 90%, yang menunjukkan bahwa metode ini dapat menilai tingkat keempukan daging sapi secara akurat (Wang dkk., 2015).

Studi ini dilakukan klasifikasi citra daging sapi dan babi. *Spatial Fuzzy C-Means* digunakan untuk segmentasi citra. GLCM dan HSV digunakan sebagai ciri hasil segmentasi. *Learning Vector Quantization 3* (LVQ3) digunakan sebagai metode klasifikasi. Parameter *Learning Vector Quantization 3* (LVQ3) yang diuji adalah variasi nilai *learning rate* dan *window value*. Data *training* yang digunakan adalah 90% dari total data dan data *testing* yang digunakan adalah 10%. *Epoch* maksimum yang digunakan adalah 1000 iterasi. Berdasarkan hasil pengujian, akurasi tertinggi adalah 91,67% (Jasril dkk., 2018).

Daging babi dan sapi merupakan sumber utama daging merah di dunia. Namun, tidak semua orang bisa makan daging babi karena latar belakang agama atau alasan lainnya. Oleh karena itu, sangat penting untuk memastikan kemurnian daging sebelum dikonsumsi. Studi ini menerapkan pembelajaran *ansambel* untuk mengoptimalkan klasifikasi pengumpulan data hidung elektronik untuk pemalsuan daging babi pada daging sapi menggunakan pembelajaran *ansambel* merupakan salah satu metode yang banyak digunakan dan paling berhasil untuk meningkatkan kinerja. Studi ini menggunakan beberapa algoritma *machine learning* dan memilih algoritma *machine learning*, yang menghasilkan hasil terbaik sebagai pengklasifikasi dasar untuk pembelajaran *ansambel* untuk mengoptimalkan klasifikasi set data hidung elektronik untuk pemalsuan daging babi dalam daging sapi. Ada tiga pembelajaran *ansambel* yang digunakan dalam penelitian ini, yaitu *hard voting*, *stacking*, dan *bagging*. Hasil percobaan menunjukkan bahwa pembelajaran *hard voting ensemble* menggunakan *k-Nearest Neighbor* (*k-NN*) sebagai basis pengklasifikasian dapat membedakan pemalsuan daging sapi, babi, dan babi dengan baik pada tujuh kelas dengan akurasi 98,33% (Malikhah dkk., 2021).

2.2.3 Jenis-jenis Wavelet

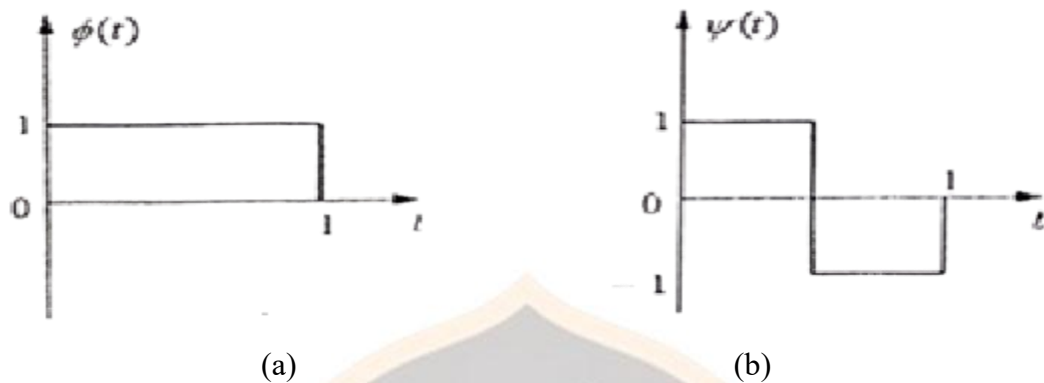
Wavelet adalah fungsi matematis yang digunakan untuk menganalisis data dalam berbagai skala. Mereka mirip dengan sinus dan kosinus dalam analisis *Fourier*, tetapi memiliki sifat yang lebih terkonsentrasi dalam domain waktu dan frekuensi. Wavelet memungkinkan pemrosesan sinyal atau data secara lokal dalam

berbagai skala, yang berarti mereka bisa mendeteksi perubahan-perubahan yang terjadi pada data dengan tingkat resolusi yang berbeda. Ini adalah keunggulan besar dibandingkan dengan transformasi *Fourier*, yang tidak memiliki kemampuan lokalitas seperti wavelet. Ada beberapa jenis *wavelet* dengan karakteristik yang berbeda-beda, termasuk wavelet *Haar*, *Daubechies* dan lainnya. Masing-masing memiliki kegunaan dan karakteristik khususnya sendiri. Penggunaan *wavelet* tersebar luas dalam bidang pemrosesan sinyal, kompresi data, analisis citra, pengolahan citra medis, dan berbagai bidang lainnya di mana analisis multi-skala diperlukan.

2.2.3.1 Wavelet *Haar*

Wavelet *Haar* adalah salah satu wavelet yang paling sederhana dan mudah dimengerti. Namanya diambil dari matematikawan Hongaria bernama Alfréd Haar, yang memperkenalkan wavelet ini pada awal abad ke-20. Wavelet *Haar* digunakan dalam transformasi *Haar*, suatu teknik transformasi *wavelet* yang mendasar. Ciri utama wavelet *Haar* adalah sifat diskritnya. Fungsinya terdiri dari dua bagian: bagian positif (1) dan bagian negatif (-1), dengan perubahan tanda mendadak di tengah. Fungsi ini dapat digunakan untuk memecah citra atau data menjadi komponen frekuensi yang berbeda pada berbagai tingkat resolusi.

Transformasi *Haar* sering digunakan dalam kompresi data karena sifatnya yang sederhana dan efisien. Pada tingkat resolusi yang lebih rendah, wavelet *Haar* dapat merepresentasikan citra dengan lebih sedikit koefisien, memungkinkan penyimpanan data yang lebih efisien. Penerapan *Haar Wavelet* melibatkan pemecahan citra menjadi subinterval dan menghitung perbedaan antara nilai rata-rata di setiap subinterval. Proses ini dapat diulangi untuk menghasilkan tingkat resolusi yang lebih tinggi. Meskipun *Haar Wavelet* sederhana, ia memiliki kegunaan praktis terutama dalam pengenalan pola, analisis sinyal, dan kompresi data dalam konteks pengolahan sinyal dan citra.



Gambar 2.1 *Wavelet Haar*, (a) Fungsi penyekalaan $\phi(t)$ dan (b) fungsi *wavelet* $\psi(t)$

Pada *wavelet Haar*, fungsi penyekalaan $\phi(t)$ merupakan fungsi kotak dengan lebar 1 pada $t = 1$, dan h_k adalah koefisien penyekalaan (R Rizal, 2012), sehingga fungsi penyekalaan di definisikan sebagai berikut:

$$\phi(t) = \sum_k h_k \sqrt{2\phi}((2t - k)) \dots \dots \dots (1)$$

$$\phi(t) = \begin{cases} 1 & \text{untuk } 0 < t < 1 \\ 0 & \text{untuk yang lain} \end{cases} \dots \dots \dots (2)$$

Fungsi penyekalaan bisa dibangun dari $\phi(2t)$ sehingga lebar fungsi kotak menjadi 0,5 pada pada $0 < t < 0,5$ demikian juga halnya pada setengah periode berikutnya:

$$\phi(t) = \phi(2t) + \phi(2t - 1) = \frac{1}{\sqrt{2}}\sqrt{2}\phi(t) + \frac{1}{\sqrt{2}}\sqrt{2}\phi(2t - 1)h_k \dots \dots \dots (3)$$

Dengan mengintegralkan kedua ruas maka akan diperoleh:

$$\int_{-\infty}^{\infty} \phi(t)dt = \int_{-\infty}^{\infty} \sum_k h_k \sqrt{2}\phi(2t - k)dt \dots \dots \dots (4)$$

Dengan mensubstitusikan $x = 2t - k$ dan $dx = 2dt$ pada ruas kanan, maka:

$$\int_{-\infty}^{\infty} \phi(t)dt = \sum_k h_k \sqrt{2} \int_{-\infty}^{\infty} \phi(x) \frac{1}{2} dx = \sum_k h_k \frac{1}{\sqrt{2}} \int_{-\infty}^{\infty} \phi(x) dx \dots \dots \dots (5)$$

Kemudian kedua ruas dibagi oleh integral, sehingga menjadi:

$$\sum_k h_k = \sqrt{2} \dots \dots \dots (6)$$

Prinsip orthogonal digunakan untuk memperoleh kondisi lainnya

$$\begin{aligned} \int \left| \phi(t)^2 dt \right. &= \int \sum_k h_k \sqrt{2} \phi(2t - k) \sum_k h_m \sqrt{2} \phi(2t - m) dt \\ &= \sum_k \sum_m h_k h_m 2 \int \phi(2t - k) \phi(2t - m) dt \\ &= \sum_k \sum_m h_k h_m \int \phi(x - k) \phi(x - m) dx \dots \dots \dots (7) \end{aligned}$$

Pada persamaan terakhir $2t$ disubstitusikan oleh x , integral pada ruas kanan sama dengan nol kecuali saat $k = m$. Dengan saat $k = m$ dapat ditentukan bahwa:

$$\sum_k h_k^2 = 1 \dots \dots \dots (8)$$

Menggunakan ortogonalitas pada saat translasi fungsi skala akan diperoleh:

$$\int \phi(t) \phi(t - m) dt = \delta_m \dots \dots \dots (9)$$

$$\begin{aligned} &\int \left[\sum_k h_k \sqrt{2} \phi(2t - k) \right] \left[\sum_l h_l \sqrt{2} \phi(2t - 2m - l) \right] dt \\ &\sum_k \sum_l h_k h_l 2 \int \phi(2t - k) \phi(2t - 2m - l) dt \dots \dots \dots (10) \end{aligned}$$

Substitusi $x = 2t$ sehingga:

$$\begin{aligned} \int \phi(t) \phi(t - m) dt &= \sum_k \sum_l h_k h_l \int \phi(x - k) \phi(x - 2m - l) dx \\ &= \sum_k \sum_l h_k h_l \delta_{k-(2m+1)} = \sum_k h_k h_m \dots \dots \dots (11) \end{aligned}$$

Sehingga diperoleh persamaan:

$$\sum_k h_k h_{k-2m} = \delta_m \dots \dots \dots (12)$$

Dengan menggunakan persamaan (3), (4) dan (9) bisa ditentukan nilai koefisien penyekalaan untuk pelewat tapis rendah.

Untuk $k = 2$, dari persamaan (3) dan (5) akan diperoleh:

$$h(0) + h(1) = \sqrt{2} \dots \dots \dots (13)$$

$$h^2(0) + h^2(1) = 1 \dots \dots \dots (14)$$

Maka nilai koefisien tapis pelewat rendah untuk *wavelet Haar* adalah:

$$h(n) = \left\{ \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right\} \dots \dots \dots (15)$$

Sedangkan untuk fungsi *wavelet* $\psi(t)$ didefinisikan sebagai berikut:

$$\psi(t) = \sum_{n=0} g(n) \sqrt{2} \phi(2t - 1) \dots \dots \dots (16)$$

Seperti fungsi penyekalaan, fungsi *wavelet* dapat dinyatakan sebagai berikut:

$$\psi(t) = \phi(2t) + \phi(2t - 1) = \frac{1}{\sqrt{2}} \sqrt{2} \phi(t) - \frac{1}{\sqrt{2}} \sqrt{2} \phi(2t - 1) \dots \dots \dots (17)$$

Nilai koefisien tapis pelewat tinggi *wavelet Haar* adalah:

$$g(n) = \left\{ \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right\} \dots \dots \dots (18)$$

Wavelet Haar adalah salah satu yang paling sederhana dan mudah dipahami karena terdiri dari fungsi basis yang sederhana. Cocok untuk deteksi jenis daging dalam citra daging karena responsnya yang tajam terhadap perubahan mendadak. *Wavelet Haar* dengan responsnya yang tajam terhadap perubahan, dapat membantu dalam menyoroti perbedaan intensitas yang signifikan antara berbagai jenis daging yang ingin dideteksi.

Perhitungan *Wavelet Haar*

Perhitungan transformasi *Wavelet Haar* melibatkan langkah-langkah yang sederhana tetapi efektif. Mari kita lakukan perhitungan untuk dekomposisi sinyal menggunakan *Wavelet Haar* pada sinyal diskrit dengan Panjang $N = 8$. Perhitungan ini akan membagi sinyal menjadi subsinyal aproksimasi (A) dan *detail* (D). Langkah-langkah perhitungan *Wavelet Haar* untuk dekomposisi adalah sebagai berikut:

1. Sinyal Asli: mari kita anggap kita memiliki sinyal diskrit dengan panjang 8, misalnya: $x = [3, 7, 1, 4, 6, 2, 8, 5]$
2. Tahap 1 (Level 1): dalam level pertama, kita akan menghitung subsinyal aproksimasi dan detail.

a) Subsinyal Aproksimasi (A_1):

$$A_1 = \frac{x[0] + x[1]}{2}, \frac{x[2] + x[3]}{2}, \frac{x[4] + x[5]}{2}, \frac{x[6] + x[7]}{2}$$

$$A_1 = [5, 2.5, 4, 6.5]$$

b) Subsinyal Detail (D_1):

$$D_1 = x[0] - A_1[0], x[1] - A_1[1], x[2] - A_1[2], x[3] - A_1[3]$$

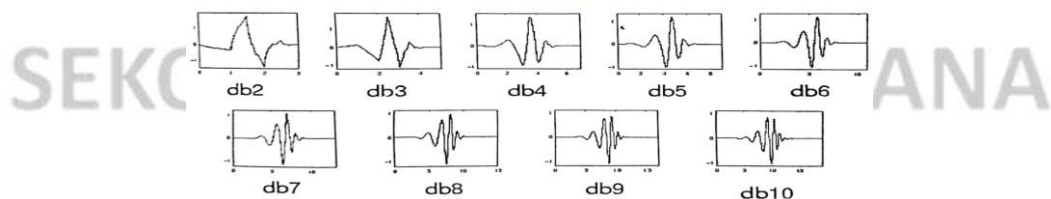
$$D_1 = [-2, 4.5, -3, -2.5]$$

3. Hasil Level 1: setelah menghitung subsinyal aproksimasi dan *detail* pada Level 1, kita mendapatkan dua hasil, yaitu subsinyal aproksimasi A_1 dan subsinyal detail D_1 .

Dalam contoh ini, kita melakukan dekomposisi pada level 1 untuk mendapatkan aproksimasi A_1 dan detail D_1 dari sinyal asli x . Perhitungan dilakukan dengan mengikuti aturan-aturan dasar dari transformasi *Wavelet Haar*, yaitu menghitung rata-rata dan perbedaan antara elemen-elemen sinyal.

2.2.3.2 Wavelet Daubechies

Wavelet Daubechies memiliki nama pendek db dan untuk order N dituliskan dengan dbN. Order *Wavelet Daubechies* $N = 1$ disebut juga *Haar*, $2 \leq N \leq 45$. Panjang *Wavelet Daubechies* adalah $2N$ (R Rizal, 2012). Sebagai contoh db2 memiliki tapis dengan Panjang 4.



Gambar 2.2 Fungsi penyekalaan pada *Wavelet Daubechies*

Gambar 2.2 menunjukkan panjang tapis dari beberapa jenis *wavelet Daubechies* dari $N = 2$ sampai dengan $N = 10$. Dengan menggunakan persamaan

(3), (5) dan (9) bisa ditentukan nilai koefisien penyekala h_k untuk pelewat tapis rendah dengan k = nilai diskret dari sumbu waktu t . Nilai k maksimum adalah sama dengan $(N - 1)$ atau dinyatakan sebagai $0 \leq k \leq (N - 1)$.

Untuk $k = 4$,

$$h_0 + h_1 + h_2 + h_3 = \sqrt{2} \dots \dots \dots (19)$$

$$h^2(0) + h^2(1) + h^2(2) + h^2(3) = 1 \dots \dots \dots (20)$$

$$h_0 h_2 + h_1 h_3 = 0 \dots \dots \dots (21)$$

Maka koefisien fungsi penyekalaan untuk *Daubechies* order 2 dengan panjang tapis 4 adalah:

$$h_0 = \frac{1 + \sqrt{2}}{4\sqrt{2}}, h_1 = \frac{3 + \sqrt{3}}{4\sqrt{2}}, h_2 = \frac{3 - \sqrt{3}}{4\sqrt{2}}, h_3 = \frac{1 - \sqrt{3}}{4\sqrt{2}} \dots \dots \dots (22)$$

$$h_0 = \frac{1 + \sqrt{3}}{4\sqrt{2}} = 0,4829 \dots \dots \dots (23)$$

$$h_1 = \frac{3 + \sqrt{3}}{4\sqrt{2}} = 0,83651 \dots \dots \dots (24)$$

$$h_2 = \frac{3 - \sqrt{3}}{4\sqrt{2}} = 0,2241 \dots \dots \dots (26)$$

$$h_3 = \frac{1 - \sqrt{3}}{4\sqrt{2}} = -0,1294 \dots \dots \dots (27)$$

Sedangkan nilai koefisien fungsi *wavelet g* adalah:

$$g[N - 1 - n] = (-1)^n \cdot h(n) \dots \dots \dots (28)$$

$$g_0 = h_3 \dots \dots \dots (29)$$

$$g_1 = -h_2 \dots \dots \dots (30)$$

$$g_2 = h_1 \dots \dots \dots (31)$$

$$g_3 = -h_0 \dots \dots \dots (32)$$

Wavelet Daubechies order 3 dengan panjang filter 6, ditentukan dua parameter yakni α dan β . Pada wavelet jenis ini, nilai dari koefisien dihasilkan dari $\alpha = 1,3598$ dan $\beta = -0,7821$. Maka nilai dari koefisien penyekalaan atau koefisien tapis pelewat rendah untuk db3 sebagai berikut:

$$h(0) = \frac{[(1 + \cos(\alpha) + \sin(\alpha))(1 - \cos(\beta) - \sin(\beta) + 2\sin(\beta)\cos(\alpha))]}{4\sqrt{2}}$$

$$= 0,33267 \dots \dots \dots (33)$$

$$h(1) = \frac{[(1 + \cos(\alpha) - \sin(\alpha))(1 + \cos(\beta) - \sin(\beta) - 2\sin(\beta)\cos(\alpha))]}{4\sqrt{2}} \\ = 0,33267 \dots \dots \dots (34)$$

$$h(2) = \frac{[(1 + \cos(\alpha - 3) + \sin(\alpha - \beta))]}{2\sqrt{2}} = 0,45987 \dots \dots \dots (35)$$

$$h(3) = \frac{[(1 + \cos(\alpha - 3) - \sin(\alpha - \beta))]}{2\sqrt{2}} = 0,13501 \dots \dots \dots (36)$$

$$h(4) = \frac{1}{\sqrt{2}} - h(0) - h(2) = -0,085441 \dots \dots \dots (37)$$

$$h(5) = \frac{1}{\sqrt{2}} - h(1) - h(3) = 0,035226 \dots \dots \dots (38)$$

Sedangkan nilai koefisien wavelet untuk db3 adalah:

$$g(0) = -h(5) \dots \dots \dots (39)$$

$$g(1) = -h(4) \dots \dots \dots (40)$$

$$g(2) = -h(3) \dots \dots \dots (41)$$

$$g(3) = h(2) \dots \dots \dots (42)$$

$$g(4) = -h(1) \dots \dots \dots (43)$$

$$g(5) = h(0) \dots \dots \dots (44)$$

Tabel 2.2 Menunjukkan koefisien skala *Daubechies* untuk db1 dan db2.

Tabel 2.2 Koefisien skala *Daubechies*

	DB1	DB2
h(0)	0,7071	0,483
h(1)	0,7071	0,8365
h(2)		0,2241
h(3)		-0,1294

Perhitungan *Wavelet Daubechies*

Perhitungan transformasi *wavelet* menggunakan *wavelet Daubechies* melibatkan serangkaian langkah matematis yang kompleks. Salah satu metode yang umum digunakan adalah dengan menggunakan *filter low-pass* (LP) dan *high-pass* (HP)

untuk melakukan dekomposisi dan rekonstruksi pada sinyal. Dekomposisi sinyal diskrit dengan panjang 8 menggunakan *wavelet Daubechies* dengan orde 4 (DB4).

Dekomposisi transformasi *Daubechies*:

1. Sinyal asli diasumsikan memiliki sinyal diskrit dengan panjang 8: $x = [3, 7, 1, 4, 6, 2, 8, 5]$. Penyaringan dengan penyaring koefisien *Low-Pass* (LP) dan koefisien *High-Pass* (HP) dari *wavelet Daubechies* db4 memiliki koefisien yang telah ditentukan. Misalnya, untuk db4, filter LP dan HP bisa diwakili oleh koefisien $h_0 = [0,482962913145, 0,836516303738, 0,224143868042, -0,129409522551]$ dan koefisien $h_1 = [-0,129409522551, -0,224143868042, 0,836516303738, -0,482962913145]$ secara berturut-turut.
2. Melakukan konvolusi sinyal asli dengan filter LP dan HP untuk mendapatkan subsinyal aproksimasi (A) dan detail (D).
 - a) Subsinyal Aproksimasi (A1): $A_1 = x * h_0$
 - b) Subsinyal *Detail* (D1): $D_1 = x * h_1$
3. Menurunkan jumlah sampel dalam subsinyal aproksimasi dan *detail*. Misalnya, dengan menyimpan setiap nilai kedua dari subsinyal.

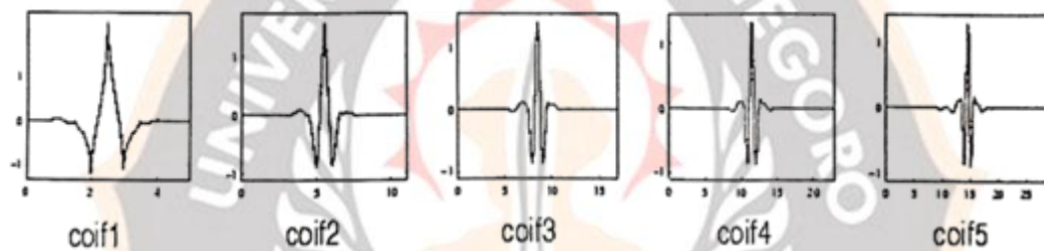
Rekonstruksi transformasi *Daubechies*:

1. Interpolasi, mengembalikan jumlah sampel pada tingkat resolusi yang lebih tinggi dengan melakukan interpolasi pada subsinyal aproksimasi dan *detail*.
2. Penggabungan subsinyal, menggabungkan subsinyal aproksimasi dan *detail* yang direkonstruksi untuk mendapatkan sinyal asli yang direkonstruksi.

Perhitungan *Wavelet Daubechies* melibatkan filter dan operasi matematis seperti konvolusi, pengambilan sampel yang lebih rendah, dan interpolasi. Langkah-langkah ini memungkinkan dekomposisi dan rekonstruksi sinyal pada berbagai tingkat resolusi menggunakan *wavelet Daubechies* tertentu.

2.2.3.3 Wavelet *Coiflets*

Wavelet *Coiflets* memiliki nama *Coif* dan untuk order N dituliskan dengan *CoifN*. Panjang tapis wavelet *Coiflets* adalah $6N$. sebagai contoh *Coif1* memiliki tapis dengan panjang 6 (R Rizal, 2012). Citra 2.3 menunjukkan fungsi penyekalaan dan dan fungsi wavelet *Coiflets*. Sedangkan fungsi skala *Coiflets* yang memiliki nama pendek *Coif* untuk order N dituliskan dengan *CoifN*. Wavelet *Coiflets* memiliki order $N = 1, 2, 3, \dots, 5$. Fungsi wavelet ini mempunyai $2N$ harga momen menuju 0 dan fungsi skala mempunyai $2N-1$ harga menuju 0. Kedua fungsi mempunyai hubungan dari panjang $6N-1$.



Gambar 2.3 Wavelet *Coiflets*

Pada citra 2.3 dapat pula ditunjukkan panjang tapis dari wavelet *Coiflets* dari $N = 1$ sampai $N = 5$. Nilai koefisien tapis pelewat rendah untuk wavelet *Coiflets* order 1 adalah:

$$h_0 = \frac{1 - \sqrt{7}}{16\sqrt{2}} = -0,0727 \dots \dots \dots (45)$$

$$h_1 = \frac{5 - \sqrt{7}}{16\sqrt{2}} = 0,3378 \dots \dots \dots (46)$$

$$h_2 = \frac{7 + \sqrt{7}}{8\sqrt{2}} = 0,8526 \dots \dots \dots (47)$$

$$h_3 = \frac{7 - \sqrt{7}}{8\sqrt{2}} = 0,3849 \dots \dots \dots (48)$$

$$h_4 = \frac{1 - \sqrt{7}}{16\sqrt{2}} = -0,0727 \dots \dots \dots (49)$$

$$h_5 = \frac{-3 + \sqrt{7}}{16\sqrt{2}} = -0,0157 \dots \dots \dots (50)$$

Perhitungan Wavelet Coiflet

Perhitungan transformasi menggunakan *Wavelet Coiflets* melibatkan serangkaian langkah matematis yang mirip dengan perhitungan transformasi menggunakan *wavelet Daubechies*. Saya akan memberikan citraan umum mengenai proses transformasi *Wavelet Coiflets*. Dekomposisi sinyal diskrit dengan panjang 8 menggunakan *Coiflets-1* sebagai contoh. Dekomposisi Transformasi *Coiflets*:

1. Sinyal asli memiliki sinyal diskrit dengan panjang 8: $x = [3, 7, 1, 4, 6, 2, 8, 5]$.
2. Penyaringan dengan penyaring koefisien *Low-Pass* (LP) dan koefisien *High-Pass* (HP) dari *Coiflets-1* memiliki koefisien yang telah ditentukan. Koefisien-koefisien ini akan digunakan untuk menghitung subsinyal aproksimasi (A) dan detail (D).
3. Melakukan konvolusi sinyal asli dengan filter LP dan HP untuk mendapatkan subsinyal aproksimasi (A1) dan detail (D1).
4. Pengambilan sampel yang lebih rendah dalam subsinyal untuk mencapai resolusi yang lebih rendah.

Rekonstruksi transformasi *Daubechies*:

1. Interpolasi yaitu mengembalikan jumlah sampel pada tingkat resolusi yang lebih tinggi dengan melakukan interpolasi pada subsinyal aproksimasi dan detail.
2. Penggabungan subsinyal yaitu menggabungkan subsinyal aproksimasi dan detail yang direkonstruksi untuk mendapatkan sinyal asli yang direkonstruksi.

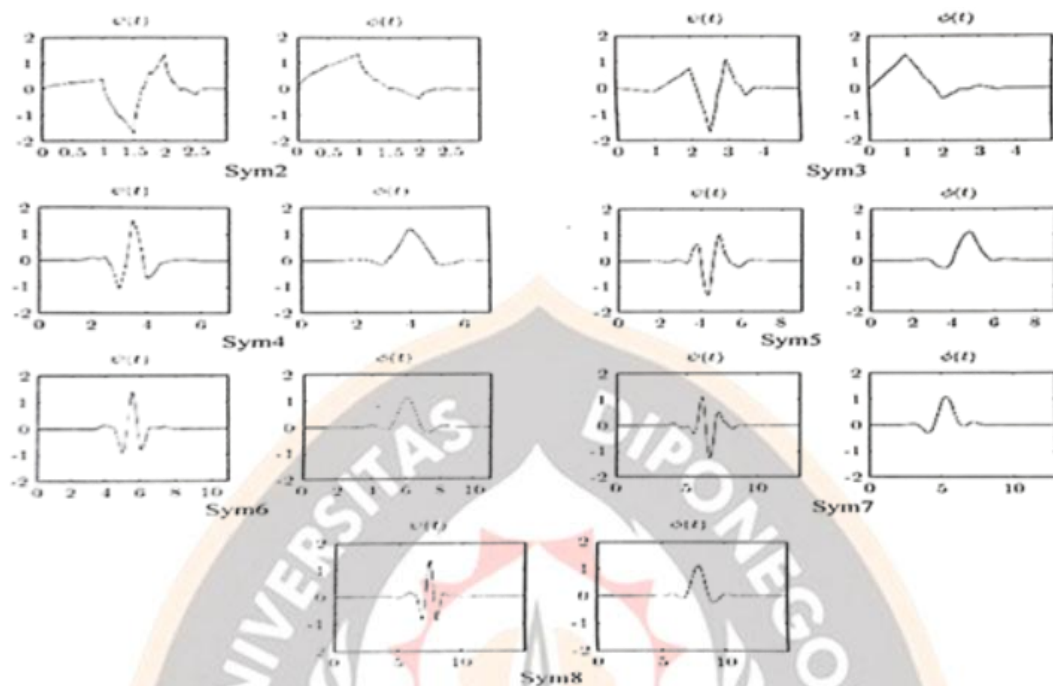
Proses ini memungkinkan dekomposisi dan rekonstruksi sinyal pada berbagai tingkat resolusi menggunakan *wavelet Coiflets-1*. Setiap iterasi akan menghasilkan tingkat resolusi yang lebih rendah dari sinyal asli dan memungkinkan analisis sinyal pada berbagai skala frekuensi. Perhitungan spesifiknya melibatkan operasi matematis seperti konvolusi, pengambilan sampel yang lebih rendah, dan interpolasi dengan memanfaatkan koefisien dari fungsi penyekalaan dan *wavelet Coiflets-1*.

2.2.3.4 Wavelet *Symlet*

Symlet adalah salah satu jenis *wavelet* yang orthogonal dan terdukung secara kompak, yang dikenalkan pertama kali oleh I. *Daubechies* sebagai modifikasi terhadap keluarga *wavelet* db. *Symlet* memiliki sifat hampir sama simetri dan memiliki tingkat asimetris terkecil. *Wavelet Symlet* adalah keluarga *wavelet* yang memiliki kemiripan dengan *wavelet Daubechies*. Mereka dibangun dengan menggunakan fungsi penyekalaan (*scaling function*) dengan rata-rata nol, namun memiliki karakteristik yang berbeda dalam ruang frekuensi. *Symlet*, singkatan dari "*symmetric orthogonal*", menunjukkan bahwa *wavelet* ini memiliki sifat simetri dalam domain waktu dan frekuensi (R Rizal, 2012).

Seperti halnya *Daubechies Wavelet*, *Symlet* juga memiliki banyak variasi yang diberi label dengan angka, misalnya *Symlet 2*, *Symlet 3*, dan seterusnya. Setiap jenis *Symlet* memiliki orde yang berbeda dan karakteristik khususnya sendiri dalam hal jumlah koefisien dan kemampuan untuk merepresentasikan sinyal pada berbagai tingkat resolusi. Seperti halnya *wavelet* lainnya, *Symlet* juga digunakan dalam berbagai aplikasi seperti kompresi citra, pengolahan sinyal, analisis data, dan pengolahan citra. Pemilihan jenis *Symlet* yang tepat tergantung pada kebutuhan analisis dan karakteristik dari data yang ditangani. Sama seperti *Daubechies*, *Symlet* dapat digunakan dalam transformasi *wavelet* diskrit untuk menganalisis sinyal dalam berbagai skala frekuensi. Tapis-tapis penyekala yang bersesuaian dengan *Symlet* mendekati bentuk tapis-tapis *fase-linier*. Sifat-sifat *Symlet* hampir sama dengan sifat-sifat dari keluarga *wavelet* db. *Symlet* dan fungsi-fungsi penyekala untuk order 2 sampai dengan order 8 ditunjukkan pada Gambar 2.4 dibawah ini.

SEKOLAH PASCASARJANA



Gambar 2.4 Wavelet Symlet

Perhitungan Wavelet Symlet

Perhitungan transformasi menggunakan *Wavelet Symlet* melibatkan serangkaian langkah matematis yang mirip dengan perhitungan menggunakan *wavelet Daubechies* atau *Coiflets*. Saya akan memberikan citraan umum mengenai proses transformasi *Wavelet Symlet*. Dekomposisi sinyal diskrit dengan panjang 8 menggunakan *Symlet-2* sebagai contoh. Dekomposisi transformasi *Symlet*:

1. Sinyal asli memiliki sinyal diskrit dengan panjang 8: $x = [3, 7, 1, 4, 6, 2, 8, 5]$.
2. Penyaringan dengan penyaring koefisien *Low-Pass* (LP) dan koefisien *High-Pass* (HP) dari *Symlet-2* memiliki koefisien yang telah ditentukan. Koefisien-koefisien ini akan digunakan untuk menghitung subsinyal aproksimasi (A) dan detail (D).
3. Melakukan konvolusi sinyal asli dengan filter LP dan HP untuk mendapatkan subsinyal aproksimasi (A1) dan detail (D1).
4. Pengambilan sampel yang lebih rendah dalam subsinyal untuk mencapai resolusi yang lebih rendah.

Rekonstruksi transformasi *Daubechies*:

1. Interpolasi yaitu mengembalikan jumlah sampel pada tingkat resolusi yang lebih tinggi dengan melakukan interpolasi pada subsinyal aproksimasi dan *detail*.
2. Penggabungan subsinyal yaitu menggabungkan subsinyal aproksimasi dan *detail* yang direkonstruksi untuk mendapatkan sinyal asli yang direkonstruksi.

Proses ini memungkinkan dekomposisi dan rekonstruksi sinyal pada berbagai tingkat resolusi menggunakan *wavelet Symlet-2*. Setiap iterasi akan menghasilkan tingkat resolusi yang lebih rendah dari sinyal asli dan memungkinkan analisis sinyal pada berbagai skala frekuensi. Perhitungan spesifiknya melibatkan operasi matematis seperti konvolusi, *down sampling*, dan interpolasi dengan memanfaatkan koefisien dari fungsi penyekalaan dan *wavelet Symlet-2*.

2.2.4 Pola *Wavelet Haar*

Wavelet Haar merupakan metode yang lazim digunakan dalam indentifikasi obyek. Nama Haar sendiri mengacu pada *Haar wavelet*, sebuah fungsi matematika yang berbentuk kotak dan memiliki prinsip seperti pada fungsi *fourier*. Prinsip *Haar* adalah mengenali obyek berdasarkan nilai sederhana dari tekstur daging tetapi bukan merupakan nilai piksel dari *imege* obyek tersebut. Metode *Wavelet Haar* ini memiliki kelebihan yaitu komputasinya sangat cepat, karena hanya bergantung pada jumlah piksel dalam persegi, bukan setiap nilai dari sebuah citra. *Wavelet Haar* memproses citra dalam bentuk sebuah kotak persegi dengan ukuran 500 x 500 piksel.

2.2.5 Ekstraksi Ciri

Ekstraksi ciri dalam konteks pengolahan citra adalah proses mengidentifikasi, memilih dan mengubah informasi penting dari data mentah menjadi representasi numerik yang lebih sederhana dan dapat diinterpretasikan. Proses ini memungkinkan kita untuk menganalisis data secara lebih efektif atau menggunakannya sebagai masukan untuk sistem yang lebih kompleks seperti pengenalan pola atau klasifikasi. Dalam konteks pengolahan citra tekstur jenis

daging, ekstraksi ciri adalah tentang mengidentifikasi ciri-ciri yang penting dari citra tekstur jenis daging yang akan digunakan dalam analisis atau pengenalan.

2.2.6 *Gray Level Co-occurrence Matrix (GLCM)*

GLCM adalah metode statistik orde kedua yang menghitung frekuensi pasangan piksel yang memiliki tingkat keabuan yang sama dalam sebuah citra dan menerapkan pengetahuan tambahan yang diperoleh dengan menggunakan hubungan piksel spasial (Manivannan dkk., 2012). Matriks kejadian bersama menyematkan distribusi transisi skala abu-abu menggunakan informasi edge. Karena sebagian besar informasi yang diperlukan untuk menghitung nilai ambang batas tertanam dalam GLCM, ini muncul sebagai teknik yang sederhana namun efektif. Pertimbangkan I sebagai citra dengan tingkat keabuan terkuantisasi 0 hingga $L-1$, L dianggap sebagai 256. GLCM dihitung berdasarkan dua parameter: jarak relatif (d) antara pasangan piksel dan orientasi relatif (ϕ) di antara piksel tersebut. Dalam formulasi *Harris lick*, lingkungan piksel dibuat menggunakan piksel pusat dan empat piksel perifer, yang berjarak sama pada interval sudut 45° . Untuk jarak tertentu d , koordinat relatif piksel di perifer piksel pusat adalah: $(0, d)$, $(-d, d)$, $(-d, 0)$, $(0, -d)$ (Bianconi & Fernández, 2014). Nilai d dianggap sebagai satu kesatuan untuk menyederhanakan dan mengurangi kompleksitas komputasi. Posisi relatif piksel mengcitirakan ciri citra dengan baik. Misalkan $d = (a, b)$ merupakan vektor perpindahan yang menunjukkan posisi relatif dari piksel-piksel yang terletak pada koordinat (x, y) dan $(x + a, y + b)$, dimana a dan b adalah bilangan bulat. Dalam GLCM, jumlah baris dan kolom sama dengan L . Misal C adalah matriks $L \times L$, dan (i, j) elemen matriks menyatakan jumlah pasangan piksel dari citra I pada posisi relatif d dan orientasi ϕ , sehingga piksel pertama memiliki tingkat keabuan i dan piksel kedua memiliki tingkat keabuan j , yang berada dalam hubungan spasial linier. Menariknya, elemen *matriks co-occurrence* $m(a, b|d, \phi)$ terdiri dari nilai probabilitas statistik orde kedua yang mewakili transisi antara tingkat keabuan a dan b pada perpindahan tertentu d serta sudut tertentu ϕ . C dihitung untuk masing-masing dari empat arah atau sudut standar (horizontal: 00, diagonal: 450, vertikal:

900, dan anti-diagonal: 1350) dan jarak d . Secara keseluruhan, GLCM dihitung dengan mengambil rata-rata dari semua arah sebagai (Pare et al., 2017):

$$GLCM = \frac{1}{4}([C_{d,0^\circ}] + [C_{d,45^\circ}] + [C_{d,90^\circ}] + [C_{d,135^\circ}]) \dots \dots \dots (51)$$

Pertimbangkan nilai ambang tunggal T untuk mempartisi matriks *co-occurrence* citra menjadi empat segmen BB , BF , FF , dan FB . Wilayah BB mengcitarkan perubahan tingkat keabuan di dalam latar belakang. Wilayah FF mewakili transisi tingkat abu-abu di dalam latar depan. Wilayah BF dan FB menunjukkan transisi bersama melintasi batas antara latar belakang dan latar depan. Berdasarkan frekuensi pasangan piksel, GLCM dapat menghitung ciri yang paling relevan dari citra seperti kontras, korelasi, energi, entropi, varians, dan momen perbedaan terbalik (Haralick, R., Shanmugam, K., 1973). Untuk bi-level *thresholding*, satu nilai *threshold* dihitung menggunakan:

$$T = \frac{1}{n} \sum_{m=0}^{L-1-q} \sum_{n=m+q}^{L-1} \left(\frac{m+n}{2} \right) GLCM(m, n) \dots \dots \dots (52)$$

Dimana,

$$n = \sum_{m=0}^{L-1-q} \sum_{n=m+q}^{L-1} GLCM(m, n) \dots \dots \dots (53)$$

Konsep bi-level *thresholding* dapat dimodifikasi menjadi multilevel *thresholding* dengan menggunakan beberapa nilai *threshold* direpresentasikan sebagai:

$$T_1 = \arg \max \left(\frac{1}{n_1} \sum_{m=0}^{q_1} \sum_{n=q_1+1}^{q_2} \frac{m+n}{2} GLCM(m, n) \right) \dots \dots \dots (54)$$

$$T_2 = \arg \max \left(\frac{1}{n_2} \sum_{m=q_1+1}^{q_2} \sum_{n=q_2+1}^{q_2} \frac{m+n}{2} GLCM(m, n) \right) \dots \dots \dots (55)$$

$$T_{k-1} = \arg \max \left(\frac{1}{n_{k-1}} \sum_{m=q_{k-2}+1}^{q_2} \sum_{n=q_{k-1}+1}^{q_2} \frac{m+n}{2} GLCM(m, n) \right) \dots (56)$$

Dimana:

$$n_1 = \sum_{m=0}^{q_1} \sum_{n=q_1+1}^{q_2} GLCM(m, n) \dots \dots \dots (57)$$

$$n_2 = \sum_{m=q_1+1}^{q_2} \sum_{n=q_2+1}^{q_3} GLCM(m, n) \dots \dots \dots (58)$$

$$n_{k-1} = \sum_{m=q_{k-2}+1}^{q_{k-1}} \sum_{n=q_{k-1}+1}^{L-1} GLCM(m, n) \dots \dots \dots (59)$$

Besaran tepi diwakili oleh $q_1, q_2, \dots, q_{k-1}$ sesuai dengan nilai ambang batas. Magnitudo tepi ini dioptimalkan untuk menghasilkan nilai ambang batas terbaik yang sesuai. Kisaran penjumlahan menghitung ambang batas di area tertentu di GLCM yang terletak pada $n - m \geq q$ yang dibatasi. Ambang batas yang diperoleh optimal ketika penjumlahan atas q dimaksimalkan. Dalam GLCM, pasangan piksel yang besaran tepinya lebih besar atau sama dengan q terlibat dalam perhitungan ambang batas. g didefinisikan sebagai jumlah total pasangan piksel dalam GLCM dengan besaran tepi lebih besar atau sama dengan q . Pemilihan q sangat penting untuk meningkatkan hasil thresholding karena nilai q yang optimum akan mendapatkan luas perhitungan pada batas objek. Selanjutnya, perhitungan dibatasi pada segitiga karena simetri GLCM. Untuk peningkatan akurasi dan kecepatan perhitungan, algoritma cerdas yang disebut *Cuckoo Search* (CS) digunakan untuk mendapatkan ambang optimal dengan memaksimalkan fungsi tujuan:

$$[T_1, T_2, \dots, T_{k-1}] = \arg \max \{f(q_1, q_2, \dots, q_{k-1})\} \dots \dots \dots (60)$$

Perhitungan *Gray Level Co-occurrence Matrix* (GLCM)

Perhitungan GLCM melibatkan langkah-langkah sebagai berikut:

1. Pemilihan sudut dan jarak yaitu tentukan arah (misalnya, 0° , 45° , 90° , dan 135°) serta jarak yang ingin peneliti gunakan untuk menghitung matriks. Sudut:
 - 1) 0° (Horizontal) yaitu membandingkan piksel ke kanan.
 - 2) 45° (Diagonal Kanan Bawah) yaitu membandingkan piksel ke kanan bawah.
 - 3) 90° (Vertikal): Membandingkan piksel ke bawah.

4) 135^0 (diagonal kiri bawah) yaitu membandingkan piksel ke kiri bawah.

Jarak: Jarak dapat diukur dalam jumlah piksel antara pasangan piksel yang dibandingkan. Jarak 1: sekali piksel terpisah. Jarak 2: dua piksel terpisah, dan seterusnya.

2. Pembuatan matriks, dalam setiap sudut dan jarak untuk setiap piksel dalam citra, hitung nilai pasangan piksel dengan intensitas yang sesuai. Misalnya, jika jarak adalah 1 piksel ke kanan, hitung seberapa sering pasangan piksel dengan perbedaan intensitas tertentu muncul dalam sudut itu. Misalkan kita memiliki citra *grayscale* sebagai berikut: citra Input: $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 1 & 2 \\ 1 & 2 & 1 \end{bmatrix}$. Kita akan menghitung GLCM untuk jarak 1 piksel ke kanan (sudut 0^0).

- 1) Pilih sudut dan jarak misalkan sudut yang dipilih adalah horizontal (0^0) dengan jarak 1 piksel ke kanan.

- 2) Hitung GLCM lakukan iterasi melalui setiap piksel dalam citra dan catat pasangan intensitas pada jarak dan sudut yang ditentukan.

Misalnya, untuk citra di atas, untuk jarak 1 piksel ke kanan, pasangan intensitas pikselnya adalah: (1, 2), (2, 1), (2, 1), (1, 2), (2, 1), (1, 2).

Kemudian, kita hitung frekuensi kemunculan pasangan ini dalam matriks GLCM: $\begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix}$. Keterangan: pasangan (1, 2) muncul 2 kali. Pasangan (2, 1) juga muncul 2 kali.

Pasangan lainnya tidak muncul pada jarak dan sudut yang dipilih.

3. Normalisasi GLCM dengan cara membagi setiap entri matriks dengan jumlah total entri dalam matriks. Hal ini dilakukan untuk menghasilkan distribusi probabilitas dari pasangan intensitas piksel. Langkah-langkah normalisasi GLCM melibatkan antara lain:

- 1) Menghitung jumlah total kemunculan yaitu jumlahkan semua nilai dalam matriks GLCM untuk mendapatkan jumlah total kemunculan pasangan piksel.

- 2) Membagi setiap elemen dengan jumlah total yaitu bagi setiap elemen dalam matriks GLCM dengan jumlah total yang dihitung pada langkah pertama.

Contoh GLCM sebelum normalisasi sebagai berikut: $\begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix}$. Langkah-langkah normalisasi sebagai berikut: jumlah total kemunculan pasangan piksel

adalah $0 + 2 + 2 + 0 = 4$. Normalisasi dengan cara bagi setiap elemen matriks GLCM dengan jumlah total (4). GLCM setelah normalisasi sebagai berikut: $[[0/4, 2/4], [2/4, 0/4]]$. GLCM setelah normalisasi sebagai berikut: $[[0.0, 0.5], [0.5, 0.0]]$.

4. Ekstraksi ciri dari GLCM yang dihasilkan, sejumlah ciri statistik dapat diekstraksi seperti energi, kontras, korelasi, homogenitas, entropi, yang mencerminkan pola atau tekstur dalam citra. Beberapa ciri yang umum diekstraksi dari GLCM antara lain:

- 1) Energi yaitu jumlah kuadrat dari nilai-nilai dalam GLCM, mengukur homogenitas tekstur. Semakin tinggi nilainya, semakin homogen tekstur citra.
- 2) Kontras yaitu mengukur perbedaan antara intensitas yang berdekatan. Nilai yang lebih tinggi menunjukkan perbedaan yang lebih tajam antara piksel-piksel tetangga.
- 3) Korelasi yaitu mengukur sejauh mana intensitas antara pasangan piksel terkait. Nilai mendekati 1 menunjukkan keterkaitan yang kuat antara piksel-piksel.
- 4) Homogenitas yaitu mengukur sejauh mana distribusi intensitas seragam dalam citra. Semakin tinggi nilainya, semakin seragam tekstur citra.
- 5) Entropi yaitu mengukur tingkat ketidakpastian dalam distribusi intensitas. Nilai yang lebih tinggi menunjukkan tekstur yang lebih kompleks atau acak.

Untuk mengekstraksi ciri dari GLCM yang sudah dinormalisasi, peneliti perlu melakukan perhitungan sesuai dengan rumus yang sesuai dengan masing-masing ciri. Sebagai contoh, untuk menghitung beberapa ciri sebagai berikut:

- 1) Energi:

$$Energi = \sum_{i=1}^N \sum_{j=1}^N GLCM(i, j)^2 \dots \dots \dots (61)$$

Perhitungan matematis energi dari sebuah citra grayscale kecil (3x3) menggunakan rumus energi: langkah 1: membaca citra dan membentuk matrik GLCM. Misalkan citra *grayscale* yang ingin dianalisis memiliki intensitas piksel sebagai berikut:

[100, 120, 140]

[90, 110, 130]

[80, 100, 120]

Ubah citra menjadi matrik GLCM (*Gray Level Co-occurrence Matrix*) untuk jarak dan arah tertentu. Dalam contoh ini, misalkan jarak = 1 dan arah = 0 (horizontal). Matrik GLCM akan menjadi:

[0, 1, 0]

[1, 2, 1]

[0, 1, 0]

Langkah 2: menghitung energi. Gunakan rumus berikut untuk menghitung energi:

$$Energi = \text{sum}(glcm)$$

Dimana: GLCM adalah matrik GLCM. Perhitungan:

$$Energi = 0 + 1 + 0 + 1 + 2 + 1 + 0 + 1 + 0 = 6$$

Hasil: nilai energi untuk citra ini adalah 6. Nilai energi ini menunjukkan tingkat keseragaman dalam citra. Semakin tinggi nilai energi, semakin seragam citranya.

2) Kontras:

$$Kontras = \sum_{i=1}^N \sum_{j=1}^N (i - j)^2 \cdot GLCM(i, j) \dots \dots \dots (62)$$

Perhitungan matematis kontras dari sebuah citra:

a. Kontras rata-rata, metode ini menghitung kontras rata-rata dari seluruh piksel dalam citra. Rumusnya adalah:

$$Kontras \text{ rata - rata} = (1/N) * \sum I_i - \mu)^2$$

Dimana:

N adalah total jumlah piksel dalam citra

I_i adalah intensitas piksel ke-i

μ adalah intensitas rata-rata citra

Misalkan kita memiliki citra grayscale dengan ukuran 3x3 dan intensitas pikselnya sebagai berikut:

[100, 120, 140]

[90, 110, 130]

[80, 100, 120]

Hitung intensitas rata-rata:

$$\mu = (100 + 120 + 140 + 90 + 110 + 130 + 80 + 100 + 120) / 9 = 110$$

Hitung perbedaan kuadrat intensitas:

$$[100 - 110]^2 = 100$$

$$[120 - 110]^2 = 100$$

$$[140 - 110]^2 = 900$$

$$[90 - 110]^2 = 400$$

$$[110 - 110]^2 = 0$$

$$[130 - 110]^2 = 400$$

$$[80 - 110]^2 = 900$$

$$[100 - 110]^2 = 100$$

$$[120 - 110]^2 = 100$$

Hitung kontras rata-rata:

$$\text{Kontras rata-rata} = (1 / 9) * (100 + 100 + 900 + 400 + 0 + 400 + 900 + 100 + 100) = 400 / 9 \approx 44.44$$

- b. Kontras rata-rata, metode ini menghitung kontras untuk setiap piksel dalam citra dengan mempertimbangkan piksel-piksel di sekitarnya.

Rumus umum untuk kontras lokal adalah:

$$\text{Kontras lokal}(p) = \sum (I_i - I_p)^2 / (N - 1)$$

Dimana:

p adalah piksel yang dihitung kontrasnya

I_i adalah intensitas piksel tetangga

I_p adalah intensitas piksel p

N adalah total jumlah piksel tetangga

Misalkan kita ingin menghitung kontras lokal untuk piksel (1, 1) pada citra grayscale 3x3 di atas. Piksel tetangga dari (1, 1) adalah:

[100, 120]

[90, 110]

Hitung kontras lokal untuk (1, 1):

$$\text{Kontras lokal (1, 1)} = ((120 - 100)^2 + (90 - 110)^2) / (2 - 1) = 100 / 1 = 100.$$

3) Korelasi:

$$\text{Korelasi} = \frac{\sum_{i=1}^N \sum_{j=1}^N (i - \mu)(j - \mu) \cdot GLCM(i, j)}{\sigma^2} \dots \dots \dots (63)$$

Perhitungan matematis korelasi dari sebuah citra *grayscale* kecil (3x3) menggunakan rumus korelasi rata-rata: langkah 1: membaca citra dan membentuk matrik GLCM. Misalkan citra *grayscale* yang ingin dianalisis memiliki intensitas piksel sebagai berikut:

[100, 120, 140]

[90, 110, 130]

[80, 100, 120]

Ubah citra menjadi matrik GLCM (*Gray Level Co-occurrence Matrix*) untuk jarak dan arah tertentu. Dalam contoh ini, misalkan jarak = 1 dan arah = 0 (horizontal). Matrik GLCM akan menjadi:

[[0, 1, 0],

[1, 2, 1],

[0, 1, 0]]

Langkah 2: menghitung rata-rata intensitas dan variansi. Hitung rata-rata intensitas (μ_i) untuk setiap baris matrik GLCM.

$$\mu_i = [1, 2, 1]$$

Hitung rata-rata intensitas (μ_j) untuk setiap kolom matrik GLCM.

$$\mu_j = [0, 1, 2]$$

Hitung varian intensitas (σ_i) untuk setiap baris matrik GLCM.

$$\sigma_i = [1, 1, 1]$$

Hitung varian intensitas (σ_j) untuk setiap kolom matrik GLCM.

$$\sigma_j = [1, 1, 1]$$

Langkah 3: menghitung korelasi rata-rata. Gunakan rumus berikut untuk menghitung korelasi rata-rata:

$$\begin{aligned} & \text{Korelasi_rata_rata} \\ &= (\text{sum}(\text{glcm} * (\text{mu_i}[:, \text{np.newaxis}] - \text{mu_j}) * (\text{mu_j} \\ & - \text{mu_i}) / (\text{sigma_i}[:, \text{np.newaxis}] * \text{sigma_j}))) / N \end{aligned}$$

Dimana:

GLCM adalah matrik GLCM

mu_i dan mu_j adalah vektor rata-rata intensitas baris dan kolom

sigma_i dan sigma_j adalah vektor varian intensitas baris dan kolom

N adalah total jumlah piksel

Dalam contoh ini, $N = 9$ (jumlah elemen dalam matrik GLCM)

Perhitungan:

$$\begin{aligned} & \text{Korelasi_rata_rata} \\ &= ((1 * (1 - 2) * (2 - 1) + 2 * (2 - 2) * (2 \\ & - 1) + 1 * (1 - 2) * (2 - 1)) / 9) \\ & \text{Korelasi_rata_rata} \approx 0.222 \end{aligned}$$

Hasil: nilai korelasi rata-rata untuk citra ini adalah 0.222. Nilai ini menunjukkan bahwa terdapat korelasi positif yang lemah antara intensitas piksel yang berdekatan dalam arah horizontal.

4) Homogenitas:

$$\text{Homogenitas} = \sum_{i=1}^N \sum_{j=1}^N \frac{\text{GLCM}(i, j)}{1 + |i - j|} \dots \dots \dots (64)$$

Perhitungan matematis homogenitas dari sebuah citra *grayscale* Kecil (3x3).

Misalkan citra *grayscale* dengan intensitas piksel sebagai berikut:

[100, 120, 140]

[90, 110, 130]

[80, 100, 120]

Metode 1: homogenitas rata-rata. Langkah 1: menghitung intensitas rata-rata (μ). Jumlahkan semua nilai intensitas piksel:

$$100 + 120 + 140 + 90 + 110 + 130 + 80 + 100 + 120 = 890$$

Bagikan total dengan jumlah piksel (9): $\mu = 890 / 9 = 98.89$. Langkah 2: menghitung perbedaan kuadrat intensitas. Hitung selisih antara intensitas setiap piksel (I) dengan intensitas rata-rata (μ):

$$\begin{aligned} &[1,11; 21,11; 41,11] \\ &[-8,89; 11,11; 31,11] \\ &[-18,89; 1,11; 21,11] \end{aligned}$$

Kuadratkan selisih tersebut:

$$\begin{aligned} &[1,2321; 444,4421; 1687,2321] \\ &[79,3841; 123,21; 967,2321] \\ &[355,6481; 1,2321; 444,4421] \end{aligned}$$

Langkah 3: menghitung homogenitas rata-rata (H). Jumlahkan semua nilai perbedaan kuadrat intensitas:

$$1232,1 + 444,4421 + 1687,2321 + 79,3841 + 123,21 + 967,2321 + 355,6481 + 1,2321 + 444,4421 = 5369,4593$$

Bagikan total dengan jumlah piksel (9): $H = 5369,4593 / 9 = 596,6066$.

Metode 2: homogenitas variansi (H_v). Langkah 1: sama dengan langkah 1 pada homogenitas rata-rata. Langkah 2: menghitung variansi intensitas (σ^2). Hitung selisih antara intensitas setiap piksel (I) dengan intensitas rata-rata (μ):

$$\begin{aligned} &[1,11; 21,11; 41,11] \\ &[-8,89; 11,11; 31,11] \\ &[-18,89; 1,11; 21,11] \end{aligned}$$

Kuadratkan selisih tersebut:

$$\begin{aligned} &[1,2321; 444,4421; 1687,2321] \\ &[79,3841; 123,21; 967,2321] \\ &[355,6481; 1,2321; 444,4421] \end{aligned}$$

Hitung rata-rata dari nilai perbedaan kuadrat intensitas:

$$\sigma^2 = (1,2321 + 444,4421 + 1687,2321 + 79,3841 + 123,21 + 967,2321 + 355,6481 + 1,2321 + 444,4421) / 9 = 140,2897$$

Langkah 3: Menghitung Homogenitas Variansi (H_v)

$$H_v = 1 / (1 + \sigma^2) = 1 / (1 + 140,2897) = 0,00713$$

5) Entropi:

$$Entropi = \sum_{i=1}^N \sum_{j=1}^N GLCM(i,j) \cdot \log(GLCM(i,j)) \dots \dots \dots (65)$$

Perhitungan matematis entropi dari sebuah citra *grayscale* Kecil (3x3).

Misalkan citra grayscale dengan intensitas piksel sebagai berikut:

[100, 120, 140]

[90, 110, 130]

[80, 100, 120]

Langkah 1: menghitung distribusi probabilitas citra. Hitung jumlah piksel untuk setiap nilai intensitas:

Intensitas	Citra
100	2
110	1
120	2
130	1
140	1

Bagikan citra dengan total jumlah piksel (9) untuk mendapatkan probabilitas:

Intensitas	Probabilitas
100	0,2222
110	0,1111
120	0,2222
130	0,1111
140	0,1111

Langkah 2: menghitung entropi (H). Gunakan rumus entropi *Shannon*:

$$H = -\sum(p \cdot \log_2(p))$$

Dimana:

p adalah probabilitas untuk setiap nilai intensitas

$\log_2$ adalah logaritma basis 2

Hasil: entropi: 1,84644.

Penjelasan nilai entropi:

Nilai entropi tinggi menunjukkan bahwa citra memiliki banyak nilai intensitas yang berbeda dan distribusi probabilitas yang lebih merata. Citra dengan entropi tinggi umumnya terlihat lebih kompleks dan bertekstur. Nilai entropi rendah menunjukkan bahwa citra memiliki sedikit nilai intensitas yang berbeda dan distribusi probabilitas yang tidak

merata. Citra dengan entropi rendah umumnya terlihat lebih seragam dan datar.

2.3 *k-Nearest Neighbors (k-NN)*

k-Nearest Neighbors (k-NN) adalah salah satu algoritma yang digunakan dalam pembelajaran mesin untuk klasifikasi dan regresi. Ini termasuk dalam kategori algoritma pembelajaran yang disebut "algoritma pembelajaran berbasis instansi" atau "algoritma pembelajaran malas (*lazy learning*)". *k-NN* mengasumsikan bahwa data dengan ciri yang serupa cenderung memiliki label atau kelas yang sama. Ketika diberikan data baru, *k-NN* mencari titik-titik data terdekat (tetangga terdekat) dalam ruang ciri untuk menentukan kelas atau nilai yang sesuai.

k-NN menggunakan metrik jarak seperti *Euclidean* atau metrik jarak lainnya untuk mengukur seberapa dekat atau jauhnya suatu titik data dengan titik lainnya dalam ruang citra. Parameter K dalam *k-NN* menentukan jumlah tetangga terdekat yang akan digunakan untuk membuat prediksi. Nilai K harus dipilih dengan hati-hati karena dapat mempengaruhi kinerja model. Untuk klasifikasi, *k-NN* melakukan mayoritas citra tekstur jenis daging dari K tetangga terdekatnya untuk menentukan kelas dari data yang diprediksi. Untuk regresi, nilai dari tetangga terdekatnya diambil rata-ratanya untuk memprediksi nilai kontinu.

k-NN digunakan dalam berbagai aplikasi seperti klasifikasi teks, pengenalan pola, sistem rekomendasi dan banyak lagi. Pemilihan parameter yang tepat, pemrosesan data yang baik, dan pemilihan metrik jarak yang sesuai sangat penting untuk hasil yang optimal saat menggunakan *k-NN*.

Perhitungan *k-Nearest Neighbors (k-NN)*

Perhitungan *k-Nearest Neighbors (k-NN)* melibatkan langkah-langkah untuk menemukan tetangga terdekat dari data yang ingin diprediksi. Pilih metrik jarak seperti *Euclidean* atau metrik jarak lainnya untuk mengukur jarak antara titik data. Pemilihan jumlah Tetangga (K) dengan cara tentukan jumlah tetangga terdekat (nilai K) yang akan digunakan dalam model *k-NN*. Pemilihan K yang tepat dapat mempengaruhi kinerja model. Penghitungan jarak, dengan cara hitung jarak antara

titik data yang ingin diprediksi dengan setiap titik data dalam *dataset* latih menggunakan metrik jarak yang dipilih. Identifikasi tetangga terdekat dengan cara pilih K titik data terdekat berdasarkan jarak yang telah dihitung sebelumnya. Titik-titik ini akan menjadi tetangga terdekat dari data yang ingin diprediksi. Prediksi kelas atau nilai untuk klasifikasi, gunakan mayoritas citra tekstur jenis daging dari K tetangga terdekat untuk menentukan kelas dari data yang ingin diprediksi. Evaluasi model menggunakan teknik validasi atau metrik evaluasi seperti akurasi untuk mengukur kinerja model *k-NN* pada *dataset* uji. Perhitungan *k-NN* dapat diimplementasikan dengan menggunakan berbagai alat atau bahasa pemrograman seperti *google colab research*. Penting untuk memilih metrik jarak yang sesuai dan memahami bagaimana memilih parameter K agar model *k-NN* dapat memberikan hasil yang baik pada data yang diberikan.

2.4 Jarak *Euclidean*

Jarak *Euclidean* adalah metrik jarak yang umum digunakan dalam berbagai bidang seperti ilmu komputer, statistik, dan matematika. Ini adalah pengukuran jarak antara dua titik dalam ruang n – dimensi. Ruang n – dimensi, jarak *Euclidean* antara dua titik P dan Q didefinisikan sebagai:

$$d(P, Q) = \sqrt{\sum_{i=1}^n (P_i - Q_i)^2} \dots \dots \dots (66)$$

Di sini, P dan Q adalah dua vektor atau titik dalam ruang citra n – dimensi. P_i dan Q_i adalah koordinat dari *vector* P dan Q pada dimensi ke i dalam ruang n – dimensi. Misalnya, kita memiliki dua titik P dan Q dalam ruang (dua dimensi (dua koordinat x dan y) yaitu titik P dengan koordinat $P(x_1, y_1)$ dan titik Q dengan koordinat $Q(x_2, y_2)$. Jarak *Euclidean* antara P dan Q dalam ruang dua dimensi ($n = 2$) akan dihitung sebagai:

$$d(P, Q) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \dots \dots \dots (67)$$

Jarak *Euclidean* adalah ukuran jarak yang paling umum digunakan karena sifatnya yang intuitif dan memiliki interpretasi geometris yang mudah dipahami dalam ruang n – dimensi. Ini juga sering digunakan dalam algoritma pembelajaran

mesin seperti *k-Nearest Neighbors* (*k-NN*) untuk mengukur kedekatan antara titik-titik data.

Perhitungan *Euclidean*

Perhitungan jarak *Euclidean* antara dua titik dalam ruang n – dimensi melibatkan langkah-langkah untuk menghitung jarak berdasarkan koordinat atau nilai dari setiap dimensi. Misalkan kita memiliki dua vector P dan Q dalam ruang n – dimensi:

$$P = (p_1, p_2, \dots, p_n) \dots \dots \dots (68)$$

$$Q = (q_1, q_2, \dots, q_n) \dots \dots \dots (69)$$

Langkah-langkah untuk menghitung jarak *Euclidean* antara vektor P dan Q adalah sebagai berikut:

- 1) Pengurangan koordinat dengan cara kurangkan setiap koordinat vektor P dengan koordinat yang sesuai dari vektor Q :

$$(p_1 - q_1), (p_2 - q_2), \dots, (p_n - q_n) \dots \dots \dots (70)$$

- 2) Kuadrat dari selisih koordinat yaitu kuadratkan hasil pengurangan koordinat dari setiap dimensi:

$$(p_1 - q_1)^2, (p_2 - q_2)^2, \dots, (p_n - q_n)^2 \dots \dots \dots (71)$$

- 3) Penjumlahan kuadrat dengan cara jumlahkan semua hasil kuadrat dari langkah sebelumnya:

$$\sum_{i=1}^n (p_i - q_i)^2 \dots \dots \dots (72)$$

- 4) Pengakaran kuadrat dengan cara ambil akar kuadrat dari jumlah hasil kuadrat sebelumnya untuk mendapatkan jarak *Euclidean*:

Ini adalah cara umum untuk menghitung jarak *Euclidean* antara dua vektor dalam ruang n – dimensi. Langkah-langkah ini digunakan untuk mengukur jarak antara dua titik dalam konteks yang lebih tinggi, seperti dalam algoritma *k-Nearest Neighbors* (*k-NN*) atau banyak aplikasi ilmu data dan analisis statistik lainnya.

Misalkan:

$$\text{Vektor A: } (a_1, a_2, \dots, a_n)$$

Vektor B: $(b_1, b_2, \dots, b_n)$

Dimensi ruang: N. Langkah-langkah:

1. Pengurangan koordinat: hitung selisih koordinat di setiap dimensi:

$$d_1 = b_1 - a_1$$

$$d_2 = b_2 - a_2$$

...

$$d_n = b_n - a_n$$

2. Kuadrat selisih koordinat: kuadratkan hasil selisih koordinat di setiap dimensi:

$$d_{1_kuadrat} = d_1 \times d_1$$

$$d_{2_kuadrat} = d_2 \times d_2$$

...

$$d_{n_kuadrat} = d_n \times d_n$$

3. Penjumlahan kuadrat: jumlahkan semua hasil kuadrat dari langkah 2:

$$\text{Jarak_kuadrat} = d_{1_kuadrat} + d_{2_kuadrat} + \dots + d_{n_kuadrat}$$

4. Pengakaran kuadrat: hitung akar kuadrat dari jarak kuadrat untuk mendapatkan jarak *Euclidean*:

$$\text{Jarak_euclidean} = \sqrt{\text{jarak_kuadrat}}$$

2.5 Matrik Konfusi

Matriks konfusi adalah alat evaluasi yang berguna dalam memahami kinerja model klasifikasi. Ini mengcitrakan performa model dengan membandingkan prediksi model dengan nilai sebenarnya pada dataset uji. Matriks konfusi biasanya digunakan pada masalah klasifikasi yang memiliki kelas target yang sudah diketahui. Contoh matriks konfusi pada klasifikasi citra tekstur jenis daging, misalkan kita memiliki dua kelas Positif (P) dan Negatif (N). Dalam konteks matriks konfusi:

- 1) *True Positive* (TP): jumlah prediksi yang benar untuk kelas P.
- 2) *True Negative* (TN): jumlah prediksi yang benar untuk kelas N.
- 3) *False Positive* (FP): jumlah kesalahan prediksi untuk kelas P (model memprediksi P tetapi seharusnya N).

- 4) *False Negative* (FN): jumlah kesalahan prediksi untuk kelas N (model memprediksi N tetapi seharusnya P).

Tabel 2.1 Matriks Konfusi

	Prediksi Positif (P)	Prediksi Negatif (N)
Aktual Positif (P)	<i>True Positive</i> (TP)	<i>False Negative</i> (FN)
Aktual Negatif (N)	<i>False Positive</i> (FP)	<i>True Negative</i> (TN)

Penerapan matriks konfusi antara lain:

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN}$$

Akurasi mengukur sejauh mana model dapat mengklasifikasikan data dengan benar secara keseluruhan, yaitu dengan membandingkan jumlah prediksi yang benar (TP dan TN) dengan total jumlah prediksi (TP, TN, FP, dan FN)..

Perhitungan Matrik Konfusi

Perhitungan matriks konfusi melibatkan pengklasifikasian hasil prediksi dari model dengan nilai sebenarnya pada dataset uji. Misalkan kita memiliki model klasifikasi yang memprediksi apakah suatu email adalah *spam* (positif) atau bukan *spam* (negatif) berdasarkan ciri-ciri tertentu. Setelah model tersebut diterapkan pada dataset uji, kita mendapatkan hasil prediksi dan nilai sebenarnya dari dataset. Contoh tabel matriks monfusi, misalkan hasil prediksi model dan nilai sebenarnya untuk 100 contoh adalah seperti berikut:

Dari 100 contoh:

- a) Model memprediksi 70 contoh sebagai *spam* (positif).
- b) Model memprediksi 30 contoh bukan *spam* (negatif).

Dari hasil prediksi tersebut:

- a) 50 contoh memang benar-benar *spam* (*True Positive*, TP).
- b) 10 contoh diprediksi sebagai *spam* tetapi sebenarnya bukan *spam* (*False Positive*, FP).
- c) 5 contoh diprediksi sebagai bukan *spam* tetapi sebenarnya adalah *spam* (*False Negative*, FN).

d) 35 contoh memang benar-benar bukan *spam* (*True Negative*, TN).

Dengan menggunakan nilai-nilai ini, kita dapat menghitung metrik evaluasi seperti akurasi yang memberikan pemahaman tentang kinerja model dalam membedakan antara kelas spam dan bukan spam. Metrik-metrik ini kemudian dapat memberikan wawasan yang berguna tentang kekuatan dan kelemahan model klasifikasi citra tekstur jenis daging.

2.6 Teknik Kombinasi Fitur

Penjumlahan vektor fitur (*feature vector summation*), metode ini menggabungkan beberapa fitur dari dataset ke dalam satu vektor dengan melakukan operasi penjumlahan atau agregasi (Jain, A. K., dkk., (2000)). Jika sebuah dataset memiliki dua set fitur yang berbeda, misalnya *wavelet* (W) dan GLCM (G), maka kita dapat menggabungkannya dengan:

$$F_{\text{gabungan}} = W + G$$

atau secara komponen:

$$F_{\text{gabungan}} = (W_1, W_2, \dots, W_n, G_1, G_2, \dots, G_m)$$

di mana:

$W_1, W_2, \dots, W_n$ adalah fitur yang diperoleh dari transformasi wavelet.

$G_1, G_2, \dots, G_m$ adalah fitur dari GLCM.

2.7 Validasi *K-Fold*

Validasi silang *K-Fold* adalah teknik yang umum digunakan dalam evaluasi kinerja model untuk mengurangi bias dari pemisahan dataset menjadi data latih dan data uji. Ini membagi dataset menjadi K bagian (biasanya 5 atau 10), yang disebut lipatan (*folds*), dan melakukan pelatihan dan pengujian model sebanyak K kali.

Langkah-langkah untuk validasi silang *K-Fold*:

- 1) Pembagian dataset yaitu dataset dibagi menjadi K lipatan tanpa pengulangan data. Misalnya, jika $K = 5$, *dataset* dibagi menjadi 5 bagian dengan ukuran yang serupa.
- 2) Iterasi melalui lipatan yaitu salah satu lipatan digunakan sebagai data uji, sementara yang lain digunakan sebagai data latih. Ini dilakukan secara

berulang sebanyak K kali sehingga setiap lipatan digunakan sebagai data uji tepat satu kali.

- 3) Pengujian model yaitu model dilatih pada data latih dan dievaluasi pada data uji untuk setiap iterasi. Metrik evaluasi seperti akurasi dan lainnya dihitung untuk setiap iterasi.
- 4) Perhitungan rata-rata yaitu metrik evaluasi yang dihitung dari setiap iterasi digunakan untuk menghitung rata-rata dan variasi (bias dan *variance*) kinerja model.

Validasi silang *K-Fold* memberikan cara yang lebih dapat diandalkan untuk mengukur kinerja model dari pada metode pemisahan data latih dan uji tunggal, karena memanfaatkan *dataset* secara lebih efisien dan memberikan estimasi yang lebih konsisten terhadap kinerja model secara keseluruhan.

2.8 Kamera Ponsel Cerdas

Kamera Ponsel cerdas dapat digunakan secara independen untuk memberikan hasil pemosisian dalam ruangan yang akurat. Sistem pemosisian berbasis kamera ponsel cerdas bergantung pada peta lantai eksperimental dan pose kamera. Tantangan dalam pelokalan berbasis kamera ponsel cerdas adalah akurasi bergantung pada kecepatan perubahan arah pengguna (Poulose & Han, 2019). Berikut ini tabel spesifikasi kamera ponsel cerdas Samsung A33 yang digunakan dalam pengambilan sampel berbagai jenis daging untuk penelitian.

Tabel 2.2 Spesifikasi Kamera Ponsel Cerdas Samsung A33

<i>Layar Resolusi</i>	1080 x 2400 Piksel
<i>CPU</i>	Octa-core (2x2.4 GHz Cortex-A78 & 6x2.0 GHz Cortex-A55)
<i>RAM</i>	8 GB
<i>Memori Internal</i>	256 GB
<i>Kamera Utama</i> (Jumlah Kamera 4)	48 MP 8 MP 5 MP 2 MP
<i>Kamera Depan</i>	13 MP

2.9 Standar Nasional Indonesia (SNI 3932:2008)

Penelitian yang dilakukan di sini mencoba mengidentifikasi tekstur daging sapi menggunakan teknik pengolahan citra mengacu pada Standar Nasional Indonesia (SNI 3932:2008) tentang karkas dan kualitas daging sapi. Skor warna didasarkan pada skor warna standar yang paling dekat dengan warna yang diamati. Warna standar daging sapi terdiri dari sembilan skor warna, berkisar dari merah muda ke merah tua, disajikan dalam Gambar 2.5 di bawah ini:



Sumber: Tesis “Identifikasi Citra Untuk Mengidentifikasi Jenis Daging Sapi Dengan Menggunakan Transformasi Wavelet Haar” (Kiswanto, 2012)

Gambar 2.5 Warna Standar Daging Sapi

