IMPLEMENTASI METODE SVM DAN GABOR FILTER PADA APLIKASI PENGENALAN CITRA KAYU JATI UNTUK PEMBUATAN MEUBEL BERBASIS MOBILE (Studi Kasus PD Meubel Amoy Jaya)

Fachmi Miftahillah[1], Irfan Maliki[2]

[1,2] Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatiukur 112 – 114 Bandung
E-mail : fachmimiftahillah11@gmail.com[1], irfanmaliki007@gmail.com[2]

ABSTRAK

Kayu sebagai salah satu bahan kontruksi banyak digunakan dalam berbagai keperluan, antara lain sebagai bahan untuk pengolahan meubel. PD Meubel Amoy Jaya merupakan industri yang bergerak dibidang pengolahan kayu sebagai bahan kontruksi meubel. Kayu yang digunakan diantaranya adalah kayu jati. Kayu jati banyak digunakan karena memiliki karakteristik yang berbeda. Kurangnya pengetahuan tentang kayu jati dari konsumen juga semakin menyulitkan dalam memilih bahan dasar untuk pembuatan meubel secara tepat guna.

Gabor filter merupakan metode pengenalan objek untuk memunculkan ciri-ciri tekstur khusus dari citra yang telah dikonvolusi terhadap kernel. *Support Vector Machines* (SVM) adalah metode klasifikasi supervised karena ketika proses pelatihan, diperlukan target pembelajaran tertentu. Dalam pengolahan citra, tekstur dapat digunakan untuk klasifikasi terhadap citra yaitu untuk mengidentifikasi dan mengenali suatu citra dengan tekstur tertentu berdasarkan satu set kelas atau klasifikasi tekstur yang diberikan.

Berdasarkan hasil pengujian aplikasi pengenalan kayu jati berbasis mobile menggunakan metode *gabor filter* dan *support vector machines*, didapat hasil akurasi pengenalan sebesar 92%. Dan hasil survei kuesioner didapatkan hasil persentase sebesar 90%. Sehingga aplikasi tersebut dapat membantu konsumen PD Meubel Amoy Jaya dalam memilih kayu jati dan usianya secara tepat untuk pembuatan meubel.

Kata kunci: Kayu, Gabor Filter, Support Vector Machines, Klasifikasi, Mobile

1. PENDAHULUAN

Kayu sebagai salah satu bahan kontruksi banyak digunakan dalam berbagai keperluan, antara lain sebagai bahan bangunan gedung, jembatan, barang kerajinan dan termasuk juga sebagai bahan untuk pengolahan meubel. PD Meubel Amoy Jaya merupakan industri yang bergerak dibidang pengolahan kayu sebagai bahan kontruksi meubel. Jenis kayu yang dijadikan sebagai bahan dasar pembuatan meubel di perusahaan tersebut salah satunya adalah kayu jati. Dengan tingkat pengaruh usia dari suatu kayu, akan menghasilkan kualitas meubel yang berbeda-beda. Banyak konsumen dari perusahaan tersebut yang membuat meubel menggunakan kayu jati karena memiliki karakteristik yang berbeda yaitu : tingkat kekuatan kayu, tingkat keawetan kayu, tingkat keindahan kayu, serta tingkat kemudahan proses pengerjaannya [24]. Berdasarkan hasil wawancara kepada bapak Hasanudin Jafar selaku pemilik perusahaan meubel tersebut, kurangnya pengetahuan tentang kayu jati dari konsumen semakin menyulitkan dalam memilih kayu jati sebagai bahan dasar untuk membuat meubel secara tepat guna sesuai usia kayu tersebut. Oleh karena itu, aplikasi pengenalan kayu jati sangat dibutuhkan bagi konsumen di PD Meubel Amoy Jaya sehingga dapat memilih kayu jati secara tepat guna untuk pembuatan meubel.

Citra adalah gambar pada bidang dua dimensi, maka sebuah citra merupakan dimensi special atau bidang yang berisi informasi warna yang tidak bergantung waktu [8]. Tekstur adalah salah satu elemen dasar citra. Elemen dasar ini berupa ciri-ciri atau sifat yang terdapat dalam suatu citra yang membentuk suatu pola dengan interval jarak dan arah tertentu secara berulang-ulang yang memenuhi sebagian besar atau seluruh bidang citra. Dalam pengolahan citra, tekstur dapat digunakan untuk klasifikasi terhadap citra yaitu untuk mengidentifikasi dan mengenali suatu citra dengan tekstur tertentu berdasarkan satu set kelas atau klasifikasi tekstur yang diberikan. Setiap daerah pada citra tekstur tersebut memiliki karakteristik yang unik. Untuk mendapat karakteristik tersebut dibutuhkan ekstraksi ciri yang berguna untuk mengambil ciri-ciri yang berbeda dari tiap daerah pada citra tekstur tersebut.

Dengan menggunakan teknologi mobile sebagai basis untuk pengenalan citra, maka proses pengenalan akan lebih mudah dan efisien [17], sehingga dapat mempercepat dalam menghasilkan informasi dari citra yang akan dikenali. Terlebih dengan zaman yang semakin maju, memungkinkan teknologi ini dapat dimiliki oleh semua orang, khususnya konsumen dan pemilik dari perusahaan PD Amoy Jaya sehingga dapat membantu dalam memberikan informasi kepada konsumen dalam

memilih kayu jati untuk pembuatan meubel sesuai usianya.

Berdasarkan penelitian [18] dengan menggunakan gabor filter menghasilkan keakuratan sebesar 79,31%. Dan hasil dari penelitian [19] dalam pengklasifikasian menggunakan metode SVM menghasilkan keakuratan sebesar 90,67%. Dari hasil penelitian dan latar belakang masalah terebut, maka penulis dalam penelitian ini akan mengimplementasikan metode Gabor Filter dan Support Vector Machines pada aplikasi pengenalan kayu jati berbasis mobile.

1.1 Maksud dan Tujuan

Maksud dari penelitian ini adalah membuat aplikasi berbasis mobile dengan mengimplementasikan metode *Gabor Filter* dan *Support Vector Machine* (SVM) dalam pemilihan kayu jati dan usianya.

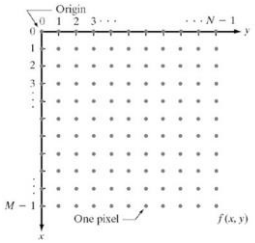
Adapun tujuan dalam penelitian ini adalah membantu konsumen PD Meubel Amoy Jaya dalam memilih kayu jati dan usianya untuk pembuatan meubel.

2. ISI PENELITIAN

2.1 Citra

Citra adalah gambar pada bidang dua dimensi, maka sebuah citra merupakan dimensi special atau bidang yang berisi informasi warna yang tidak bergantung waktu [8]. Citra digital merupakan suatu gambar yang tersusun dari piksel, dimana tiap piksel merepresentasikan warna (tingkat keabuan untuk gambar hitam putih) pada suatu titik di gambar [2]. Secara umum, pengolahan citra digital menunjukkan pemroresan gambar 2 dimensi menggunakan computer. Citra digital merupakan sebuah larik (array) yang berisi nilai – nilai real maupun komplek yang direpresentasikan dengan deretan bit tertentu.

Sebuah citra dapat didefinisikan sebagai fungsi f(x,y) berukuran M baris dan N kolom, dengan x dan y adalah koordinat spasial, dan amplitudo f di titik koordinat (x,y) dinamakan intensitas atau tingkat keabuan dari citra pada titik tersebut. Apabila nilai x,y dan nilai amplitudo f secara keseluruhan berhingga (finite) dan bernilai diskrit maka dapat dikatakan bahwa citra tersebut adalah citra digital.



Gambar 1 Koordinat Citra Digital

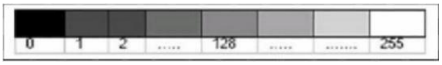
2.2 Resolusi Citra

Resolusi citra merupakan tingkat detail suatu citra. Semakin tinggi resolusi citra maka akan semakin tinggi pula tingkat detail dari citra tersebut. Satuan dalam pengukuran resolusi citra dapat berupa ukuran fisik (jumlah garis per tinggi citra). Resolusi sebuah citra dapat diukur dengan berbagai cara :

1. Resolusi Piksel
   Resolusi piksel merupakan perhitungan jumlah piksel dalam sebuah citra digital. Sebuah citra dengan tinggi N dan lebar M piksel berarti memiliki resolusi sebesar M x N. Resolusi piksel akan memberikan dua buah angka integer yang secara berurutan akan mewakili jumlah piksel lebar dan jumlah piksel tinggi dari citra tersebut.
2. Resolusi Spasial
   Resolusi spasial menunjukkan seberapa dekat jarak setiap garis pada citra. Jarak tersbut tergantung dari sistem yang menciptakan citra tersebut. Resolusi spasial menghasilkan jumlah piksel per satuan panjang.
3. Resolusi Spektral
   Sebuah citra digital membedakan intensitas ke dalam beberapa spectrum. Citra multi spectrum akan memberikan spectrum atau panjang gelombang yang lebih baik yang akan digunakan untuk menampilkan warna.
4. Resolusi Temporal
   Resolusi temporal berkaitan dengan video. Suatu video merupakan kumpulan frame statis yang berupa citra yang berurutan dan ditampilkan secara cepat. Resolusi temporal memberikan jumlah frame yang dapat ditampilkan setiap satuan frame per second (fps)..

2.3 Citra *Grayscale*

Grayscale adalah warna-warna piksel yang berada dalam rentang gradasi warna hitam dan putih. Intensitas grayscale biasanya disimpan sebagai data citra 8 bit atau 256 intensitas warna gray dari nilai 0 (hitam) dan 255 (putih). Citra grayscale merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pikselnya, dengan kata lain nilai bagian RED = GREEN = BLUE. Nilai tersebut digunakan untuk menunjukan tingkat intensitas. Warna yang dimiliki adalah warna dari hitam, keabuan, dan putih. Tingkatan keabuan disini merupakan warna abu dengan berbagai tingkatan dari hitam hingga mendekati putih [13]. Model warna ini menyimpan informasi warna sebesar 8 bit. Jadi warna yang bisa ditampilkan pada mode warna greyscale berjumlah sampai 256. Skala perhitungan numeriknya bisa dengan pendekatan nilai brightness yang berskala 0 sampai 255 seperti pada Gambar 3 berikut:



Gambar 2 Komposisi Warna Grayscale

Untuk mengkonversikan citra yang memiliki warna RGB ke derajat keabuan bisa menggunakan rumus 2.1 :

$$G = 0.3.R + 0.59.G + 0.11.B \quad (1)$$

Dimana :

G = Citra Greyscale
R = Red
G = Green
B = Blue
R = nilai red * 15 / 255
G = nilai green * 15 / 255
B = nilai blue * 15 / 255
Contoh dari Hasil dari proses grayscaling ditunjukkan pada Gambar 3 dibawah ini :



Gambar 3 Contoh Citra Grayscale

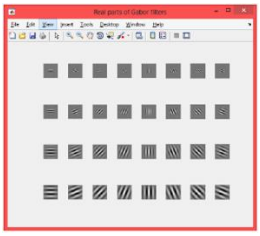
2.4 Metode Ekstraksi Gabor Filter

Metode *gabor filter* merupakan metode pengenalan objek yang bertujuan untuk memunculkan ciri-ciri tekstur khusus dari citra yang telah dikonvolusi terhadap kernel. Sebagai filter digunakan *gabor filter kernel* 2D yang diperoleh dengan memodulasi gelombang *sinus* 2D pada frekuensi dan orientasi tertentu dengan *Gaussian envelope*. Persamaan dasar fungis *gabor filter* 2D ditunjukkan pada persamaan (2.1), dimana σ adalah standar deviasi dari *Gaussian envelope* pada dimensi *x* dan *y*. Lamda (λ) dan tetha (θ) adalah panjang dan orientasi dari gelombang sinus 2D. Penyebaran dari *Gaussian envelope* didefinisikan dalam bentuk dari gelombang *sinus* λ. Rotasi dari $x - y$ sebesar sudut θ menghasilkan *gabor filter* pada orientasi θ [20].

$$g(x, y; \lambda, \theta, \varphi, \sigma, \gamma) = \exp\left(-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2}\right) \cos\left(2\pi \frac{x'}{\lambda} + \psi\right) \quad (2)$$

Dimana :
$x' = x \cos\theta + y \sin\theta$
$y' = -x \sin\theta + y \cos\theta$
σ = standar deviasi *Gaussian envelope*
λ = panjang gelombang *sinus* pada 2D dimensi
θ = derajat orientasi gelombang *sinus* pada 2D dimensi
γ = aspek ratio spasial
ψ = *phase offset* (sebagai faktor kosinus pada fungsi gabor)

Psi (ψ) memiliki nilai derajat antara -180, -90, 0, 90, 180. Nilai gamma (γ) yaitu 0,5. Sigma (σ) tidak memiliki nilai yang dapat ditetapkan, tetapi dapat diganti dengan nilai bandwidth (b) yaitu 1. Panjang gelombang (λ) dan derajat orientasi (θ) berbeda diterapkan pada satu titik tertentu (x,y), maka didapatkan banyak respon *filter* untuk titik tersebut, missal : digunakan empat panjang gelombang dan delapan orientasi, maka akan dihasilkan 32 respon *filter* untuk tiap titik citra yang dikomvolusikan dengan *filter* tersebut.

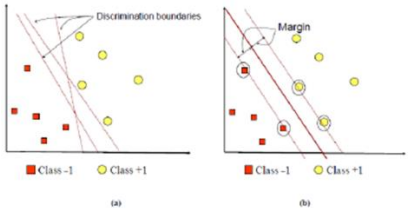


Gambar 4 Contoh Real Part Gabor Filter

2.5 Metode *Support Vector Machines* (SVM)

Support Vector Machines (SVM) merupakan metode klasifikasi *supervised* karena ketika proses pelatihan, diperlukan target pembelajaran tertentu. SVM juga merupakan algoritma yang bekerja menggunakan pemetaan nonlinear untuk mengubah data pelatihan asli ke dimensi yang lebih tinggi. Dalam hal ini dimensi baru, akan mencari *hyperplane* untuk memisahkan secara linear dan dengan pemetaan nonlinear yang tepat ke dimensi yang cukup tinggi, data dari dua kelas selalu dapat dipisahkan dengan *hyperplane* tersebut. SVM menemukan *hyperplane* ini menggunakan *support vector* dan *margin* [21].

SVM muncul pertama kali pada tahun 1992 oleh Vladimir Vapnik bersama rekannya Berhand Boser dan Isabelle Guyon. Dasar untuk SVM sudah sejak tahun 1960-an (termasuk karya awal oleh Vapnik dan Alexei Chervonenkis pada teori belajar statistik).



Gambar 5 Proses menemukan hyperplane oleh metode SVM dengan memisahkan kedua class -1 dan +1

Hyperplane pemisah terbaik antara kedua class dapat ditemukan dengan mengukur *margin hyperplane* tersebut dan mencari titik maksimalnya. *Margin* adalah jarak antara *hyperplane* tersebut dengan *pattern* terdekat dari masing-masing *class*. *Pattern* yang paling dekat ini disebut sebagai *support vector*. Garis *solid* pada gambar 5-b menunjukkan *hyperplane* yang terbaik, yaitu yang terletak tepat pada tengah-tengah kedua *class*, sedangkan titik merah dan kuning yang berada dalam lingkaran hitam adalah hitam adalah *support vector*. Usaha untuk mencari lokasi *hyperplane* ini merupakan inti dari proses pembelajaran pada SVM [22].

Konsep SVM dapat dijelaskan secara sederhana sebagai usaha mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua buah *class* pada input *space*. Gambar 5-a memperlihatkan beberapa *pattern* yang merupakan anggota dari dua buah *class* +1 dan -1. *Pattern* yang tergabung pada *class* -1

disimbolkan dengan warna merah (kotak), sedangkan *pattern* +1 disimbolkan dengan warna kuning (lingkaran). Problem klasifikasi dapat diterjemahkan dengan usaha menemukan garis (*hyperplane*) yang memisahkan antara kedua kelompok tersebut. Dalam linear Support Vector Machine permisah merupakan fungsi linear. Data latih dinyatakan oleh (x_i, y_i) dan x_i = { x_1, x_2, ... x_{iq}} merupakan atribut (fitur) set untuk data latih kelas ke-i. Untuk yi € {-1,1} menyatakan label kelas. Pendefinisian persamaan suatu hyperplane pemisah yang dituliskan dengan:

$$w * x_i + b = 0 \quad (3)$$

Data xi yang terbagi ke dalam dua kelas, yang termasuk kelas -1 (sampel negatif) didefinisikan sebagai vektor yang memenuhi pertidaksamaan (4) berikut.

$$w * x_i + b < 0 \text{ untuk } y_i = -1 \quad (4)$$

Sedangkan yang termasuk kelas +1 (sampel positif) memenuhi pertidaksamaan (5) berikut.

$$w * x_i + b >= 0 \text{ untuk } y_i = +1 \quad (5)$$

Dimana:
xi = data input
yi = label yang diberikan
w = nilai dari bidang normal
b = posisi bidang relatif terhadap pusat koordinat

Parameter w dan b adalah parameter yang akan dicari nilainya. Bila label data yi = -1, maka pembatas menjadi persamaan (6) berikut:

$$w * x_i + b \leq -1 \quad (6)$$

Bila label data yi = +1, maka pembatas menjadi persamaan (7) berikut:

$$w * x_i + b \geq +1 \quad (7)$$

Margin terbesar dapat dicari dengan cara memaksimalkan jarak antar bidang pembatas kedua kelas dan titik terdekatnya, yaitu 2/|w|. Hal ini dirumuskan sebagai permasalahan *quadratic programming* (QP) *problem* yaitu mencari titik minimal persamaan (8) dengan memperhatikan persamaan (9) berikut:

$$min\ \tau(w) = 1/2||w||^2 \quad (8)$$

$$yi(w * xi + b) - 1 \geq 0, (i=1,..,n) \quad (9)$$

Permasalahan ini ini dapat dipecahkan dengan berbagi teknik komputasi. Lebih mudah diselesaikan dengan mengubah persamaan (2.8) ke dalam fungsi *Lagrangian* pada persamaan (10), dan menyederhanakannya menjadi persamaan (11) berikut:

$$L(w,b,a) = 1/2||w||^2 - \sum_{i=1}^{n} a_i(y_i((w^T x_i + b) - 1)) \quad (10)$$

$$L(w,b,a) = 1/2||w||^2 - \sum_{i=1}^{n} a_i y_i((w^T x_i + b) - 1)) + \sum_{i=1}^{n} a_i \quad (11)$$

Dimana a_i adalah *lagrange multiplier* yang bernilai nol atau positif ($a_i \geq 0$). Nilai optimal dari persamaan (11) dapat dihitung dengan meminimalkan L terhadap *w*, *b* dan *a*. Dapat dilihat pada persamaan (12) sampai (14) berikut:

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{n} a_i y_i x_i = 0 \quad (12)$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{n} a_i y_i = 0 \quad (13)$$

$$\frac{\partial L}{\partial a} = \sum_{i=1}^{n} a_i y_i (w^T x_i + b) - \sum_{i=1}^{n} a_i = 0 \quad (14)$$

Maka masalah Lagrange untuk klasifikasi dapat dinyatakan pada persamaan (15) berikut:

$$Min\ L(w,b,a) = 1/2||w||^2 - \sum_{i=1}^{n} a_i y_i (w^T x_i + b) - \sum_{i=1}^{n} a_i \quad (15)$$

Dengan memperhatikan persamaan (16) dan (17) berikut :

$$w - \sum_{i=1}^{n} a_i y_i x_i = 0 \quad (16)$$

$$\sum_{i=1}^{n} a_i y_i = 0 \quad (17)$$

Model persamaan (15) diatas merupakan model primal *Lagrange*. Sedangkan dengan memaksimalkan L terhadap a_i, persamannya menjadi persamaan (18) berikut:

$$Max \sum_{i=1}^{n} a_i a_j y_i y_j^T x_i x_j^T \quad (18)$$

Dengan memperhatikan persamaan (19) berikut:

$$\sum_{i=1}^{n} a_i y_i = 0, a_i \geq 0\ (i,j = 1, \dots, n) \quad (19)$$

Untuk mencari nilai x_i dan y_i dapat dilakukan ketika sudah didapatkan nilai ekstraksi ciri dari pembobotan dan inisialisasi kelas . Hasil dari pembobotan ekstraksi diubah ke dalam bentuk format data svm, sedangkan data kelas menjadi label data svm.

Untuk mendapatkan nilai a_i, langkah pertama adalah mengubah setiap abstrak menjadi nilai vektor (*support vector*) = $\binom{x}{y}$. Kemudian nilai vektor dari setiap bobot ekstraksi dimasukkan ke persamaan (20) kernel *trick phi* berikut.

$$\phi\begin{bmatrix} x \\ y \end{bmatrix} = \begin{cases} \sqrt{x_n^2 + y_n^2} > 2\ maka \begin{bmatrix} \sqrt{x_n^2 + y_n^2} - x + |x - y| \\ \sqrt{x_n^2 + y_n^2} - y + |x - y| \end{bmatrix} \\ \sqrt{x_n^2 + y_n^2} \leq 2\ maka \begin{bmatrix} x \\ y \end{bmatrix} \end{cases} \quad (20)$$

Nilai *x* didapatkan dari persamaan (21) kernel linear untuk *x* berikut :

$$\sum_{i=1,j=1}^{n} x_i x_j^T, (i,j = 1, \dots, n) \quad (21)$$

Nilai *y* didapatkan dari persamaan (2.22) kernel linear untuk *y* berikut :

$$\sum_{i=1,j=1}^{1} y_i y_j^T, (i,j = 1, \dots, n) \quad (22)$$

Untuk mendapatkan jarak tegak lurus yang optimal dengan mempertimbangkan vektor positif, maka hasil perhitungan dari substitusi nilai *x* dan nilai *y* ke persamaan (20) diberi nilai bias = 1 . Kemudian cari parameter a_i, dengan terlebih dahulu mencari nilai fungsi setiap abstrak menggunakan persamaan (23), lalu mencari nilai a_i pada persamaan linear menggunakan persamaan (24) dengan memperhatikan *i,j* = 1,..,*n* berikut:

$$\sum_{i=1,j=1}^{n} a_i S_i^T S_j \quad (23)$$

$$\sum_{i=1,j=1}^{n} a_i S_i^T S_j = y_i \quad (24)$$

Setelah parameter a_i didapatakan, kemudian masukkan ke persamaan (25) berikut:

$$W = \sum_{i=1,j=1}^{n} a_i S_i \quad (25)$$

Hasil yang didapatkan menggunakan persamaan (25), selanjutnya digunakan persamaan (26) untuk mendapatkan nilai *w* dan *b*.

$$y = wx + b \quad (26)$$

Sedemikian sehingga didapatkanlah nilai *w* dan *b* atau nilai *hyperplane* untuk mengklasifikasikan kedua kelas.

2.6 Mobile

2.6.1 Mobile Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi (Nazruddin, 2012:1) dan menurut Firdan (2011:1) android adalah sistem operasi untuk telepon seluler yang berbasiskan linux. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri sehingga dapat digunakan oleh bermacam peranti penggerak.

Berdasarkan pemaparan pendapat oleh Nazruddin dan Firdan dapat diambil kesimpulan, android adalah suatu sistem operasi untuk *mobile* berbasis linux yang bersifat *open source* sehingga memberikan pengembang untuk melakukan pengembangan sesuai dengan yang diharapkannya sehingga sangat cepat perkembangan sistem operasi ini.

2.6.2 Kelebihan Android

Kelebihan yang dimiliki android adalah sebagai berikut :

1. User Friendly
   Kalimat ini sangat melekat pada sistem operasi Windows milikinya Microsoft, ibaratnya kita dengan sangat mudah mengoperasikan komputer hanya dengan belajar beberapa hari bahkan beberapa jam saja, dan ini juga melekat pada Android yang berjalan pada Smartphone.
2. Tampilan
   Dari segi tampilan, Android tidak kalah bagusnya dari IOS milik Apple, karena memang dari awal android hampir mengusung teknologi IOS, hanya saja ini versi murahnya.
3. Aplikasi
   Untuk aplikasi anda akan disajikan jutaan pilihan aplikasi yang menarik dari yang gratis hingga berbayar, dan bias mendownloadnya di Google Play.

2.6.3 Kekurangan Android

Kekurangan dari android sendiri adalah sebagai berikut :

1. Update System
   Untuk melakukan update system tidaklah mudah, harus menunggu dari masing-masing vendor untuk merilis update versi yang terbaru.
2. Batterai Cepat Habis
   Ini sering terjadi jika menyalakan paket data dan menggunakan widget serta aplikasi yang berjalan secara berlebihan, untuk mengatasinya pengguna harus mengurangi waktu aktifitas aplikasi pada smartphone.
3. Lag
   Kekurangan ini berkaitan dengan spesifikasi dari masing-masing perangkat, namun ada kalanya android ini tidak bersahabat dengan beberapa aplikasi dikarenakan RAM ataupun Processornya yang kurang memadai.

2.7 Analisis Pre-Processing Citra

2.7.1 Resize Image

Resize adalah tahapan dimana ukuran citra inputan diperkecil dari ukuran aslinya, tahap ini bertujuan untuk mendapatkan inputan yang sama dari semua citra inputan. Karena citra inputan yang akan diujikan memiliki ukuran yang besar yaitu 1200 x 1600 piksel, resize yang dilakukan yaitu dengan mengubah citra menjadi ukuran 200 x 200 piksel.

2.7.2 Proses *Grayscaling*

Tahap *grayscaling* citra adalah tahapan untuk menyederhanakan model citra. Tiga layer pada citra berwarna, R, G, dan B diubah menjadi satu *layer grayscale*. Untuk mengubah citra berwarna yang memiliki nilai matrik masing-masing R, G, dan B menjadi citra grayscale dengan membagi jumlah ketiga layer, dengan persamaan (1).

2.8 Analisis Metode Gabor Filter

Persamaan dasar fungsi *gabor filter* 2D ditunjukkan pada persamaan (2), dimana σ = 1 adalah standar deviasi dair *Gaussian envelope* pada dimensi x dan y. λ = 2 dan $\theta = 90^0$ adalah panjang dan orientasi dari gelombang sinus 2D, π = 3.14, $\gamma = 0{,}5$, dan $\psi = 0$ karena memiliki nilai derajat antara -180,-90,0,90,180.

Setelah melakukan proses perhitungan, maka diperoleh hasil nilai ekstraksi ciri pada citra kayu jati dengan 4 x 4 piksel seperti dibawah ini.

(x,y)	2	3	4	5
45	-0.77	-0.77	-0.48	-0.32
46	-0.41	-0.29	-0,2	-0.38
47	-0.41	-0.5	-0.52	-0.57
48	-0.32	-0.27	-0.48	-0.29

Setelah mendapatkan hasil dari nilai ekstraksi ciri citra, selanjutnya dilakukan pencarian terhadap nilai bobot dari ciri citra dengan mencari nilai mean dan nilai standar deviasi.

a. *Mean*

$$\mu = \frac{\sum x_i}{n} \quad (27)$$

Dimana :

μ = mean

n = banyaknya data

$\sum x_i$ = jumlah nilai data

= -0.77 + (-0.77) + (-0.48) + (-0.32) + (-0.41) + (-0.29) + (-0.2) + (-0.38) + (-0.41) + (-0.5) + (-0.52) + (-0.57) + (-0.32) + (-0.27) + (-0.48) + (-0.29) / 16 = -0,43625

b. Standar deviasi

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i-\mu)^2}{n-1}} \quad (28)$$

Dimana :

s = standar deviasi

x_i = nilai data

n = banyaknya data

Diketahui :
μ = -0,43625
Ditanyakan *s ?*
Perhitungan :

Tabel 1 Perhitungan Standar Deviasi

x_i	$x_i - \mu$	$(x_i - \mu)^2$
-0.77	-0.33375	0.111389063
-0.41	0.02625	0.000689063
-0.41	0.02625	0.000689063
-0.32	0.11625	0.013514063
-0.77	-0.33375	0.111389063
-0.29	0.14625	0.021389063
-0.5	-0.06375	0.004064063
-0.27	0.16625	0.027639063
-0.48	-0.04375	0.001914063
-0.2	0.23625	0.055814063
-0.52	-0.08375	0.007014063
-0.48	-0.04375	0.001914063
-0.32	0.11625	0.013514063
-0.38	0.05625	0.003164063
-0.57	-0.13375	0.017889063
-0.29	0.14625	0.021389063
$\sum_{i=1}^{n}(x_i - \mu)^2$		0.413375

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i-\mu)^2}{n-1}}$$

$$s = \sqrt{\frac{0.413375}{16-1}}$$

$$s = \sqrt{\frac{0.413375}{15}} = 0{,}166007028$$

Jadi nilai bobot vektor ciri dari citra kayu jati 4 x 4 piksel adalah (-0.43625, 0.166007028).

2.9 Analisis Umur Kayu Jati

Pada penelitian ini umur kayu jati diamati secara manual dengan melihat hasil dari ekstraksi gabor filter yang dapat dibedakan berdasarkan hasil dari nilai bobot mean. Dimana nilai mean merupakan hasil dari penjumlahan nilai ekstraksi gabor filter, sehingga nilai tersebut dapat mewakili nilai ciri suatu citra. Penelitian ini, umur kayu jati yang digunakan yaitu berumur < 6 tahun dan berumur > 6 tahun. Umur kayu jati yang berumur < 6 tahun mempunyai nilai bobot mean < 0,4 sedangkan umur kayu jati > 6 tahun mempunyai nilai bobot mean > 0,4. Tabel berikut ini adalah contoh dari gambar kayu jati yang berumur < 6 tahun dan > 6 tahun dengan nilai bobot mean yang dihasilkannya.

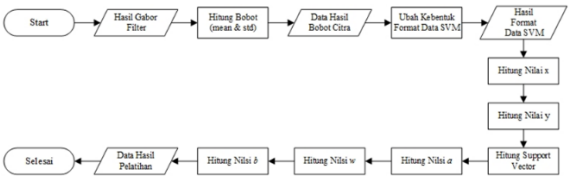
2.10 Tahap Pelatihan SVM

Pada tahap pelatihan ini, bertujuan untuk menentukan nilai dari vektor *a*, w dan nilai b untuk mendapatkan *hyperplane* terbaik. Setelah mendapatkan nilai bobot ciri dari seluruh piksel kayu jati, pada proses pelatihan ini data akan diubah kedalam bentuk data vektor svm. Pada penelitian ini untuk format representasi data input klasifikasi SVM menggunakan format [*kelas urutan_bobot$_1$:bobot$_1$ urutan_bobot$_n$:bobot$_n$*] [30]. Masukan yang pertama diberikan inisialisasi label +1 atau -1 yang menyatakan sebagai dua label awal yang diberikan. Tabel 2 berikut ini adalah contoh pengubahan nilai bobot ciri citra kedalam vektor.

Tabel 2 Contoh Pengubahan Nilai Bobot Menjadi Vektor

Nama Citra / Class	Train1.jpg / Kayu Jati
Nilai Bobot Citra	[0.4794115724185327, 0.31935630333301923]
Vektor	[1 1:0.4794115724185327, 2:0.31935630333301923]

Dari contoh tabel 2 dimana angka 1 pada karakter pertama menyatakan data tersebut masuk kedalam label positif. Kemudian angka 1 berikutnya merupakan indeks ke-n dalam vektor, selanjutnya angka 0.4794115724185327 adalah nilai bobot. Proses ini akan berulang sampai semua bobot pada citra training dirubah kedalam bentuk vektor. Adapun *flowchart* dari tahapan pelatihan pada SVM dapat dilihat pada Gambar 6 dibawah ini.



Gambar 6 Flowchart SVM Training

Pada penelitian ini menggunakan beberapa fungsi yang ada pada LibSVM. Proses pertama dalam LibSVM ini adalah memberikan mzodel pada data training yaitu berupa model label +1 dan -1.

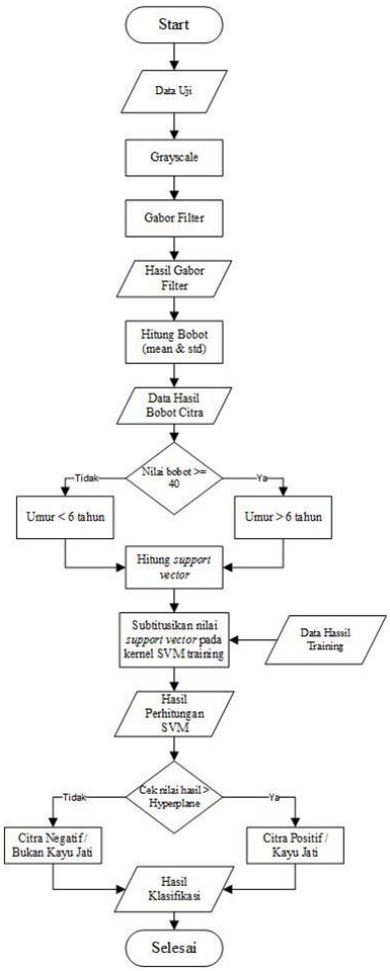
Akurasi model yang akan dihasilkan dari proses pelatihan dengan SVM sangat bergantung pada fungsi kernel serta parameter yang digunakan. Oleh karena itu performansinya dapat dioptimasi dengan mencari (mengestimasi) parameter terbaik.

Setelah data training dihitung dengan menggunakan parameter yang telah ditentukan, kemudian disimpan dalam fungsi class *svm_model* yang nantinya akan digunakan dalam klasifikasi yang menghasilkan prediksi.

2.11 Tahap Pengujian SVM

Setelah mendapatkan nilai w dan hyperplane pada data training, selanjutnya dapat ditentukan data yang aka diujikan termasuk kedalam kelas +1 atau -

1 dengan dilakukan perhitungan yang sama yaitu dari tahap pre-processing sampai mendapatkan nilai bobot dari citra kemudian mencari nilai vector pada citra yang akan diujikan. Gambar 7 berikut ini adalah *flowchart* pengujian pada svm untuk menghasilkan klasifikasi.

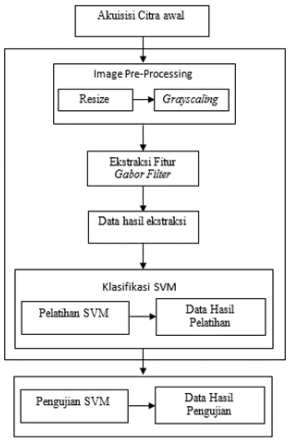


Gambar 7 Flowchart Klasifikasi SVM

Pada LibSVM, setelah menyimpan hasil nilai perhitungan parameter SVM pada *svm_model*, selanjutnya adalah menghitung prediksi dari method library svm yaitu *svm_predict_probability* dengan menggunakan parameter method *svm_model* yang telah diproses sebelumnya. Setelah proses perhitungan prediksi selesai, maka akan didapatkan citra hasil klasifikasi yang termasuk kedalam kelas +1 atau -1.

2.12 Deskripsi Alur Proses

Deskripsi secara umum dalam alur proses pengenalan citra ini adalah dimulai dari tahap akuisisi citra sampai pada tahapan akhir yaitu mengklasifikasikan citra. Pada gambar 8 berikut ini adalah tahapan proses yang akan dilakukan.



Gambar 8 Alur Proses Pengenalan Citra Kayu Jati

2.13 Implementasi

2.13.1 Implementasi Perangkat Lunak

Implementasi perangkat lunak yang digunakan dalam membangun aplikasi ini dapat dilihat pada tabel 3 berikut:

Tabel 3 Implementasi Perangkat Lunak

No	Perangkat Lunak	Keterangan
1	Android Studio versi 2.0	Perangkat lunak untuk membuat aplikasi android
2	Android SDK	Untuk pengembangan aplikasi android
3	Android NDK	Untuk mengeksekusi kode dalam bahasa C dan C++ kedalam aplikasi android
4	Java Development Kit v.8	Untuk kompilasi kode program
5	Java Runtime Edition	Sebagai platform untuk menjalankan sistem
6	LibSVM	Library untuk klasifikasi menggunakan metode Support Vector Machines (SVM)
7	Windows 10	Sebagai sistem operasi
8	Android v.6 (Marshmallow)	Sebagai sistem operasi pada perangkat lunak android

2.13.2 Implementasi Perangkat Keras

Implementasi perangkat keras yang digunakan dapat dilihat pada tabel 4 berikut :

Tabel 4 Implementasi Perangkat Keras

No	Perangkat Keras	Spesifikasi
1	Processor	Intel® Core™ i5-3210M 2,50 Ghz
2	VGA	2GB
3	RAM	8GB
4	Harddisk	500GB
5	Monitor	LCD 14”
6	Keyboard	Standard Port USB / PS2
7	Mouse	Standard Port USB / PS2

2.14 Pengujian

2.14.1 Pengujian pengenalan citra kayu jati

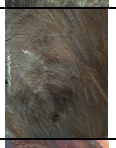
Untuk membuktikan keakuratan terhadap pengenalan kayu jati maka dilakukan pengujian terhadap gambar uji yang sesuai (kayu jati) dengan citra training dan terhadap gambar uji yang tidak sesuai dengan citra training (bukan kayu jati) dengan diinputkan melalui kamera ataupun gallery dengan memperhatikan tabel-tabel berikut :

Tabel 5 Hasil Pengujian Gambar Uji Citra Kayu Jati

No	Gambar Uji	Gambar Training	Deskripsi Citra	Hasil
1			Kayu Jati Umur > 6 Tahun	Berhasil
2			Kayu Jati Umur > 6 Tahun	Berhasil
3			Kayu Jati Umur < 6 Tahun	Berhasil
4			Kayu Jati Umur < 6 Tahun	Berhasil
5		Tidak Ada	Bukan Kayu Jati	Gagal
6		Tidak Ada	Bukan Kayu Jati	Gagal
7			Kayu Jati Umur < 6 Tahun	Berhasil
8			Kayu Jati Umur < 6 Tahun	Berhasil
9			Kayu Jati Umur > 6 Tahun	Berhasil
10			Kayu Jati Umur < 6 Tahun	Berhasil
11			Kayu Jati Umur < 6 Tahun	Berhasil
12			Kayu Jati Umur < 6 Tahun	Berhasil
13			Kayu Jati Umur < 6 Tahun	Berhasil
14			Kayu Jati Umur < 6 Tahun	Berhasil
15			Kayu Jati Umur < 6 Tahun	Berhasil

Berdasarkan pengujian dari contoh 15 gambar uji yang diinputkan, didapatkan presentasi untuk pengenalannya sebesar 86,66% dengan citra yang dikenali sebanyak 13 buah citra sebagai kayu jati.

Tabel 6 Hasil Pengujian Gambar Uji Citra Bukan Kayu Jati

No	Gambar Uji	Gambar Training	Deskripsi Citra	Hasil
1		Tidak Ada	Bukan Kayu Jati	Berhasil
2		Tidak Ada	Bukan Kayu Jati	Berhasil
3		Tidak Ada	Bukan Kayu Jati	Berhasil
4		Tidak Ada	Bukan Kayu Jati	Berhasil
5		Tidak Ada	Bukan Kayu Jati	Berhasil
6		Tidak Ada	Bukan Kayu Jati	Berhasil

No	Gambar Uji	Gambar Training	Deskripsi Citra	Hasil
7		Tidak Ada	Bukan Kayu Jati	Berhasil
8		Tidak Ada	Bukan Kayu Jati	Berhasil
9		Tidak Ada	Bukan Kayu Jati	Berhasil
10		Tidak Ada	Bukan Kayu Jati	Berhasil

Berdasarkan pengujian yang dilakukan pada gambar uji selain kayu jati, didapatkan hasil bahwa gambar uji tersebut bukan merupakan kayu jati yang artinya aplikasi dapat mengenali 100% bahwa gambar uji yang diinputkan tersebut tidak sesuai dengan citra training yang ada.

2.14.2 Kesimpulan Pengujian

Berdasarkan hasil pengujian yang telah dilakukan, untuk fungsionalitas dari aplikasi memberikan output yang diharapkan. Namun, untuk gambar uji yang terdapat backgroundnya harus terlebih dahulu dilakukan cropping dengan minimal 200 x 200 piksel sebelum diinputkan. Dari hasil penelitian, persentase keberhasilan dapar dihitung menggunakan rumus umum = (P/n)*100%, dimana P adalah jumlah gambar pada satu kategori citra kayu yang dimaksud, sedangkan n adalah jumlah total input gambar. Sehingga dapat diperoleh hasil keseluruhan dari pengujian pengenalan citra kayu jati sebesar : (23/25)*100% = 92%.

3. PENUTUP

3.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat ditarik kesimpulan untuk hasil implementasi metode *Support Vector Machines* (SVM) dan ekstraksi *Gabor Filter* untuk aplikasi pengenalan citra kayu jati didapatkan hasil akurasi yang baik yaitu sebesar 92%. Dengan tingkat akurasi baik dan aplikasi yang berbasis mobile, sehingga dapat membantu konsumen PD Meubel Amoy Jaya dalam memilih kayu jati dan usianya secara tepat yang akan digunakan untuk pembuatan meubel.

3.2 Saran

Berdasarkan hasil implementasi dan pengujian support vector machines dan metode ekstraksi gabor filter pada aplikasi pengenalan citra kayu jati untuk pembuatan meubel, terdapat beberapa usulan yang diharapkan dapat dikembangkan dalam penelitian berikutnya, diantaranya :

1. Untuk metode support vector machines, perbanyak data training karena metode ini merupakan metode klasifikasi yang ketika proses pengenalan diperlukan target pembelajaran. Sehingga semakin banyak data training yang diinputkan akan semakin meningkatkan tingkat akurasi.
2. Perlu dikembangkan kembali dalam proses pengolahan citra khususnya menggunakan mobile agar proses pengolahan data training dapat disimpan dalam database server atau media pemrosesan yang lebih cepat lainnya sehingga proses pengenalan akan lebih cepat dibandingkan dengan pengolahan data training pada penyimpanan mobile tersebut.
3. Mengembangkan metode ekstraksi ciri lainnya selain gabor filter, agar dapat mengenali citra selain dari teksturnya. Seperti dari sudut pengambilan gambar, warna, bentuk , dan lain sebagainya.
4. Diharapkan pada penelitian berikutnya metode SVM dan gabor filter dapat digunakan untuk mengenali selain citra kayu.

DAFTAR PUSTAKA

[1] Amit, Yali. (2002). 2D object recognition and detection: models, algorithms and networks. Cambridge: MIT Press.

[2] M. S. Nixon and A. S. Aguado, Feature Extraction and Image Processing, First ed. London: Newnes, 2002.

[3] F. Achmad, "Klasifikasi Kayu Dengan Menggunakan Local Binary Pattern dan Naive Bayes-Classifier".

[4] A. J. Arriawati, "Klasifikasi Citra Tekstur Menggunakan k-Nearest Neighbour Berdasarkan Ekstraksi Ciri Metode Matriks Kookurensi," Universitas Diponegoro, Semarang, 2011.

[5] D. Putra, Pengolahan Citra Digital, Yogyakarta: Andi, 2010.

[6] Jogiyanto. Pengenalan Komputer Andi Yogyakarta. 2006

[7] J. Han, M. Kamber, and J. Pei, Data Mining Concepts and Techniques, 3rd ed. Waltham, United States of America: Morgan Kaufmann, 2012.

[8] bayuaji.staff.gunadarma.ac.id/Downloads/files/32096/UML.pdf, akses tanggal 10 April 2016, 1.35 WIB

[9] Rizzky, Alfyan. "Perancangan Berorientasi Objek". Tersedia di : https://www.academia.edu/6475264/PERANCANGAN_BERORIENTASI_OBJEK. [Diakses tanggal 12 November 2016]

[10] K. Hamilton and R. Miles, "Learning UML 2.0", United Stated of America: O'reilly, 2006.
[11] D. Putra, "Pengolahan Citra," Andi, Yogyakarta, 2010.
[12] Soerianegara, I. dan RHMJ. Lemmens (eds.). 2002. Sumber Daya Nabati Asia Tenggara 5(1): Pohon penghasil kayu perdagangan yang utama. PROSEA – Balai Pustaka. Jakarta. ISBN 979-666-308-2. Hal. 7
[13] http://www.dephut.go.id/INFORMASI/PROPINSI/SUMSEL/Jenis_Kayu_Dagang.html. [Diakses tanggal 12 November 2016]
[14] Kusuma. H, Wirawan, Purwanto. D, "Single Scale Retinex Histogram Remapping untuk perbaikan recognition rate pada Eigenspace-based Face Recognition", Seminar on Intelligent Technology and Its applications, Oktober 2010.
[15] Yonarisa, Farida. (2012) *Mobile Apps.* [Online] Blog STMIK AMIKOM Yogyakarta. Tersedia di : http://blog.akakom.ac.id/faridayonarisa/2012/09/07/mobile-apps/. [Diakses tanggal 12 November 2016]
[16] D. E. Kurniawan, "Rancang Bangun Sistem Pengenalan Wajah Menggunakan Filter Gabor," Universitas Dipenogoro, Semarang, 2012
[17] S. N. Endah dan M. R. Widyanto, "Klasifikasi Pose Skeleton Pada Manusia dengan Support Vector Machine
[18] J. Domke, "Gabor Filter Visualization," University Of Maryland, Maryland, 2005.
[19] P. P. Widodo, R. T. Handayanto dan Herlawati, Penerapan Data Mining Dengan Matlab, Bandung: Rekayasa Sains, 2013.
[20] A. S. Nugroho, A. B. Witarto dan D. Handoko, "Support Vector Machines - Teori dan Aplikasinya dalam Bioinformatika," IlmuKomputer.Com, 2003. [Online]. Available: asnugroho.net/papers/ikcsvm.pdf. [Diakses 8 Februari 2015].
[21] Sugeng. (2016) Menghitung Varian dan Standar Deviasi. [Online] Rumus Statistik. Tersedia di : http://www.rumusstatistik.com/2013/07/varian-dan- standar-deviasi simpangan.html. [Diakses tanggal 12 Desember 2016]
[22] Mahfudz dkk., t.t., Sekilas Jati. Puslitbang Biotek dan Pemuliaan Tanaman Hutan. Jogyakarta.
[23] Chang, C.C and Lin, C.J.2013.*LIBSVM: A Library for Support Vector Machines*. Department of Computer Science National Taiwan University. Taiwan. [online]. Tersedia pada: http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf. [Diunduh pada 25 Februari 2016].
[24] Hsu, Chih-Wei et al. *A Practical Guide to Support Vector Classification*.
[25] Department of Computer Science and Information Engineering, National Taiwan University. 2004.
[26] Sembiring, K. (2007, September). Penerapan Teknik Support Vector Machine Untuk Pendeteksian Intrusi Pada Jaringan. *S1 Teknik Informatika, Sekolah Tinggi Elektro dan Informatika, ITB.*
[27] Amalia, A. N. (2016). *Implementasi Support Vector Machine (SVM) Pada Klasifikasi Laporan Skripsi (Studi Kasus Teknik Informatika).* Bandung: Perpustakaan UNIKOM.
[28] Putri. A. D. (2013). Klasifikasi Dokumen Teks Menggunakan Metode Support Vector Macchine Dengan Pemilihan Fitur Chi-Square.