



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

Adapun beberapa telaah literatur yang berhubungan dalam mengimplementasikan verifikasi tanda tangan, yaitu verifikasi tanda tangan, *preprocessing*, algoritma *Discrete Wavelet Transform*, algoritma *K-Nearest Neighbor*, dan evaluasi sistem.

2.1 Verifikasi Tanda Tangan

Verifikasi tanda tangan merupakan proses untuk memverifikasi seseorang dengan menggunakan tanda tangan. Kesulitan utama yang diamati dalam verifikasi tanda tangan adalah ketidakkonsistenan tanda tangan individu setiap kali melakukan tanda tangan. Terdapat beberapa faktor kunci yang mempengaruhi tanda tangan seseorang (Bača, dkk., 2011):

- Keadaan fisik dan psikologis orang tersebut - termasuk penyakit, cedera, ketakutan, detak jantung, usia seseorang, ketenangan, niat baik, dan lainnya.
- Posisi tubuh - tidak sama jika orang tersebut berdiri atau duduk saat menandatangani dokumen, kemana orang tersebut melihat pada saat itu, dan lainnya.
- Permukaan tulisan dan bahan tulisan (pena) - tanda tangan akan terlihat berbeda pada berbagai jenis kertas. Tanda tangan juga akan terlihat berbeda jika diambil dengan tablet digital atau pena khusus. Menulis dengan pena, pensil, *stylus* juga memengaruhi tanda tangan seseorang.

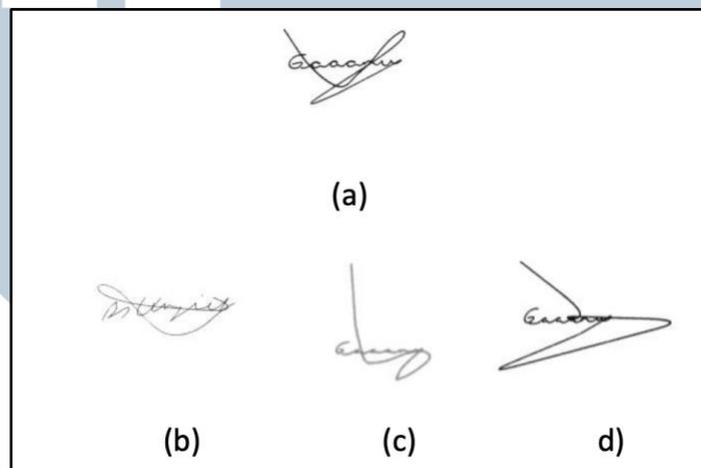
- Tujuan penandatanganan – hasil tanda tangan biasanya berbeda secara signifikan jika diambil di lingkungan formal dan informal.
- Faktor lingkungan-lingkungan dan orang-orang yang mengelilingi penanda tangan. Faktor ini termasuk kebisingan, pencahayaan, suhu, kelembaban, dan lainnya.

Metode verifikasi tanda tangan dapat diklasifikasikan berdasarkan informasi masukan tanda tangan menjadi dua kategori, *online* dan *offline*. Metode verifikasi tanda tangan *online* merujuk pada penggunaan fungsi-fungsi waktu dalam proses penandatanganan secara dinamis. Metode verifikasi tanda tangan *offline* merujuk pada penggunaan gambar statis dari tanda tangan (Jain, dkk., 2006). Pada metode verifikasi tanda tangan *online*, pengguna membutuhkan perangkat berupa pena khusus yang biasa disebut dengan *stylus* untuk membuat tanda tangannya. Hasil dari data masukan tanda tangan pengguna berupa informasi dinamis dimana pada informasi tersebut didapatkan data kecepatan, tekanan, jumlah goresan, dan lainnya. Sedangkan pada metode verifikasi tanda tangan *offline*, pengguna tidak memerlukan pena khusus untuk mendapatkan data masukan berupa gambar tanda tangan, melainkan data masukan didapat dari alat pemindai ataupun kamera. Hasil dari data masukan tanda tangan pengguna berupa informasi statis dimana pada informasi tersebut tidak terdapat data kecepatan, tekanan, jumlah goresan dan lainnya. Untuk itu dapat dikatakan metode verifikasi tanda tangan *offline* lebih sulit untuk diverifikasi dibandingkan pada metode *online*.

Secara umum, terdapat tiga jenis pemalsuan yang terjadi pada verifikasi tanda tangan *offline* (Ghandali dan Moghaddam, 2009):

- *Random forgery*: terjadi ketika penandatanganan hanya tahu nama orang yang tanda tangannya akan ditandatangani. Sampel acak diambil dari tanda tangan asli milik orang lain.
- *Simple forgery*: terjadi ketika si pemalsu tahu bentuk tanda tangan orang tersebut, tetapi belum banyak mempraktikkannya.
- *Skilled forgery*: tiruan yang masuk akal dari model tanda tangan asli.

Tiga jenis pemalsuan yang berbeda ini ditunjukkan pada Gambar 1.



Gambar 2.1 Jenis Pemalsuan (a) Tanda Tangan Asli (b) *Random Forgery* (c) *Simple Forgery* (d) *Skilled Forgery* (Al-maqaleh dan Musleh, 2016)

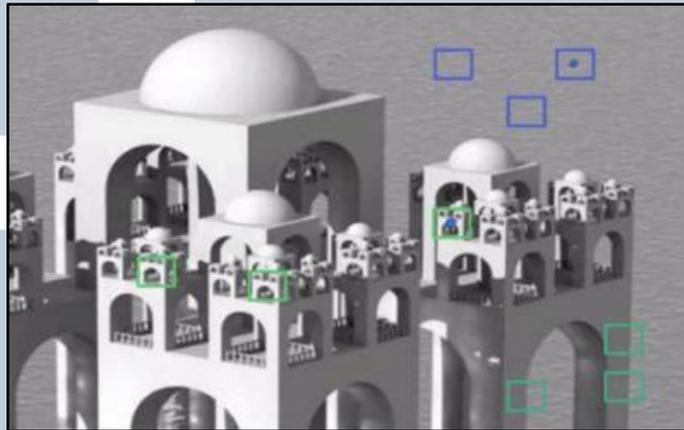
2.2 Preprocessing

Preprocessing merupakan sebuah proses pengolahan citra sebelum citra tersebut melalui proses *feature extraction*. Terdapat beberapa tahap *preprocessing* citra, antara lainnya adalah sebagai berikut.

2.2.1 Noise Filter

Ketika citra tanda tangan dipindai dengan menggunakan alat pemindai, maka *noise* citra juga ikut masuk melalui alat pemindai atau dari kertas yang menjadi *background* tanda tangan. Untuk itu, *noise* citra harus dihilangkan terlebih

dahulu sebelum citra tanda tangan melalui proses *feature extraction* (Gupta dan Hans, 2012). Salah satu metode yang dapat digunakan untuk menghilangkan *noise* citra adalah dengan menggunakan algoritma *Non-Local Means Denoising* dengan beberapa optimisasi komputasi.



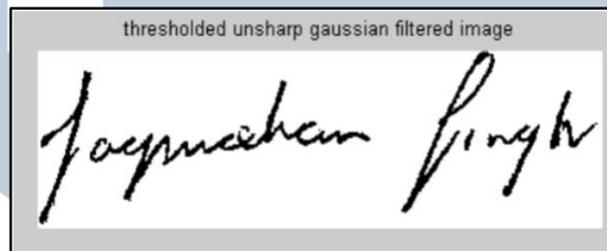
Gambar 2.2 Citra Algoritma *Non-Local Means Denoising* (Mordvintsev dan Rahman, 2013)

Secara sederhana, algoritma *Non-Local Means Denoising* bekerja dengan sebuah *window* kecil pada gambar yang mencari tambalan yang sama yang berada di tempat lain dalam gambar. Kemudian tambalan gambar-gambar yang sama ini dirata-ratakan dan ganti piksel tersebut dengan hasil nilai rata-rata. Pada Gambar 2.2 dapat dilihat bahwa area pada kotak warna biru tampak serupa dan area pada kotak warna hijau juga tampak serupa. Jadi pikselnya diambil, kemudian buat sebuah area kotak kecil di sekitarnya, cari area kotak yang serupa pada gambar, rata-rata semua area kotak, dan ganti piksel dengan hasil rata-rata yang didapat. Teknik ini membutuhkan lebih banyak waktu dibandingkan dengan teknik pengaburan lainnya, namun hasilnya sangat baik.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

2.2.2 Thresholding

Tujuan dari proses *thresholding* adalah untuk membuat citra tanda tangan keabuan menjadi citra tanda tangan biner dengan cara mengubah nilai piksel menjadi 1 apabila nilai piksel di atas *threshold* dan mengubah nilai piksel menjadi 0 apabila nilai piksel di bawah *threshold*. Ada beberapa metode berbeda untuk memilih *threshold*, pengguna dapat secara manual memilih nilai *threshold*, atau dengan menggunakan algoritma yang dapat menghitung nilai *threshold* secara otomatis (Gupta dan Hans, 2012). Gambar 2.3 merupakan hasil proses *thresholding* yang dilakukan oleh Gupta dan Hans (2012) dengan menggunakan *matlab*.



Gambar 2.3 Citra *Threshold* (Gupta dan Hans, 2012)

2.2.3 Thinning

Thinning merupakan sebuah proses untuk menipiskan garis pada tanda tangan secara morfologi pada citra biner. Hasil dari proses ini adalah sebuah kerangka citra tanda tangan (Brahmane, dkk., 2016). *Thinning* bertujuan untuk memperjelas suatu bentuk citra tanda tangan dengan menghilangkan bagian-bagian garis tanda tangan yang memiliki penebalan.

2.3 Discrete Wavelet Transform

Wavelet adalah bentuk gelombang dari durasi terbatas yang efektif yang memiliki nilai rata-rata nol (Chowdhury dan Khatun, 2012). Transformasi *wavelet*

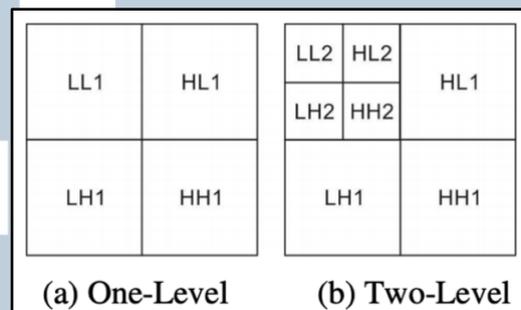
menguraikan sinyal menjadi seperangkat fungsi dasar. Fungsi-fungsi dasar ini disebut *wavelet*. *Wavelet* diperoleh dari prototipe *wavelet* tunggal yang disebut *mother wavelet* dengan pelebaran dan pergeseran (Blessie, ddk., 2011)

Wavelet melakukan dekomposisi sinyal menjadi *band-band* frekuensi kemudian analisis sinyal dilakukan pada *band* tersebut. DWT (*Discrete Wavelet Transform*) menggunakan hasil filter dari potongan frekuensi untuk menganalisis skala sinyal yang berbeda. Sinyal dilewatkan pada serangkaian *high pass filter* untuk menganalisis frekuensi tinggi dan serangkaian *low pass filter* untuk menganalisis frekuensi rendah. DWT menggunakan dua set fungsi, yaitu fungsi skala dan fungsi *wavelet*, dimana setiap fungsinya berhubungan dengan *low pass filter* dan *high pass filter* (Suprpto, dkk., 2014)

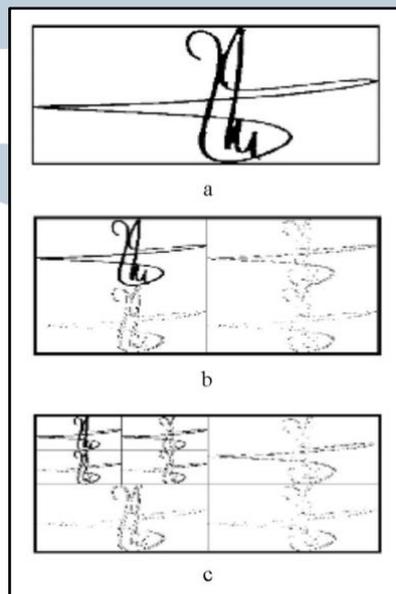
DWT menguraikan gambar menjadi empat sub gambar (LL, LH, HL, HH) ketika sebuah sampel didekomposisi. Salah satu dari sub gambar (LL) adalah versi yang dihaluskan dari gambar asli yang berhubungan dengan pita frekuensi rendah dan tiga sub gambar lainnya (LH, HL, HH) adalah detail yang berhubungan dengan pita frekuensi tinggi. Pita-pita ini mewakili tepi horisontal, vertikal dan diagonal gambar. DWT dengan tingkat dekomposisi N memiliki pita frekuensi $3N + 1$ dengan pita frekuensi tinggi $3N$ (Ghandali dan Moghaddam, 2009).

Pada Gambar 2.4(a). *subband* yang berlabel LH1, HL1 dan HH1 mewakili koefisien skala *wavelet* terbaik yaitu gambar detail. Sedangkan *subband* LL1 sesuai dengan koefisien *coarse level* yaitu, perkiraan gambar. Untuk mendapatkan *coarse level* berikutnya dari koefisien *wavelet*, *subband* LL1 sendiri selanjutnya didekomposisi dan diambil sampelnya. Hal ini menghasilkan dekomposisi *wavelet* level dua seperti yang ditunjukkan pada Gambar 2.4(b). Demikian pula, untuk

mendapatkan dekomposisi lebih lanjut, LL2 akan digunakan. Proses ini berlanjut hingga beberapa skala akhir tercapai (Arivazhagan, dkk., 2003). Contoh gambar tanda tangan yang telah didekomposisi menggunakan DWT terdapat pada Gambar 2.5.



Gambar 2.4 Dekomposisi DWT (a) Level Satu (b) Level Dua (Arivazhagan, dkk., 2003)



Gambar 2.5 (a) Tanda Tangan Asli (b) Dekomposisi Level Satu Menggunakan DWT (c) Dekomposisi Level Dua Menggunakan DWT (Ghandali dan Moghaddam, 2009)

Haar Discrete Wavelet Transform merupakan *wavelet* yang tertua dan sederhana. Dalam *Haar DWT* terdapat dua proses yaitu *forward* dan *inverse*. *Forward* berguna untuk memecah atau dekomposisi gambar, sedangkan *inverse* berguna untuk membentuk kembali pecahan-pecahan gambar dari proses *forward* menjadi gambar seperti semula (Hidayatno, dkk., 2012). Pada penelitian ini, proses

yang digunakan hanya *forward* karena *Haar* DWT digunakan untuk mendapatkan ekstraksi fitur dari gambar. Setiap langkah dalam proses *forward* transformasi *Haar* menghitung rata-rata dan diferensiasi. Fungsi penskalaan dan *wavelet* untuk *wavelet Haar* diberikan masing-masing dalam Rumus 2.1 dan Rumus 2.2. Dimana S_0 hingga S_{n-1} adalah data gambar, a adalah rata-rata, dan d adalah diferensiasi (Dhabe dan Sanjekar, 2010).

$$a = \frac{S_i + S_{i+1}}{2} \quad \dots(2.1)$$

$$d = \frac{S_i - S_{i+1}}{2} \quad \dots(2.2)$$

Berdasarkan penelitian yang dilakukan oleh Dhabe dan Sanjekar (2010), tahapan implementasi dari *Haar* DWT adalah sebagai berikut:

- a. Pertama, transformasi *wavelet* diterapkan pada setiap baris nilai piksel dengan menghitung fungsi rata-rata terlebih dahulu yang dilanjutkan dengan menghitung fungsi diferensiasi setelah fungsi rata-rata dihitung semua pada satu baris. Ulangi operasi ini sampai semua baris diproses.
- b. Kedua, transformasi *wavelet* diterapkan pada setiap kolom nilai piksel dengan menghitung fungsi rata-rata terlebih dahulu yang dilanjutkan dengan menghitung fungsi diferensiasi setelah fungsi rata-rata dihitung semua pada satu kolom. Ulangi operasi ini sampai semua kolom diproses.
- c. Untuk mendapatkan level dekomposisi berikutnya, kedua langkah diatas diulang hanya pada kuadran yang mengandung rata-rata atau *subband LL*.

2.4 K-Nearest Neighbor

Algoritma *K-Nearest Neighbor* adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling

dekat dengan objek tersebut. *K-Nearest Neighbor* (KNN) adalah suatu metode yang menggunakan algoritma *supervised* dimana hasil dari *query instance* yang baru diklasifikasikan berdasarkan mayoritas dari kategori pada KNN. Tujuan dari algoritma ini adalah mengklasifikasikan obyek baru berdasarkan atribut dan *training sample* (Yustanti, 2012).

KNN mengklasifikasikan suatu kasus dengan cara mengukur *distance* atau jarak dari data *training* yang sudah dikumpulkan dan mencari jarak data yang berdekatan. Nilai mayoritas dari data yang berdekatan tersebut untuk menjadi sebuah nilai prediksi untuk data baru selanjutnya (Khamar, 2013). *Pseudocode* dari algoritma KNN dapat dilihat pada Gambar 2.6.

```
for semua sampel yang belum diketahui unsample(i)
  for semua sampel yang diketahui sample(j)
    hitung jarak antara unsample(i) dan sample(j)
  end for
tentukan K
cari sampel yang sesuai sample(J1),...,sample(JK)
tentukan unsample(i) ke kelas yang muncul paling sering
end for
```

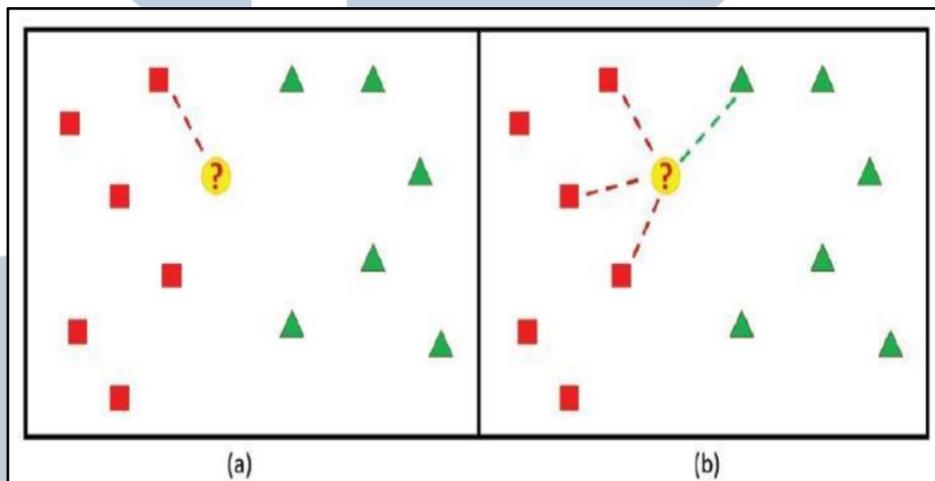
Gambar 2.6 Pseudocode Algoritma KNN

Algoritma *K-Nearest Neighbor* secara sederhana dapat diterapkan dengan langkah-langkah sebagai berikut (Yustanti, 2012):

- a. Tentukan parameter K (jumlah tetangga terdekat).
- b. Hitung jarak antara permintaan (data *testing*) dan semua data *training*.
- c. Urutkan jarak tersebut dan tentukan tetangga terdekat berdasarkan jarak terdekat ke-K.
- d. Tentukan kategori dari tetangga terdekat.

- e. Gunakan kategori mayoritas dari kategori tetangga terdekat sebagai nilai prediksi.

Pada Gambar 2.7 diperlihatkan KNN dengan aturan untuk $K=1$ dan $K=4$ untuk sebuah set sampel yang dibagi menjadi 2 kelas. Sampel yang kelasnya belum diketahui akan diklasifikasikan berdasarkan nilai mayoritas sampel data *training* sejumlah K sampel terdekat. Pada Gambar 2.7(a) digunakan hanya satu buah sampel, sedangkan pada Gambar 2.7(b) digunakan empat buah sampel. Pada Gambar 2.7(a), sampel dapat diklasifikasikan sebagai kelas berwarna merah karena satu sampel terdekat adalah sampel berwarna merah. Pada Gambar 2.7(b), sampel dapat diklasifikasikan sebagai kelas berwarna merah karena mayoritas dari empat sampel terdekat adalah berwarna merah.



Gambar 2.7 (a) KNN dengan $K=1$; (b) KNN dengan $K=4$ (Bolandraftar dan Imandoust, 2013)

Dekat atau jauhnya tetangga biasanya dihitung berdasarkan *Euclidean Distance* yang direpresentasikan pada Rumus 2.3 sebagai berikut:

UNIVERSITAS
MULTIMEDIA
NUSANTARA

$$D(a, b) = \sqrt{\sum_{k=1}^d (a_k - b_k)^2} \quad \dots(2.3)$$

dimana matriks $D(a,b)$ adalah jarak skalar dari kedua *vector* a dan b dari matriks dengan ukuran d dimensi (Kartika, dkk., 2017).

Menurut Yustanti (2012), algoritma KNN memiliki kelebihan yaitu ketangguhan terhadap *training data* yang memiliki banyak *noise* dan efektif apabila *training data*-nya besar. Sedangkan kelemahan dari algoritma KNN yaitu KNN perlu menentukan nilai dari parameter K (jumlah tetangga terdekat), *training* berdasarkan jarak tidak jelas mengenai jenis jarak apa yang harus digunakan dan atribut mana yang harus digunakan untuk mendapatkan hasil terbaik, dan biaya komputasi cukup tinggi karena diperlukan perhitungan jarak dari tiap *query instance* pada keseluruhan *training sample*.

2.5 Evaluasi Sistem

Analisis hasil dari verifikasi tanda tangan dilakukan berdasarkan berbagai parameter evaluasi dengan menghitung nilai FAR (*False Acceptance Rate*), FRR (*False Rejection Rate*), dan *Accuracy*. Semua metrik evaluasi ini dihitung berdasarkan *confusion matrix* di bawah ini (Davis dan Goadrich, 2006).

Tabel 2.1 *Confusion Matrix*

	Actual Positive	Actual Negative
Predicted Positive	TP	FP
Predicted Negative	FN	TN

Berdasarkan *confusion matrix* pada Tabel 2.1, *True Positive* (TP) merupakan kondisi ketika sistem mengenali citra tanda tangan asli sebagai tanda tangan yang asli. *False Positive* (FP) merupakan kondisi ketika sistem mengenali tanda tangan

palsu sebagai tanda tangan asli. *False Negative* (FN) merupakan kondisi ketika sistem mengenali tanda tangan asli sebagai tanda tangan palsu. *True Negative* (TN) merupakan kondisi ketika sistem mengenali tanda tangan palsu sebagai tanda tangan palsu.

a. FAR (False Acceptance Rate)

FAR merupakan perhitungan dalam keamanan biometrik untuk mengetahui tingkat kesalahan sistem dalam menerima tanda tangan yang palsu.

$$FAR = \frac{FP}{FP+TN} \quad \dots(2.4)$$

b. FRR (False Rejection Rate)

FRR merupakan perhitungan dalam keamanan biometrik untuk mengetahui tingkat kesalahan sistem dalam menolak tanda tangan yang asli.

$$FRR = \frac{FN}{FN+TP} \quad \dots(2.5)$$

c. Accuracy

Accuracy merupakan perhitungan untuk mengetahui tingkat performa dari sistem.

$$Accuracy = 100 - \frac{FAR+FRR}{2} \quad \dots(2.6)$$

UMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA