

# APLIKASI BIOMETRIKA PENGENALAN CITRA SIDIK JARI DENGAN METODE *MINUTIAE* DAN *ARTIFICIAL NEURAL NETWORK BACKPROPAGATION*

Sandri Titik Wulandari<sup>1</sup>, Ernawati<sup>2</sup>, Endina Putri Purwandari<sup>3</sup>

<sup>1,2,3</sup>Program Studi Teknik Informatika, Fakultas Teknik, Universitas Bengkulu.  
Jl. WR. Supratman Kandang Limun Bengkulu 38371A INDONESIA  
(telp: 0736-341022; fax: 0736-341022)

<sup>1</sup>sandrititikwulandari@gmail.com

<sup>2</sup>ernawati@unib.ac.id

<sup>3</sup>endinaputri@unib.ac.id

*Abstrak:* Penelitian biometrika sidik jari dilakukan karena setiap individu memiliki perbedaan sidik jari bahkan pada individu kembar identik. Penelitian ini membangun sebuah aplikasi biometrika pengenalan citra sidik jari dengan metode *Minutiae* dan *Artificial Neural Network Backpropagation* untuk pengenalan sidik jari. Citra yang digunakan sebagai objek penelitian adalah citra sidik jari telunjuk dan jari tengah dari kedua tangan dari *ATVS-FakeFingerprint Database*. Aplikasi ini dibangun dalam bahasa pemrograman Matlab dan dirancang dengan Diagram *Unified Modeling Language* (UML) dan pengembangan sistem yang digunakan adalah *prototyping*. Hasil yang diperoleh pada penelitian ini adalah nilai *Genuine Acceptance Rate* (GAR) sebesar 100% untuk citra latih dijadikan citra uji. Untuk citra uji diluar *database* adalah 67%. Citra uji dari sidik jari yang pengambilan menggunakan *flat capacitive sensor by Precise Biometrics model Precise 100* sebesar 55%. Pada pengujian citra uji yang terkena serangan yaitu rotasi 90° sebesar 40%, Citra uji serangan *Noise Gaussian* 10 % adalah 40%, *Blur Gaussian ratio* 2.0 sebesar 70%, dan citra uji yang terpotong sebesar 30%.

Kata kunci : Pengenalan Citra, Sidik Jari, *Minutiae*, *Artificial Neural Network*, *Backpropagation*

**Abstract :** Fingerprint biometrics research was carried out because each individual has difference fingerprints even identical twins. This research designed a biometrics application recognition fingerprint by using minutiae and Artificial Neural Network Backpropagation method for recognizing fingerprint. Image used as a research object were image index and middle fingerprints of both hands of ATVs-FakeFingerprint Database. This application was designed in Matlab programming language and with a diagram of the Unified Modeling Language (UML) and the development system used was prototyping. The result of the research was the Genuine Acceptance Rate (GAR) value of 100% for training was used as test image. For the test outside the database was 67%. Test image of fingerprint was captured using flat capacitive sensor by Precise Biometrics Precise 100 models was 55%. On testing the test rotation 90° attack was 40%, the test Gaussian

**Noise attack 10% was 40%, Blur Gaussian ratio 2.0 was 70%, and test cut was 30%.**

**Keywords: Recognition, Fingerprint, Minutiae, Artificial Neural Network, Backpropagation**

## I. PENDAHULUAN

Semakin berkembangnya teknologi komputer, salah satunya teknologi pengolahan citra digital yang memberikan kemudahan untuk memproses suatu citra agar dapat dimanfaatkan dalam berbagai bidang. Untuk saat ini waktu pengenalan ciri seseorang harus lebih cepat. Berbagai cara dapat dilakukan untuk membedakan identitas seseorang dengan orang lain dengan mengidentifikasi ciri seseorang.

Sistem identifikasi dengan memanfaatkan karakteristik biometrik saat ini mencapai perkembangan yang luar biasa dalam menggantikan sistem identifikasi secara konvensional. Sistem ini didasarkan pada karakteristik alami manusia, yaitu karakteristik fisiologis dan karakteristik perilaku seperti wajah, sidik jari, telapak tangan, iris, retina mata, DNA, dan tanda tangan

Sidik jari dapat digunakan sebagai sumber informasi untuk proses pengenalan identitas manusia. Teknik identifikasi dengan sidik jari sudah digunakan sejak akhir abad ke-19 dan telah diketahui bahwa setiap individu memiliki perbedaan bahkan pada individu kembar identik yang memiliki susunan DNA yang hampir sama

Menurut Jain (dalam Purwandari,2008) menyatakan bahwa sidik jari memiliki beberapa kelebihan dibandingkan sistem biometrik lainnya yaitu: 1) unik artinya setiap individu memiliki ciri khas sidik jari tersendiri; 2) permanen artinya sidik jari tidak cepat mengalami perubahan dalam periode waktu tertentu; 3) efektif artinya memiliki kemampuan kecepatan akurasi; 4) aman artinya

sidik jari tidak mungkin dapat dipinjamkan, dicuri dan dipalsukan. Sidik jari juga memiliki reliabilitas paling tinggi dan telah digunakan oleh ahli forensik dalam investigasi kriminal.

Pola sidik jari normal terdiri dari garis-garis dan spasi. Garis-garis dinamakan *ridge* sedangkan spasi di antara dua garis dinamakan *valley*. Ciri yang unik dari sidik jari ini dinamakan *minutiae*. *Minutiae* ini terdiri dari *ending*, *core* dan *bifurcation*.

Berdasarkan permasalahan yang telah diuraikan di atas, maka penulis tertarik untuk melakukan penelitian dengan judul “**Aplikasi Biometrika Pengenalan Citra Sidik Jari Dengan Metode *Minutiae* dan *Artificial Neural Network*”**”.

## II. LANDASAN TEORI

### A. Pengolahan Citra Digital

Pengolahan citra adalah pemrosesan citra, khususnya dengan menggunakan komputer, menjadi citra yang kualitasnya lebih baik. Umumnya, operasi-operasi pada pengolahan citra diterapkan pada citra bila perbaikan atau memodifikasi citra perlu dilakukan untuk meningkatkan kualitas penampakan atau untuk menonjolkan beberapa aspek informasi yang terkandung di dalam citra, kemudian elemen di dalam citra perlu dikelompokkan, dicocokkan, atau diukur, sebagian citra perlu digabung dengan bagian citra yang lain [1].

### B. Pengenalan Sidik Jari

Pengenalan sidik jari (*fingerprint recognition*) menurut Bambang (dalam Verawati, 2014) merupakan teknologi yang amat sering dan umum digunakan oleh khalayak ramai dalam identifikasi identitas seseorang, bahkan telah menjadi teknologi yang cukup diandalkan karena efektifitas

dan penggunaannya yang mudah. Sidik jari (*fingerprint*) adalah hasil reproduksi tapak jari baik yang sengaja diambil, di capkan dengan tinta, maupun bekas yang ditinggalkan pada benda karena pernah tersentuh dengan kulit telapak tangan atau kaki. Karena keunikannya itulah sidik jari (*fingerprint*) dapat digunakan untuk menjadi identitas utama yang digunakan dalam mengenali seseorang [2].

### C. Image Processing

*Image Preprocessing* merupakan tahapan awal dari proses verifikasi sidik jari. Hal tersebut untuk meningkatkan kejelasan struktur *ridge* dan dapat meningkatkan kinerja metode ekstraksi *minutiae*, proses ini *preprocessing* ini terdiri dari tahapan yaitu :

#### 1. Cropping

*Cropping* pada pengolahan citra berarti memotong satu bagian dari citra sehingga diperoleh citra yang diharapkan. Ukuran pemotongan citra tersebut berubah sesuai dengan ukuran citra yang diambil. *Cropping* dilakukan pada koordinat  $(x, y)$  sampai pada koordinat  $(m, n)$ .

#### 2. Thresholding

*Thresholding* adalah konversi dari citra hitam-putih menjadi citra biner dilakukan dengan operasi pengambangan (*thresholding*).

$$f_{B(i,j)} = \begin{cases} 1, & f_g(i,j) \leq T \\ 0, & \text{lainnya} \end{cases} \dots\dots\dots(1)$$

#### 3. Thinning

*Thinning* berguna untuk mengurangi *thresholded* citra output yang dihasilkan dari *edge detector*, menjadi garis dengan ukuran ketebalan satu piksel saja.

#### 4. Resizing

Operasi pengubahan ukuran atau *resizing* pada citra digital adalah operasi memperbesar

maupun memperkecil ukuran citra. Teknik yang digunakan dalam operasi ini adalah teknik interpolasi [3].

### D. Minutiae

Minutia (*Minutiae*) menurut Bambang (dalam Verawati, 2014) berasal dari bahasa latin "minutus" yang berarti kecil, minutia adalah perpotongan guratan-guratan (*ridge*) kulit yang membentuk sidik jari manusia

Semua point pada *minutiae* dapat didefinisikan dengan rumus berikut :

$$CN = 0.5 \sum_{i=1}^9 |P_i - P_{i+1}|, P_9 = P_1 \dots\dots\dots(2)$$

Dimana  $P_i$  adalah piksel di P. Semua piksel akan beralawanan dengan arah jarum jam [2].

### E. BackPropagation Neural Network (BPNN)

BPNN merupakan algoritma pelatihan terbimbing yang mempunyai banyak lapisan. BPNN menggunakan *error output* untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward*). Untuk mendapatkan *error* ini, tahap perambatan maju (*forward propagation*) [4].

#### 1. Arsitektur Backpropagation

Arsitektur algoritma *Backpropagation* terdiri dari tiga layer, yaitu *input layer*, *hidden layer*, dan *output layer*.. Algoritma *Backpropagation* ini digunakan fungsi aktivasi *sigmoid biner*, karena output yang diharapkan bernilai antara 0 sampai 1.

#### 2. Inisialisasi Bobot

Inisialisasi Bobot Terdapat dua cara untuk menginisialisasi bobot, yaitu inisialisasi secara random

#### 3. Proses Feed Forward

Pada dasarnya proses algoritma *Backpropagation* terdiri dari komputasi maju (*feed forward*) dan komputasi balik

(*Backpropagation*). Algoritma proses *feed forward* adalah sebagai berikut:

- a. Unit input ( $x_i, i = 1, 2 \dots n$ )
- Menerima *input*  $X_i$
  - Mengirimkannya ke semua unit *layer* di atasnya (*Hidden layer*).

- b. Unit Hidden ( $z_j, j = 1, 2 \dots n$ )
- Menghitung semua sinyal *input* dengan bobotnya

$$z_{-in_j} = v_{oj} + \sum x_i v_{ij} \dots\dots\dots (3)$$

- Menghitung nilai aktivasi setiap unit *hidden* sebagai output unit *hidden*

$$z_j = f(z_{-in_j}) = \frac{1}{1 + e^{-z_{-in_j}}} \dots\dots\dots (4)$$

- c. Unit Output ( $y_k, k = 1, 2, \dots n$ )
- Menghitung semua sinyal *input*-nya dengan bobotnya:

$$y_{in_k} = w_{ok} + \sum z_j w_{jk} \dots\dots\dots (5)$$

4. Algoritma Proses *Backpropagation*

- a) Unit Output ( $y_k, k = 1, 2, \dots m$ )
- Menghitung besarnya koreksi bobot unit *output*:

$$\delta_k = (T_k - y_k) f'(y_{in_k}) \dots\dots\dots (6)$$

- Menghitung besarnya koreksi bias *output*:

$$\Delta w_{ok} = \alpha \delta_k \dots\dots\dots (8)$$

- b) Unit *Hidden* ( $z_j = 1, 2, \dots p$ )
- Menghitung semua koreksi *error*:

$$\delta_{-in_j} = \sum \delta_k W_{jk} \dots\dots\dots (9)$$

- Menghitung nilai aktivasi koreksi *error*:

$$\delta_j = \delta_{-in_j} f'(z_{-in_j}) \dots\dots\dots (10)$$

- Menghitung koreksi bobot unit *hidden*:

$$\Delta v_{ij} = \alpha \frac{\partial E(v_{ij})}{\partial v_{ij}} = \alpha \delta_j X_i \dots\dots\dots (11)$$

- Menghitung koreksi *error* bias unit *hidden*:

$$\Delta V_{oj} = \alpha \delta_j \dots\dots\dots (12)$$

- c) *Update* bobot dan bias
- Unit *Output* ( $y_k, k = 1, 2, \dots m$ )
- Meng-*update* bobot dan biasnya ( $j = 0, \dots p$ ):

$$w_{jk} = w_{jk} + \Delta w_{jk}$$

$$w_{ok} = w_{ok} + \Delta w_{ok} \dots\dots\dots (11)$$

- Unit *hidden* ( $z_j, j = 1, \dots p$ )
- Meng-*update* bobot dan biasnya ( $i = 0, \dots, n$ ):

$$V_{ij} = V_{ij} + \Delta V_{ij}$$

$$V_{oj} = V_{oj} + \Delta V_{oj} \dots\dots\dots (12)$$

5. *Stopping Condition*

Terdapat dua kondisi *stopping* pada algoritma *Backpropagation* ini, yaitu:

$$E = 0 \sum_{k=0}^k (T_k - Y_k)^2 \dots\dots\dots (13)$$

F. *Matrix Laboratory* (Matlab)

*Matrix Laboratory* (Matlab) merupakan salah satu bahasa pemrograman yang dikembangkan oleh MathWorks. Matlab tidak hanya berfungsi sebagai bahasa pemrograman, tetapi juga sebagai alat visualisasi, yang berhubungan langsung dengan ilmu Matematika. Oleh Karena itu, Matlab semakin banyak digunakan oleh para *progammer* dalam membuat program [5].

G. *Uninfied Modeling Language* (UML)

*Unifed Modeling Language* (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OO). Berdasarkan sifatnya, jenis diagram UML ada 9, diantaranya *class diagram, sequence diagram, usecase diagram, activity diagram, object diagram, component*

diagram, collaboration diagram, dan deployment diagram [6].

### III. METODE PENELITIAN

#### A. Jenis Penelitian

Pada penelitian ini menggunakan penelitian terapan. Penelitian terapan merupakan penelitian yang dilakukan dengan tujuan menerapkan, menguji, dan mengevaluasi kemampuan suatu teori yang diterapkan dalam memecahkan masalah-masalah praktis [7].

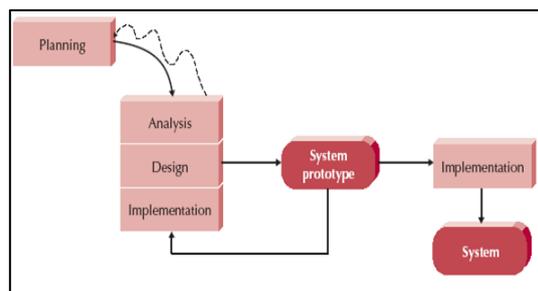
Penelitian ini juga dilakukan untuk mengetahui tingkat akurasi penggunaan metode *Minutiae* dan *Artificial Neural Network Backpropagation* dalam ekstraksi citra sidik jari dan menghitung selisih jarak antara citra masukan dengan citra *database*.

#### B. Metode Pengumpulan Data

Dalam pengumpulan data pustaka yang dilakukan dengan mengumpulkan data dan informasi yang digunakan sebagai acuan dalam pembuatan aplikasi pengenalan citra sidik jari. Data dan informasi berupa laporan penelitian, buku-buku ilmiah, skripsi, jurnal dan sumber-sumber tertulis lainnya yang dapat berhubungan dengan pemahaman metode yang digunakan dalam penelitian ini dan pembuatan aplikasi pada Matlab.

#### C. Metode Pengembangan Sistem

Pengembangan aplikasi yang digunakan pada penelitian ini adalah metode *prototyping*. Metodologi *prototyping* melakukan tahap analisis, perancangan dan implementasi secara bersamaan, dan ketiga tahap tersebut dilakukan secara berulang dalam siklus hingga sistem selesai atau lengkap Yang ditunjukkan pada Gambar 1 [8].



Gambar 1. Metodologi *Prototyping* (Dennis, 2005)

Adapun tahapan yang dilakukan dalam pengembangan aplikasi ini secara garis besar adalah sebagai berikut :

##### 1. Perencanaan

Perencanaan dimulai dengan menentukan aplikasi yang akan dibangun dalam penelitian ini, yaitu aplikasi yang dapat melakukan pengenalan sidik jari menggunakan *Minutiae* dan *Artificial Neural Network Backpropagation*.

##### 2. Analisis Kebutuhan

Analisa kebutuhan bertujuan untuk mendefinisikan format seluruh perangkat lunak, mengidentifikasi semua kebutuhan dan garis besar sistem yang akan dibuat.

Aplikasi yang akan dibangun membutuhkan adanya antarmuka sebagai bentuk interaksi manusia dan komputer dengan memberikan masukan dan menampilkan keluaran. Adapun analisis kebutuhan aplikasi yang terkait dengan hal tersebut adalah sebagai berikut:

###### a.) Kebutuhan masukan

Data masukan yang dibutuhkan dalam aplikasi ini adalah sebuah citra digital dengan format BMP dan berukuran awal 300x 300 dan ketika di *resize* 120 x 90.

###### b) Kebutuhan keluaran

Data keluaran yang diharapkan dari aplikasi ini adalah informasi mengenai

identitas berdasarkan deteksi sidik jari yang dilakukan.

c) Kebutuhan antarmuka

Kebutuhan antarmuka bagi aplikasi ini ialah kemudahan bagi pengguna untuk memahami dan menggunakan aplikasi ini sesuai dengan masalah yang menjadi topik penelitian.

3. Perancangan Aplikasi

Perancangan adalah tahap konseptualisasi yang mengharuskan analisis dalam perancangan perangkat lunak untuk benar-benar mengetahui hal-hal yang menjadi kebutuhan dan harapan pengguna sehingga dapat memuaskan kebutuhan dan harapannya. Pembuatan aplikasi dalam penelitian ini memerlukan beberapa perangkat lunak dan perangkat keras yang dapat menunjang penelitian berjalan dengan baik, yaitu :

a. Perangkat Lunak

Perangkat lunak yang mendukung dalam penelitian ini adalah berupa Sistem Operasi *Windows Seven (10) 64 bit*, *Matlab*, *Microsoft Word 2010*, *Microsoft Office Visio 2010* untuk pembuatan diagram alir sistem, *MySQL* untuk merancang *database*, *Google Chrome*, dan *Astah Community* untuk pembuatan UML perancangan sistem.

b. Perangkat Keras

Sedangkan perangkat keras yang digunakan untuk mendukung dalam penelitian ini adalah 1 unit Laptop Toshiba dengan spesifikasi monitor VGA atau SVGA dan *processor Intel® Core™ i3-2350M*, RAM 2 GB, *Hardisk 500 GB* dan printer *Canon PIXMA ip2770*.

4. Membangun *Prototyping*

Membangun *prototyping* dengan membuat perancangan sementara yang berfokus pada penyajian kepada pengguna (misalnya dengan membuat format masukan dan keluaran).

5. Evaluasi *Prototyping*

Evaluasi *prototyping* dilakukan untuk mengetahui apakah *prototyping* yang sudah dibangun telah sesuai dengan keinginan. Jika sudah sesuai maka dilanjutkan dengan implementasi aplikasi. Jika tidak *prototyping* direvisi dengan mengulang langkah sebelumnya dimulai dari analisis kebutuhan.

6. Implementasi

Tahap ini merupakan tahapan nyata dalam pengerjaan aplikasi berdasarkan *prototyping* yang telah disepakati.

7. Pengujian

Setelah aplikasi menjadi perangkat lunak yang siap pakai atau lengkap, aplikasi ini harus diuji sebelum digunakan. Pengujian yang dilakukan dalam penelitian ini menggunakan dua metode pengujian yaitu *white box testing* dan *black box testing*.

a. *White Box Testing*

Pengujian ini dilakukan dengan meneliti kode-kode program yang telah dibuat dan menganalisis apakah ada kesalahan atau tidak. *Testing* yang dilakukan untuk mengetahui kesesuaian suatu komponen terhadap desain [9].

b. *Black Box Testing*

Pengujian ini dilakukan dengan mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari aplikasi yang telah dibuat. Pengujian *Black Box* adalah dengan *Equivalence Partitioning*. *Equivalence Partitioning* dapat diasumsikan bahwa masukan yang sama

akan menghasilkan respon yang sama pula [10].

#### 8. Penggunaan dan Pemeliharaan

Pada tahap ini aplikasi yang telah diuji siap digunakan oleh pengguna. Jika terdapat pengembangan fungsional dari aplikasi yang diinginkan oleh pengguna, maka dapat dilakukan pemeliharaan lanjutan.

### IV. ANALISIS DAN PERANCANGAN

#### A. Identifikasi Masalah

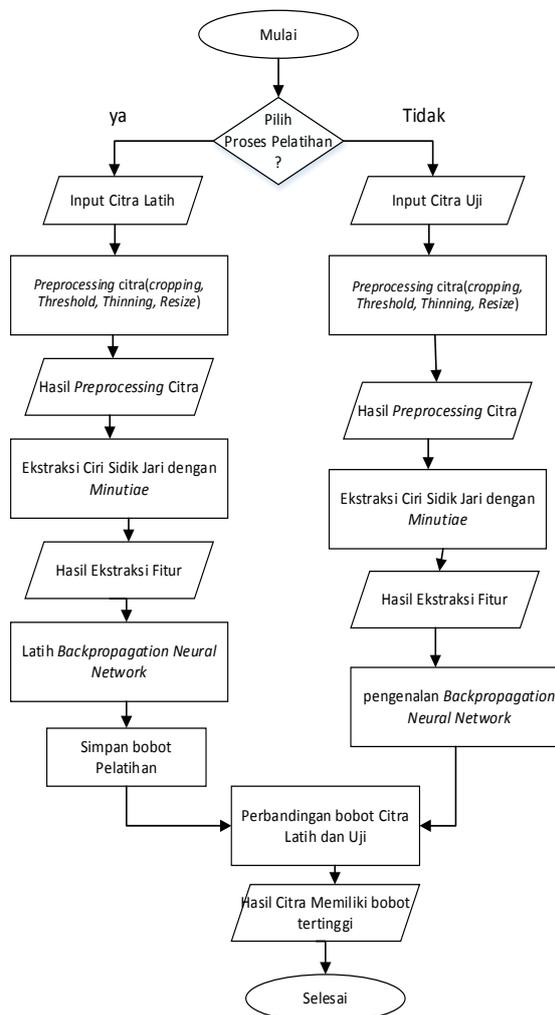
Semakin berkembangnya teknologi manusia membutuhkan berbagai macam aplikasi yang berguna membantu mempermudah pekerjaan manusia. Aplikasi tersebut dibuat sesuai dengan fungsi dan tujuan awal masing-masing. Salah satu aplikasi yang membantu pekerjaan manusia adalah aplikasi pengenalan citra melalui sidik jari. Cara kerja aplikasi pengenalan citra sidik jari mencari data yang memiliki kemiripan yang paling dekat dengan citra latih yang ada didalam *database*.

Agar tidak terjadi lagi kesalahan dalam pengenalan citra sidik jari, maka dibuatlah aplikasi pengenalan ini. Dalam pengenalan ini ada beberapa teknik untuk *preprocessing* yang dilakukan yaitu *cropping*, *threshold*, *thinning* dan *resize*. Kemudian, melakukan ekstraksi fitur menggunakan metode *Minutiae*. Setelah mendapatkan nilai-nilai dapat ekstraksi fitur untuk melakukan proses pengenalan melalui *Artificial Neural Network*. Hal tersebut untuk mendapatkan tingkat akurasi yang tinggi dalam pengenalan citra sidik jari.

#### B. Cara Kerja sistem

Secara garis besar cara kerja sistem yang dibangun untuk aplikasi biometrika pengenalan citra sidik jari dengan metode *minutiae* dan

*artificial neural network backpropagation* dalam Gambar 2.



Gambar 2 Diagram Alir Aplikasi pengenalan Citra Sidik Jari

#### C. Perancangan Model UML (Unified Modeling Language)

Perancangan model UML ditunjukkan untuk memberikan gambaran secara umum tentang sistem yang akan dibangun. Perancangan sistem Aplikasi Biometrika Pengenalan Citra Sidik Jari Dengan Metode *Minutiae* dan *Artificial Neural Network Backpropagation* ini menggunakan model diagram UML, yaitu :

## 1. Use Case Diagram

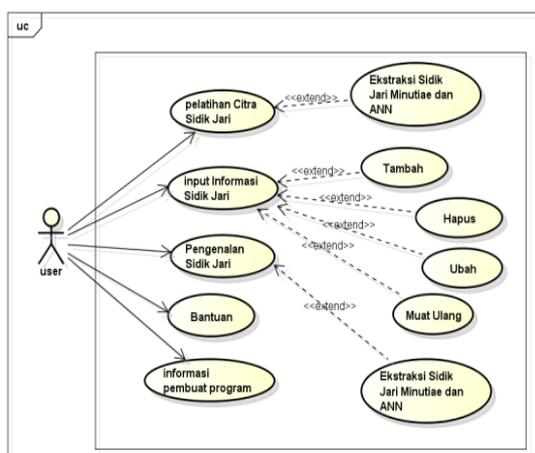
*Use case diagram* adalah diagram yang bekerja mendeskripsikan tipikal interaksi antara pengguna dengan sebuah sistem melalui sebuah cerita bagaimana sebuah sistem digunakan. *Use case diagram* terdiri dari sebuah aktor dan interaksi yang dilakukannya. Aktor tersebut dapat berupa manusia, perangkat keras, sistem lain, ataupun yang berinteraksi dengan sistem.

### a. Identifikasi aktor dan interaksinya dengan sistem

Aplikasi biometrika pengenalan citra sidik jari metode *Minutiae* dan *Artificial Neural Network Backpropagation* digunakan oleh satu pengguna. Pengguna sistem ini disebut dengan *actor*. Pengguna sistem ini adalah pengguna umum (*user*).

### b. Pembuatan Use case Diagram

*Use case diagram* dari sistem yang dibangun seperti Gambar 3 sebagai berikut :

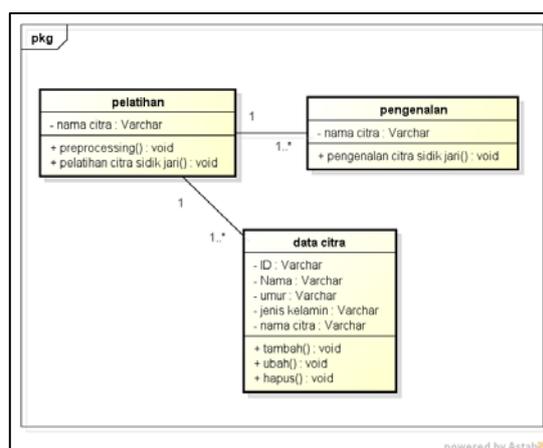


Gambar 3 Use case Diagram

Pada Gambar 3 menjelaskan *Use Case Diagram* dari aplikasi. *User* dapat melakukan proses data citra yaitu tambah, ubah, hapus dan muat ulang citra sidik jari, kemudian proses pelatihan citra sidik jari. Proses

pelatihan terdiri dari *preprocessing* citra yaitu : *cropping*, *threshold*, *thinning* dan *resize*. Setelah itu ekstraksi ciri dari sidik jari dengan *Minutiae* dan pelatihan dengan *Backpropagation*. Kemudian proses pengenalan, proses ini sama dengan proses yang ada pada pelatihan.

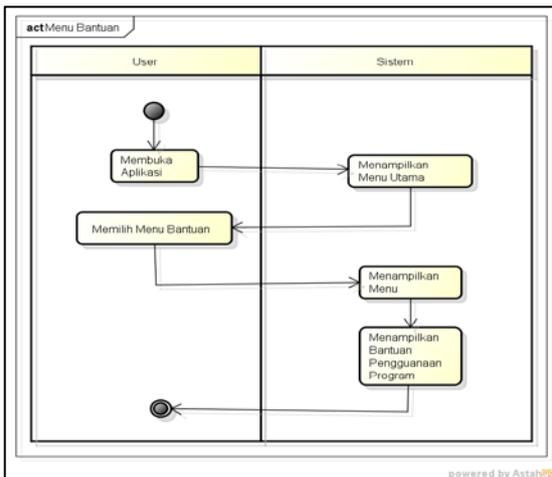
## 2. Class Diagram



Gambar 4 Class Diagram

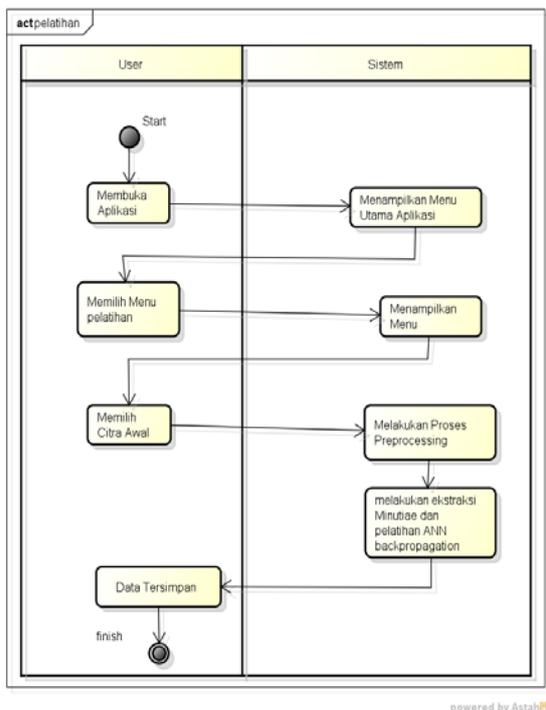
Pada Gambar 4.5 *Class Diagram* pada aplikasi pengenalan citra sidik jari. Pada aplikasi ini terdapat *Class* data citra yang mempunyai atribut nama citra, nama, jenis kelamin dan umur. Proses yang dapat dilakukan pada aplikasi ini yaitu proses tambah citra, ubah citra, hapus citra dan ulangi citra. *Class* pelatihan hanya mempunyai atribut nama citra dan dapat melakukan proses pelatihan untuk citra. Pada aplikasi ini banyak citra dapat melakukan satu pelatihan. Sedangkan kelas pengenalan adalah *class* yang digunakan untuk melakukan proses pengenalan citra sidik jari. *Class* pelatihan dapat mengenali dari beberapa data pelatihan.

### 3. Activity Diagram



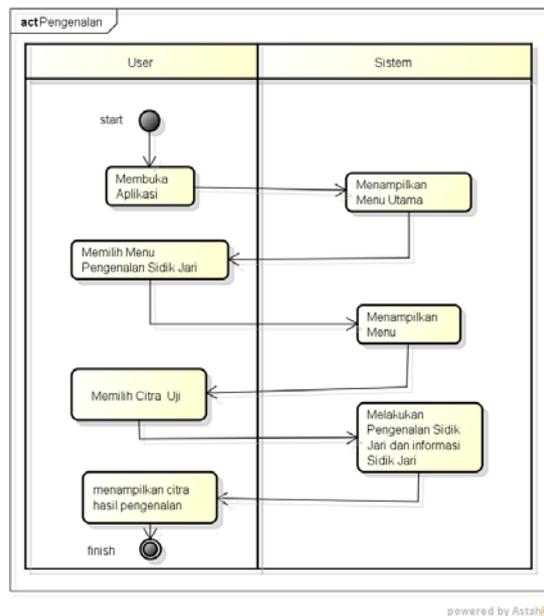
Gambar 5 Activity Diagram Bantuan

Gambar 5 adalah *activity diagram* tentang petunjuk sistem. *Activity diagram* petunjuk sistem diawali dari pengguna atau *user* memilih menu petunjuk sistem pada menu utama sistem. Dan setelah memilih menu petunjuk sistem maka akan tampil informasi cara penggunaan sistem yang dapat mempermudah dalam penggunaan aplikasi ini.



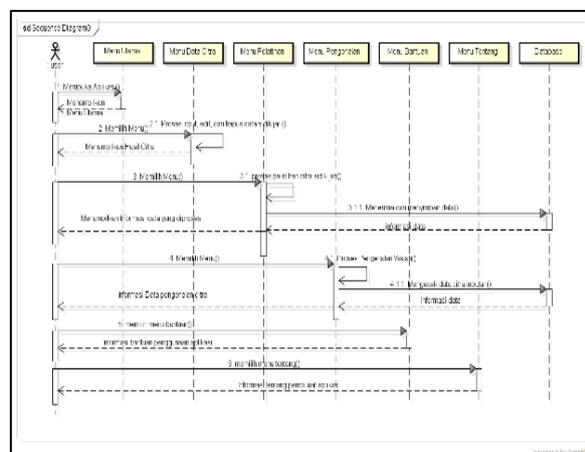
Gambar 6 Activity Diagram Pelatihan Citra

Gambar 6 merupakan aktifitas untuk pemilihan menu pelatihan citra yang akan digunakan oleh *user*. Aktifitas ini dimulai dari pengguna atau *user* mengakses aplikasi biometrika pengenalan citra sidik jari, kemudian pengguna memilih menu pelatihan. Maka *user* akan menginputkan citra yang akan mendapatkan proses *preprocessing*. Kemudian citra tersebut akan disimpan didalam *database* sebagai citra latih pada aplikasi ini.



Gambar 7 Activity diagram pengenalan sistem

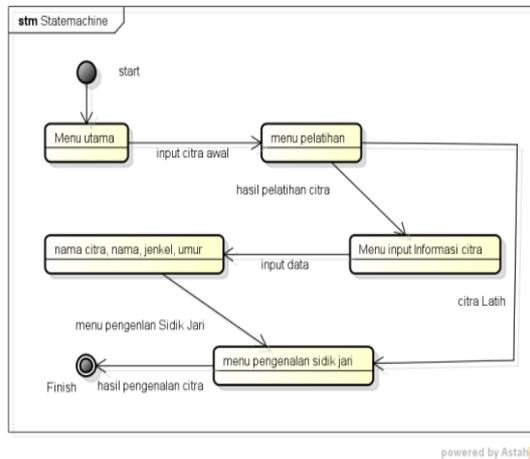
### 4. Sequence Diagram



Gambar 8 Activity diagram pengenalan sistem

Gambar 8 merupakan gambaran *sequence diagram* proses pengenalan sidik jari pada aplikasi biometrika pengenalan sidik jari.

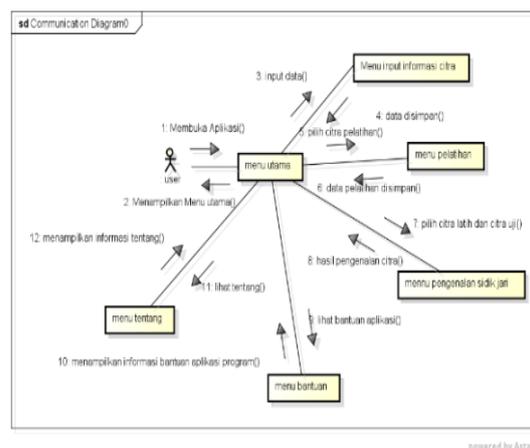
### 5. State Chart Diagram



Gambar 9 State Chart Diagram

*State chart diagram* menghubungkan *event-event* dan *state-state*. Ketika suatu *event* diterima, *state* berikutnya bergantung pada *state* yang sekarang ada. Gambar 9 menggambarkan diagram *state* untuk aplikasi pengenalan citra sidik jari.

### 6. Communication Diagram atau Collaboration Diagram

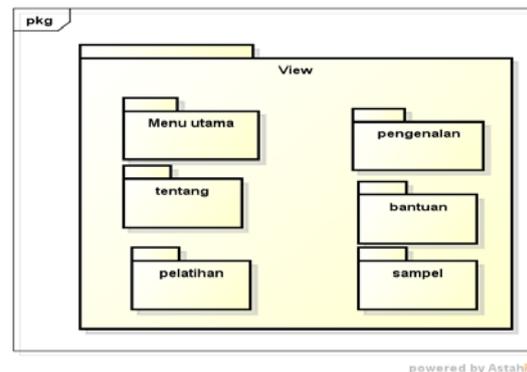


Gambar 10. Communication Diagram

Pada *communication diagram* setiap *event* mengirimkan informasi dari suatu objek ke

objek yang lainnya. *User* akan memasukkan data yang diperlukan oleh aplikasi pengenalan sidik jari. Kemudian aplikasi akan mengenali citra yang dijadikan citra uji. Gambar 10 adalah *communication diagram* aplikasi biometrika pengenalan citra sidik jari.

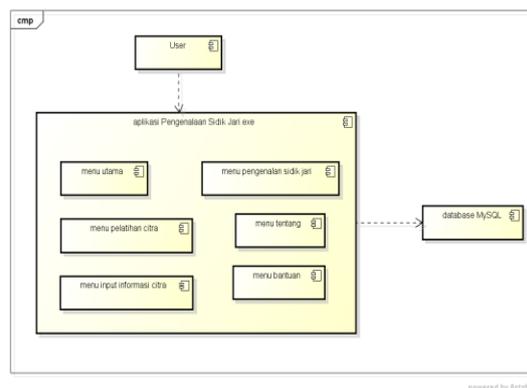
### 7. Package Diagram



Gambar 11. Package Diagram

Pada Gambar 11 terdapat beberapa paket yang mendukung aplikasi biometrika pengenalan citra sidik jari paket views merupakan paket pendukung dalam aplikasi yang dibangun. Hal ini dilakukan untuk mempermudah dalam mencari letak kesalahan dalam membangun aplikasi yang dibangun.

### 8. Component Diagram

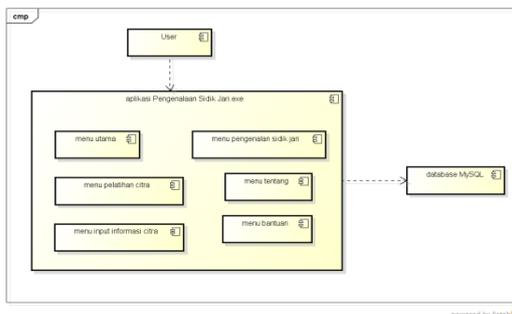


Gambar 12. Component Diagram

Pada Gambar 12 terdapat beberapa komponen yang dapat mendukung dalam pembuatan aplikasi biometrika pengenalan citra sidik jari, yaitu komponen pengguna,

komponen Aplikasi Biometrika Pengenalan Citra Sidik Jari dan komponen database MySQL. Komponen pendukung tersebut berguna untuk membantu dalam pembuatan aplikasi yang dibangun.

### 9. Component Diagram



Gambar 12. Component Diagram

Pada Gambar 12 terdapat beberapa komponen yang dapat mendukung dalam pembuatan aplikasi biometrika pengenalan citra sidik jari, yaitu komponen pengguna, komponen Aplikasi Biometrika Pengenalan Citra Sidik Jari dan komponen database MySQL. Komponen pendukung tersebut berguna untuk membantu dalam pembuatan aplikasi yang dibangun.

## V. PEMBAHASAN

### A. Menu Utama

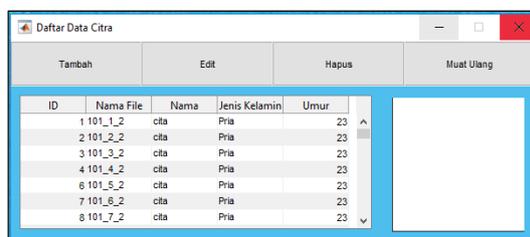
Menu utama pada aplikasi ini muncul pertama kali saat aplikasi ini dijalankan. Menu utama ditunjukkan pada Gambar 14.



Gambar 14. Menu Utama

### B. Menu Data Citra

Menu Data citra akan muncul setelah memilih submenu data citra 'Data Citra' pada menu utama ditekan. Berikut adalah tampilan dari menu tambah citra yang dapat dilihat pada Gambar 15.



Gambar 15. Menu Data Citra

### C. Menu Tambah Citra

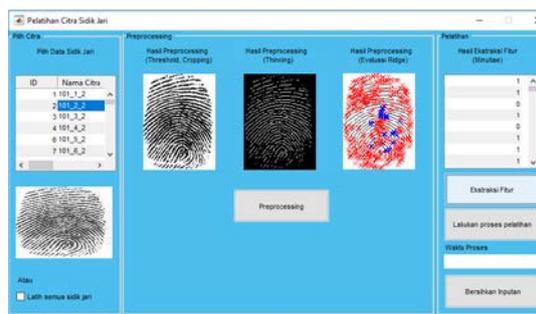
Menu cari citra akan muncul saat tombol 'tambah' dan 'ubah' pada menu data citra ditekan. Berikut adalah tampilan dari menu cari citra yang dapat dilihat pada Gambar 16.



Gambar 16. Menu Tambah Citra

### D. Menu Pelatihan Citra

Menu pelatihan citra adalah menu yang digunakan *user* untuk melakukan pelatihan pada citra sidik jari. Menu pelatihan cira dapat ditunjukkan pada Gambar 17.



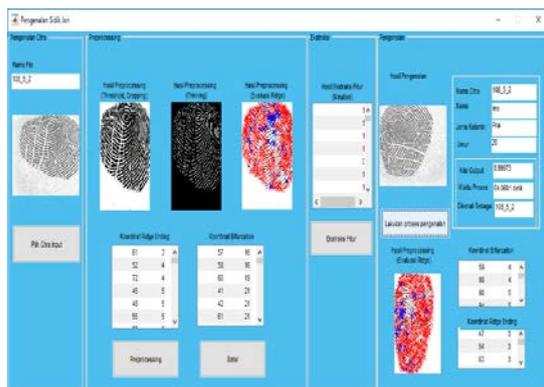
Gambar 17. Menu Pelatihan Citra

Pada menu pelatihan ini terdapat proses *preprocessing* yaitu *cropping*, *threshold*, *thinning* dan *resize* citra sidik jari menjadi 120 x 90.

Kemudian melakukan ekstraksi fitur *Minutiae* yang akan menghasilkan vektor berupa 1 untuk piksel berwarna hitam dan 0 untuk piksel berwarna putih. Vektor-vektor ini akan digunakan sebagai inputan pada *Backpropagation*. Kemudian nilai bobot yang dihasilkan dari *Backpropagation* akan disimpan pada *database*.

#### E. Menu Pengenalan Citra

Menu pengenalan citra adalah menu yang digunakan untuk melakukan proses pengenalan citra uji sidik jari. Menu pengenalan sidik jari dapat ditunjukkan pada Gambar 18.



Gambar 18 Menu Pengenalan Citra

Pada Gambar 18 menampilkan menu pengenalan dari aplikasi. Pada pengenalan tersebut nampak citra uji yang diujikan dengan melalui proses *cropping*, *threshold*, *thinning* dan *resize* menjadi 120x90 piksel. Setelah itu citra akan di ekstraksi dengan *Minutiae* untuk mendapatkan titik-titik dari *ridge ending* dan *bifurcation*.

Pada ekstraksi *Minutiae* ada ekstraksi *Minutiae* akan mencari 8 titik piksel tetangga dari titik pusat *ridge ending* atau *bifurcation*. Dari ekstraksi ini akan mendapatkan hasil berupa vektor yang bernilai 1 untuk piksel berwarna hitam dan 0

untuk piksel berwarna putih. Kemudian vektor tersebut akan disimpan dan dijadikan bobot input pada *Backpropagation*.

Setelah itu akan dilakukan perhitungan dengan *Backpropagation* untuk mendapatkan nilai *output* yang mendekati 1 yang akan dijadikan *output* untuk pengenalan citra. Pada aplikasi ini akan diketahui waktu proses yang dihasilkan selama proses berlangsung dan tingkat kecocokan dari citra uji dan citra latih yang didapat dari membandingkan nilai-nilai piksel yang sama dari citra uji dan citra latih.

## VI. PENUTUP

### A. Kesimpulan

Berdasarkan analisa perancangan sistem, implementasi, dan pengujian sistem, maka dapat disimpulkan bahwa :

1. Penelitian ini telah menghasilkan aplikasi biometrika pengenalan citra berbasis dekstop yang dapat digunakan sebagai referensi pengenalan identitas tambahan seseorang. Pada aplikasi yang dihasilkan diperoleh waktu 20 menit dalam proses pelatihan. Hal ini disebabkan oleh koordinat *ridge ending* dan *bifurcation* yang diekstraksi pada citra sidik jari berjumlah 9900 titik koordinat.
2. Berdasarkan hasil pengujian, aplikasi mampu mengenali citra sidik jari dengan metode *Minutiae* dan *Artificial Neural Network backpropagation* menunjukkan hasil pengujian sebagai berikut :
  - a) Persentase keberhasilan pengujian citra latih (di dalam *database*) yang dijadikan citra uji adalah 100%. Dari hasil pengujian ini dapat dibuktikan bahwa aplikasi dapat berfungsi secara optimal dalam mengenali citra

uji. Persentase keberhasilan pengujian dari citra uji di luar *database* mampu mengenali dengan yang ada didalam *database* adalah 67%. Dari hasil pengujian ini dapat dibuktikan bahwa aplikasi belum dapat berfungsi secara optimal. Karena posisi dan jarak dari citra uji yang sangat berbeda dengan citra latih yang menyebabkan koordinat *ridge ending* dan *bifurcation* dari citra uji dan citra latih sangat berbeda.

- b) Persentase keberhasilan pengujian citra uji dari *Flat capacitive sensor by Precise Biometrics Model Precise 100 SC* adalah 55%. Dari hasil pengujian ini dapat dibuktikan bahwa aplikasi belum dapat berfungsi secara optimal. Karena posisi, jarak pengambilan citra, area citra sidik jari dan ukuran citra dari citra uji *Flat Capacitive Sensor By Precise Biometrics Model Precise 100 SC* yang sangat berbeda dengan citra latih dari *Flat Optical Sensor Biometrika Fx2000* yang menyebabkan nilai-nilai piksel dan koordinat *ridge ending* dan *bifurcation* dari citra uji dan citra latih sangat berbeda.
- c) Persentase keberhasilan pengujian dengan citra uji diberi serangan yaitu, pada serangan rotasi 90° adalah 40% karena posisi citra yang sangat berbeda yang menyebabkan nilai-nilai piksel dan koordinat *ridge ending* dan *bifurcation* dari citra uji dan citra latih sangat berbeda. Pada *Noise Gaussian* 10% adalah 40%

karena adanya titik-titik *noise* yang membuat nilai piksel citra berubah. *Blur Gaussian* adalah 70% karena adanya titik-titik *blur* yang membuat nilai piksel citra berubah. Pada citra terpotong adalah 30% disebabkan oleh ukuran sehingga posisi piksel menjadi berubah. Dari hasil pengujian ini dapat dibuktikan bahwa aplikasi belum dapat berfungsi secara optimal.

#### B. Saran

Berdasarkan hasil dan pembahasan yang telah dijabarkan sebelumnya, maka saran yang dapat diberikan untuk pengembangan penelitian selanjutnya adalah sebagai berikut:

1. Aplikasi akan semakin baik apabila mampu menangani *noise* dan *cropping* sidik jari.
2. Penggunaan metode *Minutiae* belum dapat digunakan secara optimal dalam ekstraksi fitur, sebaiknya dapat menggunakan metode lain seperti PCA karena efisien dalam pengenalan pola atau metode lainnya pada penelitian selanjutnya.
3. Agar aplikasi ini dapat terus berkembang, ada baiknya penelitian ini dilanjutkan dengan menambahkan fitur untuk mempercepat proses pelatihan.

#### REFERENSI

- [1] Purwandari, E. P. (2008). *Aplikasi Transformasi Wavelet Dalam Peningkatan Rekonstruksi Citra Sidik Jari*. Bengkulu: Universitas Bengkulu.
- [2] Verawati, R. (2014). Pengenalan Sidik Jari Berdasarkan Struktur *Minutiae* Dengan Metode *BackPropagation*. *Jurnal Pelita Informatika Volume 7, Nomor 1*.
- [3] Munir, Rinaldi. (2004). *Pengolahan Citra Digital dengan Pendekatan Algoritmik*. Informatika, Bandung.

- [4] Nurmila Nazla, Sarwoko Eko Adi, dan Sugiharto Aris. 2010. Algoritma Backpropagation Neural Network untuk Pengenalan Pola Karakter Huruf Jawa. *Jurnal Masyarakat Informatika Volume 1, Nomor 1*.
- [5] Paulus, E & Nataliani, Y. 2007. *GUI Matlab*. ogyakarta :Andi
- [6] Widodo, P. P., & Herlawati. (2011). *Menggunakan UML*. Bandung: Informatika.
- [7] Sugiyono. (2014). *Metode Penelitian Kuantitatif, Kualitatif, dan R&D*. Bandung: Alfabeta.
- [8] Dennis, A. W. (2005). *Sistem Analysis and Design with UML Version 2.0*. United States of America: John Willey & Sons, Inc.
- [9] Rosa, & Shalahuddin, M. (2011). *Modul Pembelajaran Rekayasa Perangkat Lunak*. Bandung: Modula
- [10] Lewis, W. E.(2005). *Software Testing and Continuous Quality Improvement Second Edition*. Boca Raton London New York Wahington, D.C.: CRC Press LLC.