

## BAB III

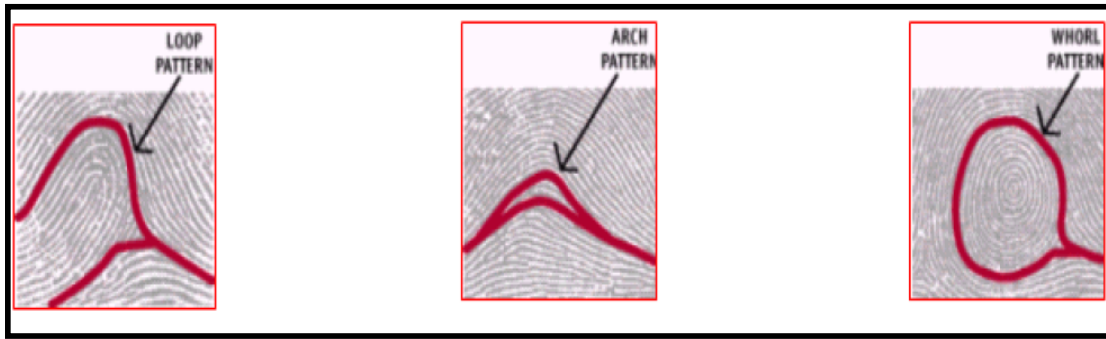
### LANDASAN TEORI

#### 3.1 Sidik jari

Sidik jari adalah hasil reproduksi tapak jari yang baik yang sengaja diambil, dicapkan dengan tinta, maupun bekas yang ditinggalkan pada benda karena pernah tersentuh kulit telapak tangan atau kaki. Kulit telapak adalah kulit pada bagian telapak tangan mulai dari pangkal pergelangan sampai kesemua ujung jari, dan kulit bagian dari telapak kaki mulai dari tumit sampai ujung jari yang mana pada daerah tersebut terdapat garis halus menonjol yang keluar satu sama lain yang dipisahkan oleh celah atau alur yang membentuk struktur tertentu. Fungsi dari sidik jari itu sendiri adalah untuk memberikan gaya gesek lebih besar agar jari dapat memegang benda-benda lebih erat

Teknologi identifikasi sidik jari berdasarkan fakta bahwa setiap sidik jari adalah unik. Verifikasi system menggunakan kontur dan *flat image* dari jari dan membandingkannya. Sidik jari manusia biasanya diklasifikasikan berdasarkan Henry System :

1. *Loop kiri*
2. *Loop kanan*
3. *Arch*
4. *Tented arch*
5. *Whorl*



Gambar 3.1 Pola sidik jari

Biasanya 2/3 jari berbentuk loop, 1/3 berbentuk *whorl*, dan 5-10% berbentuk

*Arches*.



Gambar 3.2 Sidik jari struktur hubungan gesekan

Sensor sidik jari akan menangkap kontur kulit jari. Kontur jari adalah hal yang sangat penting karena bisa menghindari kecurangan pemalsuan dengan menggunakan *foto copy* sidik jari, dengan karet, atau cetakan yang lain.

Sifat yang dimiliki sidik jari antara lain :

1. *Perennial nature*, yaitu guratan-guratan pada sidik jari yang melekat pada kulit manusia seumur hidup.
2. *Immutability*, yaitu sidik jari seseorang tidak pernah berubah, kecuali mendapatkan kecelakaan yang serius.
3. *Individuality*, pola sidik jari adalah unik dan berbeda untuk setiap orang.

Ciri khas sidik jari yang digunakan adalah guratan sidik jari yang dapat diidentifikasi dengan cara menganalisis detail dari guratan-guratan sidik jari yang dinamakan dengan “*minutiae*”

### **3.2 AFIS (*Automated Fingerprint Identification Systems*)**

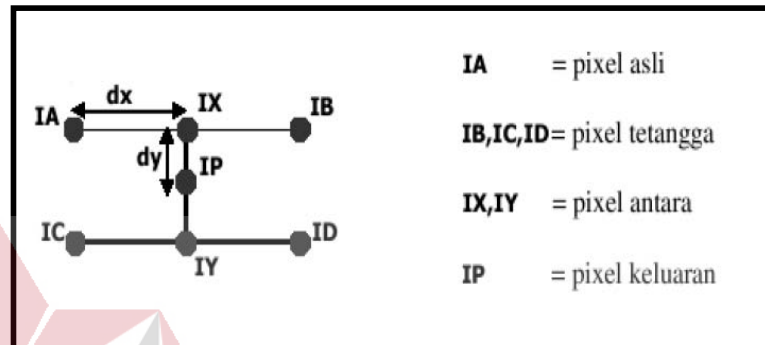
AFIS merupakan proses pengkalsifikasian sidik jari yang berbasis komputer. AFIS merupakan otomatisasi dari teori Galton-Henry, meskipun pada perkembangannya ada beberapa modifikasi dan penyesuaian pada teori (Carlton, 2003).

Secara garis besar AFIS banyak menggunakan algoritma *image processing* untuk mengkalsifikasi dan mengidentifikasi sidik jari, seperti contohnya untuk mendeteksi Galton Detail. Berikut merupakan beberapa algoritma *image processing* yang digunakan dalam aplikasi ini.

#### **3.2.1 Interpolasi Bilinier**

Proses Interpolasi ini diperlukan untuk perbesaran citra yang nantinya dapat digunakan untuk pelaporan dan analisa manual. Metode Interpolasi Bilinear pada proses registrasi citra menggunakan dua persamaan linier dimana proses

Interpolasi dilakukan dengan memperhitungkan pengaruh distribusi tingkat keabuan *pixel* (*gray value*) tetangga atau perkiraan rata-rata tertimbang dari empat *pixel* dalam proses Interpolasi (purwadhi, 2001). *Pixel* yang mempengaruhi adalah sejumlah empat *pixel* yang mengelilinginya.



Gambar 3.3 Interpolasi Biner

Gambar diatas menjelaskan dimana nilai kecerahan IX berada diantara dua pixel A dan B, IX berjarak  $dx$  dari pixel A, maka diberikan persamaan :

$$IX = IA + (IB - IA) dx$$

Cara yang sama untuk IY :

$$IY = IC + (ID - IC) dx$$

Dengan demikian, dapat diperoleh nilai kecerahan suatu pixel yang , diletakkan pada citra keluaran (terkoreksi) dengan nilai kecerahan IP, yang mengikuti persamaan :

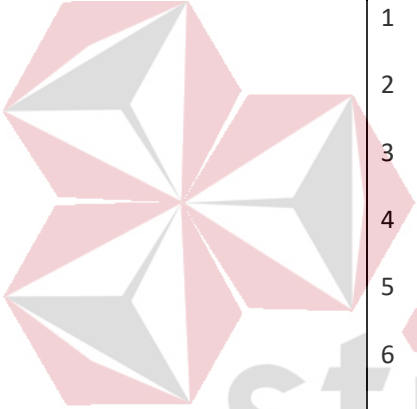
$$IP = IX + (IY - IX) dy$$

Sehingga jarak  $dx$  dan  $dy$  adalah jarak terdekat dari empat buah pixel yang mengelilinginya dan diperoleh dari fungsi transformasi koordinat.

### 3.2.2 Image Enhancement

Proses ini digunakan untuk mempertajam dan memperbaiki citra input sidik jari. Ada banyak algoritma tentang *image enhancement*, namun pada aplikasi ini yang akan digunakan adalah *Histogram Equalization*.

Hal pertama yang dilakukan pada metode ini adalah membangun *table* frekuensi dari semua nilai *pixel* dalam citra. Seperti contoh :



Gray level	Frequency
0	5
1	10
2	15
3	20
4	16
5	12
6	15
7	7
Total 8	100

Tabel 3.1 Contoh tabel *Histogram*

Setiap frekuensi dalam table mencerminkan banyaknya *pixel* dengan nilai gray level tersebut dala sebuah citra.

### 3.2.3 Thresholding

*Thresholding* adalah menentukan suatu ambang batas (*threshold*) dimana *threshold* tersebut nantinya menggolongkan nilai dari *gray value* menjadi dua

bagian, untuk kemudian mengubahnya menjadi hitam atau putih sehingga didapatkan citra hitam putih. Tujuan dari mengubah citra menjadi hitam putih adalah untuk memudahkan mendeteksi titik-titik minus seperti pada teori Galton-Henry.

### 3.3 DSS (*Decision Support System*)

Menurut Little (1970) mendefinisikan DSS (*Decision Support System*) atau Sistem Pendukung Keputusan sebagai sekumpulan prosedur berbasis model untuk data pemrosesan dan penilaian guna membantu pihak manajer mengambil keputusan.

Menurut Alter (1980) mendefinisikan Sistem Pendukung keputusan dengan membandingkannya dengan sistem elektronik data processing tradisional pada lima dimensi. Bonzze yang merupakan salah satu pemikir dalam bidang DSS (*Decision Support System*) menjelaskan bahwa DSS sebagai sistem yang berbasis komputer yang terdiri dari tiga komponen yang saling berinteraksi : sistem bahasa, sistem pengetahuan, dan sistem pemrosesan masalah.

Jenis-jenis keputusan menurut Herbert A. Simon, yaitu

- a. Keputusan Terprogram, bersifat berulang dan rutin, sedemikian sehingga suatu prosedur pasti telah dibuat ulang untuk menanganinya.
- b. Keputusan Tak Terprogram, bersifat baru, tidak terstruktur dan jarang konsekuensi. Tidak ada metode pasti untuk menangani masalah ini.

Tahap-tahap pengambilan keputusan menurut Simon, yaitu

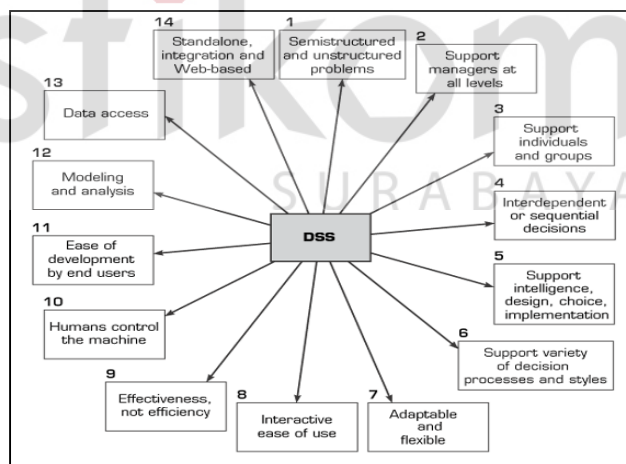
- a. Kegiatan Intelijen, mengamati lingkungan mencari kondisi-kondisi yang perlu diperbaiki

- b. Kegiatan merancang, menemukan, mengembangkan dan menganalisis berbagai alternative tindakan yang mungkin
- c. Kegiatan memilih, memilih satu rangkaian tindakan tertentu dari beberapa yang tersedia
- d. Kegiatan menelaah, menilai pilihan-pilihan yang lalu.

Tujuan pelaksanaan Sistem Pendukung Keputusan :

- a. Membantu manajer atau pemimpin tertinggi dalam pembuatan keputusan untuk memecahkan masalah semi terstruktur
- b. Mendukung penilaian manajer atau pemimpin tertinggi bukan mencoba menggantikan fungsi dari pemimpin
- c. Meningkatkan efektifitas pengambilan keputusan manajart atau pemimpin daripada efesuensinya.

Karakteristik & Kemampuan DSS (*Decision Support System*)



Gambar 3.4 Gambar karakteristik dan kapabilitas DSS

1. DSS memberi dukungan bagi pengambil keputusan untuk menyelesaikan masalah yang semi terstruktur atau tidak terstruktur.

2. Mendukung berbagai tingkatan manajemen yang berbeda.
3. Untuk individu dan juga bagi kelompok orang.
4. Untuk keputusan yang berurutan atau saling berkaitan.
5. Mendukung berbagai fase pengambilan keputusan, *intelligence*, *design*, *choice* dan *implementation*.
6. Mendukung pengambilan keputusan dan *style* yang berbeda-beda.
7. DSS dapat beradaptasi sepanjang masa ( sehingga pengambil keputusan harus reaktif ) dan fleksibel.
8. Mudah digunakan.
9. Mengutamakan efektifitas daripada efisiensi
10. Pengambil keputusan memiliki kontrol menyeluruh terhadap semua langkah proses pengambilan keputusan dalam menyelesaikan masalah. DSS secara khusus ditujukan untuk mendukung dan tak menggantikan pengambil keputusan.
11. DSS mengarah pada pembelajaran, yaitu mengarah pada kebutuhan baru dan penyempurnaan sistem
12. Pengguna harus mampu menyusun sendiri sistem yang sederhana.
13. DSS biasanya mendayagunakan pelbagai model ( standar atau sesuai keinginan pengguna ) dalam menganalisis berbagai keputusan.
14. DSS dalam pengembangannya dilengkapi dengan komponen *knowledge* yang bisa memberikan solusi yang efisien dan efektif.



### 3.4 DFD (*Data Flow Diagram*)

Menurut *Jogianto (1999:700)* DFD (*Data Flow Diagram*) adalah diagram yang digunakan untuk menggambarkan sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau data tersebut disimpan.

Keuntungan menggunakan DFD (*Data Flow Diagram*) adalah memudahkan pemakai yang kurang menguasai bidang komputer untuk mengerti sistem yang akan dikerjakan atau dikembangkan. Simbol-simbol menggunakan simbol dari Gane and Sarson dapat dilihat pada gambar

Beberapa hal yang perlu mendapat perhatian lebih tentang *Data Flow Diagram* (DFD) adalah sebagai berikut :

1. Antara sumber data tidak boleh langsung saling berhubungan.
2. Diperbolehkan untuk mengambil sumber data yang sama, dengan tujuan untuk menyederhanakan permodelan.
3. Hindari dialog-dialog yang tidak perlu dalam *Data Flow Diagram* (DFD).
4. Untuk memudahkan membaca DFD, maka penggambaran DFD disusun berdasarkan tingkatan atau level dari atas ke bawah, yaitu :

#### a. Diagram Konteks

Merupakan diagram paling atas yang terdiri dari suatu proses dan menggambarkan ruang lingkup proses. Hal yang digambarkan dalam diagram konteks adalah hubungan terminator dengan sistem dan juga sistem dalam proses. Sedangkan hal yang tidak

digambarkan dalam diagram konteks adalah hubungan antar *terminator* dan *data store*.

b. Diagram *Zero* (Level 0)

Merupakan diagram yang berada diantara Diagram Konteks dan Diagram Detail serta menggambarkan proses utama dari DFD. Hal yang digambarkan dalam Diagram *Zero* adalah proses utama dari sistem serta hubungan *Entity*, Proses, alur data, dan *data store*.

c. Diagram Detail

d. Merupakan penguraian dalam proses yang ada dalam Diagram *Zero*. Diagram yang paling rendah dan tidak dapat diuraikan lagi.

Data Flow Diagram (DFD) memiliki empat komponen, yaitu :

a) Terminator atau *External Entity* atau Kesatuan Luar

Terminator mewakili entitas *external* yang berkomunikasi dengan sistem yang sedang dikembangkan. Terminator merupakan kesatuan di lingkungan sistem yang dapat berupa orang, organisasi atau sistem lainnya yang berada di lingkungan luar sistem yang akan memberikan input maupun output dari sistem. Biasanya terminator ini dikenal dengan nama entitas (*external*), sumber atau tujuan (*source* and *sink*). Terminator dapat juga berupa departemen, divisi, atau sistem diluar sistem yang berkomunikasi dengan sistem yang dikembangkan. Ada tiga hal penting yang harus diingat tentang terminator :

1. Terminator merupakan bagian atau lingkungan luar sistem.  
Alur data yang menghubungkan terminator dengan berbagai proses sistem menunjukkan hubungan sistem dengan dunia luar.
2. Profesional sistem tidak dapat mengubah isi atau cara kerja, organisasi atau prosedur yang berkaitan dengan terminator.
3. Hubungan yang ada antar terminator yang satu dengan yang lain tidak dapat digambarkan pada DFD.

b) Proses

Proses sering dikenal dengan nama *Bubble*, fungsi atau informasi. Komponen proses menggambarkan bagian dari sistem. Yang mentransformasikan input ke output, atau dapat dikatakan bahwa komponen proses menggambarkan transformasi satu *input* atau lebih menjadi *output*. Dilambangkan dengan lingkaran, atau empat persegi panjang tegak dengan sudut tumpul.

c) Data Store (Penyimpan Data)

Data store digunakan sebagai saran untuk pengumpulan data. Data store disimbolkan dengan dua garis *horizontal* yang parallel dimana tertutup pada salah satu ujungnya atau dua garis *horizontal*. Suatu nama perlu diberikan pada data store menunjukkan nama dari filenya.

Data store ini biasanya berkaitan dengan penyimpanan seperti : *file* atau *database* yang berkaitan dengan penyimpanan secara

komputerisasi. Data *store* juga berkaitan dengan penyimpanan data.

### 3.5 *Entity Relationship Diagram (ERD)*

Relasi Entiti adalah suatu alat untuk mempresentasikan model data yang ada pada sistem dimana terdapat *Entity* dan *Relationship*. *Entity* merupakan objek yang ada dan terdefiniskan di dalam suatu organisasi dapat abstrak atau nyata, misal dapat berupa orang, objek atau waktu kejadian. Setiap *entity* mempunyai atribut atau karakteristik *entity* tersebut. Adapun elemen-elemen dari ERD ini adalah :

1. Entitas adalah sesuatu yang dapat diidentifikasi di dalam lingkup pemakai, sesuatu yang penting bagi pemakai dari sistem yang akan dikembangkan.
2. Atribut, entitas memiliki atribut yang berfungsi untuk menjelaskan karakteristik dari entitas.
3. Pengidentifikasian, data-data entitas memiliki nama yang berfungsi untuk mengidentifikasikan mereka. Sebuah identifikasi dapat bersifat unik atau tidak unik.
4. Hubungan atau relasi berfungsi untuk menunjukkan hubungan satu entitas dengan entitas lain. Hubungan ini boleh memiliki atribut. Banyaknya entitas dalam suatu relasi menunjukkan tingkat dari relasi yang bersangkutan, namun yang banyak digunakan dalam aplikasi-aplikasi adalah model yang menggunakan relasi tingkat dua atau yang disebut dengan hubungan biner. Hubungan biner ini memiliki tiga tipe

yaitu hubungan biner satu ke satu, biner satu ke banyak dan hubungan biner banyak ke banyak.

Sedangkan *relationship* adalah hubungan yang mewujudkan pemetaan antar *entity*. Fungsi untuk hubungan yang mewujudkan pemetaan antar *entity*.

Jenis *Relationship* diagram dapat berbentuk :

a. *One to One*

Yaitu relasi satu lawan satu yang terjadi bila satu record yang ada dalam satu *entity/table* hanya punya satu relasi pada *file* lain. Misalnya suatu departemen hanya mengerjakan satu jenis pekerjaan saja dan satu pekerjaan hanya dikerjakan oleh satu departemen saja.

b. *One to Many*

Yaitu relasi satu lawan banyak yang terjadi bila *record* dengan kunci tertentu pada satu *file* mempunyai relasi banyak *record* pada file lain. Misalnya suatu pekerjaan hanya dikerjakan oleh satu departemen saja, namun suatu departemen dapat mengerjakan beberapa macam pekerjaan sekaligus.

c. *Many to Many*

Yaitu relasi banyak lawan banyak yang terjadi bila kedua *file* saling mempunyai relasi banyak *record* pada *file* yang lain. Misalnya satu departemen mampu mengerjakan banyak pekerjaan, juga satu pekerjaan dapat ditangani oleh banyak departemen.