Desain Metode *Polar Fuzzy Vault* untuk Proteksi Data Sidik Jari

Yohanes I. Riskajaya[1], Tohari Ahmad[2]

***Abstract*— Authentication concept using password or token has been used widely but still has weaknesses, because password or token can be easily forgotten or shared with other users. The use of biometric data as an authentication tool can be a solution to the problem. It is because biometric data is relatively permanent and unique, thus, the risk of losing or forgetting the password is reduced. Consequently, a protection method to secure the biometric data is needed. A number of studies have been done to secure the biometric data, ranging from classic encryption method to data transformation.**

In this paper, a method is developed to create a cancelable fingerprint templates by combining pair-polar on the selection and matching with modified fuzzy vault on data transformation. The selection procecss is carried out in two stages. The first stage selects the reference minutiae points and its respective neighbors which close to the reference. A feature vector set is adopted from pair-polar method with an addition of the ridge type of the neighbors for later use in the matching process. The second stage generates a template, which is developed by implementing extended fuzzy vault with combining four regular fuzzy vault sets into one vault. This method gains the most optimum accuracy up to 99% of GAR and 0.03% of FAR. The use of ridge type in the matching process can reduce FAR from 1.03% to 0.03%, and the use of extended fuzzy vault can reduce FAR from 0.28% to 0.03%.

***Intisari*— Konsep autentikasi dengan menggunakan kata sandi atau token sudah lama digunakan, tetapi masih memiliki kelemahan, yaitu dapat hilang atau terlupakan. Penggunaan data biometrik sebagai sarana autentikasi dapat menjadi solusi karena data biometrik bersifat relatif permanen dan unik. Namun, sebagai konsekuensi, diperlukan metode untuk melindungi data biometrik tersebut. Sejumlah penelitian telah dilakukan untuk mengamankan data biometrik, mulai dari menerapkan enkripsi hingga melakukan transformasi.**

Pada makalah ini, dikembangkan metode untuk membuat *cancelable template* pada sidik jari dengan mengadaptasi metode seleksi *pair-polar* dan transformasi data *fuzzy*. Proses seleksi dilakukan dalam dua tahap. Pertama, seleksi awal pemilihan titik *minutiae* referensi dilanjutkan pemilihan titik ketetanggaan terhadap titik referensi. Vektor fitur yang dibuat dengan *pair-polar* ditambahkan dengan fitur tipe *ridge* titik tetangga untuk kemudian digunakan dalam proses pencocokan. Pada akhirnya, pembuatan *template* dikembangkan dengan metode *extended fuzzy vault* dengan menggabungkan empat himpunan *fuzzy vault* standar menjadi satu *vault*. Pengembangan metode ini menghasilkan akurasi paling optimal dengan GAR 99% dan FAR 0,03%. Penggunaan tipe *ridge* dalam proses pencocokan mampu mengurangi persentase FAR dari 1,03% menjadi 0,03%, dan penerapan *extended fuzzy vault* juga mampu menekan FAR dari 0,28% menjadi 0,03%.

***Kata Kunci*— *Fuzzy vault*, keamanan data, keamanan informasi, sidik jari, proteksi data.**

I. PENDAHULUAN

Cara autentikasi klasik dengan menggunakan kata sandi atau token masih banyak digunakan. Namun, konsep autentikasi tersebut masih menyimpan beberapa permasalahan, salah satunya adalah pengguna dapat kehilangan atau lupa dengan kata sandinya. Permasalahan lainnya berkaitan dengan mudahnya kata sandi atau token dipindahtangankan kepada pengguna lain, sehingga sistem tidak dapat menjamin keabsahan pengguna yang terautentikasi.

Untuk mengatasi masalah tersebut, data biometrik menjadi alternatif yang dapat digunakan untuk proses autentikasi. Sifat data biometrik yang unik dan relatif permanen serta melekat pada diri manusia (pengguna) menjadi alasan yang kuat untuk menggantikan sistem autentikasi menggunakan kata sandi. Sejumlah penelitian telah dilakukan untuk merancang sebuah proses autentikasi yang aman dan akurat menggunakan data biometrik, khususnya data sidik jari.

Data biometrik sendiri bersifat privat, sehinga dalam menyimpan data biometrik pada basis data, khususnya dalam hal ini adalah data sidik jari, seharusnya tidak boleh berupa data asli. Beberapa cara dilakukan untuk mengamankan data sidik jari ini, salah satunya dengan cara pengamanan umum menggunakan teknik enkripsi. Namun, teknik enkripsi ini masih dinilai sangat rentan terhadap masalah keamanan, karena pada waktu proses pencocokan saat autentikasi berlangsung, data sidik jari pada basis data akan terlebih dahulu didekripsi sehingga kembali pada bentuk aslinya.

Ada beberapa metode yang dikerjakan untuk menghasilkan *template* data sidik jari yang dapat sewaktu-waktu dihapus dan dibuat kembali untuk data biometrik yang sama, dengan bentuk yang *template* berbeda, sehingga dapat disebut *cancelable template*. Sebuah penelitian mengkategorikannya dalam empat cara yaitu sebagai berikut [1].

1. *Biometric Salting*, yaitu sebuah teknik yang menambahkan "*salt*" (berupa informasi acak khusus terhadap pengguna tertentu) untuk meningkatkan entropi dari *template* biometrik.
2. *Biometric Key Generation*, yaitu sebuah teknik untuk menurunkan sebuah kunci dari sinyal biometrik [2].
3. Skema *Fuzzy*, yaitu teknik yang menggabungkan *data helper* yang bersifat publik dengan data biometrik untuk membuat sebuah *template*. Ada beberapa macam metode *fuzzy*, yaitu *fuzzy commitment*, *fuzzy extractor* [3], dan *fuzzy vault* [4], [5].

[1] *Dosen, Departemen Informatika, Universitas Internasional Semen Indonesia* (*UISI*), *Gresik, INDONESIA, (e-mail: yohanes.riskajaya@uisi.ac.id)*

[2] *Dosen, Jurusan Teknik Informatika, Institut Teknologi Sepuluh Nopember* (*ITS*), *Surabaya, INDONESIA (telp: 031-5939214; e-mail: tohari@if.its.ac.id)*

4. Transformasi *non-invertible*, merupakan sebuah teknik untuk mengubah (transformasi) data biometrik dengan fungsi satu arah [6].

Dari keempat kategori tersebut, skema *fuzzy* dan transformasi *non-invertible* masih sangat menarik untuk diteliti. Transformasi *non-invertible* sendiri dapat berupa transformasi Cartesian, transformasi polar, dan transformasi fungsional [1]. Penggunaan skema *fuzzy* tampak menjanjikan untuk menghasilkan akurasi pencocokan yang lebih tinggi. Selain itu, skema *fuzzy* juga memberi nilai tambah tersendiri, karena juga berfungsi menyembunyikan suatu data rahasia yang dapat berupa kunci suatu algoritme enkripsi simetris.

Pada makalah ini, fitur tipe *ridge* digunakan sebagai salah satu kriteria pembanding pada saat pencocokan dan diterapkannya konsep *extended fuzzy vault*. Sebuah *vault* terdiri atas himpunan titik-titik yang diperoleh dari gabungan beberapa fungsi polinomial.

Struktur makalah ini selanjutnya disusun sebagai berikut. Bagian II memaparkan penelitian yang berhubungan dengan metode yang diusulkan. Bagian III menjelaskan metode yang diusulkan, sedangkan hasil uji coba serta analisisnya didiskusikan pada bagian IV. Terakhir, yakni kesimpulan, ada pada bagian V.

II. *FUZZY VAULT* UNTUK SIDIK JARI

Beberapa penelitian sebelumnya yang berkaitan dengan penelitan ini dapat disampaikan sebagai berikut.

A. Fuzzy Vault

Teknik pengamanan data sidik jari dengan *fuzzy vault* pernah diimplementasikan pada aplikasi *smartcard* [7], kemudian disempurnakan lagi dengan memanfaatkan titik *minutiae* dari sidik jari untuk membuat *vault* dan menyembunyikan kunci AES 128 bit di dalamnya [4]. Dalam membuat *template*, titik *minutiae* terlebih dahulu disejajarkan dengan menggunakan acuan orientasi dari *singular point* (*core point*). Penggunaan *core point* sebagai acuan orientasi banyak digunakan dalam berbagai penelitian, termasuk di antaranya transformasi dengan menggunakan garis proyeksi [8] dan transformasi geometrik [9]. Saat ini, penggunaan *core point* sebagai acuan orientasi semakin dikurangi karena keberadaannya kurang stabil, dapat diakibatkan karena pada saat pengambilan data sampel tidak optimal atau pengguna tidak memiliki *core point*.

Dengan tidak digunakannya *core point*, maka proses autentikasi hanya mengandalkan informasi dari titik *minutiae*, tetapi bukan hanya satu (titik *minutiae* referensi), melainkan keseluruhan titik *minutiae*. Keuntungannya adalah tidak perlu dilakukan rotasi untuk penjajaran, sehingga lebih fleksibel. Kekurangannya ada pada tingkat kompleksitas komputasi yang pasti akan lebih rumit dibandingkan dengan penggunaan satu titik *minutiae* referensi yaitu *core point*. Penelitian lain menggunakan *helper data* yang diekstrak dari data sidik jari dan digunakan untuk menyejajarkan menggunakan algoritme *Iterative Close Point* (ICP) [10]. Proses persejajaran juga dapat dihilangkan dengan menghitung keterkaitan hubungan antara satu titik *minutiae* dengan yang lainnya, lalu diamankan dengan *fuzzy vault* [11].

B. Koordinat Polar

Koordinat polar merupakan bentuk koordinat yang dibentuk dari vektor $(r, \theta$, dengan nilai r merupakan jarak radial dari titik asal (0, 0) pada koordinat Cartesian menuju titik (x, y) dan θ merupakan koordinat *angular* yang diukur dari sudut sumbu x terhadap r berlawanan arah jarum jam. Sejumlah penelitian menerapkan transformasi data sidik jari pada domain koordinat polar [12], [13] dan juga gabungan pada koordinat Cartesian dan polar [6]. Fitur komposit yang pernah diusulkan juga diperoleh dengan menggunakan konsep koordinat polar [11]. Sedangkan pada penelitian lain, diterapkan perolehan fitur murni dari koordinat polar yang kemudian direpresentasikan dalam bentuk *bit-string* [14].

Penentuan vektor fitur dalam koordinat polar dilakukan dengan terlebih dahulu menentukan titik *minutiae* referensi sebagai pusat koordinat [3], [6], [11] - [13]. Orientasi titik referensi tersebut dijadikan acuan sumbu x positif, sehingga seluruh titik *minutiae* lainnya yang ada di sekitar titik referensi dapat diposisikan secara relatif berdasarkan koordinat titik referensi. Pada salah satu penelitian diperkenalkan sebuah kerangka kerja yang menggambarkan keterkaitan dua arah antara titik pusat (referensi) terhadap titik tetangga yang masing-masing memiliki sudut orientasi, sehingga diperoleh vektor koordinat polar yang terdiri atas (r, α, β) dengan jarak r dan sudut α merupakan elemen dari koordinat polar yang dihitung dari titik pusat ke titik tetangga, sedangkan sudut β merupakan elemen sudut koordinat polar yang dihitung dari titik tetangga ke titik pusat [12]. Oleh karena itu, kerangka kerja tersebut dinamakan sebagai *pair-polar* karena memiliki keterkaitan antar titik secara dua arah (berpasangan).

C. Extended Fuzzy Vault

Vault yang dihasilkan dari algoritme standar *fuzzy vault* pada dasarnya merupakan sebuah himpunan titik-titik koordinat (x, y) dengan y merupakan nilai hasil evaluasi suatu fungsi polinomial P terhadap nilai x, sehingga $y = P(x)$. Pada himpunan titik-titik yang berkorelasi dengan fungsi polinomial P tersebut kemudian ditambahkan titik-titik palsu atau yang sering disebut sebagai *chaff point* yang merupakan titik (x, y) dengan nilai $y \neq P(x)$.

Pada *extended fuzzy vault*, proses pembuatan *vault* sama dengan *fuzzy vault* biasa, hanya saja beberapa *vault* yang terbentuk dari sejumlah fungsi polinomial yang berbeda $(P1, P2, \ldots, Pn)$ digabungkan menjadi satu untuk membentuk satu *vault* baru beranggotakan himpunan titik-titik yang dibentuk dari *n* fungsi polinomial yang berbeda [15]. Keuntungan dari konsep ini adalah dapat mereduksi ukuran penyimpanan data sekaligus mengurangi usaha untuk membuat titik *chaff*, karena titik-titik yang diperoleh dari satu fungsi polinomial akan dianggap sebagai titik *chaff* bagi fungsi polinomial lainnya.

III. METODE *POLAR FUZZY VAULT*

Dari beberapa referensi terdahulu, banyak penelitian berfokus pada pengembangan metode pembuatan *template* sidik jari dengan memanfaatkan fitur lokal, yaitu titik

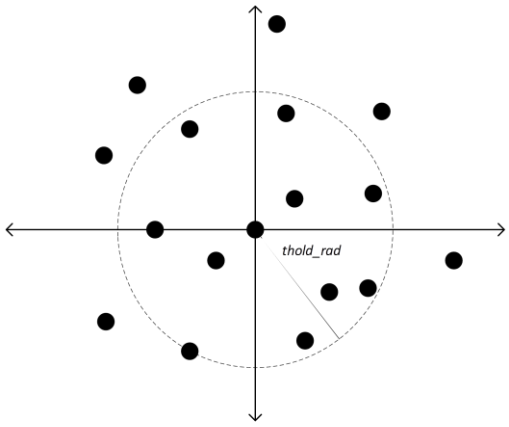
minutiae. Beberapa penelitian menerapkan konsep yang mirip dalam melakukan transformasi fitur untuk memperoleh suatu himpunan vektor ketetanggaan yang terdiri atas fitur koordinat relatif dalam bentuk koordinat polar [11], [12]. Namun, fitur komposit digunakan selanjutnya untuk pembentukan *fuzzy vault* yang menjadi fungsi transformasi untuk mengamankan data sidik jari sekaligus menyimpan sebuah data rahasia [11]. Pada penelitian sebelumnya, diperkenalkan fitur *pair-polar* yang kemudian diacak kembali menggunakan fungsi transformasi searah dalam koordinat polar, yang fungsi-fungsi transformasi tersebut akan menjadi kunci untuk membangun *template* dan *query* [12].

Makalah ini menggabungkan kelebihan dari keterkaitan yang dibentuk fitur *pair-polar* [12] dengan peluang kecocokan dengan akurasi tinggi yang diperoleh dari konsep *fuzzy vault* [11], dengan menambahkan konsep *extended fuzzy vault* [15] untuk menambah efektivitas penyimpanan data *template*. Selain itu, fitur tipe *ridge* titik *minutiae* tetangga juga ditambahkan dalam vektor *pair-polar* untuk selanjutnya digunakan dalam proses pencocokan. Hal ini diperlukan karena tidak dimungkinkan menyimpan informasi tipe *ridge* untuk masing-masing titik *minutiae* referensi seperti pada penelitian sebelumnya karena sesuai konsep *extended fuzzy vault* [11], [12]. Penggunaan tipe *ridge* juga bertujuan untuk lebih memperketat batasan kemiripan antara dua vektor, sehingga diharapkan dapat menekan kesalahan dalam identifikasi sidik jari. Untuk penjelasan lebih lanjut mengenai metode yang diusulkan, keseluruhan proses dibagi menjadi tiga bagian yaitu [12]:

1. seleksi titik *minutiae*,
2. pembuatan *template* sidik jari, dan
3. pencocokan.

A. Seleksi Titik Minutiae

Langkah pertama dalam membuat sebuah *template* data sidik jari yang akan disimpan dalam basis data adalah dengan melakukan seleksi fitur. Tujuan dari seleksi fitur ini sendiri adalah untuk membatasi jumlah fitur titik *minutiae* yang akan diproses selanjutnya. Pembatasan ini selain berguna untuk mengurangi beban proses, juga untuk meningkatkan peluang akurasi pada saat proses pencocokan nantinya. Dengan merujuk pada penelitian sebelumnya, dalam makalah ini proses seleksi akan dibagi menjadi dua tahap, yaitu memilih titik-titik *minutiae* yang kemudian akan dijadikan sebagai titik referensi berdasarkan metode seleksi [12], lalu memilih titik-titik tetangga dari masing-masing titik referensi [16]. Langkah-langkah tersebut dapat dijelaskan sebagai berikut.



Gbr. 1 Radius area ketetanggaan terhadap titik referensi.

1) Seleksi Titik Referensi: Seleksi titik *minutiae* untuk dijadikan titik referensi telah dipaparkan pada penelitian sebelumnya dengan konsep memilih titik-titik yang ada pada *data set* awal yang memenuhi jarak minimal tertentu satu sama lain [12]. Perhitungan jarak antar titik didefinisikan sebagai berikut.

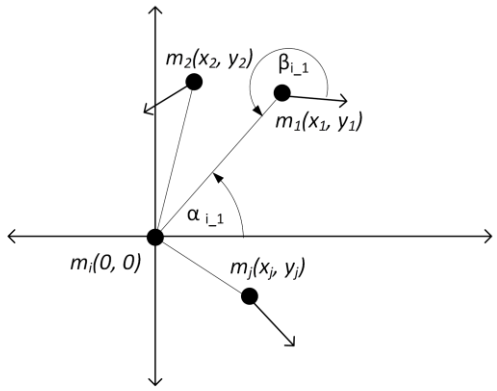
$$dis(m_i, m_j) = t_1 \Delta r + t_2 \Delta \alpha. \quad (1)$$

Dalam hal ini, (1) menghitung jarak antara dua titik *minutiae*, dengan Δr merupakan jarak koordinat antara dua titik yang dapat diperoleh dari $\Delta r = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$, sedangkan $\Delta \alpha$ merupakan selisih orientasi antara dua titik yang dapat diperoleh dari perhitungan $\Delta \alpha = \min\left(|\theta_i - \theta_j|, (360 - |\theta_i - \theta_j|)\right)$, dengan t_1 dan t_2 adalah konstanta pembobotan untuk Δr dan $\Delta \alpha$ yang berturut-turut bernilai 1 dan 0,2 [12].

Jarak minimal yang memenuhi syarat dalam proses seleksi ditentukan dengan konstanta $thold\_dis_1$, sehingga untuk setiap titik-titik *minutiae* yang terpilih sebagai titik referensi memenuhi $dis(m_i, m_j) \leq thold\_dis_1$ [12]. Namun, berbeda dengan penelitian terdahulu, titik-titik yang tidak terpilih sebagai titik referensi selanjutnya tidak diabaikan, tetapi akan diikutsertakan sebagai anggota dari titik-titik tetangga.

2) Seleksi Titik Ketetanggaan: Setelah diperoleh titik-titik referensi dari proses seleksi tahap pertama, tahap seleksi selanjutnya adalah memilih titik-titik tetangga untuk masing-masing titik referensi yang dipilih. Pada seleksi tahap kedua, untuk masing-masing titik referensi akan dibatasi area yang menjadi anggota titik-titik tetangga [16]. Pembatasan area tersebut dilakukan dengan menggunakan batasan jarak radius tertentu dari titik *minutiae* referensi yang dinotasikan dengan $thold\_rad$ seperti ditunjukkan pada Gbr. 1. Dengan nilai $thold\_rad$ yang tetap untuk semua titik *minutiae* referensi, maka diperoleh sejumlah titik-titik *minutiae* tetangga yang berada di dalam radius tersebut.

Untuk membuka sebuah *fuzzy vault* dan memperoleh data rahasia yang disembunyikan di dalamnya, dibutuhkan himpunan titik sejumlah $d + 1$, dengan d adalah derajat atau pangkat tertinggi dari fungsi polinomial pembentuk *fuzzy vault* tersebut [11]. Pada makalah ini akan dibentuk *fuzzy vault* dengan menggunakan fungsi polinomial berderajat 9, sehingga untuk menjaga adanya peluang keberhasilan dalam membuka *fuzzy vault*, jumlah titik *minutiae* tetangga yang tercakup dalam radius harus diperoleh minimal 9 + 1 = 10 titik. Jika dalam radius tersebut terdapat titik *minutiae* tetangga di bawah 10 titik, maka titik *minutiae* referensi tersebut dibatalkan dari anggota titik referensi. Pada makalah ini digunakan nilai $thold\_rad$ = 150, yang merupakan batas terkecil, yaitu seluruh sidik jari yang ada dapat diproses untuk pembuatan *template* [16].



Gbr. 2 Cara memperoleh vektor ketetanggaan dari suatu titik *minutiae* referensi.

B. Pembuatan Template Sidik Jari

1) Perubahan Bentuk Fitur dalam Koordinat Polar: Setelah melalui proses seleksi tahap pertama, semua titik *minutiae* yang terpilih menjadi titik referensi selanjutnya akan berperan sebagai titik pusat dengan titik-titik *minutiae* tetangga yang ditentukan pada proses seleksi titik ketetanggaan. Pada langkah selanjutnya, berdasarkan titik referensi sebagai pusat dengan titik-titik tetangga yang berada di radius titik referensi tersebut, dibentuklah vektor yang menghubungkan titik pusat (referensi) dengan titik-titik tetangga di sekitarnya. Dengan mengacu pada konsep *pair-polar*, didefinisikan titik-titik *minutiae* dari suatu jari i adalah m_i dan himpunan titik-titik referensi disebut BS_i [12]. Dari masing-masing $m_i \in BS_i$, diperoleh sebuah vektor fitur (r, α, β) yang diperoleh dari posisi relatif titik-titik tetangga terhadap titik referensi yang disajikan dalam bentuk koordinat polar.

Gbr 2 menunjukkan perolehan nilai (r, α, β) seperti yang telah dijelaskan. Posisi koordinat dibuat secara relatif terhadap titik *minutiae* referensi, dengan sumbu x-positif merupakan orientasi dari titik *minutiae* referensi yang menjadi titik pusat (0, 0) [12]. Nilai r dan α sendiri merupakan elemen standar konversi titik dalam koordinat Cartesian menjadi koordinat polar, sedangkan nilai β merupakan selisih sudut yang memanfaatkan fitur orientasi dari titik tetangga, sehingga ada keterkaitan lebih antara titik pusat (referensi) dengan titik tetangga.

Berbeda dengan penelitian sebelumnya yang hanya menggunakan fitur (r, α, β), dalam makalah ini ditambahkan fitur t yang merupakan tipe *ridge* titik *minutiae* tetangga, sehingga masing-masing titik referensi memiliki himpunan vektor $v_{i\_j} = \{r_{i\_j}, \alpha_{i\_j}, \beta_{i\_j}, t_j\}$ [12]. Fitur t ini nantinya akan digunakan dalam proses pencocokan dengan data *query*.

Seperti yang sudah dijelaskan sebelumnya, titik-titik *minutiae* tetangga ditentukan dengan batasan area tetangga yang dievaluasi berdasarkan radius tertentu sehingga memenuhi $r_{i\_j} \leq thold\_rad$ seperti diilustrasikan pada Gbr. 1 sebelumnya. Dengan demikian, diperoleh himpunan vektor fitur v_{i_j} untuk masing-masing titik referensi $m_i \in BS_i$ sejumlah k titik, yang selanjutnya akan disebut Vms_{i_k} yang dinyatakan dalam (2). Masing-masing anggota himpunan Vms_{i_k} dapat memiliki jumlah vektor yang berbeda tergantung pada perolehan titik tetangga (j titik) yang tercakup dalam radius tersebut.

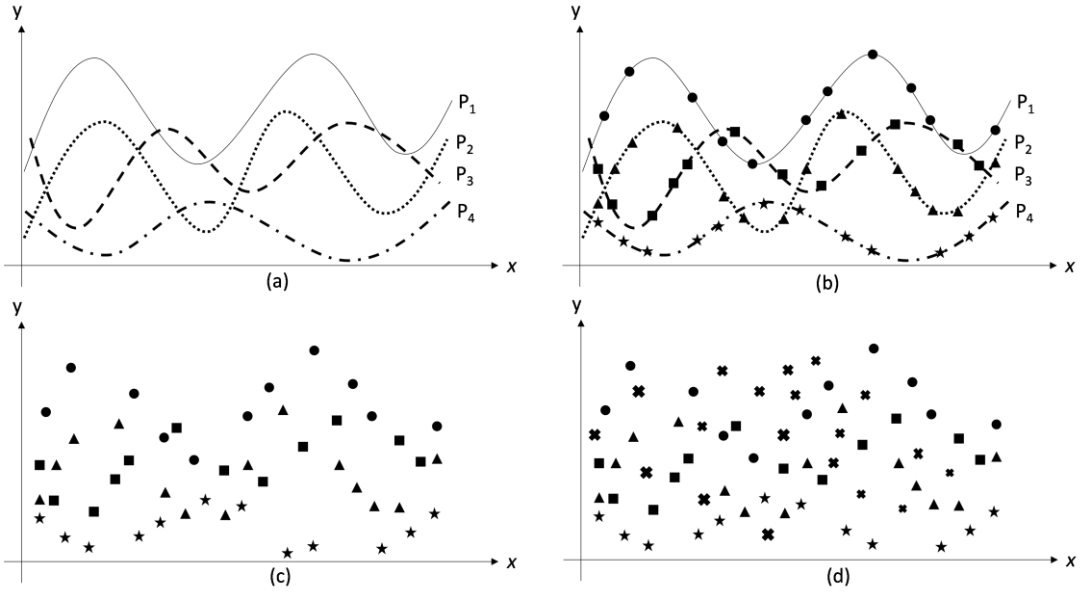
$$Vms_{i_k} = \begin{Bmatrix} (r_{i_{k\_1}}, \alpha_{i_{k\_1}}, \beta_{i_{k\_1}}, t_1), \\ (r_{i_{k\_2}}, \alpha_{i_{k\_2}}, \beta_{i_{k\_2}}, t_2), \\ (r_{i_{k\_3}}, \alpha_{i_{k\_3}}, \beta_{i_{k\_3}}, t_3), \\ \vdots \\ (r_{i_{k\_j}}, \alpha_{i_{k\_j}}, \beta_{i_{k\_j}}, t_j) \end{Bmatrix}. \quad (2)$$

2) Penguncian dengan Extended Fuzzy Vault: Setelah mendapatkan himpunan vektor Vms_{i_k}, proses transformasi selanjutnya dalam membuat *template* adalah dengan menggunakan teknik *extended fuzzy vault* [15]. Konsep serupa yang menerapkan *fuzzy vault* pada sidik jari dipaparkan dalam penelitian sebelumnya yang memanfaatkan fitur komposit [11]. Pembentukan elemen vektor fitur komposit sendiri (d, φ, θ) tidak jauh berbeda dengan elemen vektor fitur pada *pair-polar*, hanya saja pada penurunan salah satu fitur sudut ada sedikit perbedaan [12].

Adapun proses enkripsi *fuzzy vault* pada makalah ini mengadaptasi penelitian sebelumnya dengan penyesuaian kombinasi fitur sebagai berikut [11].

a. Pada masing-masing himpunan, vektor ketetanggaan Vms_{i_k} dimasukkan ke dalam fungsi hash untuk menggabungkan fitur (r, α, β, t)untuk menghasilkan nilai x yang berukuran 16-bit, yang nantinya akan dimasukkan dalam fungsi polinomial $P(x)$. Fungsi hash dalam makalah ini menggunakan algoritme Pearson Hash [17]. Sebelum melalui proses hash, keempat elemen vektor dibaca sebagai teks (*string*) lalu digabungkan, sehingga membentuk satu rangkaian teks angka. Contohnya adalah sebagai berikut.
   Data masukan:
   r = 120.12; α = 92.46; β = 111.59; t = 1.
   Data keluaran:
   x = *hash*('120.1292.46111.591')
   x = 43591 (16-bit integer).
b. Proses pembentukan fungsi polinomial diawali dengan sistem membangkitkan pesan rahasia S dengan panjang 144 bit secara acak setara banyaknya jumlah titik *minutiae* referensi dalam himpunan BS_i.
c. Masing-masing pesan S ditambahkan pengkodean CRC 16 bit sehingga menjadi SC yang memiliki panjang 160 bit. Data SC tersebut kemudian dibagi rata menjadi 10, yaitu $(c_{10}c_9c_8c_7c_6c_5c_4c_3c_2c_1)$, sehingga masing-masing terdiri atas 16 bit data yang akan digunakan sebagai koefisien fungsi polinomial orde 9. Fungsi polinomial yang dihasilkan akan berbentuk seperti (3) berikut.

$$P(x) = c_{10}x^9 + c_9x^8 + c_8x^7 + c_7x^6 + c_6x^5 + c_5x^4 + c_4x^3 + c_3x^2 + c_2x + c_1. \quad (3)$$



Gbr. 3 Ilustrasi *Extended Fuzzy Vault*, terinspirasi dari [15]. (a) Penggabungan beberapa fungsi polinomial, (b) *Plot* titik-titik pada fungsi polinomial, (c) *Vault* dengan titik-titik asli, (d) *Vault* dengan ditambahkan titik *chaff*.

Sebagai contoh, digunakan jumlah bit yang lebih kecil, yaitu SC yang memiliki panjang 10 bit:

$SC = 1001101011_2$,

kemudian dibagi menjadi lima bagian menjadi:

$c5 = 10_2, c4 = 01_2, c3 = 10_2,$

$c2 = 10_2, c1 = 11_2$, maka

$P(x) = 2x^4 + x^3 + 2x^2 + 2x + 3.$

Semua nilai x hasil penggabungan fitur dimasukkan ke dalam fungsi polinomial $P(x)$ untuk memperoleh titik p. Untuk menyederhanakan perhitungan, maka semua evaluasi polinomial dihitung dalam domain Galois *field* $GF(2^{16})$ sehingga hasil perhitungan semua merupakan bilangan integer 16-bit [4].

d. Himpunan Vms_{i_k} akan memiliki vektor baru yaitu $\{(r_{i_k\_j}, \alpha_{i_k\_j}, \beta_{i_k\_j}, t_j), p_{i_k}\}_{i=1}^{l}$. Jika diasumsikan sebuah koordinat, maka vektor tersebut merupakan titik (x, y) dengan $x = (r_{i_k\_j}, \alpha_{i_k\_j}, \beta_{i_k\_j}, t_j)$, dan $y = p_{i_k}$.

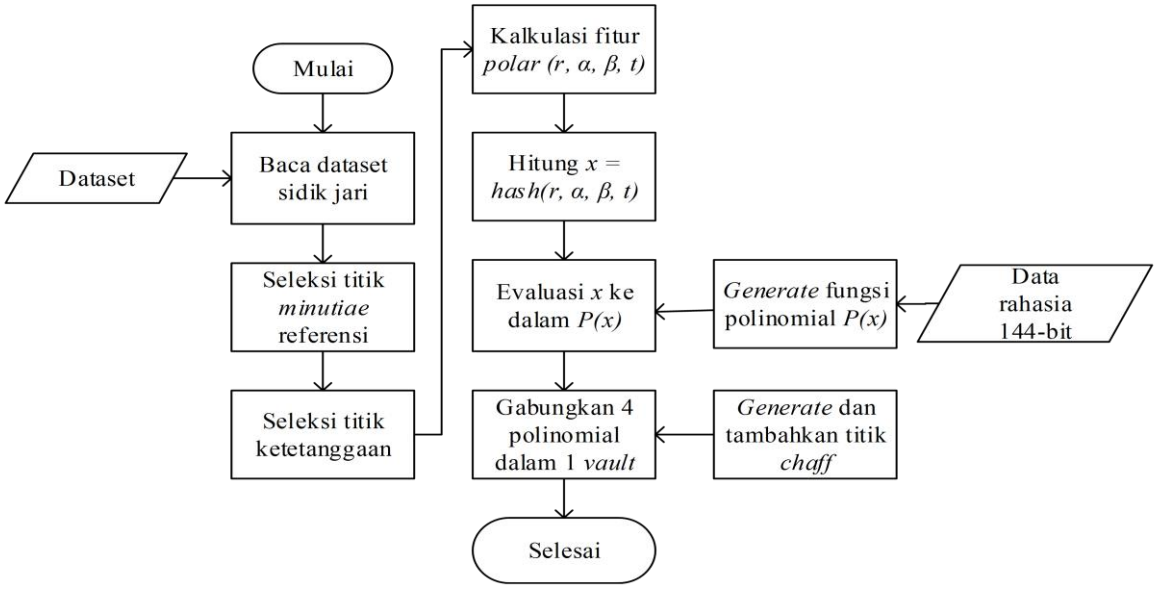
e. Selanjutnya adalah penerapan konsep *extended fuzzy vault* dengan sejumlah fungsi polinomial dari himpunan vektor fitur yang berbeda dimasukkan dalam satu wadah/*vault* [15]. Pada penelitian ini dimasukkan maksimal empat fungsi polinomial berbeda dalam satu *vault* dan minimal satu fungsi polinomial untuk polinomial sisanya jika total fungsi polinomial tidak habis dibagi 4. Maka jumlah *vault* dalam satu *template* sidik jari adalah $jml\_vault = \lceil l/4 \rceil$, dengan l = jumlah titik referensi atau jumlah fungsi polinomial. Gbr. 3(a) dan Gbr. 3(b) menunjukkan ilustrasi penggabungan polinomial dalam satu *vault*.

f. Titik *chaff* ditambahkan pada *vault* hingga terdapat titik gabungan sejumlah 200 titik. Titik c*haff* dibuat secara acak untuk membentuk vektor $\{(r_{chaff}, \alpha_{chaff}, \beta_{chaff}, t_{chaff}), p_{chaff}\}$, dengan $P\left(Hash(r_{chaff}, \alpha_{chaff}, \beta_{chaff}, t_{chaff})\right) \neq p_{chaff}$, sehingga titik tersebut tidak dapat menjadi anggota himpunan titik-titik pembentuk fungsi polinomial. Selain itu, untuk menjamin jarak antara $(r_{chaff}, \alpha_{chaff}, \beta_{chaff}, t_{chaff})$ dan $(r_{i_k\_j}, \alpha_{i_k\_j}, \beta_{i_k\_j}, t_j)$ pada masing-masing himpunan *vault* tidak terlalu berdekatan, baik jarak antar titik *chaff* itu sendiri maupun dengan titik asli yang dibentuk dari fungsi polinomial, sehingga tidak menyebabkan kesalahan identifikasi terhadap titik yang asli saat proses pencocokan dikarenakan tingkat kedekatannya lebih baik. Untuk nilai ambang ditentukan hanya di akhir menggunakan $thold\_dis_{chaff}$. Gbr. 3(c) dan Gbr. 3(d) menunjukkan ilustrasi pemberian titik *chaff* sekaligus *vault template* sidik jari yang disimpan dalam basis data.

Hasil *vault* yang diperoleh selanjutnya disimpan dalam basis data sebagai *template*. Gbr. 4 menunjukkan diagram alir pembuatan *template* sidik jari, mulai dari proses seleksi hingga terbentuknya *vault* dan Gbr. 5 menunjukkan bagan urutan transformasi data dalam pembuatan *template*. Jika suatu saat diperlukan, misalnya ketika *template* yang dibuat sebelumnya sudah bocor/tidak rahasia, maka pembuatan *template* ini dapat dilakukan ulang sehingga menghasilkan *template* yang berbeda walaupun dengan jari yang sama, karena data rahasia yang digunakan sebagai pembentuk fungsi polinomial serta titik *chaff* yang diproduksi untuk pembuatan *extended fuzzy vault* dapat berbeda dari pembuatan *template* sebelumnya.

C. Pencocokan dan Dekripsi Fuzzy Vault

Pada subbab ini dijelaskan proses pencocokan antara sidik jari *template* dengan sidik jari *query*. Keputusan akhir sidik jari *query* merupakan jari yang sama dengan *template* atau tidak sangat bergantung pada proses ini. Untuk menyiapkan data sidik jari pada *query*, beberapa proses yang sama



Gbr. 4 Pembuatan *template* sidik jari.

dilakukan seperti pada saat pembuatan *template*, yaitu proses seleksi fitur hingga transformasi ke dalam bentuk vektor *pair-polar* yang dalam makalah ini ditambahkan elemen tipe *ridge*, sehingga diperoleh vektor (r, α, β, t). Data *query* hanya diproses sampai tahap tersebut, sehingga menghasilkan himpunan vektor ketetanggaan Vms'_{i_k} sejumlah titik referensi yang ada dan memiliki anggota vektor ketetanggaan yang berjumlah minimal sepuluh. Dengan demikian, jumlah kelompok himpunan vektor pada *query* akan berbeda dengan jumlah himpunan vektor pada *vault template*, karena pada *vault template* sebelumnya telah terjadi proses penggabungan dalam penerapan *extended fuzzy vault*. Langkah selanjutnya adalah proses pencocokan untuk membuka data rahasia dalam *vault template*.

Proses pencocokan diawali dengan mencari titik-titik kandidat pada *vault template* dengan mengukur kedekatan vektor fitur sidik jari *query*. Sebuah titik pada *vault* pada dasarnya adalah vektor fitur yang sama dengan *query* yaitu (r, α, β, t), hanya terdapat tambahan elemen p_{i_k} dari proses pembuatan *extended fuzzy vault*. Langkah-langkah pencocokan sekaligus proses *vault decoding* adalah sebagai berikut.

1) Perbandingan Vektor: Pada setiap vektor fitur pada Vms_{i_k} dilakukan proses pencocokan pada semua anggota himpunan *vault* untuk memperoleh minimal sepuluh titik yang mirip sebagai kandidat. Misalnya dalam membandingkan vektor *template* $v_{i\_j} = (r_{i\_j}, \alpha_{i\_j}, \beta_{i\_j}, t_j)$ dan vektor *query* $v'_{i\_k} = (r'_{i\_k}, \alpha'_{i\_k}, \beta'_{i\_k}, t'_k)$. Maka, pertama kali dilakukan perbandingan antara tipe kedua vektor sama yang berarti $t_j = t'_k$. Jika berbeda, maka langsung dianggap bukan vektor yang sama. Tipe *ridge* dari tiap titik ketetanggaan ikut dibandingkan, berbeda dengan penelitian sebelumnya yang hanya membandingkan tipe *ridge* dari titik *minutiae* referensi antara data *template* dan data *query* [12]. Pembandingan tipe *ridge* titik referensi tidak dimungkinkan karena konsep *fuzzy vault* tidak menyimpan informasi yang dimiliki titik referensi.

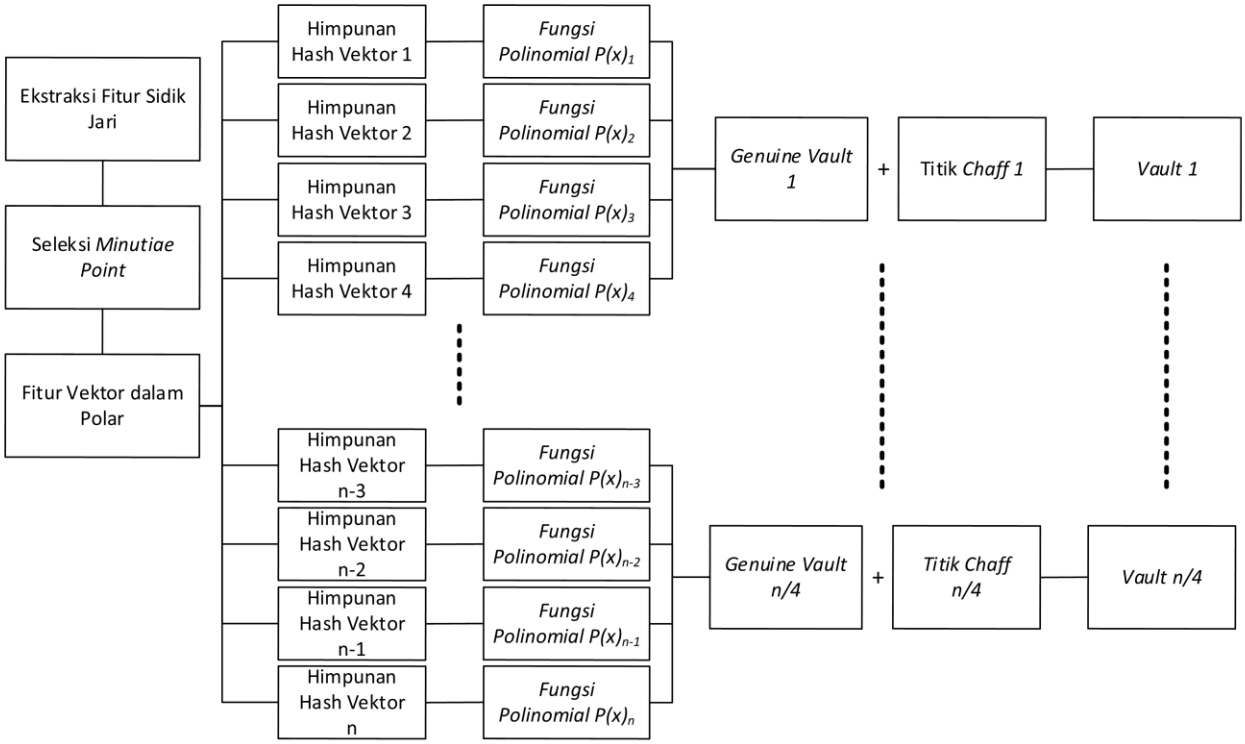
2) Hasil Perbandingan Vektor: Jika tipe *ridge* keduanya sama, maka dilakukan perhitungan kemiripan seperti halnya yang dilakukan penelitian sebelumnya [12]. Jika terdapat nilai yang tidak memenuhi batas, maka kedua vektor dinyatakan tidak mirip.

3) Pengecekan Nilai: Jika nilai pada langkah nomor 2 terpenuhi, maka langkah selanjutnya adalah menggabungkan ketiga nilai tersebut dengan (4) dengan nilai $wgh_r, wgh_\alpha, wgh_\beta$ adalah 1, 0.2, 0.2 [12]:

$$\Delta f = \Delta r \times wgh_r + \Delta\alpha \times wgh_\alpha + \Delta\beta \times wgh_\beta. \quad (4)$$

4) Urutan Titik Kandidat Vektor: Dari keseluruhan titik kandidat vektor yang diperoleh, selanjutnya titik-titik kandidat tersebut akan diurutkan berdasarkan perolehan nilai Δf mulai dari yang terkecil, yang berarti tingkat kedekatannya paling baik. Dari titik-titik kandidat yang telah diurutkan tersebut diambil maksimal $n_{toleransi}$ titik dengan nilai ≥ 10. Notasi $n_{toleransi}$ merupakan jumlah kandidat titik yang dipilih setelah diurutkan yang nilainya tidak harus tepat 10 untuk memberikan toleransi kesalahan terhadap sepuluh titik kandidat terbaik, sehingga peluang keberhasilan dalam membuka *vault* lebih tinggi. Dalam makalah ini diambil nilai $n_{toleransi}$ = 13 titik yang berarti ada toleransi kesalahan tiga titik, sehingga dapat dilakukan interpolasi polinomial dengan mengambil kombinasi sepuluh titik dari maksimal 13 titik yang ada hingga ditemukan salah satu kombinasi berhasil membuka data rahasia yang benar.

5) Perubahan Vektor: Setelah mendapatkan kombinasi kandidat yang diperoleh dari langkah sebelumnya, dilakukan proses perhitungan interpolasi dengan terlebih dahulu mengubah vektor $\{r, \alpha, \beta, t\}$ menjadi satu nilai x menggunakan fungsi hash. Selanjutnya, dengan pasangan nilai p dari vektor $\{(r_{i_{k\_j}}, \alpha_{i_{k\_j}}, \beta_{i_{k\_j}}, t_j), p_{i_k}\}_{i=1}^{l}$ atau bisa disebut sebagai y dalam koordinat Cartesian, dilakukan interpolasi Lagrange.



Gbr. 5 Proses pembuatan *template* sidik jari dengan *Extended Fuzzy Vault*.

Interpolasi membutuhkan $n + 1$ titik, dengan n adalah pangkat tertinggi dari fungsi polinomial yang ingin diperoleh. Karena dalam makalah ini digunakan fungsi polinomial dengan pangkat maksimal 9, maka hanya diperlukan sepuluh titik (x, y) dari seluruh kandidat yang ada.

Contoh perhitungan interpolasi Lagrange untuk fungsi polinomial derajat 3 adalah sebagai berikut, jika titik-titik masukan adalah:

$$x_1 = 1, y_1 = 12$$
$$x_2 = 3, y_2 = 82$$
$$x_3 = 5, y_3 = 264$$
$$x_4 = 7, y_4 = 606$$

$$P(x) = \frac{(x-3)(x-5)(x-7)}{(1-3)(1-5)(1-7)}12 + \frac{(x-1)(x-5)(x-7)}{(3-1)(3-5)(3-7)}82 + \frac{(x-1)(x-3)(x-7)}{(5-1)(5-3)(5-7)}264 + \frac{(x-1)(x-3)(x-5)}{(7-1)(7-3)(7-5)}606$$

Maka, diperoleh:

$$P(x) = \frac{8x^3+40x^2+16x+32}{8}. \quad (5)$$

Jadi, diperoleh fungsi polinomial $P(x) = x^3 + 5x^2 + 2x + 4$, sehingga dapat diambil koefisiennya, yaitu (1,5,2,4). Dalam makalah ini, proses perhitungan tetap dilakukan dalam *domain* Galois *Field* $GF(2^{16})$, sehingga seluruh data masukan dan keluaran merupakan bilangan bulat positif 16-bit.

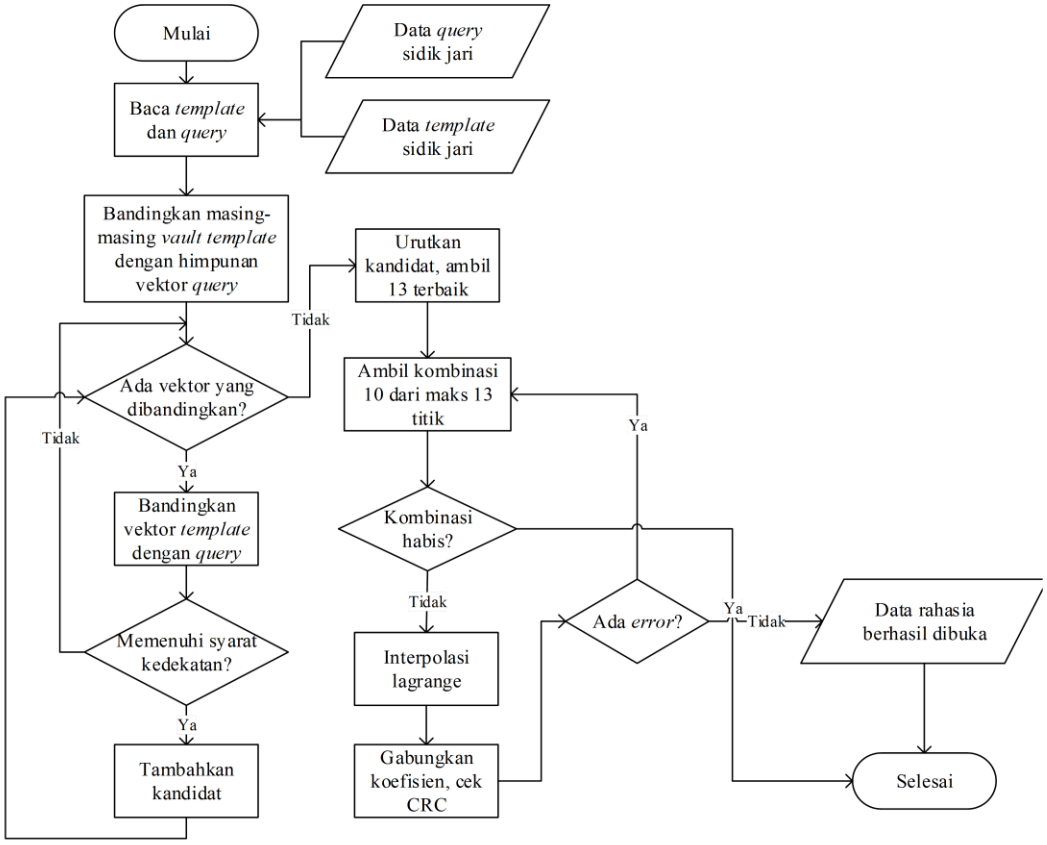
6) Pembentukan Fungsi Polinomial: Jika interpolasi berhasil dilakukan, maka akan diperoleh fungsi polinomial orde 9 dengan sepuluh koefisien. Kesepuluh koefisien tersebut kemudian diubah kembali ke dalam bentuk biner yang masing-masing akan berukuran 16 bit, lalu digabungkan untuk membentuk satu kesatuan pesan rahasia $SC' = c'_{10}c'_9c'_8c'_7c'_6c'_5c'_4c'_3c'_2c'_1$ dengan panjang 160 bit. Selanjutnya, dilakukan proses deteksi galat menggunakan CRC. Jika dalam proses deteksi tidak ada galat, maka pesan rahasia S' dapat diperoleh dari SC' dengan menghilangkan 16 bit paling belakang [11].

Ilustrasi penggabungan koefisien yang diperoleh dari contoh sebelumnya adalah sebagai berikut.

Konversi: $(1,5,2,4)_{10} = (001,101,010,100)_2$.

Penggabungan: $SC' = (001101010100)_2$.

Gbr. 6 menunjukkan diagram alir proses *vault decoding* seperti yang dijabarkan pada langkah 1 sampai 5. Proses tersebut dilakukan berulang sebanyak data yang ada pada *template* dan *query* hingga mendapatkan jumlah minimal S' yang benar, yang dinotasikan sebagai η, sehingga sistem dapat menyatakan sidik jari *query* tersebut cocok. Keempat proses tersebut dilakukan secara berulang hingga proses pencocokan telah dilakukan untuk semua data *vault template* dengan data *query*. Dalam iterasi pencocokan, jika satu himpunan vektor pada *query* telah berhasil membuka satu data rahasia, maka selanjutnya tidak perlu dilakukan pencocokan untuk data *vault template* yang lain. Begitu pula titik-titik kandidat yang termasuk dalam $n_{toleransi}$ titik terbaik akan diabaikan dalam pencocokan jika telah berhasil membuka satu data rahasia.



Gbr. 6 Proses pencocokan dan *vault decoding*.

IV. UJI COBA DAN ANALISIS

Data uji coba yang digunakan diambil dari *data set* FVC2002 DB2A yang terdiri atas seratus data dari jari yang berbeda dengan masing-masing terdiri atas delapan pengambilan citra (impresi) yang berbeda [18]. Data sidik jari dalam *data set* yang disebutkan di atas masih dalam bentuk citra *grayscale*, sehingga sebelum digunakan harus terlebih dahulu melalui proses ekstraksi fitur. Makalah ini tidak membahas cara kerja ekstraksi fitur titik *minutiae* dari sebuah citra sidik jari. Oleh karena itu, untuk melakukan proses ekstraksi fitur, digunakan aplikasi yang sudah ada, yakni Verifinger. Data uji coba yang digunakan sama seperti penelitian-penelitian sebelumnya, yaitu menggunakan impresi 1 dan 2, sehingga jumlah perbandingan untuk mendapatkan persentase *False Acceptance Rate* (FAR) sebesar 9.900 perbandingan, dan 100 perbandingan untuk memperoleh persentase *Genuine Acceptance Rate* (GAR) [10] - [12]. Jadi, total perbandingan dalam sekali percobaan adalah sebesar 10.000 perbandingan.

Uji coba dilakukan dengan sejumlah skenario yang berbeda dengan beberapa parameter yang digunakan untuk seleksi titik referensi dan proses pencocokan. Beberapa skenario tersebut antara lain uji coba dengan penerapan *fuzzy vault* biasa, uji coba dengan penerapan *extended fuzzy vault*, uji coba tanpa membandingkan fitur tipe *minutiae*, uji coba dengan perubahan parameter jumlah vektor kandidat saat interpolasi, dan uji coba dengan menggunakan parameter ambang jarak antar fitur sudut [11].

A. Performa Akurasi

Skenario pertama adalah uji coba akurasi dengan penerapan *non-extended fuzzy vault*, yang berarti tidak ada penggabungan himpunan *vault* dari beberapa fungsi polinomial. Hasil akurasi *non-extended fuzzy vault* akan dibandingkan secara langsung dengan penerapan *extended fuzzy vault*. Parameter yang digunakan untuk kedua metode tersebut sama, kecuali untuk parameter n_vault yang menentukan jumlah penggabungan himpunan polinomial. Tabel I menunjukkan perbandingan akurasi hasil uji coba skenario pertama. Hasil akurasi menunjukkan perolehan nilai GAR yang sama di antara kedua metode untuk jumlah data rahasia η yang berhasil dibuka mulai 1 sampai dengan 6. Yang membedakan antara keduanya adalah perolehan nilai FAR, yaitu metode *extended fuzzy vault* lebih baik dengan perolehan FAR yang lebih rendah. Pada $\eta = 3$ metode *extended fuzzy vault* memperoleh akurasi dengan nilai GAR 98% dan FAR 0,01%, sedangkan metode *non-extended fuzzy vault* menghasilkan akurasi dengan nilai GAR 98% dan FAR 0,1%.

Tabel II menyajikan data perbandingan akurasi untuk skenario kedua, yaitu perbandingan akurasi antara penggunaan dan tanpa penggunaan tipe *minutiae* saat pembandingan. Penggunaan tipe *minutiae* dalam proses pencocokan sedikit lebih unggul akurasinya karena perolehan nilai GAR secara keseluruhan lebih tinggi. Selain itu, pada perolehan nilai FAR, penggunaan tipe *minutiae* juga

TABEL I
AKURASI PENCOCOKAN ANTARA *NON-EXTENDED* DAN *EXTENDED FUZZY VAULT*

η	*Non-extended Fuzzy Vault*		*Extended Fuzzy Vault*	
	GAR (%)	**FAR (%)**	**GAR (%)**	**FAR (%)**
1	99	1,76	99	0,54
2	99	0,28	99	0,03
3	98	0,10	98	0,01
4	98	0,02	98	0,01
5	98	0,01	98	0
6	96	0	96	0

TABEL II
PERBANDINGAN AKURASI BERDASARKAN PENGGUNAAN TIPE *MINUTIAE*

η	Dengan Tipe *Minutiae*		Tanpa Tipe *Minutiae*	
	GAR (%)	**FAR (%)**	**GAR (%)**	**FAR (%)**
1	99	0,54	99	4,74
2	99	0,03	98	1,03
3	98	0,01	98	0,28
4	98	0,01	98	0,09
5	98	0	97	0,03
6	96	0	95	0

TABEL III
PERBANDINGAN AKURASI BERDASARKAN JUMLAH $N_{TOLERANSI}$

η	$n_{toleransi}$ = 13		$n_{toleransi}$ = 10	
	GAR (%)	**FAR (%)**	**GAR (%)**	**FAR (%)**
1	99	0,54	99	0,05
2	99	0,03	97	0
3	98	0,01	92	0
4	98	0,01	86	0
5	98	0	80	0
6	96	0	71	0

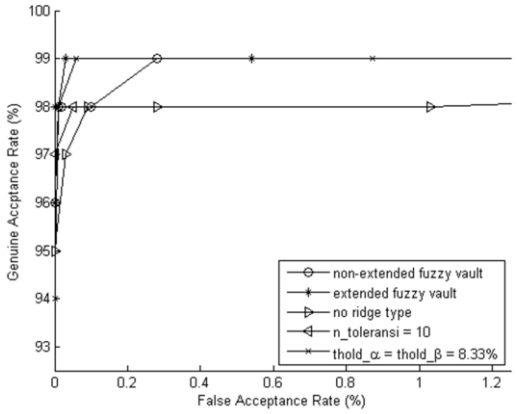
TABEL IV
AKURASI PENCOCOKAN BERDASARKAN THOLD_A DAN THOLD_B

η	thold_ α = thold_β = 7,5%		thold_α = thold_β = 8,33%	
	GAR (%)	**FAR (%)**	**GAR (%)**	**FAR (%)**
1	99	0,54	99	0,87
2	99	0,03	99	0,06
3	98	0,01	98	0,01
4	98	0,01	94	0
5	98	0	90	0
6	96	0	88	0

menghasilkan persentase yang lebih kecil, yang berarti kesalahan identifikasi lebih terminimalkan. Pada $\eta = 2$ diperoleh nilai GAR 99% dan FAR 0,03% untuk penggunaan tipe *minutiae*, sedangkan tanpa tipe menghasilkan nilai GAR 98% dan FAR 1,03%. Selisih akurasi yang cukup jauh antara keduanya, terutama untuk nilai FAR.

Uji coba skenario ketiga membandingkan penggunaan nilai $n_{toleransi} = 10$dengan nilai $n_{toleransi} = 13$ yang dipilih dalam makalah ini. Semakin besar nilai $n_{toleransi}$, peluang keberhasilan dalam perolehan data rahasia semakin tinggi.

Namun, jika terlalu besar, maka proses komputasi akan terlalu lama karena diperlukan komputasi kombinatorial



Gbr. 7 Grafik ROC dari beberapa skenario.

sebesar $C_{n_{toleransi}}^{10}$, dan kemungkinan terjadi kesalahan identifikasi juga semakin besar. Hasil uji coba pada Tabel III menunjukkan nilai GAR yang diperoleh maksimal 99% dengan FAR 0,05% dengan $\eta = 1$, sedangkan jika diambil $\eta = 2$, maka hanya mampu mencapai GAR 97% dengan FAR adalah 0%.

Hasil uji coba terakhir adalah penggunaan nilai parameter nilai ambang selisih sudut $thold\_\alpha$ dan $thold\_\beta$ menjadi sebesar 8,33%, yang diambil dari penelitian sebelumnya, karena elemen dari fitur komposit mirip dengan fitur *pair-polar* yang juga digunakan dalam makalah ini [11], [12]. Tabel IV menampilkan hasil perbandingan dengan nilai GAR yang diperoleh 99% dengan FAR 0,06%. Namun, dengan $\eta = 3$ diperoleh GAR yang sedikit lebih rendah, yaitu 98%, tetapi FAR juga lebih rendah, yaitu menjadi 0,01%. Jika dibandingkan dengan akurasi yang diperoleh dengan $thold\_\alpha = thold\_\beta = 7{,}5\%$, akurasinya lebih rendah dengan nilai perolehan data rahasia η yang sama.

Secara keseluruhan, akurasi yang dihasilkan paling optimal adalah GAR 99% dengan FAR 0,03%, dengan data rahasia minimal yang diperoleh adalah 2. Gbr. 7 menunjukkan grafik ROC dari akurasi keempat skenario uji coba seperti yang dijelaskan sebelumnya.

B. Analisis Keamanan

Metode yang diusulkan dalam makalah ini dibuat dengan teknik *extended fuzzy vault* yang menggabungkan beberapa fungsi polinomial ke dalam satu *vault*. Seperti diketahui, satu fungsi polinomial mewakili satu himpunan vektor fitur polar yang merupakan representasi hubungan antara satu titik *minutiae* referensi dengan titik-titik *minutiae* tetangga.

Dengan digabungkannya anggota himpunan titik-titik antar satu fungsi polinomial dengan dengan fungsi polinomial yang lain, maka akan lebih sulit mengelompokkan anggota vektor fitur yang merupakan satu ikatan dengan titik *minutiae* referensi tertentu. Usaha untuk melakukan rekonstruksi titik-titik *minutiae* cenderung lebih rumit.

Untuk memecahkan hanya satu data rahasia saja, penyerang dapat melakukan *brute force* dengan mencoba kemungkinan kombinasi sepuluh titik dari 200 titik yang ada pada *vault*

untuk metode *non-extended fuzzy vault*. Kompleksitasnya akan berkurang jika dalam satu *vault* terdapat empat data rahasia, yang berarti kemungkinan diperoleh sepuluh titik dari grup yang sama juga lebih besar, karena jumlah titik *chaff* yang relatif sedikit. Maksimum titik tetangga pada tiap titik *minutiae* referensi adalah 51 titik, sehingga dari keseluruhan kemungkinan kombinasi $C_{10}^{200} \approx 2{,}245 \times 10^{16}$ dan di antaranya $C_{10}^{51} \approx 1{,}28 \times 10^{10}$ kombinasi adalah benar. Maka, dengan empat polinomial di dalamnya, peluang didapatkan sepuluh data yang benar adalah $4 \times (1{,}28 \times 10^{10}) / 2{,}245 \times 10^{16} \approx 2{,}28 \times 10^{-6}$.

C. Kelebihan dan Kekurangan

Kelebihan pada metode ini adalah proses pencocokan dapat berjalan lebih ketat, sehingga dapat meminimalkan nilai persentase FAR yang muncul ketika nilai persentase GAR sudah cukup tinggi. Selain itu, nilai persentase GAR dapat diperoleh cukup tinggi hingga 99% dengan FAR 0,03% untuk minimal dua data rahasia yang diperoleh. Proses pembandingan pada metode ini lebih sederhana karena tidak perlu menggunakan sistem hierarki [11]. Jika dibandingkan dengan metode *pair-polar* yang memperoleh nilai GAR 90% dengan FAR 0,01%, metode ini sedikit lebih baik dengan memperoleh nilai GAR 98% dengan FAR 0,01% [12]. Bahkan, dengan enam data rahasia minimal yang bisa dibuka masih dapat mencapai GAR 96% dengan FAR 0%.

Kekurangan dari metode ini adalah rendahnya akurasi jika jumlah data rahasia yang harus dibuka lebih dari satu. Dari hasil uji coba, perolehan GAR untuk jumlah data rahasia minimal 6 ($\eta = 6$) adalah 96%, lebih rendah dari perolehan pada penelitian sebelumnya [11]. Pada proses seleksi tahap kedua, untuk beberapa titik *minutiae* referensi ada yang tidak bisa mendapatkan jumlah minimal sepuluh titik. Posisi titik ketetanggaan yang saling berjauhan dapat menjadi faktor kegagalan pembuatan *template*, sehingga dengan batas ambang radius yang ditentukan ($thold\_rad$) tidak dapat mencakup jumlah titik tetangga minimal sepuluh. Akibatnya, dalam pembuatan *template* ada sidik jari yang hanya menyisakan satu titik referensi untuk diproses, sehingga hanya ada satu data rahasia yang disembunyikan. Hal tersebut menyebabkan dalam hasil uji coba, maksimal GAR yang diperoleh adalah 99%, tidak dapat mencapai 100% seperti pada penelitian sebelumnya [12]. Penerapan skema *extended fuzzy vault* mengakibatkan proses pencocokan menjadi lebih banyak jika tidak diketemukan pasangan *template* dengan *query* yang cocok. Selain itu, informasi mengenai titik referensi tidak dapat diikutsertakan dalam *template*, sehingga penentuan kemiripan harus diperketat.

V. KESIMPULAN

Semakin populernya penggunaan data sidik jari sebagai sarana autentikasi ke dalam sebuah sistem menuntut keamanan yang lebih baik dalam penyimpanan data *template* sidik jari dalam basis data. Pada makalah ini diusulkan pengembangan metode transformasi data sidik jari dengan menentukan titik ketetanggaan berdasarkan area yang dibatasi radius tertentu terhadap suatu titik referensi. Selanjutnya, vektor fitur ketetanggaan yang diperoleh dengan koordinat polar ditambahkan fitur tipe *minutae* untuk digunakan pada proses pencocokan.

Proses pengamanan data dengan teknik *extended fuzzy vault* dipilih untuk membuat *cancelable template*. Hasil uji coba menunjukkan, akurasi yang dihasilkan dengan menambahkan pembandingan fitur tipe *ridge* pada saat pencocokan lebih baik jika dibandingkan tanpa penggunaan tipe *minutiae*. Hal itu ditunjukkan pada penurunan kesalahan deteksi penerimaan sidik jari yang berbeda yang diukur dengan perhitungan FAR. Begitu pula penerapan *extended fuzzy vault* mampu menekan kesalahan deteksi penerimaan sidik jari yang berbeda.

REFERENSI

[1] N. K. Ratha, S. Chikkerur, J. H. Connell and R. M. Bolle, "Generating Cancelable Fingerprint Templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, 2007.

[2] Y.-J. Chang, W. Zhung and T. Chen, "Biometrics-based cryptographic key generation," *IEEE ICME*, vol. 3, pp. 2203-2206, 2004.

[3] K. Xi, J. Hu dan F. Han, "An Alignment Free Fingerprint Fuzzy Extractor using Near-equivalent Dual Layer Structure Check (NeDLSC) Algorithm," *Proc. 6th ICIEA*, Beijing, 2011.

[4] U. Uludag, S. Pankanti and A. K. Jain, "Fuzzy vault for fingerprints," *Lecture Notes in Computer Science*, vol. 3546, pp. 310-319, 2005.

[5] U. Uludag and A. Jain, "Securing Fingerprint Template: Fuzzy Vault with Helper Data," *Proc. 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, 2006.

[6] T. Ahmad dan F. Han, "Cartesian and Polar Transformation-based Cancelable Fingerprint Template," *Proc. IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society*, Melbourne, 2011.

[7] T. C. Clancy, N. Kiyavash and D. J. Lin, "Secure Smartcard Based Fingerprint Authentication," *Proc. 2003 ACM SIGMM WBMA*, Berkley, California, 2003.

[8] T. Ahmad and J. Hu, "Generating cancelable biometric templates using a projection line," Proc. *11th ICARCV*, Singapore, 2010.

[9] Y. Sutcu, H. T. Sencar and N. Memon, "A Geometric Transformation to Protect Minutiae-Based Fingerprint Templates," *SPIE*, vol. 6539, pp. 65390E-1-8, 2007.

[10] K. Nandakumar, A. K. Jain and S. Pankanti, "Fingerprint-Based Fuzzy Vault: Implementation and Performance," *Information Forensics and Security*, vol. 2, no. 4, p. 744–757, 2007.

[11] K. Xi and J. Hu, "Biometric Mobile Template Protection: A Composite Feature based Fingerprint Fuzzy Vault," *Proc. ICC*, Dresden, 2009.

[12] T. Ahmad, J. Hu and S. Wang, "Pair-polar coordinate-based cancelable fingerprint templates," *Pattern Recognition*, vol. 44, no. 10-11, pp. 2555-2564, 2011.

[13] Z. Jin, A. B. J. Teoh, T. S. Ong and C. Tee, "Fingerprint Template Protection with Minutiae-based Bit-string for Security and Privacy Preserving," *Expert Systems with Applications*, vol. 39, pp. 6157-6167, 2012.

[14] M. V. Prasad and C. S. Kumar, "Fingerprint template protection using multiline neighboring relation," *Expert Systems with Applications*, vol. 41, no. 14, p. 6114–6122, 2014.

[15] J. Bringer, H. Chabanne and M. Favre, "Fuzzy Vault for Multiple Users," *Progress in Cryptology - AFRICACRYPT 2012*, A. Mitrokotsa and S. Vaudenay, Eds., Springer Berlin Heidelberg, 2012, pp. 67-81.

[16] Y. I. Riskajaya and T. Ahmad, "Pengembangan Metode Seleksi Titik Minutiae pada Sidik Jari dengan Radius Ketetanggaan," *Jurnal Ilmiah Teknologi Informasi*, vol. 13, no. 1, 2015.

[17] Preston L. Bannister. (2009) "Extending the Pearson Hash Function to Larger Value," . [Online], http://bannister.us/weblog/2009/extending-the-pearson-hash-function-to-larger-values/ , tanggal akses: 20 Desember 2014.

[18] (2002) BioLab - University of Bologna, "FVC2002 Fingerprint verification competition,". [Online], http://bias.csr.unibo.it/fvc2002/download.asp, tanggal akses: 14 Oktober 2014.