

## BAB 2 TRANSFORMASI KARHUNEN-LOEVE DAN JENIS INTERPOLASI

Sebuah pola dapat didefinisikan sebagai entitas yang terdefinisi dan dapat diberi suatu identifikasi atau nama [FAT99]. Seperti pengenalan pola, pengenalan sudut pandang juga mengumpulkan informasi pola-pola dari berbagai macam sudut pandang obyek yang disusun dalam sebuah basis data pola sudut pandang.

Untuk mengekstraksi ciri ke dalam sebuah ruang eigen diperlukan sebuah pendekatan yang bernama transformasi Karhunen-Loeve (KLT). Dengan menggunakan transformasi ini dimensi data dapat dikurangi dan dapat meminimalkan kesalahan dalam perhitungan. Dalam penelitian sebelumnya, telah dikembangkan 2 transformasi data yang berbeda dengan menggunakan transformasi Karhunen-Loeve. Dua cara transformasi tersebut adalah sebagai berikut:

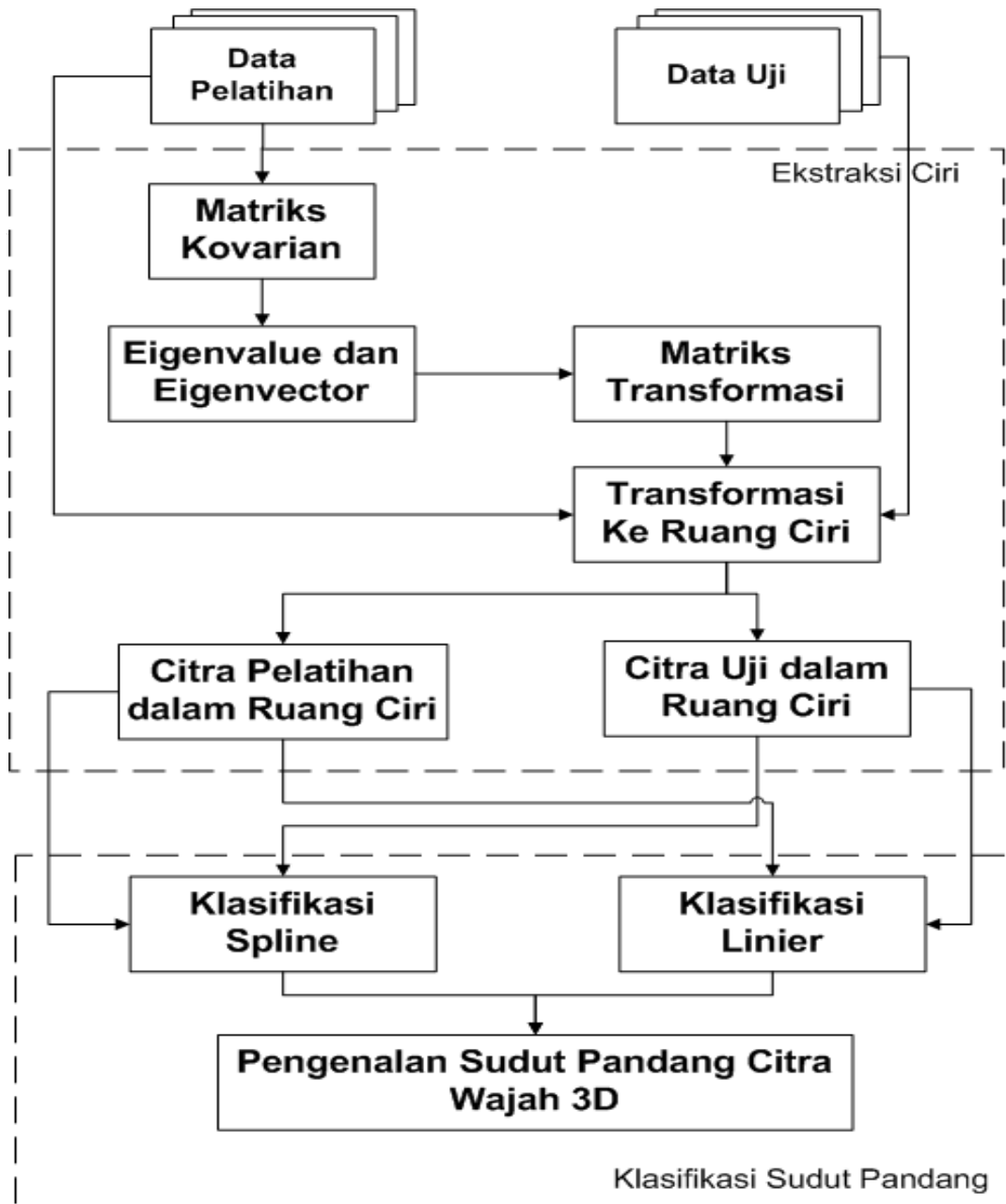
- *Fully* KLT, yaitu semua citra pelatihan ditransformasikan ke dalam satu buah ruang eigen sehingga dalam data ruang eigen terbentuk beberapa kelas sudut pandang. Citra uji yang diujikan juga ditransformasikan ke dalam ruang eigen tersebut
- *Partially* KLT, yaitu beberapa citra acuan untuk suatu kelas sudut pandang yang sama ditransformasikan ke dalam suatu ruang eigen. Sedangkan untuk kelas sudut pandang yang lain juga ditransformasikan ke dalam ruang eigen yang lain. Citra uji yang diujikan ditransformasikan ke dalam masing-masing ruang eigen yang ada dan dilakukan proses klasifikasi

Pada penelitian ini penulis melakukan proses pembentukan ruang ciri berdasarkan kelas wajah dan kelompok sudut pandang horizontal/vertikal. Penjelasan mengenai proses pembentukan ruang ciri akan dijelaskan pada Bab 3.

Setelah dilakukan proses pembentukan ruang ciri dengan menggunakan transformasi Karhunen-Loeve, titik-titik uji yang ditransformasikan ke setiap ruang ciri diklasifikasikan menggunakan klasifikasi sudut pandang. Untuk melakukan proses klasifikasi sudut pandang digunakan interpolasi linier dan interpolasi spline. Interpolasi ini digunakan untuk menghubungkan titik-titik ciri yang berada dalam ruang ciri. Dengan menghubungkan setiap titik-titik ciri dapat di-*generate* titik-titik ciri dari citra dengan sudut pandang yang berada diantara titik-titik ciri citra pelatihan.

Dalam penelitian ini penulis membandingkan antara interpolasi linier dan interpolasi spline sebagai pembentuk ruang ciri. Dengan membandingkan kedua interpolasi tersebut diharapkan dapat mengetahui jenis interpolasi spline terbaik sebagai pembentuk garis ciri. Proses pembentukan garis ciri dan klasifikasi sudut pandang akan dijelaskan pada bab 3.

Diagram dibawah ini dapat memperjelas proses penentuan sudut pandang citra wajah 3 dimensi dalam sistem.



**Gambar 2.1** Diagram Sistem Pengenalan Sudut Pandang Wajah 3 Dimensi

### 2.1 Transformasi Karhunen-Loeve (Principal Component Analysis)

Pengolahan sebuah citra dengan menggunakan data citra dalam ruang spasial memerlukan sumber daya yang sangat besar. Selain itu kemungkinan kesalahan dalam pengolahan data meningkat dengan semakin besarnya dimensi yang kita gunakan. Untuk itu diperlukan sebuah representasi data citra dengan dimensi kecil namun data tidak mengurangi informasi yang terkandung dalam data tersebut.

Salah satu metode untuk mereduksi dimensi sebuah data citra adalah Transformasi Karhunen-Loeve. Metode ini mampu mentransfer suatu citra dari ruang spasial (dimensi besar) ke ruang yang mempunyai dimensi lebih kecil,

ruang ini disebut dengan ruang eigen. Dimensi yang ada dalam ruang spasial direduksi dengan mencari vektor-vektor eigen yang paling berpengaruh dari data yang diindikasikan dengan nilai eigen yang besar dari vektor tersebut. Vektor-vektor eigen tersebut dibangkitkan dari sebuah matriks kovarian yang merupakan representasi dari distribusi data spasial citra [KRI02].

Tujuan dari transformasi ini adalah mengubah data X dengan dimensi D menjadi data Y dengan dimensi L yang lebih kecil. Dengan kata lain kita mencari matriks Y dengan melakukan transformasi Karhunen-Loeve (PCA) terhadap matriks X.

$$Y = PCA \{X\}$$

Transformasi ini mempunyai beberapa langkah untuk melakukan transformasi data citra dari ruang spasial ke dalam ruang eigen. Langkah-langkah transformasi Karhunen-Loeve adalah sebagai berikut:

∴ Menyusun dataset

Apabila kita memiliki data citra dengan ukuran MxN piksel maka data tersebut mempunyai MxN variabel. Kita susun data vektor kolom tersebut menjadi vektor tunggal yang berdimensi MxN

$$\vec{X}_i = \begin{bmatrix} x_{11} \\ x_{12} \\ x_{13} \\ \vdots \\ \vdots \\ x_{im-3} \\ x_{im-2} \\ x_{im} \end{bmatrix}$$

∴ Menghitung mean data

Kita cari vektor rata-rata  $\vec{m}$  dari vektor  $\vec{X}_i$  yang berjumlah N

$$\vec{m} = \frac{\sum_{i=1}^N \vec{X}_i}{N}$$

∴ Mencari matriks kovarian

Matriks kovarian merupakan matriks yang menggambarkan hubungan satu dimensi dengan dimensi yang lain. Matriks kovarian merupakan hasil dari *outer product* dari matriks deviasi mean. Persamaan matriks kovariannya adalah sebagai berikut

$$C = \frac{\sum_{i=1}^N (\vec{x}_i - \vec{m})(\vec{x}_i - \vec{m})^T}{N}$$

∴ Menyusun nilai eigen dan matriks eigen

Matriks kovarian merupakan matriks bujur sangkar maka dapat diperoleh suatu himpunan vektor eigen yang orthonormal [ANT99]. Dengan

menggunakan metode jacobi terhadap matriks kovarian dapat diperoleh vektor-vektor eigen dan nilai-nilai eigennya. Kita definisikan matriks E sebagai kumpulan vektor-vektor eigen dari matriks kovarian dan matriks  $\vec{\lambda}$  sebagai kumpulan nilai-nilai eigen dari matriks kovarian.

$$E = \left[ \begin{array}{c|c|c|c} \vec{e}_1 & \vec{e}_2 & \cdots & \vec{e}_D \end{array} \right] = \begin{bmatrix} e_{11} & e_{12} & \cdots & e_{1D} \\ e_{21} & e_{22} & \cdots & e_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ e_{D1} & e_{D2} & \cdots & e_{DD} \end{bmatrix}$$

$$\vec{e} = \begin{bmatrix} e_{1j} \\ \vdots \\ e_{Dj} \end{bmatrix}$$

$$\vec{\lambda} = [\lambda_1 \quad \lambda_2 \quad \cdots \quad \lambda_D]$$

Kemudian vektor  $\vec{e}_j$  kita urutkan berdasarkan nilai eigennya dengan  $\vec{e}_1$  adalah vektor dengan nilai eigen terbesar dan  $\vec{e}_D$  merupakan nilai eigen terkecil.

∴ Pilih irisan matriks eigen sebagai vektor basis

Untuk mendapatkan matriks di ruang eigen dengan jumlah dimensi yang lebih kecil maka kita harus memilih K ( $K < N$ ) vektor eigen pertama yang telah diurutkan berdasarkan nilai eigen terbesar. Vektor-vektor eigen yang mempunyai nilai eigen besar mempunyai peran penting dalam proses transformasi dimensi [SMI02]. Hal ini disebabkan karena semakin tinggi nilai eigen maka tinggi pula lebar distribusi data pada vektor basis tersebut sehingga dengan mengabaikan vektor basis dengan nilai eigen yang rendah (mendekati nol) tidak mempengaruhi akurasi informasi dari data.

Dengan kita memilih K buah vektor eigen yang mempunyai nilai terbesar maka kita akan mendapatkan matriks transformasi W dengan ukuran KxD yang merupakan irisan dari matriks E ( $K < D$ )

$$W = \left[ \begin{array}{c|c|c|c} \vec{e}_1 & \vec{e}_2 & \cdots & \vec{e}_K \end{array} \right] = \begin{bmatrix} e_{11} & e_{12} & \cdots & e_{1K} \\ e_{21} & e_{22} & \cdots & e_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ e_{K1} & e_{K2} & \cdots & e_{KK} \end{bmatrix}$$

Jika K bernilai 20, berarti  $W = \left[ \begin{array}{c|c|c|c} \vec{e}_1 & \vec{e}_2 & \cdots & \vec{e}_{20} \end{array} \right]$  maka akan didapat matriks di ruang eigen yang mempunyai ukuran dimensi 20xD.

∴ Melakukan standarisasi data menggunakan *z-scores*

Sebelum dilakukan transformasi ke dalam ruang eigen harus dilakukan standarisasi terhadap data citra. Hal ini berfungsi untuk mengukur banyaknya standar deviasi yang berada di atas atau di bawah mean.

$$Z_i = \frac{\vec{X}_i - \vec{m}}{\sigma}$$

∴ Proyeksikan data *z-scores* ke basis yang baru

Untuk melakukan proyeksi ke ruang eigen maka kita akan melakukannya dengan mengalikan vektor transformasi  $W$  dengan data asli yang sudah distandarisasi. Kita akan mendapatkan data baru  $Y$  dengan dimensi yang lebih kecil tanpa mengurangi informasi yang terkandung di dalam data.

$$Y = W * Z = \text{PCA}(X)$$

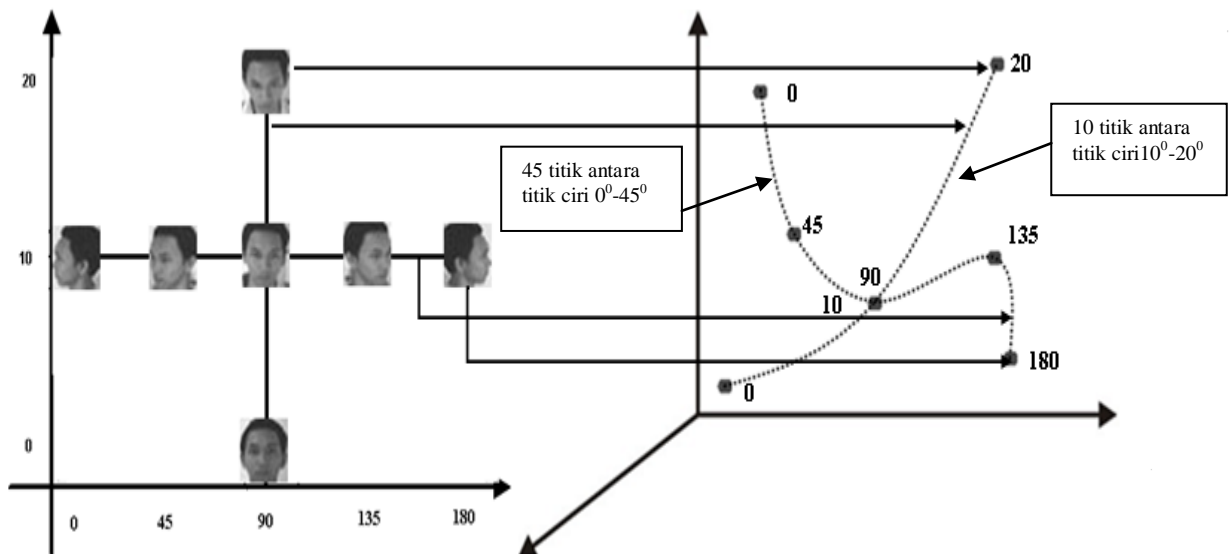
## 2.2 Jenis Interpolasi

Di bidang matematika, khususnya analisa numerik, interpolasi merupakan sebuah metode untuk menghasilkan titik data baru pada sebuah rentang sebuah data yang telah diketahui.

Dalam bidang teknik atau ilmu pengetahuan interpolasi digunakan sebagai pembentuk fungsi pendekatan yang dapat memenuhi titik sebuah data. Interpolasi sendiri mempunyai beberapa jenis yaitu interpolasi linier, interpolasi polinomial, dan interpolasi spline.

Interpolasi dalam penelitian ini digunakan untuk membentuk garis ciri yang menghubungkan antara satu titik ciri ke titik ciri yang lain. Garis ciri dibentuk dari titik-titik yang di-generate menggunakan interpolasi dimana titik-titik tersebut mempunyai jarak yang sangat pendek satu sama lain. Karena jarak satu titik ke titik yang lain sangat pendek maka mampu menghasilkan garis ciri.

Jumlah dari titik yang di-generate oleh interpolasi antara satu titik ciri ke titik ciri yang lain tergantung dari selisih sudut kedua titik ciri tersebut. Misalkan untuk titik ciri dengan sudut horizontal antara  $0^{\circ}$  dengan  $45^{\circ}$  dibagi menjadi 45 titik sedangkan untuk titik ciri dengan sudut vertikal antara  $10^{\circ}$  dengan  $20^{\circ}$  dibagi menjadi 20 titik. Setiap titik yang di-generate oleh interpolasi ini merepresentasikan perubahan sudut pandang dengan ketelitian hingga  $1^{\circ}$ . Setelah dilakukan generalisasi titik-titik ciri baru antara 2 titik ciri kemudian dilakukan penghitungan jarak antara titik uji dengan titik ciri dan titik-titik ciri baru menggunakan jarak Euclid. Titik yang mempunyai jarak minimum dengan titik uji merupakan sudut pandang tebakan dari sistem.



Gambar 2.2 Ilustrasi Pembentukan Titik-titik Generalisasi dengan Menggunakan Interpolasi Spline

Dalam penelitian ini penulis ingin membandingkan akurasi dari penggunaan interpolasi linier dan interpolasi spline sebagai penghubung antara titik ciri satu dengan titik ciri yang lainnya. Berikut ini penulis akan menjelaskan masing-masing interpolasi yang digunakan dalam penelitian ini.

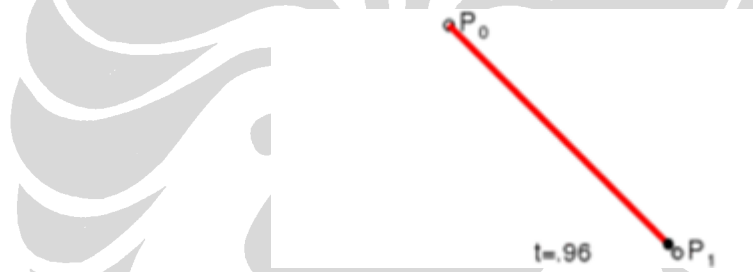
### 2.2.1 Interpolasi Linier

Salah satu interpolasi yang paling sederhana adalah interpolasi linier. Interpolasi linier dapat dibentuk dengan cepat namun memberikan hasil yang kurang presisi. Selain itu interpolasi ini tidak *differentiable* pada titik ke  $k$  dan memiliki tingkat kesalahan yang cukup tinggi bila dibandingkan interpolasi yang lain.

Interpolasi linier merupakan interpolasi yang akan menghasilkan garis lurus sebagai penghubung antara titik satu dengan titik yang lain. Misalkan titik pertama adalah  $P_0$  dan titik kedua adalah  $P_1$  maka interpolasi ini harus memenuhi persamaan berikut ini

$$B(t) = P_0 + (P_1 - P_0)t$$

Variabel  $t$  dalam persamaan tersebut menggambarkan titik ke  $t$  yang di-*generate* oleh interpolasi linier antara  $P_0$  ke  $P_1$ .



Gambar 2.3 Interpolasi Linier yang Menghubungkan Titik  $P_0$  dan  $P_1$

### 2.2.2 Interpolasi Spline

Spline merupakan sebuah bagian polinomial yang memenuhi kondisi kontinuitas diantara bagian-bagiannya. Spline juga sering dikaitkan dengan fungsi matematis yang digunakan oleh aplikasi yang membutuhkan interpolasi data. Spline dapat digunakan untuk melakukan interpolasi dalam berbagai dimensi, baik dalam dimensi rendah maupun dimensi tinggi. Penghalusan spline dapat dipandang sebagai proses generalisasi dari interpolasi spline dimana fungsi spline digunakan untuk meminimalisasi kombinasi aproksimasi error atas data yang diamati.

Interpolasi spline menggunakan polinomial dengan *degree* yang rendah sehingga mampu membentuk garis dengan halus. Tidak seperti polinomial yang lain, interpolasi spline menghasilkan kesalahan yang rendah dan menghasilkan kurva yang lebih halus.

### 2.2.2.1 Bezier Spline

Kurva Bezier pertama kali dipublikasikan oleh seorang insinyur berkebangsaan Perancis pada tahun 1962 yang digunakan untuk mendesain rangka mobil. Untuk membentuk kurva Bezier digunakan algoritma de Casteljau yang ditemukan oleh Paul de Casteljau pada tahun 1959. Algoritma ini merupakan metode yang cukup stabil untuk mengevaluasi kurva Bezier.

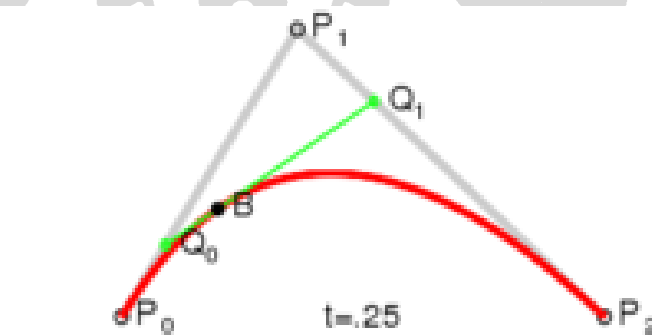
Kurva Bezier sering digunakan untuk memperhalus kurva dalam bidang grafik komputer. Kurva ini dibentuk dengan melakukan interpolasi antara titik-titik kontrolnya dan dapat dilakukan beberapa transformasi dengan melakukan transformasi tersebut terhadap titik-titik kontrolnya.

Kurva Bezier memiliki beberapa jenis, tergantung dari jumlah titik kontrolnya. *Quadratic* dan *Cubic* Bezier merupakan beberapa jenis kurva Bezier yang digunakan dalam bidang grafik computer karena untuk mengevaluasi kurva dengan *degree* lebih tinggi membutuhkan sumber daya yang lebih tinggi. Dalam penelitian ini penulis menggunakan *Quadratic* Bezier untuk menghubungkan titik-titik ciri di dalam ruang ciri wajah.

*Quadratic* Bezier dibentuk dari 3 buah titik kontrol dan harus memenuhi persamaan berikut ini:

$$B(t) = (1-t)^2 P_0 + 2t(1-t) P_1 + t^2 P_2$$

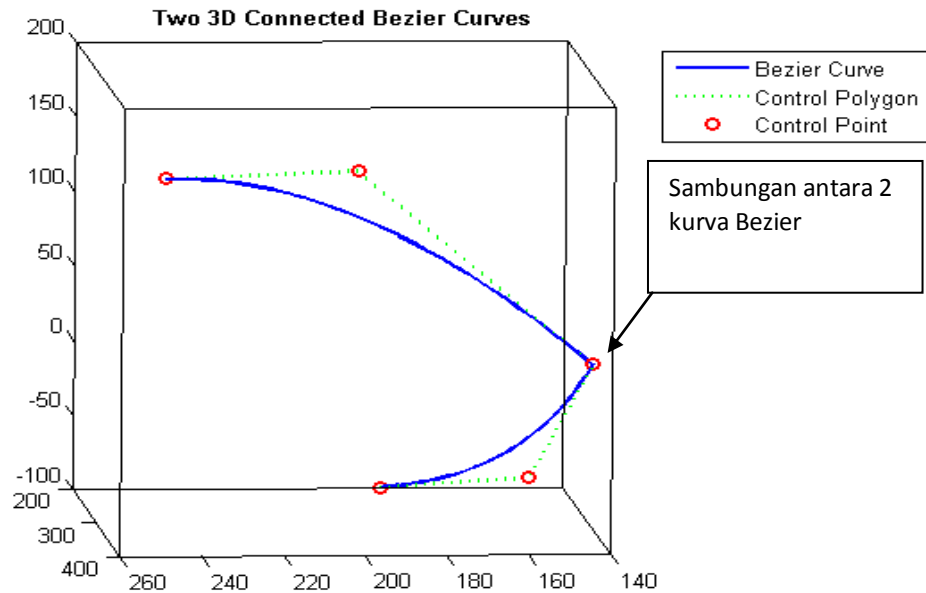
Untuk satu kurva *Quadratic* Bezier terbentuk titik tengah  $Q_0$  dan  $Q_1$  sedemikian rupa selama  $t$  berubah-ubah nilainya.



**Gambar 2.4 Pembentukan *Quadratic* Bezier Spline**

Keterangan gambar:

- Titik  $Q_0$  berubah-ubah dari titik  $P_0$  ke  $P_1$  dan menggambarkan kurva linier (garis lurus)
- Titik  $Q_1$  berubah-ubah dari titik  $P_1$  ke  $P_2$  dan menggambarkan kurva linier (garis lurus)
- Titik  $B(t)$  berubah-ubah dari  $Q_0$  ke  $Q_1$  dan menggambarkan kurva *Quadratic* Bezier



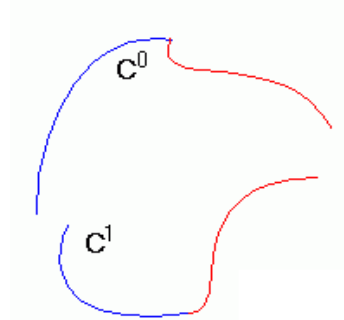
**Gambar 2.5 Proses Penyambungan Bezier Spline dalam Ruang 3 Dimensi**

Karena jumlah dari titik ciri lebih dari 3 buah maka harus dilakukan penyambungan di setiap titik akhir dari kurva Bezier. Hal ini dilakukan untuk menghemat biaya komputasi dan meningkatkan ketelitian generalisasi titik-titik di dalam kurva.

### 2.2.2.2 Cardinal Spline

Cardinal spline merupakan interpolasi spline yang menggunakan *tension* untuk membentuk sebuah polygon atau poly-line. Cardinal spline menggunakan posisi dari titik sekarang, sepanjang titik sebelum dan sesudahnya serta menyeimbangkan posisinya dengan menggunakan *tension*. Hal ini akan memperhalus garis dan membuat jalur garis lebih halus dalam melewati titik-titik kontrolnya.

Interpolasi Cardinal spline merupakan modifikasi dari *Quadratic* Bezier spline yang menggunakan proses penyambungan dengan kontinuitas  $C^1$ . Kurva kontinu pada  $C^k$  apabila semua turunan ke  $k$  dari kurva juga kontinu.



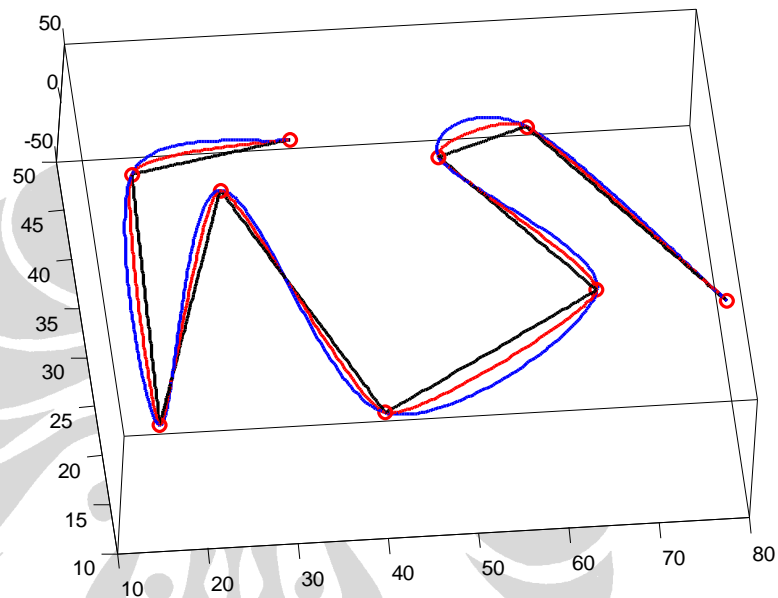
**Gambar 2.6 Perbandingan Hasil Penyambungan Bezier Spline dengan kontinuitas  $C^0$  dan  $C^1$**



Sebuah Cardinal spline dapat dibentuk minimal 4 titik dan harus memenuhi persamaan berikut ini:

$$P(t) = \begin{pmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -c & 0 & c & 0 \\ 0 & -c & 0 & c \end{bmatrix} \begin{bmatrix} x_{-1} \\ x_0 \\ x_1 \\ x_2 \end{bmatrix} \end{pmatrix} \begin{bmatrix} 2t^3 - 3t^2 + 1 \\ -2t^3 + 3t^2 \\ t^3 - 2t^2 + t \\ t^3 - t^2 \end{bmatrix}$$

3D Cardinal Spline



**Gambar 2.7 Cardinal Spline dengan  $c = 1$  (hitam),  $c = 0.5$  (merah), dan  $c = 0$  (biru)**

Dengan  $0 \leq c \leq 1$  dan nilai  $c$  dapat mempengaruhi kehalusan dari garis yang dihasilkan. Ketika nilai  $c = 1$  maka akan menghasilkan sambungan kurva yang paling halus dan jika  $c = 0$  akan menghasilkan sambungan kurva yang paling kasar.

## **BAB 3 PEMBENTUKAN RUANG CIRI DAN KLASIFIKASI SUDUT PANDANG**

Manusia mempunyai kemampuan istimewa untuk mengenali wajah seseorang yang pernah mereka temui atau lihat. Kemampuan ini masih mampu mengenali wajah seseorang dengan tepat walaupun sudah terjadi beberapa perubahan seperti usia, ekspresi, sikap dan tambahan beberapa aksesoris di wajahnya. Seiring dengan meningkatnya kebutuhan akan sistem pengenalan wajah 3 dimensi yang handal banyak peneliti yang menyumbangkan kemampuannya untuk mengembangkan sistem pengenalan wajah 3 dimensi dengan berbagai sudut pandang.

Sistem pengenalan wajah 3 dimensi merupakan sistem yang mengenali wajah manusia dengan membandingkan citra asing dengan model wajah yang tersedia di *database* sistem. Selain itu sistem juga harus mampu mengklasifikasikan citra wajah sebagai *outlier* jika citra wajah tersebut tidak terdapat dalam *database* sistem.

Sistem pengenalan wajah 3 dimensi yang handal harus mampu mengenali wajah dari seseorang dengan berbagai sudut pandang atau pose. Hal ini terbukti bahwa pengenalan wajah dapat ditingkatkan dengan melakukan pelatihan pengenalan wajah dan penebakan sudut pandang bersama-sama. Untuk melengkapi metode pengenalan obyek 3D menggunakan *Cylindrical Structure of Hidden Layer Neural Network* (CSHL) [KUS01], penulis mengembangkan sistem pengenalan sudut pandang dengan menggunakan interpolasi spline sebagai penghubung titik-titik ciri dalam ruang eigen.

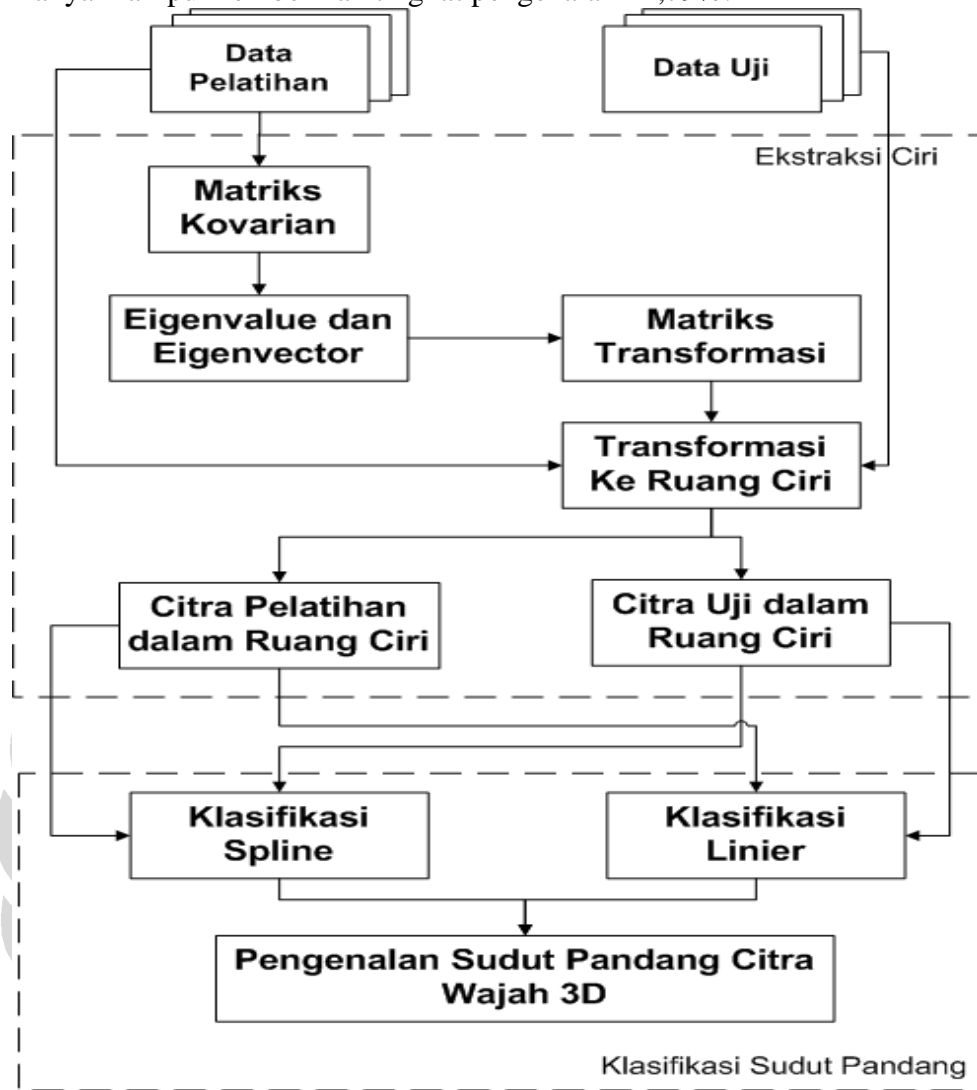
Sistem ini terdiri dari 2 subsistem yaitu ekstraksi ciri dan klasifikasi sudut pandang. Sebagai metode ekstraksi ciri, penulis menggunakan transformasi Karhunen-Loeve untuk mengubah vektor citra wajah dari domain spasial ke bentuk vektor lain di ruang eigen yang memiliki dimensi yang lebih kecil. Sedangkan sebagai klasifikasi sudut pandang kami menggunakan klasifikasi linier (menggunakan interpolasi linier untuk menghubungkan titik-titik ciri) dan klasifikasi spline (menggunakan interpolasi spline untuk menghubungkan titik-titik ciri). Diagram pada gambar 3.1 dapat memperjelas proses penentuan sudut pandang citra wajah 3 dimensi dalam sistem ini.

### **3.1 Pembentukan Ruang Ciri**

#### **3.1.1 Pembentukan Ruang Ciri Berdasarkan Kelas Data Wajah Pelatihan**

Pada penelitian sebelumnya telah dilakukan beberapa pendekatan untuk membentuk ruang ciri. Pada penelitian Lina [LIN04] dilakukan pembentukan ruang ciri dengan menggunakan semua citra pelatihan sebagai titik ciri dan menarik garis ke semua titik ciri (lihat gambar 2.1). Hasil dari penelitian ini

menunjukkan bahwa pembentukan ruang ciri dengan menggunakan pendekatan ini hanya mampu memberikan tingkat pengenalan 42,79%.

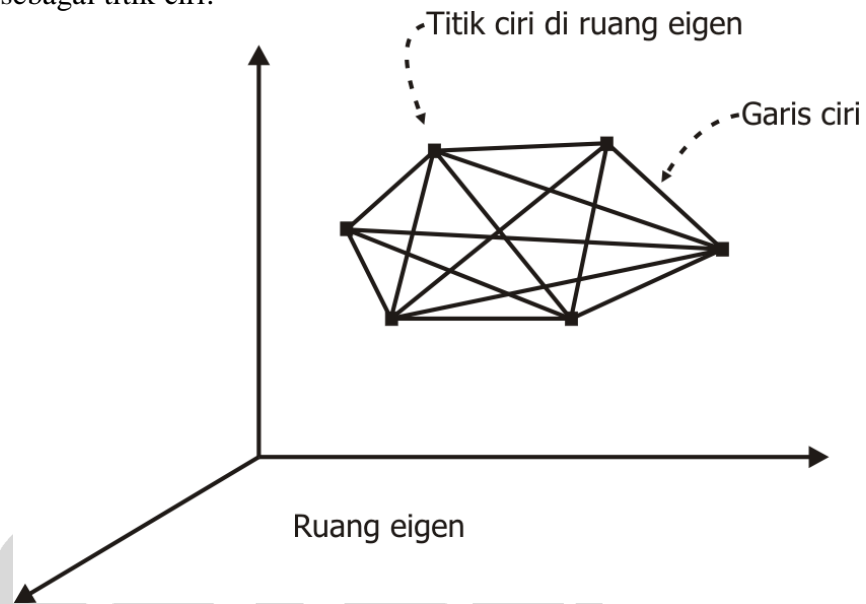


**Gambar 3.1 Diagram Sistem Pengenalan Sudut Pandang Wajah 3 Dimensi**

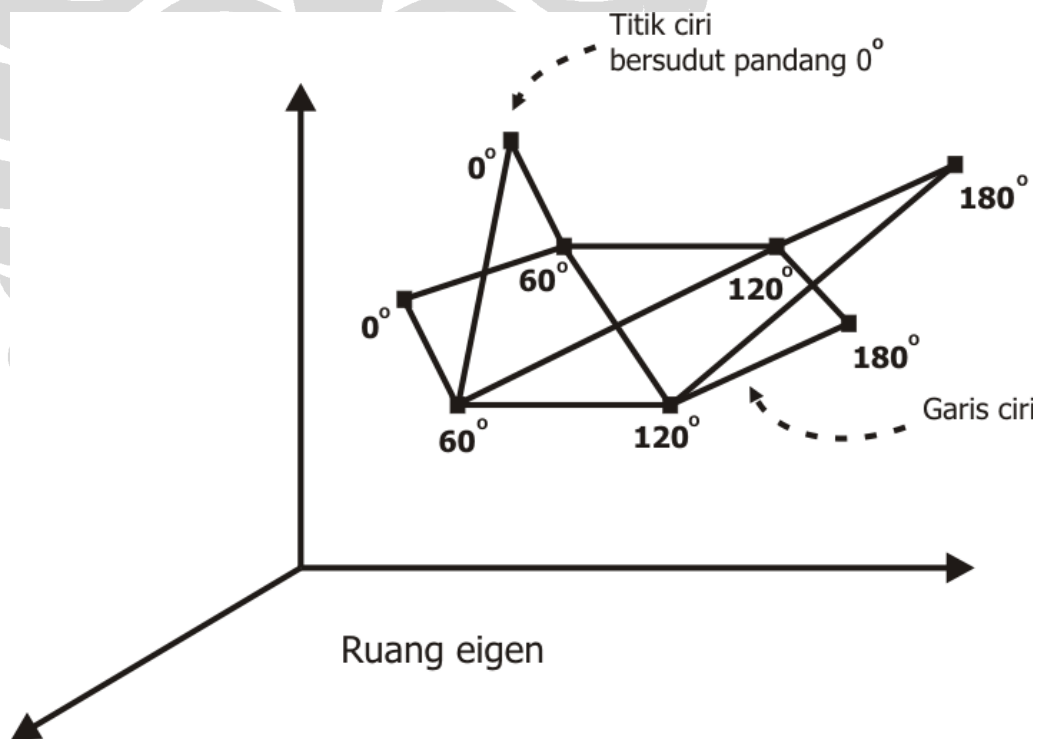
Kemudian Efrika [EFR04] melakukan modifikasi pembentukan ruang ciri dengan membatasi pembentukan ruang ciri pada beberapa sudut tertentu. Dalam salah satu percobaannya Efrika mencoba menghubungkan titik-titik ciri dengan sudut  $0^{\circ}$  ke titik-titik ciri dengan sudut  $60^{\circ}$ , titik-titik ciri dengan sudut  $60^{\circ}$  ke titik-titik ciri dengan sudut  $120^{\circ}$ , dan titik-titik ciri dengan sudut  $120^{\circ}$  ke titik-titik ciri dengan sudut  $180^{\circ}$ . Hal ini dilakukan karena menurut Efrika untuk mengenali sudut  $120^{\circ}$  tidak diperlukan garis antara sudut  $120^{\circ}$ - $120^{\circ}$  karena sudut  $120^{\circ}$  dapat dikenali pada garis ciri  $60^{\circ}$ - $120^{\circ}$  atau pada garis ciri  $120^{\circ}$ - $180^{\circ}$ . Hasil dari percobaan ini mampu meningkatkan tingkat pengenalan sistem dari 32,6% menjadi 61,8%.

Dalam penelitian selanjutnya Rusdi [RUS06] mencoba untuk melakukan pendekatan baru dalam usaha peningkatan tingkat pengenalan sistem. Rusdi mencoba untuk mengurangi jumlah titik ciri yang digunakan sehingga dapat mengurangi jumlah garis yang terbentuk dalam ruang eigen. Hal ini dilakukan

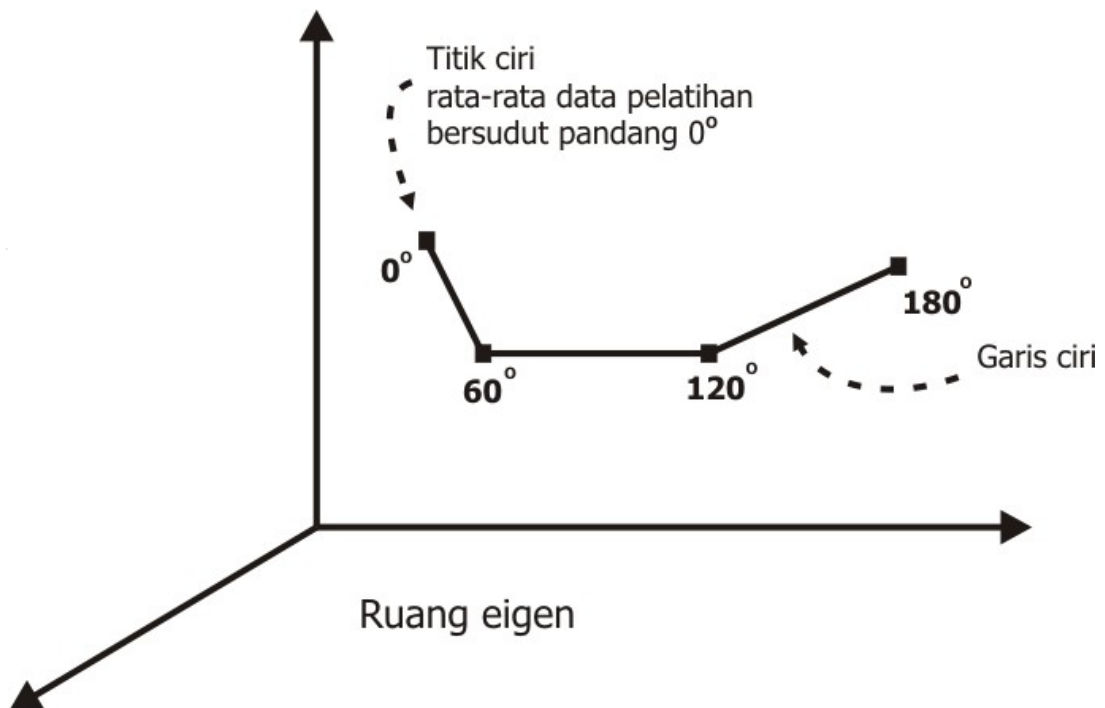
dengan harapan hanya terbentuk satu garis ciri yang saling menghubungkan tiap sudut data pelatihan dengan menjadikan nilai rata-rata dari tiap sudut data pelatihan sebagai titik ciri.



**Gambar 3.2 Pembentukan Garis Ciri dengan Menghubungkan Seluruh Titik-titik Ciri**



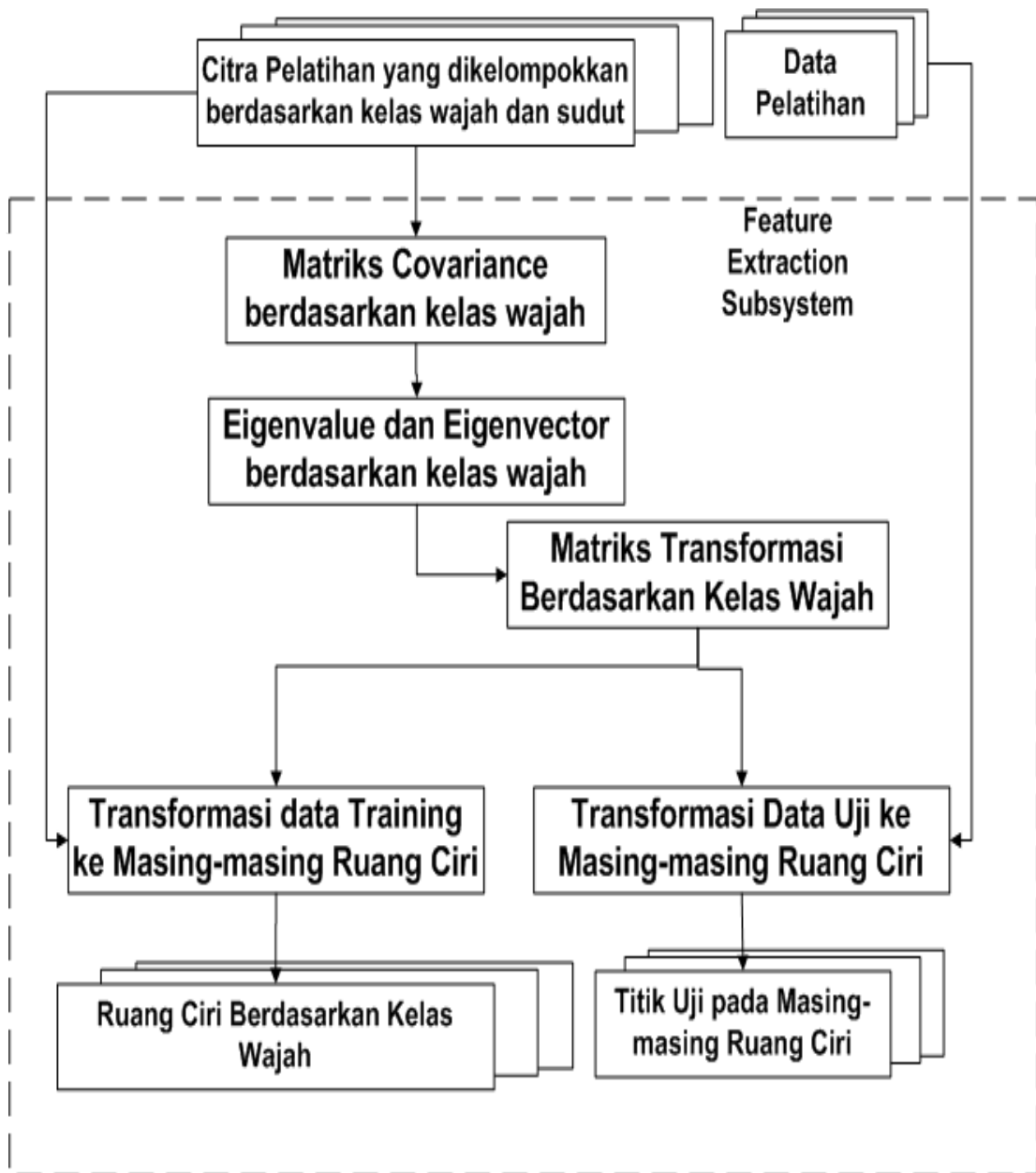
**Gambar 3.3 Pembentukan Garis Ciri dengan Menghubungkan Titik-titik Ciri dengan Selisih 60°**



**Gambar 3.4 Pembentukan Garis Ciri yang Terbatas Pada Rata-rata Tiap Sudut Pelatihan**

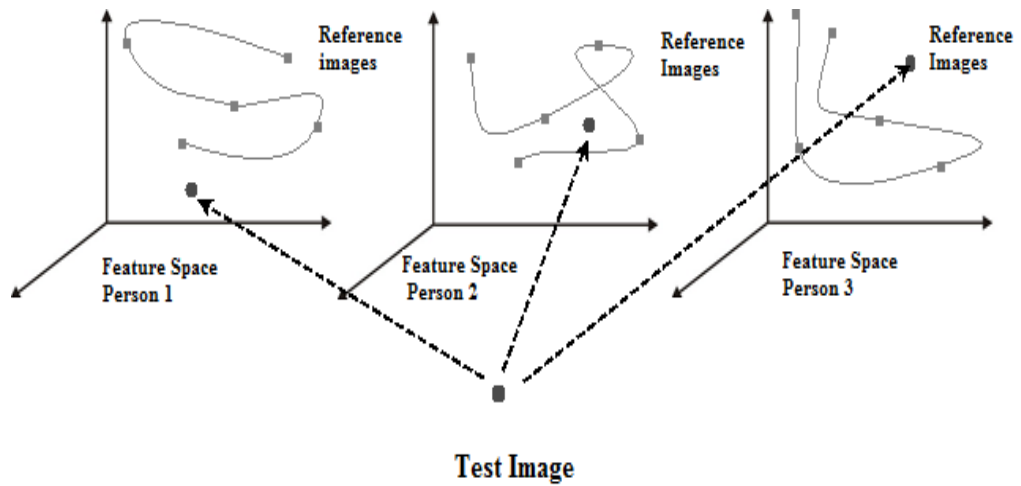
Dalam penelitian ini penulis mencoba untuk melakukan pendekatan baru dalam pembentukan ruang ciri. Dalam pendekatan ini penulis memisahkan ruang ciri berdasarkan kelas wajah dari data *training* serta berdasarkan kelas wajah dan kelompok vertikal/horizontal. Hal ini dilakukan dengan harapan mampu meningkatkan ketelitian dalam pembentukan ruang ciri.. Pada setiap ruang wajah akan terbentuk lebih dari 1 garis ciri, tergantung dari jumlah data *training*. Dalam eksperimen, digunakan 4 data wajah yaitu wajah Adhit, Inu, Mamat, dan Uyo. Setiap wajah akan membentuk ruang ciri tersendiri yang terdiri dari titik-titik ciri dengan sudut tertentu. Diagram pada gambar 2.4 dapat memperjelas pembentukan ruang ciri berdasarkan wajah data *training*.

Penelitian dilakukan untuk melakukan estimasi atas sudut pandang 3 dimensi dari sebuah citra wajah. Sistem akan menghasilkan *output* berupa sudut pandang horizontal dan vertikal. Untuk melakukan klasifikasi sudut pandang di dalam ruang ciri, penulis mengelompokkan data *training* berdasarkan sudut horizontal dan vertikal. Sebagai contoh citra wajah yang mempunyai sudut horizontal  $90^0$  dikelompokkan dengan citra sudut dengan sudut horizontal  $90^0$  dan citra wajah yang mempunyai sudut vertikal  $10^0$  dikelompokkan dengan citra sudut dengan sudut vertikal  $10^0$ .

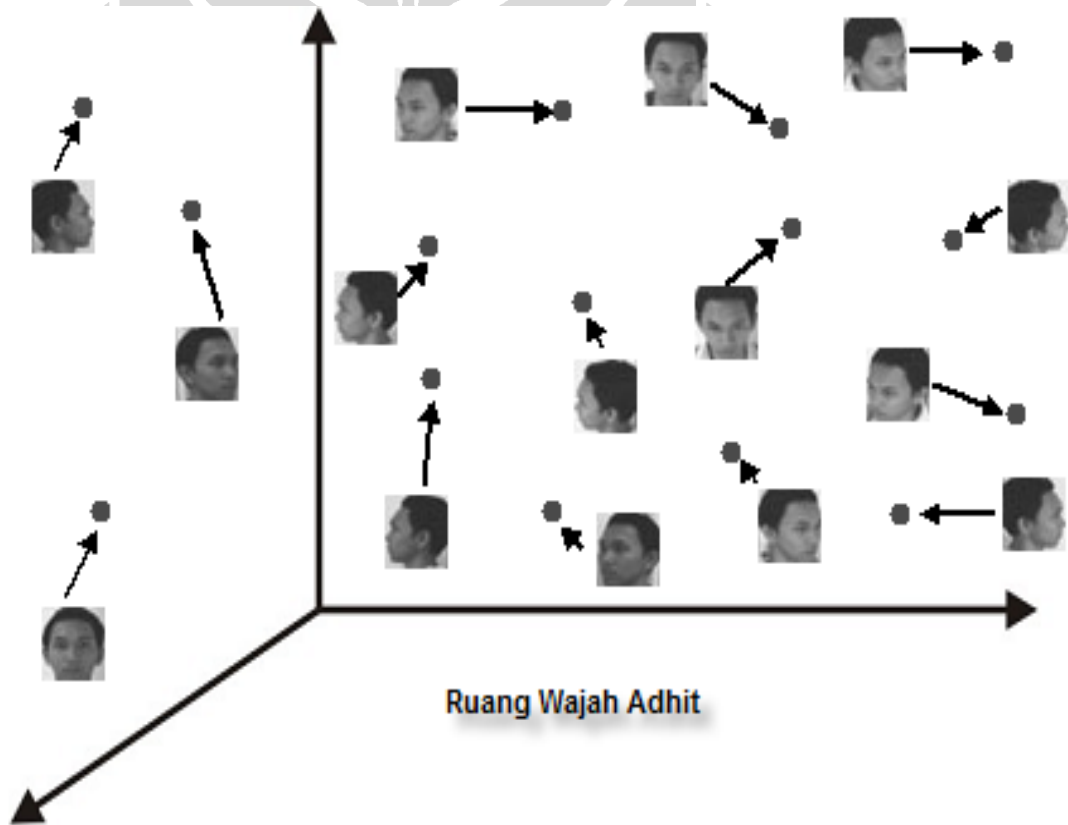


**Gambar 3.5 Diagram Pembentukan Ruang Wajah Berdasarkan Kelas Wajah**

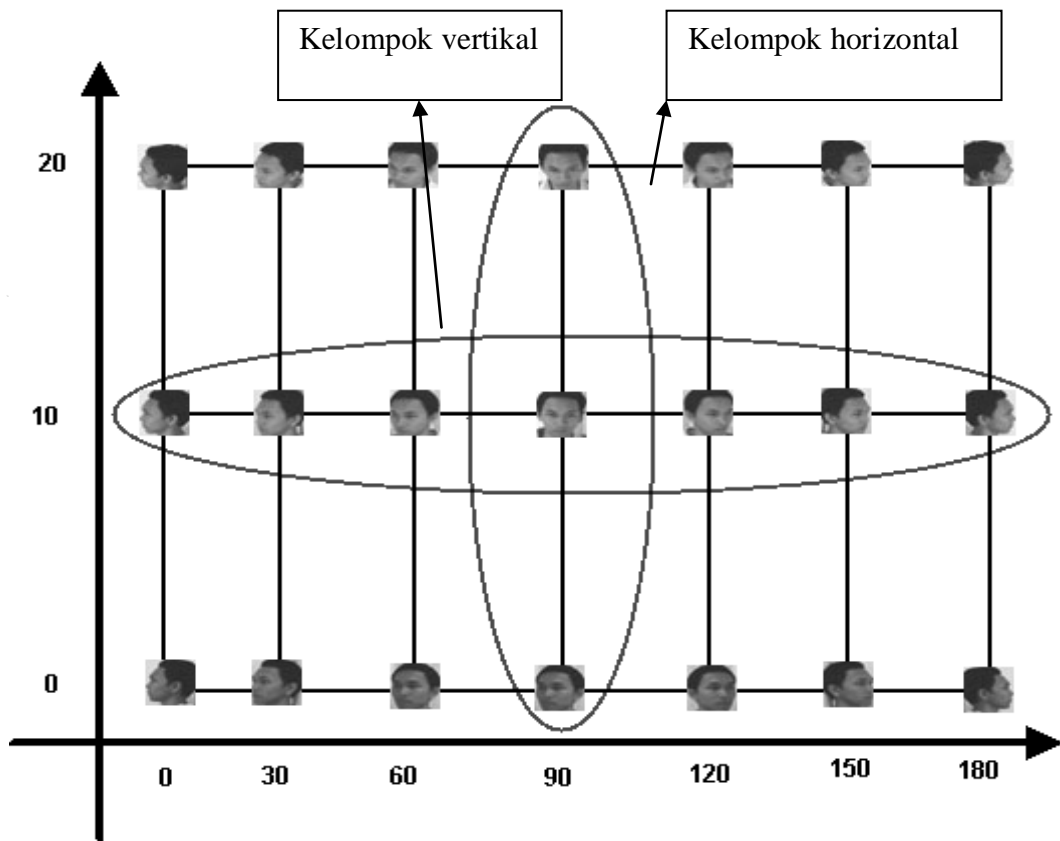
Kemudian setiap kelompok sudut, baik vertikal maupun horizontal ditransformasikan ke dalam ruang ciri dengan menggunakan Transformasi Karhunen-Loeve. Selanjutnya setiap kelompok sudut tersebut dihubungkan dengan menggunakan interpolasi spline atau linier dan dilakukan perhitungan terhadap titik-titik yang diujikan.



Gambar 3.6 Ilustrasi Pembentukan Ruang Ciri Berdasarkan Kelas Wajah dalam Data *Training* dan Transformasi Titik Uji ke Masing-masing Ruang Wajah



Gambar 3.7 Ilustrasi Titik Ciri dalam Ruang Wajah Adhit



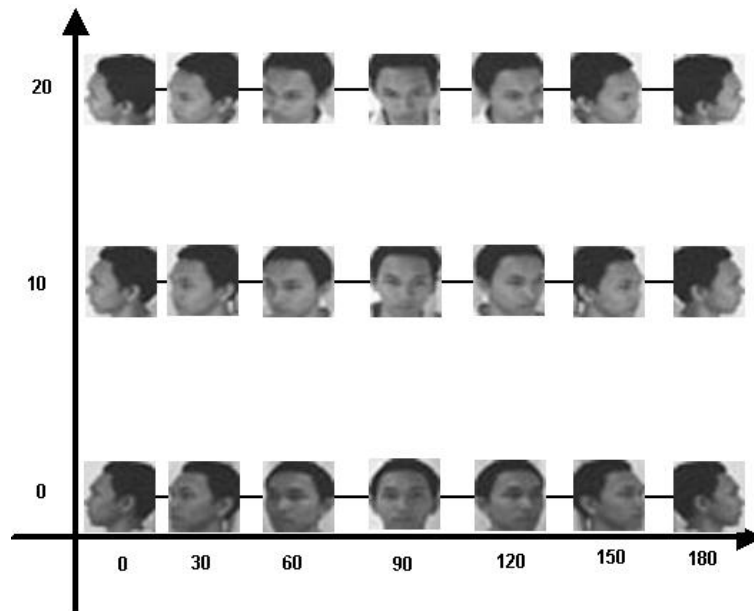
**Gambar 3.8 Ilustrasi Pengelompokan Data Berdasarkan Sudut Vertikal dan Horizontal yang Sama**

### 3.1.2 Pembentukan Ruang Ciri Berdasarkan Kelas Data Wajah Pelatihan dan Kelompok Sudut Vertikal atau Kelompok Sudut Horizontal

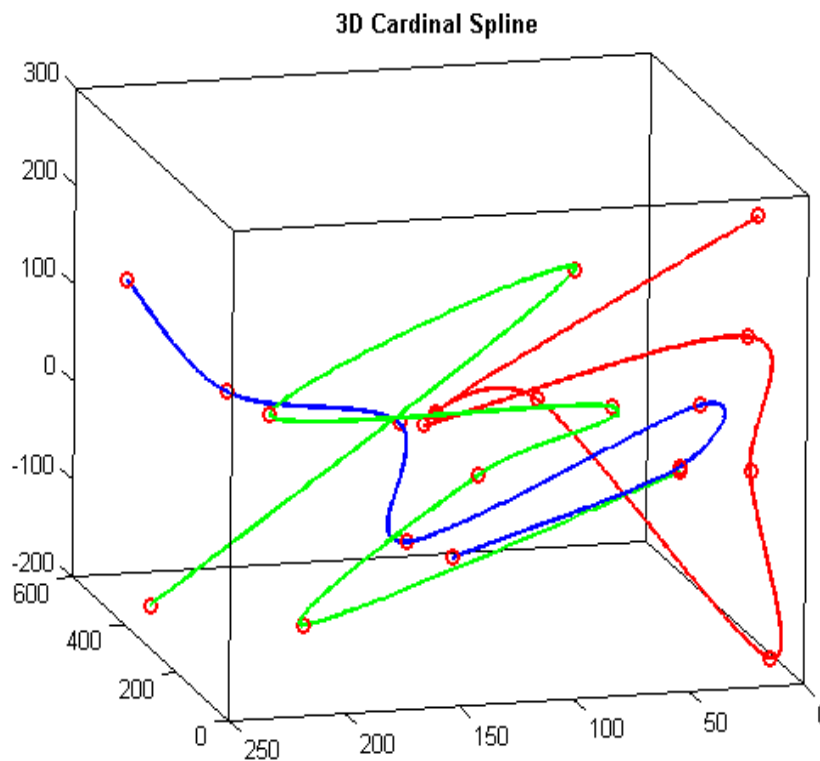
Berdasarkan penelitian Efrika [EFR04], pengurangan garis ciri yang tidak perlu dapat meningkatkan pengenalan sistem. Hal ini disebabkan karena semakin menurunnya kesalahan penebakan pada garis ciri yang salah. Berdasarkan penelitian tersebut penulis mencoba untuk memisahkan ruang ciri tidak saja berdasarkan kelas wajah dalam data *training*, namun juga memisahkan antara ruang ciri dengan kelompok sudut horizontal yang sama dan kelompok sudut vertikal yang sama. Ilustrasi pembentukan ruang ciri berdasarkan kelas wajah dan kelompok sudut vertikal atau horizontal dapat dilihat pada gambar 2.8 dan 2.9.

Dengan pembentukan ruang ciri dengan menggunakan metode ini, dalam satu ruang ciri hanya terdapat garis-garis ciri yang menghubungkan data-data yang mempunyai kelompok sudut horizontal atau vertikal yang sama saja. Dengan pendekatan ini diharapkan dapat meningkatkan keakuratan perhitungan dan dapat mengurangi kesalahan dalam menebak sudut antara sudut horizontal dan vertikal. Diagram pada gambar 2.10 dapat memperjelas pembentukan ruang ciri berdasarkan kelas wajah dan kelompok data vertikal atau horizontal.

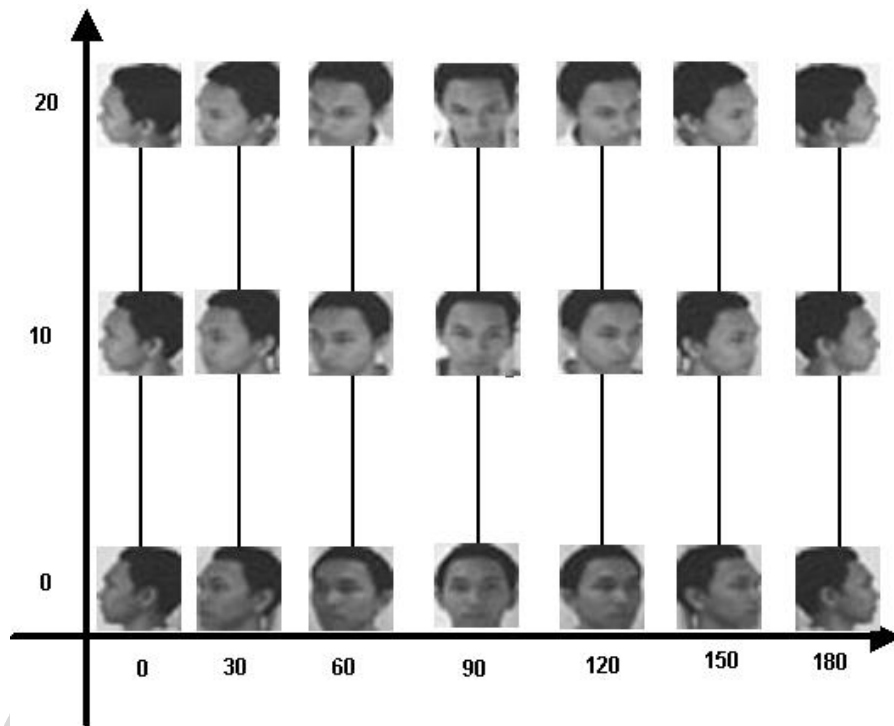




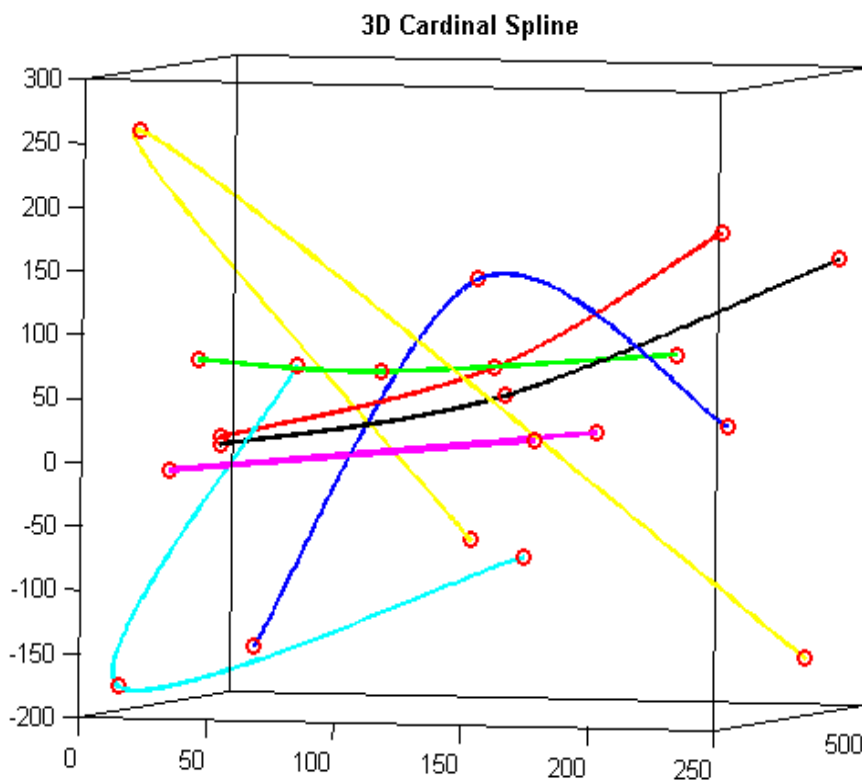
**Gambar 3.9 Ilustrasi Ruang Ciri Wajah dengan Kelompok Sudut Vertikal**



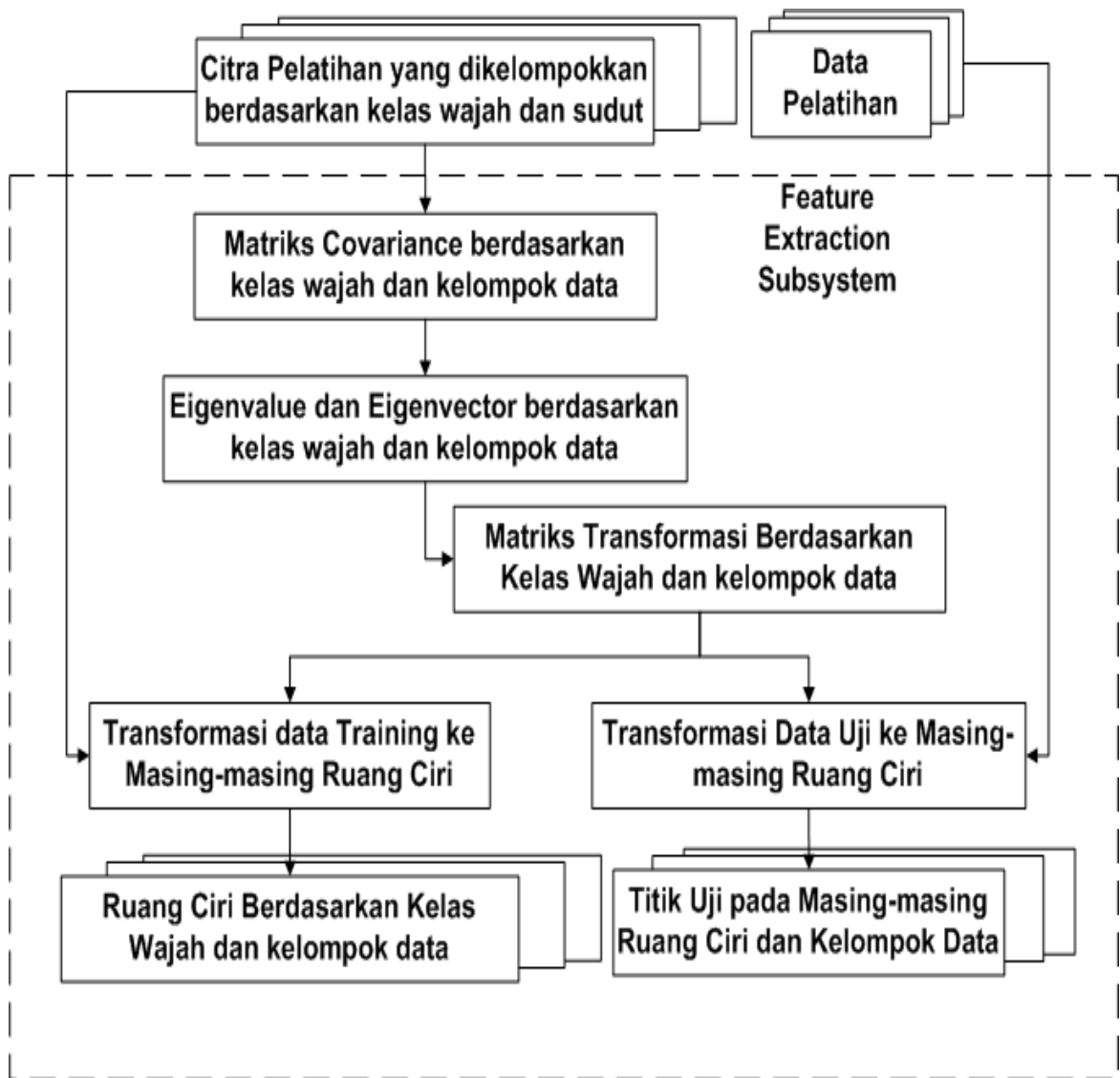
**Gambar 3.10 Ilustrasi Titik-titik Ciri Pada Kelompok Vertikal yang Dihubungkan Menggunakan Interpolasi Spline (Cardinal Spline)**



**Gambar 3.11 Ilustrasi Ruang Ciri Wajah dengan Kelompok Sudut Horizontal**



**Gambar 3.12 Ilustrasi Titik-titik Ciri Pada Kelompok Horizontal yang Dihubungkan Menggunakan Interpolasi Spline (Cardinal Spline)**

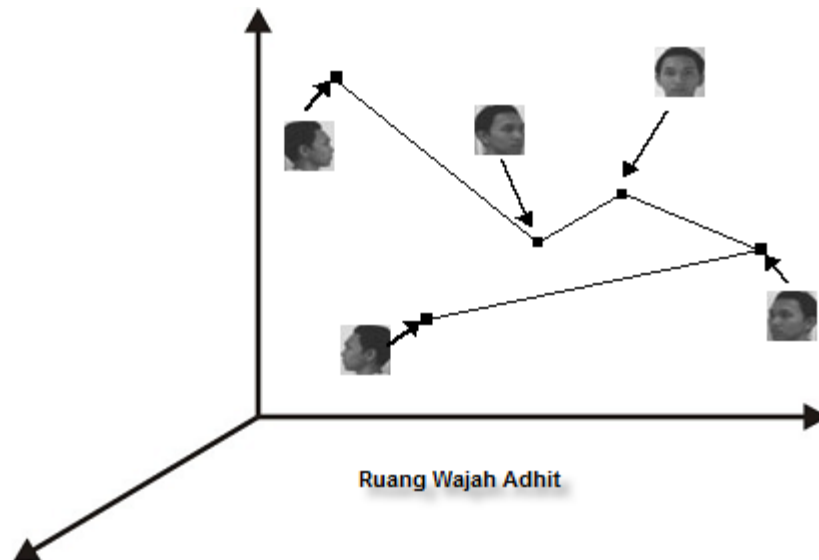


**Gambar 3.13 Diagram Pembentukan Ruang Wajah Berdasarkan Kelas Wajah dan Kelompok Sudut**

### 3.2 Klasifikasi Sudut Pandang

#### 3.2.1 Skema Klasifikasi Menggunakan Interpolasi Linier

Pada skema klasifikasi dengan menggunakan interpolasi linier ini, titik-titik ciri yang berada pada ruang eigen akan dihubungkan dengan menggunakan garis lurus. Hal ini dilakukan untuk mendapatkan generalisasi dari titik-titik ciri yang akan meningkatkan keakuratan dan tingkat pengenalan sistem.



**Gambar 3.14 Titik-titik Ciri yang Dihubungkan Menggunakan Interpolasi Linier pada Kelompok Vertikal yang Sama**

Kemudian dilakukan penghitungan jarak euclid dan dicari jarak euclid minimal antara titik uji yang telah ditransformasikan ke dalam ruang ciri dengan garis hasil interpolasi linier yang terbentuk di dalam ruang ciri.

Berikut ini adalah algoritma dari proses klasifikasi dengan menggunakan interpolasi linier:

1. Hubungkanlah setiap titik-titik pelatihan yang telah dikelompokkan berdasarkan sudut vertikal dan sudut horizontal yang sama dengan menggunakan interpolasi linier sehingga terbentuk garis sesuai dengan gambar 3.14. Garis yang terbentuk harus memenuhi persamaan dibawah ini

$$LX(t) = EXTr_i + (EXTr_{i+1} - EXTr_i) t$$

$$LY(t) = EYTr_i + (EYTr_{i+1} - EYTr_i) t$$

2. Hitung jarak vertikal dan horizontal antara titik uji dengan n titik yang telah di-generate dengan interpolasi linier dengan menggunakan jarak Euclid

$$XDl_i = \sqrt{\sum_{i=1}^n (LX(t)_i - EXTs_i)^2}$$

$$YDl_i = \sqrt{\sum_{i=1}^n (LY(t)_i - EYT_s_i)^2}$$

3. Dari semua jarak yang telah dihitung, carilah jarak horizontal dan vertikal minimum. Titik dimana mempunyai jarak horizontal dan vertikal yang minimum merupakan sudut tebakan dari sistem

$$XL_i = \text{Min} (XDl_i)$$

$$YL_i = \text{Min} (YDl_i)$$

### 3.2.2 Skema Klasifikasi Menggunakan Interpolasi Spline

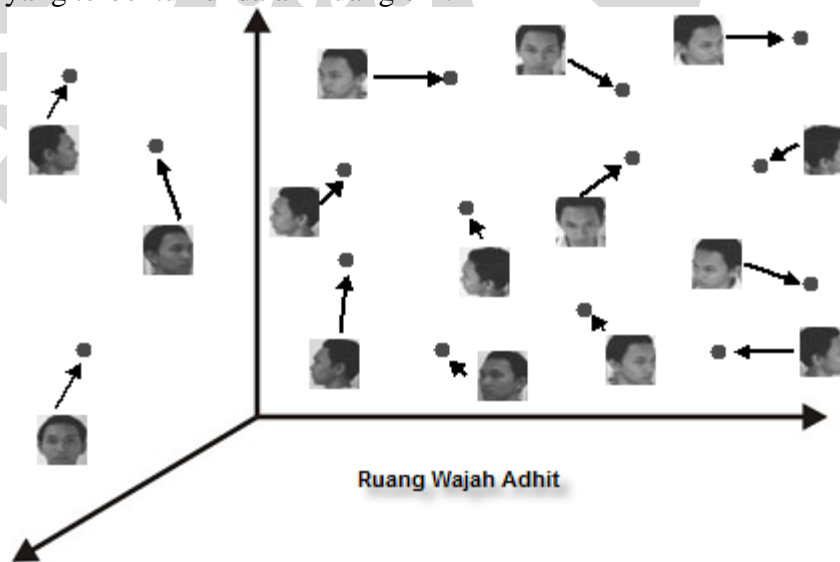
Pada skema sebelumnya, interpolasi linier digunakan untuk mengklasifikasikan suatu obyek. Informasi yang ada akan direpresentasikan ke dalam ruang eigen dalam bentuk titik-titik ciri. Untuk meningkatkan pengenalan sistem maka dilakukan generalisasi titik-titik ciri dengan membentuk garis (*feature line*). *Feature line* akan memberikan lebih banyak variasi perubahan ciri dari sebuah obyek.

Dalam metode klasifikasi dengan menggunakan interpolasi linier titik-titik ciri akan di hubungkan dengan menggunakan garis lurus. Garis yang terbentuk merupakan hasil dari titik-titik ciri yang saling dihubungkan, baik dari kelas obyek yang sama maupun berbeda. Selanjutnya dilakukan dengan cara mencari jarak minimum antara titik ciri uji dengan garis ciri.

Berbeda dengan metode diatas yang menggunakan garis lurus untuk menghubungkan titik-titik ciri yang ada dalam ruang eigen, pada penelitian kali ini penulis mencoba untuk menggunakan interpolasi spline untuk menghubungkan titik-titik ciri tersebut. Penelitian ini mencoba untuk membandingkan pengaruh jenis interpolasi antara interpolasi linier dengan interpolasi spline (Bezier spline dan Cardinal spline) dalam peningkatan pengenalan sistem penentu sudut pandang.

Garis dalam ruang ciri dibentuk dengan menghubungkan antara satu titik ciri dengan titik ciri lain pada kelas wajah yang sama dengan menggunakan interpolasi spline.

Kemudian dilakukan penghitungan jarak euclid dan dicari jarak euclid minimal antara titik uji yang telah ditransformasikan ke dalam ruang ciri garis spline yang terbentuk di dalam ruang ciri.



**Gambar 3.15 Ilustrasi Titik-titik ciri dalam Ruang Wajah Adith**

Berikut ini adalah klasifikasi suatu obyek citra dengan menggunakan Bezier spline. Algoritma dari proses klasifikasi dengan menggunakan Bezier spline adalah sebagai berikut:

1. Hubungkanlah setiap titik-titik pelatihan yang telah dikelompokkan berdasarkan sudut vertikal dan sudut horizontal yang sama dengan menggunakan Bezier spline sehingga terbentuk garis sesuai dengan gambar 3.16. Garis yang terbentuk harus memenuhi persamaan dibawah ini

$$BX(t)_i = (1-t)^2 EXTr_i + 2t(1-t) EXTr_{i+1} + t^2 EXTr_{i+2}$$

$$BY(t)_i = (1-t)^2 EYTr_i + 2t(1-t) EYTr_{i+1} + t^2 EYTr_{i+2}$$

2. Hitung jarak vertikal dan horizontal antara titik uji dengan n titik yang telah di-generate dengan interpolasi Bezier spline dengan menggunakan jarak Euclid

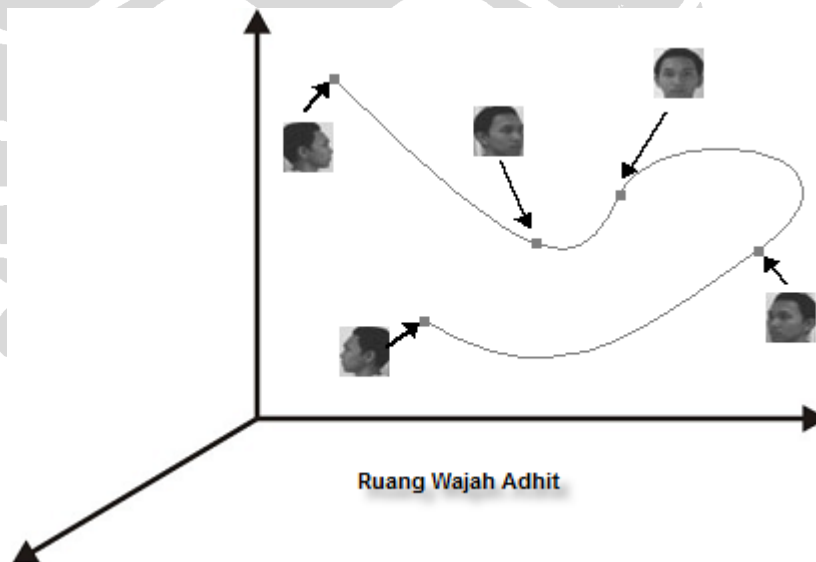
$$XDb_i = \sqrt{\sum_{i=1}^n (BX(t)_i - EXTs_i)^2}$$

$$YDb_i = \sqrt{\sum_{i=1}^n (BY(t)_i - EYTs_i)^2}$$

3. Dari semua jarak yang telah dihitung, carilah jarak horizontal dan vertikal minimum. Titik dimana mempunyai jarak horizontal dan vertikal yang minimum merupakan sudut tebakan dari sistem

$$XB_i = \text{Min} (XDb_i)$$

$$YB_i = \text{Min} (YDb_i)$$



**Gambar 3.16 Titik-titik Ciri yang Dihubungkan Menggunakan Interpolasi Spline pada Kelompok Vertikal yang Sama**

Sedangkan untuk klasifikasi dengan menggunakan Cardinal spline mempunyai sedikit perbedaan dalam cara menghubungkan titik-titik ciri pelatihan. Berikut ini adalah algoritma klasifikasi dengan menggunakan Cardinal spline:

1. Hubungkanlah setiap titik-titik pelatihan yang telah dikelompokkan berdasarkan sudut vertikal dan sudut horizontal yang sama dengan

menggunakan Cardinal spline sehingga terbentuk garis lengkung. Garis yang terbentuk harus memenuhi persamaan dibawah ini

$$CX(t)_i = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -c & 0 & c & 0 \\ 0 & -c & 0 & c \end{bmatrix} \begin{bmatrix} EXTr_{i-1} \\ EXTr_i \\ EXTr_{i+1} \\ EXTr_{i+2} \end{bmatrix} \begin{bmatrix} 2t^3 - 3t^2 + 1 \\ -2t^3 + 3t^2 \\ t^3 - 2t^2 + t \\ t^3 - t^2 \end{bmatrix}$$

$$CY(t)_i = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -c & 0 & c & 0 \\ 0 & -c & 0 & c \end{bmatrix} \begin{bmatrix} EYTr_{i-1} \\ EYTr_i \\ EYTr_{i+1} \\ EYTr_{i+2} \end{bmatrix} \begin{bmatrix} 2t^3 - 3t^2 + 1 \\ -2t^3 + 3t^2 \\ t^3 - 2t^2 + t \\ t^3 - t^2 \end{bmatrix}$$

2. Hitung jarak vertikal dan horizontal antara titik uji dengan n titik yang telah di-generate dengan interpolasi Cardinal spline dengan menggunakan jarak Euclid

$$XDc_i = \sqrt{\sum_{i=1}^n (CX(t)_i - EXTs_i)^2}$$

$$YDc_i = \sqrt{\sum_{i=1}^n (CY(t)_i - EYT_s_i)^2}$$

3. Dari semua jarak yang telah dihitung, carilah jarak horizontal dan vertikal minimum. Titik dimana mempunyai jarak horizontal dan vertikal yang minimum merupakan sudut tebakan dari sistem

$$XC_i = \text{Min} (XDc_i)$$

$$YC_i = \text{Min} (YDc_i)$$