

## **BAB 2**

### **PENGENALAN IRIS, PENENTUAN LOKASI IRIS, DAN PEMBUATAN VEKTOR MASUKAN**

Pengenalan suatu objek tentu saja tidak bisa dilakukan tanpa persiapan sama sekali. Ada beberapa proses yang perlu dilakukan sebelum suatu objek bisa digunakan untuk proses pengenalan. Bab ini akan membahas mengenai pengenalan iris secara umum dan beberapa proses yang termasuk di dalamnya. Beberapa proses tersebut diantaranya adalah penentuan lokasi iris dan pembuatan vektor masukan. Bab ini juga akan membahas mengenai *Principal Component Analysis* yang digunakan untuk mereduksi dimensi dari vector masukan.

#### **2.1 Pengenalan Iris**

Pengenalan iris bukan merupakan hal yang baru dalam biometrik. Sebelum penelitian ini dilakukan sudah banyak penelitian yang dilakukan mengenai pengenalan iris dan hasilnya cukup memuaskan. Beberapa hasil penelitian yang telah dilakukan sebelumnya bahkan ada yang telah menjadi sebuah sistem yang sudah dipatenkan dan dipakai di berbagai penjuru dunia. Penelitian tersebut dilakukan oleh John Daugman.

Berdasarkan literatur yang telah dibaca, ada beberapa tahap yang dilalui dalam pengenalan iris. Tahap tersebut berbeda-beda dalam setiap penelitian. Tetapi secara umum tahap-tahap tersebut adalah sebagai berikut

1. Pengumpulan data. Pada tahap ini, gambar-gambar iris yang akan digunakan untuk pengenalan dikumpulkan.
2. Penentuan lokasi iris. Gambar yang telah dikumpulkan pada tahap sebelumnya umumnya masih belum terfokus pada bagian iris saja. Gambar yang terkumpul biasanya adalah gambar mata. Oleh karena itu lokasi iris pada gambar mata tersebut perlu ditentukan terlebih dahulu.

3. Pembuatan vektor masukan. Setelah lokasi iris ditentukan, dibuat vektor masukan. Pembuatan vektor masukan dilakukan dengan mengambil nilai piksel dari gambar mata pada bagian yang telah iris yang telah ditentukan lokasinya.
4. Pengenalan. Tahap ini adalah tahap yang menentukan apakah suatu vektor masukan dikenali atau tidak.

Dalam penelitian ini sebenarnya ingin dilakukan pengumpulan data sendiri. Namun karena keterbatasan sumber daya yang ada, hal tersebut tidak dapat dilakukan. Mungkin untuk penelitian selanjutnya dapat dilakukan pengumpulan data sendiri.

Karena tidak dapat mengumpulkan data sendiri, maka dibutuhkan sumber data lain. Dari literatur yang telah dibaca, ditemukan beberapa sumber data iris yang bisa digunakan dalam penelitian ini. Beberapa diantaranya adalah basis data dari Universitas Palacky dan basis data CASIA. Setelah membandingkan beberapa sumber data tersebut, akhirnya diputuskan untuk menggunakan basis data CASIA. Keputusan ini diambil karena basis data CASIA banyak digunakan dalam penelitian-penelitian terdahulu dan basis data CASIA memiliki kualitas yang baik.

Basis data CASIA dibagi menjadi tiga, CASIA-IrisV3-Interval, CASIA-IrisV3-Lamp dan CASIA-IrisV3-Twins. Masing-masing bagian terdiri atas gambar iris dari beberapa orang. Masing-masing orang terbagi lagi menjadi dua bagian yaitu untuk mata kanan dan kiri.

Pada penelitian ini diambil gambar iris mata kiri dua puluh orang dari CASIA-IrisV3-Lamp. Gambar kedua puluh orang tersebut dipilih karena memiliki ukuran iris yang relatif sama. Hal ini memudahkan dalam proses lokalisasi iris dan pembuatan vektor masukan dari gambar iris.

## **2.2 Penentuan Lokasi Iris**

Dalam penelitian ini, data yang digunakan belum berupa gambar iris saja tetapi gambar mata secara keseluruhan. Karena itu untuk membuat vektor masukan perlu ditentukan lokasi iris pada setiap gambar. Penentuan lokasi iris ini dilakukan secara manual dengan dua metode.

### 2.2.1 Metode Cropping

Dengan metode ini, dari setiap gambar mata dibuat sebuah gambar baru yang berisi irisnya saja. Pada setiap gambar, bagian irisnya diukur secara manual. Setelah itu gambar dipotong pada bagian yang telah diukur sehingga didapatkan irisnya saja.

Metode ini memakan waktu yang cukup lama. Karena keterbatasan waktu yang dimiliki, gambar yang berhasil di-crop hanya untuk sepuluh kelas yang masing-masing terdiri atas dua puluh gambar.

### 2.2.2 Metode Titik Tengah

Untuk membuat vektor masukan, diperlukan titik tengah dari iris. Oleh karena itu maka sebenarnya proses *cropping* tidak diperlukan. Selain karena memakan waktu yang cukup lama, hasil gambar yang didapatkan juga kadang-kadang kurang baik.

Dengan metode titik tengah, tidak ada pemotongan sama sekali. Setiap gambar hanya dicari titik tengah irisnya lalu dengan informasi titik tengah tersebut dapat dibuat vektor masukannya.

## 2.3 Pembentukan Vektor Masukan

Sebelum masuk proses pengenalan, suatu gambar perlu diubah menjadi sebuah vektor masukan yang berdimensi satu. Metode yang biasa digunakan adalah dengan mengambil nilai setiap dari setiap piksel pada gambar lalu disusun sehingga membentuk suatu vektor satu dimensi.

Misalkan terdapat suatu gambar berukuran  $3 \times 3$  piksel. Untuk membentuk suatu vektor masukan dari gambar tersebut caranya adalah sebagai berikut.

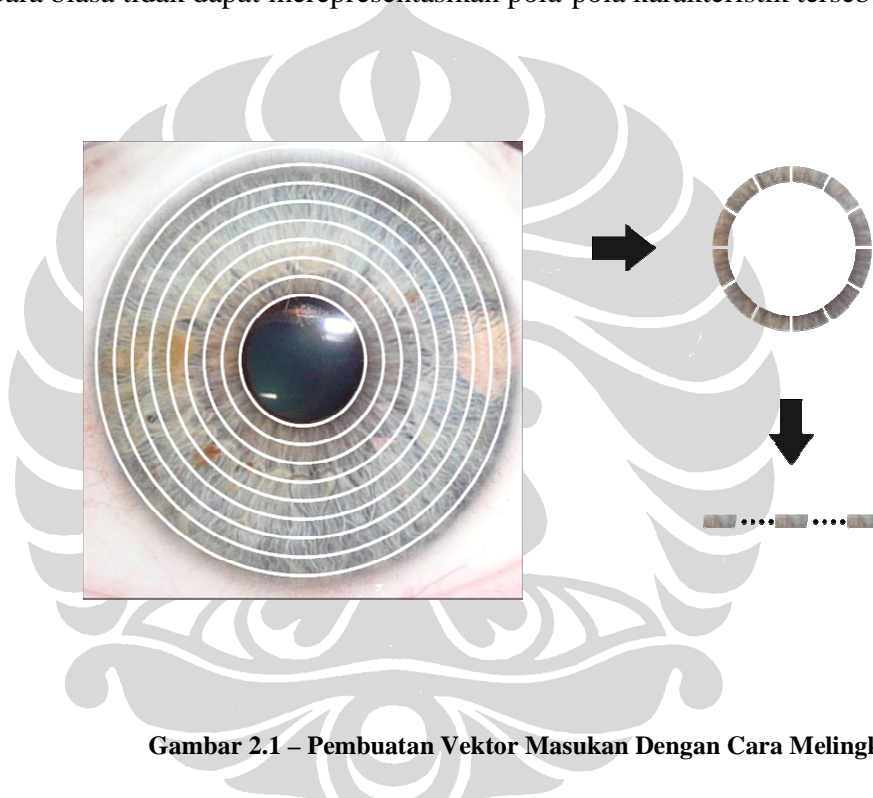
- Ambil nilai *grayvalue* untuk masing-masing piksel untuk membuat vektor  $G$ .

$$G = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix} \quad (2.1)$$

- Buat sebuah vektor satu dimensi dari  $G$  yaitu  $A$ .

$$A = [p_{11}, p_{12}, p_{13}, p_{21}, p_{22}, p_{23}, p_{31}, p_{32}, p_{33}] \quad (2.2)$$

Untuk iris, vektor masukan yang dibentuk dengan cara biasa tersebut akan menghasilkan tingkat pengenalan yang kurang baik. Tingkat pengenalan yang kurang baik ini disebabkan oleh bentuk iris yang berupa lingkaran sehingga pola-pola karakteristik yang digunakan untuk membedakan antara iris juga berbentuk lingkaran. Cara biasa tidak dapat merepresentasikan pola-pola karakteristik tersebut.



**Gambar 2.1 – Pembuatan Vektor Masukan Dengan Cara Melingkar**

Agar vektor masukan yang dibentuk dapat merepresentasikan pola-pola tersebut maka perlu dilakukan pengambilan nilai grayvalue untuk setiap piksel dengan cara melingkar. Dalam penelitian ini vektor masukan dibentuk dengan memetakan suatu gambar iris ke dalam koordinat polar lalu dipetakan lagi ke koordinat kartesius sehingga menghasilkan sebuah vektor dua dimensi.

Pertama-tama perlu ditentukan titik pusat dari iris, titik tersebut berada di pupil. Titik pusat dari iris tersebut adalah titik pusat dari koordinat polar ( $r = 0, \theta = 0$ ) yang akan digunakan untuk mengambil nilai *grayvalue* setiap piksel pada iris secara melingkar.

Kemudian tentukan radius iris  $R$  dan radius pupil  $r$ . Radius pupil perlu ditentukan karena vektor masukan hanya akan dibuat dari iris saja. Karakteristik yang menjadi pembeda antar kelas tidak terdapat pada pupil maka bagian pupil dapat dihilangkan untuk memaksimalkan pengenalan.

Besarnya radius iris dan pupil menentukan banyaknya lingkaran yang diperlukan untuk membuat vektor masukan. Besarnya radius pupil pada masing-masing gambar berbeda-beda sehingga vektor masukan dibuat dengan radius pupil yang berbeda-beda juga. Radius pupil yang digunakan dalam penelitian ini adalah 30, 40, 50, 60, dan 70 piksel. Radius pupil ini didapatkan berdasarkan pengukuran langsung pada beberapa gambar. Banyaknya radius pupil yang digunakan ditujukan untuk mengetahui seberapa besar radius pupil dapat mempengaruhi tingkat pengenalan iris.

Untuk setiap nilai  $i = [r, 2, \dots, R]$  buat sebuah vektor  $g_i = [p_{ij}, p_{ij}, \dots, p_{ij}]$ ,  $p_{ij}$  adalah nilai grayvalue dari piksel pada koordinat  $(i \cos(j), i \sin(j))$  dengan  $j = [0, \Delta\theta, 2\Delta\theta, \dots, n\Delta\theta], n\Delta\theta = 360$ .

Vektor-vektor  $g_i$  membentuk sebuah vektor  $G$  seperti pada 2.3.

$$G = \begin{bmatrix} g_i \\ \vdots \\ g_R \end{bmatrix} \quad (2.3)$$

$$G = \begin{bmatrix} p_{1,0} & \cdots & p_{1,360} \\ \vdots & \vdots & \vdots \\ p_{R,0} & \cdots & p_{R,360} \end{bmatrix} \quad (2.4)$$

Ukuran vektor ini dapat dihitung dengan persamaan 2.5.

$$R \times \frac{360}{\Delta\theta} \quad (2.5)$$

Vektor ini kemudian dibentuk menjadi sebuah vektor masukan satu dimensi  $A$  seperti pada 2.6.

$$A = [p_{1,0}, \dots, p_{1,360}, \dots, p_{R,0}, \dots, p_{R,360}] \quad (2.6)$$

Dalam penelitian ini penentuan lokasi iris pada gambar tidak dilakukan secara otomatis. Karena itu untuk memudahkan proses pembuatan serta menyeragamkan ukuran vektor masukan penulis memilih data yang memiliki radius yang sama. Proses pemilihan data tersebut dilakukan secara manual, yaitu dengan mengukur gambar satu persatu. Dari hasil pengukuran tersebut didapatkan radius rata-rata adalah 90 piksel maka data yang digunakan untuk penelitian adalah data dengan radius 90 piksel

Untuk membuat vektor masukan, sudut putar  $\Delta\theta$  yang semakin kecil tentu akan menghasilkan vektor masukan yang lebih baik. Tetapi dimensi vektor masukan yang didapatkan juga akan semakin besar. Vektor masukan dengan dimensi besar akan mempengaruhi proses pengenalan sehingga dimensinya harus diperkecil terlebih dahulu. Proses memperkecil dimensi ini dilakukan dengan *Principal Component Analysis* (PCA).

Secara teori, besarnya dimensi vektor masukan tidak berpengaruh terhadap proses PCA. Tetapi dalam penelitian ini, mesin yang digunakan memiliki daya komputasi yang terbatas, sehingga proses PCA hanya bisa dijalankan pada vektor masukan yang berdimensi kurang lebih 3600.

Karena keterbatasan tersebut maka vektor masukan tidak bisa dibuat dengan sudut putar yang paling kecil yaitu satu derajat. Ukuran sudut putar menjadi bergantung kepada radius pupil. Untuk radius pupil 30, 40, 50, 60, dan 70 piksel sudut putar yang digunakan adalah 6, 5, 4, 3, dan 2 derajat. Penggunaan sudut putar tersebut akan menghasilkan vektor masukan dengan dimensi 3600 sehingga dapat dikecilkan dengan PCA.

## 2.4 Principal Component Analysis

Banyak penelitian menemui keadaan dimana data yang diteliti dimensinya terlalu besar dan rumit. Pada beberapa penelitian, sifat data seperti ini menyebabkan percobaan menjadi tidak efisien dan hasilnya juga sering kurang memuaskan. Penelitian ini juga mengalami masalah tersebut. Vektor masukan yang dibuat berdimensi cukup besar sehingga perlu diperkecil dimensinya. Untuk mengatasi masalah ini ada banyak metode yang dapat digunakan. Salah satunya adalah dengan menggunakan *Principle Component Analysis*.

*Principle Component Analysis* (PCA) adalah suatu metode untuk mengambil informasi terpenting dari sekumpulan data yang terlalu besar dan rumit. Hal ini dapat diwujudkan dengan mereduksi dimensi data tersebut dengan cara mentransformasinya menjadi sekumpulan vektor ortogonal yang mewakili distribusinya.

Misalkan terdapat sekumpulan data gambar  $X = [x_1, x_2, \dots, x_M]$ .  $X$  berukuran  $M \times N$  dimana  $M$  adalah banyaknya gambar dan  $N$  adalah dimensi data (banyaknya piksel yang terdapat dalam gambar). Nilai  $N$  yang terlalu besar akan menyebabkan pemrosesan data menjadi kurang maksimal. PCA dapat mengatasi masalah ini dengan mentransformasi data tersebut sehingga ukurannya menjadi  $M \times n$ , dimana nilai  $n$  lebih kecil dari  $N$ .

Metode PCA yang umum digunakan adalah dengan mencari matriks kovarian dari data. Untuk data  $X$  dapat dicari matriks kovariannya dengan langkah-langkah sebagai berikut.

1. Buat vektor rata-rata dari  $X$  yaitu  $x_r$ .
2. Kurangi  $X$  dengan rata-ratanya ( $x_r$ ) seperti pada persamaan 2.7, sehingga didapatkan  $X'$

$$X' = [(x_1 - x_r), (x_2 - x_r), \dots, (x_M - x_r)] \quad (2.7)$$

$$X' = [x'_1, x'_2, \dots, x'_M] \quad (2.8)$$

3. Matriks kovariannya dapat dihitung dengan persamaan 2.9

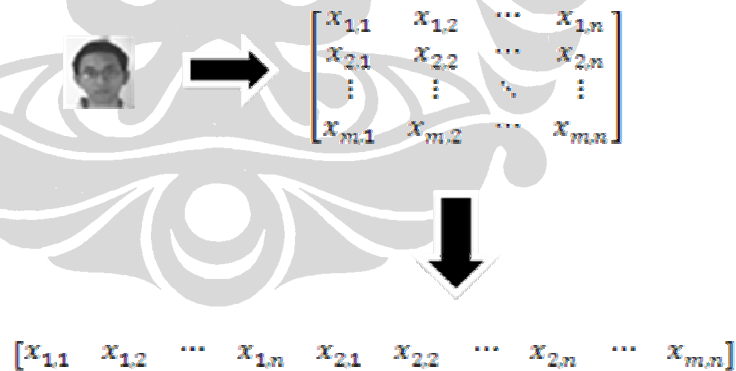
$$C = \sum_{i=1}^M x_i x_i^T = X' X'^T \quad (2.9)$$

Langkah selanjutnya adalah mencari vektor eigen dan nilai eigen dari matriks kovarian  $C$ . Vektor eigen lalu diurutkan secara menurun sesuai dengan nilai eigennya. Setelah itu vektor eigen yang nilai eigennya kecil (tidak terlalu signifikan) dihilangkan. Vektor eigen sisanya menjadi matriks transformasi yang akan digunakan untuk mentransformasi data sehingga data tereduksi dimensinya.

Pada penelitian ini, PCA dilakukan terhadap data yang telah standar. Pembuatan data menjadi standar dilakukan dengan  $z$ -score.

PCA diimplementasikan melalui program Matlab. program Matlab ini dibuat oleh M Irvan Ginandjar. Algoritma yang digunakan adalah sebagai berikut (Irvan Ginandjar, 2009)

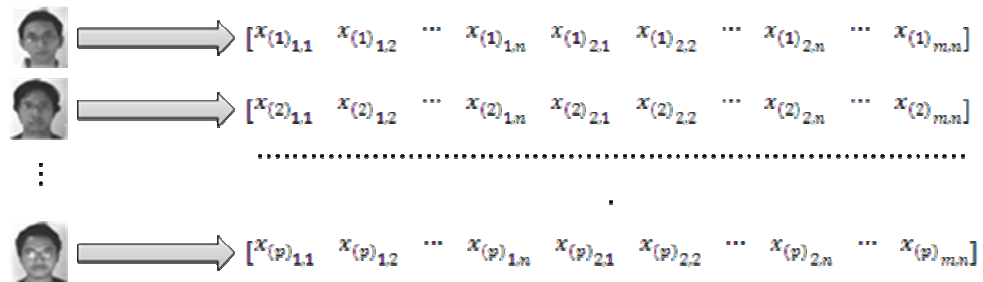
- Susun data acuan. Misal gambar berukuran  $m \times n$ , susun data tersebut menjadi vektor sepanjang  $m \times n$  seperti yang ditunjukkan oleh Gambar 2.2.



Gambar 2.2 – Pengubahan dari Gambar menjadi Vektor

Begitu pula dengan gambar selanjutnya, kemudian susun dibawah data gambar sebelumnya seperti yang dapat dilihat pada Gambar 2.3.

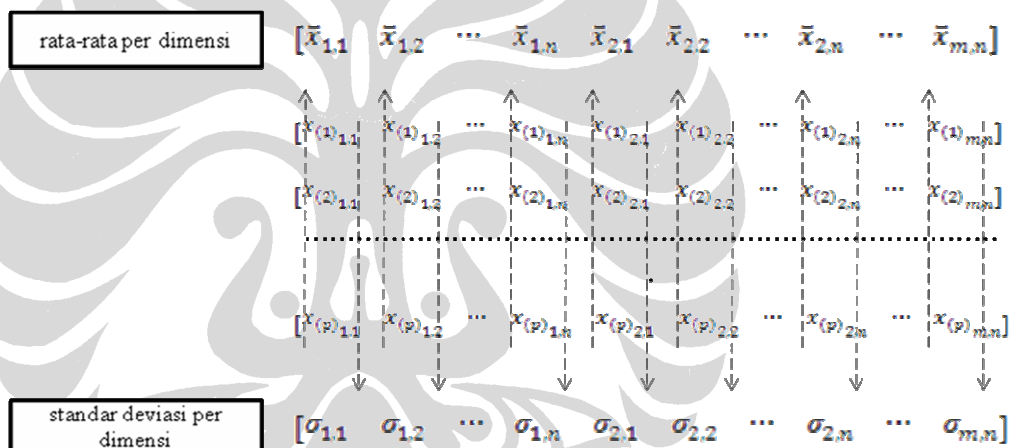




Gambar 2.3 – Penyusunan Vektor

Variabel  $p$  adalah banyaknya data.

- Cari rata-rata dan standar deviasi masing-masing dimensi. Gambar 2.4 menjelaskan tentang pengambilan nilai rata-rata dan standar deviasi masing-masing dimensi:



Gambar 2.4 – Pengambilan Rata-Rata dan Standar Deviasi masing-Masing Dimensi

Hal ini dilakukan untuk menormalisasi data pada tahap selanjutnya.

- Normalisasi data dengan ( $z$ -score). Normalisasi masing-masing elemen matriks dengan menggunakan informasi rata-rata dan standar deviasi tiap dimensi dengan Persamaan 2.10

$$z^{(p)}_{m,n} = \frac{x^{(p)}_{m,n} - \bar{x}_{m,n}}{\sigma_{m,n}} \quad (2.10)$$

- Cari kovarian dari data normal. Cari kovarian dengan aturan pada Persamaan 2.11:

$$cov(X, Y) = \frac{\sum_{i=1}^k (X_i - \bar{X})(Y_i - \bar{Y})}{(k - 1)} \quad (2.11)$$

$$C^{n \times n} = (c_{i,j}, c_{i,j} = cov(Dim_i, Dim_j))$$

$$covMat = \begin{bmatrix} cov(D_1, D_1) & cov(D_1, D_2) & \dots & cov(D_1, D_{n^2}) \\ cov(D_2, D_1) & cov(D_2, D_2) & \dots & cov(D_2, D_{n^2}) \\ \vdots & \vdots & \ddots & \vdots \\ cov(D_{n^2}, D_1) & cov(D_{n^2}, D_2) & \dots & cov(D_{n^2}, D_{n^2}) \end{bmatrix}$$

- Hitung nilai Eigen dan vektor Eigen dari masing-masing hasil kovarian. Hitung nilai Eigen dengan Persamaan 2.12 berikut:

$$\det \begin{bmatrix} c_{1,1} - \lambda & c_{1,2} & \dots & c_{1,n^2} \\ c_{2,1} & c_{2,2} - \lambda & \dots & c_{2,n^2} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n^2,1} & c_{n^2,2} & \dots & c_{n^2,n^2} - \lambda \end{bmatrix} = 0 \quad (2.12)$$

$$\longrightarrow \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_{n^2} \end{bmatrix}$$

Ambil diagonal dari matriks nilai Eigen.

$$eigVal = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_{n^2} \end{bmatrix} \quad (2.7)$$

Hitung vektor Eigen mengikuti aturan pada Persamaan 2.8:

$$\begin{bmatrix} c_{1,1} & c_{1,2} & \dots & c_{1,n^2} \\ c_{2,1} & c_{2,2} & \dots & c_{2,n^2} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n^2,1} & c_{n^2,2} & \dots & c_{n^2,n^2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n^2} \end{bmatrix} = \lambda_i \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n^2} \end{bmatrix}, eig_i = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n^2} \end{bmatrix} \quad (2.8)$$

$$eigMat = (eig_1, eig_2, \dots, eig_{n^2})$$

Transpos vektor Eigen, pasangkan vektor Eigen dengan nilai Eigen yang sesuai. Lalu urutkan vektor Eigen sesuai dengan nilai Eigen-nya dengan yang terbesar di atas.

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{bmatrix} \rightarrow \begin{bmatrix} x_{1,1} & x_{2,1} & \dots & x_{n,1} \\ x_{1,2} & x_{2,2} & \dots & x_{n,2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1,m} & x_{2,m} & \dots & x_{n,m} \end{bmatrix} = \begin{bmatrix} \lambda_1 & x_{1,1} & x_{2,1} & \dots & x_{n,1} \\ \lambda_2 & x_{1,2} & x_{2,2} & \dots & x_{n,2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \lambda_n & x_{1,m} & x_{2,m} & \dots & x_{n,m} \end{bmatrix} \downarrow \text{urutkan}$$

Gambar 2.5 – Pemasangan Vektor Eigen dengan Nilai Eigen yang Sesuai lalu Diurutkan Sesuai Nilai Eigen

- Misal jumlah data adalah  $m \times n = i$  dan dimensi adalah  $p$ . Untuk memotong dimensi, ambil data dari vektor teratas sepanjang  $k$  ( $0 < k < p$ ) dari vektor Eigen yang sudah terurut. Maka akan didapatkan matriks berukuran  $i \times k$  yang merupakan matriks transformasi.
- Kalikan data acuan yang sudah dinormalisasi dengan matriks transformasi, data baru adalah data hasil perkalian tersebut.
- Sementara itu data uji langsung dikalikan dengan matriks transformasi yang sesuai.

Dalam penelitian ini, dimensi vektor masukan yang besarnya 3600 dicecilkan menjadi 50. Angka 50 tersebut didapatkan dari hasil pengamatan pada grafik nilai eigen terhadap dimensi vektor masukan yang dihasilkan oleh proses PCA.