

## Visualisasi Data Iris Menggunakan Analisis Komponen Utama dan Analisis Komponen Utama Kernel

### *Visualization of Iris Data Using Principal Component Analysis and Kernel Principal Component Analysis*

<sup>1</sup>Ismail Djakaria, <sup>2</sup>Suryo Guritno, <sup>2</sup>Sri Haryatmi Kartiko  
<sup>1</sup>Mathematics Department, Gorontalo State University, Gorontalo  
<sup>2</sup>Mathematics Department, Gadjah Mada University, Yogyakarta

#### ABSTRACT

Principal component analysis (PCA) is a method used to reduce dimensionality of the dataset. However, the use of PCA failed to carry out the problem of non-linear and non-separable data. To overcome this problem such data is more appropriate to use PCA method with the kernel function, which is known as the kernel PCA (KPCA). In this paper, Iris dataset visualized with PCA and KPCA, that contains are the length and the width of sepal and petal.

Keywords : KPCA, non-linear, non-separable, Iris flower dataset

#### PENDAHULUAN

Untuk menghindari adanya permasalahan, seperti kemungkinan hilangnya beberapa informasi hubungan antar variabel terikat dan gagalnya pengawasan terhadap beberapa hipotesis yang melibatkan beberapa variabel yang diperlukan, maka analisis multivariat dianggap sangat berperan (Darlington 1997; Johnson & Dean 2002). Salah satu metode yang ada dalam analisis multivariat adalah *principal component analysis* (PCA), di samping metode lain seperti analisis diskriminan, analisis faktor, *multidimensional scaling*, dan lain-lain.

*Principal component analysis* (PCA) telah dipelajari sejak awal abad 20, Pearson tahun 1901 dan Hottelling tahun 1935 telah mempelajarinya melalui metode komputasi praktis (Tipping & Bishop 1999). PCA merupakan suatu metode yang banyak digunakan untuk mereduksi sejumlah dimensi, misalkan  $p$ , dari sebuah *dataset* (variabel), menjadi  $k$  variabel baru, dengan  $k \leq p$ . Setiap  $k$  variabel baru hasil reduksi merupakan kombinasi linier dari  $p$  variabel asal, dengan variansi yang dimiliki oleh  $p$  variabel asal, sebagian besar dapat diterangkan/dimiliki oleh  $k$  variabel baru. PCA akan cukup efektif jika antar variabel asal memiliki korelasi yang cukup tinggi. Tujuan PCA adalah mendapatkan vektor yang menyatakan kembali *dataset noisy*

yang sangat berarti. PCA berasumsi bahwa transformasi dapat dilakukan secara linier. Di sisi lain, PCA merupakan suatu transformasi ortogonal dari sistem koordinat dimana data diuraikan. Nilai koordinat baru yang merepresentasikan data disebut *principal component* (Schölkopf *et al.* 1996). Sedangkan menurut Xuejie (2006) dan Min (2006), PCA adalah suatu metode non-parametrik yang menggunakan aljabar linier untuk mengekstrak informasi atau pola yang relevan dari sebuah *dataset* dengan atribut multivariat.

Aplikasi PCA banyak ditemukan di sejumlah bidang, di antaranya, Raychaudhuri, *et al* (2000) menyelidiki penerapan PCA dalam meringkas eksperimen *microarray*, dengan melihat runtun waktu sporulasi. Contoh lain penggunaan PCA pada data berdimensi lebih tinggi, khususnya data *microarray*, dapat dilihat dalam Alter *et al.* (2000).

Tujuan utama PCA (Härdle & Simar 2003) adalah mereduksi dimensi observasi, dengan mengambil sebuah elemen vektor terobservasi dan membuang yang lainnya. PCA, juga dikatakan transformasi Karhunen-Loève atau transformasi Hotelling, merupakan sebuah teknik untuk menyederhanakan suatu *dataset*, dengan mereduksi dimensi *dataset* menjadi dimensi yang lebih kecil untuk analisis dengan tetap mempertahankan sebanyak mungkin tampilan variasi dalam *dataset* asal (Lin 2006). Misalkan, vektor random  $\mathbf{X}' = [\mathbf{X}_1, \dots, \mathbf{X}_p]$

berdistribusi normal multivariat dengan mean  $\mu$  dan matriks kovariansi  $\Sigma$ , serta akar karakteristik  $\lambda_1 \geq \dots \geq \lambda_p \geq 0$ . *Principal component* (PC) adalah kombinasi linier dari  $p$  variabel yang diobservasi, atau dapat ditulis

$$\mathbf{Y} = \mathbf{A}\mathbf{X},$$

dengan  $\mathbf{Y}' = (\mathbf{Y}_1, \dots, \mathbf{Y}_p)$ ,  $\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1p} \\ a_{21} & a_{22} & \dots & a_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{p1} & a_{p2} & \dots & a_{pp} \end{bmatrix}$ , dan  $\mathbf{X}' = (\mathbf{X}_1, \dots, \mathbf{X}_p)$ .

Sehingga masing-masing PC dapat ditulis sebagai

$$\mathbf{Y}_1 = a_{11}\mathbf{X}_1 + a_{21}\mathbf{X}_2 + \dots + a_{p1}\mathbf{X}_p = \mathbf{a}'_1\mathbf{X}$$

$$\mathbf{Y}_2 = a_{12}\mathbf{X}_1 + a_{22}\mathbf{X}_2 + \dots + a_{p2}\mathbf{X}_p = \mathbf{a}'_2\mathbf{X}$$

⋮

$$\mathbf{Y}_p = a_{1p}\mathbf{X}_1 + a_{2p}\mathbf{X}_2 + \dots + a_{pp}\mathbf{X}_p = \mathbf{a}'_p\mathbf{X}$$

dengan  $\text{Var}(\mathbf{Y}_i) = \mathbf{a}'_i\mathbf{S}\mathbf{a}_i$  dan  $\text{Cov}(\mathbf{Y}_i, \mathbf{Y}_k) = \mathbf{a}'_i\mathbf{S}\mathbf{a}_k$ .

Vektor koefisien dari PC1 dipilih sedemikian rupa sehingga  $\text{Var}(\mathbf{Y}_1)$  maksimum di antara vektor koefisien yang lain. Selanjutnya, PC2 adalah kombinasi linier dari variabel terobservasi yang bersifat ortogonal terhadap PC1 dan memiliki variansi yang minimum dari variansi sebelumnya dan maksimum dalam PC2. Jadi, PCA juga merupakan metode untuk mendapatkan koefisien-koefisien dari himpunan kombinasi linier yang memenuhi:

- ada tepat  $p$  PC, yaitu sebanyak variabel yang diobservasi, dan setiap PC merupakan kombinasi linier variabel-variabel tersebut;
- setiap PC saling ortogonal, saling bebas, yang berarti bahwa koefisien-koefisiennya bersifat ortogonal, dan skor komponennya tidak berkorelasi;
- PC diekstrak dengan urutan yang semakin kecil.

Salah satu cara mendapatkan koefisien seluruh PC secara simultan adalah dengan melakukan dekomposisi akar karakteristik-vektor karakteristik terhadap matriks korelasi antara variabel yang diobservasi. Cara yang lain adalah melakukan dekomposisi nilai singular (SVD) terhadap matriks data yang berukuran  $n \times p$ . Akar karakteristik adalah variansi, sedangkan vektor karakteristik adalah koefisien-koefisien PC.

Meskipun PCA merupakan suatu metode yang sangat baik untuk mereduksi sejumlah dimensi variabel dan sangat baik pula untuk menemukan *feature*, namun ada beberapa keterbatasan-keterbatasan, antara lain:

- Berdasarkan asumsi bahwa PCA hanya dapat mengatasi masalah hubungan antara setiap data yang linier, sedangkan kenyataannya dalam dalam setiap situasi hubungan data ada yang *non-linear* dan *non-separable*.
- PCA hanya memperhatikan mean dan variansi data, hal ini berarti bahwa PCA tidak memandang data berdistribusi probabilitas eksponensial (misalnya Gaussian).
- Principal component* tidak diurutkan dari variansi besar ke variansi yang lebih kecil, tetapi variansi yang paling besar tidak merepresentasikan hal yang penting untuk kluster data.

Keterbatasan yang lebih menonjol adalah PCA tidak dapat digunakan untuk data *non-linear* dan *non-separable*, sehingga perlu dilakukan transformasi terhadap semua input data ke ruang *feature*.

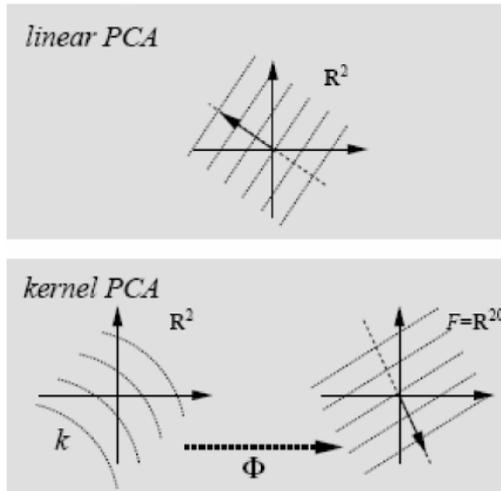
Misalkan  $\mathbf{x}_i \in \mathbb{R}^n$ ,  $i = 1, 2, \dots, m$  merupakan input data. Matriks kovariansi untuk input data tersebut di ruang  $\mathbb{R}^n$  adalah

$$\mathbf{A} = \frac{1}{m} \sum_{i=1}^m \mathbf{x}_i \mathbf{x}'_i.$$

Berdasarkan prinsip kerja dari PCA, untuk menyelesaikan masalah klasifikasi pola sama artinya dengan menyelesaikan masalah *eigen-value*  $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$ , untuk *eigen value*  $\lambda \geq 0$  dan *eigen vector*  $\mathbf{v} \in \mathbb{R}^n \setminus \{0\}$ . Matriks kovariansi umumnya tidak diketahui, untuk mendapatkannya, digunakan estimasi berdasarkan input data. Hal ini dapat dilakukan jika input data terdiri dari data linier. Sedangkan untuk data *non-linear* dan *non-separable*, PCA tidak dapat digunakan. Salah satu cara agar PCA dapat digunakan adalah dengan melakukan transformasi semua input data ke ruang *feature*. Maka untuk mengatasi masalah ini muncul metode kernel PCA (KPCA). Kernel PCA merupakan PCA yang diaplikasikan pada input data yang telah ditransformasikan ke ruang *feature*. Misalkan  $\Phi : \mathbb{R}^n \rightarrow F$  fungsi yang memetakan semua input data  $\mathbf{x}_i \in \mathbb{R}^n$ , berlaku  $\Phi(\mathbf{x}_i) \in F$ . Berdasarkan transformasi ini, terlihat bahwa ruang *feature* dibangun oleh vektor-vektor  $\{\Phi(\mathbf{x}_1), \Phi(\mathbf{x}_2), \dots, \Phi(\mathbf{x}_m)\}$ . Sehingga semua

vektor di ruang *feature* dapat dinyatakan sebagai kombinasi linier dari vektor-vektor  $\{\Phi(x_1), \Phi(x_2), \dots, \Phi(x_m)\}$ . Dengan demikian, matriks kovariansi di ruang *feature* untuk vektor  $\{\Phi(x_1), \Phi(x_2), \dots, \Phi(x_m)\}$  dapat dituliskan sebagai

$$C = \frac{1}{m} \sum_{j=1}^m \Phi(x_j) \Phi(x_j)' \quad (1)$$



Gambar 1. Ide Dasar KPCA (Schölkopf et al. 1996).

dan masalah *eigen-value* di ruang *feature*  $F$  dapat dinyatakan sebagai:

$$\lambda \mathbf{v} = C \mathbf{v} \quad (2)$$

di mana  $\mathbf{v}$  *eigen vector* dari  $C$  dalam ruang  $F$ . Jadi, dengan argumen yang sama, Persamaan (2) ekuivalen dengan,

$$\lambda \langle \Phi(x_k), \mathbf{v} \rangle = \langle \Phi(x_k), C \mathbf{v} \rangle, \forall k = 1, 2, \dots, m$$

Karena semua vektor di ruang *feature*  $F$  dapat dinyatakan sebagai kombinasi linier dari vektor-vektor  $\{\Phi(x_1), \Phi(x_2), \dots, \Phi(x_m)\}$ , maka *eigen-vector*, yang merupakan solusi dari masalah *eigen-value*  $\lambda \mathbf{v} = C \mathbf{v}$ , juga dapat dinyatakan sebagai kombinasi linier dari kombinasi linier vektor-vektor  $\{\Phi(x_1), \Phi(x_2), \dots, \Phi(x_m)\}$ . Hal ini berarti, ada konstanta-konstanta  $\alpha_i$  di mana  $i = 1, 2, \dots, m$ , sedemikian sehingga

$$\mathbf{v} = \sum_{i=1}^m \alpha_i \Phi(x_i) \quad (4)$$

Dengan mensubstitusi Persamaan (1) dan (4) ke Persamaan (3) dihasilkan:

$$\lambda \langle \Phi(x_k), \sum_{i=1}^m \alpha_i \Phi(x_i) \rangle = \langle \Phi(x_k), \frac{1}{m} \sum_{j=1}^m \Phi(x_j) \Phi(x_j)' \sum_{i=1}^m \alpha_i \Phi(x_i) \rangle$$

$$\lambda \sum_{i=1}^m \alpha_i \langle \Phi(x_k), \Phi(x_i) \rangle = \frac{1}{m} \sum_{i=1}^m \alpha_i \left\langle \Phi(x_k), \sum_{j=1}^m \Phi(x_j) \Phi(x_j)' \Phi(x_i) \right\rangle$$

$$\lambda \sum_{i=1}^m \alpha_i \Phi(x_i) \Phi(x_k)' = \frac{1}{m} \left[ \sum_{i=1}^m \alpha_i \Phi(x_i) \right] \left[ \sum_{j=1}^m \Phi(x_j) \Phi(x_j)' \right] \Phi(x_k)'$$

$$\forall k = 1, 2, \dots, m \quad (5)$$

Misalkan

$$k(x_i, x_j) = k_{ij} = \Phi(x_i) \Phi(x_j)' \quad (6)$$

dan  $K = [k_{ij}]$  matriks ukuran  $m \times m$ . Jika Persamaan (5) dilakukan pergantian semua suku dalam bentuk  $\Phi(x_i) \Phi(x_j)'$  dengan bentuk  $k_{ij}$ , maka dapat dibuktikan bahwa Persamaan (5) akan menjadi

$$\lambda m K \alpha = K^2 \alpha \quad (7)$$

di mana  $\alpha$  adalah vektor kolom dengan entri  $\alpha_1, \alpha_2, \dots, \alpha_m$ .

Solusi dari Persamaan (7), dapat dihitung dengan cara menyelesaikan

$$\lambda m \alpha = K \alpha \quad (8)$$

Misalkan  $\alpha^k$  adalah *eigen-vector* ke- $k$  dari masalah *eigen-value* pada Persamaan (7). Jika semua vektor  $\mathbf{v} \in F$  dilakukan normalisasi, yaitu harus memenuhi  $\mathbf{v}' \mathbf{v} = 1$ , maka dengan menggunakan Persamaan (4), (6), dan (8) akan dihasilkan

$$\begin{aligned} \mathbf{v}' \mathbf{v} &= \sum_{i,j=1}^m \alpha_i^k \alpha_j^k \langle \Phi(x_i), \Phi(x_j) \rangle \\ &= \sum_{i,j=1}^m \alpha_i^k \alpha_j^k \Phi(x_i) \Phi(x_j)' = (\alpha^k)' K \alpha^k \\ &= \lambda_k (\alpha^k)' \alpha^k = 1 \end{aligned} \quad (9)$$

Untuk mengekstraksi PC, maka semua peta dari input vektor  $\mathbf{z}$ , yaitu  $\Phi(\mathbf{z})$ , harus diproyeksikan ke vektor  $\mathbf{v}$  yang telah dinormalisasi. Untuk menghitung proyeksi tersebut digunakan

$$\mathbf{v}' \Phi(\mathbf{z}) = \sum_{i=1}^m \alpha_i^k k(\mathbf{x}, \mathbf{z}) \quad (10)$$

Ada beberapa kernel yang sudah dikenal untuk dipilih namun masih harus dikonstruksi satu demi satu, antara lain:

- *Kernel linier*:  $k(\mathbf{X}, \mathbf{Z}) = \mathbf{X}' \mathbf{Z}$
- *Kernel polinomial*:  $k(\mathbf{X}, \mathbf{Z}) = (\langle \mathbf{X}, \mathbf{Z} \rangle + b)^p$ , di mana  $p$  orde maksimal dari monomial dan  $b$  adalah bias menuju orde monomial terkecil.
- *Kernel Gaussian*:  $k(\mathbf{X}, \mathbf{Z}) = \exp\left(\frac{-\|\mathbf{X} - \mathbf{Z}\|^2}{2\sigma^2}\right)$ , di mana  $\sigma > 0$ .
- *Kernel sigmoid*:  $k(\mathbf{X}, \mathbf{Z}) = \tanh(\alpha \langle \mathbf{X}, \mathbf{Z} \rangle + \beta)$  di mana  $\alpha > 0$  dan  $\beta < 0$ .

Kernel PCA (Seng 2006) digunakan untuk memvisualisasi data diabetes Pima-Indian. Dalam artikelnya, Seng menyimpulkan bahwa PCA dan KPCA dapat digunakan sebagai alat

untuk memvisualisasi data dengan dimensi yang berbeda dari atribut eksplanatorinya. Meskipun tidak untuk memisahkan dua kelas data secara total, akan tetapi dapat memvisualisasi data berdasarkan seluruh atribut eksplanatori

## METODE

Visualisasi data didahului dengan melakukan penelaahan referensi yang terkait, melalui perpustakaan. Selanjutnya, dilakukan *searching* data melalui internet. Data yang digunakan adalah data sekunder, yaitu *Iris Data Set* yang disumbangkan oleh Marshall (1988). Untuk membedakan metode PCA dan KPCA dalam data ini, dianalisis menggunakan beberapa *software program*, antara lain, *R*, *Minitab 13*, *SPSS 12.0 for Windows*, dan *S-Plus 2000*.

## HASIL DAN PEMBAHASAN

Data Iris yang divisualisasikan di sini meliputi 150 contoh, terdiri atas tiga kelas masing-masing 50 contoh. Data ini dibagi atas 5 atribut, yang terdiri dari 4 atribut prediktif (numerik), *sepal length*, *sepal width*, *petal length*, *petal width*, yang semuanya diukur dalam cm, dan satu atribut kelas yang terbagi atas tiga jenis, *Iris Setosa*, *Iris Versicolour*, dan *Iris Virginica*. Deskripsi data ini dapat dilihat pada Tabel 1.

Tabel 1. Deskripsi data iris.

Variabel	Min	Max	Mean	SD	Korelasi Kelas
Sepal Length	4,30	7,90	5,8433	0,82807	0,7826
Sepal Width	2,00	4,40	3,0573	0,43587	-0,4194
Petal Length	1,00	6,90	3,7580	1,76530	0,9490
Petal Width	0,10	2,50	1,1993	0,76224	0,9565

Dari Tabel 1 tampak bahwa *petal length* (panjang daun) dan *petal width* (lebar daun) memiliki korelasi yang sangat tinggi. Sedangkan, *Scatterplot* masing-masing variabel dapat dilihat pada Gambar 2.

Gambar 2 menunjukkan bahwa setiap variabel tidak dapat dipisah garis linier secara sempurna, terutama antara *species versicolour* dengan *species virginica*. Di samping itu, antara *petal length* dengan *petal width*

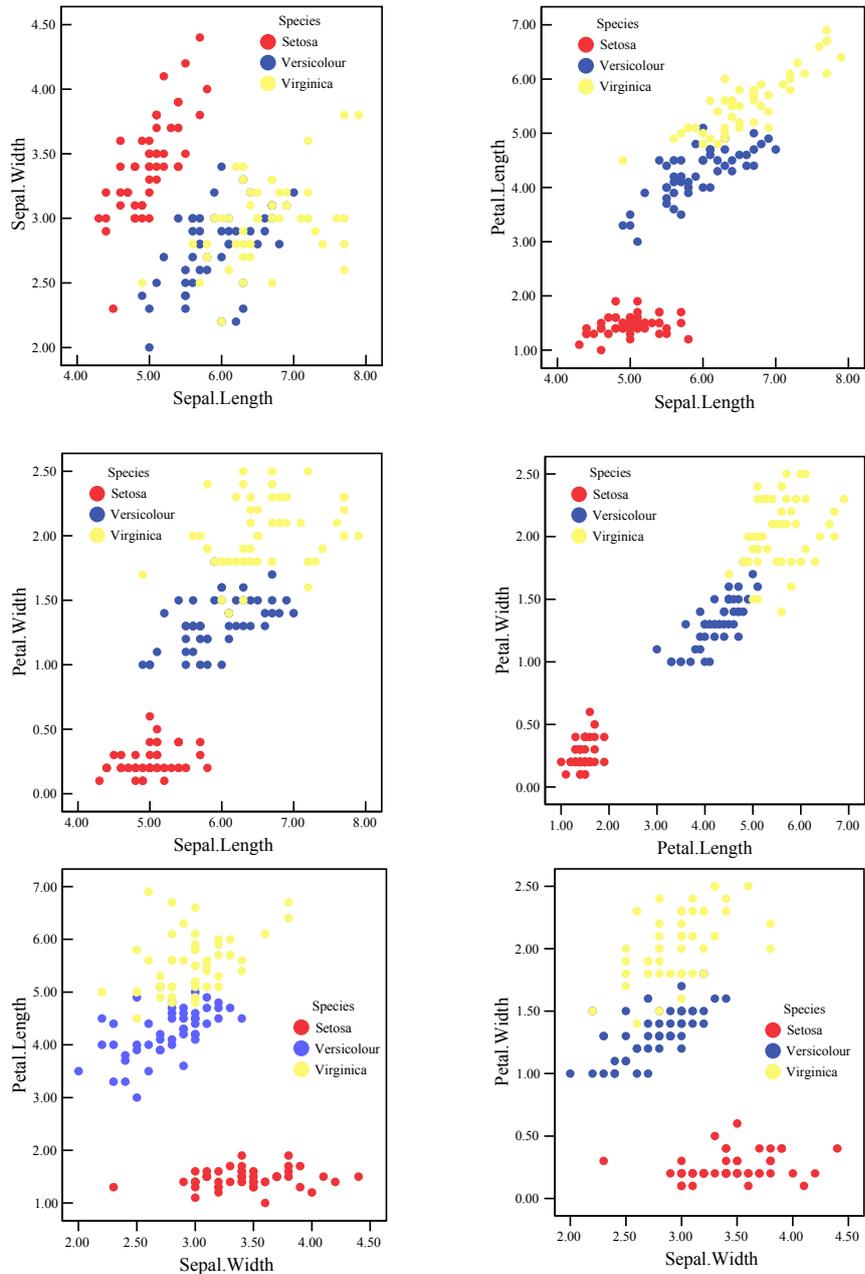
menunjukkan hubungan yang sangat tinggi, dan *species virginica* lebih besar (dari segi ukuran, panjang daun dan lebar daun) dibandingkan dengan *species* lainnya.

PCA dilaksanakan pada *dataset* setelah distandarisasi data yang menggunakan algoritma PCA baku. Setiap *eigen-value* diperoleh melalui *software*, dihubungkan dengan PC dan tingkat pengenalan masing-masing (Tabel 2).

Gambar 3 menunjukkan kontribusi kumulatif PC dari uraian data. Gambar ini memperlihatkan bahwa hingga tiga PC pertama diperlukan untuk menjelaskan sekitar 97% data, dan dua PC pertama bisa menjelaskan sekitar 90% data.

Proyeksi data Iris berturut-turut untuk dua dan tiga PC ditunjukkan pada Gambar 4. Dapat dilihat pada Gambar 4 bahwa terdapat kluster data menurut kelasnya (*species Setosa*, *Versicolour* dan *Virginica*), namun ketiga *species* itu tidak linier *separable*. Oleh karena klasifikasi data *non-linear* dan *non-separable*, maka digunakan KPCA untuk meninjau atau mengenal pola terbaik. Fungsi kernel yang digunakan adalah *radial basis function* (RBF), atau dikenal pula dengan kernel Gaussian. Beberapa nilai argumen ( $\sigma$ ) digunakan untuk melihat salah satu yang menghasilkan pola terbaik. Hasil visualisasi dari beberapa kondisi ini disajikan pada Gambar 5. Dari beberapa nilai argumen, ada kecenderungan berhimpitan satu jenis data dengan jenis data lainnya. Kondisi ini tidak lebih baik digunakan untuk visualisasi separasi kelas-kelas yang berbeda. Nilai argumen ( $\sigma$ ) yang dipikirkan (diperhatikan) lebih baik adalah  $\sigma = 0,001$ , karena nilai argumen ini memberikan separasi antar jenis data dengan baik. Hal ini tampak jelas pada Gambar 5, dimana untuk argumen yang diperhatikan, memberikan separasi linier secara sempurna.

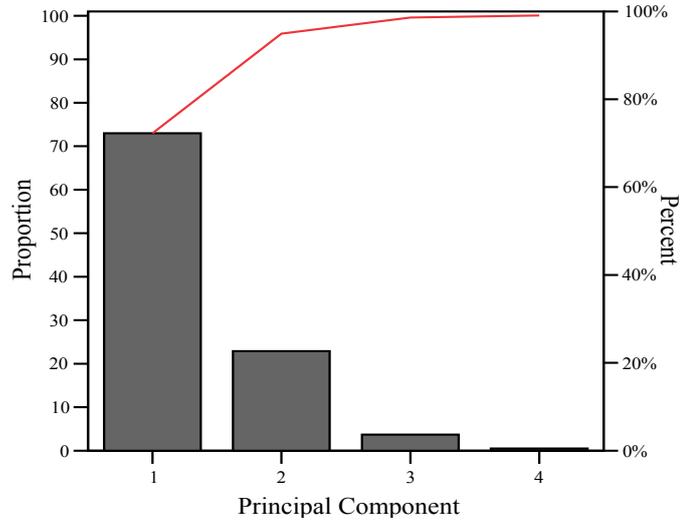
Untuk memvisualisasikan separasi secara lebih rinci, dipilih *scatterplot* dua PC pertama menggunakan KPCA dengan kernel RBF,  $\sigma = 0,001$ , seperti ditunjukkan di dalam Gambar 6(a), dan *scatterplot* 3-D untuk tiga PC pertama menggunakan KPCA dengan kernel RBF,  $\sigma = 0,001$ , yang ditunjukkan di dalam Gambar 6(b). Dengan membandingkan Gambar 4(a) dan 4(b) dengan Gambar 6(a) dan 6(b) berturut-turut, tampak bahwa KPCA memberikan separasi yang lebih baik daripada PCA linier.



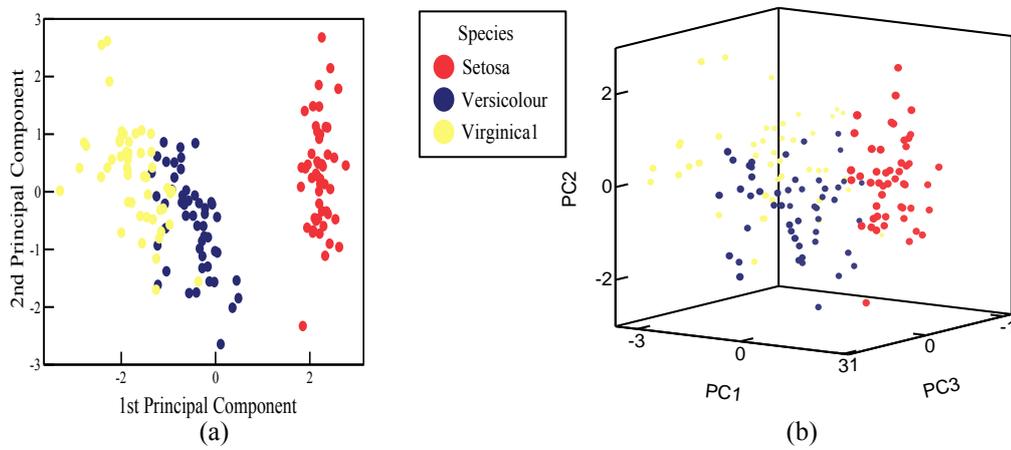
Gambar 2. Scatterplots antar variabel.

Tabel 2. Eigen-value setiap PCA.

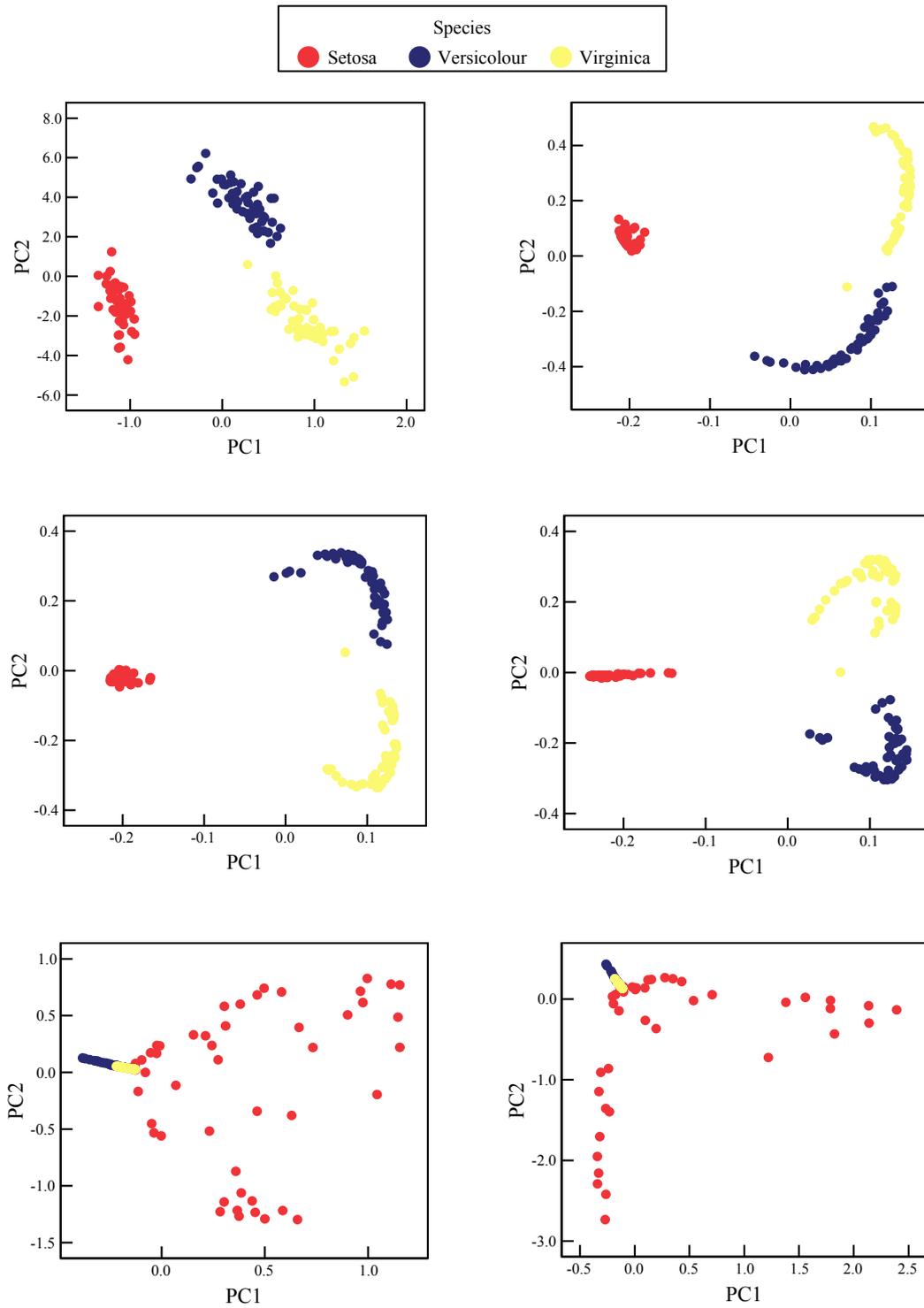
PC	Eigen-value	Tingkat Pengenalan
1	2.9185	73.0%
2	0.9140	22.8%
3	0.1468	3.7%
4	0.0207	0.5%



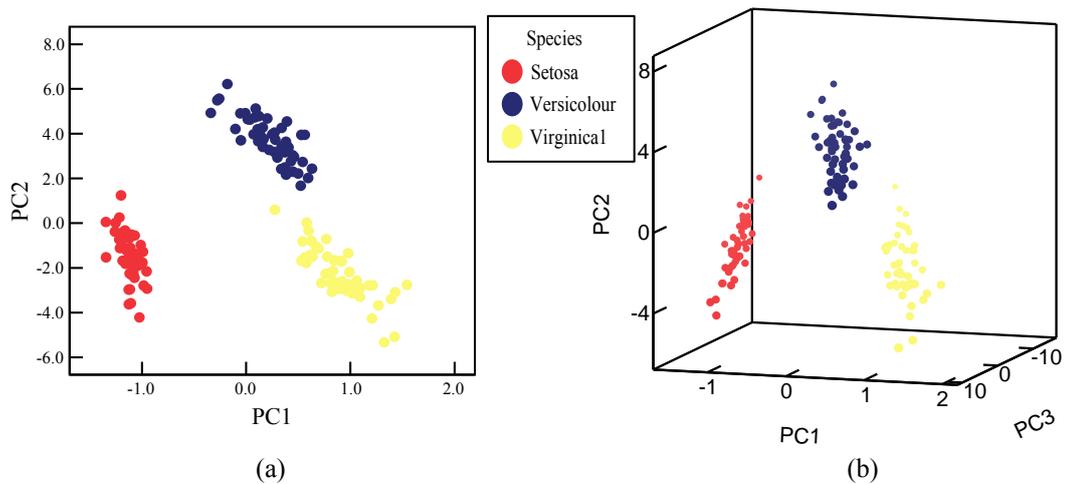
Gambar 3. Diagram pareto untuk kontribusi PC.



Gambar 4. Data Iris dalam (a) dua PC pertama dan (b) tiga PC pertama.



Gambar 5. Scatterplot KPCA menggunakan kernel RBF dengan beberapa nilai  $\sigma$ . Gambar 5. Scatterplot KPCA menggunakan kernel RBF dengan beberapa nilai  $\sigma$ .



Gambar 6. Data Iris dalam (a) Dua PC pertama dan (b) Tiga PC pertama menggunakan KPCA dengan  $\sigma = 0,001$ .

### KESIMPULAN

Principal component analysis (PCA) dan kernel principal component analysis (KPCA) adalah dua metode yang digunakan sebagai alat untuk memvisualisasi data dari dimensi-dimensi berbeda dalam atribut bersifat pengenalan. Namun, tidak secara total dapat memisahkan dua kelas data, yang dapat memvisualisasikan data secara bersamaan dengan mempertimbangkan semua atribut. Kegagalan dalam memisahkan data seperti ini tidak dapat dihubungkan dengan kelemahan-kelemahan metode itu sendiri.

Mengingat data adalah bagian yang sangat penting di dalam visualisasi, sehingga hal ini dapat menjadi bahan pertimbangan dalam menerapkan metode yang sesuai. Pengambilan data yang tidak mempunyai pola yang baik maka mengakibatkan hasil dalam ruang *feature* yang tidak bisa diekstrak secara sempurna. Fasilitas yang mendukung visualisasi juga akan sangat membantu dalam mengkonfirmasi signifikansi pengenalan pola *dataset*.

### DAFTAR PUSTAKA

- Alter O, Brown PO & Botstein D. 2000. Singular value decomposition for genome-wide expression data processing and modeling. *Proc Natl Acad Sci USA*, **97**: 10101-10106.
- Darlington RB. 1997. Multivariate Analysis. <http://www.psych.cornell.edu/darlington/manova.htm>.
- Härdle W & Simar L. 2003. *Applied Multivariate Statistical Analysis*. Berlin: Method and Data Technologies.
- Johnson, Richard A & Wicheren DW. 2002. *Applied Multivariate Statistical Analysis*, 5<sup>ed</sup>. New Jersey: Prentice-Hall, Inc.
- Lin W. 2006. Visualization of PCA and FDA on Waveform Data. *Computational Intelligence: Methods and Applications*.
- Marshall Michael. 1988. Iris Dataset. *UCI, Machine Learning Respository*. <http://archive.ics.uci.edu/ml/>
- Min W. 2006. PCA, Kernel PCA and MDS to Visualize the Zoo Data. *Computational Intelligence: Methods and Applications*, Nanyang Technological University.
- Raychaudhuri, Soumya, Stuart JM & Altman RB. 2000. Principal Component Analysis to Summarize Microarray Experiments: Application to Sporulation Time Series. *Pacific Symposium on Biocomputing*, vol. 5: 452-463.
- Schölkopf B, Alexander S & Klaus-RM. 1996. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Technical Report No. 44*, Nanyang Technological University.
- Seng CY. 2006. *Visualization of Pima-Indian-Diabetes Data Using Principal Component Analysis and Kernel Principal Component Analysis*. Nanyang Technological University: Computational Intelligence: Methods and Applications.
- Tipping ME & Christopher MB. 1999. Probabilistic Principal Component Analysis. *Journal of the Royal Statistical Society, Series B*, **61**, Part 3, 611-622.
- Xuejie Z. 2006. PCA and Kernel PCA in Visualizations of Glass Identification Data. *Computational Intelligence: Methods and Applications*. Nanyang Technological University.