BAB I
PENDAHULUAN

1.1. Latar Belakang

Analisis komponen utama atau *Principal Component Analysis* (PCA) merupakan teknik statistika yang banyak digunakan dan dibahas pada buku–buku analisis multivariat baik teori maupun aplikasi. Kegunaan PCA adalah untuk mereduksi dimensi data yang saling berkorelasi satu sama lain, selain itu PCA juga digunakan untuk mengatasi masalah multikolinieritas dan secara luas digunakan untuk menganalisis data yang berdimensi tinggi. PCA merupakan analisis tahap awal sebelum melakukan analisis diskriminan, analisis klaster atau teknik multivariat yang lain (Hubert, 2003).

Classic Principal Component Analysis (CPCA) adalah suatu metode dengan pendekatan klasik untuk menjelaskan struktur matriks kovarian melalui sejumlah kecil komponen yang tidak saling berkorelasi, dimana komponen utama merupakan kombinasi linier dari peubah asal sehingga mempunyai ragam maksimum. Pada pendekatan klasik, komponen utama ditentukan melalui persamaan lagrange. Pada kenyataanya, data seringkali mengandung beberapa pengamatan pencilan, oleh karena itu reduksi dimensi dengan CPCA menjadi tidak dapat diandalkan bila pencilan muncul dalam data (Sujatmiko, 2005).

Pada penelitian sebelumnya (Sujatmiko, 2005), peneliti membandingkan antara *Robust Principal Component Analysis* (ROBPCA) dengan penduga MVE dan ROBPCA dengan penduga *Minimum Covariant Determinant* (MCD) menggunakan data simulasi yang mengandung pencilan, diperoleh kesimpulan bahwa ROBPCA dengan MCD lebih baik digunakan untuk data mengandung pencilan daripada MVE. Pada penelitian ini, peneliti mencoba menganalisis data pencilan dengan ROBPCA menggunakan algoritma FAST-MCD dengan teorema *C-Step* menggunakan data sekunder. Menurut Maronna (2006), ROBPCA merupakan metode untuk mendapatkan komponen utama yang tidak mengandung pencilan, pendekatan pertama dilakukan dengan mengganti matriks kovarian yang klasik dengan penduga yang *robust*. Rousseeuw (2004) memperkenalkan MCD yang merupakan penduga yang *affine equivariant* dan *breakdown point* yang tinggi untuk mengganti matriks kovarian yang klasik.

1.2. Rumusan Masalah

Rumusan masalah dari penelitian ini adalah:

1. Bagaimana perubahan banyaknya komponen utama yang terbentuk jika menggunakan metode CPCA dan ROBPCA?
2. Bagaimana perubahan proporsi keragaman yang dapat dijelaskan oleh CPCA dan ROBPCA, dengan komponen utama yang sama banyak?

1.3. Batasan Masalah

Penelitian ini hanya mempelajari cara mendapatkan komponen utama dengan metode ROBPCA menggunakan penduga MCD dan mengetahui besarnya proporsi keragaman data yang dapat dijelaskan oleh komponen utama. Data yang digunakan pada penelitian ini adalah data multivariat yang mengandung pencilan.

1.4. Tujuan Penelitian

Penelitian ini bertujuan untuk:

1. Mengetahui perubahan banyaknya komponen utama yang terbentuk jika menggunakan metode CPCA dan ROBPCA
2. Mengetahui perubahan proporsi keragaman yang dapat dijelaskan oleh CPCA dan ROBPCA, dengan komponen utama yang sama banyak.

1.5. Manfaat Penelitian

Manfaat yang diharapkan dalam penelitian ini adalah agar metode ROBPCA dapat digunakan sebagai alternatif penyelesaian masalah dari data multivariat yang mengandung pencilan.

BAB II
TINJAUAN PUSTAKA

2.1. Identifikasi Pencilan

Pencilan adalah suatu pengamatan yang menyimpang dari pengamatan yang lain sehingga menimbulkan kecurigaan bahwa pengamatan tersebut berasal dari distribusi yang berbeda (Marazzi, 1993). Secara garis besar tujuan dari identifikasi pencilan adalah memisahkan distribusi pencemar dari distribusi dasar sehingga dapat diketahui pengamatan–pengamatan yang termasuk pencilan (Sujatmiko, 2005).

Untuk identifikasi pencilan pada data multivariat didasarkan pada jarak kuadrat mahalanobis (MSD = *Mahalanobis Square Distance*).

$$MSD_i = (\boldsymbol{x_i} - \bar{\boldsymbol{x}})'\boldsymbol{S}^{-1}(\boldsymbol{x_i} - \bar{\boldsymbol{x}}),\ \mathbf{i} = 1, 2, \ldots n \quad \ldots\ldots(2.1)$$

dengan $\bar{\boldsymbol{x}}$ merupakan vektor rata–rata berukuran *px1*, *n* adalah banyaknya pengamatan dan **S** adalah matriks kovarian berukuran *pxp*. Distribusi untuk MSD_i adalah χ_p^2. Jika $MSD_i > \chi^2_{p,(0{,}975)}$ maka pengamatan tersebut didefinisikan sebagai pengamatan pencilan.

Pencilan yang disebabkan oleh peubah prediktor dinamakan *leverage* (Suryana, 2005). *Leverage* sangat sulit diketahui karena :

1. Visualisasi seperti *scatter plot* tidak mampu menjelaskan secara utuh dalam gambar
2. Beberapa pencilan dalam data membentuk efek *masking*.

Pengukuran jarak ortogonal (*orthogonal distance*/OD) dilakukan dengan tujuan untuk mengetahui jenis observasi. OD dapat dinyatakan dengan rumus sebagai berikut (Hubert, 2003):

$$OD_i = \left\| \boldsymbol{x_i} - \bar{\boldsymbol{x}} - \boldsymbol{B_{p,q}}\boldsymbol{sk_i} \right\| \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2.2)$$

di mana:

$\boldsymbol{x_i}$: vektor pengamatan ke-i (*px1*)
$\bar{\boldsymbol{x}}$: vektor rata-rata (*px1*)
$\boldsymbol{B_{p,q}}$: matriks *loading* komponen utama(pxq)
$\boldsymbol{sk_i}$: vektor skor komponen pengamatan ke-i (*qx1*)
p : banyaknya peubah yang diamati
q : banyaknya komponen utama yang terbentuk.

Kombinasi jarak mahalanobis (MD) dan OD menghasilkan jenis observasi sebagai berikut (Hubert, 2003):

Tabel 2.1 jenis observasi berdasarkan kombinasi RM dan OD

Jarak	MD kecil	MD besar
OD besar	*Orthogonal outlier*	*Bad leaverage*
OD kecil	Observasi *regular*	*Good leverage*

Keterangan:
MD: Mahalanobis *Distance*
OD: *Orthogonal Distace*

2.2. Analisis Komponen Utama

Metode ini pertama kali ditemukan oleh Hotteling (1993) yang pada dasarnya bertujuan untuk menerangkan struktur kovarian melalui kombinasi linier dari peubah-peubah bebas dengan cara mereduksi data dan menginterpretasikannya. Meskipun dari *p* peubah asal dapat diturunkan *p* buah komponen utama untuk menerangkan variansi total sistem, namun seringkali variansi total itu dapat diterangkan secara memuaskan oleh sejumlah kecil komponen utama, misalkan q buah komponen utama (Gaspersz, 1995 dalam Astutik, 2007).

Analisis komponen utama dari *p* peubah bebas, dalam bentuk vektor dituliskan: $\boldsymbol{X} = (X_1, X_2, X_3, ..., Xp)$ ditransformasi menjadi vektor peubah $\boldsymbol{K} = (K_1, K_2, K_3, \ldots, K_q)$ dalam bentuk persamaan sebagai berikut :

$$K_k = b_{k1}X_1 + b_{k2}X_2 + b_{k3}X_3 + \ldots + b_{kp}X_p$$

di mana :

$$b_{k1}^2 + b_{k2}^2 + b_{k3}^2 + \cdots + b_{kp}^2 = 1$$

Var $(K_k) = \boldsymbol{b_k'X'Xb_k}$ untuk $k = 1, 2, 3, \ldots, q$

Sedemikian sehingga peubah-peubah $K_1, K, K_3, \ldots, K_q$ bebas satu dengan yang lain dan vektor peubah baru $\boldsymbol{K} = (K_1, K_2, K_3, \ldots, K_q)$ menjelaskan sebesar mungkin proporsi dari variansi vektor peubah semula $\boldsymbol{X} = (X_1, X_2, X_3, ..., X_p)$ (Astutik, 2007).

2.2.1. Matriks Masukan

Apabila unit satuan seluruh peubah yang akan dianalisis memiliki unit satuan yang sama, maka matriks satuan yang tepat digunakan adalah matriks kovarian atau matriks ragam peragam. Jika unit satuan peubah–peubah yang akan dianalisis berbeda maka matriks masukan yang tepat digunakan adalah matriks korelasi. Matriks kovarian juga dapat digunakan untuk data yang peubahnya mempunyai satuan berbeda, tetapi data harus ditransformasi ke dalam normal baku (Astutik, 2007).

2.2.1.1. Matriks Kovarian

Menurut Johnson (2002) jika $\boldsymbol{B}$ didefinisikan sebagai matriks konstan berukuran *pxq*, maka komponen utama didefinisikan sebagai kombinasi linier dari *p* peubah asal yang dinyatakan dalam bentuk persamaan matriks:

$$\boldsymbol{K}_{(\boldsymbol{q},\mathbf{1})} = \boldsymbol{B}_{(\boldsymbol{q},\boldsymbol{p})}\boldsymbol{x}_{(\boldsymbol{p},\mathbf{1})}$$

$$\underset{(q \times 1)}{\begin{bmatrix} K_1 \\ K_2 \\ \vdots \\ K_q \end{bmatrix}} = \underset{(qxp)}{\begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1p} \\ b_{21} & b_{22} & \cdots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{q1} & b_{q2} & \dots & b_{qp} \end{bmatrix}} \quad \underset{(p \times 1)}{\begin{bmatrix} \boldsymbol{X_1} \\ \boldsymbol{X_2} \\ \vdots \\ \boldsymbol{X_p} \end{bmatrix}}$$

di mana:

$\boldsymbol{k_k}$ = komponen utamake-*k* dari peubah asal

$\boldsymbol{x}$ = vektor kolom peubah asal

$\widetilde{\boldsymbol{b}}_{\boldsymbol{kj}}$ = *loading* komponen utama ke-*k* untuk peubah ke-*j* (vektor eigen).

Komponen utama ini tergantung pada matriks kovarian ($\boldsymbol{S}$) atau matriks korelasi ($\boldsymbol{R}$) dari peubah asal X_1, X_2, ...,X_p. Secara umum kombinasi linier yang dimaksud adalah:

$$K_1 = b_{11}X_1 + b_{12}X_2 + \dots + b_{1p}X_p = \boldsymbol{b_1^T X}$$

$$K_2 = b_{21}X_1 + b_{22}X_2 + \dots + b_{2p}X_p = \boldsymbol{b_2^T X}$$

$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots$$

$$K_q = b_{q1}X_1 + b_{q2}X_2 + \dots + b_{qp}X_p = \boldsymbol{b_q^T X}$$

Berdasarkan transformasi tersebut didapatkan ragam masing-masing komponen utama dan kovarian yaitu:

$$\text{Var}\,(k_k) = \mathbf{b}_{\boldsymbol{k}}^{\boldsymbol{T}}\boldsymbol{S}\boldsymbol{b_k} \quad \dots\dots(2.3)$$

$$\text{Cov}\,(k_k, k_j) = \mathbf{b}_{\boldsymbol{k}}^{\boldsymbol{T}}\boldsymbol{S}\boldsymbol{b_k} \quad k, j = 1, 2, \dots, p \quad \dots\dots(2.4)$$

di mana $k \neq j$

Menurut Maronna (2006) komponen utama yang pertama berupa kombinasi linier $K_1 = \boldsymbol{b_1^T X}$ yang bertujuan memaksimumkan Var(K_1) dengan batasan $\boldsymbol{b_1^T b_{\ 1}}$ =1 dan akan didapatkan bahwa:

Var(K_1)= $\mathbf{b_1^T} \sum_x \boldsymbol{b_1} = S_{k1}^2 = \lambda_1$

dimana λ_1 merupakan akar ciri terbesar pertama dari matriks $\boldsymbol{S}$ selanjutnya komponen utama pertama dapat ditulis:

$K_1 = b_{11}X_1 + b_{12}X_2 + \ldots + b_{1p}X_p = \boldsymbol{b_1^T X}.$

Sedangkan komponen utama kedua berupa kombinasi linier $K_2 = \boldsymbol{b_2^T X}$ yang bertujuan memaksimukan Var(K_2) yang tidak berkolerasi dengan komponen utama pertama tetapi bersifat ortogonal dengan komponen utama pertama. Oleh karena itu harus memenuhi batasan

$\boldsymbol{b_2^T b_2} = 1$ dan Cov $(K_1, K_2) = 0$

akan didapatkan

Var $(K_2) = \mathbf{b_2^T} \sum_x \boldsymbol{b_2} = S_{k2}^2 = \lambda_2$

dengan λ_2 merupakan akar ciri terbesar kedua dari matriks $\boldsymbol{S}$.

Selanjutnya komponen utama kedua dapat ditulis:

$K_2 = b_{12}X_1 + b_{22}X_2 + \ldots + b_{p2}X_p = \boldsymbol{b_2^T X}$

Jadi komponen utama ke-i merupakan kombinasi linier $k_i = \boldsymbol{b_i^T X}$ yang bertujuan memaksimumkan Var(k_i) dan tidak berkolerasi dengan komponen utama yang lain tetapi bersifat ortogonal dengan komponen utama yang lain. Oleh karena itu harus memenuhi:

$\boldsymbol{b_k^T b_k} = 1$ dan Cov $(k_k, k_j) = 0$

2.2.1.2. Matriks Korelasi

Bila komponen utama dihasilkan dari matriks kovarian, maka komposisi dari komponen utama tergantung pada ukuran satuan yang digunakan untuk mengukur peubah-peubah tersebut. Untuk mengatasi persoalan tersebut maka dapat ditempuh dengan membentuk komponen utama dari matriks korelasi. Bila *p* peubah asal tidak meggunakan satuan pengukuran yang sama maka peubah tersebut perlu ditransformasikan ke skor baku. Pembakuan peubah asal X ke dalam peubah baku Z dapat dilakukan dengan (Johnson, 2002):

$$Z_i = \frac{(X_i - \bar{X})}{\sqrt{\sigma_{ii}}} \; ; i = 1,2,...,n \quad \text{(2.5)}$$

Matriks $\mathbf{Z} = (\boldsymbol{V}^{1/2})^{-1} (\boldsymbol{x_i} - \overline{\boldsymbol{x}})$ dengan **V** adalah matriks diagonal simpangan baku yaitu:

$$(\boldsymbol{V}^{1/2})^{-1} = \begin{bmatrix} 1/\sqrt{s_{11}} & 0 & \dots & 0 \\ 0 & 1/\sqrt{s_{22}} & \dots & 0 \\ . & . & . & . \\ 0 & 0 & \dots & 1/\sqrt{s_{pp}} \end{bmatrix}$$

E(**Z**) = 0

$$\text{Cov}(\mathbf{Z}) = (\boldsymbol{V}^{1/2})^{-1} \boldsymbol{S} (V^{1/2})^{-1} = \boldsymbol{R} \quad \text{(2.6)}$$

Dengan demikian komponen utama dari $\boldsymbol{Z}$ dapat ditentukan dari vektor eigen matriks korelasi $\boldsymbol{R}$ peubah asal $\boldsymbol{X}$. Model komponen utama ke-p dapat ditulis:

$$K_1 = b_{11}X_1 + b_{12}X_2 + \dots + b_{1p}X_p = \boldsymbol{b_1^T X}$$
$$K_2 = b_{21}X_1 + b_{22}X_2 + \dots + b_{2p}X_p = \boldsymbol{b_2^T X}$$
$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots$$
$$K_q = b_{q1}X_1 + b_{q2}X_2 + \dots + b_{qp}X_p = \boldsymbol{b_q^T X}$$

Cara menentukan harga vektor-vektor b_k sama seperti penggunaan matriks kovarian, juga vektor-vektor ciri dan nilai akar ciri yang didapat dari matriks korelasi ($\boldsymbol{R}$).

2.2.2. Koefisien Pembobot *(Loading)* Komponen Utama

Loading komponen utama adalah hal yang penting dalam analisis komponen utama.

$$k_k = b_{k1}X_1 + b_{k2}X_2 + \dots\dots + b_{kp}X_p$$
$$k = 1, 2, \dots, q$$

Koefisien b_{kj} menunjukkan kontribusi peubah ke–j terhadap komponen utama ke–k dan tanda (positif atau negatif) menunjukkan arah pengaruh. *Loading* komponen utama dapat digunakan untuk interpretasi peubah komposit yang dihasilkan (Astutik, 2007).

2.2.3. Perhitungan *Loading* Komponen Utama

Mencari komponen utama $k_k = b_{k1}X_1 + b_{k2}X_2 + \ ... \ + b_{kp}X_p$ berarti mencari vektor koefisien $b_k = (b_{k1}, b_{k2}, \ldots, b_{kp})$ sedemikian sehingga $\boldsymbol{b_k^T X^T X b_k}$ yang merupakan ragam komponen utama, mencapai nilai maksimum jika $\boldsymbol{b_k^T b_k} = 1$. Permasalahan ini dapat diselesaikan dengan menggunakan fungsi *Lagrange* (Johnson, 2002). Maksimum *Lagrange* $\boldsymbol{b_k^T X^T X b_k}$ dengan kendala $\boldsymbol{b_k^T b_k} = 1$ atau $\boldsymbol{b_k^T b_k} - 1 = 0$

Dengan demikian dapat dibentuk fungsi *Lagrange* sebagai berikut:

$$L = \boldsymbol{b_k^T X^T X b_k} - \lambda_k(\boldsymbol{b_k^T b_k} - 1) \quad (2.7)$$

Apabila L diturunkan terhadap vektor $\boldsymbol{b}$ kemudian dinormalisasi maka diperoleh:

$$\frac{\partial L}{\partial \boldsymbol{b}} = 2\boldsymbol{b_k^T X^T X} - 2\lambda_k \boldsymbol{b_k} = 0$$

$$2(\boldsymbol{X^T X} - \lambda_k \boldsymbol{I})\boldsymbol{b_k} = 0$$

$$(\boldsymbol{X^T X} - \lambda_k \boldsymbol{I})\boldsymbol{b_k} = 0 \quad (2.8)$$

Persamaan (2.8) akan menghasilkan jawab *non trivial*. Matriks $(\boldsymbol{X^T X} - \lambda_k \boldsymbol{I})$ merupakan matriks *singular* dimana determinan dari matriks tersebut sama dengan nol

$$\left|\boldsymbol{X^T X} - \lambda_k \boldsymbol{I}\right| = 0 \quad (2.9)$$

Dengan menyelesaikan persamaan (2.9) akan diperoleh akar-akar karekteristik $\lambda_1, \lambda_2, \ldots, \lambda_q$ di mana $\lambda_1 > \lambda_2 > \cdots > \lambda_q > 0$. Setiap akar karakteristik λ_k akan menentukan b_k yaitu vektor karakteristik yang dinormalkan dari $\boldsymbol{X^T X}$.

Penentuan akar karakteristik mana dari *p* buah akar karakteristik yang ada untuk dipergunakan dalam komponen utama pertama adalah sebagai berikut:

Kembali pada persamaan (2.8) yaitu

$$(\boldsymbol{X^T X} - \lambda_k \boldsymbol{I})\boldsymbol{b_k} = 0$$

$$\boldsymbol{X^T X b_k} - \lambda_k \boldsymbol{I b_k} = 0$$

$$\boldsymbol{X^T X b_k} = \lambda_k \boldsymbol{I b_k} \quad (2.10)$$

Jika kedua sisi persamaan dikalikan dengan $\boldsymbol{b_k^T}$ akan menjadi

$$\boldsymbol{b_k^T X^T X b_k} = \boldsymbol{b_k^T} \lambda_k \boldsymbol{I b_k}$$

Dengan mengingat batasan yang telah diberikan yaitu $\boldsymbol{b_k^T b_k} = 1$ maka diperoleh:

$$\boldsymbol{b_k^T X^T X b_k} = \lambda_k \quad (2.11)$$

Dengan demikian dapat diketahui bahwa ragam setiap komponen utama berpadanan dengan nilai setiap akar karakteristik yang ada. Agar ragam komponen utama pertama maksimum, maka harus dipilih akar karakteristik terbesar dari matriks $\boldsymbol{X^T X}$ untuk digunakan dalam komponen utama pertama sehingga komponen utama pertama ini akan menjelaskan bagian terbesar dari keragaman yang dikandung oleh gugusan data yang telah dibakukan. Komponen-komponen utama lainnya menjelaskan proporsi keragaman yang semakin lama semakin kecil.

Secara umum vektor koefisien komponen utama ke-k (k = 1, 2, …, q) yaitu b_k ditentukan dengan cara menyelesaikan sistem persamaan (2.10). Agar komponen utama pertama mempunyai ragam yang maksimum maka di antara akar karakteristik yang ada dipilih akar karakteristik dengan nilai terbesar sehingga diperoleh b_1 yaitu vektor koefisien komponen utama pertama. Demikian juga untuk mendapatkan vektor koefisien komponen utama kedua b_2 yaitu dengan memilih akar karakteristik terbesar kedua untuk dimasukkan ke dalam persamaan (2.10), demikian seterusnya hingga diperoleh vektor koefisien komponen utama terakhir (Astutik, 2007).

2.2.4. Penentuan Komponen Utama yang Digunakan

Analisis lebih lanjut didasarkan pada komponen pokok yang bermakna yang memenuhi kriteria tertentu. Kriteria yang harus dipenuhi untuk memperoleh komponen utama yang dianggap bermakna adalah sebagai berikut:

1. Memilih akar karakteristik yang lebih besar dari 1 ($\lambda_j \geq 1$). Kriteria ini ditemukan oleh Kaiser (1958), Drapper and Smith (1981) dalam Astutik, 2007.
2. Memilih k buah komponen utama yang memberikan sumbangan terbesar terhadap keragaman data.

$$\frac{\sum_{j=1}^{k} \lambda_j}{\sum_{j=1}^{p} \lambda_j} > 0{,}80 \qquad (2.12)$$

   dalam hal ini p adalah banyaknya peubah semula (Johnson, 2002).
3. *Scree Plot*

Scree plot adalah plot antara nilai eigen dan komponen utama yang terbentuk. Melalui *scree plo*t dapat diketahui seberapa banyak komponen utama yang terbentuk dari *p* peubah asal.

2.2.5. Elipsoid

Fungsi ketepatan normal multivariat akan konstan pada permukaannya jika kuadrat jarak Q=$(\boldsymbol{x_i}-\overline{\boldsymbol{x}})^{\boldsymbol{T}}\boldsymbol{S}^{-\mathbf{1}}(\boldsymbol{x_i}-\overline{\boldsymbol{x}})$ konstan, dan disebut *contour* (Astutik, 2007).

$$f(x)=\frac{1}{(2\pi)^{p/2}\sqrt{|S|}}e^{-\frac{1}{2}Q}$$

Contour dari kepekatan konstan distribusi normal bivariat adalah elipsoid (Johnson,2002). Peluang kepekatan konstan *contour* dapat dirumuskan sebagai berikut:

$$(\boldsymbol{x_i}-\overline{\boldsymbol{x}})^{\boldsymbol{T}}\boldsymbol{S}^{-\mathbf{1}}(\boldsymbol{x_i}-\overline{\boldsymbol{x}})=c^2$$

Persamaan di atas merupakan permukaan ellipsoid yang mempunyai sumbu $\pm c\sqrt{\lambda_i}b_i$ dan berpusat di $\overline{\boldsymbol{x}}$. Sumbu setiap elipsoid adalah searah dengan vektor eigen dari $\mathbf{S}^{-\mathbf{1}}$ sedangkan panjang sumbu berpadanan dengan nilai eigen dari $\mathbf{S}^{-\mathbf{1}}$**.** Jika $\mathbf{S}$ definit positif, sehingga $\mathbf{S}^{-\mathbf{1}}$ ada maka:

$$\boldsymbol{Sb}=\boldsymbol{\lambda b} \text{ menyatakan } \mathbf{S}^{-\mathbf{1}}\boldsymbol{b}=\frac{\mathbf{1}}{\boldsymbol{\lambda}}\boldsymbol{b}$$

di mana $(\lambda,\boldsymbol{b})$ adalah pasangan nilai eigen dan vektor eigen dari $\boldsymbol{S}$ yang bersesuaian dengan $(\frac{1}{\lambda},\boldsymbol{b})$ dari $\mathbf{S}^{-\mathbf{1}}$ (Johnson, 2002). Elipsoid padat nilai $\boldsymbol{x}$ memenuhi:

$$(\boldsymbol{x_i}-\overline{\boldsymbol{x}})^{\boldsymbol{T}}\boldsymbol{S}^{-\mathbf{1}}(\boldsymbol{x_i}-\overline{\boldsymbol{x}})\leq \chi^2_{p;0,975}$$

di mana:

$\boldsymbol{x_i}$: vektor pengamatan ke-i

$\overline{\boldsymbol{x}}$: vektor rata-rata

$\chi^2_{p;0,975}$: *cut off* elipsoid

2.2.6. Skor Komponen Utama

Apabila komponen utama telah diperoleh, tahap berikutnya adalah menghitung skor komponen dari setiap individu yang akan digunakan dalam analisis lebih selanjutnya. Jika vektor skor dari individu ke-i adalah (X_{i1}, X_{i2},….,X_{ip}) atau dalam bentuk matriks data sebagai berikut (Johnson, 2002):

$$\mathbf{X} = \begin{bmatrix} X_{11} & \cdots & X_{1p} \\ \vdots & \ddots & \vdots \\ X_{n1} & \cdots & X_{np} \end{bmatrix}$$

Maka skor komponen dari individu ke-i pada komponen utama $k_{\ k}$ yang dihasilkan dari matriks kovarian adalah

$$sk_{ik} = (b_{1k}, b_{2k}, \ldots, b_{pk}) \begin{bmatrix} X_{i1} - & \bar{X}_1 \\ X_{i2} - & \bar{X}_2 \\ . & \\ . & \\ X_{1p} - & \bar{X}_p \end{bmatrix}$$

atau

$$sk_{ik} = \boldsymbol{b}_{\boldsymbol{k}}^{\prime}\ (\boldsymbol{x}_{\boldsymbol{i}}\text{-}\bar{\boldsymbol{x}}) \quad \ldots\ldots\ldots\ldots (2.13)$$

di mana:

sk_{ik} : skor komponen ke-*k* dari individu ke-*i*
$\boldsymbol{b}_{\boldsymbol{k}}^{\prime}$: vektor pembobot komponen utama ke-*k*
$\boldsymbol{x}_{\boldsymbol{i}}$: vektor data individu ke - *i*
$\bar{\boldsymbol{x}}$: vektor nilai rata–rata dari peubah asal

Jika komponen utama dihasilkan dari matriks korelasi $\boldsymbol{R}$, maka rumus skor komponen utama dari individu ke–i yaitu:

$$sk_{\boldsymbol{ik}} = \boldsymbol{b}_{\boldsymbol{k}}^{\prime}\boldsymbol{V}^{-\mathbf{1/2}}(\boldsymbol{X}_{\boldsymbol{i}}\text{-}\bar{\boldsymbol{X}}) \quad \ldots\ldots\ldots\ldots (2.14)$$

di mana

$$\boldsymbol{V}^{-1/2} = \text{diag}\ (\ 1/\sqrt{S_{11}}, \ldots, 1/\sqrt{S_{pp}}\ ),\ S_{kk} \quad \ldots\ldots\ldots\ldots (2.15)$$

menyatakan elemen diagonal ke- k dari **S**, k = 1, 2,…, *q*.

2.3. Analisis Komponen Utama *Robust* (ROBPCA)

ROBPCA adalah suatu metode yang bertujuan untuk menghasilkan komponen utama yang tidak dipengaruhi oleh adanya pengamatan pencilan (Hubert, 2003). Langkah awal sebelum melakukan analisis dengan metode ROBPCA adalah pendeteksian pencilan menggunakan *Mahalanobis distance* (Hubert, 2003). Prinsip dasar dari ROBPCA adalah mengganti matriks kovarian dan vektor rata-rata dengan penduga yang *robust* bagi matriks kovarian dan vektor rata-rata. Pendugaan terhadap matriks kovarian dan vektor rata-rata dilakukan dengan penduga *Minimum Covariance Determinant* (MCD) (Rousseeuw, 2003). Langkah-langkah untuk mendapatkan komponen utama dengan ROBPCA hampir sama dengan CPCA, yang membedakan adalah pada ROBPCA matriks kovarian dan vektor rata-rata diduga dengan MCD.

Ukuran tingkat *robust* yang sangat bermanfaat dari suatu penaksir adalah *breakdown point*. *Breakdown point* adalah jumlah pengamatan minimal yang dapat menggantikan sejumlah pengamatan semula yang berakibat pada nilai taksiran yang dihasilkan sangat berbeda dari taksiran sebenarnya (Hubert, 2007). Penduga yang *robust* dinyatakan dengan *breakdown point* (tingkat *robust*). Misalkan $\boldsymbol{X}$ adalah matriks data dan $\boldsymbol{X}_m$ adalah matriks data di mana sejumlah *m* pengamatan digantikan suatu nilai yang berbeda dari aslinya, maka *breakdown point* dari penduga **t(X)** pada data **X** adalah proporsi pencilan terkecil *m/n* yang mengakibatkan taksiran menjadi takterhingga (Hubert, 2007). *Breakdown point* penaksir kovarian **C(X)** didefinisikan sebagai proporsi pencilan terkecil *m/n* yang menyebabkan nilai eigen terbesar $\boldsymbol{\lambda_1}(\boldsymbol{C})$ mencapai tak berhingga atau nilai eigen terkecil $\boldsymbol{\lambda_p}(\boldsymbol{C})$ mencapai nol.

Penduga *robust* MCD memiliki sifat statistik yang baik harus memenuhi sifat *affine equivariant* (Rousseeuw,2003). Penduga **t**(**X**) dikatakan *affine equivariant* jika:

$$\mathbf{t}(XA + jv^T) = t(X)A + v \quad \text{.......................................}(2.16)$$

untuk semua **A** matriks nonsingular berdimensi *pxp*, di mana $\boldsymbol{v}$ adalah vektor *px1* dan $\mathbf{j} = [\mathbf{1}, \mathbf{1}, \dots, \mathbf{1}]^T$ vektor *nx1*. Penduga kovarian MCD bersifat *affine equivariant* jika:

$$C(XA+jv^T) = A^T C(X) A \text{...}(2.17)$$

Jika ***X*** adalah sebuah vektor random di mana elemen baris menyatakan n observasi dan banyaknya kolom menyatakan p peubah , , …., .

2.3.1 Penduga *Robust Minimum Covariance Determinant* (MCD)

Penduga *Robust* MCD merupakan penduga vektor rata-rata t**(X)** dan matriks kovarian **C(X)**. Nilai **C(X)** dipilih dari sebagian pengamatan yang meminimumkan determinan matriks kovarian (Suryana, 2007). Menurut Hubert (2007), MCD memiliki sifat statistik yang baik karena memenuhi sifat *affine equivariant*. MCD juga merupakan penaksir *robust* dengan *breakdown point* tinggi. **C(X)** berukuran (*pxp*) yang simetris dan definit positif dari sub sampel berukuran *h* pengamatan, dengan $[(n+p+1)/2] < h \leq n$ dan merupakan nilai integer terkecil dari $[(n+p+1)/2]$ dengan ketentuan bahwa

$$\mathbf{t(X)} = 1/h \sum_{i=1}^{h} \boldsymbol{x}_j \quad \text{...(2.18)}$$

$$\mathbf{C(X)} = 1/h\text{-}1 \sum_{i=1}^{h} (\boldsymbol{x}_i - \boldsymbol{t}_1)(\boldsymbol{x}_i - \boldsymbol{t}_1)^T \quad \text{..............(2.19)}$$

Metode MCD mencari himpunan bagian dari ***X*** sejumlah elemen . Misalkan himpunan bagian tersebut adalah . Terdapat kombinasi yang harus ditemukan untuk mendapatkan penaksir MCD. Untuk *n* kecil, penaksir MCD cepat ditemukan, tetapi jika *n* besar maka banyak kombinasi sub sampel yang harus ditemkan untuk mendapatkan penaksir MCD. Keterbatasan ini menghantarkan penemuan tentang algoritma FAST-MCD oleh Rousseeuw dan Van Driessen (2004). Salah satu aspek penting dalam algoritma FAST-MCD adalah teorema *C-Steps.*

2.3.2. Algoritma FAST-MCD dengan *C-Steps*

Diketahui *X* merupakan himpunan data sejumlah *n* pengamatan ***X*** = (X_1, X_2, X_3, …, terdiri dari *p* peubah. Misal dengan jumlah elemen tetapkan

$$\boldsymbol{t}_1 = \frac{1}{h} \sum_{i \in H_1} \boldsymbol{x}_i \quad \text{...(2.20)}$$

dan

$$\boldsymbol{C_1} = \frac{1}{\mathrm{h}}\sum_{\mathrm{i}\in\mathrm{H}_1}(\mathbf{x_i} - \mathbf{t_1})\,(\mathbf{x_i} - \mathbf{t_1})^{\mathrm{T}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2.21)$$

Jika det($\boldsymbol{C_1}$) $\neq 0$, didefinisikan jarak relatif

$$d_1(i) = \sqrt{(\mathbf{x_i} - \mathbf{t_1})^{\mathrm{T}}\mathbf{C_1^{-1}}(\mathbf{x_i} - \mathbf{t_1})} \quad \text{untuk } i = 1, \ldots, n \ldots\ldots\ldots(2.22)$$

Selanjutnya ambil H_2 sedemikian sehingga

$\{d_1(i); i \in H_2\} = \{(d_1)_{1:n}, \ldots, (d_1)_{h:n}\}$ di mana $(d_1)_{1:n} \leq (d_1)_{2:n} \ldots \leq (d_1)_{h:n}$ menyatakan urutan jarak, kemudian menghitung $\boldsymbol{t_2}$ dan $\boldsymbol{C_2}$ berdasarkan himpunan H_2, maka det($\boldsymbol{C_2}$) $\leq$ det($\boldsymbol{C_1}$) dan akan sama jika dan hanya jika $\boldsymbol{t_1} = \boldsymbol{t_2}$ dan $\boldsymbol{C_1} = \boldsymbol{C_2}$. Asumsikan det($\boldsymbol{C_2}$) > 0, selanjutnya $d_2(i) = d_{(t_2;C_2)}(i)$ untuk semua $i = 1, \ldots, n$ dengan menggunakan $\#H_2 = h$ dan definisi $(\boldsymbol{t_2}, \boldsymbol{C_2})$, diperoleh

$$\frac{1}{hp}\sum_{i\in H_2} d_2^2(i) = \frac{1}{hp}\,\mathrm{tr}\sum_{i\in H_2}(\mathbf{x_i} - \mathbf{t_2})^{\mathrm{T}}\mathbf{C_2^{-1}}(\mathbf{x_i} - \mathbf{t_2})$$

$$= \frac{1}{hp}\,tr\sum_{i\in H_2}\mathbf{C_2^{-1}}(\mathbf{x_i} - \mathbf{t_2})\,(\mathbf{x_i} - \mathbf{t_2})^{\mathrm{T}}$$

$$\frac{1}{hp}\sum_{i\in H_2} d_2^2(i) = \frac{1}{p}\,tr\mathbf{C_2^{-1}}\mathbf{C_2} = \frac{1}{p}\mathrm{tr}(\mathbf{I}) = 1 \ldots\ldots\ldots\ldots\ldots(2.23)$$

selanjutnya

$$\lambda = \frac{1}{hp}\sum_{i\in H_2} d_1^2(i) = \frac{1}{hp}\sum_{i=1}^{h}(d_1^2)_{i:n} \leq \frac{1}{hp}\sum_{j\in H_1} d_1^2(j)1 \ldots\ldots(2.24)$$

di mana $\lambda > 0$

dengan menggabungkan antara (2.23) dan (2.24) dihasilkan

$$\frac{1}{hp}\sum_{i\in H_2} d^2_{(\boldsymbol{t_1},\lambda\boldsymbol{C_1})}(i) = \frac{1}{hp}\sum_{i\in H_2}(\mathbf{x_i} - \mathbf{t_1})^{\mathrm{T}}\frac{1}{\lambda}\mathbf{C_1^{-1}}(\mathbf{x_i} - \mathbf{t_1})$$

$$= \frac{1}{\lambda hp}\sum_{i\in H_2} d_1^2(i) = \frac{\lambda}{\lambda} = 1$$

akibatnya det($\lambda\boldsymbol{C_1}$) $\leq$ det($\boldsymbol{C_1}$). Berdasarkan rumus (2.23) diperoleh pertidaksamaan det(C_2) $\leq$ det(λC_1), sehingga det(C_2) $\leq$ det(λC_1) $\leq$ det(C_1). Bentuk lain penaksir MCD adalah dengan menggunakan pembobot. Pengamatan yang tidak disertakan dalam perhitungan penaksir rata-rata dan kovarian MCD diberi bobot nol, lainnya diberi bobot sama dengan satu (Suryana,2005). Penaksir MCD dihitung dengan

$$\boldsymbol{t}_{MCD} = \frac{\sum_{i=1}^{n} w_i \boldsymbol{x_i}}{\sum_{i=1}^{n} w_i} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2.25)$$

$$\boldsymbol{C}_{MCD} = \frac{\sum_{i=1}^{n} w_i(\mathbf{x_i}-\mathbf{t}_{\mathrm{MCD}})(\mathbf{x_i}-\mathbf{t}_{\mathrm{MCD}})^{\mathrm{T}}}{\sum_{i=1}^{n} w_1 - 1} \quad \text{.........................(2.26)}$$

dengan

$$w_i = \begin{cases} 1 \text{ jika } (\boldsymbol{x_i} - \boldsymbol{t}_{MCD})^T \boldsymbol{C}_{MCD}^{-1} (\boldsymbol{x_i} - \boldsymbol{t}_{MCD}) \le \chi^2_{p,0.975} \\ 0 \text{ lainnya} \end{cases}$$

Identifikasi pencilan dengan persamaan (2.1) tidak berjalan maksimal bila data mengandung lebih dari satu pengamatan pencilan. Terdapat dua kondisi yang berkaitan dengan pencilan yaitu *masking* dan *swamping*. *Masking* terjadi ketika suatu pengamatan pencilan tidak teridentifikasi karena adanya pengamatan lain yang berdekatan. *Swamping* terjadi ketika pengamatan baik, teridentifikasi sebagai pencilan. Oleh karena itu dikembangkan metode $RMSD_i$ yang *robust* untuk vektor rata-rata dan matriks kovarian untuk mengatasi masalah *masking* dan *swamping.*

$$RMSD_i = (\boldsymbol{x_i} - \boldsymbol{t}(\boldsymbol{X}))'(\boldsymbol{C}(\boldsymbol{X}))^{-1}(\boldsymbol{x_i} - \boldsymbol{t}(\boldsymbol{X})) \text{.................(2.27)}$$

i = 1, 2, ….*n*;

di mana

p : banyaknya peubah

$RMSD_i$: *Robust Mahalanobis Square Distance* untuk setiap pengamatan ke-i

dengan **t(X)** adalah penduga yang *robust* untuk vektor rata-rata dan **C(X)** adalah penduga yang *robust* untuk matriks kovarian (Sujatmiko, 2005). Jarak *robust* dihitung setelah diperoleh taksiran rata-rata dan matriks kovarian yang *robust* menggunakan MCD.

2.3.3. Indikator Pembanding

Pada Penelitian ini, indikator perbandingan yang digunakan adalah besarnya proporsi keragaman yang dapat dijelaskan oleh penduga nilai eigen, jika kedua metode (CPCA dan ROBPCA) menghasilkan komponen utama terpilih yang sama banyak.

$$\frac{\sum_{k=1}^{q} \lambda_k}{\sum_{j=1}^{p} \lambda_k} > 80\% \text{ ....(2.28)}$$

di mana p adalah banyaknya peubah asal (Johnson, 2002)

BAB III
METODE PENELITIAN

3.1. Sumber Data

Data yang digunakan dalam penelitian ini adalah data sekunder. Data yang diambil merupakan data multivariat (Lampiran 1-3). Pengolahan data dilakukan dengan bantuan software MATLAB 7.0.1 dan S-PLUS 6.

Tabel 3. 1 Data pengeluaran RT berdasarkan pendapatan RT

No	Peubah Pengamatan	Sumber	*n*	*p*
1	1. konsumsi beras (rupiah/bulan) 2. konsumsi umbi (rupiah/bulan) 3. konsumsi ikan (rupiah/bulan) 4. konsumsi daging (rupiah/bulan) 5. konsumsi telur (rupiah/bulan) 6. konsumsi kacang-kacangan (rupiah/ bulan) 7. konsumsi buah (rupiah/bulan) 8. konsumsi sayuran (rupiah/bulan) 9. konsumsi makanan dan minuman jadi (rupiah/bulan) 10. konsumsi bahan pangan lain (rupiah/bulan)	Hartanto (2006)	91	10
2	1. konsumsi beras (rupiah/minggu) 2. konsumsi umbi (rupiah/minggu) 3. konsumsi ikan (rupiah/minggu) 4. konsumsi daging (rupiah/minggu) 5. konsumsi telur dan susu (rupiah/ minggu) 6. konsumsi sayur-sayuran (rupiah/minggu) 7. konsumsi kacang-kacangan (rupiah/minggu) 8. konsumsi buah-buahan (rupiah/minggu) 9. konsumsi minyak dan lemak (rupiah/minggu) 10. konsumsi bahan minuman (rupiah/minggu) 11. konsumsi bumbu-bumbuan	Data hasil survei Sosial Ekonomi Nasional di Kota Kupang (2005)	608	26

	(rupiah/minggu) 12. konsumsi konsumsi lainnya (rupiah/minggu) 13. konsumsi makanan dan minuman jadi (rupiah/minggu) 14. konsumsi minuman alkohol (rupiah/minggu) 15. konsumsi tembakau dan sirih (rupiah/minggu) 16. pengeluaran sewa rumah (rupiah/bulan) 17. pengeluaran listrik (rupiah/bulan) 18. pengeluaran untuk pemeliharaan rumah (rupiah/bulan) 19. pengeluaran barang dan jasa (rupiah/bulan) 20. pengeluaran pendidikan (rupiah/bulan) 21. pengeluaran kesehatan (rupiah/bulan) 22. pengeluaran pakaian (rupiah/bulan) 23. pengeluaran barang tahan lama (rupiah/bulan) 24. pengeluaran pajak dan iuran (rupiah/bulan) 25. pengeluaran asuransi (rupiah/bulan) 26. pengeluaran pesta (rupiah/bulan)			
3	1. konsumsi beras (rupiah/minggu) 2. konsumsi umbi (rupiah/minggu) 3. konsumsi ikan (rupiah/minggu) 4. konsumsi daging (rupiah/minggu) 5. konsumsi telur dan susu (rupiah/ minggu) 6. konsumsi sayur-sayuran (rupiah/minggu) 7. konsumsi kacang-kacangan (rupiah/minggu) 8. konsumsi buah-buahan (rupiah/minggu) 9. konsumsi minyak dan lemak (rupiah/minggu) 10. konsumsi minuman beralkohol	Data hasil survei Sosial Ekonomi Nasional di Kota Sorong (2002)	340	20

	(rupiah/minggu) 11. konsumsi minuman non alcohol (rupiah/minggu) 12. konsumsi tembakau dan sirih (rupiah/minggu) 13. pengeluaran sewa rumah (rupiah/bulan) 14. pengeluaran perumahan dan fasilittasnya (rupiah/bulan) 15. pengeluaran biaya pendidikan (rupiah/bulan) 16. pengeluaran transportasi (rupiah/bulan) 17. pengeluaran barang dan jasa (rupiah/bulan) 18. pengeluaran pajak dan iuran RT (rupiah/bulan) 19. pengeluaran kesehatan (rupiah/bulan) 20. pengeluaran pesta dan upacara(rupiah/bulan)			

Keterangan: n : banyaknya responden (rumah tangga)
p : banyaknya peubah/peubah yang diamati
RT : Rumah Tangga

3.2. Metode

Langkah-langkah analisis yang dilakukan dalam penelitian ini adalah:

1. Mendeteksi pencilan menggunakan rumus (2.1)
2. Jika tidak terdapat pengamatan pencilan data dianalisis dengan CPCA, jika terdapat pengamatan pencilan, maka dilanjutkan ke langkah 3.
3. Menganalisis data dengan metode ROBPCA

3.1.Mendapatkan matriks kovarian yang *robust* dengan metode MCD atau FAST-MCD.

3.1.1.Menentukan himpunan awal X_{1h} dengan jumlah elemen himpunan bagian sejumlah h pengamatan yang berbeda sehingga terdapat ${}^{n}_{\mathrm{h}}C$ sub sampel, h nilai integer terkecil dari $[n+p+1]/2$

3.1.2. Untuk setiap sub sampel, dihitung vektor rata-rata dan metriks kovarian $(\boldsymbol{t_1}, \boldsymbol{C_1})$ dengan rumus (2.21) dan (2.22)

3.1.3. Menghitung jarak Mahalanobis, mengambil h pengamatan dengan jarak terkecil kemudian menjadikan h pengamatan tersebut sebagai elemen himpunan $\boldsymbol{X}_{2h}$.

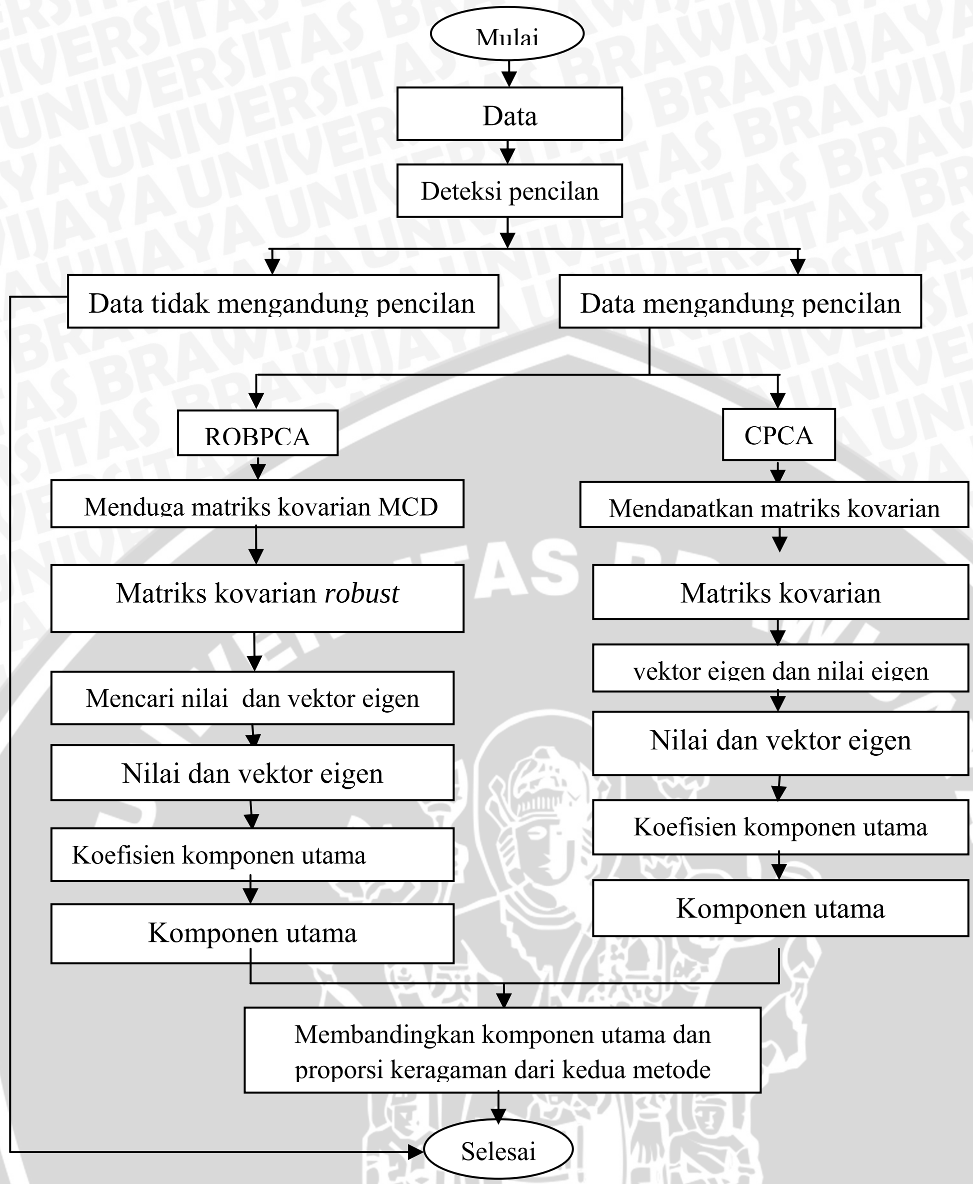
3.1.4. Menghitung $(\boldsymbol{t_2}, \boldsymbol{C_2})$, kemudian kembali ke langkah (3.1.2) sampai diperoleh determinin minimum dari matriks kovarian yang telah konvergen sehingga didapat $(.\boldsymbol{t_{MCD}}, \boldsymbol{C_{MCD}})$

3.2. Menentukan koefisien komponen utama *robust* menggunakan rumus (2.9)

3.3. Menentukan banyaknya komponen utama yang digunakan menggunakan rumus (2.13)

3.4. Menentukan skor komponen utama *robust* menggunakan rumus (2.14) atau (2.15)

4. Membandingkan komponen utama dan proporsi keragaman yang dihasilkan jika menggunakan kedua metode tersebut.
5. Langkah-langkah analisis dijelaskan melalui diagram alir penelitian (Gambar 3).



Gambar 3.1. Skema metode penelitian

BAB IV
HASIL DAN PEMBAHASAN

4.1.Identifikasi Pencilan

Hasil identifikasi pencilan pada penelitian ini dapat dilihat pada Tabel 4.1.

Tabel 4.1 Banyaknya pengamatan pencilan pada data penelitian menggunakan jarak mahalanobis

No	Data	n	p	Banyak pencilan	*Cut off* $\left(\sqrt{\chi^2_{(p;0,975)}}\right)$
1	I	91	10	6	4,5258
2	II	608	26	75	6,4748
3	III	340	20	62	5,8455

Sumber: output MATLAB 7.0.1

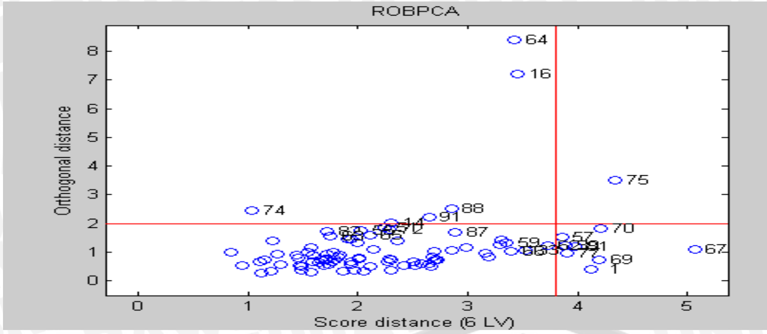
Keterangan

n : banyaknya pengamatan (responden/rumah tangga)

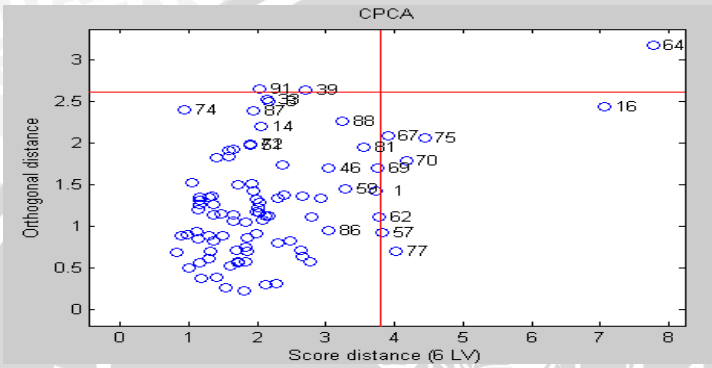
p : banyaknya peubah yang diamati

cut off : batas pendefinisian pencilan

Berdasarkan Tabel 4.1, dapat diketahui bahwa data I terdapat 6 pengamatan dengan jarak mahalanobis (MD) > *cut off* $\chi^2_{(10;0,975)}$, sehingga 6 pengamatan tersebut teridentifikasi sebagai pencilan. Data II terdapat 75 pengamatan dengan nilai MD > $\chi^2_{(26;0,975)}$, sehingga 75 pengamatan tersebut teridentifikasi sebagai pengamatan pencilan dari 608 pengamatan. Data III terdapat 62 pengamatan yang mempunyai nilai MD > $\chi^2_{(20;0.975)}$, sehingga 62 pemgamatan tersebut teridentifikasi sebagai pencilan. Pengklasifikasian pencilan dapat dilihat pada Gambar 1-2 dan Lampiran 24-25.



Gambar 4.1. Plot antara jarak ortogonal dan jarak mahalanobis dengan ROBPCA untuk data I



Gambar 4.2. Plot antara jarak ortogonal dan jarak mahalanobis dengan CPCA untuk data I

Berdasarkan Gambar 1, ROBPCA menghasilkan beberapa *orthogonal outlier* (observasi ke-16, 64, 74, 88, 91), 1 titik *bad leverage* (observasi ke-75), dan beberapa *good leverage* (observasi ke-1, 57, 67, 69, 70, 77, 81). Berdasarkan Gambar 2, CPCA menghasilkan 1 titik *bad leverage* (observasi ke-64), dan menghasilkan *good leverage* (observasi ke-1, 16 67, 69, 70, 75, 77). Visualisasi *outlier map* antara jarak ortogonal dan jarak *robust* untuk data penelitian II dan III dapat dilihat pada Lampiran 24 dan 25. Berdasarkan lampiran 24 dan 25, dapat diketahui bahwa jumlah *orthogonal outlier* dan *bad leverage* yang dideteksi ROBPCA lebih banyak daripada CPCA.

4.2. Matriks masukan

Data penelitian I menggunakan 10 peubah pengeluaran konsumsi rumah tangga dengan satuan yang sama, yaitu rupiah/bulan, artinya besarnya pengeluaran konsumsi rumah tangga selama satu bulan dalam satuan rupiah. Oleh karena itu matriks masukan yang digunakan adalah matriks kovarian. Pengukuran peubah pada Lampiran 7 menghasilkan *rank* yang berbeda jauh. Dengan sifat seperti itu, analisis komponen utama sebaiknya didasarkan pada data yang distandardisasi. Data sebelum dan sesudah standardisasi dapat dilihat pada Lampiran 4. Data penelitian II dan III, mempunyai satuan peubah yang berbeda, dan mempunyai *rank* data yang berbeda nyata (Lampiran 8 dan 9), sehingga analisis komponen utama didasarkan pada data yang distandardisasi (Lampiran 5 dan 6), dan matriks masukan yang digunakan adalah matriks kovarian.

4.3. Nilai eigen dari matriks kovarian

Nilai eigen dalam analisis komponen utama berkaitan erat dengan *loading* komponen utama yang terbentuk. Hal ini dapat dilihat pada rumus (2.12), yang dapat diartikan bahwa ragam setiap komponen utama berpadanan dengan nilai setiap akar karakteristik yang ada. *Loading* komponen utama data penelitian pertama menggunakan metode ROBPCA dengan penduga MCD dapat dilihat pada Lampiran 13. Tanda positif atau negatif pada *loading* komponen utama hanya menunjukkan arah.

Komponen utama pertama pada data penelitian I, peubah konsumsi makanan dan minuman jadi (X_9) memberikan kontribusi terbesar terhadap komponen utama pertama, dengan nilai *loading* sebesar 0,4878. Pada komponen utama kedua, peubah konsumsi beras memberikan kontribusi terbesar terhadap komponen utama kedua, dengan nilai *loading* sebesar 0,7699. Komponen utama pertama pada data penelitian II, peubah konsumsi sayur-sayuran (X_6) memberikan kontribusi terbesar komponen utama pertama, dengan nilai *loading* sebesar 0,3107. Pada komponen utama kedua, peubah untuk pengeluaran asuransi memberikan kontribusi terbesar terhadap komponen utama kedua, dengan nilai *loading* sebesar 0,5381. Komponen utama pertama pada data penelitian III, peubah konsumsi ikan (X_3) memberikan kontribusi terbesar terhadap komponen utama pertama, dengan nilai *loading* sebesar 0,3822. Pada komponen utama kedua, peubah untuk pengeluaran transportasi memberikan kontribusi terbesar terhadap komponen utama kedua, dengan nilai *loading* sebesar 0,5348. Nilai eigen dalam analisis komponen utama merupakan salah satu kriteria untuk menentukan banyaknya komponen utama yang dipilih. Komponen utama yang dipilih harus mempunyai nilai eigen lebih besar satu (λ_k >1).

Banyaknya komponen utama yang mempunyai *eigenvalue* > 1 yang dihasilkan metode CPCA dan ROBPCA dapat dilihat pada Tabel 4.2, tabel selengkapnya dapat dilihat pada Lampiran 10-12.

Tabel 4.2 Banyaknya komponen utama terpilih yang mempunyai *eigenvalue* > 1

No	Data	Banyak komponen dengan *eigenvalue* > 1	
		CPCA	ROBPCA
1	I	7	4
2	II	6	5
3	III	6	5

Sumber: output MATLAB 7.0.1

Keterangan

CPCA : *Classic Principal Component Analysis*

ROBPCA: *Robust Principal Component Analysis*

Berdasarkan Tabel 4.2, dapat diketahui bahwa pada data penelitian I terdapat 7 komponen utama dengan *eigenvalue* lebih besar satu menggunakan metode CPCA, dan menghasilkan 4 komponen utama jika menggunakan ROBPCA. Data penelitian II dan III, CPCA menghasilkan 6 komponen utama dengan $\lambda > 1$ dan ROBPCA menghasilkan 5 komponen utama. Jadi POBPCA menghasilkan komponen utama yang lebih sedikit dari CPCA.

4.4. Proporsi keragaman

Proporsi keragaman merupakan salah satu kriteria penting dalam menentukan komponen utama yang terpilih. Dalam penelitian ini, dipilih k buah komponen utama yang memberikan sumbangan terbesar terhadap total keragaman data, dengan persentase keragaman kumulatif ≥ 80% seperti pada rumus (2.13). Persentase keragaman kumulatif menggunakan metode CPCA dan ROBPCA dapat dilihat pada Tabel 4.3, penjabaran selengkapnya dapat dilihat pada Lampiran 10-12.

Tabel 4.3. Persentase keragaman kumulatif dari *q* buah komponen utama yang terpilih

Data	Proporsi kumulatif (%)		Banyak komponen	
	CPCA	ROBPCA	CPCA	ROBPCA
I	81,85	81,76	7	6
II	81,80	81,16	17	14
III	80,17	80,87	12	12

Sumber: output MATLAB 7.0.1

Keterangan

CPCA : *Classic Principal Component Analysis*

ROBPCA: *Robust Principal Component Analysis*

k : banyaknya komponen utama yang terpilih

Berdasarkan Tabel 4.3, dapat diketahui bahwa pada data penelitan I jika menggunakan metode CPCA terdapat 7 komponen utama terpilih yang mampu menjelaskan keragaman total data sebesar 81,95%, sedangkan ROBPCA menghasilkan 6 komponen utama terpilih yang mampu menjelaskan keragaman total data sebesar 81,76%. Data penelitian II, metode CPCA menghasilkan 17 komponen utama yang mampu menjelaskan total keragaman data sebesar 81,80%, sedangkan ROBPCA menghasilkan 14 komponen utama yang mampu menjelaskan total keragaman data sebesar 81,16%. Data penelitian III, CPCA dan ROBPCA menghasilkan komponen utama yang sama, yaitu 12 komponen utama, tetapi total keragaman yang dapat dijelaskan oleh komponen utama *robust* lebih besar dibandingkan CPCA (80,17%), yaitu 80,87%.

Persentase keragaman kumulatif dengan *q* buah komponen utama terpilih yang sama banyak antara CPCA dan ROBPCA dapat dilihat pada Tabel 4.4.

Tabel 4.4 Persentase keragaman kumulatif dengan q buah komponen utama terpilih yang sama banyak antara CPCA dan ROBPCA

Data	Proporsi kumulatif (%)		Banyak komponen	
	CPCA	ROBPCA	CPCA	ROBPCA
I	75,42	81,76	6	6
II	73,64	81,16	14	14
III	80,17	80,87	12	12

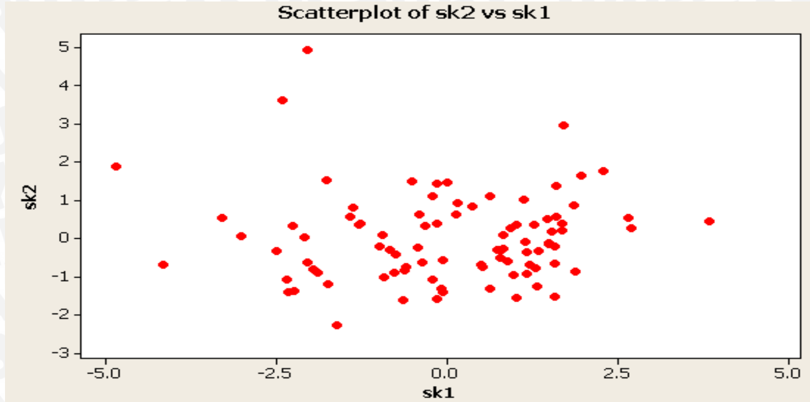
Sumber: output MATLAB 7.0.1

Keterangan
CPCA : *Classic Principal Component Analysis*
ROBPCA: *Robust Principal Component Analysis*

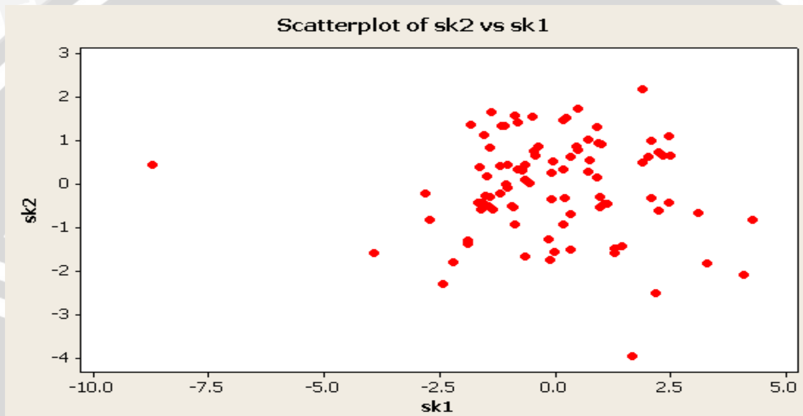
Berdasarkan Tabel 4.4, dapat diketahui bahwa pada data penelitian I dengan jumlah komponen utama yang sama banyak, ROBPCA memberikan kontribusi yang lebih besar terhadap keragaman total data, yaitu sebesar 81,76% daripada CPCA yang hanya memberikan kontribusi sebesar 75,42%. Data penelitian II, dengan jumlah komponen utama yang sama banyak, ROBPCA memberikan kontribusi lebih besar terhadap keragaman total data, yaitu sebesar 81,16% daripada CPCA yang hanya memberikan kontribusi sebesar 73,64% terhadap keragaman total data. Data penelitian III, dapat diketahui bahwa dengan jumlah komponen yang sama banyak, ROBPCA memberikan kontribusi yang lebih besar terhadap keragaman total data, yaitu sebesar 80,87% daripada CPCA yang hanya mampu memberikan kontribusi sebesar 80,17% terhadap keragaman total data. Jadi dengan banyak komponen utama yang lebih sedikit atau sama banyak dari CPCA, ROBPCA telah mampu menjelaskan keragaman total data ≥ 80%.

4.5. Skor Komponen Utama

Skor komponen utama adalah data baru hasil analisis komponen utama yang akan digunakan untuk analisis selanjutnya. Visualisasi antara 2 skor komponen utama yang memberikan kontribusi terbesar terhadap keragaman data, ditunjukkan melalui plot berikut ini:



Gambar 4.3 Plot antara skor komponen utama 1 dan skor komponen utama 2 untuk data I dengan CPCA



Gambar 4.4 Plot antara skor komponen utama 1 dan skor komponen utama 2 untuk data I dengan ROBPCA

Kedua komponen utama tidak saling berkorelasi, maka skor antara kedua komponen utama dapat digambarkan pada bidang datar untuk menentukan *group* antar pengamatan. Berdasarkan Gambar 4 , dapat diketahui bahwa pengelompokan data cenderung lebih dekat dengan komponen utama 1, artinya kumpulan data tersebut lebih mendekati karakteristik dari komponen utama 1. Pengelompokan data pada Gambar 3 tidak begitu kelihatan, karena titik-titik data lebih cenderung memencar, selain itu pada Gambar 3 terdapat lebih banyak observasi yang letaknya jauh dari observasi yang lain, bila dibandingkan dengan Gambar 4. Plot antara skor komponen utama 1 dan skor komponen utama 2 untuk data II dan III dapat dilihat pada Lampiran 30.

4.6. Jarak *Robust*

Jarak *robust* dihitung setelah didapatkan rata-rata dan matriks kovarian *robust* dengan MCD.

Tabel 4.5 Banyaknya pengamatan pencilan pada data penelitian menggunakan jarak mahalanobis *robust*

No	Data	n	p	Banyak pencilan	*Cut off* $\left(\sqrt{\chi^2_{(p;0,975)}}\right)$
1	I	91	10	10	4,5258
2	II	608	26	118	6,4748
3	III	340	20	68	5,8455

Berdasarkan Tabel 4.4 dan *score outlier map* (Lampiran 15) dapat diketahui data I terdapat 10 pengamatan dengan jarak mahalanobis (MD) > *cut off* $\chi^2_{(10;0,975)}$, sehingga 10 pengamatan tersebut teridentifikasi sebagai pencilan. Data II terdapat 118 pengamatan dengan nilai MD > $\chi^2_{(26;0,975)}$, sehingga 118 pengamatan tersebut teridentifikasi sebagai pengamatan pencilan dari 608 pengamatan. Data III terdapat 68 pengamatan yang mempunyai nilai MD > $\chi^2_{(20;0,975)}$, sehingga 68 pemgamatan tersebut teridentifikasi sebagai pencilan.

4.7. Menentukan metode yang lebih baik untuk data penelitian

Berdasarkan kriteria persentase keragaman kumulatif (ukuran yang mampu menjelaskan keragaman total data) ≥ 80% pada data penelitian I, dengan jumlah komponen utama yang sama (6 komponen utama), metode ROBPCA mampu menjelaskan total keragaman data yang lebih besar (81,76%) bila dibandingkan dengan CPCA yang hanya mampu menjelaskan total keragaman sebesar 75,42%. Data penelitian II, dengan jumlah komponen utama yang sama (14 komponen utama), metode ROBPCA mampu menjelaskan total keragaman data lebih besar (81,16%) bila dibandingkan dengan CPCA yang hanya mampu menjelaskan total keragaman sebesar 73,64%. Demikian juga untuk data penelitian III, dengan jumlah komponen utama yang sama (12 komponen utama), metode ROBPCA mampu menjelaskan total keragaman data lebih besar (80,87%) bila dibandingkan dengan CPCA yang hanya mampu menjelaskan total keragaman sebesar 80,17%. Berdasarkan criteria ini, ROBPCA lebih baik digunakan untuk data multivariat mengandung pencilan.

Berdasarkan kriteria (λ_i>1), metode ROBPCA juga menghasilkan jumlah komponen yang lebih sedikit daripada CPCA. Pada data penelitian I, metode ROBPCA menghasilkan 4 komponen dengan nilai eigen yang lebih besar 1, sedangkan CPCA menghasilkan 7 komponen utama, demikian juga untuk data penelitian II dan III, metode ROBPCA menghasilkan 5 komponen utama, sedangkan CPCA menghasilkan 6 komponen utama. Hal ini juga dapat dilihat pada *scree plot* antara *eigenvalue* dan jumlah komponen (Lampiran 21-23). Berdasarkan kriteria (λ_i>1), ROBPCA lebih baik digunakan untuk data multivariat mengandung pencilan.

Cara menentukan jumlah komponen yang dipilih menggunakan *scree plot*, dengan memperhatikan patahan siku dari *scree plot*. Untuk kedua metode, setelah komponen kedua, perubahan *eigenvalue* antar komponen cukup kecil. Dengan menggunakan kriteria *scree plot*, data penelitian I metode ROBPCA dan CPCA sama-sama membutuhkan 2 komponen utama, data penelitian II metode ROBPCA dan CPCA sama-sama membutuhkan 5 komponen utama, sedangkan untuk data penelitian III, kedua metode sama-sama membutuhkan 4 komponen utama. Berdasarkan kriteria *scree plot* kedua metode memberikan kesimpulan yang sama, karena berdasarkan kriteria ini dihasilkan konponen utama yang sama banyak.

Pada Lampiran 16, 18 dan 20, tampak batas *cut off ellipse* dari ROBPCA lebih kecil dibandingkan CPCA. *Ellipse* (Lampiran 16,18 dan 20) yang melebar dipengaruhi mean dari data yang tidak *robust* terhadap pencilan. Hal ini menunjukkan bahwa ROBPCA lebih *robust* dari pada CPCA. Berdasarkan kriteria- kriteria yang telah disebutkan, maka untuk data penelitian yang mengandung pencilan, metode ROBPCA baik digunakan untuk data pencilan karena menghasilkan vektor rata-rata dan matriks kovarian yang *robust* terhadap pencilan, dengan penduga MCD.

BAB V
KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil analisis dalam penelitian ini, dapat ditarik kesimpulan:

1. Berdasarkan kriteria eigenvalue > 1, ROBPCA menghasilkan komponen utama yang lebih sedikit daripada CPCA untuk ketiga data yang digunakan dalam penelitian ini, sedangkan berdasarkan kriteria proporsi keragaman kumulatif ROBPCA menghasilkan komponen utama yang lebih sedikit untuk data I dan II, dan untuk data III ROBPCA dan CPCA menghasilkan komponen utama yang sama banyak. Dengan jumlah jomponen utama yang lebih sedikit atau sama banyak daripada CPCA, ROBPCA telah mampu menjelaskan keragaman total data ≥ 80%.
2. Dengan jumlah komponen utama yang sama, metode ROBPCA memberikan kontribusi lebih besar terhadap total keragaman data daripada metode CPCA (untuk semua data yang digunakan dalam penelitian ini.

5.2. Saran

Untuk penelitian selanjutnya, peneliti disarankan untuk menggunakan data berdimensi tinggi, karena metode ROBPCA selain dapat diterapkan untuk data berdimensi rendah juga dapat diterapkan untuk data berdimensi tinggi. Metode ROBPCA juga dapat dibandingkan dengan metode PCA yang lain seperti: RAPCA (Hubert *et al*, 2003), *spherical* (SPHER) dan *ellipsoidal* (ELL) PCA (Locantore *et al*, 1999). Untuk penelitian lebih lanjut, peneliti disarankan untuk menggunakan regresi ROBPCA untuk mendapatkan kesimpulan yang lebih informatif secara inferensia.

DAFTAR PUSTAKA

Astutik, S.2007. *Diklat Analisis Multivariat*. Malang: Fakultas MIPA Universitas Brawijaya

Debruine, M and M. Hubert. 2003. *The Influence Function of stahel-Donodo Type Methods for Robust PCA* Begium: K.U.Leuven.

Hartanto, E. 2006. *Analisis Pola Konsumsi Pangan pada Rumah Tangga Pedesaan (Studi Kasus Pada Desa Tamansari Kecamatan Karanglewas Kabupaten Banyumas Jawa Tengah)*. Tesis tidak Diterbitkan. Malang: Fakultas MIPA Universitas Brawijaya.

Hubert, M., P. J. Rousseeuw, and K. V. Branden. 2003. "ROBPCA: *A New Approach to Robust Principal Component*". *Technometrics*. http://www.wis.kuleuven.ac.be/stat/ROBPCA.html. 9 Oktober 2008.

Hubert, M., P. J. Rousseeuw, and Stevan. 2007. *High-Breakdown Robust Multivariate Methods*. Belgia: K.U.Leuven. http://www.wis.kuleuven.ac.be/stat/*robust multivatiate*.html. 9 Oktober 2008.

Jollifle, I.T. 2002. *Principal Component Analysis*. U.S.A: Springer-Verlag New York, Inc.

Johnson, R.A., and D. W. Wichern. 2002. *Applied Multivariate Statistical Analysis*. 5 th Ed. New Jersey: Prentice Hall.

Josept, F. H., R. E. Anderson, R. L.Tatham, and W. C. Black. 1998. *Multivariate Data Analysis*. New Jersay: Prentice Hall, Inc.

Li, Y., L. Xu, J. Morphett, and R. Jakobs. 2004. "*An Integrated Algorithm of Incremental and Robust* PCA". *BTexact Technologies*.(diakses 15 Oktober 2008).

Marazzi, A. 1993. *Algorithms, Routines and S functions for Robust Statistics*. California: Wadworth, Inc.

Maronna, R. A., R. D. Martin, and V. J. Yohai. 2006. *Robust Statistics*. John Wiley & Sons Ltd: England.

Sujatmiko, I. 2005.*Analisis Komponen Utama dengan Menggunakan Matriks Varian-Kovarian yang Robust*. Tesis tidak Diterbitkan. Surabaya: Fakultas MIPA *Institude* Teknologi Sepuluh November.

Suryana. 2007.*"Analisis Data Outlier Pada Data Pengeluaran Rumah Tangga Di Kota Kupang Nusa Tenggara Timur Tahun 2005 dengan Metode ROBPCA"*. Paper Diterbitkan. Surabaya: Fakultas MIPA *Institude* Teknologi Sepuluh November.

Verboven, S.,and M. Hubert. 2004. *LIBRA: A MATLAB Library for Robust Analysis*. http://www.wis.kuleuven.ac.be/stat/*robust*.html. 16 Oktober 2008.

Zuo, Y. 2005. *Robust Location and Scatter Estimators in Multivariate Analysis*. Michigan *State Univercity*: *Department of Statistics and Probability*.

Lampiran 1 Data Penelitian I

Data hasil penelitian mahasiswa tentang pola konsumsi rumah tangga Pedesaan (studi kasus pada Desa Tamansari Kecamatan Karanglewas Kabupaten Banyumas Jawa Tengah) berdasarkan pendapatan RT per bulan dalam satuan rupiah.

No	X_1	X_2	X_3	X_4	X_5
1	96200	35400	2800	0	0
2	78500	14000	45000	0	11600
3	97000	10500	22000	5000	11600
4	74100	5000	15000	0	15800
…	....	……….	……….	………	……….
90	93350	9800	76900	93900	70400
91	83600	5000	90300	1E+05	30100

No	X_6	X_7	X_8	X_9	X_{10}
1	10400	0	11000	500	37200
2	14450	22500	18000	25000	38900
3	10000	9000	20900	40000	39900
4	15000	0	22500	25000	77300
…	…….	………	……..	……..	……..
90	19500	46900	27300	1E+05	71350
91	6000	21000	15000	1E+05	47100

Sumber: Hartanto (2006)

Keterangan

X_1 : Konsumsi beras (rupiah/bulan)
X_2 : Konsumsi umbi (rupiah/bulan)
X_3 : konsumsi ikan (rupiah/bulan)
X_4 : Konsumsi daging (rupiah/bulan)
X_5 : Konsumsi telur (rupiah/bulan)
X_6 : Konsumsi kacang-kacangan (rupiah/bulan)
X_7 : Konsumsi buah-buahan (rupiah/bulan)
X_8 : Konsumsi sayur-sayuran (rupiah/bulan)
X_9 : Konsumsi makanan dan minuman jadi (rupiah/bulan)
X_{10} : Konsumsi bahan pamgan lain (rupiah/bulan)

Lampiran 2 Data Penelitian II

Data besarnya pengeluaran konsumsi rumah tangga setiap minggu dalam satuan rupiah untuk 15 peubah pertama dan setiap bulan dalam satuan rupiah untuk 11 peubah berikutnya, yang merupakan hasil survei Sosial Ekonomi Nasional di Kota Kupang tahun 2005.

No	X_1	X_2	X_3	X_4	…….	X_{26}
1	52000	0	2000	2500	…….	0
2	28000	0	7500	0	…….	0
3	43500	0	6000	8000	…….	0
4	29500	0	0	0	…….	0
5	44500	3000	4000	10000	…….	0
6	56000	0	2500	10000	…….	0
7	43000	1000	2500	7500	…….	0
8	33800	0	3000	10000	…….	0
9	22500	1500	2000	4000	…….	0
10	17500	0	3000	0	…….	0
11	50000	1000	2500	12000	…….	0
12	13500	0	0	3000	…….	0
…	………	………	………	………	………	………
608	15000	0	7500	0	…….	0

Sumber: data survei Kota Kupang (2005)

Keterangan

X_1 : Konsumsi beras (rupiah/minggu)
X_2 : Konsumsi umbi (rupiah/minggu)
X_3 : konsumsi ikan (rupiah/minggu)
X_4 : Konsumsi daging (rupiah/minggu)
... : ....
X_{20} : Pengeluaran sewa rumah (rupiah/bulan)
… : ……………………………………………
X_{20} : Pengeluaran pesta (rupiah.bulan)

Lampiran 3 Data Penelitian III

Data besarnya pengeluaran konsumsi rumah tangga setiap minggu dalam satuan rupiah untuk 12 peubah pertama dan setiap bulan dalam satuan rupiah untuk 8 peubah berikutnya, yang merupakan hasil survei Sosial Ekonomi Nasional di Kota Sorong tahun 2002.

No	X_1	X_2	X_3	X_4	……	X_{20}
1	49000	4000	129900	0	……	15000
2	45500	10000	140000	40000	....	83300
3	42000	0	40000	10000	……	13000
4	42000	35000	80000	60000	……	35000
5	35000	10000	70000	0	……	15000
6	38500	5000	70000	0	……	15000
7	35000	5000	90100	20000	……	8000
8	58000	5000	140000	0	……	40000
9	35000	5000	35000	0	……	15000
10	112000	10000	140000	75000	……	500000
11	18000	5000	35000	0	……	20000
12	24000	5000	35000	20000	……	0
…	………..	………	……….	………..	……	………
340	35000	8000	35000	0	……	0

Sumber: data survei Kota Sorong (2002)

Keterangan

X_1 : Konsumsi beras (rupiah/minggu)
X_2 : Konsumsi umbi (rupiah/minggu)
X_3 : konsumsi ikan (rupiah/minggu)
X_4 : Konsumsi daging (rupiah/minggu)
... : ....
X_{13} : Pengeluaran sewa rumah (rupiah/bulan)
… : ……………………………………………
X_{20} : Pengeluaran pesta dan upacara (rupiah/bulan)

Lampiran 4 Hasil *Standardize* Data Penelitian I

No	Z_1	Z_2	Z_3	Z_4	.....	Z_{10}
1	0,62102	3,97621	-0,71982	-1,08218	.....	-0,75003
2	-0,04409	0,67643	-0,0218	-1,08218	.....	-0,69629
3	0,65108	0,13674	-0,40224	-0,92999	.....	-0,66467
4	-0,20943	-0,71133	-0,51803	-1,08218	.....	0,51781
5	0,76756	0,05964	0,22631	-1,08218	.....	-0,72948
6	-0,90366	0,05964	-0,10451	-1,08218	.....	-0,74450
7	-0,14743	0,05964	-0,26991	-1,08218	.....	0,53836
8	0,32416	-0,55714	-0,10451	-1,06392	.....	-1,65112
9	-1,02860	-0,71133	-0,76614	-0,35167	.....	-0,58879
10	1,49655	2,44968	-0,26991	-1,08218	.....	0,17950
...	....	....	....	....	.....	....
90	0,51392	0,02881	0,50585	1,77593	.....	0,32969
91	0,10997	-0,71133	0,72749	2,58254	.....	-0,43703

Sumber: output MATLAB 7.0.1

Keterangan
Z_1 : Nilai hasil standardisasi dari X_1
Z_2 : Nilai hasil standardisasi dari X_2
Z_3 : Nilai hasil standardisasi dari X_3
Z_4 : Nilai hasil standardisasi dari X_4
... : ....
Z_{20} : Nilai hasil standardisasi dari X_{10}

Lampiran 5 Hasil *Standardize* Data Penelitian II

No	Z_1	Z_2	Z_3	Z_4	....	Z_{26}
1	1,26386	-0,45345	-0,99421	-0,50910	....	-0.1906
2	-0,19027	-0,45345	-0,5876	-0,69814	....	-0,1906
3	0,74886	-0,45345	-0,69849	-0,09321	....	-0,1906
4	-0,09938	-0,45345	-1,14207	-0,69814	....	-0,1906
5	0,80945	0,54000	-0,84635	0,05802	....	-0,1906
6	1,50622	-0,45345	-0,95725	0,05802	....	-0,1906
7	0,71857	-0,1223	-0,95725	-0,13102	....	-0,1906
8	0,16115	-0,45345	-0,92028	0,05802	....	-0,1906
....	....	....	....	....	....	....
607	-0,85674	-0,45345	-0,77242	-0,69814	....	-0,1906
608	-0,97792	-0,45345	-0,5876	-0,69814	....	-0,1906

Sumber: output MATLAB 7.0.1

Keterangan
Z_1 : Nilai hasil standardisasi dari X_1
Z_2 : Nilai hasil standardisasi dari X_2
Z_3 : Nilai hasil standardisasi dari X_3
Z_4 : Nilai hasil standardisasi dari X_4
... : ....
Z_{20} : Nilai hasil standardisasi dari X_{26}

Lampiran 6 Hasil *Standardize* Data Penelitian III

No	Z_1	Z_2	Z_3	Z_4	......	Z_{20}
1	0,29458	-0,38386	1,91619	-0,43173	......	-0,37808
2	0,14540	0,25617	2,17209	1,42399	......	0,18118
3	-0,00377	-0,81055	-0,36151	0,03220	......	-0,39445
4	-0,00377	2,92297	0,65193	2,35184	......	-0,21431
5	-0,30213	0,25617	0,39857	-0,43173	......	-0,37808
6	-0,15295	-0,27719	0,39857	-0,43173	......	-0,37808
7	-0,30213	-0,27719	0,90782	0,49613	......	-0,43539
8	0,67818	-0,27719	2,17209	-0,43173	......	-0,17337
9	-0,30213	-0,27719	-0,48819	-0,43173	......	-0,37808
10	2,97978	0,25617	2,17209	3,04773	......	3,59319
...	....	....	....	......	......	....
339	0,59294	0,25617	-1,12158	0,68170	......	1,95555
340	-0,30213	0,04283	-0,48819	-0,43173	......	-0,50090

Sumber: output MATLAB 7.0.1

Keterangan
Z_1 : Nilai hasil standardisasi dari X_1
Z_2 : Nilai hasil standardisasi dari X_2
Z_3 : Nilai hasil standardisasi dari X_3
Z_4 : Nilai hasil standardisasi dari X_4
... : ....
Z_{20} : Nilai hasil standardisasi dari X_{20}

Lampiran 7 *Rank* Data Penelitian I Sebelum dan Sesudah *Standardize*.

Rank data sebelum *standardize*

Peubah	Max	Min	Rank
X_1	184400	34350	150050
X_2	35400	500	34900
X_3	554900	0	554900
X_4	152500	0	152500
X_5	117600	0	117600
X_6	105000	0	105000
X_7	102500	0	102500
X_8	51900	6300	45600
X_9	143200	0	143200
X_{10}	169650	4850	164800

Rank data setelah *standardize*

Peubah	Max	Min	Rank
X_1	3,93527	-1,70310	5,63837
X_2	3,97621	-1,40521	5,38142
X_3	8,41233	-0,76614	9,17847
X_4	3,55959	-1,08218	4,64177
X_5	3,43045	-1,51827	4,94872
X_6	7,46682	-1,23454	8,70136
X_7	3,99863	-1,24940	5,24803
X_8	3,80359	-1,75227	5,55586
X_9	2,80097	-1,61199	4,41296
X_{10}	3,43763	-1,77284	5,21047

Sumber: output MATLAB 7.0.1

Lampiran 8 *Rank* Data Penelitian II Sebelum dan Sesudah *Standardize*.

Rank data sebelum *standardize*

Peubah	Max	Min	Rank
X_1	154000	3000	151000
X_2	20000	0	20000
X_3	91000	0	91000
X_4	91000	0	91000
X_5	88500	0	88500
X_6	70000	0	70000
X_7	45000	0	45000
X_8	54000	0	54000
X_9	77500	0	77500
X_{10}	185000	0	185000
X_{11}	65000	0	65000
X_{12}	40000	0	40000
X_{13}	100000	0	100000
X_{14}	20000	0	20000
X_{15}	84000	0	84000
X_{16}	1000000	7500	992500
X_{17}	2550000	2500	2547500
X_{18}	300000	0	300000
X_{19}	860000	1000	859000
X_{20}	850000	0	850000
X_{21}	3020000	0	3020000
X_{22}	750000	0	750000
X_{23}	1245000	0	1245000
X_{24}	150000	0	150000
X_{25}	150000	0	150000
X_{26}	750000	0	750000

Lampiran 8 (lanjutan)

Rank data setelah *standardize*

Peubah	Max	Min	Rank
X_1	7,44392	-1,70499	9,14891
X_2	6,16958	-0,45345	6,62303
X_3	5,58560	-1,14207	6,72767
X_4	6,18292	-0,69814	6,88106
X_5	6,56205	-0,71767	7,27972
X_6	6,84809	-1,63653	8,48462
X_7	7,40125	-0,88009	8,28134
X_8	8,26410	-0,54680	8,81090
X_9	11,87980	-1,02140	12,90120
X_{10}	19,71570	-0,99750	20,71320
X_{11}	16,61670	-0,98720	17,60390
X_{12}	8,19957	-0,56688	8,76645
X_{13}	9,18025	-0,70506	9,88531
X_{14}	14,29490	-0,09650	14,39140
X_{15}	5,46944	-0,62834	6,09778
X_{16}	6,39222	-0,76500	7,15722
X_{17}	20,12800	-0,56190	20,68990
X_{18}	16,38270	-0,14580	16,52850
X_{19}	9,67547	-0,72844	10,40391
X_{20}	8,92743	-0,47649	9,40392
X_{21}	15,79890	-0,23760	16,03650
X_{22}	10,77640	-0,54010	11,31650
X_{23}	16,75100	-0,26890	17,01990
X_{24}	13,89810	-0,49710	14,39520
X_{25}	11,94820	-0,30600	12,25420
X_{26}	16,43870	-0,19060	16,62930

Sumber: output MATLAB 7.0.1

Lampiran 9 *Rank* Data Penelitian III Sebelum dan Sesudah *Standardize*.

Rank data sebelum *standardize*

Peubah	Max	Min	Rank
X_1	140000	7000	133000
X_2	70000	0	70000
X_3	175000	0	175000
X_4	175000	0	175000
X_5	193000	0	193000
X_6	210000	0	210000
X_7	93100	0	93100
X_8	105000	0	105000
X_9	112000	0	112000
X_{10}	124000	0	124000
X_{11}	50000	0	50000
X_{12}	70000	0	70000
X_{13}	256000	0	256000
X_{14}	210000	0	210000
X_{15}	2000000	0	2000000
X_{16}	7618000	0	7618000
X_{17}	3220000	38000	3182000
X_{18}	4000000	0	4000000
X_{19}	7660600	0	7660600
X_{20}	1000000	0	1000000

Lampitan 9 (lanjutan)

Rank data setelah *standardize*

Peubah	Max	Min	Rank
X_1	4,17320	-1,49555	5,66875
X_2	6,65649	-0,81055	7,46704
X_3	3,05884	-1,37494	4,43378
X_4	7,68701	-0,43173	8,11874
X_5	5,97876	-0,92385	6,90261
X_6	8,59066	-1,22567	9,81633
X_7	8,08026	-0,72603	8,80629
X_8	7,55409	-0,68596	8,24005
X_9	8,95262	-1,21987	10,17249
X_{10}	8,86486	-1,10711	9,97197
X_{11}	7,66645	-1,22771	8,89416
X_{12}	7,08507	-0,82523	7,91030
X_{13}	6,62990	-0,81485	7,44475
X_{14}	5,59368	-0,66329	6,25697
X_{15}	5,67339	-0,64193	6,31532
X_{16}	3,50431	-1,47038	4,97469
X_{17}	7,34950	-1,49292	8,84242
X_{18}	9,29112	-0,47305	9,76417
X_{19}	5,77926	-1,05827	6,83753
X_{20}	7,68727	-0,50090	8,18817

Sumber: output MATLAB 7.0.1

Lampiran 10 Nilai Eigen, Proporsi Keragaman dan Proporsi Keragaman Kumulatif Menggunakan CPCA dan ROBPCA untuk Data Penelitian I

	CPCA		
KU	Eigenvalue	Proporsi keragaman	Proporsi kumulatif
1	**5,3859**	0,340590891	0,340590891
2	**1,5624**	0,098802282	0,439393173
3	**1,4976**	0,094704491	0,534097664
4	**1,2291**	0,077725220	0,611822884
5	**1,1863**	0,075018655	0,686841539
6	**1,0647**	0,067328974	0,754170514
7	**1,0325**	0,065292726	**0,819463240**
8	0,9849	0,062282621	0,881745861
9	0,9431	0,059639293	0,941385154
10	0,9269	0,058614846	1,000000000

ROBPCA			
No	Eigenvalue	Proporsi keragaman	Proporsi kumulatif
1	4.3834	0.375008555	0.375008555
2	1.2296	0.105194716	0.480203272
3	1.1796	0.100917117	0.581120389
4	1.0343	0.088486414	0.669606803
5	0.9113	0.077963521	0.747570324
6	0.8188	0.070049962	0.817620286
7	0.7203	0.061623092	0.879243378
8	0.6788	0.058072685	0.937316063
9	0.5783	0.049474711	0.986790774
10	0.1544	0.013209226	1

Sumber: output MATLAB 7.0.1

Keterangan
KU : Komponen Utama
CPCA : *Classic Principal Component analysis*
ROBPCA : *Robust Principal Component Analysis*

Lampiran 11 Nilai Eigen, Proporsi Keragaman dan Proporsi Keragaman Kumulatif Menggunakan CPCA dan ROBPCA untuk Data Penelitian II

CPCA			
KU	Eigenvalue	Proporsi keragaman	Proporsi kumulatif
1	**5,3859**	0,207150000	0,207150000
2	**1,5624**	0,060092308	0,267242308
3	**1,4976**	0,057600000	0,324842308
4	**1,2291**	0,047273077	0,372115385
5	**1,1863**	0,045626923	0,417742308
6	**1,0647**	0,040950000	0,458692308
7	1,0325	0,039711538	0,498403846
8	0,9849	0,037880769	0,536284615
9	0,9431	0,036273077	0,572557692
10	0,9269	0,035650000	0,608207692
11	0,8806	0,033869231	0,642076923
12	0,8780	0,033769231	0,675846154
13	0,8055	0,030980769	0,706826923
14	0,7702	0,029623077	0,736450000
15	0,7188	0,027646154	0,764096154
16	0,7077	0,027219231	0,791315385
17	0,6932	0,026661538	**0,817976923**
18	0,6544	0,025169231	0,843146154
19	0,6151	0,023657692	0,866803846
20	0,5867	0,022565385	0,889369231
21	0,5489	0,021111538	0,910480769
22	0,5270	0,020269231	0,930750000
23	0,4815	0,018519231	0,949269231
24	0,4739	0,018226923	0,967496154
25	0,4355	0,016750000	0,984246154
26	0,4096	0,015753846	1,000000000

Lampitan 11 (lanjutan)

	ROBPCA		
KU	Eigenvalue	Proporsi keragaman	Proporsi kumulatif
1	**5,2186**	0,252666541	0,252666541
2	**1,4291**	0,069192073	0,321858614
3	**1,2186**	0,059000392	0,380859006
4	**1,1155**	0,054008647	0,434867653
5	**1,0418**	0,050440348	0,485308002
6	0,9757	0,047240015	0,532548017
7	0,9276	0,044911180	0,577459197
8	0,8704	0,042141754	0,619600951
9	0,7527	0,036443128	0,656044078
10	0,7412	0,035886337	0,691930416
11	0,6640	0,032148581	0,724078996
12	0,6246	0,030240969	0,754319966
13	0,6127	0,029664812	0,783984778
14	0,5713	0,027660368	**0,811645146**
15	0,5471	0,026488687	0,838133833
16	0,4836	0,023414237	0,861548070
17	0,4542	0,021990791	0,883538862
18	0,4354	0,021080560	0,904619422
19	0,4180	0,020238113	0,924857534
20	0,3869	0,018732358	0,943589893
21	0,2919	0,014132787	0,957722680
22	0,2385	0,011547344	0,969270024
23	0,1805	0,008739185	0,978009209
24	0,1589	0,007693388	0,985702597
25	0,1556	0,007533613	0,993236210
26	0,1397	0,006763790	1,000000000

Sumber: output MATLAB 7.0.1

Lampiran 12 Nilai Eigen, Proporsi Keragaman dan Proporsi Keragaman Kumulatif Menggunakan CPCA dan ROBPCA untuk Data Penelitian III

	CPCA		
KU	Eigenvalue	Proporsi keragaman	Proporsi kumulatif
1	**3,9617**	0,198085	0,198085
2	**1,9879**	0,099395	0,297480
3	**1,5241**	0,076205	0,373685
4	**1,1713**	0,058565	0,432250
5	**1,1334**	0,056670	0,488920
6	**1,0571**	0,052855	0,541775
7	0,9844	0,049220	0,590995
8	0,9484	0,047420	0,638415
9	0,9358	0,046790	0,685205
10	0,8612	0,043060	0,728265
11	0,7770	0,038850	0,767115
12	0,6921	0,034605	**0,801720**
13	0,6345	0,031725	0,833445
14	0,5943	0,029715	0,863160
15	0,5368	0,026840	0,890000
16	0,5235	0,026175	0,916175
17	0,4911	0,024555	0,940730
18	0,4379	0,021895	0,962625
19	0,3855	0,019275	0,981900
20	0,3620	0,018100	1,000000

Lampiran 12 (lanjutan)

	ROBPCA		
KU	Eigenvalue	Proporsi keragaman	Proporsi kumulatif
1	**3,3797**	0,196882227	0,196882227
2	**1,7535**	0,102149003	0,299031232
3	**1,1413**	0,066485690	0,365516922
4	**1,0816**	0,063007905	0,428524825
5	**1,0206**	0,059454390	0,487979215
6	0,9423	0,054893074	0,542872289
7	0,9074	0,052859997	0,595732286
8	0,8870	0,051671609	0,647403895
9	0,7911	0,046085016	0,693488911
10	0,7045	0,041040190	0,734529101
11	0,6433	0,037475023	0,772004124
12	0,6298	0,036688590	**0,808692714**
13	0,5521	0,032162227	0,840854941
14	0,5203	0,030309738	0,871164679
15	0,4296	0,025026069	0,896190748
16	0,4214	0,024548383	0,920739131
17	0,3907	0,022759975	0,943499106
18	0,3684	0,021460903	0,964960008
19	0,3310	0,019282190	0,984242198
20	0,2705	0,015757802	1,000000000

Sumber: output MATLAB 7.0.1

Lampiran 13 *Loadding* Komponen Utama ($\widetilde{b}_k$) Menggunakan ROBPCA untuk Data I

PC1	PC2	PC3	PC4	PC5
0.0832	-0.7699	0.1913	-0.2859	0.2386
-0.3287	-0.3639	-0.3666	-0.3992	-0.3148
0.1305	-0.0895	-0.0381	0.0204	-0.0436
0.4780	0.0475	-0.1069	-0.0596	-0.5816
0.4123	0.1780	-0.1208	-0.4106	0.6154
0.0202	-0.2344	0.0740	0.2777	0.1324
0.4355	-0.2406	0.1837	-0.0786	-0.2792
0.1768	-0.3376	-0.1926	0.7087	0.1418
0.4878	0.0679	0.0165	-0.0224	-0.0188
-0.1011	0.0381	0.8515	-0.0189	-0.0957

PC6	PC7	PC8	PC9	PC10
-0.3608	-0.0520	-0.2459	-0.1776	0.0135
0.5938	0.1145	-0.0012	0.0121	0.0267
0.0506	-0.1182	0.1098	0.0228	-0.9706
-0.2141	0.4840	-0.3676	0.0180	-0.0223
0.2335	0.4122	0.0972	0.0484	-0.0183
0.1401	0.0580	-0.2436	0.8729	0.0144
0.0221	-0.1392	0.7377	0.1855	0.1903
0.2757	0.2758	0.0846	-0.3652	0.0528
0.4144	-0.6261	-0.4066	-0.1246	0.1080
0.3826	0.2744	-0.0836	-0.1408	-0.0729

Sumber: output MATLAB 7.0.1

Keterangan

KU : Komponen Utama
0,0832 : Nilai *loading* untuk peubah 1 pada KU/PC 1
-0,3287 : Nilai *loading* untuk peubah 2 pada KU/PC 1
-0,7699 : Nilai *loading* untuk peubah 1 pada KU/PC 2
PC1 : *Principal Component* 1/komponen utama pertama
PC2 : *Principal Component* 2/komponen utama kedua
.....
PC10 : *Principal Component* 10/komponen utama kesepuluh

Lampiran 14 *Loadding* Komponen Utama ($\widetilde{b}_k$) Menggunakan ROBPCA untuk Data II

PC1	PC2	PC3	PC4	PC5	PC6
0.2413	-0.0960	0.3382	-0.2358	-0.0241	0.2177
0.2294	-0.3520	-0.1768	0.0400	0.0445	-0.0146
0.2513	0.1707	0.2492	-0.2583	-0.0245	0.2203
0.2543	0.2051	-0.1289	0.1439	0.2259	0.2667
0.2967	-0.1902	-0.0579	0.0300	-0.0287	-0.0088
0.3107	0.0953	0.1518	-0.1976	-0.1427	-0.0702
0.2733	-0.0538	0.0141	-0.1740	-0.1158	-0.0942
0.2524	0.1176	-0.0855	0.0134	0.2658	0.2585
0.1529	-0.0157	0.1288	-0.1168	-0.0426	0.0064
0.1361	-0.0204	0.1294	-0.1367	-0.0780	0.0355
0.1552	0.0020	0.1820	-0.1257	-0.0094	0.0361
0.2604	-0.2366	-0.1003	-0.0544	0.0238	-0.1321
0.2255	-0.2422	-0.1597	0.0767	0.0276	0.0662
-0.0078	0.0108	-0.0825	-0.4583	0.6936	-0.5097
0.0878	0.2343	-0.4046	-0.4425	-0.3957	-0.0994
0.2150	0.4314	0.0138	0.3103	0.0688	-0.2429
0.1357	0.1725	0.0328	0.1206	0.0801	0.0224
0.0182	-0.0301	-0.0648	-0.0545	-0.0239	0.0487
0.2238	0.2172	0.1281	0.2452	0.1948	0.0416
0.0969	-0.1106	0.4987	0.2137	-0.2263	-0.5517
0.0171	0.0067	0.0284	-0.0289	-0.0332	0.0103
0.2123	-0.3810	-0.2404	0.2181	-0.0213	-0.0906
0.1167	-0.1876	-0.0502	0.1176	0.0789	0.0518
0.1877	0.1692	-0.1116	0.1916	0.0255	-0.1815
0.1601	0.2649	-0.3530	0.0541	-0.2865	-0.2133
0.0419	-0.0891	-0.0221	-0.0247	-0.0460	-0.0228

Keterangan

PC/KU : *Principal Component/*Komponen Utama
0,2413 : Nilai *loading* untuk peubah 1 pada KU/PC 1
0,2294 : Nilai *loading* untuk peubah 2 pada KU/PC 1
-0,0960 : Nilai *loading* untuk peubah 1 pada KU/PC 2

Lampiran 14 (Lanjutan)

PC7	PC8	PC9	PC10	PC11	PC12
-0.0197	0.1411	0.2780	0.3956	-0.2863	0.0588
0.2939	0.1638	0.0424	0.2398	-0.0421	0.1123
-0.1485	0.2312	-0.1140	0.0422	0.4892	-0.0261
-0.2441	-0.0458	0.1529	-0.0583	-0.4718	0.2346
0.1281	-0.1088	-0.2587	-0.2259	-0.2145	0.0566
0.1595	0.1068	0.0215	-0.0956	-0.2519	-0.2824
0.4077	0.0098	-0.3063	-0.1199	0.1975	0.0980
-0.3493	-0.0484	-0.1825	-0.3548	0.2088	-0.0844
0.0071	0.0355	0.0195	0.0331	-0.0235	-0.0467
-0.0611	-0.0794	0.0287	0.0133	0.0003	-0.0378
-0.0162	-0.0965	-0.0406	-0.0278	-0.0155	-0.0453
0.0359	-0.0065	0.1785	-0.4773	-0.0208	0.3930
-0.1903	-0.5205	-0.3628	0.4419	0.0233	-0.0931
-0.0895	0.0507	-0.0146	0.1162	-0.0333	-0.0568
-0.1082	-0.4181	0.3433	0.0409	0.1152	0.0681
0.3700	-0.1564	0.1376	0.0280	0.0503	-0.2146
-0.0125	-0.0520	0.0912	0.0283	-0.0074	-0.0179
-0.0149	-0.0219	0.0040	-0.2441	-0.2528	-0.6752
0.0784	-0.0537	0.1906	0.1193	0.2409	0.1179
-0.4188	-0.1537	-0.0233	-0.0746	-0.0433	0.0691
-0.0101	-0.0258	0.0239	0.0097	-0.0550	-0.1126
-0.1889	0.2437	0.4504	0.0109	0.2883	-0.2772
0.0071	-0.0437	-0.0825	0.0750	0.0939	-0.1812
0.0550	0.0069	-0.0220	0.1528	0.0017	0.0436
-0.2729	0.5381	-0.3332	0.1728	-0.1417	0.0510
-0.0896	0.0348	0.1407	-0.0085	0.0685	-0.0643

Keterangan

PC/KU : *Principal Component/*Komponen Utama
-0,0197 : Nilai *loading* untuk peubah 1 pada KU/PC 1
0,2939 : Nilai *loading* untuk peubah 2 pada KU/PC 1
0,1411 : Nilai *loading* untuk peubah 1 pada KU/PC 2

Lampiran 14 (lanjutan)

PC13	PC14	PC15	PC16	PC17	PC18
0.0315	-0.1430	0.3359	-0.1896	0.1099	0.0707
-0.6025	0.3797	0.0886	0.1530	0.0411	-0.1363
-0.0763	0.0770	-0.3798	0.0843	0.1628	-0.3734
0.2865	0.3604	-0.1859	0.3006	0.0080	-0.1133
-0.1262	0.0533	-0.5111	-0.4544	0.0503	0.2864
-0.0188	-0.1441	-0.0530	0.0989	-0.4903	0.1103
0.4467	0.2645	0.2951	0.3063	0.0661	0.2085
-0.2594	0.1251	0.4901	-0.2036	-0.1869	0.1398
0.0725	0.0082	-0.0617	-0.1249	-0.0371	-0.0223
0.0832	0.0007	0.0060	-0.1385	0.0859	-0.0250
0.0209	0.0069	-0.0865	-0.0562	-0.0058	-0.1138
-0.0117	-0.5161	0.0994	0.1012	0.1156	-0.3420
0.0418	-0.2814	0.0018	0.1933	-0.1736	-0.1753
0.0334	0.0145	-0.0772	-0.0006	0.0028	0.0455
-0.1129	0.1440	-0.0197	-0.0258	0.0916	0.1312
-0.0415	0.0145	0.0309	-0.1020	-0.1939	-0.3631
-0.0163	0.0334	0.0812	-0.1025	0.0227	0.0259
-0.1469	-0.0672	0.0505	0.2903	0.4973	-0.0402
-0.1913	-0.3217	-0.1577	0.3185	0.1745	0.5717
-0.1162	0.2472	0.0607	0.1296	0.0700	0.0324
-0.0033	-0.0449	0.0011	-0.0093	0.0097	0.0318
0.2743	0.0944	-0.1128	-0.0262	-0.1797	0.0681
0.2547	0.0009	-0.0072	-0.1344	0.2520	0.0586
0.1430	0.0037	0.1663	-0.3897	0.4279	-0.1067
-0.0396	-0.2002	0.0287	0.0553	0.0062	0.0578
0.0294	-0.0478	0.0415	-0.0904	-0.1304	0.0099

Keterangan

PC/KU : *Principal Component*/Komponen Utama
0,0315 : Nilai *loading* untuk peubah 1 pada KU/PC 1
-0,6025 : Nilai *loading* untuk peubah 2 pada KU/PC 1
-0,1430 : Nilai *loading* untuk peubah 1 pada KU/PC 2

Lampiran 14 (lanjutan)

PC19	PC20	PC21	PC22	PC23	PC24
-0.3119	-0.0682	-0.2659	0.0004	-0.0049	0.0618
0.1295	-0.0477	0.1238	-0.0163	0.0150	-0.0531
0.0093	0.0422	-0.2327	-0.0074	-0.0105	0.0173
0.0437	-0.0144	-0.0170	-0.0723	-0.0688	-0.0497
-0.2668	0.0624	-0.1600	-0.0250	-0.0124	0.0168
0.5238	0.0206	-0.1803	0.0678	0.0817	-0.0868
-0.1776	0.1048	-0.0149	-0.0553	0.0097	0.0330
-0.0026	-0.0317	-0.0468	0.0330	-0.0724	0.0045
0.1088	0.0003	0.5233	0.0964	-0.6475	0.4398
-0.0686	0.0584	0.4185	0.3023	-0.0433	-0.7619
0.0177	-0.0529	0.4529	-0.2741	0.5473	0.1477
0.0077	-0.0718	-0.0042	0.0467	0.0053	0.0385
-0.0355	0.1522	-0.0423	0.0223	-0.0351	0.0270
-0.0533	-0.0113	-0.0090	0.0070	0.0098	-0.0150
0.0256	-0.1282	-0.0682	0.0136	0.0025	0.0639
-0.3475	-0.2002	-0.0619	-0.0196	-0.1257	-0.0613
-0.0545	0.0805	0.2662	0.1548	0.4429	0.2477
-0.1205	0.1183	0.0068	-0.0432	-0.0564	0.0147
0.0456	0.0130	0.0917	-0.0540	-0.0616	-0.0573
0.0049	-0.0658	-0.0953	0.0019	-0.0290	0.0256
-0.0636	0.0592	-0.0006	0.0251	0.0994	0.2602
-0.1216	0.1681	-0.0235	0.1695	0.0851	0.0489
0.2825	-0.7871	-0.0430	-0.0421	0.0288	-0.0249
0.4647	0.4225	-0.1233	-0.0655	-0.0029	-0.0103
-0.1702	-0.1474	0.1378	-0.0114	0.0393	-0.0191
-0.0355	0.0842	0.0974	-0.8612	-0.1464	-0.1975

Keterangan

PC/KU : *Principal Component/*Komponen Utama
-0,3119 : Nilai *loading* untuk peubah 1 pada KU/PC 1
0,1295 : Nilai *loading* untuk peubah 2 pada KU/PC 1
-0,0682 : Nilai *loading* untuk peubah 1 pada KU/PC 2

Lampiran 14 (lanjutan)

PC25	PC26
-0.0848	-0.0641
0.0675	0.0240
0.0134	0.1169
0.0531	-0.0171
-0.0387	0.0325
-0.0243	0.0667
0.0040	0.0362
0.0211	-0.1190
-0.0662	0.0093
0.1732	0.0684
-0.0823	-0.5251
0.0240	0.0536
-0.0379	0.0388
0.0078	0.0228
-0.0313	-0.0178
0.0179	-0.1006
-0.1619	0.7136
-0.1127	0.0149
0.0223	-0.1010
-0.0029	0.0262
0.9430	0.0376
-0.0455	-0.1386
0.0545	0.1119
0.0230	-0.1247
0.0054	-0.0295
0.0621	0.3078

Sumber: output MATLAB 7.0.1

Keterangan

-0,0848 : Nilai *loading* untuk peubah 1 pada KU/PC 1
0,0675 : Nilai *loading* untuk peubah 2 pada KU/PC 1
-0,0641 : Nilai *loading* untuk peubah 1 pada KU/PC 2
PC1 : *Principal Component* 1/komponen utama pertama
.....
PC26 : *Principal Component* 26/komponen utama ke-26

Lampiran 15 *Loadding* Komponen Utama ($\widetilde{b}_k$) Menggunakan ROBPCA untuk Data III

PC1	PC2	PC3	PC4	PC5	PC6
0.2509	-0.1443	0.1484	0.4329	0.1767	-0.1472
0.1848	-0.2009	0.1679	0.0541	0.3362	-0.4234
0.3822	-0.0803	0.1852	0.1975	-0.0626	0.2099
0.2553	-0.0512	-0.2797	-0.2087	-0.0132	0.0517
0.2958	0.0129	-0.4281	-0.1257	-0.0665	-0.0014
0.2997	-0.0619	-0.2008	-0.0023	0.0834	0.1012
0.1466	-0.0551	-0.0802	-0.0874	0.1253	-0.3230
0.2755	-0.0372	-0.2385	-0.1997	-0.0304	0.2175
0.2565	-0.1002	0.0641	0.0424	0.0750	-0.2501
0.1772	-0.0950	0.0705	0.0732	0.0940	-0.1890
0.2647	-0.0055	0.2968	0.0955	-0.3876	0.2469
0.2392	-0.1226	-0.1420	0.0520	0.1721	0.2267
0.3067	-0.0655	0.1066	-0.1760	-0.0770	0.0358
0.1170	-0.0523	0.3563	-0.2742	-0.4427	-0.2172
0.0085	0.3467	-0.0909	-0.1154	0.3454	-0.0939
0.2062	0.5348	0.0520	-0.2252	-0.0403	-0.0405
0.1661	0.4805	0.1465	-0.1358	-0.0109	-0.3005
0.0412	0.2280	0.0466	0.0982	0.3042	0.2541
0.1124	0.3979	0.1714	0.4281	0.0675	0.2241
0.0141	0.1811	-0.4803	0.5131	-0.4575	-0.3402

Keterangan

PC/KU : *Principal Component*/Komponen Utama
0,2509 : Nilai *loading* untuk peubah 1 pada KU/PC 1
0,1848 : Nilai *loading* untuk peubah 2 pada KU/PC 1
-0,1443 : Nilai *loading* untuk peubah 1 pada KU/PC 2

Lampiran 15 (lanjutan)

PC7	PC8	PC9	PC10	PC11	PC12
0.4221	-0.0001	-0.1261	0.0659	-0.3178	-0.2781
-0.3859	0.0084	0.1572	0.0742	-0.2642	0.2689
-0.0482	0.2336	-0.2310	-0.0505	-0.0448	-0.0341
-0.0200	-0.2493	0.0922	-0.1455	-0.2021	-0.0022
-0.0147	-0.0265	0.0781	0.3585	-0.3205	-0.3653
0.1153	-0.0270	-0.1566	-0.0859	0.2854	0.1518
-0.1422	0.2097	0.4290	0.3554	0.4175	-0.1624
-0.1130	-0.0490	-0.0614	-0.2072	0.2288	-0.1373
0.0364	-0.1536	-0.1253	-0.2901	0.4041	-0.1986
-0.0176	-0.0492	-0.0832	-0.1804	0.1857	-0.1188
-0.2673	0.5026	0.1050	0.0595	0.0181	-0.0644
0.4065	0.0654	0.0671	0.3267	0.0724	0.5157
-0.2352	-0.1974	0.1371	-0.2991	-0.2836	0.3366
0.5076	-0.1426	0.3655	-0.0532	0.0398	0.0499
0.2247	0.4725	0.1958	-0.4903	-0.1853	-0.0911
0.0403	0.0185	-0.0207	0.1330	-0.1004	-0.0888
-0.0372	-0.0444	-0.5039	0.2289	0.0824	0.2312
0.0391	0.0675	0.2820	-0.0225	0.1877	0.2026
-0.1306	-0.5025	0.3270	0.0304	0.0583	-0.1512
-0.0177	0.1026	0.0968	-0.1737	0.0188	0.2729

Keterangan

PC/KU : *Principal Component/*Komponen Utama
0,4221 : Nilai *loading* untuk peubah 1 pada KU/PC 1
-0,3859 : Nilai *loading* untuk peubah 2 pada KU/PC 1
-0,0001 : Nilai *loading* untuk peubah 1 pada KU/PC 2

Lampiran 15 (lanjutan)

PC13	PC14	PC15	PC16
-0.0217	-0.0748	0.2916	0.1062
0.1245	0.1965	-0.4185	-0.0418
-0.0160	-0.2698	-0.3514	0.2488
0.2703	0.1449	0.1160	0.0289
0.1950	-0.1816	-0.1165	-0.1498
-0.3869	-0.3129	-0.3404	-0.3049
-0.2507	-0.1740	0.2150	0.2484
0.0921	0.2287	-0.0576	0.5080
0.2402	0.2380	0.0185	-0.3283
-0.0189	0.0110	0.2809	-0.1382
0.1128	0.2522	0.1585	-0.1903
-0.0616	0.4320	0.1134	0.0164
-0.2787	-0.2822	0.4336	0.0078
0.1206	-0.0924	-0.2635	0.1022
-0.1095	0.1038	-0.0859	0.1428
-0.2383	0.1676	-0.0076	-0.3989
0.1913	-0.0766	0.1095	0.2983
0.5890	-0.4278	0.0980	-0.1573
-0.1733	0.1595	-0.1420	0.1475
0.0506	-0.0138	-0.0040	0.0137

Keterangan

PC/KU : *Principal Component*/Komponen Utama
-0,0217 : Nilai *loading* untuk peubah 1 pada KU/PC 1
0,1245 : Nilai *loading* untuk peubah 2 pada KU/PC 1
-0,0748 : Nilai *loading* untuk peubah 1 pada KU/PC 2

Lampiran 15 (lanjutan)

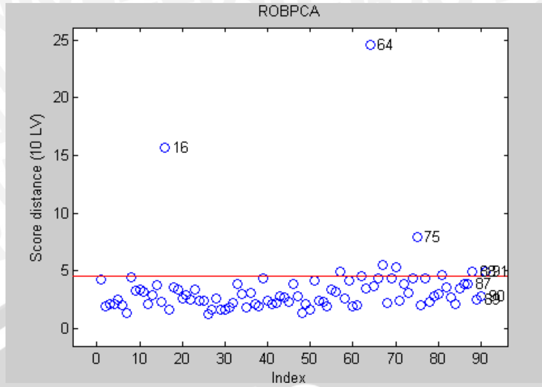
PC17	PC18	PC19	PC20
-0.1120	-0.3670	-0.0693	-0.1511
0.0209	-0.1994	-0.0772	0.0602
0.2086	0.2524	0.4944	0.0125
-0.5787	0.0154	0.4748	0.0747
0.1925	0.2914	-0.3252	0.0492
-0.3937	-0.1663	-0.2412	0.0334
-0.0970	0.0237	0.2027	-0.1182
0.2507	-0.4638	-0.2107	0.0123
0.1403	0.2631	0.0052	-0.4595
0.1485	0.0961	-0.0010	0.8271
-0.2928	-0.0291	-0.2322	0.0132
0.1555	0.2084	0.0021	0.0277
0.1994	0.1076	-0.1336	-0.2031
0.0104	-0.0267	-0.0761	0.1009
-0.0885	0.2413	-0.1353	0.0063
0.2704	-0.3713	0.3486	-0.0100
-0.2078	0.1398	-0.1876	0.0011
0.0938	-0.2043	0.0104	0.0099
-0.1036	0.1853	-0.1145	0.0485
0.0952	-0.0819	0.0639	-0.0024

Sumber: output MATLAB 7.0.1

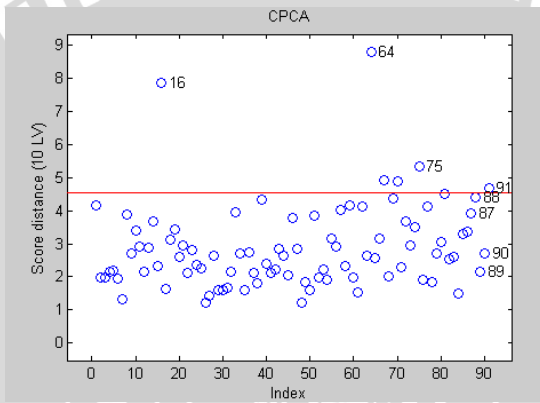
Keterangan

-0,1120 : Nilai *loading* untuk peubah 1 pada KU/PC 1
0,0209 : Nilai *loading* untuk peubah 2 pada KU/PC 1
-0,3670 : Nilai *loading* untuk peubah 1 pada KU/PC 2
PC1 : *Principal Component* 1/komponen utama pertama
PC2 : *Principal Component* 1/komponen utama kedua
.....
PC20 : *Principal Component* 20/komponen utama ke-20

Lampiran 16 *Score Outlier Map*/Jarak Mahalanobis Hasil Analisis Menggunakan CPCA dan ROBPCA untuk Data Penelitian I



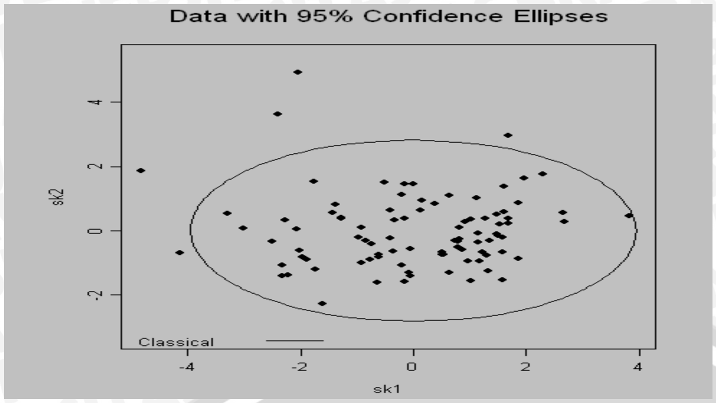
Plot nilai jarak mahalanobis (*Scor outlier map*) untuk ROBPCA

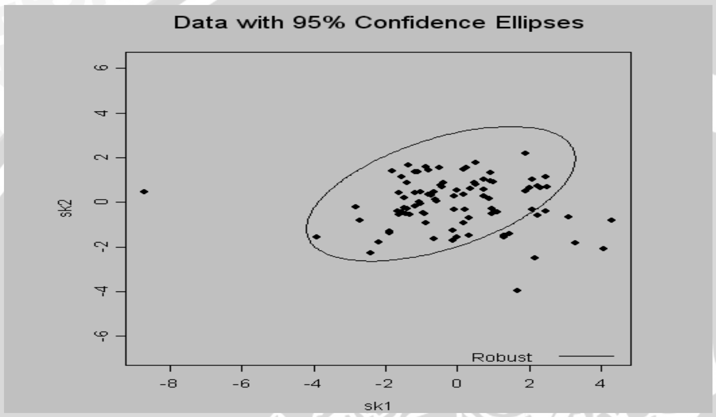


Plot nilai jarak mahalanobis (*Scor outlier map*) untuk CPCA

Sumber: output MATLAB 7.0.1

Lampiran 17 ***Tolerance Ellipse*** **Data Penelitian I**





Sumber: output dari S-PLUS 6

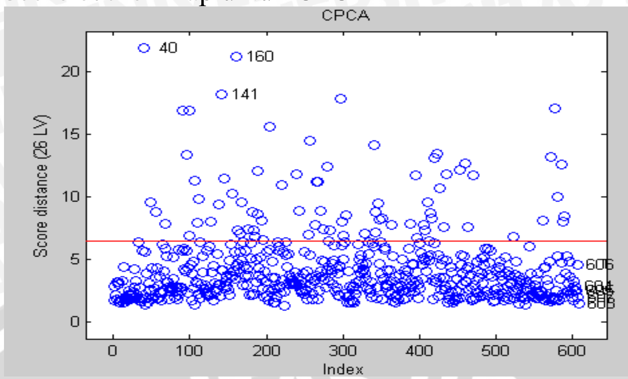
Keterangan

V_1 : Peubah pertama untuk data masukan (data yang sudah distandardisasi)

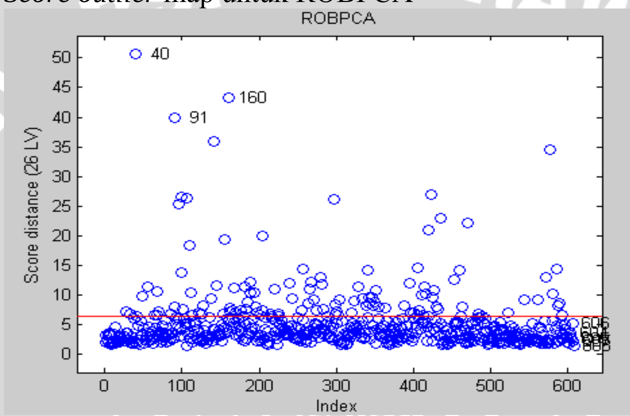
V_2 : Peubah kedua untuk data masukan (data yang sudah distandardisasi)

Lampiran 18 *Score Outlier Map*/Jarak Mahalanobis Hasil Analisis Menggunakan CPCA dan ROBPCA untuk data Penelitian II

Score outlier map untuk CPCA

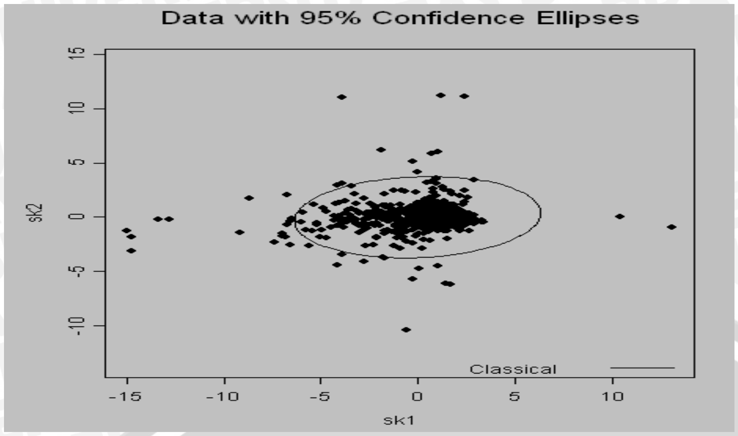


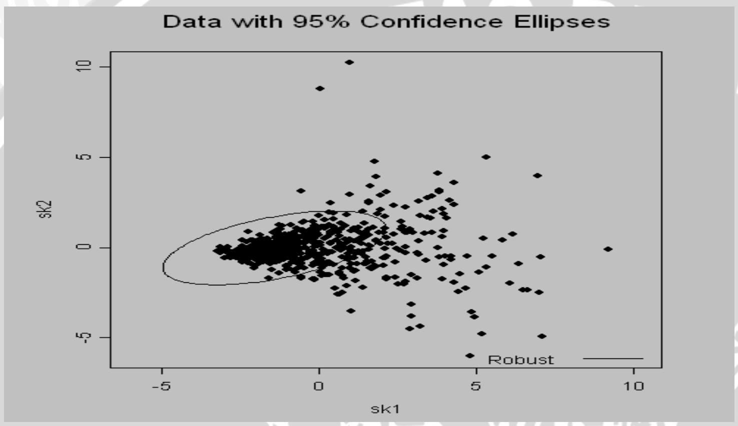
Score outlier map untuk ROBPCA



Sumber: Output MATLAB 7.0.1

Lampiran 19 *Tolerance Ellipse* untuk Data Penelitian II





Sumber: output dari S-PLUS 6

Keterangan

V_1 : Peubah pertama untuk data masukan (data yang sudah distandardisasi)

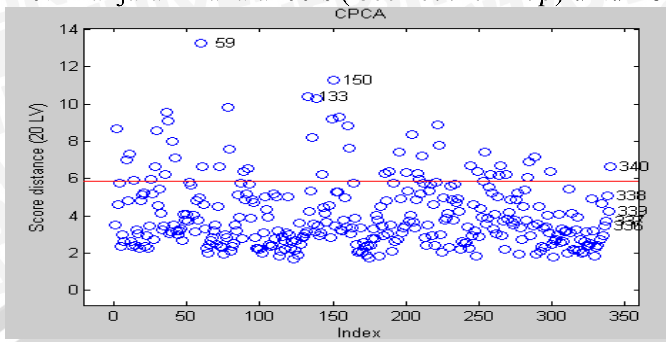
V_2 : Peubah kedua untuk data masukan (data yang sudah distandardisasi)

....

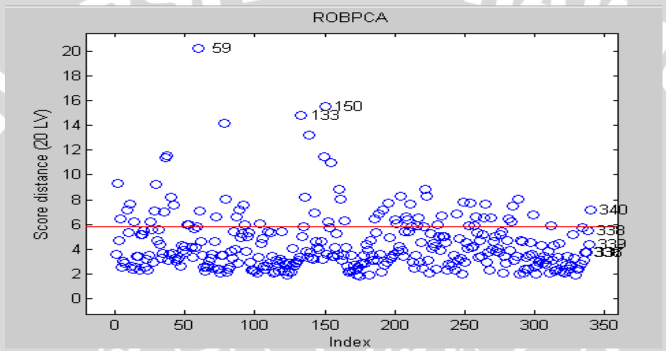
V_{26} : Peubah ke-26 untuk data masukan (data yang sudah distandardisasi)

Lampiran 20 *Score Outlier Map*/Jarak Mahalanobis Hasil Analisis Menggunakan CPCA dan ROBPCA untuk data Penelitian III

Plot nilai jarak mahalanobis (*Scor outlier map*) untuk CPCA

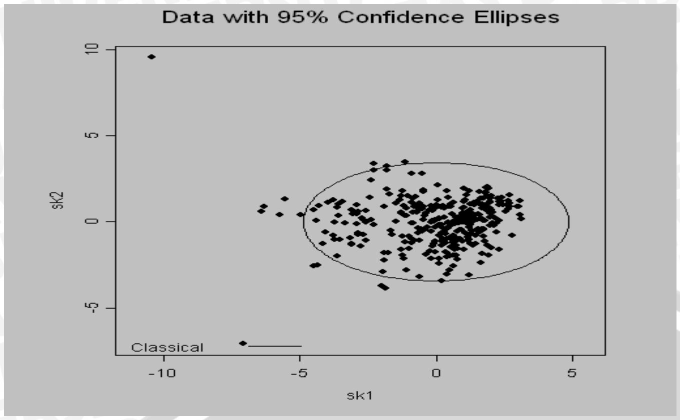


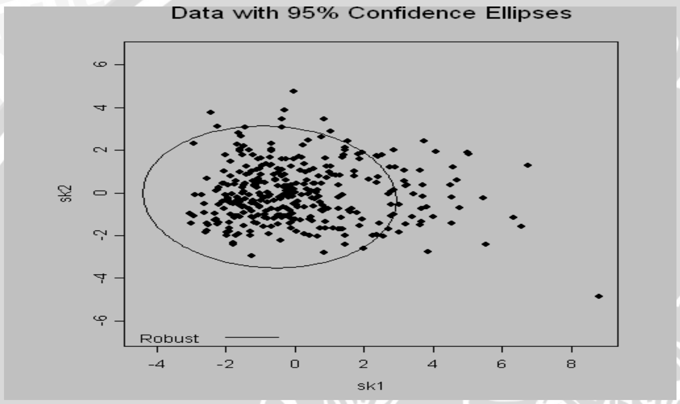
Plot nilai jarak mahalanobis (*Scor outlier map*) untuk ROBPCA



Sumber: output MATLAB 7.0.1

Lampiran 21 *Tolerance Ellipse* untuk Data Penelitian III





Sumber: output dari S-PLUS 6

Keterangan

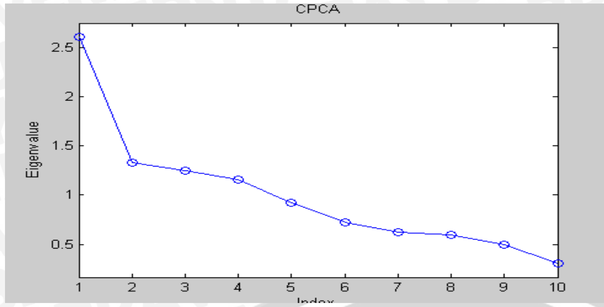
V_1 : Peubah pertama untuk data masukan (data yang sudah distandardisasi)

V_2 : Peubah kedua untuk data masukan (data yang sudah distandardisasi)

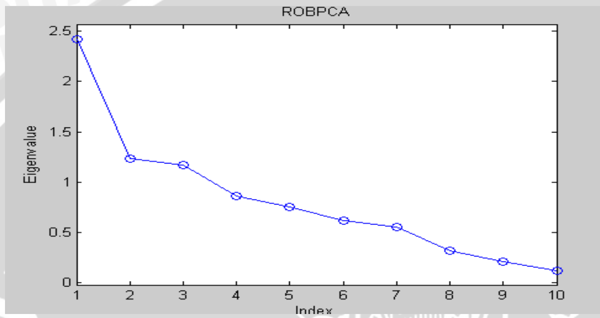
....

V_{20} : Peubah ke-20 untuk data masukan (data yang sudah distandardisasi)

Lampiran 22 *Scree Plot* untuk Data Penelitian I

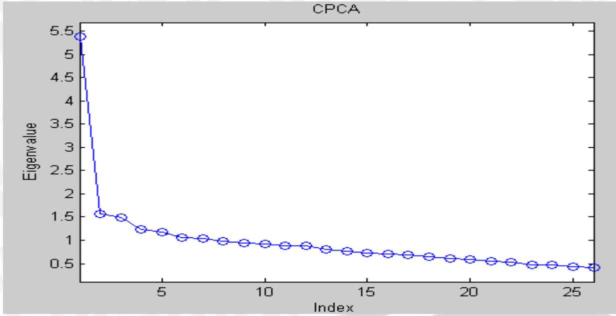


Gambar 4.1. *Scree plot* antara *eigenvalue* dan komponen utama untuk metode CPCA

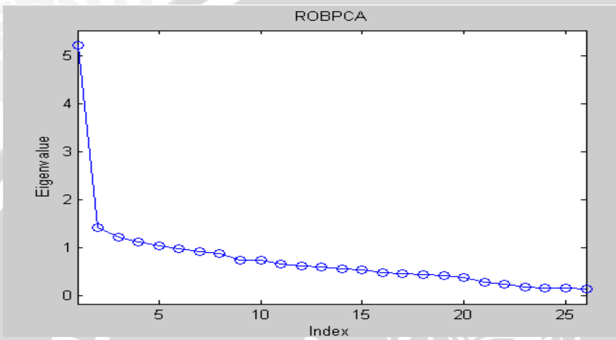


Gambar 4.2. *Scree plot* antara *eigenvalue* dan komponen utama untuk metode ROBPCA

Lampiran 23 *Scree Plo*t untuk Data Penelitian II

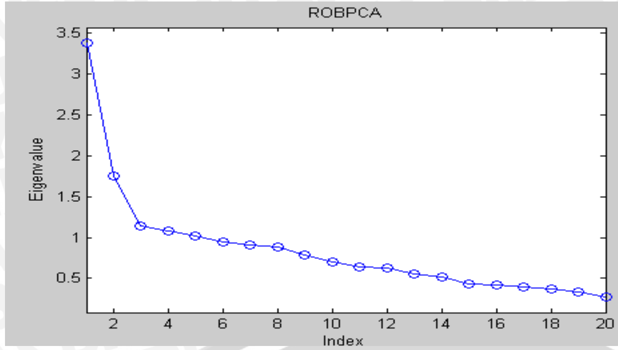


Gambar 4.3. *Scree plot* antara *eigenvalue* dan komponen utama untuk metode CPCA

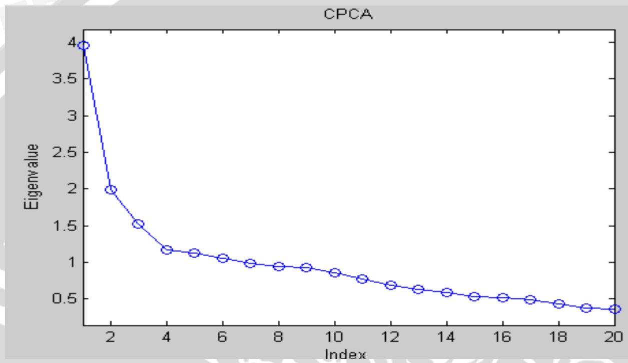


Gambar 4.4. *Scree plot* antara *eigenvalue* dan komponen utama untuk metode ROBPCA

Lampiran 24 *Scree Plo*t untuk Data Penelitian III

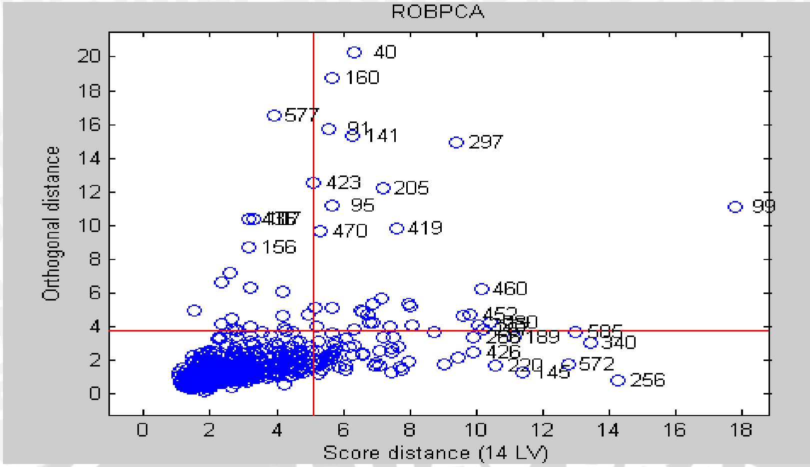


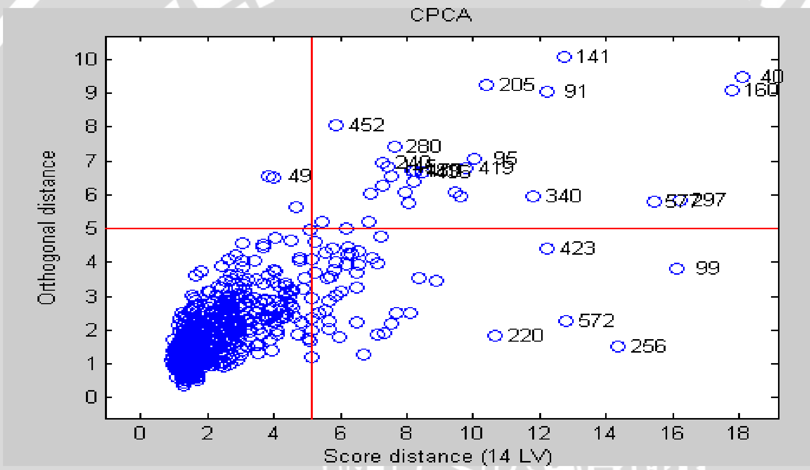
Gambar 4.5. *Scree plot* antara *eigenvalue* dan komponen utama untuk metode CPCA untuk data penelitian III



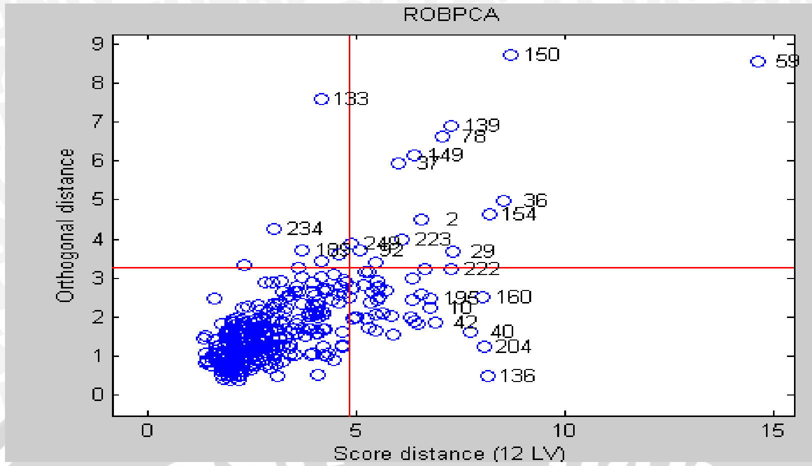
Gambar 4.6. *Scree plot* antara *eigenvalue* dan komponen utama untuk metode ROBPCA untuk data penelitian III

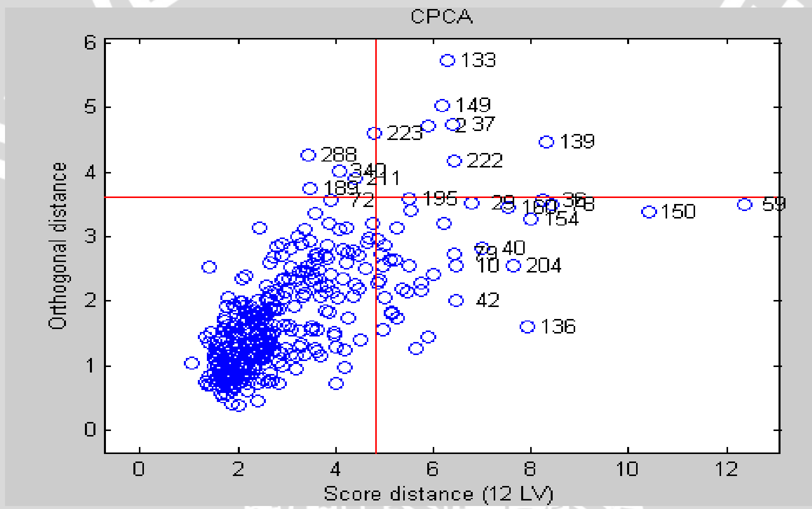
Lampiran 25 *Outlier Map* antara Jarak Mahalanobis dan Jarak Ortogonal untuk Data II





Lampiran 26 *Outlier Map* antara Jarak Mahalanobis dan Jarak Ortogonal untuk Data III





Lampiran 27 Nilai Skor Komponen untuk Data I

No	sk 1	sk2
1	-3,91	-1,57
2	-1,56	-0,42
3	-1,33	-0,56
4	-1,64	0,41
5	-1,61	-0,56
6	-1,54	1,14
7	-1,41	-0,28
8	0,17	-0,91
9	0,46	0,88
10	-2,41	-2,28
11	-0,45	0,78
12	-0,94	-0,48
13	-0,11	-1,72
14	-0,65	-1,65
…	……	……
91	2,25	0,74

Keterangan
sk 1: skor komponen utama 1
sk 2: skor komponen utama 2

Lampiran 28 Nilai Skor Komponen untuk Data II

No	sk1	sk2
1	-2,11	-0,21
2	-2,61	-0,23
3	-1,88	-0,25
4	-2,63	-0,32
5	-1,25	-0,78
6	-1,78	-0,83
7	-1,46	-0,48
8	-2,05	0,08
9	-2,43	-0,3
10	-2,84	-0,34
11	-1,27	-0,43
12	-3,24	-0,23
13	-0,4	-0,53
14	-2,43	-0,17
15	-2,24	-0,16
…	……	……
608	-2,42	0,08

Keterangan
sk 1: skor komponen utama 1
sk 2: skor komponen utama 2

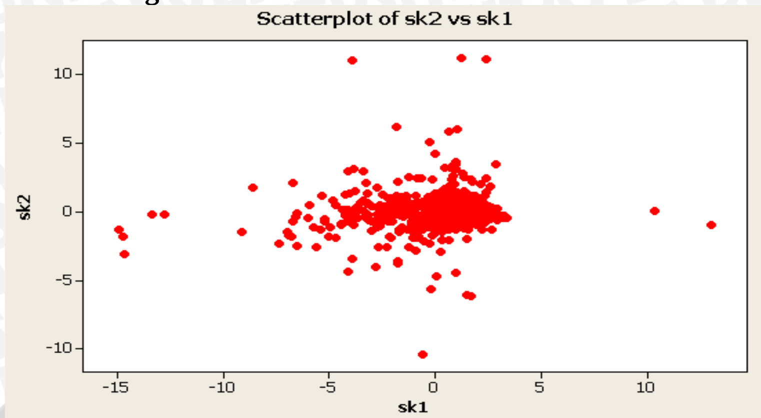
Lampiran 29 Nilai Skor Komponen untuk Data III

No	sk1	sk2
1	0.53	0.45
2	5.02	1.81
3	0.39	2.48
4	3.83	-0.68
5	0.29	0.4
6	-0.01	0.04
7	0.79	-0.84
8	2.35	-1.95
9	0.77	-0.81
10	4.1	1.92
11	-2.29	-1.23
12	-0.36	-0.14
13	2.8	-1.08
14	-3.01	-1.41
15	1,41	1,76
…	…...	……
340	0.88	-0.72

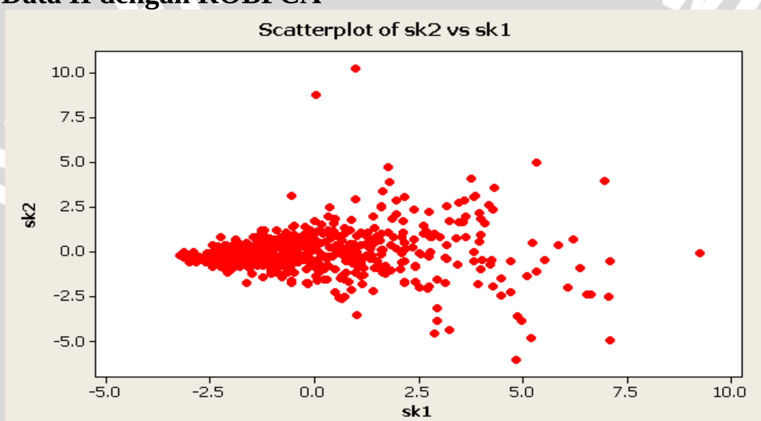
Keterangan
sk 1: skor komponen utama 1
sk 2: skor komponen utama 2

Lampiran 30 Plot antara Skor Komponen Utama 1 dan Skor Komponen Utama 2 untuk Data II dan III

Data II dengan CPCA

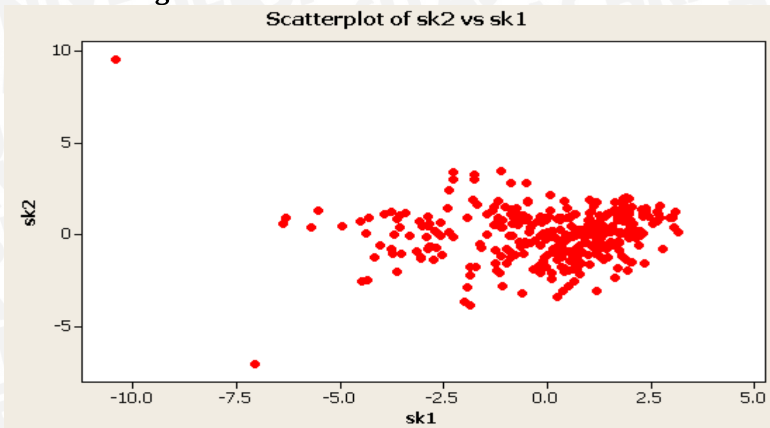


Data II dengan ROBPCA

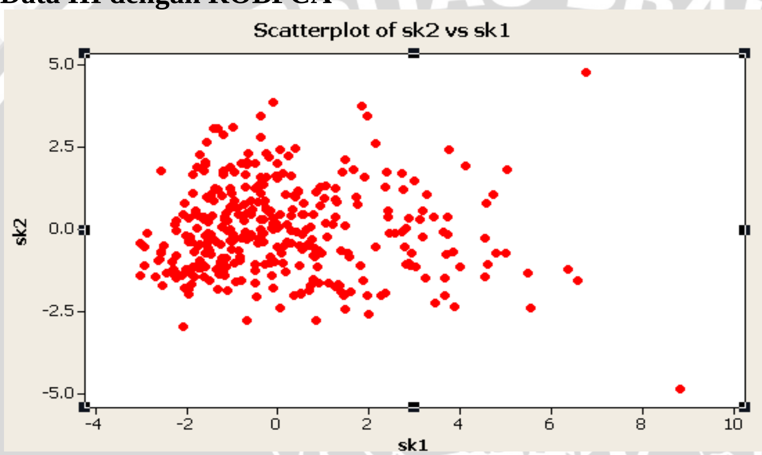


Lampiran 30 (lanjutan)

Data III dengan CPCA



Data III dengan ROBPCA



Lampiran 31 Macro MATLAB 7.0.1

```
function result=robpca(x,varargin);

%ROBPCA is a 'ROBust method for Principal Components Analysis'.
% It is resistant to outliers in the data  The robust loadings
are computed
% using projection-pursuit techniques and the MCD method.
% Therefore ROBPCA can be applied to both low and high-
dimensional data sets.
% In low dimensions, the MCD method is applied (see mcdcov.m).
%
% The ROBPCA method is described in
%  Hubert, M., Rousseeuw, P.J., Vanden Branden, K. (2005),
ROBPCA: a
%  new approach to robust principal components analysis,
Technometrics, 47, 64-79.
%
% To select the number of principal components, a robust PRESS
(predicted
% residual sum of squares) curve is drawn, based on a fast
algorithm for
% cross-validation. This approach is described in:
%
%    Hubert, M., Engelen, S. (2007),
%    "Fast cross-validation of high-breakdown resampling
algorithms for PCA",
%    Computational Statistics and Data Analysis, 51, 5013-5024.
%
% For the up-to-date references, please consult the website:
%    http://wis.kuleuven.be/stat/robust.html
%
% Required input arguments:
%            x : Data matrix (observations in the rows,
variables in the
%                columns)
%
% Optional input arguments:
%            k : Number of principal components to compute. If
k is missing,
%                or k = 0, a scree plot and a press curve are
drawn which allows you to select
%                the number of principal components.
%         kmax : Maximal number of principal components to
compute (default = 10).
%                If k is provided, kmax does not need to be
specified, unless k is larger
%                than 10.
%        alpha : (1-alpha) measures the fraction of outliers
the algorithm should
%                resist. Any value between 0.5 and 1 may be
specified (default = 0.75).
```

Lampiran 31 (lanjutan)

```
%            h : (n-h+1) measures the number of outliers the
algorithm should
%                resist. Any value between n/2 and n may be
specified. (default = 0.75*n)
%                Alpha and h may not both be specified.
%          mcd : If equal to one: when the number of variables
is sufficiently small,
%                the loadings are computed as the eigenvectors
of the MCD covariance matrix,
%                hence the function 'mcdcov.m' is automatically
called. The number of
%                principal components is then taken as k =
rank(x). (default)
%                If equal to zero, the robpca algorithm is
always applied.
%        plots : If equal to one, a scree plot, a press curve
and a robust score outlier map are
%                drawn (default). If the input argument
'classic' is equal to one,
%                the classical plots are drawn as well.
%                If 'plots' is equal to zero, all plots are
suppressed (unless k is missing,
%                then the scree plot and press curve are still
drawn).
%                See also makeplot.m
%        labsd : The 'labsd' observations with largest score
distance are
%                labeled on the outlier map. (default = 3)
%        labod : The 'labod' observations with largest
orthogonal distance are
%                labeled on the outlier map. default = 3)
%      classic : If equal to one, the classical PCA analysis
will be performed
%                (see also cpca.m). (default = 0)
%        scree : If equal to one, a scree plot is drawn. If k
is given as input, the default value is 0, else the default
value is one.
%        press : If equal to one, a plot of robust press-values
is drawn.
%                If k is given as input, the default value is
0, else the default value is one.
%    robpcamcd : If equal to one (default), the whole robpca
procedure is run (computation of outlyingness and
%                MCD).
%                If equal to zero, the program stops after the
computation of the outlyingness. The
%                robust eigenvectors then correspond with the
eigenvectors of the covariance matrix
%                of the h observations with smallest
outlyingness. This yields the same
```

Lampiran 31 (lanjutan)

```
%                PCA subspace as the full robpca, but not the
same eigenvectors and eigenvalues.
%
% I/O:
result=robpca(x,'k',k,'kmax',10,'alpha',0.75,'h',h,'mcd',1,'plo
ts',1,'labsd',3,'labod',3,'classic',0);
%  The user should only give the input arguments that have to
change their default value.
%  The name of the input arguments needs to be followed by
their value.
%  The order of the input arguments is of no importance.
%
% Examples:
%    result=robpca(x,'k',3,'alpha',0.65,'plots',0)
%    result=robpca(x,'alpha',0.80,'kmax',15,'labsd',5)
%
% The output of ROBPCA is a structure containing
%
%    result.P        : Robust loadings (eigenvectors)
%    result.L        : Robust eigenvalues
%    result.M        : Robust center of the data
%    result.T        : Robust scores
%    result.k        : Number of (chosen) principal components
%    result.h        : The quantile h used throughout the
algorithm
%    result.Hsubsets : A structure that contains H0, H1 and
Hfreq:
%                      H0 : The h-subset that contains the h
points with the smallest outlyingness.
%                      H1 : The optimal h-subset of mcdcov.
%                      Hfreq : The subset of h points which are
the most frequently selected during the mcdcov
%                              algorithm.
%    result.sd       : Robust score distances within the robust
PCA subspace
%    result.od       : Orthogonal distances to the robust PCA
subspace
%    result.cutoff   : Cutoff values for the robust score and
orthogonal distances
%    result.flag     : The observations whose score distance is
larger than result.cutoff.sd
%                      or whose orthogonal distance is larger
than result.cutoff.od
%                      can be considered as outliers and
receive a flag equal to zero.
%                      The regular observations receive a flag
1.
%    result.class    : 'ROBPCA'
%    result.classic  : If the input argument 'classic' is equal
to one, this structure
```

Lampiran 31 (lanjutan)

```
%                    contains results of the classical PCA
analysis (see also cpca.m).
%
% Short description of the method:
%
% Let n denote the number of observations, and p the number of
original variables,
% then ROBPCA finds a robust center (p x 1) of the data M and a
loading matrix P which
% is (p x k) dimensional. Its columns are orthogonal and define
a new coordinate
% system. The scores (n x k) are the coordinates of the
centered observations with
% respect to the loadings: T=(X-M)*P.
% Note that ROBPCA also yields a robust covariance matrix
(often singular) which
% can be computed as
%                          cov=out.P*out.L*out.P'
%
% To select the number of principal components, it is useful to
look at the scree plot which
% shows the eigenvalues, and the press curve which displays a
weighted sum of the squared
% cross-validated orthogonal distances.
% The outlier map visualizes the observations by plotting their
orthogonal
% distance to the robust PCA subspace versus their robust
distances
% within the PCA subspace. This allows to classify the data
points into 4 types:
% regular observations, good leverage points, bad leverage
points and
% orthogonal outliers.
%
% This function is part of LIBRA: the Matlab Library for Robust
Analysis,
% available at:
%              http://wis.kuleuven.be/stat/robust.html
%
% Written by Mia Hubert, Sabine Verboven, Karlien Vanden
Branden, Sanne Engelen
% Last Update: 17/06/2003

%
% initialization with defaults
%
data=x;
[n,p]=size(data);

% First Step: classical PCA on data
if n < p
```

Lampiran 31 (lanjutan)

```
    [P1,T1,L1,r,Xc,clm]=kernelEVD(data);
else
    [P1,T1,L1,r,Xc,clm]=classSVD(data);
end

if r==0
    error('All data points collapse!')
end

niter=100;
counter=1;
kmax=min([10,floor(n/2),r]);
k=0;
alfa=0.75;
h=min(floor(2*floor((n+kmax+1)/2)-n+2*(n-
floor((n+kmax+1)/2))*alfa),n);
labsd=3;
labod=3;
plots=1;
scree = 1;
press = 1;
mcd=1; % user wants the mcd approach (in case n>>p)
robpcamcd = 1;
cutoff  = 0.975;
% default is a structure needed for input checking
default=struct('alpha',alfa,'h',h,'labsd',labsd,'labod',labod,.
..

'k',k,'plots',plots,'kmax',kmax,'mcd',mcd,'classic',0,'scree',s
cree,'press',press,'robpcamcd',robpcamcd,'cutoff',cutoff);
list=fieldnames(default);
options=default;
IN=length(list);
i=1;
%
if nargin==2
    error('Incorrect number of input arguments!')
end
if nargin>2
    %
    %placing inputfields in array of strings
    %
    for j=1:nargin-2
        if rem(j,2)~=0
            chklist{i}=varargin{j};
            i=i+1;
        end
    end
    dummy=sum(strcmp(chklist,'h')+2*strcmp(chklist,'alpha'));
    switch dummy
    case 0 % Take on default values
```

Lampiran 31 (lanjutan)

```
        options.alpha=alfa; % 0.75
        if any(strcmp(chklist,'kmax'))
            for j=1:nargin-2 % searching the index of the
accompanying field
                if rem(j,2)~=0 % fieldnames are placed on odd
index
                    if strcmp('kmax',varargin{j})
                        I=j;
                    end
                end
            end
            options=setfield(options,'kmax',varargin{I+1});
            kmax = options.kmax;
            options.h=min(floor(2*floor((n+kmax+1)/2)-n+2*(n-
floor((n+kmax+1)/2))*alfa),n);
        else
           options.h=h;
        end
    case 3
        error('Both inputarguments alpha and h are provided.
Only one is required.')
    end

    %
    % Checking which default parameters have to be changed
    % and keep them in the structure 'options'.
    %
    while counter<=IN
        index=strmatch(list(counter,:),chklist,'exact');
        if ~isempty(index) % in case of similarity
            for j=1:nargin-2 % searching the index of the
accompanying field
                if rem(j,2)~=0 % fieldnames are placed on odd
index
                    if strcmp(chklist{index},varargin{j})
                        I=j;
                    end
                end
            end

options=setfield(options,chklist{index},varargin{I+1});
            index=[];
        end
        counter=counter+1;
    end
    options.h=floor(options.h);
    options.kmax=floor(options.kmax);
    options.k=floor(options.k);
    kmax=max(min([options.kmax,floor(n/2),r]),1);
    labod=max(0,min(floor(options.labod),n));
    labsd=max(0,min(floor(options.labsd),n));
```

Lampiran 31 (lanjutan)

```
    k=options.k;

    if k<0
        k=0;
    elseif k > kmax
        k=kmax;
        mess=sprintf(['Attention (robpca.m): The number of
principal components, k = ',num2str(options.k)...
            ,'\n is larger than kmax= ',num2str(kmax),'; k is
set to ',num2str(kmax)]);
        disp(mess)
    end
    if dummy==1 % checking input variable h
        options.alpha=options.h/n;
        if k==0
            if options.h < floor((n+kmax+1)/2 )
                options.h=floor((n+kmax+1)/2);
                options.alpha=options.h/n;
                mess=sprintf(['Attention (robpca.m): h should
be larger than (n+kmax+1)/2.\n',...
                        'It is set to its minimum value
',num2str(options.h)]);
                disp(mess)
            end
        else
            if options.h < floor((n+k+1)/2)
                options.h=floor((n+k+1)/2);
                options.alpha=options.h/n;
                mess=sprintf(['Attention (robpca.m): h should
be larger than (n+k+1)/2.\n',...
                        'It is set to its minimum value
',num2str(options.h)]);
                disp(mess)
            end
        end
        if options.h > n
            options.alpha=0.75;
            if k==0
                options.h=floor(2*floor((n+kmax+1)/2)-n+2*(n-
floor((n+kmax+1)/2))*options.alpha);
            else
                options.h=floor(2*floor((n+k+1)/2)-n+2*(n-
floor((n+k+1)/2))*options.alpha);
            end
            mess=sprintf(['Attention (robpca.m): h should be
smaller than n. \n',...
                    'It is set to its default value
',num2str(options.h)]);
            disp(mess)
        end
    elseif dummy==2 %checking input variable alpha
```

Lampiran 31 (lanjutan)

```
        if options.alpha < 0.5
            options.alpha=0.5;
            mess=sprintf(['Attention (robpca.m): Alpha should
be larger than 0.5.\n',...
                    'It is set to 0.5.']);
            disp(mess)
        end
        if options.alpha > 1
            options.alpha=0.75;
            mess=sprintf(['Attention (robpca.m): Alpha should
be smaller than 1. \n',...
                    'It is set to 0.75.']);
            disp(mess)
        end
        if k==0
            options.h=floor(2*floor((n+kmax+1)/2)-n+2*(n-
floor((n+kmax+1)/2))*options.alpha);
        else
            options.h=floor(2*floor((n+k+1)/2)-n+2*(n-
floor((n+k+1)/2))*options.alpha);
        end
    end
    h=options.h;
    alfa=options.alpha;
    dummyh = strcmp(chklist,'h');
    dummykmax = strcmp(chklist,'kmax');
    if all(dummyh == 0) & any(dummykmax) & k==0
        h = min(floor(2*floor((n+kmax+1)/2)-n+2*(n-
floor((n+kmax+1)/2))*alfa),n);
    end
    dummyscree = strcmp(chklist,'scree');
    dummypress = strcmp(chklist,'press');
    if all(dummyscree == 0)
        if k~=0
            options.scree = 0;
        end
    end
    if all(dummypress == 0)
        if k~=0
            options.press = 0;
        end
    end
    scree = options.scree;
    press = options.press;
    labsd=floor(max(0,min(options.labsd,n)));
    labod=floor(max(0,min(options.labod,n)));
    plots=options.plots;
    mcd=options.mcd;
    robpcamcd = options.robpcamcd;
    cutoff = options.cutoff;
end
```

Lampiran 31 (lanjutan)

```
%
% MAIN PART
%
X=T1;
center=clm;
rot=P1;
% Depending on n and p, perfoRD MCD or ROBPCA:
% p << n => MCD
p1=size(X,2);
if p1<=min(floor(n/5),kmax) & mcd
    options.h=h;
    [res,raw]=mcdcov(X,'h',h,'plots',0);
    [U,S,P]=svd(res.cov,0);
    L=diag(S);
    if k~=0
        options.k=min(k,p1);
    else
        bdwidth=5;
        topbdwidth=30;
        set(0,'Units','pixels');
        scnsize=get(0,'ScreenSize');
        pos1=[bdwidth, 1/3*scnsize(4)+bdwidth, scnsize(3)/2-
2*bdwidth, scnsize(4)/2-(topbdwidth+bdwidth)];
        pos2=[pos1(1)+scnsize(3)/2, pos1(2), pos1(3), pos1(4)];
        if press == 1
            outcvMcd = cvMcd(X,p1,res,h);
            figure('Position',pos1)
            set(gcf,'Name', 'PRESS curve','NumberTitle',
'off');
            plot(1:p1,outcvMcd.press,'o-')
            title('MCD')
            xlabel('number of LV')
            ylabel('R-PRESS')
        end
        if scree == 1
            figure('Position',pos2)
            screeplot(L,'MCD');
        end
        if (scree == 1)|(press == 1)
            cumperc = cumsum(L)./sum(L);
            disp(['The cumulative percentage of variance
explained by the first ',num2str(kmax),' components is:']);
            disp([num2str(cumperc')]);
            disp(['How many principal components would you like
to retain? Max = ',num2str(kmax),'. ']);
            k=input('');
        end
        % to close the figures.
        if scree == 1
            close
        end
```

Lampiran 31 (lanjutan)

```
        if press == 1
            close
        end
    end
    options.k = k;
    T=(X-repmat(res.center,size(X,1),1))*U;
    out.M=center+res.center*rot';
    out.L=L(1:options.k)';
    out.P=rot*U(:,1:options.k);
    out.T=T(:,1:options.k);
    out.h=h;
    out.k=options.k;
    out.alfa=alfa;
    out.Hsubsets.H0 = res.Hsubsets.Hopt;
    out.Hsubsets.H1 = [];
    out.Hsubsets.Hfreq = res.Hsubsets.Hfreq;
else
    % p > n => ROBPCA
    niter=100;
    seed=0;
    if h~=n

        % dim(P1): p x r
        nrich1=n*(n-1)/2;
        ndirect=min(250,nrich1);
        true = (ndirect == nrich1);
        B=extradir(T1,ndirect,seed,true); %n*ri
        for i=1:size(B,1)
            Bnorm(i)=norm(B(i,:),2);
        end
        Bnormr=Bnorm(Bnorm > 1.e-12); %ndirect*1
        B=B(Bnorm > 1.e-12,:);       %ndirect*n
        A=diag(1./Bnormr)*B;         %ndirect*n
        %projected points in columns
        Y=T1*A';%n*ndirect
        m=length(Bnormr);
        Z=zeros(n,m);
        for i=1:m
            [tmcdi,smcdi,weights]=unimcd(Y(:,i),h);
            if smcdi<1.e-12
                r2=rank(data(weights,:));
                if r2==1
                    error(['At least ',num2str(sum(weights)),'
obervations are identical.']);
                end
            else
                Z(:,i)=abs(Y(:,i)-tmcdi)/smcdi;
            end
        end
        d=max(Z');
        [ds,is]=sort(d);
```

Lampiran 31 (lanjutan)

```
        Xh=T1(is(1:h),:); % Xh contains h (good) points out of
Xcentr
        [P2,T2,L2,r2,Xm,clmX]=classSVD(Xh);
         out.Hsubsets.H0 = is(1:h);
        Tn=(T1-repmat(clmX,n,1))*P2;

    else
        P2=eye(r);
        Tn=T1;
        L2=L1;
        r2=r;
        out.Hsubsets.H0=1:n;
        Xm=T1;
        clmX=zeros(1,size(T1,2));
    end

    %dim(P2) = r x r2
    L=L2;
    kmax=min(r2,kmax);

    % choice of k:
    %-------------
    bdwidth=5;
        topbdwidth=30;
        set(0,'Units','pixels');
        scnsize=get(0,'ScreenSize');
        pos1=[bdwidth, 1/3*scnsize(4)+bdwidth, scnsize(3)/2-
2*bdwidth, scnsize(4)/2-(topbdwidth+bdwidth)];
        pos2=[pos1(1)+scnsize(3)/2, pos1(2), pos1(3), pos1(4)];

    if press == 1
        disp('The robust press curve based on cross-validation
is now computed.')
        outprMCDkmax =
projectMCD(Tn,L,kmax,h,niter,rot,P1,P2,center,cutoff);
        outprMCDkmax.Hsubsets.H0 = out.Hsubsets.H0;
        outpress = cvRobpca(data,kmax,outprMCDkmax,0,h);
        figure('Position',pos1)
        set(gcf,'Name', 'PRESS curve','NumberTitle', 'off');
        plot(1:kmax,outpress.press,'o-')
        title('ROBPCA')
        xlabel('Number of LV')
        ylabel('R-PRESS')
    end

    if scree == 1
        figure('Position',pos2)
        screeplot(L(1:kmax),'ROBPCA')
    end

    if (scree == 1)|(press == 1)
```

Lampiran 31 (lanjutan)

```
        cumperc = (cumsum(L(1:kmax))./sum(L))';
        disp(['The cumulative percentage of variance explained 
by the first ',num2str(kmax),' components is:']);
        disp([num2str(cumperc)]);
        disp(['How many principal components would you like to 
retain? Max = ',num2str(kmax),'.']);
        k=input('');
        k=max(min(min(r2,k),kmax),1);
    else
        k=min(min(r2,k),kmax);
    end
    % to close the figures.
    if scree == 1
        close
    end
    if press == 1
        close
    end
    if ~robpcamcd
        center=clm;
        out.P = P1*P2(:,1:k);
        out.M = center + clmX*P1';
        
result=struct('P',{out.P},'M',{out.M},'k',{k},'h',{h},'Hsubsets
',{out.Hsubsets});
        return
    end

    if k~=r
        XRc=T1-repmat(clmX,n,1);
        Xtilde=XRc*P2(:,1:k)*P2(:,1:k)';
        Rdiff = XRc-Xtilde;
        for i=1:n
            odh(i,1)=norm(Rdiff(i,:));
        end
        [m,s]=unimcd(odh.^(2/3),h);
        cutoffodh = sqrt(norminv(cutoff,m,s).^3);
        indexset = find(odh<=cutoffodh)';
        [P2,Th,Lh,rh,Xm,clmX]=classSVD(T1(indexset,:));
    end
    center=center+clmX*rot';
    rot=rot*P2(:,1:k);
    Tn=(T1-repmat(clmX,n,1))*P2;

    % projection, mcd
    %-----------------

    outprMCD = 
projectMCD(Tn,L,k,h,niter,rot,P1,P2,center,cutoff);

    out.T = outprMCD.T;
```

Lampiran 31 (lanjutan)

```
    out.P = outprMCD.P;
    out.M = outprMCD.M;
    out.L = outprMCD.L;
    out.k = k;
    out.h = h;
    out.alfa = alfa;
    out.Hsubsets.H1 = outprMCD.Hsubsets.H1;
    out.Hsubsets.Hfreq = outprMCD.Hsubsets.Hfreq;
end

% Classical analysis
if options.classic==1
    out.classic.P=P1(:,1:out.k);
    out.classic.L=L1(1:out.k)';
    out.classic.M=clm;
    out.classic.T=T1(:,1:out.k);
    out.classic.k=out.k;
    out.classic.Xc=Xc;
end

outprMCD = out;

% Calculation of the distances, flags
%-------------------------------------

if options.classic == 1
    outDist = CompDist(data,r,outprMCD,cutoff,out.classic);
else
    outDist = CompDist(data,r,outprMCD,cutoff);
end

out.sd = outDist.sd;
out.cutoff.sd = outDist.cutoff.sd;
out.od = outDist.od;
out.cutoff.od = outDist.cutoff.od;
out.flag = outDist.flag;
out.class = outDist.class;
out.classic = outDist.classic;
if options.classic == 1
    out.classic.sd = outDist.classic.sd;
    out.classic.od = outDist.classic.od;
    out.classic.cutoff.sd = outDist.classic.cutoff.sd;
    out.classic.cutoff.od = outDist.classic.cutoff.od;
    out.classic.class = outDist.classic.class;
    out.classic.flag = outDist.classic.flag;
end


result=struct('P',{out.P},'L',{out.L},'M',{out.M},'T',{out.T},'
k',{out.k},'h',{out.h},'Hsubsets',{out.Hsubsets},...
```

Lampiran 31 (lanjutan)

```
    'sd',
{out.sd},'od',{out.od},'cutoff',{out.cutoff},'flag',out.flag',.
..
    'class',{out.class},'classic',{out.classic});

% Plots
try
    if plots & options.classic

makeplot(result,'classic',1,'labsd',labsd,'labod',labod)
    elseif plots
        makeplot(result,'labsd',labsd,'labod',labod)
    end
catch %output must be given even if plots are interrupted
    %> delete(gcf) to get rid of the menu
end
%---------------------------------------------------------------
--------------
function outprMCD =
projectMCD(Tn,L,k,h,niter,rot,P1,P2,center,cutoff)

% these function performs the last part of ROBPCA when k is
determined.
% input :
%   Tn : the projected data
%   L  : the matrix of the eigenvalues
%   k  : the number of components
%   h  :
%   niter : the number of iterations
%   rot : the rotation matrix
%   P1, P2: the different eigenvector matrices after each
transformation
%   center : the classical center of the data

X2=Tn(:,1:k);
n = size(X2,1);
rot=rot(:,1:k);
% first apply c-step with h points from first step,i.e. those
that
% determine the covariance matrix after the c-steps have
converged.
mah=mahalanobis(X2,zeros(size(X2,2),1),'cov',L(1:k));
oldobj=prod(L(1:k));
P4=eye(k);
for j=1:niter
    [mahs,is]=sort(mah);
    Xh=X2(is(1:h),:);
    [P,T,L,r3,Xm,clmX]=classSVD(Xh);
    obj=prod(L);
    X2=(X2-repmat(clmX,n,1))*P;
    center=center+clmX*rot';
```

Lampiran 31 (lanjutan)

```
    rot=rot*P;
    mah=mahalanobis(X2,zeros(size(X2,2),1),'cov',diag(L));
    P4=P4*P;
    if ((r3==k) & (abs(oldobj-obj) < 1.e-12))
        break;
    else
        oldobj=obj;
        j=j+1;
        if r3 < k
            j=1;
            k=r3;
        end
    end
end
% dim(P4): k x k0 with k0 <= k but denoted as k
% dim X2: n x k0
% perform mcdcov on X2
[zres,zraw]= mcdcov(X2,'plots',0,'ntrial',250,'h',h,'file',0);
out.resMCD = zres;
if zraw.objective < obj
    z = zres;
    out.Hsubsets.H1 = zres.Hsubsets.Hopt;
else
    sortmah = sort(mah);
    if h==n
        factor=1;
    else
        factor = sortmah(h)/chi2inv(h/n,k);
    end
    mah = mah/factor;
    weights = mah <= chi2inv(cutoff,k);
    [center_noMCD,cov_noMCD] = weightmecov(X2,weights);
    mah = mahalanobis(X2,center_noMCD,'cov',cov_noMCD);
    z.flag = (mah <= chi2inv(cutoff,k));
    z.center = center_noMCD;
    z.cov = cov_noMCD;
    out.Hsubsets.H1 = is(1:h);
end
covf=z.cov;
centerf=z.center;
[P6,L]=eig(covf);
[L,I]=greatsort(diag(real(L)));
P6=P6(:,I);
out.T=(X2-repmat(centerf,n,1))*P6;
P=P1*P2;
out.P=P(:,1:k)*P4*P6;
centerfp=center+centerf*rot';
out.M=centerfp;
out.L=L';
out.k=k;
out.h=h;
```

Lampiran 31 (lanjutan)

```
% creation of Hfreq
out.Hsubsets.Hfreq = zres.Hsubsets.Hfreq(1:h);

outprMCD = out;
%---------------------------------------------------------------
------------------
function outDist = CompDist(data,r,out,cutoff,classic)

% Calculates the distances.
% input: data : the original data
%           r : the rank of the data
%         out is a structure that contains the results of the
PCA.
%         classic: an optional structure:
%                classic.P1
%                classic.T1
%                classic.L1
%                classic.clm
%                classic.Xc

if nargin < 5
    options.classic = 0;
else
    options.classic = 1;
end

n = size(data,1);
p = size(data,2);
k = out.k;

% Computing distances
% Robust score distances in robust PCA subspace
out.sd=sqrt(mahalanobis(out.T,zeros(size(out.T,2),1),'cov',out.
L))';
out.cutoff.sd=sqrt(chi2inv(cutoff,out.k));
% Orthogonal distances to robust PCA subspace
XRc=data-repmat(out.M,n,1);
Xtilde=out.T*out.P';
Rdiff=XRc-Xtilde;
for i=1:n
    out.od(i,1)=norm(Rdiff(i,:));
end
% Robust cutoff-value for the orthogonal distance
if k~=r
    [m,s]=unimcd(out.od.^(2/3),out.h);
    out.cutoff.od = sqrt(norminv(cutoff,m,s).^3);
else
    out.cutoff.od=0;
end
if options.classic==1
    % Mahalanobis distance in classical PCA subspace
```

Lampiran 31 (lanjutan)

```
    Tclas=classic.Xc*classic.P(:,1:out.k);

out.classic.sd=sqrt(mahalanobis(Tclas,zeros(size(Tclas,2),1),'i
nvcov',1./classic.L(1:out.k)))';
    % Orthogonal distances to classical PCA subspace
    Xtilde=Tclas*classic.P(:,1:out.k)';
    Cdiff=classic.Xc-Xtilde;
    for i=1:n
        out.classic.od(i,1)=norm(Cdiff(i,:));
    end
    out.classic.cutoff.sd=sqrt(chi2inv(cutoff,out.k)); % should
be defined after od to have the output in the correct order
    % Classical cutoff-values
    if k~=r
        m=mean(out.classic.od.^(2/3));
        s=sqrt(var(out.classic.od.^(2/3)));
        out.classic.cutoff.od = sqrt(norminv(cutoff,m,s)^3);
    else
        out.classic.cutoff.od=0;
    end
    out.classic.cutoff.sd=sqrt(chi2inv(cutoff,out.k));

out.classic.flag=((out.classic.od<=out.classic.cutoff.od)&(out.
classic.sd<=out.classic.cutoff.sd));
    out.classic.class='CPCA';
else
    out.classic=0;
end

out.class='ROBPCA';

if k~=r
    out.flag=(out.od<=out.cutoff.od)&(out.sd<=out.cutoff.sd);
else
    out.flag=(out.sd<=out.cutoff.sd);
end

outDist = out;

%---------------------------------------------------------------
--------------
function B2=extradir(data,ndirect,seed,true);

% Calculates ndirect directions through
% two random choosen data points from data

[n,p]=size(data);
B2=zeros(ndirect,p);

if true
    perm=[1 1];
```

Lampiran 31 (lanjutan)

```
end
k=1;
for ndir=1:ndirect
    if true
        k1=2;
        perm(k1)=perm(k1)+1;
        while ~(k1==1 |perm(k1) <=(n-(k+1-k1)))
            k1=k1-1;
            perm(k1)=perm(k1)+1;
            for j=(k1+1):k+1
                perm(j)=perm(j-1)+1;
            end
        end
        index=perm; % index : contains trial subsample.
    else
        [index,seed]=randomset(n,2,seed);
    end
    B2(ndir,:)=data(index(1),:)-data(index(2),:);
end

%---------------------------------------------------------------
---------------------------
function [ranset,seed]=randomset(tot,nel,seed)

% This function is called if not all (p+1)-subsets out of n
will be considered.
% It randomly draws a subsample of nel cases out of tot.

for j=1:nel
   [random,seed]=uniran(seed);
   num=floor(random*tot)+1;
   if j > 1
      while any(ranset==num)
         [random,seed]=uniran(seed);
         num=floor(random*tot)+1;
      end
   end
   ranset(j)=num;
end
```

Sumber: LIBRA (2004)

ROBUST PRINCIPAL COMPONENT ANALYSIS (ROBPCA) PADA DATA PENCILAN DENGAN PENDUGA *MINIMUM COVARIANT DETERMINANT* (MCD)

SKRIPSI

Sebagai salah satu syarat untuk memperoleh gelar Sarjana Sains
dalam bidang Statiistika

oleh:
RIRIN AIMATUS SOLIKAH
0510950048-95

PROGRAM STUDI STATISTIKA
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2009

LEMBAR PENGESAHAN SKRIPSI

***ROBUST PRINCIPAL COMPONENT ANALYSIS* (ROBPCA)**
PADA DATA PENCILAN DENGAN PENDUGA
***MINIMUM COVARIANT DETERMINANT (*MCD)**

Oleh:
RIRIN AIMATUS SOLIKAH
0510950048

Setelah dipertahankan di depan Majelis Penguji
Pada tanggal 27 April 2009
dan dinyatakan memenuhi syarat untuk memperoleh gelar
Sarjana Sains dalam bidang Statistika

Pembimbing I	Pembimbing II
Suci Astutik, S.Si, M.Si	Eni Sumarminingsih, S.Si, MM
NIP. 132 233 148	NIP. 131 300 241

Mengesahkan
Ketua Jurusan Matematika
Fakultas MIPA Universitas Brawijaya

Dr. Agus Suryanto, MSc.
NIP. 132 126 049

LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama	:	Ririn Aimatus Solikah
NIM	:	0510950048
Penulis Tugas Akhir berjudul	:	*Robust Principal Component Analysis* (ROBPCA) pada Data Pencilan dengan Penduga *Minimum Covariant Determinant* (MCD)

Dengan ini menyatakan bahwa :

1. Isi dari Tugas Akhir yang saya buat adalah benar-benar karya sendiri dan tidak menjiplak karya orang lain, selain nama-nama yang termaktub di isi dan tertulis di daftar pustaka dalam Tugas Akhir ini.
2. Apabila dikemudian hari ternyata Tugas Akhir yang saya tulis terbukti hasil jiplakan, maka saya akan bersedia menanggung segala resiko yang akan saya terima.

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 27 April 2009
Yang menyatakan,

(Ririn Aimatus Solikah)
NIM. 0510950048

***ROBUST PRINCIPAL COMPONENT ANALYSIS* (ROBPCA) PADA DATA PENCILAN DENGAN PENDUGA *MINIMUM COVARIANT DETERMINANT* (MCD)**

ABSTRAK

Classic Principal Component Analysis (CPCA) adalah suatu metode dengan pendekatan klasik untuk menjelaskan struktur matriks kovarian melalui sejumlah kecil komponen yang tidak saling berkorelasi, komponen utama merupakan kombinasi linier dari variabel asal sehingga mempunyai ragam maksimum. Pada kenyataannya, data seringkali mengandung beberapa pengamatan pencilan, oleh karena itu reduksi dimensi dengan CPCA menjadi tidak dapat diandalkan bila pencilan muncul dalam data (Sujatmiko, 2005). Pada penelitian ini peneliti mencoba untuk menganalisis data multivatiat yag mengandung pencilan dengan *Robust Principal Component Analysis* (ROBPCA). *Robust Principal Component Analysis* adalah metode analisis untuk mendapatkan komponen utama pada data multivariat yang mengandung pencilan dengan penduga MCD. Penelitian ini bertujuan untuk mendapatkan komponen utama yang *robust* dengan penduga MCD dan menentukan metode yang lebih baik (*Classic* PCA atau ROBPCA) dalam mendapatkan komponen utama pada data pencilan. Penelitian ini menggunakan dua data sekunder yang merupakan hasil survei Sosial Ekonomi Nasional di Kota Sorong dan Kupang dan data hasil penelitian mahasiswa Fakultas Pertanian Universitas Brawijaya. Berdasarkan hasil analisis yang dilakukan dalam penelitian ini, menunjukkan bahwa metode ROBPCA lebih baik dari pada CPCA pada data multivariat mengandung pencilan karena menghasilkan matriks kovarian dan rata-rata yang *robust* terhadap pencilan dengan penduga MCD yang mampu meminimumkan determinan matriks kovarian. Hal ini dapat dilihat dari besarnya keragaman yang dapat dijelaskan oleh komponen utama yang terpilih.

Kata kunci : ROBPCA, CPCA, MCD, *robust statistics*, *multivariate analysis*

ROBUST PRINCIPAL COMPONENT ANALYSIS (ROBPCA) ON OUTLIER DATA USING _MINIMUM COVARIANT DETERMINANT_ (MCD) ESTIMATOR

ABSTRACT

Classic Principal Component Analysis (CPCA) is a method with classic approach clarifying covariant matrix structure by means of few uncorrelated components; principal component is a linear combination of origin variables so that it has maximum variance. In fact, data usually contains of several outlier observations; that is why dimension reduction with CPCA becomes unreliable if the outlier emerges within data (Sujatmiko, 2005). In this research, the researcher was trying to analyze multivariate data containing outlier by *Robust Principal Component Analysis* (ROBPCA). *Robust Principal Component Analysis* is an analysis method used to obtain principal component within multivariate data containing outlier using MCD estimator. The purposes of this research are obtaining *robust* principal component using MCD estimator and determining a better method (PCA *Classic* or ROBPCA) of finding principal component within outlier data. This research used 2 (two) secondary data resulted from National Social Economic survey in Sorong and Kupang, and data resulted from research of a student from Agricultural Faculty of Brawijaya University. The result of analysis done in this research showed that ROBPCA method was better than CPCA in terms of multivariate data containing outlier because it resulted in covariant matrix and means that were *robust* to outlier with MCD estimator that was able to minimize covariant matrix determinant. This could be seen from the number of variance that may be clarified by chosen principal component.

Keywords: ROBPCA, CPCA, MCD, *robust statistics*, *multivariate analysis*

KATA PENGANTAR

Puji syukur atas segala limpahan rahmat dan karunia serta hidayah Allah SWT, hanya degan ijin-Nya lah penyusunan tugas akhir ini dapat selesai dengan baik. Sholawat serta salam semoga tetap tercurah pada junjungan kita, Nabi Muhammad SAW serta pengikutnya.

Skripsi berjudul "*Robust Principal Component Analysis* (ROBPCA) pada Data Pencilan dengan Penduga *Minimum Covariant Determinant* (MCD) " disusun sebagai salah satu syarat mendapatkan gelar Sarjana Sains dalam bidang Statstika di Fakultas MIPA Universitas Brawijaya.

Penyusunan skripsi ini tak lepas dari bantuan yang diberikan oleh beberapa pihak, maka dari itu penulis banyak mengucapkan terima kasih kepada :

1. Ibu Suci Astutik, S.Si, M.Si. selaku dosen pembimbing I dan Ibu Eni Sumarminingsih, S.Si, MM. selaku dosen pembimbing II, atas segala masukan, nasehat dan kesabaran selama proses pembimbingan skripsi.
2. Ibu Ir. Heni Kusdarwati, MS., Ibu Nurjannah, S.Si., M. Phil. dan Bapak Adji Achmad Rinaldo F, S.Si, M.Sc. selaku dosen penguji yang telah memberikan masukan pada skripsi ini.
3. Bapak Dr. Agus Suryanto MSc, selaku ketua jurusan Matematika Fakultas MIPA Universitas Brawijaya atas segala fasilitas yang telah diberikan selama studi.
4. Bapak dan Ibu tercinta, yang telah memberikan kasih sayang, doa, kepercayaan, serta dukungan moril dan materiil. Semoga Allah memberikan kesempatan pada ananda untuk mewujudkan kebahagiaan dan membalas semua kasih yang pernah tercurah.
5. Palik Muslih dan Bulik Aya yang telah banyak membantu saya secara materiil dan spiritual selama masa kuliah.
6. Mbak Suul, Palik Muchsin, Mbak Kom, Palik Komari, Paklik Abas, Mbak Emi dan saudara sepupu di Kediri (Mbak Dha, Faik, Ella, Fika dan Innul) yang selalu mendukung saya.
7. Saudara-saudaraku (Mbak Ulie, Mbak Rahma, Mbak Rida dan Icha) atas segala bantuan dan pengertian dalam membantu penyelesaian skripsi ini.

8. Sahabatku tersayang Mita (teman seperjuanganku), makasih atas *support* dan persahabatan yang mungkin tidak selamanya bisa kita jalani bersama.
9. Medi yang selalu memberikan *support*, semangat, dan sering membangunkanku untuk sholat malam, terima kasih atas semua dukungan yang kamu berikan selama ini.
10. Teman-teman mahasiswa Program Studi Statistika angkatan 2005, 2003 dan 2004 terima kasih atas diskusi dan pertemanan yang baik selama ini.
11. Staf Tata Usaha Jurusan Matematika, atas segala kerjasama dan bantuan yang baik selama ini.
12. Bapak Suryana, atas segala bantuan,materi dan ilmu yang di berikan kepada Ririn.
13. Semua pihak yang secara langsung maupun tak langsung telah membantu tersusunnya skripsi ini.

Akhir kata dengan segala keterbatasan yang dimiliki, penulis mengharap kritik dan saran yang membangun dalam upaya meningkatkan kualitas skripsi ini. Penulis berharap skripsi ini dapat memberikan manfaat kepada para pecinta statistika khususnya, dan kepada pembaca pada umumnya.

Malang, 27 April 2009

Penulis

DAFTAR ISI

DAFTAR GAMBAR

Halaman

DAFTAR TABEL

Halaman

DAFTAR LAMPIRAN

Halaman