



**IMPLEMENTASI ALGORITMA MODEL *ROBUST* FAKTOR UNTUK
TABEL DUA ARAH PADA PROGRAM R**

SKRIPSI

oleh

**Jefri Ardian
NIM 111810101015**

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2015**



**IMPLEMENTASI ALGORITMA MODEL *ROBUST* FAKTOR UNTUK
TABEL DUA ARAH PADA PROGRAM R**

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Matematika (S1)
dan mencapai gelar Sarjana Sains

oleh

**Jefri Ardian
NIM 1118101015**

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2015**

PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Allah SWT yang telah memberikan kehidupan sempurna ini;
2. Bapak Senam dan Ibu Tukinah yang telah memberikan doa, cinta dan kasih sayangnya dalam perjalanan hidup saya;
3. bapak dan ibu guru sejak taman kanak-kanak sampai perguruan tinggi, yang telah memberikan ilmu dan membimbing dengan penuh kesabaran;
4. Almamater Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam.

MOTTO

“Barangsiapa bertawakal kepada Allah niscaya Allah menjadikan baginya kemudahan dalam urusannya.”

(QS. At – Thalaq: 65)^{*)}

“Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya”

(QS. Al-Baqarah: 286)^{**)}

*) Departemen Agama Republik Indonesia. 2006. *Al-Qur'an Al – Karim dan Terjemah Bahasa Indonesia*. Kudus: MENARA KUDUS.

***) Departemen Agama Republik Indonesia. 2006. *Al-Qur'an dan Terjemahannya*. Surabaya: Duta Ilmu Surabaya.

PERNYATAAN

Saya yang bertanda tangan di bawah ini :

nama : Jefri Ardian

NIM : 111810101015

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Implementasi Algoritma Model *Robust* Faktor untuk Tabel Dua Arah pada Program R” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi manapun dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, November 2015

Yang menyatakan,

Jefri Ardian

NIM 111810101015

SKRIPSI

**IMPLEMENTASI ALGORITMA MODEL *ROBUST* FAKTOR UNTUK
TABEL DUA ARAH PADA PROGRAM R**

oleh
Jefri Ardian
NIM 111810101015

Pembimbing :

Dosen Pembimbing Utama : Dr. Alfian Futuhul Hadi, S.Si., M.Si

Dosen Pembimbing Anggota : Dian Anggraeni, S.Si., M.Si

PENGESAHAN

Skripsi berjudul “Implementasi Algoritma Model *Robust* Faktor untuk Tabel Dua Arah pada Program R” telah diuji dan disahkan pada :

hari, tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas
Jember.

TimPenguji :

Ketua,

Sekretaris,

Dr. Alfian Futuhul Hadi, S.Si., M.Si
NIP. 19740719 200012 1 001

Dian Anggraeni, S.Si., M.Si
NIP. 19820216 200604 2 002

Penguji I,

Penguji II,

Prof. Drs. I Made Tirta, M.Sc., Ph.D
NIP. 19591220 198503 1 002

M. Ziaul Arif, S.Si., M.Sc
NIP. 19850111 200812 1 002

Mengesahkan

Dekan,

Prof. Drs. Kusno, DEA., Ph.D
NIP. 19610108 198602 1 001

RINGKASAN

Implementasi Algoritma Model *Robust* Faktor untuk Tabel Dua Arah pada Program R; Jefri Ardian, 111810101015; 2015; 39 halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Robust Faktor analitik merupakan salah satu teknik statistika untuk menganalisis tentang saling ketergantungan dari beberapa variabel secara simultan pada data yang mengandung *outliers*. Tujuan *Robust* Faktor analitik adalah untuk menyederhanakan dari bentuk hubungan antara beberapa variabel yang diteliti menjadi sejumlah faktor yang lebih sedikit dari pada variabel yang diteliti.

Metode yang digunakan dalam *robust* faktor analitik pada tabel dua arah yaitu dengan menggunakan estimasi *Robust Alternating Regression* (RAR). RAR merupakan penduga iteratif untuk mendapatkan nilai penduga *loading* dan penduga skor yang dilakukan secara bergantian atau bolak-balik. *Software* statistik yang digunakan untuk analisis data menggunakan S-plus yang merupakan *software* statistik yang komersil dan program R yang merupakan *software* statistik *open-source*. Secara umum sintaks dari bahasa R adalah ekuivalen dengan bahasa pemrograman S, sehingga sebagian besar keperluan analisis statistika dan pemrograman dengan R adalah hampir identik dengan perintah yang dikenal di S-plus. Dengan menggunakan *software* statistik S-plus dan program R, penelitian ini bertujuan untuk mengevaluasi hasil analisis fungsi-fungsi yang digunakan pada *script* yang dibuat oleh Croux *et al.* (2003) dengan menggunakan estimasi *Robust Alternating Regression* (RAR).

Data yang digunakan dalam penelitian ini adalah data penelitian perbenihan padi di beberapa kabupaten di Pulau Jawa menggunakan 10 varietas dan 8 lokasi percobaan. Kemudian data diolah dengan menggunakan *script* yang disesuaikan pada *software* S-plus dan program R. Perlakuan untuk menjalankan *software* S-plus dan program R dilakukan dengan *me-restart* ulang untuk setiap menjalankan

software S-plus dan program R. Hasilnya adalah jumlah iterasi S-plus lebih sedikit dari pada program R. Biplot yang dihasilkan dari S-plus dan program R mempunyai kemiripan dengan informasi pengaruh interaksi yang diberikan S-plus sebesar 73,33% sedangkan program R sebesar 70,9%. Berdasarkan MSE, program R memiliki MSE yang lebih kecil daripada S-plus. MSE yang dihasilkan oleh S-plus sebesar 0,2786216 dan program R sebesar 0,001381457. Selain itu, program R lebih sensitif untuk mendeteksi adanya *outliers* pada data dilihat dari bobot yang diberikan pada efek baris dan efek kolom dan boxplot yang direpresentasikan dari efek baris dan efek kolom.

PRAKATA

Alhamdulillah, puji syukur kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga tugas akhir yang berjudul “Implementasi Algoritma Model *Robust* Faktor untuk Tabel Dua Arah pada Program R” dapat terselesaikan dengan baik. Tugas akhir ini disusun untuk memenuhi syarat menyelesaikan pendidikan strata satu (S1) di Jurusan Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Penyusunan skripsi ini tidak terlepas dari bantuan berbagai pihak, baik secara langsung maupun tidak langsung. Oleh karena itu, penulis menyampaikan terima kasih kepada :

1. Dr. Alfian Futuhul Hadi, S.Si., M.Si selaku Dosen Pembimbing Utama dan Dian Anggraeni S.Si., M.Si selaku Dosen Pembimbing Anggota yang telah memberikan bimbingan secara intensif dan bantuan untuk penyempurnaan skripsi ini;
2. Prof. Drs. I Made Tirta, M.Sc., Ph.D dan M. Ziaul Arif, S.Si., M.Sc selaku Dosen Penguji yang telah memberikan kritik dan saran yang membangun untuk penyempurnaan skripsi ini;
3. seluruh dosen dan karyawan Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam yang telah memberikan ilmu serta fasilitas yang membantu selama proses perkuliahan berlangsung;
4. Bapak Senam dan Ibu Tukinah yang selalu memberikan dukungan baik lahir maupun batin;
5. sahabati tercinta Niya Liyani yang telah memberikan dukungan serta motivasi dalam pembuatan skripsi ini;
6. HIMATIKA “Geokompstat” yang telah memberikan berbagai pengalaman berorganisasi selama menjadi mahasiswa di Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam

7. UKM KSR PMI Unit Universitas Jember yang memberikan berbagai pengalaman dan *softskill* yang bermanfaat dalam kehidupan;
8. para akademisi (Yulio, Haki, Darul, Rafi', Ivan, Hendri, Emil, Feri, Rian, Zulfi, Saiful) yang senantiasa memberi dukungan dalam proses menyelesaikan tugas akhir;
9. keluarga besar KRAMAT'11 yang tidak dapat saya sebut satu per satu yang selalu menemani dan memberikan dukungan dalam hal apapun.

Penulis menyadari bahwa dalam menyusun skripsi ini masih terdapat kekurangan, baik isi maupun susunannya. Oleh karena itu, penulis mengharapkan saran dan kritik demi penyempurnaan skripsi ini. Akhirnya penulis berharap semoga skripsi ini dapat memberi manfaat dan sumbangan bagi pembaca.

Jember, November 2015

Jefri ardian

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PERSEMBAHAN	ii
HALAMAN MOTTO	iii
HALAMAN PERNYATAAN.....	iv
HALAMAN PEMBIMBINGAN.....	v
HALAMAN PENGESAHAN.....	vi
RINGKASAN	vii
PRAKATA	ix
DAFTAR ISI.....	xi
DAFTAR TABEL	xiii
DAFTAR GAMBAR.....	xiv
DAFTAR LAMPIRAN	xv
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan	3
1.4 Manfaat	3
BAB 2. TINJAUAN PUSTAKA.....	4
2.1 Analisis Faktor	4
2.2 <i>Outliers</i> (Pencilan)	4
2.3 <i>Robust Principal Component Analysis (RobPCA)</i>.....	5
2.4 <i>Robust Alternating Regression (RAR) Estimator</i>	7
2.5 <i>Software S-plus</i>	9
2.6 Program R.....	10
2.7 Analisis <i>Robust Biplot</i>	10
2.8 <i>Mean Square Error (MSE)</i>.....	11

2.9 Implementasi Algoritma pada S-plus.....	12
BAB 3. METODE PENELITIAN.....	16
3.1 Data	16
3.2 Desain Penelitian	16
3.3 Struktur Fungsi Program R.....	19
BAB 4. HASIL DAN PEMBAHASAN.....	21
4.1 Implementasi Algoritma pada Program R	21
4.1.1 Algoritma Weight.w11	21
4.1.2 <i>Robust Principal Komponent Analysis</i> (RobPCA).....	23
4.1.3 Algoritma Twoway.rob	24
4.2 Fungsi-fungsi yang digunakan pada <i>Script S-plus</i> dan Program R .	31
4.3 Aplikasi <i>Script</i> pada Data Riil dengan S-plus dan Program R	32
4.4 Interpretasi Biplot.....	33
4.5 Identifikasi <i>Outliers</i>	35
BAB 5. PENUTUP.....	37
5.1 Kesimpulan	37
DAFTAR PUSTAKA	38
LAMPIRAN.....	40

DAFTAR TABEL

	Halaman
4.1 Fungsi-fungsi yang digunakan pada <i>Script</i> S-plus dan Program R.....	31
4.2 Nilai Hasil Model <i>Robust</i> Faktor	32
4.3 Pembobot bagi Pengaruh Baris terhadap Pencilan pada S-plus	35
4.4 Pembobot bagi Pengaruh Kolom terhadap Pencilan pada S-plus	35
4.5 Pembobot bagi Pengaruh Baris terhadap Pencilan pada Program R.....	36
4.6 Pembobot bagi Pengaruh Kolom terhadap Pencilan pada Program R.....	36

DAFTAR GAMBAR

	Halaman
2.1 Bagan Alur Model <i>Robust</i> Faktor Analitik pada S-plus	13
3.1 Skema alur penelitian	18
4.1 Biplot S-plus	33
4.2 Biplot Program R	34
4.3 (a) Boxplot Pengaruh Baris (Genotip) Data Pembenihan Padi dengan S-plus	35
4.3 (b) Boxplot Pengaruh Kolom (Lingkungan) Data Pembenihan Padi dengan S-plus	35
4.4 (a) Boxplot Pengaruh Baris (Genotip) Data Pembenihan Padi dengan Program R	36
4.4 (b) Boxplot Pengaruh Kolom (Lingkungan) Data Pembenihan Padi dengan Program R	36

DAFTAR LAMPIRAN

	Halaman
A. Data Perbenihan Padi	40
B. Script Data Perbenihan Padi pada S-plus	41
B.1 Script Weight.w11	41
B.2 Script Prcomp.rob dan Keluarannya	41
B.3 Script Twoway.rob dan Keluarannya	42
C. Script Data Perbenihan Padi pada Program R	55
C.1 Script Weight.w11	55
C.2 Script PcaProj dan Keluarannya	56
C.3 Script Twoway.rob dan Keluarannya	56

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Regresi merupakan suatu metode statistika yang digunakan untuk menyelidiki pola hubungan antara dua atau lebih variabel. Apabila pola hubungan yang diselidiki terdiri dari satu variabel respon dan satu variabel bebas maka disebut regresi linier sederhana. Sedangkan regresi linier berganda adalah regresi linier yang terdiri dari satu variabel respon dan lebih dari satu variabel bebas. Regresi adalah metode statistika yang baik untuk memodelkan suatu data yang tidak mengandung pencilan atau *outliers*.

Jika data yang diteliti mengandung suatu *outlier*, maka model regresi tidak memenuhi asumsi dan tidak cocok untuk data yang akan dimodelkan, karena nilai koefisien dari model regresi dipengaruhi oleh adanya *outlier*. Model yang dihasilkan tidak dapat digunakan untuk memprediksi, sebelum *outlier* yang ada dapat diatasi. Diperlukan analisis yang mampu mengakomodir adanya pencilan yaitu dengan pendekatan *robust* analitik. Regresi *robust* merupakan metode regresi yang digunakan ketika residual berdistribusi tidak normal dan mengandung beberapa *outlier* yang berpengaruh pada model (Ryan, 1997). Pada regresi *robust*, taksiran yang *robust* terhadap *outlier* akan dicari sehingga *outlier* yang ada tidak akan diabaikan. Salah satu regresi *robust* yang digunakan adalah faktor analitik.

Faktor analitik merupakan suatu teknik untuk menganalisis tentang saling ketergantungan dari beberapa variabel secara simultan dengan tujuan untuk menyederhanakan dari bentuk hubungan antara beberapa variabel yang diteliti menjadi sejumlah faktor yang lebih sedikit dari pada variabel yang diteliti (Suliyanto, 2005). Pada faktor analitik, variabel bebas dan terikatnya tidak dibedakan. Keuntungan dari penggunaan *robust* faktor analitik adalah tabel data yang mengandung pencilan dapat diperiksa lebih mudah (Hawkins *et al.*, 2001). Analisis data dengan pendekatan *robust* analitik dilakukan pada data yang akan

disajikan dalam bentuk tabel dua arah. Tabel dua arah atau komponen adalah tabel yang menunjukkan dua kategori atau dua karakteristik. Metode yang digunakan dalam *robust* faktor analitik pada tabel dua arah yaitu dengan menggunakan estimasi *Robust Alternating Regression* (RAR). RAR merupakan penduga iteratif untuk mendapatkan nilai penduga *loading* dan penduga skor yang dilakukan secara bergantian atau bolak-balik.

Beberapa penelitian sebelumnya mengenai *Robust Alternating Regression* (RAR) diantaranya Croux *et al.* (2003) dan Hadi (2011). Pada penelitian ini dengan menggunakan estimasi *Robust Alternating Regression* (RAR) akan dilakukan analisis pada tabel dua arah dengan menggunakan data pembenihan padi di beberapa kabupaten di Pulau Jawa. Data yang diperoleh akan diolah dengan menggunakan *software* statistik. Tersedia banyak *software* statistik yang bisa digunakan untuk melakukan analisis data diantaranya dengan menggunakan S-plus yang merupakan *software* statistik yang komersil dan program R yang merupakan *software* statistik *open-source*.

S-plus merupakan salah satu *software* statistik yang menggunakan bahasa pemrograman S. S-plus merupakan salah satu *software* statistik yang masih jarang penggunaannya seperti pada bidang pertanian, perikanan, ekonomi, dan lain-lain. S-plus dapat menangani data, *functions* dan *fitted models* sebagai *objects*, sehingga membuat analisis data menjadi lebih fleksibel (Burn, 1998). Sedangkan program R merupakan suatu sistem analisis data statistik dengan menggunakan bahasa R yang merupakan versi *open-source* dari bahasa pemrograman S. Fungsi dan kemampuan dari R sebagian besar diperoleh melalui *Add-on packages/library*. Suatu *library* adalah kumpulan perintah atau fungsi yang dapat digunakan untuk melakukan analisis tertentu (Suhartono, 2008). Secara umum sintaks dari bahasa R adalah ekuivalen dengan bahasa pemrograman S, sehingga sebagian besar keperluan analisis statistika dan pemrograman dengan R adalah hampir identik dengan perintah yang dikenal di S-plus. Penelitian sebelumnya mengenai *Robust Alternating Regression* (RAR) oleh Croux *et al.* (2003) menggunakan S-plus untuk analisis data. Dengan menggunakan program R, peneliti akan mentransformasi fungsi-fungsi yang digunakan pada *script* yang

dibuat oleh Croux *et al.* (2003) dengan menggunakan estimasi *Robust Alternating Regression* (RAR) dan mengevaluasi hasil penanganan *outliers software* S-plus 8.0 dan program R 2.7.0 dengan menggunakan data perbenihan padi di beberapa kabupaten di Pulau Jawa untuk mengetahui kemampuan *software*.

1.2 Rumusan Masalah

Rumusan masalah dari penelitian ini adalah sebagai berikut :

- a. bagaimana mentransformasi fungsi-fungsi yang digunakan pada *script* penanganan *outlier* pada data tabel dua arah menggunakan *robust* faktor analitik ke dalam program R yang *open-source*?
- b. bagaimana evaluasi hasil penanganan *outlier* pada data tabel dua arah menggunakan *robust* faktor analitik dengan menggunakan *software* S-plus 8.0 dan program R 2.7.0 untuk mengetahui kemampuan *software* dengan menggunakan data perbenihan padi di beberapa kabupaten di Pulau Jawa?

1.3 Tujuan

Adapun tujuan dari penelitian ini adalah sebagai berikut :

- a. mendapatkan *script* penanganan *outlier* pada data tabel dua arah menggunakan *robust* faktor analitik hasil transformasi dari S-plus ke program R,
- b. memperoleh hasil evaluasi penanganan *outlier* pada data tabel dua arah menggunakan *robust* faktor analitik pada *software* S-plus dan program R.

1.4 Manfaat

Adapun manfaat yang dapat diperoleh dari penelitian ini adalah sebagai berikut :

- a. menghindari adanya pembuangan atau pengabaian *outliers* pada *robust* faktor analitik,
- b. menyediakan alternatif analisis statistik yang *open-source* untuk menghindari adanya pembuangan atau pengabaian *outliers* pada *robust* faktor analitik.

BAB 2. TINJAUAN PUSTAKA

2.1 Analisis Faktor

Analisis faktor merupakan suatu teknik untuk menganalisis tentang ketergantungan dari beberapa variabel secara simultan dengan tujuan untuk menyederhanakan bentuk hubungan antara beberapa variabel yang diteliti menjadi sejumlah faktor yang lebih sedikit dari variabel yang diteliti. Hal ini berarti, analisis faktor dapat juga menggambarkan tentang struktur data dari suatu penelitian (Suliyanto, 2005).

Analisis faktor adalah suatu teknik interdependensi, sehingga tidak ada pembagian variabel menjadi variabel bebas dan variabel tergantung dengan tujuan utama yaitu mendefinisikan struktur yang terletak diantara variabel-variabel dalam analisis. Tujuan umum dari teknik analisis faktor adalah menemukan suatu cara untuk mereduksi informasi yang terkandung didalam sejumlah variabel-variabel original ke dalam set variabel yang lebih kecil dari dimensi-dimensi gabungan dan baru. Pada prinsipnya analisis faktor digunakan untuk mengelompokkan beberapa variabel yang memiliki kemiripan untuk dijadikan satu faktor, sehingga dimungkinkan dari beberapa atribut yang mempengaruhi satu komponen variabel dapat diringkas menjadi beberapa faktor utama yang jumlahnya lebih sedikit (Hair *et al.*, 2010).

2.2 Outlier (Pencilan)

Definisi *outlier* adalah satu atau lebih pengamatan yang nilainya menyimpang cukup jauh dari pengamatan lain. Data *outlier* adalah pengamatan yang dapat diidentifikasi secara jelas yang berbeda dari pengamatan yang lain, namun data *outlier* dapat menunjukkan karakteristik dari populasi sehingga tidak harus diabaikan atau dibuang. Menurut Kriegel (2010) suatu nilai dianggap sebagai *outliers*, apabila menyimpang lebih dari tiga kali standar deviasi. Apabila data *outlier* bermasalah, tidak mewakili populasi dan bertentangan dari tujuan

analisis, maka secara serius dapat memberikan hasil uji statistik yang berbeda. Pemeriksaan atau identifikasi *outlier* pada data harus dilakukan karena *outlier* memberikan pengaruh pada ragam dan setelah *outlier* teridentifikasi maka dapat diputuskan untuk mempertahankan atau menghapus *outlier* tersebut (Hair *et al.*, 1992).

Identifikasi *outlier* umumnya didasarkan pada *Mahalanobis Distance* (MD). Metode MD adalah suatu metode yang digunakan untuk mendapatkan suatu data dengan jarak tertentu terhadap mean data tersebut, sehingga diperoleh suatu penyebaran data yang memiliki pola terhadap nilai mean. Metode ini didasarkan pada korelasi antara variabel dengan pola yang berbeda yang dapat diidentifikasi dan dianalisis. Metode ini memiliki nilai penguat yang fleksibel sehingga mudah disesuaikan dengan perubahan kondisi. Penguat dari metode ini ada pada nilai kovariannya. Secara umum MD dapat didefinisikan sebagai berikut :

$$MD_i = \sqrt{(x_i - \bar{x})^T S^{-1} (x_i - \bar{x})} \quad (2.1)$$

dimana,

x_i = data objek ke- i

\bar{x} = vektor rata-rata

S = matriks *variance-covariance*

untuk $i = 1, \dots, n$, x_i merupakan *outlier* jika $MD > X_{\alpha;p}^2$. Prinsip MD adalah menghitung jarak di ruang multidimensional antara sebuah pengamatan dengan pusat dari semua pengamatan (Hair *et al.*, 1998).

2.3 Robust Principal Component Analysis (RobPCA)

Principal Component Analysis (PCA) atau analisis komponen utama adalah metode analisis peubah ganda yang bertujuan memperkecil dimensi peubah asal sehingga diperoleh peubah baru (komponen utama) yang tidak saling berkorelasi, tetapi menyimpan sebagian besar informasi yang terkandung pada peubah asal (Jolliffe, 2002). Terdapat tiga keunggulan analisis komponen utama yaitu memiliki kombinasi linear dengan *Mean Square Error* (MSE) optimal dalam meringkas rangkaian vektor berdimensi tinggi menjadi rangkaian vektor berdimensi rendah dan kemudian membentuknya kembali, parameter model dapat secara langsung

dihitung dari data dan peringkasan mudah dioperasikan untuk menunjukkan parameter model (Chen, 2002).

Meskipun mempunyai beberapa keunggulan, analisis komponen utama akan sangat dipengaruhi dengan adanya *outliers* pada data. Kelemahan analisis komponen utama dengan adanya *outliers* menyebabkan diperlukannya metode analisis komponen utama yang *robust* terhadap adanya *outliers*. Analisis komponen utama yang *robust* didapatkan dengan mengganti matriks varian kovarian dengan suatu estimator yang *robust*. Estimator *robust* yang diinginkan adalah memiliki *breakdown point* (tingkat *robust*) yang tinggi. *Breakdown point* merupakan proporsi terkecil dari pencemar yang mampu mempengaruhi estimator, yaitu nilainya mengalami pergeseran yang cukup jauh dari $t(X)$. Sifat lainnya yang diinginkan adalah *affine equivariant*. Suatu estimator bagi ukuran pemusatan, yaitu $t(X)$ dan ukuran simpangan, yaitu $C(X)$ adalah *affine equivariant* jika dan hanya jika untuk setiap vektor baris $b \in \mathfrak{R}^p$ (konstanta dalam ruang berdimensi p) dan setiap matriks non-singular $A_{p \times p}$ berlaku,

$$\begin{aligned} t(AX + b) &= At(X) + b, \\ C(AX + b) &= AC(X)A^t \end{aligned} \quad (2.2)$$

Estimator *robust Minimum Volume Ellipsoid* (MVE) merupakan pasangan $t(X)$ dan $C(X)$, dimana $t(X)$ merupakan vektor rata-rata dan $C(X)$ merupakan matriks $p \times p$ simetris definit-positif, dari suatu sub-sampel berukuran h pengamatan dimana *volume ellipsoid* dari sub-sampel tersebut adalah yang minimal. Dengan $h_0 \leq h \leq n$ dan h_0 merupakan nilai integer terkecil dari $\left(\frac{(n+p+1)}{2}\right)$,

$$\begin{aligned} MVE \approx \min\{m_j^{2p} \det(C(X)_j)\}^{1/2}, \\ j = 1, \dots, \binom{n}{h} \end{aligned} \quad (2.3)$$

(Rousseeuw, 1985).

Analisis komponen utama yang *robust* juga menggunakan metode *projection-pursuit* (PP). Tujuan digunakannya metode *projection-pursuit* untuk mendapatkan struktur pada data peubah ganda dengan memproyeksikannya pada subruang berdimensi lebih rendah. PP tepat digunakan untuk menganalisis data

dengan jumlah peubah yang besar. Sub-ruang berdimensi rendah dipilih dengan memaksimalkan indeks proyeksi tertentu (Huber, 1985).

2.4 Robust Alternating Regression (RAR) Estimator

Matriks Y berukuran $n \times p$ digunakan untuk melambangkan catatan yang berisi peubah pengamatan atau karakteristik (pada kolom) atas beberapa individu (objek/case) pada barisnya. Suatu skor faktor dicatat sebagai f_{ik} , dengan vektor skor ke- i diberikan sebagai $f_i = (f_{i1}, \dots, f_{ik})^T$, sedangkan *loading* vektor ke- j adalah $\lambda_j = (\lambda_{j1}, \dots, \lambda_{jk})^T$ keduanya tidak diketahui. Vektor dari seluruh *loading* dan skor faktor dilambangkan dengan $\theta = (f_1^T, \dots, f_n^T, \lambda_1^T, \dots, \lambda_p^T)$, dan

$$\hat{y}_{ij}(\theta) = \sum_{k=1}^k f_{ik} \lambda_{kj} = f_i^T \lambda_j = \lambda_j^T f_i \quad (2.4)$$

adalah nilai dugaan (*fitted value*) bagi y_{ij} , dengan memilih θ sedemikian sehingga nilai dugaan dan nilai aktual dari matriks sangat dekat, didefinisikan nilai dugaan vektor skor \hat{f}_i dan $\hat{\lambda}_j$ sebagai nilai dugaan vektor *loading*. Nilai dugaan matriks data \hat{Y} dapat didekomposisi menjadi $\hat{Y} = \hat{F} \hat{\Lambda}^T$ dengan baris-baris dari \hat{F} adalah nilai dugaan skor dan baris-baris dari $\hat{\Lambda}^T$ adalah nilai dugaan *loading*.

Rank dari \hat{Y} sebesar-besarnya adalah $k < p$, sedangkan rank dari Y adalah sebesar p . Pendekatan metode kuadrat terkecil adalah meminimumkan kuadrat sisaan :

$$\hat{\theta}_{LS} = \operatorname{argmin}_{\theta} \sum_{i=1}^n \sum_{j=1}^p (y_{ij} - \hat{y}_{ij}(\theta))^2 \quad (2.5)$$

dan menghasilkan nilai dugaan \hat{Y} yang dapat dilihat sebagai pendekatan “terbaik” (dalam kacamata kuadrat terkecil) terhadap matriks Y oleh matriks ber-*rank* k . Teorema Eckart-Young (Gower dan Hand, 1996) menyebutkan bahwa aproksimasi terbaik dapat diperoleh melalui dekomposisi nilai singular (*singular value decomposition/ SVD*) terhadap matriks $Y = UDV^T$, dengan mengganti semua nilai singular dalam matriks D dengan nol kecuali untuk k buah nilai singular terbesar pertama. Diperoleh D_k dan akhirnya $\hat{Y} = UD_kV^T$, dengan menyatakan $\hat{F} = \sqrt{n}U$ dan $\hat{\Lambda} = VD_k / \sqrt{n}$ diperoleh apa yang disebut solusi komponen utama untuk masalah faktor analisis (Johnson dan Wichern, 1998).

Matriks peragam contoh dari vektor dugaan adalah $\hat{F}^T \hat{F} = I_k$ dimana hal ini konsisten dengan asumsi awal $\text{Cov}(F) = I_k$. Nilai dugaan \hat{F} dan $\hat{\Lambda}^T$ hanya dinyatakan secara khusus untuk suatu transformasi linier, karena $\hat{Y} = (\hat{F}^T T^T)(\hat{\Lambda} T^{-1})^T$ untuk sembarang matriks T non-singular berukuran $k \times k$ matriks, maka $\hat{F}^T T^T$ dan $\hat{\Lambda} T^{-1}$ mendapat nilai yang sama bagi fungsi tujuan $\hat{\theta}_{LS}$. Meskipun demikian nilai dugaan \hat{Y} yang diperoleh bersifat unik, jika ditambahkan suatu kendala bahwa dugaan matriks peragam bagi vektor skor adalah matriks satuan, maka dugaan \hat{F} dan $\hat{\Lambda}^T$ menjadi suatu transformasi orthogonal sehingga matriks $\hat{\Lambda} \hat{\Lambda}^T$ juga unik. Karena kriteria kuadrat terkecil memberikan pembobot yang berlebih kepada residu yang bernilai besar, gagasan awal mengatasinya adalah menggunakan kriteria simpangan mutlak terkecil (*Least Absolute Deviations*) atau disebut kriteria L1 yang diketahui memberikan penduga aditif bagi tabel dua arah yang robust (Terbeck dan Davies, 1998).

Kriteria L1 menghasilkan penduga yang meminimumkan simpangan mutlak sebagai berikut :

$$\hat{\theta}_{L1} = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^n \sum_{j=1}^p |y_{ij} - \hat{y}_{ij}(\theta)| \quad (2.6)$$

Nilai \hat{F} dan $\hat{\Lambda}^T$ dapat diperoleh secara optimal jika dipenuhinilai dugaan \hat{f}_i meminimumkan $\sum_{j=1}^p |y_{ij} - \hat{f}_i^T \hat{\lambda}_j|$ dan \hat{y}_j meminimumkan $\sum_{i=1}^n |y_{ij} - \hat{f}_i^T \lambda_j|$, oleh karena itu, pendekatan ini tidaklah meminimumkan kedua jumlah mutlak pada kriteria L1 secara bersamaan, melainkan dengan menetapkan indeks- j dan skor f_i yang kemudian mencari λ_j yang meminimumkan $\sum_{i=1}^n |y_{ij} - \hat{f}_i^T \lambda_j|$. Meminimumkan $\sum_{i=1}^n |y_{ij} - \hat{f}_i^T \lambda_j|$ dilakukan satu persatu untuk setiap $j=1, \dots, p$ berkenaan dengan kriteria L1 untuk skor yang ditetapkan. Analog dengan itu, untuk *loading* λ_j yang ditetapkan, mencari f_i yang meminimumkan $\sum_{j=1}^p |y_{ij} - \hat{f}_i^T \lambda_j|$ (untuk setiap $i = 1, \dots, n$ bergantian) berkenaan dengan meminimumkan kriteria L1 pada nilai *loading* yang diberikan. Bergantian bolak-balik pada pemimuman $\sum_{i=1}^n |y_{ij} - \hat{f}_i^T \lambda_j|$ dan $\sum_{j=1}^p |y_{ij} - \hat{f}_i^T \lambda_j|$ mengarahkan pada skema pendugaan iteratif yang disebut *alternating regression* (regresi bolak-balik).

Regresi L1 sensitif terhadap pengamatan pengungkit (*leverage points*), jika terdapat skor atau *loading* yang memencil maka regresi L1 dapat sangat terpengaruh olehnya. Dengan memberikan pembobot yang bernilai lebih rendah (*downweighting*) terhadap pengamatan pengungkit didapatkan regresi L1 terboboti (*weight L1 regression*), dan menghasilkan penduga sebagai berikut :

$$\hat{\theta}_{RAR} = \operatorname{argmin}_{\theta} \sum_{i=1}^n \sum_{j=1}^p w_i(\theta) v_j(\theta) |y_{ij} - \hat{y}_{ij}(\theta)| \quad (2.7)$$

Satu fungsi objektif tunggal menduga \hat{F} dan $\hat{\Lambda}^T$ secara simultan dari baris-baris dan kolom-kolom matriks Y, untuk mendapatkan nilai dugaan $\hat{\theta}_{RAR}$ digunakan regresi bolak-balik *robust (Robust Alternating Regression)*. Penduga $\hat{\theta}_{RAR}$ tidak akan salah arah oleh adanya pengamatan *outliers*. Pembobot bagi baris, $w_i(\theta)$ didefinisikan oleh $w_i(\theta) = \min\left(1, \frac{x_k^2, 0.95}{RD_i^2}\right)$ untuk $i = 1, \dots, n$ dimana $x_k^2 0.95$ adalah batas atas bagi nilai kritis 5% sebaran khi-kuadrat dengan derajat bebas k.

2.5 Software S-plus

S-plus merupakan salah satu *software* statistik yang masih jarang penggunaannya oleh *stakeholder*, seperti pada bidang pertanian, perikanan, ekonomi, dan lain-lain. S-plus memiliki keunggulan pada beberapa point analisis, yaitu model *tree*, metode resampling, dan regresi nonparametrik (*smoothing*). S-plus dirancang oleh AT & T Bells Labs dengan menggunakan bahasa S. S-plus dapat menangani data, *functions* dan *fitted models* sebagai *objects*, sehingga membuat analisis data menjadi lebih fleksibel (Burn, 1998).

Penggunaan Microsoft Office yang kompatibel dengan S-plus, memberikan akses yang mudah untuk manipulasi data, grafik, dan statistik. Beberapa hal yang dapat dilakukan dengan menggunakan S-plus antara lain dapat memasukkan data dari hampir semua sumber, mengedit data, membuat plot dengan klik tombol, mengatur setiap detail dari grafis dan hasil yang menakjubkan, keluaran tampak profesional untuk keluaran ke dalam bentuk laporan dokumen, melakukan analisis statistik dari sistem menu, menjalankan fungsi analisis satu per satu pada baris

perintah atau dalam sejumlah fungsi menggunakan *script window*, membuat fungsi sendiri (Insighful corp., 2007).

2.6 Program R

Program R merupakan suatu sistem analisis data statistik yang komplit sebagai hasil dari kolaborasi penelitian berbagai ahli statistik di seluruh dunia. Versi awal dari R dibuat pada tahun 1992 di Universitas Auckland, New Zealand oleh Ross Ihaka dan Robert Gentleman. Pada saat ini, *source code* kernel R dikembangkan terutama oleh R Core Team yang beranggotakan 17 orang statistisi dari berbagai penjuru dunia (lihat <http://www.r-project.org/contributors.html>). Selain itu, para statistisi lain pengguna R di seluruh dunia juga memberikan kontribusi berupa kode, melaporkan bug dan membuat dokumentasi untuk R. Paket statistik R bersifat *multiplatforms*, dengan *file* instalasi *binary/file rar* yang tersedia untuk sistem operasi windows, Mac OS, Mac OS X, Linux, Free BSD, Net BSD, irix, Solaris, AIX dan HPUX. Secara umum sintaks dari bahasa R adalah ekuivalen dengan paket statistik S-plus, sehingga sebagian besar keperluan analisis statistika dan pemrograman dengan R adalah hampir identik dengan perintah yang dikenal di S-plus (Suhartono, 2008).

Fungsi dan kemampuan dari R sebagian besar diperoleh melalui *Add-on packages/library*. Suatu *library* adalah kumpulan perintah atau fungsi yang dapat digunakan untuk melakukan analisis tertentu (Suhartono, 2008). Keunggulan dari R diantaranya adalah efektif dalam pengelolaan data, terdiri dari koleksi *tools* statistik yang terintegrasi untuk analisis data dan dapat dikembangkan sesuai keperluan dan kebutuhan.

2.7 Analisis Robust Biplot

Analisis biplot adalah peragaan secara grafik dari baris dan kolom sebuah matriks data, dengan baris mewakili objek dan kolom mewakili peubah. Adapun yang diperoleh gambaran dari analisis biplot adalah kedekatan antar objek, keragaman peubah, korelasi antar peubah, dan keterkaitan peubah dengan objek (Matjjik dan Sumertajaya, 2011). Pada analisis *robust* biplot, analisis biplot dapat dibangkitkan dengan menggunakan matrik *covariance* yang *robust*. Analisis

biplot dengan menggunakan matriks *covariance* yang *robust* dilakukan dengan menduga *eigen value* dan *eigen vektor* kiri (U) dan kanan (A) sehingga hasil duga tersebut tahan terhadap data *outlier* (Hawkins *et al.*, 2001).

Gabriel (1971) mengemukakan bahwa ukuran pendekatan matriks X dengan biplot dalam bentuk :

$$\rho^2 = \frac{(\lambda_1 + \lambda_2)}{\sum_{k=1}^r \lambda_k}$$

Dengan λ_1 adalah nilai *eigen* terbesar ke-1, λ_2 adalah nilai *eigen* terbesar ke-2 dan $\lambda_k, k = 1, 2, \dots, r$ adalah nilai *eigen* ke- k . Apabila ρ^2 mendekati nilai satu, maka biplot memberikan penyajian yang semakin baik mengenai informasi data yang sebenarnya. Biplot merupakan upaya membuat gambar di ruang berdimensi banyak menjadi gambar di ruang berdimensi dua. Pereduksian dimensi ini mengakibatkan menurunnya informasi yang terkandung dalam biplot. Biplot yang dapat memberikan informasi sebesar 70% dari seluruh informasi dianggap cukup (Mattjik dan Sumertajaya, 2011).

2.8 Mean Square Error (MSE)

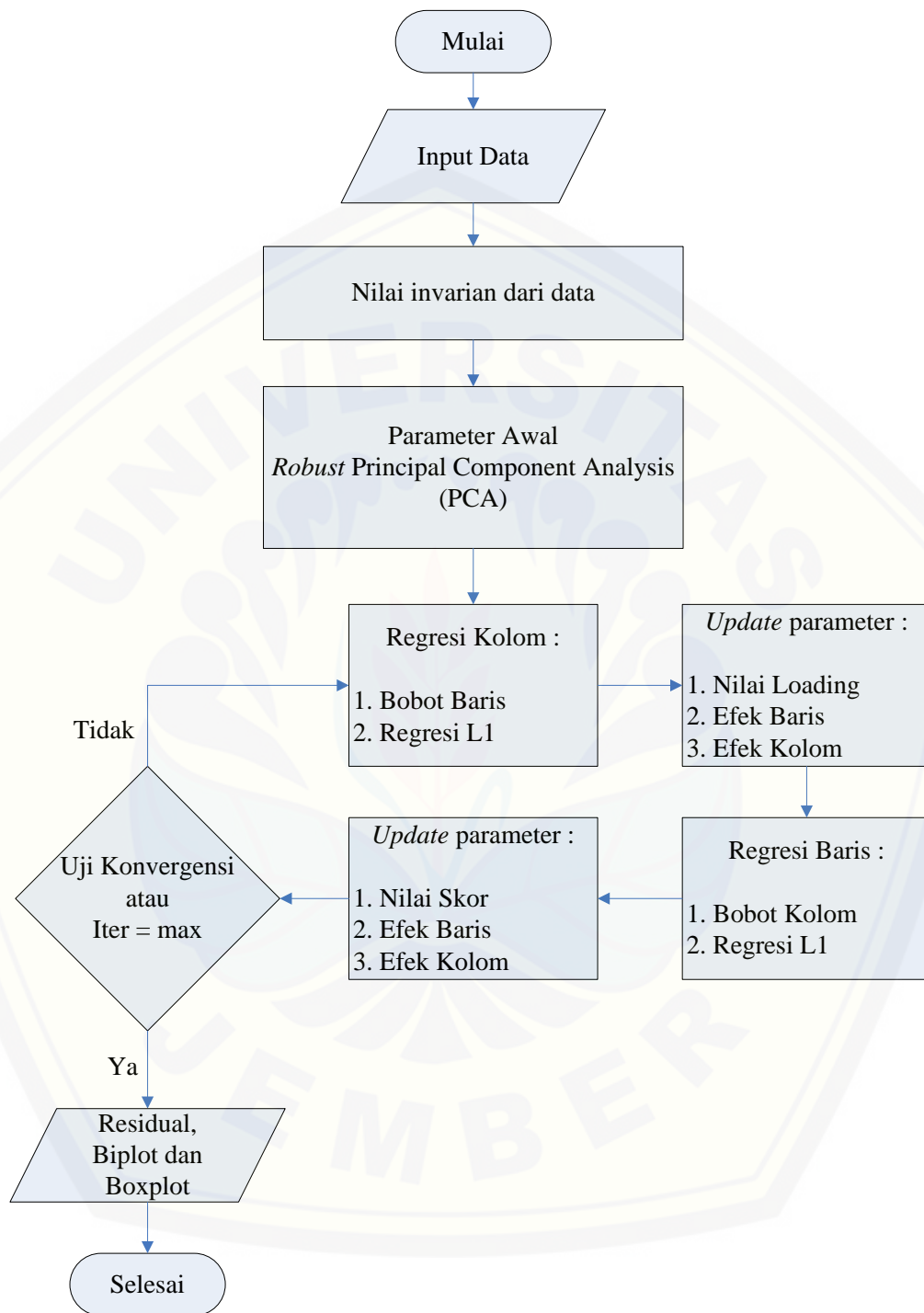
Terdapat beberapa metode yang biasa digunakan untuk menguji model manakah yang lebih baik. Namun terkadang dua metode yang berbeda dapat memberikan hasil yang berbeda, sehingga tidak ada ketetapan khusus metode manakah yang lebih baik dalam mengukur kecocokan model. MSE adalah rata-rata kuadrat *error* yang merupakan selisih antara sesuatu yang diduga dan penduganya. Masing-masing *error* atau sisaan dikuadratkan, kemudian dijumlahkan dan dibagi dengan jumlah observasi. MSE dapat digunakan untuk menentukan model yang paling baik dengan hasil dari MSE memiliki nilai yang cenderung kecil. Secara matematis MSE dapat dituliskan sebagai berikut :

$$MSE = \frac{\sum (y - \hat{y})^2}{n}$$

dengan y merupakan variabel respon, \hat{y} merupakan prediktor dan n adalah banyaknya data. Model dengan nilai MSE yang mendekati nol mengindikasikan bahwa model tersebut merupakan model yang lebih baik.






2.9 Implementasi Algoritma pada S-plus

Script robust faktor analitik yang digunakan untuk analisis data pada penelitian Croux *et al.* (2003) dengan menggunakan *software* S-plus adalah *weight.w11*, *prcomp.rob* dan *twoway.rob*. Ketiga *script* tersebut dijalankan secara bersamaan dikarenakan *script* *weight.w11* dan *prcomp.rob* merupakan bagian dari *script* *twoway.rob*. *Script* *weight.w11* berfungsi untuk menghitung nilai bobot dari baris maupun kolom data sebelum iterasi dilakukan. *Script* *prcomp.rob* berfungsi untuk menghitung nilai awal skor yang dipergunakan pada proses regresi kolom. Sedangkan *script* *twoway.rob* merupakan *script* yang digunakan untuk memodelkan data ke dalam bentuk tabel dua arah yang *robust* terhadap *outlier*. Berikut ini merupakan bagan alur model *robust* faktor analitik untuk penanganan *outliers* pada tabel dua arah menggunakan *software* S-plus :



Gambar 2.1 Bagan Alur Model *Robust* Faktor Analitik pada S-plus

Keterangan :

-  : sebagai awal dan akhir dari suatu program
-  : menunjukkan bagian arah intruksi yang dijalankan
-  : data yang akan dimasukkan ke dalam proses algoritma atau data yang dihasilkan dari proses algoritma
-  : satu atau beberapa himpunan perintah yang akan dilaksanakan secara beruntun
-  : memutuskan arah atau percabangan yang diambil sesuai dengan kondisi yang dipenuhi, yaitu benar atau salah

Bagan diatas menunjukkan bahwa alur model *robust* faktor analitik untuk penanganan *outliers* pada tabel dua arah menggunakan *software* S-plus dimulai dari menginputkan data. Dari data yang diinputkan dicari nilai *invarian* data atau nilai awal efek baris dan nilai tengah dari data berdasarkan barisnya. Kemudian dicari nilai awal dari skor menggunakan *Robust Principal Component Analysis* (RobPCA) atau *Robust Analisis Kumponen Utama* (AKU) dengan fungsi `f <- prcomp.rob(x, k, sca = "A", scores = T)$scores[, 1:k]`. Pada fungsi tersebut nilai skor yang diambil sepanjang kolom 1 sampai k . Selain itu juga didefinisikan nilai objektif sebagai nilai awal yang digunakan untuk uji kekonvergenan. Nilai awal yang diperoleh digunakan untuk mencari nilai regresi dari kolom. Pada tahap ini terlebih dahulu dihitung bobot baris dari data menggunakan fungsi `weightc1 <- weight.w11(f, n, k)` yang *downweight outlier*. Nilai bobot yang diperoleh merupakan salah satu syarat untuk melakukan regresi L1 sehingga menjadi regresi L1 yang terboboti.

Hasil dari regresi L1 yang terboboti merupakan nilai dari penduga *loading*. Setelah regresi L1 yang terboboti berdasarkan kolom selesai, dilakukan pembaruan parameter baru yang digunakan untuk regresi pada baris. Parameter yang diperbarui diantaranya adalah nilai penduga *loading*, efek baris, efek kolom, nilai skala dan rata-rata keseluruhan. Regresi pada baris dimulai dengan menghitung bobot dari kolom dengan fungsi `weightr1 <- weight.w11(1, p, k)` yang *downweight outlier*. Bobot dari kolom yang diperoleh digunakan

untuk regresi L1 pada kolom data sehingga menjadi regresi L1 yang terboboti. Fungsi dari regresi L1 yang terboboti didefinisikan sebagai `sigmfreg <- llfit (weightr1 * cbind (vecpl,1) , weightr1 * (x [i ,]- ms - b) , intercept=F)$coef`. Nilai yang dihasilkan merupakan nilai dugaan skor. Sama seperti regresi pada kolom, setelah selesai dilakukan regresi pada baris maka dilakukan pembaruan pada parameternya. Parameter yang diperbarui adalah diantaranya nilai penduga skor, efek baris, efek kolom, nilai dari skala dan nilai rata-rata keseluruhan.

Dinilai satu iterasi apabila sudah dilakukan proses regresi pada kolom dan regresi pada baris. Regresi pada kolom dan pada baris tidak dapat dilakukan secara bersamaan, nilai hasil regresi pada kolom digunakan untuk proses regresi pada baris dan hasil dari regresi baris digunakan untuk proses regresi pada kolom. Proses ini dinamakan *Robust Alternating Regression* (RAR). Setiap akhir dari iterasi dilakukan perhitungan terhadap nilai residual. Nilai residual digunakan untuk menentukan nilai objektif yang baru sebagai nilai yang akan di uji kekonvergenannya. Apabila sudah konvergen iterasi dihentikan dan apabila belum konvergen, iterasi akan berlanjut hingga iterasi maksimal. Keluaran dari *script* yang dihasilkan yaitu nilai rata-rata keseluruhan, efek baris, efek kolom, nilai skala, penduga *loading*, penduga skor, residual, bobot baris, bobot kolom, *Mean Square Error* (MSE), nilai eigen, nilai kekonvergenan, jumlah iterasi, model, metode dan fungsi pemanggil *script*. Keluaran lainnya dari *script* adalah dalam bentuk plot diantaranya biplot dan boxplot. Biplot digambarkan berdasarkan nilai residualnya dan boxplot diperoleh dari nilai efek baris dan efek kolomnya (Croux *et al*, 2003).

BAB 3. METODE PENELITIAN

3.1 Data

Data yang digunakan dalam penelitian ini adalah data penelitian perbenihan padi di beberapa kabupaten di Pulau Jawa yang digunakan oleh Aini (2015). Pada penelitian ini menggunakan 10 varietas dan 8 lokasi percobaan. Varietas padi yang diteliti didefinisikan dengan G1, G2, G3, G4, G5, G6, G7 dan G8. Sedangkan lokasi percobaannya didefinisikan dengan L1, L2, L3, L4, L5, L6, L7, L8, L9 dan L10.

3.2 Desain Penelitian

Data yang diperoleh diolah dan dianalisis dengan menggunakan S-plus dan program R. Metode analisis uji kebaikan model menggunakan *Mean Square Error* (MSE) yang dibandingkan nilai MSE-nya antara analisis dengan menggunakan S-plus 8.0 dan analisis dengan menggunakan program R 2.7.0. Alur penelitian pada tugas akhir ini dapat dijelaskan sebagai berikut :

a. *Script* Program

Script program yang digunakan pada penelitian ini diperoleh dari <http://www.statistik.tuwien.ac.at/public/filz/programs.html> yang dibuat oleh Croux *et al.* (2003). Terdapat tiga *script* yang digunakan yaitu *weight.w11*, *prcomp.rob* dan *twoway.rob* yang ketiganya merupakan *script* pada S-plus. Ketiga *script* tersebut akan diimplementasikan pada program R 2.7.0.

b. Implementasi *Script*

Script yang diperoleh dari *weight.w11*, *prcomp.rob* dan *twoway.rob* di implementasikan pada program R. Sintaks maupun fungsi-fungsi yang terdapat di dalam *script* disesuaikan dengan kondisi masing-masing program tetapi tetap mendefinisikan perintah yang sama.

c. Menginputkan Data

d. Analisis Data Menggunakan S-plus

Pada tahap ini data yang ada akan diolah berdasarkan *script* *twoway.rob*. *Script* *twoway.rob* membutuhkan *script* *weight.w11* dan *prcomp.rob* pada saat program dijalankan. Dengan kata lain, *script* *weight.w11* dan *prcomp.rob* merupakan bagian dari *script* *twoway.rob* pada saat di jalankan pada S-plus 8.0. *Script* *twoway.rob* merupakan *script* yang digunakan untuk menentukan model *robust* faktor analitik untuk penanganan *outliers* pada tabel dua arah. *Script* *weight.w11* untuk mencari nilai bobot baris maupun bobot kolom data sedangkan *script* *prcomp.rob* untuk mencari nilai awal skor sebelum iterasi.

e. Analisis Data Menggunakan Program R

Pada tahap ini data diolah dengan *script* *twoway.rob* yang telah disesuaikan dengan program R 2.7.0. Sama halnya dengan *script* S-plus, pada program R memerlukan *script* *weight.w11* dan *script* *PcaProj*. *Script* *twoway.rob* pada program R yang disesuaikan berdasarkan *script* *Croux et al.* (2003). *Script* *twoway.rob* merupakan *script* yang digunakan untuk menentukan model *robust* faktor analitik untuk penanganan *outliers* pada tabel dua arah. *Script* *weight.w11* untuk mencari nilai bobot baris maupun bobot kolom data sedangkan *script* *PcaProj* untuk mencari nilai awal skor sebelum iterasi.

f. Perbandingan *Mean Square Error* (MSE)

Model yang lebih baik dapat dilihat berdasarkan nilai MSE yang dihasilkan. Semakin kecil nilai MSE yang dihasilkan oleh model, menunjukkan bahwa model tersebut semakin baik.

g. Representasi Biplot

Pada tahap ini merepresentasikan hasil dari algoritma ke dalam bentuk biplot untuk menduga pengaruh interaksi dan menginterpretasikan model melalui biplot.

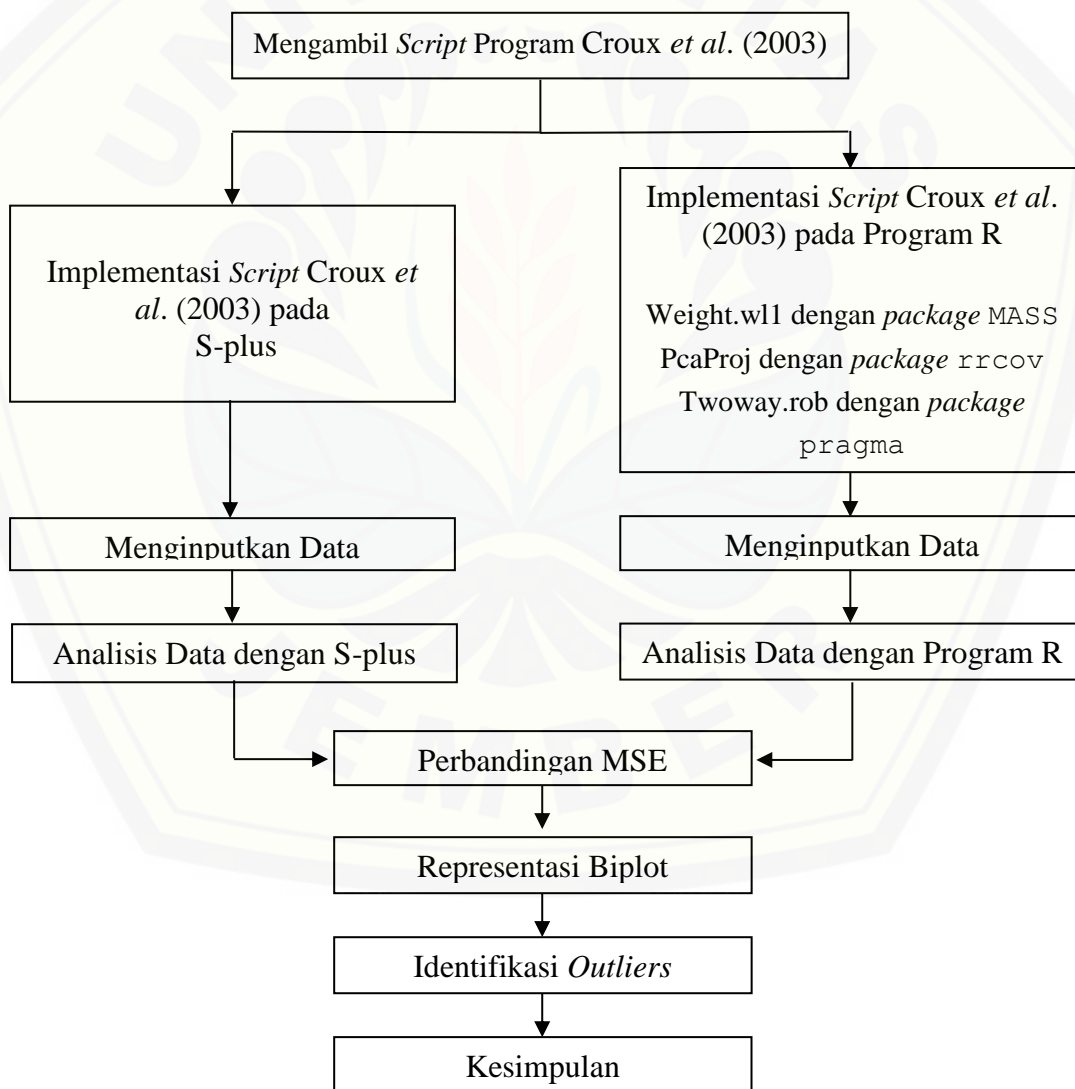
h. Identifikasi *Outliers*

Berdasarkan pembobot efek baris dan efek kolom yang diperoleh dari hasil iterasi, akan dibuat boxplot untuk menggambarkan ada tidaknya *outliers* pada data yang diolah.

i. Kesimpulan

Berdasarkan dari proses pengolahan data dapat diperoleh hasil evaluasi penanganan *outlier* pada data tabel dua arah menggunakan *robust* faktor analitik pada *software* S-plus dan program R.

Secara skematik, alur penelitian tugas akhir ini digambarkan dalam Gambar 3.1



Gambar 3.1 Skema Alur Penelitian

3.3 Struktur Fungsi Program R

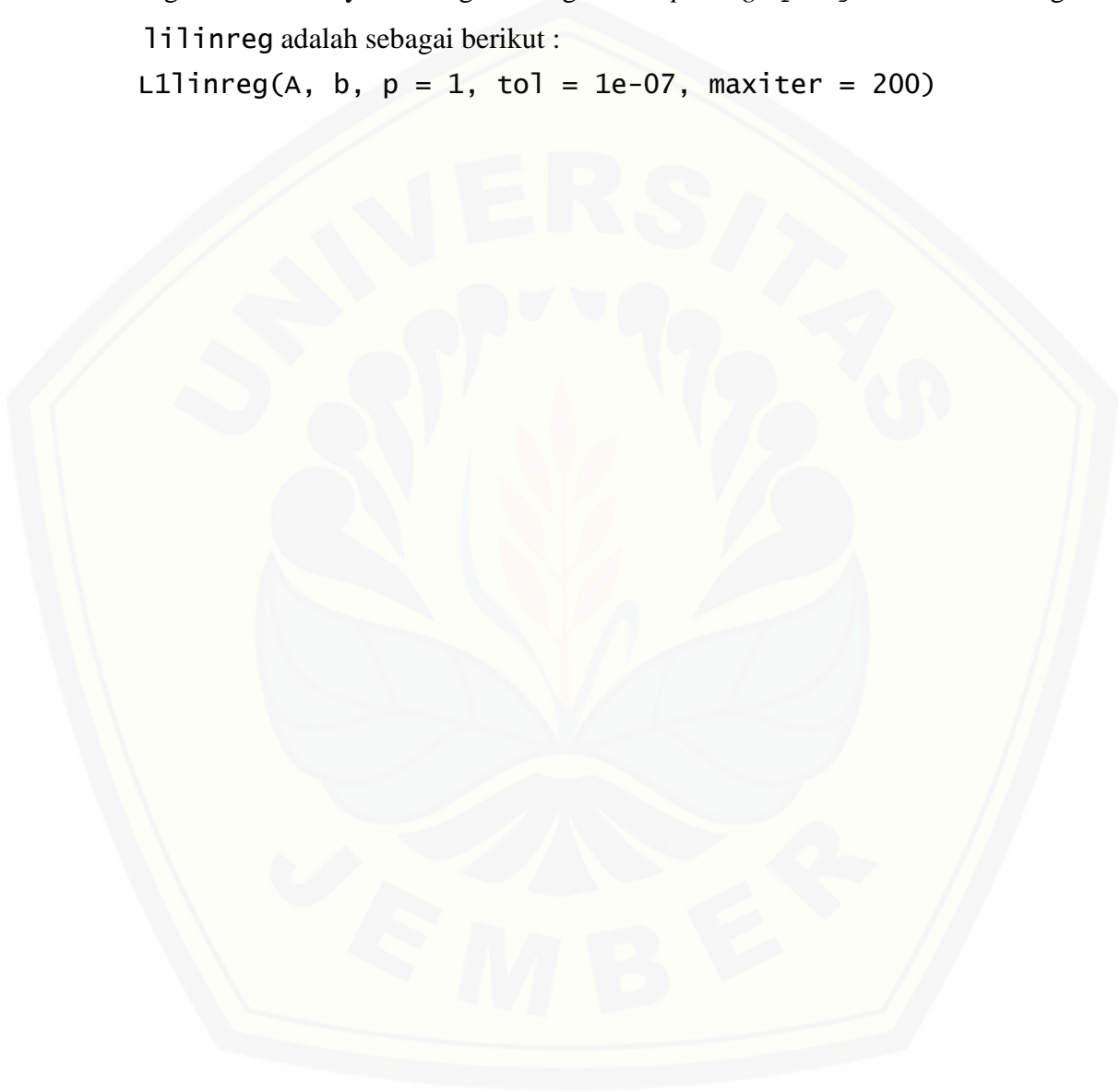
Metode yang digunakan dalam penelitian ini adalah estimasi *Robust Alternating Regression* (RAR). Proses perhitungan dipermudah dengan menggunakan bantuan salah satu *software* dalam statistika yaitu Rstudio. Cara penggunaan Rstudio mirip dengan program R. Rstudio adalah *Integrated Development Environment* (IDE) utama bagi pengembang yang menggunakan bahasa R. Uniform Resource Locator (URL) Rstudio dapat diunduh pada link <http://www.rstudio.com/ide/download/desktop>. Fungsi R yang digunakan dalam proses analisis disesuaikan dengan algoritma yang telah dibuat oleh Croux *et al.* (2003). Fungsi R yang digunakan dalam penelitian ini diantaranya ialah :

- a. `as.matrix`, digunakan untuk mengubah data yang dimasukkan ke dalam bentuk matriks. Struktur fungsi `as.matrix` adalah sebagai berikut :
`as.matrix(data)`
- b. `as.data.frame`, digunakan untuk menjadikan data yang dimasukkan sebagai kerangka data. Kerangka data ini yang menjadi struktur data fundamental yang digunakan pada proses pemodelan data. Struktur fungsi `as.data.frame` adalah sebagai berikut :
`as.data.frame(data)`
- c. `biplot`, digunakan untuk memvisualisasi hasil analisis dalam bentuk plot dua dimensi. Struktur fungsi `biplot` adalah sebagai berikut :
`biplot(x,y)`
- d. `boxplot`, digunakan untuk memvisualisasi efek baris dan efek kolom dengan plot berbentuk balok. Struktur fungsi `boxplot` adalah sebagai berikut :
`boxplot(x)`
- e. `cov.rob`, digunakan untuk menghitung penduga skala dengan *hight breakdown point* menggunakan metode *Minimum Volume Ellipsoid*. Fungsi ini terdapat pada algoritma `weight.w11` dengan menggunakan *package MASS*. Struktur fungsi `cov.rob` adalah sebagai berikut :
`cov.rob(data)`
- f. `PcaProj`, digunakan untuk analisis komponen utama yang *robust* dengan menggunakan metode *Projection Pursuit* (PP). Fungsi ini menggunakan *package rrcov*. Struktur fungsi `PcaProj` adalah sebagai berikut :

```
PcaProj(data, k = 0, kmax = ncol(data), scale=TRUE,  
na.action=na.fail, trace=FALSE)
```

- g. `l1linreg`, digunakan untuk menghitung nilai dugaan loading dan dugaan skor dengan regresi L1 (*Least Absolute Deviations*). Fungsi ini terdapat pada algoritma `twoway.rob` dengan menggunakan *package* `pragm`. Struktur fungsi `l1linreg` adalah sebagai berikut :

```
L1linreg(A, b, p = 1, tol = 1e-07, maxiter = 200)
```



BAB 5. PENUTUP

5.1 Kesimpulan

Hasil analisis yang dilakukan pada kedua software S-plus dan program R menghasilkan beberapa kesimpulan sebagai berikut :

- a. Hasil implementasi *script* Croux *et al.* (2003) pada S-plus dan program R menunjukkan beberapa perbedaan pendefinisian fungsi diantaranya adalah fungsi `rob` pada S-plus didefinisikan `cov.mve(x)`, pada program R didefinisikan `cov.rob(x)` dengan *package* MASS. S-plus menggunakan `precomp.rob` untuk *Robust Principal Component Analysis* (RobPCA), pada program R menggunakan `PcaProj` dengan *package* `rrcov`. Regresi L1 pada S-plus didefinisikan dengan `l1fit`, pada program R didefinisikan dengan `L1linreg` dengan *package* `pragm`.
- b. Hasil model robust faktor pada saat program dijalankan dengan menggunakan perlakuan yang sama untuk setiap *software*nya didapatkan bahwa hasil dari S-plus mempunyai iterasi yang lebih sedikit daripada program R. Tetapi nilai MSE dari program R lebih baik daripada S-plus untuk model *robust* faktor dengan nilai MSE-nya yang lebih kecil yaitu 0,001381457 sedangkan MSE S-plus 0,2786216. Program R dapat dikatakan lebih sensitif untuk mendeteksi adanya *outliers* .
- c. Hasil representasi biplot menggunakan S-plus tidak jauh berbeda dengan menggunakan program R. Biplot dengan S-plus memberikan informasi pengaruh interaksi sebesar 73,33% sedangkan pada program R informasi yang diberikan sebesar 70,9%. Representasi biplot yang diberikan oleh S-plus dan program R dapat dikatakan mempunyai kemiripan, sehingga informasi yang diberikan juga relatif sama. Dilihat dari plot genotip dan lingkungan yang sebagian besar sama antara lain G3, G4, G6, G7 dan G8 merupakan genotip yang relatif stabil, G1 dan G2 merupakan genotip yang relatif paling tidak stabil.

DAFTAR PUSTAKA

- Aini, E. N. 2015. "Metode Robust Singular Value Decomposition (RSVD) untuk Model AMMI dengan Data Pencilan". Tidak Diterbitkan. Skripsi. Jember : Universitas Jember.
- Borchers, H. W. 2015. Functions From Numerical Analysis and Linear Algebra, Numerical Optimization, Differential Equations, Plus Some Special Functions.[serial on line]. <https://cran.r-project.org/web/packages/pracma/index.html>. [21 September 2015].
- Burns, P. 1998. *S Poetry*. Albuquerque : University of New Mexico Press.
- Chen, C. 2002. Robust Regression and Outlier Detection with the ROBUSTREG procedure. *Statistics and Data Analysis*. **27**,265-77.
- Croux, C., Filzmozer, P., Pison, G., dan Rousseeuw, P.J. 2003. Fitting Multiplicative Models by Robust Alternating Regressions. *Statistics and Computing*. **13**, 23-36.
- Gabriel, K. R. 1971. The Biplot Graphic Display of Matrices with Application to principal component analysis. *Journal of Biometrika*. **58**, 453-467.
- Gower, J. dan Hand, D. 1996. *Biplots*. New York : Champman dan Hall.
- Hadi, A. F. 2011. Handling Outlier in Two-Way Table by Robust Alternating Regression of FANOVA Models : Towards Robust AMMI Models. *Jurnal ILMU DASAR*. **12**, 123-131.
- Hair, J. F., Anderson, R. E., Tatham, R. L., dan Black, W. C. 1992. *Multivariate Data Analysis with Reading*. 3th Edition. Macmillan Publishing Company, Inc.
- Hair, J. F., Anderson, R. E., Tatham, R. L., dan Black, W. C. 1998. *Multivariate Data Analysis*. 5th Edition. Prentice Hall, International, Inc.
- Hair, J. F., Black, W. C., Babin, B.J., dan Anderson, R. E. 2010. *Multivariate Data Analysis*. 7th Edition. NJ : Pearson Prentice Hall.
- Hawkins, D. M., Liu, Li., dan Young, S.S. 2001. Robust Singular Value Decomposition. *National Institute of Statistical Science*. Technical Report Number **122**.

- Huber, P. J. 1986. Projection Pursuit. *The Annals of Statistics*. **13**, 435-475.
- Insighful corp. 2007. *S-plus 8 for Windows User's Guide*. Seattle : Insighful Corporation.
- Johnson, R. A. dan Wichern, D. W. 1998. *Applied Multivariate Statistical Analysis*. 4th Edition. New Yersey : Prentice Hall.
- Jolliffe, I. T. 2002. *Principal Component Analysis*. 2th Edition. New York : Springer-Verlag, Inc.
- Kriegel, H. P. 2010. *Outlier Detection Techniques*. Munich : University Munchen.
- Mattjik, A. A. dan Sumertajaya, I. M. 2011. *Sidik Peubah Ganda dengan menggunakan SAS*. Bogor : IPB Press.
- Ripley, B., Venables, B., Bates, D. M., Hornil, K., Gebhardt, A., dan Firth, D. 2015. Functions and Datasets to Support Venables and Ripley, "Modern Applied Statistics with S" (4th edition, 2002). [serial on line]. <https://cran.r-project.org/web/packages/MASS/index.html>. [18 Juni 2015].
- Rousseeuw, P. J. 1985. Multivariate Estimation with High Breakdown Point. *Mathematical Statistics and Applications*. Vol. **B**, 283-297.
- Ryan, T. P. 1997. *Modern Regression Methods*. Canada : John Wiley & Sons, Inc.
- Suhartono. 2008. *Analisis Data Statistik dengan R*. Surabaya : Lab. Statistik Komputasi.
- Suliyanto. 2005. *Analisis Data Dalam Aplikasi Pemasaran*. Bogor: Ghalia Indonesia.
- Terbeck, W dan Davies, P. 1998. Interaction and outliers in the two-way analysis of variance. *The Annals of Statistics*. **26**, 1279-1305.
- Todorov, V. 2014. Robust Location and Scatter Estimation and Robust Multivariate Analysis with High Breakdown Point. [serial on line]. <https://cran.r-project.org/web/packages/rrcov/index.html>. [14 Agustus 2015].

LAMPIRAN

A. Data Perbenihan Padi

	G1	G2	G3	G4	G5	G6
L1	8,004333	9,576000	6,735000	7,997333	6,379000	7,161333
L2	15,050822	13,716323	7,773000	10,440333	5,765333	7,294333
L3	7,532333	11,866661	7,256667	8,809667	6,140667	7,094333
L4	9,773667	9,650000	7,439667	7,994000	5,314333	7,563000
L5	9,002667	6,566667	7,673667	13,133571	5,859333	7,524667
L6	7,513333	9,438333	6,452667	6,175667	6,355667	7,432667
L7	9,660667	9,118000	7,599667	8,595000	7,189667	8,112000
L8	6,699667	6,928333	6,376000	7,086000	7,088333	8,080667
L9	13,512762	8,106333	6,931333	5,880000	7,571667	8,306000
L10	7,074000	12,813361	7,249333	8,221667	5,928667	7,352000

	G7	G8
L1	8,074667	7,696000
L2	8,599333	7,201333
L3	7,481000	7,676000
L4	7,878333	7,738333
L5	10,794085	7,389667
L6	7,955333	7,715000
L7	8,591000	8,898306
L8	8,065667	7,474333
L9	8,604000	6,866667
L10	7,107667	7,720333

B. Script Data Perbenihan Padi pada S-plus

B.1 Script Weight.wl1

###Menghitung Nilai Bobot pada Data###

```
weight.wl1<-function(x, n, k, print.it = F)
{
  x <- as.data.frame(x)
  if(n == 1) {
    rob <- list(center = 0, cov = 1)
    rob$center <- median(x)
    rob$cov <- mad(x)^2
  }
  else rob <- cov.mve(x, print.it = F)
  x <- as.matrix(x)
  robdist2 <- mahalnobis(x, rob$center, rob$cov)
  weight <- qchisq(0.95, k)/robdist2
  weight <- apply(cbind(1, weight), 1, min)
  weight
}
```

B.2 Script Prcomp.rob dan Keluarannya

###Menghitung Nilai Robust Principal Component Analysis###

```
prcomp.rob<-function(X, k = 0, sca = "A", scores = F)
  $scores[, 1:k]
{
  n <- nrow(X)
  p <- ncol(X)
  if(k == 0)
    p1 <- min(n, p)
  else p1 <- k
  S <- rep(1, p1)
  V <- matrix(1:(p * p1), ncol = p1, nrow = p)
  P <- diag(p) #m <- llmedian(X)
  m <- apply(X, 2, median)
  Xcentr <- scale(X, center = m, scale = F)
  for(k in 1:p1) {
    B <- Xcentr %*% P
    Bnorm <- apply(B, 1, vecnorm)
    A <- diag(1/Bnorm) %*% B
    Y <- A %*% P %*% t(X)
    if(sca == "mad")
      s <- apply(Y, 1, mad)
    if(sca == "tau")
      s <- apply(Y, 1, scale.tau)
    if(sca == "A")
```



```

      s <- apply(Y, 1, scale.a)
      j <- order(s)[n]
      S[k] <- s[j]
      V[, k] <- A[j, ]
      if(V[1, k] < 0)
        V[, k] <- (-1) * V[, k]
      P <- P - (V[, k] %*% t(V[, k]))
    }
  if(scores) {
    list(scale = S, rotation = V, scores = Xcentr %*%
V)
  }
  else list(scale = S, rotation = V)
}

```

```

      [,1]      [,2]
1 -0.2468913 -5.181311e-001
2 -1.1927529  7.587135e+000
3 -2.6474767  2.674040e-001
4  0.2357148  1.223747e+000
5  3.3066977  1.503355e+000
6 -0.2821373 -1.646941e+000
7  0.9952834  8.493298e-001
8  1.7102814 -3.147844e+000
9  3.4998875  2.244112e+000
10 -3.7460242  1.239535e-016

```

B.3 Script Twoway.rob dan Keluarannya

Robust Faktor pada Tabel Dua Arah###

```

twoway.rob<-function(x, model = "b", k = 2, method =
"wl1fit", drawplot = T, alpha = 1,
tol= 0.0001, iter = 50, orth = T)
{
# Begin of the procedure:
call <- match.call()
if(is.matrix(x) == F)
  (stop("x is no matrix."))
n <- dim(x)[1]
p <- dim(x)[2]
if(!length(dimnames(x)[[2]]))
  dimnames(x)[[1]]<-
c("L1", "L2", "L3", "L4", "L5", "L6", "L7", "L8", "L9", "L10"
)
  dimnames(x) <- list(dimnames(x)[[1]], paste("G",
1:p, sep = ""))

```



```

        )
        x_as.matrix(x)
    if(p > n) {
# take the transposed matrix
        x <- t(x)
        transp <- p
        p <- n
        n <- transp
        transp <- T
    }
    else transp <- F
    vecn0 <- rep(0, n)
    vecn1 <- rep(1, n)
    vecp0 <- rep(0, p)
    vecp1 <- rep(1, p) #
# weights for Weighted L1 or Rob. Regression
    if((method == "wllfit") || (method == "rreg")) {
# initialize matrices for row and column weights
        weightr <- matrix(0, n, p)
        weightc <- weightr
    }
# initialize the objective function
    objold <- 100000000000 #
# choice of the regression method
    imethod <- charmatch(method, c("lsfit", "l1fit",
    "wllfit", "lmsreg",
    "ltsreg", "rreg"), nomatch = NA)
    if(!is.na(imethod))
        method <- c("lsfit", "l1fit", "wllfit", "lmsreg",
    "ltsreg",
    "rreg")[imethod]
    if(method == "wllfit") {
        if((n <= 2 * k) | (p <= 2 * k)) {
            stop("\n\nThe number of rows and columns should
at least be > 2*k for WL1fit !\n"
            )
        }
    }
}
# starting values for iteration
    if(model != "m") {
        a <- apply(x, 1, median)
        mu <- median(a)
        a <- a - mu
        b <- vecp0
    }
    if(model != "a") {
# starting with robust PCA

```

```

      f <- prcomp.rob(x, k, sca = "A", scores =
T)$scores[, 1:k] #
# starting with classical PCA
#   f <- princomp(x)$scores[, 1:k]
# starting with random values
#   f <- matrix(rnorm(n * k, 0, 1), ncol = k)
      if(k == 1) {
# k is the dimension of the multiplicatice terms
      f <- f - median(f)
      sigm <- sqrt(sum(f^2))
      f <- f/sigm
    }
    else {
      f <- f - vecn1 %*% t(apply(f, 2, median))
      sigm <- sqrt(apply(f^2, 2, sum))
      f <- t(t(f)/sigm) #
# sorting by magnitude
      f <- f[, rev(sort.list(sigm))]
      sigm <- rev(sort(sigm))
    }
    sigmnew <- sigm
    sigmf <- matrix(0, n, k)
    sigml <- matrix(0, p, k)
  }
# begin of the iteration
for(it in 1:iter) {
  cat("Iteration ", it, "\n") #
# regression of the columns
  for(j in 1:p) {
    if(method == "rreg") {
      sigmlreg <- switch(model,
        a = get(method)(vecn1, (x[, j] - mu - a),
          int= F),
        m = get(method)(f, x[, j], int = F),
        b = get(method)(f, (x[, j] - mu - a)))
      weightc1 <- sigmlreg$w
      sigmlreg <- sigmlreg$coef
    }
    if(method == "wllfit") {
      if(j == 1) {
        if(model == "a")
          weightc1 <- vecn1
        else weightc1 <- weight.wll(f, n, k)
      }
      sigmlreg <- switch(model,
        a = llfit(vecn1, (x[, j] - mu - a), int =
          F)$coef,

```

```

        m = llfit(weightc1 * f, weightc1 * (x[,
            j]), intercept = F)$coef,
        b = llfit(weightc1 * cbind(vecn1, f),
            weightc1 * (x[, j] - mu - a),
            intercept = F)$coef)
    }
    else {
        sigmlreg <- switch(model,
            a = get(method)(vecn1, (x[, j] - mu - a),
                int= F)$coef,
            m = get(method)(f, x[, j], int = F)$coef,
            b = get(method)(f, (x[, j] - mu - a))$coef)
    }
    if((method == "wllfit") || (method == "rreg"))
        weightc[, j] <- weightc1
    if(model == "a")
        b[j] <- sigmlreg
    if(model == "m")
        sigml[j, ] <- t(sigmlreg[1:k])
    if(model == "b") {
        b[j] <- sigmlreg[1]
        sigml[j, ] <- t(sigmlreg[2:(k + 1)])
    }
}
# update the parameter estimates according to the model
restrictions
if(model == "a") {
    mb <- median(b)
    b <- b - mb
    ms <- mu + mb
}
if(model == "m") {
    if(k == 1) {
        sigm <- sqrt(sum(sigml^2))
        l <- sigml/sigm
        sigm <- as.matrix(sigm)
    }
    else {
        sigm <- sqrt(apply(sigml^2, 2, sum))
        l <- t(t(sigml)/sigm)
        l <- l[, rev(sort.list(sigm))]
        sigm <- rev(sort(sigm))
    }
}
if(model == "b") {
    if(k == 1) {
        hl <- median(sigml)
    }
}

```

```

    l <- sigm1 - h1
    sigm <- sqrt(sum(l^2))
    l <- l/sigm
    a <- as.vector(a + f * h1)
  }
  else {
    h1 <- apply(sigm1, 2, median)
    l <- sigm1 - vecp1 %*% t(h1)
    sigm <- sqrt(apply(l^2, 2, sum))
    l <- t(t(l)/sigm)
    l <- l[, rev(sort.list(sigm))]
    sigm <- rev(sort(sigm))
    a <- as.vector(a + f %*% h1)
  }
  ma <- median(a)
  mb <- median(b)
  a <- a - ma
  b <- b - mb
  ms <- mu + ma + mb
}
# regression of the rows
for(i in 1:n) {
  if(method == "rreg") {
    sigmfreg <- switch(model,
      a = get(method)(vecp1, (x[i, ] - ms - b),
        int = F),
      m = get(method)(l, x[i, ], int = F),
      b = get(method)(l, (x[i, ] - ms - b)))
    weightr1 <- sigmfreg$w
    sigmfreg <- sigmfreg$coef
  }
  if(method == "wllfit") {
    if(i == 1) {
      if(model == "a")
        weightr1 <- vecp1
      else weightr1 <- weight.wll(l, p, k)
    }
    sigmfreg <- switch(model,
      a = llfit(vecp1, (x[i, ] - ms - b), int =
        F)$coef,
      m = llfit(weightr1 * l, weightr1 * (x[i,
        ]), intercept = F)$coef,
      b = llfit(weightr1 * cbind(vecp1, l),
        weightr1 * (x[i, ] - ms - b),
        intercept = F)$coef)
  }
  else {

```

```

    sigmfreg <- switch(model,
      a = get(method)(vecpl, (x[i, ] - ms - b),
        int = F)$coef,
      m = get(method)(l, x[i, ], int = F)$coef,
      b = get(method)(l, (x[i, ] - ms - b))$coef)
  }
  if((method == "wllfit") || (method == "rreg"))
    weightr[i, ] <- weightr1
  if(model == "a")
    a[i] <- sigmfreg
  if(model == "m")
    sigmf[i, ] <- t(sigmfreg[1:k])
  if(model == "b") {
    a[i] <- sigmfreg[1]
    sigmf[i, ] <- t(sigmfreg[2:(k + 1)])
  }
}
# update the parameter estimates according to the model
restrictions
  if(model == "a") {
    ma <- median(a)
    a <- a - ma
    mu <- ms + ma
  }
  if(model == "m") {
    if(k == 1) {
      sigmnew <- sqrt(sum(sigmf^2))
      f <- sigmf/sigmnew
      sigmnew <- as.matrix(sigmnew)
    }
    else {
      f <- sigmf
      sigmnew <- sqrt(apply(sigmf^2, 2, sum))
      f <- t(t(sigmf)/sigmnew)
      f <- f[, rev(sort.list(sigmnew))]
      sigmnew <- rev(sort(sigmnew))
    }
  }
}
if(model == "b") {
  if(k == 1) {
    hf <- median(sigmf)
    f <- sigmf - hf
    sigmnew <- sqrt(sum(f^2))
    f <- f/sigmnew
    sigmnew <- as.matrix(sigmnew)
    b <- as.vector(b + l * hf)
  }
}

```

```

else {
  hf <- apply(sigmf, 2, median)
  f <- sigmf - vecn1 %*% t(hf)
  sigmnew <- sqrt(apply(f^2, 2, sum))
  f <- t(t(f)/sigmnew)
  f <- f[, rev(sort.list(sigmnew))]
  sigmnew <- rev(sort(sigmnew))
  b <- as.vector(b + 1 %*% hf)
}
ma <- median(a)
mb <- median(b)
a <- a - ma
b <- b - mb
mu <- ms + ma + mb
}
# calculate the residuals
if(method == "wllfit") {
  resid <- switch(model,
    a = x - mu - a %*% t(vecp1) - vecn1 %*% t(b),
    m = x - f %*% diag(sigmnew) %*% t(1),
    b = x - mu - a %*% t(vecp1) - vecn1 %*% t(b)
      - f %*% diag(sigmnew) %*% t(1))
}
else {
  resid <- switch(model,
    a = x - mu - a %*% t(vecp1) - vecn1 %*% t(b),
    m = x - f %*% diag(sigmnew) %*% t(1),
    b = x - mu - a %*% t(vecp1) - vecn1 %*% t(b)
      - f %*% diag(sigmnew) %*% t(1))
}
if(method == "ltsreg") {
  help <- vecp0
  halfn <- floor((n + 1)/2)
  for(j in 1:p) {
    help[j] <- sum(sort(resid[, j]^2)[1:halfn])
  }
}
if(method == "lmsreg") {
  help <- vecp0
  halfn <- floor((n + 1)/2)
  for(j in 1:p) {
    help[j] <- sum(sort(abs(resid[, j]))[1:halfn])
  }
}
objnew <- switch(method,
  lsfit = sum(resid^2),
  llfit = sum(abs(resid)),

```



```

    wllfit = sum(abs(resid)),
    lmsreg = sum(sort(help)[1:floor((p + 1)/2)]),
    ltsreg = sum(sort(help)[1:floor((p + 1)/2)]),
    rreg = sum(weightc * weightr * resid^2)
  cat("Convergence criterion = ", objnew, "\n")
  if(model != "m") {
    musav <- mu
    asav <- as.vector(a)

    if(length(c("L1", "L2", "L3", "L4", "L5", "L6", "L7", "L8",
    "L9", "L10")))
      names(asav) <- dimnames(x)[[1]] <-
c("L1", "L2", "L3", "L4", "L5", "L6", "L7", "L8", "L9", "L10"
)
    bsav <- as.vector(b)
    names(bsav) <- dimnames(x)[[2]]
  }
  if(model != "a") {
    sigmasav <- sigmnew
    fnames <- paste("Fact.", 1:k)
    lsav <- l
    dimnames(lsav) <- list(dimnames(x)[[2]], fnames)
    fsav <- f

    if(length(c("L1", "L2", "L3", "L4", "L5", "L6", "L7", "L8",
    "L9", "L10")))
      dimnames(fsav) <- list(dimnames(x)[[1]] <-
c("L1", "L2", "L3", "L4", "L5", "L6", "L7", "L8", "L9", "L10"
), fnames)
    else dimnames(fsav) <- list(NULL, fnames)
  }
  ressav <- resid
  objsav <- objnew
  itersav <- it
  if((abs((objold - objnew)/max(objnew, 1)) < tol)
  || (objnew <
    tol))
    break
  objold <- objnew
  residu <- sum(ressav)^2
  MSE <- residu/(n*p)
  residu <- svd(residu)
  eigen <- residu$d
}
if(model == "a")
  if(transp == T)

```

```

    ans <- list(mu = musav, a = bsav, b = asav,
residuals
    = t(ressav), objective = objsav, iterations
    = itersav, model = "additive", method =
    method, call = call)
  else ans <- list(mu = musav, a = asav, b = bsav,
residuals
    = ressav, objective = objsav, iterations =
    itersav, model = "additive", method =
    method, call = call)
if(model == "m")
  if(transp == T)
    ans <- list(sigma = sigmasav, loadings = fsav,
    scores= lsav, residuals = t(ressav),
    objective = objsav, iterations =
    itersav, model = "multiplicative",
    method = method, call = call)
  else ans <- list(sigma = sigmasav, loadings =
    lsav, scores = fsav, residuals =
    ressav, objective = objsav,
    iterations = itersav, model =
    "multiplicative", method = method,
    call = call)
if(model == "b") {
  if(transp == T)
    ans <- list(mu = musav,weightrow = weightc1,
    weightcolumn = weightr1, a = bsav, b
    = asav, sigma = sigmasav, loadings =
    fsav, scores = lsav, residuals =
    t(ressav), objective = objsav,
    iterations=itersav,MSE=MSE,eigen=eig
    en, model = "add+mult",method =
    method, call = call,)
  else ans <- list(mu = musav,weightrow = weightc1,
    weightcolumn = weightr1, a = asav,
    b = bsav, sigma = sigmasav,
    loadings = lsav, scores = fsav,
    residuals = ressav, objective =
    objsav,iterations=itersav,MSE=MSE,
    eigen=eigen, model = "add+mult"
    ,method = method, call = call)
}
# if a plots are desired
if((drawplot == T) && (model != "a") && (k == 2)) {
  if(model == "m") {
    par(mfrow = c(2, 1))
  }
}

```

```
else {
  par(mfrow = c(2, 2))
}
if((alpha < 0) || (alpha > 1)) {
  warning("alpha for biplot has to be in the
interval [0,1] --> it is set to 1"
)
  alpha <- 1
}
if(transp == T) {
  h <- lsav
  lsav <- fsav
  fsav <- h
}
if(orth == T) {
# orthogonalize f and l for the biplot
residu<- svd(resid)
f <- residu$u[,c(1,2)]
dimnames(f) <- dimnames(fsav)
l <- residu$v[,c(1,2)]
dimnames(l) <- dimnames(lsav)
biplot.default(f,l)
}
else biplot.default(fsav %*% diag(sigmasav^(1 -
alpha)), lsav %*%
  diag(sigmasav^alpha))
title("Biplot")
persp(ressav)
title("Residuals")
if(model == "b") {
  if(transp == T)
    boxplot(bsav)
  else boxplot(asav)
  title("Row Effects")
  if(transp == T)
    boxplot(asav)
  else boxplot(bsav)
  title("Column Effects")
}
}
if((drawplot == T) && (model == "a")) {
  par(mfrow = c(2, 2))
  if(transp == T)
    boxplot(bsav)
  else boxplot(asav)
  title("Row Effects")
  if(transp == T)
```

```
        boxplot(asav)
    else boxplot(bsav)
    title("Column Effects")
    persp(ressav)
    title("Residuals")
}
ans
}

twoway.rob(data)
Iteration    1
Convergence criterion = 103.046606919109
Iteration    2
Convergence criterion = 54.0855096376769
Iteration    3
Convergence criterion = 99.2433070059973
Iteration    4
Convergence criterion = 35.7025132027403
Iteration    5
Convergence criterion = 42.1110003396302
Iteration    6
Convergence criterion = 45.2983425737412
Iteration    7
Convergence criterion = 27.8783657620036
Iteration    8
Convergence criterion = 103.521439396558
Iteration    9
Convergence criterion = 42.6823181428136
Iteration   10
Convergence criterion = 100.587216044605
Iteration   11
Convergence criterion = 63.4376729254526
Iteration   12
Convergence criterion = 48.9962729695652
Iteration   13
Convergence criterion = 93.9573587816945
Iteration   14
Convergence criterion = 39.1055013169462
Iteration   15
Convergence criterion = 35.3275281354869
Iteration   16
Convergence criterion = 34.8597476338574
Iteration   17
Convergence criterion = 34.8385208591557
Iteration   18
Convergence criterion = 34.8385207467343
$mu:
```

[1] 7.886786

\$weightrow:

1	2	3	4				5	6	7	8	9	10
1	1	1	1	0.2053153	1	1	1	1	1	1	1	1

\$weightcolumn:

1			2	3	4	5	6	7	8
1	0.2009052	1	1	1	1	1	1	1	1

\$a:

	L1		L2		L3		L4		L5
-0.201875	0.2638181	-0.06605	0.02685046	0.5723587					
	L6		L7		L8		L9		L10
-0.299476	0.701929	-0.3160419	0.3655727	-0.02685046					

\$b:

	G1		G2		G3		G4		G5
0.5662544	1.813726	-0.8669274	0.1963929	-1.571666					
	G6		G7		G8				
-0.4488213	0.3589423	-0.1963929							

\$sigma:

[1] 8.296959 5.423469

\$loadings:

	Fact. 1		Fact. 2
G1	0.0668010	-0.442588435	
G2	-0.6995618	0.713806110	
G3	-0.1423471	0.009140539	
G4	-0.6416718	-0.311072822	
G5	0.2060897	-0.009140539	
G6	0.1239499	0.050419974	
G7	-0.0668010	-0.433866621	
G8	0.1088420	0.082948759	

\$scores:

	Fact. 1		Fact. 2
L1	-0.01906740	-0.008604054	
L2	-0.41184270	0.058848847	
L3	-0.24385039	0.299747878	
L4	0.01906740	0.008604054	
L5	-0.62138624	-0.693393417	
L6	0.20268755	0.313522889	
L7	0.09548249	-0.188628449	
L8	0.15804836	-0.095015906	
L9	0.52034808	-0.119461203	

L10 -0.19527936 0.518236661

\$residuals:

	G1	G2	G3
L1	-2.569171e-001	5.565839e-010	-1.050764e-001
L2	6.703484e+000	1.133739e+000	1.028161e-007
L3	-7.186106e-008	-3.435818e-001	9.550803e-009
L4	1.303861e+000	8.100389e-010	4.150511e-001
L5	-1.342729e+000	-4.628538e+000	-6.180633e-001
L6	-4.880770e-007	6.452031e-008	-4.387400e-002
L7	1.753049e-008	-1.196384e-007	-4.944905e-009
L8	-1.753002e+000	-1.170952e+000	-1.364435e-001
L9	4.118998e+000	1.522941e+000	1.663802e-001
L10	-1.020775e-007	-4.907469e-007	-4.215139e-008

	G4	G5	G6
L1	3.053896e-009	2.979323e-001	-5.279475e-002
L2	1.977976e-007	-1.064704e-001	-4.015680e-008
L3	-3.618075e-008	3.234208e-001	-1.087706e-001
L4	-1.605111e-009	-1.059814e+000	7.622315e-002
L5	2.839253e-007	-9.913512e-008	3.429909e-001
L6	-4.545279e-009	8.986553e-003	-2.372107e-008
L7	-8.296591e-008	-2.163576e-008	-7.450777e-002
L8	-3.246245e-008	8.142951e-001	8.221885e-001
L9	2.594512e-007	-4.700024e-003	5.505300e-009
L10	-2.493853e-008	2.855233e-008	1.146915e-007

	G7	G8
L1	-4.014358e-010	2.285718e-001
L2	4.229471e-009	-4.074345e-001
L3	-1.285056e-001	1.370206e-001
L4	-3.634317e-001	-2.950065e-009
L5	3.038408e-008	9.646248e-008
L6	8.591584e-001	-5.043322e-009
L7	-7.475909e-001	5.046162e-001
L8	2.718741e-009	1.069007e-008
L9	-1.726525e-008	-1.605461e+000
L10	-1.820283e-007	1.417286e-008

\$kekonvergenan:

[1] 34.83852

\$iterations:

[1] 18

\$MSE:

[1] 0.2786216

\$eigenvalue:

```
[1] 8.929931e+000 4.182810e+000 1.612913e+000
     1.236979e+000 1.076269e+000 6.623334e-001
[7] 1.814873e-001 5.623842e-008
```

\$model:

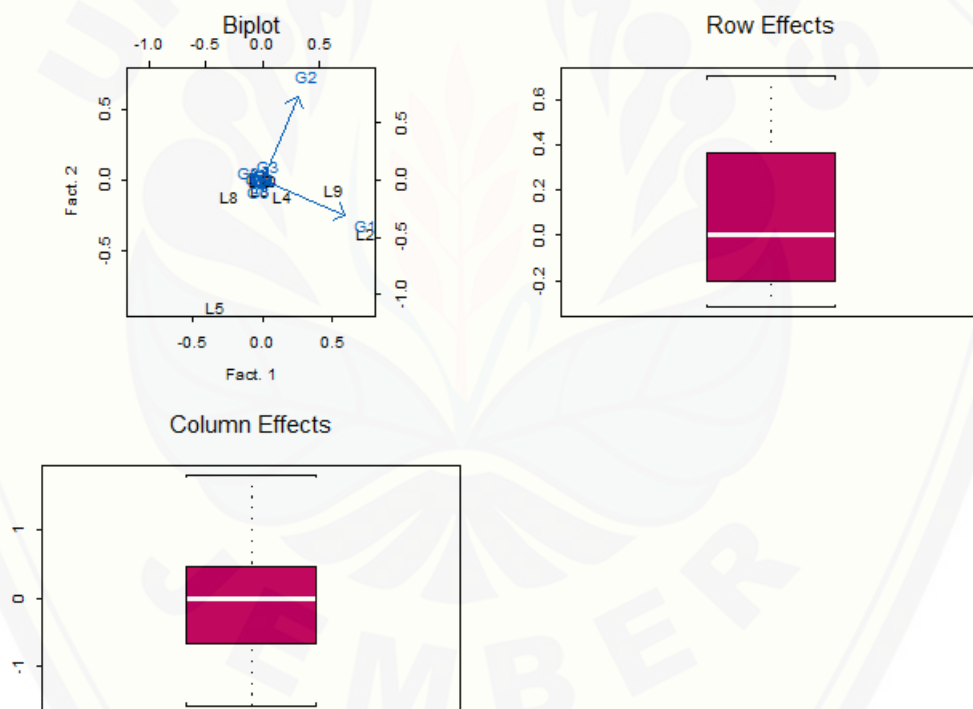
```
[1] "add+mult"
```

\$method:

```
[1] "wllfit"
```

\$call:

```
twoway.rob(x = data)
```



C. Script Data Perbenihan Padi pada Program R

C.1 Script Weight.wl1

###Menghitung Nilai Bobot pada Data###

```
weight.wl1<-function(x, n, k, print.it = F)
{
  x <- as.data.frame(x)
  if(n == 1) {
```

```

    rob <- list(center = 0, cov = 1)
      rob$center <- median(x)
      rob$cov <- mad(x)^2
    }
    else rob <- cov.rob(x)
  x <- as.matrix(x)
  robdist2 <- mahalnobis(x, rob$center, rob$cov)
  weight <- qchisq(0.95, k)/robdist2
  weight <- apply(cbind(1, weight), 1, min)
  weight
}

```

C.2 Script PcaProj dan Keluarannya

###Menghitung Nilai Robust Principal Component Analysis###

```

Pca<-PcaProj(x, k = 2, kmax = ncol(x), scale=TRUE, trace=FALSE)
> f<-attr(Pca, "scores")
> f

```

	PC1	PC2
[1,]	0.2059037	0.8063600
[2,]	1.9358075	-2.3737441
[3,]	1.4739291	0.3059358
[4,]	0.8282724	-0.7746220
[5,]	-0.5628555	-2.6697819
[6,]	-0.4045116	1.4557900
[7,]	-1.5215006	-1.5138178
[8,]	-2.2403286	1.1791935
[9,]	-2.5037144	-0.6391787
[10,]	1.5546466	0.4487833

C.3 Script Twoway.rob dan Keluarannya

Robust Faktor pada Tabel Dua Arah###

```

twoway.rob<-function(x, model = "b", k = 2, method = "wl1fit",
drawplot = T, alpha = 1, tol= 0.0001, iter = 300, orth = T)
{
  n <- dim(x)[1]
  p <- dim(x)[2]
  if(!length(dimnames(x)[[2]]))
dimnames(x)[[1]]<c("L1", "L2", "L3", "L4", "L5", "L6", "L7", "L8", "L9", "
L10")
dimnames(x) <- list(dimnames(x)[[1]], paste("G", 1:p, sep = ""))
x<-as.matrix(x)
  if(p > n) {
    # take the transposed matrix
    x <- t(x)
    transp <- p
    p <- n
    n <- transp
    transp <- T
  }
  else transp <- F
  vecn0 <- rep(0, n)
  vecn1 <- rep(1, n)
  vecp0 <- rep(0, p)
  vecp1 <- rep(1, p) #
  # weights for Weighted L1 or Rob. Regression
  if((method == "wl1fit") || (method == "rreg")) {
    # initialize matrices for row and column weights
    weightr <- matrix(0, n, p)
    weightc <- weightr
  }
}

```

```

}
# initialize the objective function
objold <- 100000000000 #
# choice of the regression method
imethod <- charmatch(method, c("lsfit", "l1fit", "w11fit",
"lmsreg", "ltsreg", "rreg"), nomatch = NA)
if(!is.na(imethod))
  method <- c("lsfit", "l1fit", "w11fit", "lmsreg", "ltsreg",
              "rreg")[imethod]
if(method == "w11fit") {
  if((n <= 2 * k) | (p <= 2 * k)) {
    stop("\n\nThe number of rows and columns should at least be >
2*k for W11fit !\n\n")
  }
}
}
# starting values for iteration
if(model != "m") {
  a <- apply(x, 1, median)
  mu <- median(a)
  a <- a - mu
  b <- vecp0
}
if(model != "a") {
  # starting with robust PCA
  Pca<-PcaProj(x, k = 2, kmax = ncol(x), scale=TRUE, trace=FALSE)
  f<-attr(Pca, "scores")
  # starting with classical PCA
  # f <- princomp(x)$scores[, 1:k]
  # starting with random values
  # f <- matrix(rnorm(n * k, 0, 1), ncol = k)
  if(k == 1) {
    # k is the dimension of the multiplicative terms
    f <- f - median(f)
    sigm <- sqrt(sum(f^2))
    f <- f/sigm
  }
  else {
    f <- f - vecn1 %*% t(apply(f, 2, median))
    sigm <- sqrt(apply(f^2, 2, sum))
    f <- t(t(f)/sigm) #
    # sorting by magnitude
    f <- f[, rev(sort.list(sigm))]
    sigm <- rev(sort(sigm))
  }
  sigmnew <- sigm
  sigmf <- matrix(0, n, k)
  sigml <- matrix(0, p, k)
}
# begin of the iteration
for(it in 1:iter) {
  cat("Iteration ", it, "\n")
  # regression of the columns
  for(j in 1:p) {
    if(method == "rreg") {
      sigmlreg <- switch(model,
                        a = get(method)(vecn1, (x[, j] - mu -
                                                a), int= F),
                        m = get(method)(f, x[, j], int = F),
                        b = get(method)(f, (x[, j] - mu - a)))
      weightc1 <- sigmlreg$w
      sigmlreg <- sigmlreg$coef
    }
    if(method == "w11fit") {

```

```

if(j == 1) {
  if(model == "a")
    weightc1 <- vecn1
  else weightc1 <- weight.w11(f, n, k)
}
sigmlreg <- switch(model,
  a = L1linreg(vecn1, (x[, j] - mu - a),
               p = 1, tol = 0.001,
               maxiter = 50)$x,
  m = L1linreg(weightc1 * f, weightc1 *
               (x[, j]), p = 1, tol =
               0.001, maxiter = 50)$x,
  b = L1linreg(weightc1 * cbind(vecn1,
                                 f), weightc1 * (x[, j] -
                                 mu - a), p = 1, tol =
                                 0.001, maxiter = 50)$x)
}
else {
  sigmlreg <- switch(model,
    a = get(method)(vecn1, (x[, j] - mu -
                           a), int=F)$coef,
    m = get(method)(f, x[, j], int =
                   F)$coef,
    b = get(method)(f, (x[, j] - mu -
                       a))$coef)
  }
if((method == "w11fit") || (method == "rreg"))
  weightc[, j] <- weightc1
if(model == "a")
  b[j] <- sigmlreg
if(model == "m")
  sigml[j, ] <- t(sigmlreg[1:k])
if(model == "b") {
  b[j] <- sigmlreg[1]
  sigml[j, ] <- t(sigmlreg[2:(k + 1)])
}
}
# update the parameter estimates according to the model
restrictions
if(model == "a") {
  mb <- median(b)
  b <- b - mb
  ms <- mu + mb
}
if(model == "m") {
  if(k == 1) {
    sigm <- sqrt(sum(sigml^2))
    l <- sigml/sigm
    sigm <- as.matrix(sigm)
  }
  else {
    sigm <- sqrt(apply(sigml^2, 2, sum))
    l <- t(t(sigml)/sigm)
    l <- l[, rev(sort.list(sigm))]
    sigm <- rev(sort(sigm))
  }
}
if(model == "b") {
  if(k == 1) {
    h1 <- median(sigml)
    l <- sigml - h1
    sigm <- sqrt(sum(l^2))
    l <- l/sigm
    a <- as.vector(a + f * h1)
  }
}

```

```

}
else {
  h1 <- apply(sigm1, 2, median)
  l <- sigm1 - vecp1 %>% t(h1)
  sigm <- sqrt(apply(l^2, 2, sum))
  l <- t(t(l)/sigm)
  l <- l[, rev(sort.list(sigm))]
  sigm <- rev(sort(sigm))
  a <- as.vector(a + f %>% h1)
}
ma <- median(a)
mb <- median(b)
a <- a - ma
b <- b - mb
ms <- mu + ma + mb
}
#
# regression of the rows
#
for(i in 1:n) {
  if(method == "rreg") {
    sigmfreg <- switch(model,
      a = get(method)(vecp1, (x[i, ] - ms -
                             b), int = F),
      m = get(method)(l, x[i, ], int = F),
      b = get(method)(l, (x[i, ] - ms -
                             b)))

    weightr1 <- sigmfreg$w
    sigmfreg <- sigmfreg$coef
  }
  if(method == "wllfit") {
    if(i == 1) {
      if(model == "a")
        weightr1 <- vecp1
      else weightr1 <- weight.wll(1, p, k)
    }
    sigmfreg <- switch(model,
      a = L1linreg(vecp1, (x[i, ] - ms -
                           b), p = 1, tol = 0.001,
                           maxiter = 50)$x,
      m = L1linreg(weightr1 * l, weightr1 *
                   (x[i, ]), p = 1, tol =
                   0.001, maxiter = 50)$x,
      b = L1linreg(weightr1 * cbind(vecp1,
                                     l), weightr1 * (x[i, ] -
                                                         ms -
                                                         b), p = 1, tol =
                                                         0.001, maxiter = 50)$x)
  }
  else {
    sigmfreg <- switch(model,
      a = get(method)(vecp1, (x[i, ] - ms -
                             b), int = F)$coef,
      m = get(method)(l, x[i, ], int =
                       F)$coef,
      b = get(method)(l, (x[i, ] - ms -
                             b))$coef)
  }
  if((method == "wllfit") || (method == "rreg"))
    weightr[i, ] <- weightr1
  if(model == "a")
    a[i] <- sigmfreg
  if(model == "m")
    sigmf[i, ] <- t(sigmfreg[1:k])
  if(model == "b") {

```

```

    a[i] <- sigmfreg[1]
    sigmf[i, ] <- t(sigmfreg[2:(k + 1)])
  }
}
# update the parameter estimates according to the model
restrictions
if(model == "a") {
  ma <- median(a)
  a <- a - ma
  mu <- ms + ma
}
if(model == "m") {
  if(k == 1) {
    sigmnew <- sqrt(sum(sigmf^2))
    f <- sigmf/sigmnew
    sigmnew <- as.matrix(sigmnew)
  }
  else {
    f <- sigmf
    sigmnew <- sqrt(apply(sigmf^2, 2, sum))
    f <- t(t(sigmf)/sigmnew)
    f <- f[, rev(sort.list(sigmnew))]
    sigmnew <- rev(sort(sigmnew))
  }
}
if(model == "b") {
  if(k == 1) {
    hf <- median(sigmf)
    f <- sigmf - hf
    sigmnew <- sqrt(sum(f^2))
    f <- f/sigmnew
    sigmnew <- as.matrix(sigmnew)
    b <- as.vector(b + 1 * hf)
  }
  else {
    hf <- apply(sigmf, 2, median)
    f <- sigmf - vecn1 %*% t(hf)
    sigmnew <- sqrt(apply(f^2, 2, sum))
    f <- t(t(f)/sigmnew)
    f <- f[, rev(sort.list(sigmnew))]
    sigmnew <- rev(sort(sigmnew))
    b <- as.vector(b + 1 %*% hf)
  }
}
ma <- median(a)
mb <- median(b)
a <- a - ma
b <- b - mb
mu <- ms + ma + mb
}
# calculate the residuals
if(method == "wllfit") {
  resid <- switch(model,
    a = x - mu - a %*% t(vecp1) - vecn1 %*%
      t(b),
    m = x - f %*% diag(sigmnew) %*% t(1),
    b = x - mu - a %*% t(vecp1) - vecn1 %*%
      t(b) - f %*% diag(sigmnew) %*% t(1))
}
else {
  resid <- switch(model,
    a = x - mu - a %*% t(vecp1) - vecn1 %*%
      t(b),
    m = x - f %*% diag(sigmnew) %*% t(1),

```



```

        b = x - mu - a %*% t(vecp1) - vecn1 %*%
            t(b) - f %*% diag(sigmnew) %*% t(1))
    }
    if(method == "ltsreg") {
        help <- vecp0
        halfn <- floor((n + 1)/2)
        for(j in 1:p) {
            help[j] <- sum(sort(resid[, j]^2)[1:halfn])
        }
    }
    if(method == "lmsreg") {
        help <- vecp0
        halfn <- floor((n + 1)/2)
        for(j in 1:p) {
            help[j] <- sum(sort(abs(resid[, j]))[1:halfn])
        }
    }
    objnew <- switch(method,
        lsfit = sum(resid^2),
        llfit = sum(abs(resid)),
        wllfit = sum(abs(resid)),
        lmsreg = sum(sort(help)[1:floor((p +
            1)/2)]),
        ltsreg = sum(sort(help)[1:floor((p +
            1)/2)]),
        rreg = sum(weightc * weightr * resid^2))
    cat("Convergence criterion = ", objnew, "\n")
    if(model != "m") {
        musav <- mu
        asav <- as.vector(a)

        if(length(c("L1", "L2", "L3", "L4", "L5", "L6", "L7", "L8", "L9", "L10")))
            names(asav) <- dimnames(x)[[1]] <-
        c("L1", "L2", "L3", "L4", "L5", "L6", "L7", "L8", "L9", "L10")
        bsav <- as.vector(b)
        names(bsav) <- dimnames(x)[[2]]
    }
    if(model != "a") {
        sigmasav <- sigmnew
        fnames <- paste("Fact.", 1:k)
        lsav <- l
        dimnames(lsav) <- list(dimnames(x)[[2]], fnames)
        fsav <- f

        if(length(c("L1", "L2", "L3", "L4", "L5", "L6", "L7", "L8", "L9", "L10")))
            dimnames(fsav) <- list(dimnames(x)[[1]] <-
        c("L1", "L2", "L3", "L4", "L5", "L6", "L7", "L8", "L9", "L10"), fnames
        )
        else dimnames(fsav) <- list(NULL, fnames)
    }
    ressav <- resid
    objsav <- objnew
    itersav <- it
    if((abs((objold - objnew)/max(objnew, 1)) < tol) || (objnew <
        tol))
        break
    objold <- objnew
    residu <- sum(ressav)^2
    MSE <- residu/(n*p)
    residu <- svd(resid)
    eigen <- residu$d
}
if(model == "a")
    if(transp == T)

```

```

ans <- list(mu = musav, a = bsav, b = asav, residuals
           = t(ressav), objective = objsav, iterations
           = itersav, model = "additive", method =
           method, call = call)
else ans <- list(mu = musav, a = asav, b = bsav, residuals =
               ressav, objective = objsav, iterations =
               itersav, model = "additive", method =
               method, call = call)
if(model == "m")
  if(transp == T)
    ans <- list(sigma = sigmasav, loadings = fsav, scores
               = lsav, residuals = t(ressav), objective =
               objsav, iterations = itersav, model =
               "multiplicative", method = method, call =
               call)
  else ans <- list(sigma = sigmasav, loadings = lsav, scores =
                 fsav, residuals = ressav, objective =
                 objsav, iterations = itersav, model =
                 "multiplicative", method = method, call = call)
if(model == "b") {
  if(transp == T)
    ans <- list(mu = musav, weightrow = weightc1,
               weightcolumn = weightr1, a = bsav, b = asav,
               sigma = sigmasav, loadings = fsav, scores =
               lsav, residuals = t(ressav), kekonvergenan =
               objsav, iterations = itersav, MSE = MSE, eigenvalue =
               eigen, model = "add+mult", method = method, call
               = call,)
    else ans <- list(mu = musav, weightrow = weightc1,
                   weightcolumn = weightr1, a = asav, b = bsav,
                   sigma = sigmasav, loadings = lsav, scores =
                   fsav, residuals = ressav, kekonvergenan =
                   objsav, iterations = itersav, MSE = MSE, eigenvalue
                   = eigen, model = "add+mult", method = method,
                   call = call)
}
# if a plots are desired
if((drawplot == T) && (model != "a") && (k == 2)) {
  if(model == "m") {
    par(mfrow = c(2, 1))
  }
  else {
    par(mfrow = c(2, 2))
  }
}
if((alpha < 0) || (alpha > 1)) {
  warning("alpha for biplot has to be in the interval [0,1] -
-> it is set to 1"
)
  alpha <- 1
}
if(transp == T) {
  h <- lsav
  lsav <- fsav
  fsav <- h
}
if(orth == T) {
  # orthogonalize f and l for the biplot
  residu <- svd(resid)
  f <- residu$u[,c(1,2)]
  dimnames(f) <- dimnames(fsav)
  l <- residu$v[,c(1,2)]
  dimnames(l) <- dimnames(lsav)
  biplot(f,l)
}

```

```
else biplot(fsav %%% diag(sigmasav^(1 - alpha)), lsav %%%
            diag(sigmasav^alpha))
title("Biplot")
if(model == "b") {
  if(transp == T)
    boxplot(bsav)
  else boxplot(asav)
  title("Row Effects")
  if(transp == T)
    boxplot(asav)
  else boxplot(bsav)
  title("Column Effects")
}
}
if((drawplot == T) && (model == "a")) {
  par(mfrow = c(2, 2))
  if(transp == T)
    boxplot(bsav)
  else boxplot(asav)
  title("Row Effects")
  if(transp == T)
    boxplot(asav)
  else boxplot(bsav)
  title("Column Effects")
}
}
ans
}
```

```
> twoway.rob(data)
```

```
Iteration 1
Convergence criterion = 43.82412
Iteration 2
Convergence criterion = 124.7225
Iteration 3
Convergence criterion = 111.8411
Iteration 4
Convergence criterion = 61.59557
Iteration 5
Convergence criterion = 41.68665
Iteration 6
Convergence criterion = 43.49681
Iteration 7
Convergence criterion = 101.9413
Iteration 8
Convergence criterion = 44.43805
Iteration 9
Convergence criterion = 91.55413
Iteration 10
Convergence criterion = 91.96248
Iteration 11
Convergence criterion = 40.81714
Iteration 12
Convergence criterion = 56.00406
Iteration 13
Convergence criterion = 119.4293
Iteration 14
Convergence criterion = 47.61318
Iteration 15
Convergence criterion = 79.67778
Iteration 16
Convergence criterion = 36.7863
Iteration 17
Convergence criterion = 90.53298
```

Iteration	18
Convergence criterion =	93.32637
Iteration	19
Convergence criterion =	88.61117
Iteration	20
Convergence criterion =	40.21362
Iteration	21
Convergence criterion =	37.25816
Iteration	22
Convergence criterion =	46.67606
Iteration	23
Convergence criterion =	140.3203
Iteration	24
Convergence criterion =	75.72437
Iteration	25
Convergence criterion =	92.15777
Iteration	26
Convergence criterion =	61.1587
Iteration	27
Convergence criterion =	39.06447
Iteration	28
Convergence criterion =	85.16064
Iteration	29
Convergence criterion =	86.94519
Iteration	30
Convergence criterion =	46.70584
Iteration	31
Convergence criterion =	101.1321
Iteration	32
Convergence criterion =	44.82349
Iteration	33
Convergence criterion =	131.5157
Iteration	34
Convergence criterion =	58.14655
Iteration	35
Convergence criterion =	50.6265
Iteration	36
Convergence criterion =	47.35395
Iteration	37
Convergence criterion =	47.48925
Iteration	38
Convergence criterion =	53.07586
Iteration	39
Convergence criterion =	107.5134
Iteration	40
Convergence criterion =	73.6206
Iteration	41
Convergence criterion =	44.27044
Iteration	42
Convergence criterion =	83.59163
Iteration	43
Convergence criterion =	39.43074
Iteration	44
Convergence criterion =	43.01964
Iteration	45
Convergence criterion =	40.65288
Iteration	46
Convergence criterion =	40.62158
Iteration	47
Convergence criterion =	40.64286
Iteration	48
Convergence criterion =	40.65731
Iteration	49
Convergence criterion =	40.66182

```

Iteration 50
Convergence criterion = 40.66478
$mu
[1] 7.830657

$weightrow
[1] 1.00000000 1.00000000 1.00000000 1.00000000 0.07975782
[6] 1.00000000 1.00000000 0.22697348 1.00000000 1.00000000

$weightcolumn
[1] 1.0000000 0.1233871 1.0000000 0.2932094 1.0000000 1.0000000
[7] 1.0000000 1.0000000

$a
      L1      L2      L3      L4      L5
-0.07145064 -0.09104237 -0.34734009 0.07145064 1.01958118
      L6      L7      L8      L9     L10
-0.18995869 0.78650420 0.44431576 0.67506835 -0.54172285

$b
      G1      G2      G3      G4      G5
-0.29913575 5.39644387 -0.09650455 2.01253298 -1.85454583
      G6      G7      G8
-0.41704456 0.15608935 0.09650455

$sigma
[1] 1825.381 1821.969

$loadings
      Fact. 1      Fact. 2
G1 -0.193223007 -1.951007e-01
G2 0.774550117 7.727797e-01
G3 0.197687891 1.971407e-01
G4 0.556468703 5.584401e-01
G5 -0.115171467 -1.152959e-01
G6 -0.027060700 -2.709835e-02
G7 -0.001076468 -4.797145e-05
G8 0.001076468 4.797145e-05

$scores
      Fact. 1      Fact. 2
L1 -0.08476824 0.08258002
L2 -0.04990277 0.05049657
L3 0.04990277 -0.05049657
L4 0.13708612 -0.13874532
L5 -0.82915237 0.83000292
L6 -0.08447481 0.08183045
L7 0.09763401 -0.10065102
L8 -0.47761439 0.47481713
L9 0.07022823 -0.07466974
L10 0.17841419 -0.17908627

$residuals
      G1      G2      G3      G4
L1 0.0005287487 -0.001205397 -3.736669e-05 0.3085138918
L2 7.9592364479 0.037071631 2.015634e-05 -0.0004610869
L3 -0.0007415412 -0.469906952 -2.756460e-04 0.0024635406
L4 1.2023072746 -2.116933506 8.838320e-04 -0.0003892479
L5 3.0438168447 -4.011632870 3.464091e-04 0.0016628057
L6 -0.5349135409 0.620138159 -5.174755e-04 -0.9300513301
L7 0.0005531060 -1.220688619 -5.218680e-04 1.2002842358
L8 -0.9514558698 -0.001675573 5.912037e-04 -1.1639902554
L9 3.5334083988 0.045581918 4.649858e-05 -0.0003044936

```


	G1	G2	G3	G4
L10	-0.6474335051	0.027212607	8.454879e-05	-0.0939410326
	G5	G6	G7	G8
L1	5.734174e-04	-0.2908790594	2.247079e-05	-3.621287e-04
L2	-3.298179e-03	-0.0001048276	6.099854e-01	-5.411426e-01
L3	3.954582e-01	-0.0000716429	-6.476269e-02	2.534922e-03
L4	-1.058974e+00	-0.0007269939	7.737851e-02	-5.175217e-01
L5	-1.095564e+00	-0.8862180614	2.310477e-01	-3.659655e-04
L6	-2.029218e-05	0.0764496588	-2.924060e-04	1.366348e-01
L7	-1.904827e-01	-0.2347554885	7.995467e-04	1.590107e-03
L8	8.772233e-04	0.0732921328	-1.262390e+00	-1.497076e-04
L9	-8.424967e-04	-0.0003012134	7.365486e-02	-1.867033e+00
L10	3.827668e-01	0.4511643162	-2.431740e-03	-3.050352e-05

\$kekonvergenan
[1] 40.66478

\$iterations
[1] 50

\$MSE
[1] 0.001381457

\$eigenvalue
[1] 9.6955376489 4.6883778729 2.0674557518 1.5915176512
[5] 0.9263556618 0.7186460967 0.5966158570 0.0007699031

\$model
[1] "add+mult"

\$method
[1] "w11fit"

\$call
function (name, ...) .Primitive("call")

