PENANGANAN PENCILAN PADA TABEL DUA ARAH DENGAN MENGGUNAKAN RCIM DAN *ROBUST* FAKTOR

TESIS

Oleh

Kurnia Ahadiyah
NIM 151820101015

JURUSAN MAGISTER MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2017

PENANGANAN PENCILAN PADA TABEL DUA ARAH DENGAN MENGGUNAKAN RCIM DAN *ROBUST* FAKTOR

TESIS

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Magister Matematika (S2)
dan mencapai gelar Magister Sains

Oleh :

Kurnia Ahadiyah
NIM 151820101015

JURUSAN MAGISTER MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2017

PERSEMBAHAN

Tesis ini saya persembahkan untuk:

1. Ayahanda Lutfillah dan Ibunda Arifatus Sholihah yang senantiasa memberi doa, semangat, motivasi, dan kasih sayang;
2. 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 tesis ini;
3. seluruh guru dan dosen sejak sekolah dasar hingga perguruan tinggi yang telah membimbing saya dan membagi ilmu dengan tulus;
4. Almamater Jurusan Matematika FMIPA Universitas Jember, SMA Negeri 1 Bangil, SMPN 1 Porong, MINU Kedung Cangkring, TK Nashrul Ummah;

MOTTO

"Sesungguhnya bersama kesukaran itu ada keringanan. Karena itu bila kau sudah selesai (mengerjakan yang lain). Dan berharaplah kepada Tuhanmu"
(Q.S Al Insyirah : 6-8)*)

*"*Urusan kita dalam kehidupan bukanlah untuk melampaui orang lain, tetapi untuk melampaui diri sendiri, untuk memecahkan rekor kita sendiri, dan untuk melampaui hari kemarin dengan hari ini*"*
*(*Stuart B. Johnson*)* **)

---

*) Departemen Agama Republik Indonesia. 2005. Alhikmah, Al-Qur'an dan terjemahannya. Bandung : CV Penerbit Diponegoro.
**) http://manubanat-kudus.sch.id/index.php/umum/195-motivasi-hidup-terbaru.

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Kurnia Ahadiyah

NIM : 151820101015

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul "Penanganan Pencilan pada Tabel Dua Arah dengan menggunakan RCIM dan *Robust* Faktor" adalah benar- benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan dalam institusi manapun dan juga 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, Juni 2017

Yang menyatakan,

Kurnia Ahadiyah

NIM 151820101015

TESIS

PENANGANAN PENCILAN PADA TABEL DUA ARAH DENGAN MENGGUNAKAN RCIM DAN ROBUST FAKTOR

Oleh

Kurnia Ahadiyah
NIM 151820101015

Pembimbing

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

PENGESAHAN

Tesis berjudul "Penanganan Pencilan pada Tabel Dua Arah dengan menggunakan RCIM (*Row Column Interaction Model*) dan *Robust* Faktor" telah diuji dan disahkan pada:

Hari, tanggal :

Tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember

Tim penguji:

Ketua,

Dr. Alfian Futuhul Hadi, S. Si, M. Si
NIP. 197407192000121001

Sekretaris,

Dian Anggraeni, S. Si, M. Si
NIP. 198202162006042002

Anggota I,

Prof. Drs. I Made Tirta M. Sc., Ph. D
NIP. 195912201985031002

Anggota II,

Dr. Mohamat Fatekurohman, S. Si., M. Si
NIP. 196906061998031001

Mengesahkan
Dekan,

Dr. Sujito, Ph. D.
NIP 196102041987111001

RINGKASAN

Penanganan Pencilan Pada Tabel Dua Arah Dengan Menggunakan RCIM dan *Robust* Faktor ; Kurnia Ahadiyah; 2017; 37 halaman; Jurusan Magister Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Outlier atau pencilan merupakan titik sampel yang memiliki karakteristik unik dan dapat diidentifikasi secara jelas karena berbeda dengan mayoritas titik sampel lainnya. Namun ada beberapa *Outlier* yang sulit teridentifikasi dikarenakan letak dan ukurannya pada suatu data. *Outlier* pada satu arah tertentu memungkinkan menjadi pengaruh dalam pengujian ketaknormalan data. Dalam hal ini diperlukan kajian sensitifitas pengaruh pencilan terhadap pengujian ketaknormalan.

Row Column Interaction Model (RCIM) yang merupakan perluasan dari konsep *Reduced-Rank Vector Generalized Models* (RR-VGLM) dimana prediktor linier yang pertama dimodelkan dengan penjumlahan dari pengaruh baris, pengaruh kolom, dan pengaruh interaksi yang mana pada pengaruh interaksi ditunjukkan seperti regresi *reduced-rank*. Data tabel dua arah berdistribusi normal yang mengandung pencilan (*outlier*) dapat juga dianalisis dengan menggunakan *robust* faktor melalui pendekatan robPCA untuk memeroleh hasil evaluasi penanganan *outlier* pada data tabel dua arah. RobPCA (*Robust* PCA) didasarkan pada metode PCA (*Principal Component Analysis*) yang mengatasi data dengan adanya pencilan (*outlier*).

Penelitian ini bertujuan untuk menunjukkan seberapa tahan model RCIM dan model *Robust* Faktor terhadap kehadiran *outlier* dalam data tabel dua arah. Ketahanan kedua model diukur dari jenis *outlier*, banyaknya *outlier* serta komposisi *outlier* pada data tabel dua arah. Dalam penelitian ini menggunakan data simulasi. Data simulasi dibentuk dalam data tabel dua arah yang berdistribusi normal dengan mengikuti model RCIM2 sehingga data awal yang terbentuk signifikan dengan Rank=2. Selanjutnya, dibangkitkan data *outlier* dengan tiga jenis *outlier* yaitu : *Pure Shift*

Outlier, Pure Point Mass Outlier, Shift Point Mass Outlier. Ketiga jenis *outlier* tersebut dimasukkan ke dalam data tabel dua arah dengan beberapa kombinasi banyaknya *outlier* dengan cara dua jenis penempatan yaitu secara *Scattered* dan *Single Environment*. Pada penempatan secara *Scattered* komposisi *outlier* sebesar 2%, 5%, dan 10% dari data, sedangkan pada penempatan secara *Single Environment* komposisi *outlier* sebesar 5%, 10%, dan 15%.

Berdasarkan penelitian yang menggunakan data tersebut, diperoleh hasil bahwa model RCIM mampu mempertahankan model sebaik model RCIM2 meskipun data terpengaruh oleh adanya pencilan. Semakin banyak komposisi pencilan serta jenis pencilan yang beragam pada data model RCIM, model RCIM mengalami penurunan rank namun masih mampu memodelkan data sebaik model data awal pada rank-rank tertentu. Sedangkan pada model *Robust* Faktor semakin banyak komposisi *outlier* semakin besar nilai MSE pada *Scattered* dan semakin kecil pada *Single Environment*. Sehingga dapat dikatakan bahwa pada *Scattered* model *Robust* Faktor semakin jelek dalam memodelkan data *outlier* sedangkan pada *Single Environment* model *Robust* Faktor semakin baik dalam memodelkan data *outlier*. Dengan dua kesimpulan yang berbeda maka dapat dikatakan bahwa model *Robust* Faktor tidak konsisten dalam menangani data *outlier*, hal ini tergantung pada penempatan *outlier* serta banyaknya *outlier* dalam data. Perbandingan nilai MSE dari model RCIM dengan model *Robust* Faktor menunjukkan bahwa nilai MSE dari model RCIM secara keseluruhan lebih kecil dari model *Robust* Faktor sehingga model RCIM lebih bagus dalam memodelkan data *outlier* daripada model *Robust* Faktor.

PRAKATA

Puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat serta hidayahNya sehingga tesis yang berjudul "Penanganan Pencilan pada Tabel Dua Arah dengan menggunakan RCIM dan *Robust* Faktor " dapat terselesaikan. Tesis ini disusun untuk memenuhi salah satu syarat dalam menyelesaikan pendidikan strata 2 (S2) di Jurusan Magister Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember. Sholawat serta salam semoga selalu tercurahkan keharibaan beliau nabi Muhammad SAW yang telah menjadi pembawa rahmatan lil'alamin.

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

1. Ayahanda Lutfillah dan Ibunda Arifatus Sholihah yang selalu memberi doa serta dukungan baik lahir maupun batin;
2. Adik tercinta Alvin Nashihin Nazih dan Arisy Khalilur Rahman yang telah memberikan dukungan serta motivasi dalam pembuatan tesis ini;
3. 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 tesis ini;
4. Prof. Drs. I Made Tirta, M. Sc., Ph. D dan Dr. Mohamat Fatekurohman, S.Si., M.Si selaku Dosen Penguji yang telah memberikan kritik dan saran yang membangun untuk penyempurnaan tesis ini;

5. Dr. Sujito, Ph. D. selaku Dekan Fakultas MIPA Universitas Jember dan Dr. Alfian Futuhul Hadi, S.Si, M. Si selaku Ketua Jurusan Magister Matematika Fakultas MIPA Universitas Jember atas izin penelitian serta penggunaan fasilitas yang mendukung dalam penyelesaian tesis ini;
6. seluruh dosen dan karyawan Jurusan Magister Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam yang telah memberikan ilmu serta fasilitas yang membantu selama proses perkuliahan berlangsung;
7. Sahabat Dimas, Sahabat Ulum, Sahabati Anita, Sahabat Susilo, dan seluruh Sahabat-sahabati PMII Rayon FMIPA yang tidak bisa saya sebut satu per satu yang selalu senantiasa menemani, memberi dukungan, semangat perjuangan, serta saran dalam proses menyelesaikan tugas akhir;
8. teman-teman Chaponiks (Surur, Syukma, Onne, Putri, Muafa, Karinda, Amanah, Almh.Nadiya) yang senantiasa memberi dukungan dalam proses menyelesaikan tugas akhir;
9. keluarga besar MATGHIC '10 yang selalu memberikan dukungan dalam hal apapun.
10. teman-teman kos 41 A (Yudis, Leli, Aini, Faida, Ana) yang selalu memberikan dorongan dan semangat dalam situasi dan kondisi apapun.

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

Jember, Juni 2017 Kurnia Ahadiyah

DAFTAR ISI

Halaman

DAFTAR TABEL

DAFTAR GAMBAR

Halaman

DAFTAR LAMPIRAN

Halaman

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Dalam beberapa analisis data, sering ditemukan data yang melibatkan banyak variabel. Hal yang perlu diperhatikan dari data tersebut adalah adanya pencilan (*outlier*) atau tidak pada data, karena sedikit banyak akan berpengaruh terhadap hasil analisis data. *Outlier* atau pencilan merupakan titik sampel yang memiliki karakteristik unik dan dapat diidentifikasi secara jelas karena berbeda dengan mayoritas titik sampel lainnya.

Menurut Hawskin (1980), *outlier* adalah sebuah pengamatan yang menyimpang sangat besar dari pengamatan lainnya yang dibangun oleh langkah yang berbeda. Johnson (1992) mendefinisikan *outlier* sebagai sebuah pengamatan pada himpunan data dimana memunculkan ketidakkonsistenan terhadap sisa dari himpunan data. Pada umumnya, *outlier* dapat terjadi karena kesalahan manusia, kesalahan instrumen, perilaku curang, perubahan perilaku sistem atau kesalahan sistem, dan penyimpangan alami di dalam populasi. Kehadiran *outlier* sering kali berdampak buruk terhadap analisis data, karena *outlier* mampu menyimpangkan uji-uji statistik yang didasarkan pada dua penaksir klasik yaitu rerata sampel dan kovariansi sampel.

Dalam ilmu statistika, *outlier* dalam data dibagi menjadi dua, yaitu *outlier* univariat dan *outlier* multivariat. *outlier* univariat merupakan *outlier* yang disebabkan oleh variabel terikat atau variabel dependen sedangkan *outlier* multivariat merupakan *outlier* yang disebabkan oleh sekumpulan variabel bebas atau variabel independen. Metode pendeteksian *outlier* juga dibagi menjadi metode univariat (*univariat methods*) dan metode multivariat (multivariat *methods*). Deteksi *outlier* dengan metode univariat dapat dilakukan dengan menentukan nilai batas yang akan dikategorikan sebagai data *outlier* yaitu dengan cara mengkonversi nilai data ke dalam skor standardized atau yang biasa disebut *z-score*, yang memiliki nilai rata-rata (mean) sama dengan nol dan standar deviasi sama dengan satu, sedangkan konsep dasar dalam pendeteksian *outlier* melalui metode multivariat adalah dengan mengukur jarak setiap titik ke pusat datanya. Titik sampel yang memencil akan memiliki nilai jarak yang besar relatif terhadap mayoritas titik yang lain. Titik yang memencil ini patut dicurigai sebagai *outlier* (Bengal, 2005). Pada data yang mengandung *outlier* univariat, *outlier* dapat diidentifikasi secara mudah melalui grafik, tetapi dalam data yang mengandung *outlier* multivariat, *outlier* menjadi sulit diidentifikasi. *Outlier* univariat maupun multivariat dapat terjadi pada beberapa macam jenis data dengan metode analisis statistik yang beragam.

Pada penelitian ini, penulis lebih fokus terhadap penelitian *outlier* univariat pada jenis data tabel dua arah dikarenakan pendeteksian *outlier* pada data satu arah telah banyak diteliti serta pendeteksian *outlier* pada data tabel dua arah membutuhkan metode yang lebih kompleks daripada data satu arah. Dalam model statistika, terdapat beberapa metode yang dapat digunakan untuk analisis data yang mengandung *outlier*. *Outlier* pada satu arah tertentu memungkinkan menjadi pengaruh dalam pengujian ketaknormalan. Dalam hal ini diperlukan kajian sensitifitas pengaruh *outlier* terhadap pengujian ketaknormalan. Aini (2015) telah meneliti metode multivariat yaitu *Robust* SVD (RobRSVD) pada model AMMI dimana model tersebut digunakan pada data yang menyebar secara normal. Pada penguraian matriks interaksi model AMMI diperoleh nilai eigen, vektor eigen kiri dan kanan secara iteratif melalui metode penguraian nilai singular dengan *outlier* L1 dan implementasi dari norma ini disebut sebagai *alternating L1 regression*. Metode ini digunakan untuk menduga nilai eigen, vektor eigen kiri dan kanan sehingga hasil dugaannya tahan terhadap *outlier*. Menurut Yee dan Hadi (2014), *Row Column Interaction Model* (RCIM) merupakan perluasan dari konsep *Reduced-Rank Vektor Generalized Linear Models* (RR-VGLM). RCIM merupakan suatu model yang dapat diterapkan sama halnya dengan GAMMI yaitu dapat menganalisis tabel dua arah dengan matriks responnya berdistribusi lebih meluas daripada model AMMI.

Di sisi lain, Ardian (2015) telah meneliti data multivariat berdistribusi normal yang mengandung *outlier* dengan menggunakan *Robust* faktor melalui pendekatan robPCA untuk memeroleh hasil evaluasi penanganan *outlier* pada data tabel dua arah. RobPCA

(*Robust* PCA) didasarkan pada metode PCA (*Principal Component Analysis*) yang mengatasi data dengan adanya *outlier*. PCA (*Principal Component Analysis*) merupakan analisis multivariat yang dapat digunakan untuk memperkecil dimensi suatu peubah. Dalam PCA, komponen pertama merupakan komponen yang mempunyai varians terbesar, sedangkan komponen kedua merupakan orthogonal komponen pertama yang memaksimalkan varians dari titik data yang di proyeksikan, dan komponen selanjutnya sesuai dengan eigen vektor matriks kovarian. Akibatnya, komponen pertama PCA sering mengarah pada titik *outlier* dan tidak mengarah pada varians pengamatan lainnya karena variannya lebih kecil. Oleh karena itu, reduksi data berdasarkan PCA menjadi tidak dapat diandalkan jika ada *outlier* dalam data (Hubert, Rousseeuw, &Branden, 2005).

Model RCIM yang merupakan perluasan model AMMI adalah model yang sering digunakan untuk analisis Interaksi *Genotype* × Lingkungan pada tabel dua arah. Yee dan Hadi (2014) melakukan penelitian dengan model RCIM tanpa adanya *outlier* pada data, sedangkan dalam upaya merakit sifat-sifat unggul pada suatu *genotype*, *outlier* justru menjadi sesuatu yang berharga, oleh karena itu mengabaikan *outlier* tidaklah bijaksana. Untuk itu diperlukan metode yang relatif ”tegar” (*robust*) terhadap adanya *outlier*. Berdasarkan uraian tersebut, peneliti ingin melakukan penelitian terkait penanganan *outlier* pada tabel dua arah dengan dua metode yaitu metode *robust* SVD (RobSVD) pada model RCIM dan metode *robust* PCA (robPCA) pada analisis faktor dengan memodifikasi nilai *outlier* pada data tabel dua arah.

1.2 Rumusan Masalah

Adapun rumusan masalah dalam penelitian ini adalah membandingkan pemodelan suatu data tabel dua arah yang mengandung *outlier* (*outlier*) dan mencari model terbaik dengan menggunakan model RCIM (*Row Column Interaction Model*) dan *robust* faktor serta mengidentifikasi seberapa besar batas kedua metode tersebut terhadap *outlier*. Pada penelitian ini, metode *robust* pada model RCIM terletak pada penguraian nilai singular tiap-tiap model yaitu dengan menggunakan metode RobRSVD sedangkan pada analisis faktor metode *robust* terletak analisis komponen utama dengan menggunakan metode RobPCA. Batas ketahanan kedua model tersebut terhadap *outlier* dilihat dari besar *outlier* yang dibangkitkan, serta presentase banyak dan posisi *outlier* yang ada pada data tabel dua arah. Rumusan masalah pada penelitian ini dititikberatkan pada :

1. Bagaimana model RCIM dalam menangani data yang mengandung *outlier* serta ketahanan terhadap besar *outlier* yang dibangkitkan, presentase banyak, dan posisi *outlier* yang ada pada data tabel dua arah.
2. Bagaimana model *Robust* Faktor dalam menangani data yang mengandung *outlier* serta ketahanan terhadap besar *outlier* yang dibangkitkan, presentase banyak, dan posisi *outlier* yang ada pada data tabel dua arah.
3. Bagaimana perbandingan ketahanan model RCIM dan model *Robust* Faktor terhadap *outlier* pada data tabel dua arah.

1.3 Tujuan Penelitian

Adapun tujuan dalam penelitian ini adalah mendapatkan metode terbaik yang dapat digunakan untuk analisis data tabel dua arah yang mengandung *outlier* dengan membandingkan dua metode, yaitu model RCIM dan *robust* faktor serta mengetahui pengaruh besar, presentase, dan posisi *outlier* pada tabel dua arah.

1.4 Manfaat Penelitian

Adapun manfaat dalam penelitian ini adalah dapat mengantisipasi adanya *outlier* pada data tabel dua arah khususnya data interaksi *Genotype* × Lingkungan sehingga dapat memaksimalkan potensi genetik suatu tanaman serta diperoleh suatu varietas yang mempunyai kemampuan adaptasi yang baik terhadap kondisi lingkungan yang berbeda

1.4 Batasan Masalah

Adapun batasan masalah pada penelitian ini adalah data tabel dua arah yang digunakan berdistribusi normal dengan kajian utama pada dekomposisi interaksi melalui beberapa simulasi *outlier*.

BAB 2. TINJAUAN PUSTAKA

2.1. Pencilan (*Outlier*)

Pencilan (*outlier*) didefinisikan sebagai bagian dari data pengamatan yang mempunyai pola berbeda dari sebagian besar data pengamatan. *Outlier* merupakan data yang muncul dengan karakteristik unik yang terlihat sangat jauh berbeda dari observasi-observasi lainnya dan muncul dalam bentuk nilai ekstrim baik untuk sebuah variabel tunggal atau variabel kombinasi. *Outlier* pada data dapat menyebabkan ketakhomogenan matriks varian kovarian. Menurut Hadi *et al* (2009) menyebutkan bahwa pencilan memnerikan efek mengaburkan data ($masking$) dan kesalahan mengidentifikasi data non *outlier* sebagai *outlier* ($swamping$). Sebaiknya jumlah *outlier* dalam data tidak lebih dari 50%. Adanya *outlier* dapat menyebabkan beberapa penyimpangan, diantaranya adalah :

1. Residual yang besar dari model yang terbentuk
2. Varian data menjadi lebih besar
3. Rentang yang lebar pada wilayah kepercayaan (*confidence region*)

Suatu pengamatan dianggap *outlier* jika pengamatan tersebut bernilai lebih besar dari rata-rata ditambahkan dengan tiga kali standar deviasi (Kriegel,2010). Secara matematis pengamatan dianggap *outlier* jika :

$$y_i \geq \bar{y} + 3.stdev(y_i) \qquad (2.1)$$

Menurut Rockie dan Woodruff (1996), *Shift Outlier* didefinisikan sebagai penempatan data yang buruk ke dalam data yang baik dengan distribusi kedua data sama.Data disebut data yang baik jika data berdistribusi normal dengan mean μ_0 dan varian Σ_0 . Sedangkan data yang buruk jika data tersebut mempunyai mean $\mu_0 + \mu$ dan varian Ω. Dengan konsep tersebut Rockie dan Woodruff membagi jenis outlier menjadi tiga jenis :

1. *Pure Shift Outlier*, yaitu *outlier* yang menggeser rata-rata data yang baik.
2. *Pure Point Mass Outlier*, yaitu *outlier* yang merubah varian data yang baik.
3. *Shift Point Mass Outlier*, yaitu *outlier* yang menggeser rata-rata dan merubah varian suatu data yang baik.

2.2. Model RCIM (*Row Column Interaction Model*)

Yee dan Hastie (2003) memperkenalkan kelas *Reduced - Rank Vektor Generalized Linear Models* (RR-VGLMs) dimana mengaplikasikan konsep regresi *reduced rank* ke dalam kelas VGLM. Misalkan data terdiri dari $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ untuk $i = 1, \ldots, n$ dimana $\boldsymbol{x}_i$ merupakan vektor variabel penjelas untuk observasi ke$-i$ dan $\boldsymbol{y}_i$ merupakan variabel respon (memungkinkan dalam bentuk vektor). Umumnya VGLM mirip dengan GLM, namun VGLM menyediakan prediktor linier yang lebih dari satu. VGLM didefinisikan sebagai sebuah model dimana,

$$f(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{B}) = f(\boldsymbol{y}, \eta_1, \ldots, \eta_M) \tag{2.2}$$

Untuk beberapa fungsi $f(.), \boldsymbol{B} = (\boldsymbol{\beta}_1\ \ \boldsymbol{\beta}_2\ \ldots \boldsymbol{\beta}_M)$ adalah matriks koefisien regresi yang tidak diketahui dengan ukuran $p \times M$. VGLM dapat menangani sebanyak M prediktor linier tergantung dari model yang dipaskan dimana persamaan salah satu prediktor linier ke$-j$ adalah sebagai berikut :

$$\eta_j(\boldsymbol{x}) = \boldsymbol{\beta}_j^T \boldsymbol{x} = \sum_{k=1}^{p} \beta_{(j)k} x_k\,,\ j = 1. \ldots, M \tag{2.3}$$

η_j pada VGLM diaplikasikan secara langsung pada parameter distribusi seperti pada GLM, secara umum,

$$\eta_j(\boldsymbol{x}) = g_j\big(\theta_j\big),\ \ l = 1, \ldots, M \tag{2.4}$$

g_j merupakan parameter fungsi *link* karena θ_j merupakan banyak parameter dan tidak dibatasi untuk menjadi mean seperti pada GLM. Pada persamaan 2.4 mencakup lebih dari model GLM, dapat dituliskan sebagai berikut :

$$\boldsymbol{\eta}_i = \begin{pmatrix} \eta_1(x_i) \\ \vdots \\ \eta_M(x_i) \end{pmatrix} = \boldsymbol{B}^T(\boldsymbol{x}_i) = \begin{pmatrix} \boldsymbol{\beta}_1^T \boldsymbol{x}_i \\ \vdots \\ \boldsymbol{\beta}_M^T \boldsymbol{x}_i \end{pmatrix} \tag{2.5}$$

dengan $\boldsymbol{B}$ merupakan matriks $p \times M$ dari koefisien regresi. Pada banyak kasus, koefisien regresi duhubungkan satu sama lain. seperti contoh, beberapa $\boldsymbol{\beta}_j^T \boldsymbol{x}$ mungkin sama, nol, atau penambahan ke jumlah tertentu. Pada situasi ini mungkin dilakukan dengan menggunakan matriks *constraint*. VGLMs secara umum diberikan

$$\boldsymbol{\eta}(\boldsymbol{x}_i) = \sum_{k=1}^{p} \boldsymbol{\beta}_{(k)}\, x_{ik} = \sum_{k=1}^{p} \boldsymbol{H}_k \boldsymbol{\beta}_{(k)}^{*}\, x_{ik} \tag{2.6}$$

dengan $\boldsymbol{H}_1$, $\boldsymbol{H}_2$, ... , $\boldsymbol{H}_p$merupakan matriks *constraint* yang diketahui dari *full column-rank* dan $\boldsymbol{\beta}_{(k)}^{*}$ merupakan vektor yang berisi himpunan koefisien regresi yang tereduksi.dari parameter yang diketahui. Dengan tidak ada *constraint*, $\boldsymbol{H}_1 = I_M$ dan $\boldsymbol{\beta}_{(k)}^{*} = \boldsymbol{\beta}_{(k)}$. Kemudian

$$\boldsymbol{B}^T = (\,\boldsymbol{H}_1 \boldsymbol{\beta}_{(1)}^{*} \quad \boldsymbol{H}_2 \boldsymbol{\beta}_{(2)}^{*} \quad \ldots. \quad \boldsymbol{H}_p \boldsymbol{\beta}_{(p)}^{*}) \tag{2.7}$$

Partisi $\boldsymbol{x}_i$ kedalam $(\boldsymbol{x}_{1i}^T, \boldsymbol{x}_{2i}^T)^T$ dan $\mathbf{B} = (\boldsymbol{B}_1^T, \boldsymbol{B}_2^T)^T$. Jika B_2 mempunyai banyak koefisien regresi sehingga kita dapat mereduksi dengan menggunakan konsep regresi *reduced-rank*. Sehingga RR-VGLM diberikan

$$\eta_i = \boldsymbol{B}_1^T \boldsymbol{x}_{1i} + \boldsymbol{B}_2^T \boldsymbol{x}_{2i} \tag{2.8}$$

dengan $\boldsymbol{B}_2$ dengan *reduced-rank regression*

$$\boldsymbol{B}_2^T = \boldsymbol{A}\boldsymbol{C}^T \tag{2.9}$$

C dan **A** merupakan matriks *full column rank* dengan ukuran $p_2 \times R$ dan $M \times R$. sehingga,

$$\eta_i = \boldsymbol{B}_1^T x_{1i} + \boldsymbol{AC}^T \boldsymbol{x}_{2i} = \boldsymbol{B}_1^T x_{1i} + \boldsymbol{A}v_i \tag{2.10}$$

dengan $v = C^T x_2$ adalah vektor R variabel laten. Untuk membuat parameter yang unik, umumnya dengan cara menjalankan matriks *corner constraint* pada $\boldsymbol{A}$. Dengan menggunakan GRC model (Goodman's RC model) akan dijelaskan mengenai RCIM.

RCIM merupakan bagian dari RR-VGLM dimana prediktor linier yang pertama dimodelkan dengan penjumlahan dari pengaruh baris pengaruh kolom, dan pengaruh interaksi yang mana pada pengaruh interaksi ditunjukkan seperti regresi *reduced-rank*. Sehingga sebuah model RCIM secara umum didefinisikan seperti RR-VLGM yang diaplikasikan pada variabel respon Y sebagai berikut :

$$g_1(\theta_1) \equiv \eta_{1ij} = \mu + \alpha_i + \gamma_j + \sum_{r=1}^{R} c_{ir}\, a_{jr} \tag{2.11}$$

dengan $R \leq \min(M, p_2)$. Hal ini berarti parameter pertama pada model statistik yang berhubungan dengan variabel respon pada matriks. Setelah ditransformasikan, hasilnya sama dengan penjumlahan dari *intercept*, pengaruh baris dan kolom, serta interaksi yang berbentuk $\boldsymbol{A^T C}$. Adapun beberapa fungsi keluarga VGAM yang digunakan dalam `rcim()` sebagaimana ditunjukkan pada tabel 2.1.

Uninormal dapat disebut dengan *univariate normal distribution*. Model ini merupakan model linier yang mempunyai fungsi link $\eta = (\mu, \log \sigma)^T$. Dimana μ adalah prediktor linier pertama dan log dari standar deviasi adalah sebagai prediktor linier yang kedua. Selama fungsi tersebut mengandung asumsi homoskedastisitas, maka uji heteroskedastisitas dapat digunakan dengan $H_0: a_{21} = 0$ vs $H_1: a_{21} \neq 0$ karena $K_2 = e^{a_{21}}$ dan fungsi varian $Var(Y) \equiv V(\mu) = K_1^2 . K_2^{2\mu}$. Adapun fungsi yang digunakan untuk mendapatkan varian model uninormal adalah sebagai berikut

$$\log \sigma = \theta_0^* + \theta_1^* \mu \tag{2.12}$$

Dimana θ_0^* dan θ_1^* merupakan parameter yang di*fit*kan seperti pada VGLM atau RR-VGLM. (Yee, 2015)

Tabel 2.1. Jenis Fungsi yang digunakan dalam RCIM

Jenis Distribusi	Keterangan
`binomialff(multiple.response=TRUE)`	Model Rasch
`negbinomial()`	Model GRC dengan overdispersi pada regresi Poisson
`poissonff()`	Model GRC
`uninormal()`	ANOVA dua arah (tanpa ulangan)
`zipoissonff()`	Model GRC dengan beberapa nilai nol

2.3. *Robust* SVD

Untuk mereduksi dimensi data berdasarkan keragaman data (nilai eigen) terbesar dengan mempertahankan informasi optimum diperlukan penguraian nilai singular. Penguraian nilai singular (*Singular Value Decomposition*) merupakan salah satu konsep aljabar matriks dan konsep *eigen decomposition* yang terdiri dari nilai eigen λ dan vektor eigen. Hasil yang didapatkan dari penguraian nilai singular adalah vektor eigen kanan, vektor eigen kiri, dan nilai eigen (Sumertajaya & Mattjik, 2011). Namun, berbeda halnya pada data yang mengandung *outlier*. Cara untuk mendapatkan vektor eigen kanan dan vektor eigen kiri serta nilai eigen adalah adalah penguraian nilai singular dengan pencilan pada L_1 dan implementasi dari norma ini disebut dengan *alternating* L_1 *regression*. Metode ini digunakan untuk menduga nilai eigen, vektor eigen kiri dan kanan sehingga hasil dugaannya tahan terhadap *outlier* (Warsito, 2009).

Adapun algoritma dari *alternating* L_1 *regression* sebagai berikut :

1. Dimulai dengan menentukan dugaan awal untuk vektor eigen kiri a_1
2. Untuk tiap-tiap kolom j, dengan $j = 1, 2, \dots, p$, tentukan koefisien regresi L1 yaitu c_j dengan meminimumkan $\sum_{i=1}^{n} |x_{ij} - c_j a_{i1}|$.
3. Hitung hasil dugaan vektor eigen kanan $b_1 = \frac{c}{||c||}$ dimana $||.||$ adalah norma *euclidean*.
4. dengan menggunakan dugaan dari vektor eigen kanan, perhalus dugaan vektor eigen kiri. Untuk masing-masing baris , dengan $i = 1, 2, \dots, n$, tentukan koefisien regresi L1 yaitu d_i dengan meminimumkan $\sum_{j=1}^{p} |x_{ij} - d_i b_{j1}|$.
5. Hitung hasil dugaan vektor eigen kiri $a_1 = \frac{d}{||d||}$.
6. iteraskan sampai konvergen.

(Hawkins, *et. al*. 2001)

2.4. *Robust* Faktor

Analisis faktor merupakan analisis tentang saling ketergantungan antara variabel-variabel, dengan tujuan untuk menemukan himpunan variabel-variabel baru yang lebih sedikit jumlahnya daripada variabel semula. Dalam analisis faktor, variabel-variabel dalam jumlah besar dikelompokan dalam sejumlah faktor yang mempunyai sifat dan karakteristik yang hampir sama, sehingga lebih mempermudah pengolahan. Pengelompokan dilakukan dengan mengukur hubungan sekumpulan variabel dan selanjutnya menempatkan varibel-variabel yang berkorelasi tinggi dalam satu faktor, dan variabel-variabel lain yang mempunyai korelasi relatif lebih rendah ditempatkan pada faktor yang lain.

Robust faktor merupakan salah satu metode yang dapat mengatasi adanya *outlier* pada data tabel dua arah. Adapun algorima robust faktor menggunakan tiga algoritma, yaitu algoritma `Weight.wl1`, *Robust Principle Component Analysis*, dan algorima `Twoway.rob`. Adapun penjelasan dari tiap-tiap algoritma akan dijelaskan sebagai berikut:

2.4.1 Algoritma `Weight.wl1`

Algoritma ini digunakan untuk mendapatkan bobot dari masing-masing baris maupun kolom untuk regresi L1 terboboti (*Weight L1 regression*) dari data. Adapun langkah-langkah nya sebagai berikut :

1. Data dibentuk ke dalam data frame.
2. Jika nilai baris/kolom bernilai 1, maka :
   - Vektor rata-rata diperoleh dari median data.
   - Matriks peragam diperoleh dari penduga skala data yang robust kemudian dikuadratkan.
3. Jika bernilai lebih dari 1, maka vektor rata-rata dan matriks peragam diperoleh dari nilai penduga skala dengan *high breakdown point* menggunakan metode *Minimum Volume Ellipsoid*.
4. Nilai vektor rata-rata dan matriks peragam yang diperoleh digunakan untuk mengidentifikasi outlier berdasarkan *Mahalanobis Distance* (MD) dengan ruang berdimensi k.
5. Nilai pembobot baris maupun kolom diperoleh dari nilai sebaran khi kuadrat dengan batas atas bagi nilai kritis 5% berderajat bebas k dibagi dengan nilai dari *Mahalanobis Distance* (MD).

2.4.2. Robust Principle Component Analysis (RobPCA)

Metode RobPCA adalah suatu metode PCA yang *robust* terhadap keberadaan pencilan (*outlier*) pada data. Komponen *Loading* dihitung dengan menggunakan teknik *Projection-Pursuit* (PP) dan metode *Minimum Covariance Determinant* (MCD). Teknik PP digunakan untuk reduksi dimensi awal kemudian estimator MCD diaplikasikan menghasilkan estimasi yang lebih akurat. Menurut Hubert *et al* (2005), metode RobPCA dapat didefinisikan sebagai berikut :

Jika dasumsikan data asal berupa suatu matriks $\boldsymbol{X}$ berukuran $n \times p$ dimana n jumlah pengamatan dan p jumlah variable asal maka meode RobPCA dilakukan dengan 3 langkah berikut :

1. Pertama, dilakukan pre-proses data sehingga transformasi data berada pada sebuah subruang dengan dimensi paling tinggi $n-1$.
2. Kemudian dibentuk matriks kovarian S_0 yang digunakan untuk memilih jumlah komponen k sehingga menghasilkan subruang berdimensi k yang cocok dengan data.
3. Titik-titik data kemudian diproyeksikan ke subruang ini kemudian lokasi dan matriks sebarannya diestimasi secara *robust* dan dihitung nilai eigen $I_1, I_2, \dots, I_k$. Maka didapat vektor eigen yang bersesuaian adalah sejumlah k komponen utama yang *robust*.

2.4.3. Algoritma `Twoway.rob`

Algoritma ini digunakan untuk memodelkan data ke dalam bentuk tabel dua arah yang *robust* terhadap *outlier*. Adapun model persamaannya sebagai berikut :

$$x_{ij} = \mu + a_i + b_j + \sum_{t=1}^{k} f_{it} \, . \sigma_t \, . l_{jt} + e_{ij} \qquad (2.13)$$

dengan :

$i = 1, \dots, n$

$j = 1, \dots, p$

$x_{ij} = (i, j)$ dari tabel dua arah

$\mu =$ rata-rata keseluruhan

$a_i =$ efek baris

$b_j =$ efek kolom

Adapun algoritma `Twoway.rob` adalah :

1. Mendefinisikan panjang baris dan kolom data yang dimasukkan
2. Mendefinisikan nama untuk baris dan nama untuk kolom.

3. Jika panjang baris data lebih kecil daripada panjang kolom data, maka dilakukan transpose pada data untuk menukar baris dan kolom data.
4. Mendefinisikan matriks bobot baris, bobot kolom dan nilai awal fungsi objektif.
5. Mencari nilai invarian terhadap perubahan nilai efek baris data.
6. Melakukan prosedur robust analisis komponen utama (RobPCA) dengan estimasi berbasis *Projection Pursuit* (PP). Nilai yang dihasilkan diambil sebagai nilai awal bagi skor.
7. Proses iterasi RobPCA sebagai berikut :
   - Hitung bobot baris yang *downweight outlier*. Kemudian menghitung nilai penduga loading dengan menggunakan regresi L1 terboboti. Yang diperbarui dari regresi tersebut diantaranya adalah penduga loading, nilai efek baris, nilai efek kolom.
   - Cara yang sama dilakukan untuk menghitung bobot kolom yang *downweight outlier*. Kemudian nilai penduga skor dengan menggunakan regresi L1 yang terboboti. Yang diperbarui dari regresi tersebut diantaranya adalah penduga *loading*, nilai efek baris, nilai efek kolom.
   - Nilai fungsi tujuan atau objektif yang diperoleh berdasarkan iterasi sebelum dan sesudahnya akan dibandingkan. Jika tidak ada perbedaan dalam fungsi tujuan berdasarkan toleransi yang didefinisikan, maka proses iterasi dihentikan. Jika ada perbedaan maka iterasi diulang sampai iterasi maksimal.
8. Setelah diperoleh nilai konvergen / iterasi mencapai maksimal, maka hasil keluaran dari algoritma diantaranya residual, biplot dan boxplot.

2.5. Interpretasi Biplot

Biplot merupakan teknik statistika deskriptif yang digunakan untuk menyajikan secara simultan n objek pengamatan dan p atribut dalam ruang bidang datar sehingga ciri-ciri objek pengamatan dan atribut serta posisi antara keduanya dapat dianalisis. Biplot pertama kali diperkenalkan oleh Gabriel pada tahun 1971. Menurut Gabriel (1971), Biplot merupakan metode yang sering digunakan dalam analisis multivariat untuk menggambarkan elemen baris dan kolom ke dalam suatu grafik.

Dasar dari analisis Biplot adalah SVD (*Singular Value Decomposition)* dari suatu matriks. SVD (*Singular Value Decomposition*) dari suatu matriks merupakan teorema dasar yang banyak digunakan dalam perhitungan matriks. SVD (*Singular Value Decomposition*) banyak digunakan karena secara komputerisasi lebih efisien dalam menghasilkan skor komponen utama pertama dan kedua.

Menurut Gabriel (1971), kebaikan biplot dapat ditentukan dengan ukuran pendekatan matriks X dalam bentuk:

$$\rho^2 = \frac{(\lambda_1+\lambda_2)}{\sum_{k=1}^{r}\lambda_k} \tag{2.14}$$

dengan λ_1 adalah nilai eigen terbesar ke-1, λ_2 adalah nilai eigen terbesar ke-2 dan λ_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 mampu memberikan informasi sebesar 70% dari seluruh informasi dianggap cukup (Mattjik & Sumertadjaya, 2011).

2.6. Goodness Of Fit Model

Dalam model statistika, ada beberapa metode yang digunakan untuk menguji model manakah yang lebih baik. Namun terkadang dua metode yang berbeda dapat memberikan hasil yang berbeda juga, sehingga tidak ada ketetapan khusus metode manakah yang lebih baik dalam mengukur kecocokan model (*goodness of fit model*). MSE merupakan nilai yang diharapkan dari kuadrat *error*. *Error* yang ada menunjukkan seberapa besar perbedaan hasil estimasi dengan nilai yang akan diestimasi. Model dapat dikatakan model yang lebih baik jika mempunyai nilai MSE yang lebih kecil.

Secara matematis MSE dapat dituliskan sebagai berikut :

$$MSE = \frac{1}{N}\sum_{t=1}^{N}(y_t - \hat{y}_t)^2 \qquad (2.15)$$

dengan :

MSE = *Mean Squared Error*

N = Jumlah Populasi

y_t = Nilai Aktual

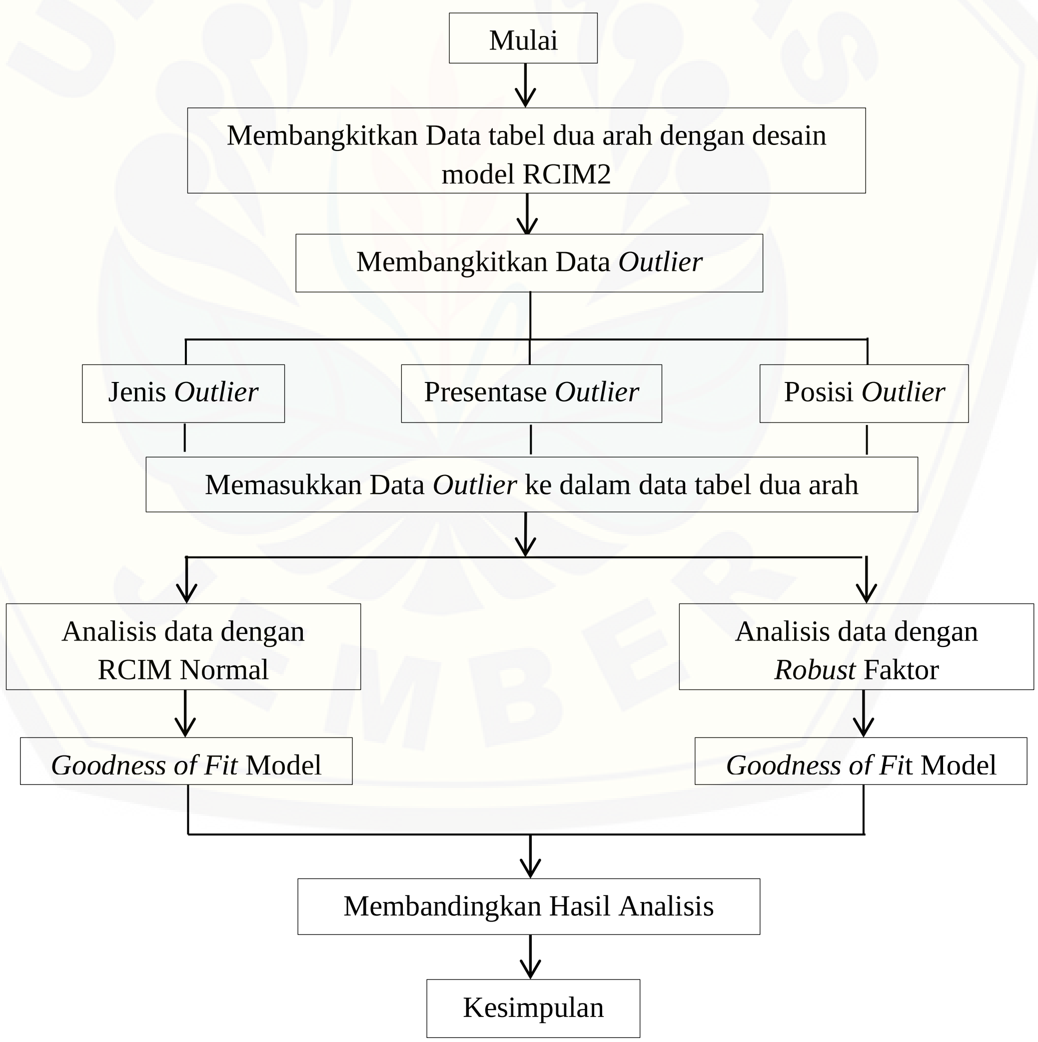
$\hat{y}_t$ = Nilai Prediksi

BAB 3. METODE PENELITIAN

Pada bab ini akan dijelaskan mengenai metode penelitian sebagai konsep dasar. Adapun metode yang digunakan dalam penelitian ini meliputi data dan analisis data dengan menggunakan model RCIM dan model *robust* faktor pada data berdistribusi normal.

3.1 Data

Adapun data yang digunakan dalam penelitian ini adalah data simulasi dengan menggunakan program R. Langkah-langkah simulasi dan desain penelitian keseluruhan dalam penelitian ini disajikan dalam diagram alir dibawah ini :



Data penelitian ini menggunakan data tabel dua arah dimana variabel respon mengikuti sebaran normal. Data diperoleh dengan cara simulasi. Data tabel dua arah dibangkitkan secara acak dengan ukuran $n = 50$ baris/genotipe dan $p = 8$ kolom/lingkungan. Interaksi baris dan kolom dijelaskan dengan dua suku multiplikatif atau disebut dengan RCIM2 dimana RCIM2 merupakan model RCIM yang signifikan pada Rank=2. Adapun langkah simulasi data tabel dua arah RCIM2 sebagai berikut :

1. Membuat sebuah matrix $\boldsymbol{X}$ dengan $n = 50$ baris/genotipe dan $p = 8$ kolom/lingkungan dengan pengamatan mengukuti distribusi uniform $[-0.5,0.5]$.
2. Matriks $\boldsymbol{X}$ diuraikan dengan menggunakan metode SVD (*Singular Value Decomposition*) dan diperoleh matriks U, V, dan D.
3. Membangkitkan rata-rata total(μ), pengaruh utama baris (α) dan pengaruh utama kolom (β) dengan ketentuan $\mu \sim N(15,3), \alpha \sim N(5,1), \beta \sim N(5,1)$.
4. Membangkitkan data tabel dua arah $\boldsymbol{Y}$ dengan struktur RCIM2 ; $Y = 1_I 1_J^T \mu + \alpha_I 1_J^T + 1_I \beta_J^T + 28 \times U[,1]D[1,1]V[,1]^T + 1_I \beta_J^T + 15 \times U[,2]D[2,2]V[,2]^T$ dengan $A[,i]$ menunjukkan kolom ke$-i$ pada matriks A dan $A[i,i]$ menunjukkan baris ke $-i$ dan kolom ke $-i$ pada matriks A dimana $i = 1,2$. (Dias dan Krzanowski, 2003).

Pada data tabel dua arah diatas selanjutnya ditambahkan beberapa *outlier* dengan tiga jenis *outlier* yang didasarkan pada penelitian Rockie dan Woodruff (1996), ketiga jenis tersebut adalah :

1. *Pure Shift Outlier* . *Outlier* ini dibangkitkan dari distribusi normal $N(\mu_j + k\sigma_j, \sigma_j^2)$.
2. *Pure Point Mass Outlier*. *Outlier* ini dibangkitkan dari distribusi normal $N(\mu_j, \sigma_j^2/100)$.
3. *Shift Point Mass Outlier*. *Outlier* ini dibangkitkan dari distribusi normal $N(\mu_j + k\sigma_j, \sigma_j^2/100)$.

dimana nilai $k = 4{,}10$ serta μ_j dan σ_j diambil dari rata-rata dan standart deviasi pada sampel data awal. Ketiga jenis *outlier* tersebut kemudian ditempatkan pada data tabel dua arah RCIM2 dengan beberapa ketentuan yang disarankan oleh penelitian Rodrigues, *et.al* (2015) sebagai berikut :

1. *Scattered*

   Outlier ditempatkan untuk merepresentasi posisi secara acak (*at random*) dengan cara memilih baris/*genotype* secara acak kemudian memilih kolom/lingkungan secara acak. Selanjutnya *outlier* ditempatkan pada baris dan kolom yang telah terpilih secara acak. *Outlier* yang lain ditempatkan dengan cara yang sama. Pada simulasi ini komposisi *outlier* dibangkitkan sebanyak 2%, 5%, dan 10% dari data.

2. *Single Environment*

   Outlier ditempatkan secara acak dengan cara memilih kolom/lingkungan secara acak kemudian *outlier* ditempatkan pada baris/*genotype* yang terpilih secara acak sampai memenuhi seluruh elemen pada kolom/lingkungan tersebut. Pada simulasi ini komposisi *outlier* dibangkitkan sebanyak 5%, 10%, dan 15% dari data.

3.2. Analisis Data

Analisis data dilakukan dengan bantuan software R studio versi 8.1 Data tersebut akan dimodelkan dan dibandingkan melalui dua model, yaitu Model RCIM dan Model *Robust* Faktor. Adapun langkah-langkah pemodelan yang dilakukan adalah sebagai berikut :

3.2.1. Model RCIM

Model RCIM yang digunakan adalah model RCIM dengan distribusi normal dimana fungsi yang digunakan adalah

```
rcim(data, uninormal, Svd.arg=TRUE, Alpha=0.5, Rank=0, trace=TRUE)
```

Adapun langkah-langkah pemodelan RCIM :

1. Mencari model *Row Column Interaction Model* (RCIM) dengan fungsi rcim. Model multiplikatif diperoleh dengan menggunakan Rank = 1, 2, 3, ... dst pada fungsi tersebut.

2. Mencari nilai log-*Likelihood* dan nilai MSE pada tiap model *full* rank pada data awal tanpa *outlier*.
3. Mencari nilai log-*Likelihood* dan nilai MSE pada tiap model *full* rank di setiap data dengan modifikasi *outlier*.
4. Interpretasi biplot pada data tanpa *outlier* dan data dengan modifikasi *Outlier* dengan menggunakan *Robust* SVD para rank=0.
5. Membandingkan nilai log-*Likelihood*, pada data tanpa *outlier* terhadap data dengan *outlier*.

3.2.2. Model *Robust* Faktor

Pada model *robust* faktor terdapat tiga algoritma yang dipakai, yaitu, Algoritma `Weight.wl1`, RobPCA, dan algoritma `Twoway.rob`. Adapun langkah pemodelan dengan *robust* factor :

1. Mendapatkan bobot dari masing-masing baris maupun kolom dari data dengan menggunakan regresi L1 terboboti (weight L1 regression). Fungsi yang digunakan adalah `cov.rob(data)`
2. Memodelkan data ke dalam tabel dua arah yang *robust* terhadap pencilan dengan algoritma `Twoway.rob` dimana langkah-langkah yang dilakukan sebagai berikut :
   - Mendefinisikan panjang baris dan kolom data yang dimasukkan
   - Mendefinisikan nama untuk baris dan nama untuk kolom
   - Jika panjang baris data lebih kecil daripada panjang kolom data, maka dilakukan transpose pada data untuk menukar baris dan kolom data.
   - Mendefinisikan matriks bobot baris, bobot kolom, dan nilai awal fungsi objektif.
   - Mencari nilai invariant terhadap perubahan nilai efek baris data.

- Melakukan prosedur *robust* analisis komonen utama (RobPCA) dengan esimasi berbasis *Projection Pursuit* (PP) dengan menggunakan fungsi `PcaProj` pada paket `rrcov`. Nilai yang dihasilkan dari RobPCA diambil sebagai nilai awal bagi skor.
- Setelah diperoleh nilai konvergensi yang maksimum pada iterasi RobPCA, maka hasil keluaran dari algoritma diantaranya residual, biplot, dan boxplot.

BAB 5. PENUTUP

5.1. Kesimpulan

Dari hasil dan pembahasan, diperoleh kesimpulan sebagai berikut:

1. Semakin banyak komposisi *outlier* pada data tabel dua arah maka model RCIM semakin kompleks dan mampu memodelkan data tersebut. Hal ini terlihat dari penurunan rank pada model tersebut. Model RCIM dengan *outlier* masih mampu mempertahankan model sebaik RCIM2 (Model Rank=2) pada rank-rank tertentu.
2. Pada penempatan *outlier* secara *Scattered* dimana *outlier* ditambahkan secara menyebar, semakin tinggi komposisi *outlier* pada data maka semakin tidak mampu model tersebut memodelkan data dengan *outlier*. Pada penempatan outlier secara S*ingle Environment* dimana *outlier* ditambahkan lingkungan tertentu, semakin tinggi komposisi *outlier* pada data maka semakin mampu model tersebut memodelkan data dengan *outlier*, sehingga dapat disimpulkan bahwa model *Robust* Faktor mampu tahan terhadap *outlier* dengan kondisi beberapa data tertentu saja. Hal ini terlihat dari nilai MSE pada *Scattered* semakin naik ketika komposisi *outlier* bertambah sedangkan pada *Single Environment* nilai MSE semakin kecil ketika komposisi *outlier* bertambah.
3. Model RCIM lebih baik dalam mengepas model data tabel dua arah yang mengandung *outlier* daripada model *Robust* faktor. Hal ini ditunjukkan dengan lebih konsistennya model RCIM daripada model *Robust* Faktor dalam pengepasan model pada saat dilakukan beberapa simulasi data dengan penempatan o*utlier* yang berbeda. Selain itu, nilai MSE pada semua model RCIM lebih kecil dari model *Robust* Faktor.

5.2. Saran

Penambahan *outlier* pada suatu data sedikit banyak mempunyai dampak pada analisis data. Sebaiknya dilakukan analisis lebih lanjut dengan menggunakan dimensi data yang lebih besar dalam analisis RCIM dengan distribusi data selain distribusi normal.

DAFTAR PUSTAKA

Aini, E. N., Hadi, A. F., dan Anggraeni, D. 2015. "Metode Robust Singular Value Decomposition (RSVD) untuk Model AMMI dengan Data Pencilan". *Berkala saintek*. Jember : Universitas Jember.

Ardian, J., Hadi, A. F., dan Anggraeni, D. 2015. "Implementasi Algoritma Model *Robust* Faktor untuk Tabel Dua Arah pada Program R". *Berkala saintek*. Jember : Universitas Jember.

Dias, C. T. S., dan Krzanowski, W. J. 2003. Model selection and cross validation in additive main effect and multiplicative interaction models, *Crop Science*, 43, 169-175.

Eeuwijk, V. F. A. 1995. Multiplicative Interaction in Generalized Linear Models. *Biometrics* 51 : 1017-1032.

Gabriel, K. R. 1971. The Biplot Graphic Display of Matrices with Aplication to principal component analysis. *Journal of Biometrica*. 58, 453-467.

Hadi, A. S., Imon, A. H. M .R., & Werner, M. (2009). Detection of Outliers. *WIREs Computational Statistics*, 1, 57-70.

Hawkins, D. M., Liu, Li., dan Young, S.S. 2001. Robust Singular Value Decomposition. *National Institute of Statistical Science*.Technical ReportNumber 122.

Huber, P. J. 1986. Projection Pursuit. *The Annals of Statistics*. 13, 435-475.

Hubert, Rousseeuw, & Branden. 2005. A Comparison of Three Procedures for *Robust* PCA in High Dimensions. *Austrian Journal of Statistics*, 34, 117-126.

Johnson R., 1992. Applied Multivariate Statistical Analysis. Prentice Hall.

Kriegel, H. P. 2010. *Outlier Detection Techniques*. Munich : University Munchen.

Mattjik, A. A, dan Sumertajaya, I. M. 2011. *Sidik Peubah Ganda dengan Menggunakan SAS*. Edisi Pertama. Bogor: IPB Press.

Rocke, D. M. dan Woodruff, D. L. 1996. Identification of *Outlier*s in multivariate data. *Journal of the American Statistical Association*, 91, 1047-1061.

Rodrigues, P. C., Monteiro, A., & Lourenco, V. M. 2015. A *Robust* AMMI model for the Analysis of Genotipe by Environment Data. *Bioinformatics Advance Access*.

Ryan, T. P. 1997. *Modern Regression Methods*. Canada : John Wiley & Sons,Inc.

Warsito. 2009. *Biplot Dengan Dekomposisi Nilai Singular Biasa Dan Kekar Untuk Pemetaan Provinsi Berdasarkan Prestasi Mahasiswa IPB*. Tidak Diterbitkan. Tesis. Bogor: Program Pasca Sarjana Institut Pertanian Bogor.

Yee. T. W., & Hastie, T. J. 2003. Reduced-rank vector generalized linear models. *Stat Model* 3: 15-41.

Yee, T. W., & Hadi, A. F. (2014). Row–column interaction models, with an R implementation. *Computational Statistics, 29* (6), 1427-1445.

Yee, T. W. 2015. Vector Generalized Linear and Additive Model. New York : Springer

LAMPIRAN A. ANALISIS DEVIAN DATA AWAL

Analisis Devian RCIM1

MODEL	DB	DEVIAN	RATAAN DEVIAN	RASIO RATAAN DEVIAN	P-VALUE
lokasi	7	775.9	110.8428571	9.903407527	2.53E-10
gen	49	774.094	15.79783673	1.411479452	0.04610965
RCIM1	55	796.356	14.4792	1.293664039	0.09468617
residual	288	3223.41	11.19239583		

Analisis Devian RCIM2

MODEL	DB	DEVIAN	RATAAN DEVIAN	RASIO RATAAN DEVIAN	P-VALUE
lokasi	7	775.9	110.8428571	10.70621509	2.11E-10
gen	49	774.094	15.79783673	1.525899299	0.02129436
RCIM1	55	796.356	14.4792	1.398533325	0.04721334
RCIM2	53	790.424	14.91366038	1.440497475	0.03612638
residual	235	2432.986	10.35313191		

Analisis Devian RCIM3

MODEL	DB	DEVIAN	RATAAN DEVIAN	RASIO RATAAN DEVIAN	P-VALUE
lokasi	7	775.9	110.8428571	11.14893743	2.10E-10
gen	49	774.094	15.79783673	1.588998135	0.01547443
RCIM1	55	796.356	14.4792	1.456365336	0.034712
RCIM2	53	790.424	14.91366038	1.500064783	0.02643246
RCIM3	51	603.656	11.83639216	1.190543071	0.2041658
residual	184	1829.33	9.94201087		

LAMPIRAN B. PERBANDINGAN NILAI LOG-*LIKELIHOOD* DATA AWAL DENGAN DATA *OUTLIER* DENGAN PENEMPATAN SECARA *SCATTERED*

JENIS *OUTLIER*	MODEL	No *Outlier*	*SCATTERED*					
			k=4			k=10		
			2%	5%	10%	2%	5%	10%
	null	-1293.063	-1338.87	-1386.18	-1444.94	-1336.42	-1386.82	-1437.06
	baris	-1256.016	-1309.2	-1358.69	-1424.51	-1306.93	-1359.32	-1416.82
	kolom	-1145.113	-1237.87	-1316.73	-1399.18	-1233.52	-1320.29	-1390.15
	Rank 0	-1095.194	-1218.15	-1308.65	-1402.91	-1213.32	-1312.62	-1393.39
	Rank 1	-710.016	-1038.48	-810.467	-1215.82	-1017.58	-773.674	-1213.19
Pure Shift Outlier	Rank 2	-201.844	-680.173	-810.753	-1096.28	-829.102	-709.279	-1027.06
	Rank 3	52.130	-538.122	-437.911	-612.247	-558.953	-488.99	-587.758
	Rank 4	510.206	-175.986	-199.188	-426.789	-29.8546	-231.949	-356.445
	Rank 5	1318.320	225.9196	-48.4321	-290.333	78.20795	-12.5309	-190.948
	Rank 6	1443.205	547.3791	250.044	126.2594	460.0764	281.6163	153.8383
	Rank 7	1874.689	806.2054	508.2333	326.3908	759.587	486.8921	364.0556
	null		-1339.67	-1385.85	-1442.86	-1335.87	-1384.49	-1437.9
	baris		-1310.07	-1358.09	-1422.45	-1306.22	-1356.82	-1417.72
	kolom		-1239.18	-1316.14	-1396.51	-1232.68	-1315.83	-1391.21
	Rank 0		-1219.7	-1307.9	-1400.07	-1212.25	-1307.65	-1394.52
	Rank 1		-979.564	-1123.76	-1465.04	-1086.82	-919.537	-1127.47
Shift Point Mass Outlier	Rank 2		-782.478	-925.765	-944.621	-773.052	-665.56	-1104.31
	Rank 3		-373.933	-453.161	-835.895	-450.047	-416.76	-772.636
	Rank 4		-154.577	-467.61	-399.914	-183.193	-311.663	-371.272
	Rank 5		226.0287	-59.5137	-274.825	140.9769	-78.3366	-206.901
	Rank 6		533.9327	272.6247	82.95489	463.9365	362.2678	100.3888
	Rank 7		769.9002	577.6343	348.2562	802.0456	490.4168	348.072

JENIS *OUTLIER*	MODEL	No *Outlier*	***SCATTERED***		
			Tanpa Pengaruh *k*		
			2%	**5%**	**10%**
Pure Shift Outlier	null	-1293.063	-1351.3	-1407.78	-1474.47
	baris	-1256.016	-1322.64	-1381.27	-1455.63
	kolom	-1145.113	-1257.88	-1347.09	-1435.78
	Rank 0	-1095.194	-1241.36	-1341.83	-1442.04
	Rank 1	-710.016	-1282.82	-817.145	-1239.74
	Rank 2	-201.844	-535.043	-808.962	-1297.93
	Rank 3	52.130	-687.659	-589.454	-640.457
	Rank 4	510.206	-141.039	-318.691	-665.782
	Rank 5	1318.320	271.1503	-53.6062	-287.454
	Rank 6	1443.205	533.9327	237.5721	115.1806
	Rank 7	1874.689	769.9002	395.1707	297.2196

LAMPIRAN C. PERBANDINGAN NILAI LOG-*LIKELIHOOD* DATA AWAL DENGAN DATA *OUTLIER* DENGAN PENEMPATAN SECARA *SINGLE ENVIRONMENT*

JENIS *OUTLIER*	MODEL	No *Outlier*	*SINGLE ENVIRONTMENT*					
			k=4			k=10		
			5%	10%	15%	5%	10%	15%
Pure Shift Outlier	null	-1293.063	-1388.39	-1442.58	-1462.46	-1392.01	-1440.55	-1461.37
	baris	-1256.016	-1361.52	-1420.4	-1445.29	-1358.97	-1420.03	-1453.79
	kolom	-1145.113	-1210.22	-1186.19	-1139.21	-1281.45	-1254.86	-1096.82
	Rank 0	-1095.194	-1184.46	-1145.71	-1076.93	-1299.68	-1245.56	-1061.6
	Rank 1	-710.016	-653.235	-866.654	-857.899	-988.793	-979.209	-759.654
	Rank 2	-201.844	-408.935	-625.879	-496.946	-591.398	-554.353	-561.266
	Rank 3	52.130	-216.701	-245.383	-171.556	96.77387	-477.748	-222.263
	Rank 4	510.206	235.958	202.475	380.1064	635.974	976.5453	187.5273
	Rank 5	1318.320	739.335	506.0237	532.8337	754.4617	1025.081	1075.324
	Rank 6	1443.205	1303.957	690.2813	1307.912	1188.854	678.455	1172.757
	Rank 7	1874.689	1281.127	1199.682	1580.763	1506.219	1562.892	1799.265
Shift Point Mass Outlier	null		-1388.96	-1444.34	-1464.66	-1390.07	-1441.38	-1461.8
	baris		-1361.97	-1422.06	-1447.26	-1357.01	-1420.76	-1454.33
	kolom		-1210.57	-1186.17	-1136.78	-1279.2	-1254.84	-1094.43
	Rank 0		-1184.84	-1145.81	-1070.78	-1296.54	-1247.03	-1058.91
	Rank 1		-653.344	-687.989	-801.664	-990.005	-982.883	-742.297
	Rank 2		-365.842	-602.191	-474.771	-606.187	-530.782	-645.972
	Rank 3		-227.827	-317.571	-163.372	-49.6726	-461.28	-193.959
	Rank 4		212.4024	305.0485	695.7448	565.4766	1041.291	631.2192
	Rank 5		756.2569	359.8705	690.396	776.5348	521.2173	520.73
	Rank 6		1083.163	810.6183	1315.909	1448.688	762.5987	1126.084
	Rank 7		1188.038	1197.775	1685.331	1197.419	1633.763	1932.457

JENIS *OUTLIER*	MODEL	No *Outlier*	*SINGLE ENVIRONMENT*		
			Tanpa Pengaruh *k*		
			5%	10%	15%
Pure Shift Outlier	null	-1293.063	-1412.49	-1477.23	-1502.59
	baris	-1256.016	-1387.08	-1457.32	-1488.27
	kolom	-1145.113	-1235.72	-1204.58	-1135.42
	Rank 0	-1095.194	-1221.05	-1171.08	-1068.1
	Rank 1	-710.016	-669.472	-783.667	-772.153
	Rank 2	-201.844	-441.891	-649.718	-464.19
	Rank 3	52.130	138.6001	-268.639	-137.993
	Rank 4	510.206	251.0658	-123.592	373.3047
	Rank 5	1318.320	747.5191	567.5379	773.3142
	Rank 6	1443.205	1178.485	1085.022	1104.042
	Rank 7	1874.689	1262.581	1285.308	1618.958

LAMPIRAN D. SYNTAX SIMULASI DATA

```
library(RobRSVD)
library(readr)
```

Membentuk Data Tabel Dua Arah dengan Model RCIM2

```
set.seed(5216)
baris = 50
kolom = 8
x = runif(baris*kolom, min = -0.5, max = 0.5)
mx = matrix(x, baris, kolom)
SVDx = svd(mx)
d = SVDx$d
U = SVDx$u
V = SVDx$v

mu = rnorm(n = 1, mean = 15, sd = 3)
alpha = rnorm(baris, mean = 5, sd = 1)
beta = rnorm(kolom, mean = 8, sd = 2)

satuI = rep(1,baris)
satuJ = rep(1,kolom)

dim(satuI%*%t(satuJ))

y = satuI%*%t(satuJ)*mu + alpha%*%t(satuJ) +
  satuI%*%t(beta) + 28*U[,1]%*%t(V[,1])*d[1] +
  satuI%*%t(beta) + 15*U[,2]%*%t(V[,2])*d[2]

y = as.vector(y)
gen = seq(1,baris)
gen = rep(paste("Gen",formatC(gen, width = 3, flag = 0),sep = "."),
kolom)
env = c()
for (i in 1:kolom) {
  env = c(env, rep(paste("Env",formatC(i, width = 3, flag = 0),sep =
"."), baris))
}

# simpan boxplot Y
bxplt = "Boxplot Data Belum Terkontaminasi"
jpeg(paste(bxplt,".jpg", sep = ""))
boxplot(y, main = bxplt)
dev.off()

df.sim = data.frame(geno = gen, envi = env, yield = y)
df.sim2 = tapply(df.sim$yield, list(df.sim$geno,df.sim$envi), mean)
```

Membangkitkan Nilai Outlier

```
meanx = apply(df.sim2,2,mean)
varx = apply(df.sim2,2,sd)

## SIMULAI I

osk1 = c(8,20,40) # outliernya
k = c(4,10)

set.seed(9832)

# jenis outlier
outlierx = "pso"
df.svd = data.frame(outlier = c(), noo = c(), k = c(), eigen_1 = c(),
eigen_2 = c(), Total = c())
rank_awal = 0
rank_akhir = 0

# tentukan outlier
hasil = data.frame(outlier = c(), noo = c(), k = c(), Model = c(),
logLik = c(), MSE = c(), AIC = c(), BIC = c())
for (i in 1:length(osk1)){
  no = osk1[i]
  set.seed(4367)
  randGen = sample(1:baris, no, replace = F)
  randEnv = rep(sample(1:kolom, kolom, replace = F),
ceiling(max(osk1)/kolom))
  print(paste("simulasi ",no, sep = ""))

  ppmox = 0
  for (j in 1:length(k)) {
    ppmox = ppmox+1
    kx = k[j]
    pso = c()
    ppmo = c()
    spmo = c()
    datax_pso = df.sim2
    datax_ppmo = df.sim2
    datax_spmo = df.sim2

    for (l in 1:no){
      s.meanx = sample(meanx,1)
      s.varx = sample(varx,1)

      randG = randGen[l]
      randE = randEnv[l]

      # pure shift outlier
      outlier1 = rnorm(1, mean = (s.meanx + k*s.varx), sd = s.varx)
      pso = c(pso, outlier1)
      datax_pso[randG, randE] = outlier1

      # pure point-mass outlier
```

```
      outlier2 = rnorm(1, mean = s.meanx, s.varx/100)
      ppmo = c(ppmo, outlier2)
      datax_ppmo[randG, randE] = outlier2

      # shift-point-mass outlier
      outlier3 = rnorm(1, mean = (s.meanx + k*s.varx), sd = 
s.varx/100)
      spmo = c(spmo, outlier3)
      datax_spmo[randG, randE] = outlier3

    }
    name1 = paste("pso_",kx,"_",no,sep = "")
    name2 = paste("ppmo_",kx,"_",no,sep = "")
    name3 = paste("spmo_",kx,"_",no,sep = "")
    assign(name1, pso)
    assign(name2, ppmo)
    assign(name3, spmo)

    dname = paste("Data_pso",no,kx,sep = "_")
    assign(dname, datax_pso)
    df.sim[dname] = as.vector(datax_pso)

    dname = paste("Data_ppmo",no,kx,sep = "_")
    assign(dname, datax_ppmo)
    df.sim[dname] = as.vector(datax_ppmo)

    dname = paste("Data_spmo",no,kx,sep = "_")
    assign(dname, datax_spmo)
    df.sim[dname] = as.vector(datax_spmo)

    for (dx in 1:3) {
      print(c(dx,ppmox))
      if (dx == 1) {
        datax = datax_pso
        namxx = name1
        ooo = paste("pure shift outlier")
        bxplt = paste("Data dengan Pure Shift Outlier \n",no," 
Outlier, k = ",k," - Skenario I", sep="")
        bxplt2 = paste("Data dengan Pure Shift Outlier ",no," 
Outlier, k = ",k," - Skenario I", sep="")
        jpeg(paste(bxplt2,".jpg", sep = ""))
        boxplot(as.vector(datax), main = bxplt)
        dev.off()
      } else if (dx == 2 & ppmox == 1) {
        datax = datax_ppmo
        namxx = name2
        ooo = paste("pure point-mass outlier")
        bxplt = paste("Data dengan Pure Point-Mass Outlier \n",no," 
Outlier - Skenario I", sep="")
        bxplt2 = paste("Data dengan Pure Point-Mass Outlier ",no," 
Outlier - Skenario I", sep="")
        jpeg(paste(bxplt2,".jpg", sep = ""))
```

```
        boxplot(as.vector(datax), main = bxplt)
        dev.off()
      } else if (dx == 3) {
        datax = datax_spmo
        namxx = name3
        ooo = paste("shift-point-mass outlier")
        bxplt = paste("Data dengan Shift-Point-Mass Outlier \n",no,"
Outlier, k = ",k," - Skenario I", sep="")
        bxplt2 = paste("Data dengan Shift-Point-Mass Outlier ",no,"
Outlier, k = ",k," - Skenario I", sep="")
        jpeg(paste(bxplt2,".jpg", sep = ""))
        boxplot(as.vector(datax), main = bxplt)
        dev.off()
      } else {
        next
      }

      dat.xy.2a = datax
      ax = rownames(dat.xy.2a)
      by = colnames(dat.xy.2a)
      la = length(ax)
      lb = length(by)

      ma = c()
      for (ix in 1:(lb)) {
        ma=c(ma,ax)
      }

      mb = c()
      for (ix in 1:lb) {
        for (j in 1:la) {
          mb = c(mb,by[ix])
        }
      }

      datax = dat.xy.2a
      baris = dim(dat.xy.2a)[1]
      kolom = dim(dat.xy.2a)[2]


      # wadah untuk menyimpan hasilnya

      for (n in rank_awal:rank_akhir) {
        print(paste("Mengerjakan model dengan rank",n, "pada data",
namxx, sep = " "))
```

LAMPIRAN E. SYNTAX MODEL RCIM DAN *ROBUST* FAKTOR

```
##MEMODELKAN DENGAN RCIM

library(VGAM)
modelfull=vglm(yield~1,uninormal,data=df.sim)
modelgen=vglm(yield~geno,uninormal,data=df.sim)
modellokasi=vglm(yield~envi,uninormal,data=df.sim)

a=rcim(df.sim2, uninormal, Svd.arg=TRUE, Rank=0, wzepsilon=4,
trace=TRUE)
b=rcim(df.sim2, uninormal, Svd.arg=TRUE, Rank=1, wzepsilon=4,
trace=TRUE)
c=rcim(df.sim2, uninormal, Svd.arg=TRUE, Rank=2, wzepsilon=4,
trace=TRUE)
d=rcim(df.sim2, uninormal, Svd.arg=TRUE, Rank=3, wzepsilon=4,
trace=TRUE)
e=rcim(df.sim2, uninormal, Svd.arg=TRUE, Rank=4, wzepsilon=4,
trace=TRUE)
f=rcim(df.sim2, uninormal, Svd.arg=TRUE, Rank=5, wzepsilon=4,
trace=TRUE)
g=rcim(df.sim2, uninormal, Svd.arg=TRUE, Rank=6, wzepsilon=4,
trace=TRUE)
h=rcim(df.sim2, uninormal, Svd.arg=TRUE, Rank=7, wzepsilon=4,
trace=TRUE)

##PENGURAIAN  NILAI SINGULAR DENGAN ROBUST SVD
library(RobRSVD)
bbb=a@residuals
a1=bbb[,1]+bbb[,2]
a2=bbb[,3]+bbb[,4]
a3=bbb[,5]+bbb[,6]
a4=bbb[,7]+bbb[,8]
a5=bbb[,9]+bbb[,10]
a6=bbb[,11]+bbb[,12]
a7=bbb[,13]+bbb[,14]
a8=bbb[,15]+bbb[,16]
zzz=matrix(c(a1,a2,a3,a4,a5,a6,a7,a8),10)
colnames(zzz)=c("L1","L2","L3","L4","L5","L6","L7","L8")
rownames(zzz)=c("G1","G2","G3","G4","G5","G6","G7","G8","G9","G10")
svd.a=RobRSVD(zzz,irobust=TRUE)
svd.a
l=svd.a$s
aa=svd.a$u
bb=svd.a$v
bt=t(bb)
X=zzz-(l*aa%*%bt)
z=svd(X)
A=z$v[,c(1,2)]## Col
U=z$u[,c(1,2)] ## Row
cc=rbind(A,U)
```

```
row.names(A)=row.names(t(data_outlier_baris))
row.names(U)=row.names(data_outlier_baris)

##BIPLOT
plot(1, type = "n", xlim = range(cc[,1]),
  ylim = range(cc[,2]), xlab = "KUI 1", ylab = "KUI 2",
  main = "Biplot RCIM normal with outlier (OUTLIER BARIS)")
  abline(h = 0)
  abline(v = 0)
  points(A[, 1], A[, 2], type = "p", col = "red")
  text(A[, 1], A[, 2], labels = row.names(A),
  adj = c(0.5, 0.5), col = "red")
  for (i in 1:15) {
  x2 <- U[, 1]
  y2 <- U[, 2]
  x21 <- c(0, x2[i])
  y21 <- c(0, y2[i])
  points(x21, y21, type = "l", col = "blue")
  text(U[, 1], U[, 2], labels = row.names(U))
  }

##MEMODELKAN DENGAN ROBUST FAKTOR

library(MASS)
library(pracma)
library(rrcov)
library(pcaPP)
library(robustbase)
library(mvtnorm)
```

Menghitung Nilai Bobot pada Data

```
x = df.sim2
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 <- mahalanobis(x, rob$center, rob$cov)
weight <- qchisq(0.95, k)/robdist2
weight <- apply(cbind(1, weight), 1, min)
weight
}
```

Menghitung Nilai Robust Principal Component Analysis

```
Pca<-PcaProj(x, k = 2,kmax = ncol(x),scale=TRUE, trace=FALSE)
```

```
f<-attr(Pca,"scores")
f
```

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]

	var_kolom = colnames(x)
	var_baris = rownames(x)

	if(!length(dimnames(x)[[2]]))
# dimnames(x)[[1]]<- var_baris
# dimnames(x) <- list(var_baris, var_kolom)
	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","wl1fit","lmsreg","ltsreg","rreg"), nomatch
= NA)
if(!is.na(imethod))
method <- c("lsfit", "l1fit", "wl1fit", "lmsreg", "ltsreg",
"rreg")[imethod]
if(method == "wl1fit") {
if((n <= 2 * k) | (p <= 2 * k)) {
stop("\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
Pca<-PcaProj(x, k = 2,kmax = ncol(x),scale=TRUE, trace=FALSE)
rf<-attr(Pca,"scores")
# starting with classical PCA
f <- princomp(x)$scores[, 1:k]
# starting with random values
#      f <- matrix(rnorm(n * k, 0, 1), ncol = k)
if(k == 1) {
# k is the dimension of the multiplicatice terms
f <- f - median(f)
sigm <- sqrt(sum(f^2))
f <- f/sigm
}
else {
f <- f - vecn1 %*% t(apply(f, 2, median))
sigm <- sqrt(apply(f^2, 2, sum))
f <- t(t(f)/sigm) #
# sorting by magnitude
f <- f[, rev(sort.list(sigm))]
sigm <- rev(sort(sigm))
}
sigmnew <- sigm
sigmf <- matrix(0, n, k)
sigml <- matrix(0, p, k)
}
# begin of the iteration
for(it in 1:iter) {
cat("Iteration    ", it, "\n")
# regression of the columns
for(j in 1:p) {
if(method == "rreg") {
sigmlreg <- switch(model,
      a = get(method)(vecn1, (x[, j] - mu -
            a), int= F),
      m = get(method)(f, x[, j], int = F),
      b = get(method)(f, (x[, j] - mu - a)))
weightc1 <- sigmlreg$w
sigmlreg <- sigmlreg$coef
}
if(method == "wl1fit") {
if(j == 1) {
      if(model == "a")
            weightc1 <- vecn1
```

```
      else weightc1 <- weight.wl1(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 == "wl1fit") || (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 parameternestimates 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 <- sigml - hl
            sigm <- sqrt(sum(l^2))
            l <- l/sigm
            a <- as.vector(a + f * hl)
      }
      else {
            hl <- apply(sigml, 2, median)
            l <- sigml - vecp1 %*% t(hl)
            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 %*% hl)
      }
      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 == "wl1fit")
{ if(i == 1) {
if(model == "a")
weightr1 <- vecp1
else weightr1 <- weight.wl1(l, 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 == "wl1fit") || (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 + l %*% hf)
}
ma <- median(a)
mb <- median(b)
a <- a - ma
b <- b - mb
mu <- ms + ma + mb
}
# calculate the residuals
if(method == "wl1fit") {
resid <- switch(model,
a = x - mu - a %*% t(vecp1) - vecn1 %*% t(b),
m = x - f %*% diag(sigmnew) %*% t(l),
b = x - mu - a %*% t(vecp1) - vecn1 %*%
t(b) - f %*% diag(sigmnew) %*% t(l))
}
else {
resid <- switch(model,
a = x - mu - a %*% t(vecp1) - vecn1 %*% t(b),
m = x - f %*% diag(sigmnew) %*% t(l),
b = x - mu - a %*% t(vecp1) - vecn1 %*% t(b) - f %*%
diag(sigmnew) %*% t(l))
}
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),
l1fit = sum(abs(resid)),
wl1fit = 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(var_kolom))
names(asav)<-dimnames(x)[[1]]<- var_baris
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(var_kolom))
dimnames(fsav)<-list(dimnames(x)[[1]]<- var_baris, 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, weightcolomn = 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,
weightcolomn = 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(x)
```