NEWTON - RAPHSON METHOD
PARAMETER ESTIMATION

DATA *SQUASHING* UNTUK MODEL REGRESI LOGISTIK

SKRIPSI

DIAN ANDRIANI

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS AIRLANGGA
SURABAYA

2005

DATA *SQUASHING* UNTUK MODEL REGRESI LOGISTIK

SKRIPSI

Sebagai Salah Satu Syarat untuk Memperoleh Gelar Sarjana Sains Bidang Matematika di Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Airlangga

Oleh :

DIAN ANDRIANI
NIM : 080112269

Tanggal Lulus : 18 Agustus 2005

Disetujui Oleh :

Pembimbing I	Pembimbing II
Drs. ETO WURYANTO, DEA	**RIMULJO HENDRADI, S.Si, M.Si**
NIP 131 933 015	**NIP 132 161 178**

LEMBAR PENGESAHAN SKRIPSI

Judul	**: DATA *SQUASHING* UNTUK MODEL REGRESI LOGISTIK**
Penyusun	**: DIAN ANDRIANI**
NIM	**: 080112269**
Tanggal Ujian	**: 18 Agustus 2005**

Disetujui Oleh :

Pembimbing I	Pembimbing II
Drs. ETO WURYANTO, DEA	**RIMULJO HENDRADI, S.Si, M.Si**
NIP 131 933 015	**NIP 132 161 178**

Mengetahui :

Ketua Jurusan Matematika

FMIPA Universitas Airlangga

DEPARTEMEN PENDIDIKAN NASIONAL UNIVERSITAS AIRLANGGA JURUSAN MATEMATIKA MATEMATIKA DAN ILMU PENGETAHUAN ALAM

Drs. MOH. IMAM UTOYO, M.Si

NIP 131 801 397

PEDOMAN PENGGUNAAN SKRIPSI

"Selama masih hidup masalah seberat apapun

pasti ada jalan keluarnya

sebab segala sesuatu yang ada di dunia tiada yang abadi"

"Bersukacitalah dalam pengharapan,
sabarlah dalam kesesakan, dan
bertekunlah dalam doa"

(Roma 12:12)

Thank's To

Ucapan terima kasih saya berikan kepada :

1. Tuhan, atas segala rahmat-Nya
2. Ibu dan Bapak (Alm) serta saudara2ku, atas dukungan & doa2nya
3. Drs. Eto Wuryanto,DEA dan Rimuljo Hendradi, S.Si, M,Si selaku pembimbing I dan pembimbing II yang telah membimbing penulis dalam menyelesaikan skripsi ini.
4. Drs.Ardi Kurniawan, M.Si dan Drs. Suliyanto, M.Si selaku dosen penguji yang telah memberikan saran perbaikan.
5. Herry Suprajitno, S.Si, M.Si selaku dosen wali, makasih buat nasihat2nya
6. Bapak dan Ibu Dosen pengajar di linkungan FMIPA UNAIR, khususnya jurusan matematika.
7. My best friends, Vonny (jangan lupa nasehat dari orang lain n' belajarlah mandiri), Lelly (jangan lama2 kalo' liburan skripsi, kalo' perlu sesuatu dating aja langsung ke rumah), Tria (dah keenakan kerja, jangan lupa ma' skripsinya), Nikhen (selamat dah lulus di ItS, semoga yang Unair juga cepet selesai), Ira.R (kalo' ada yang gak ngerti Tanya ma' dosen aja, selamat dah lulus di ItS), Erni (yang rutin kalo' bimbingan n' jangan putus asa), Dian. R (kita jadi wisuda bareng, nich), Trip (cepet nyusul yang laen ngerjain skripsi). Makasih semuanya 'tuk jadi tempat curhat n' buat persahabatan qta selama ini !!

8. Teman2 '01, Nina, Ida, Seha, Dewi (q dah mo' nyusul wisuda), Sugeng, Hamzah, Juned (kapan nyusul ?), Zaenal (cepetan maju), Noor Ira (makasih atas jawaban2nya), Nita.M, Raras (jangan menyerah !), Dina (undangan q tunggu di rumah), Ribut (rajin bimbingan, ya), Daniel (makasih doanya), Henry (makasih buat backup-an cd n' skripsinya mulai dikerjain), Agus (skripsi dah selesai, cepetan lulus), Trisni (jangan lupa ngerjain skripsi), Zum', Arif"S", Didin, Dhani, Hanafi, Yudi, Ipung, 'tuk Semuanya sukses, ya!!
9. Adekq Tika 'bio (makasih buat semua perhatiannya), Anna 'bio (selalu ceria, ya), Heppy 'bio (makasih nasihatnya), Kury 'bio (kapan2 maen ke rumah,,ya), Nike 'kimia (makasih doanya), Mba' Tere ' fisika (makasih doanya)
10. Angkatan '98, '99, '00, makasih buat pinjeman buku2nya n' dukungannya, angkatan '02 makasih dukungannya
11. Mas Edi, Mas Milan , dan Mas Udin, makasih 'tuk kelancaran administrasi

PRAKATA

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas berkat dan anugerah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul "Data *Squashing* untuk Model Regresi Logistik".

Dalam kesempatan ini penulis mengucapkan terima kasih kepada semua pihak yang telah membantu dalam penyusunan skripsi ini hingga selesai, khususnya kepada Drs. Eto Wuryanto, DEA dan Rimuljo Hendradi, S.Si, M.Si. selaku dosen pembimbing serta kepada Herry Suprajitno, S.Si, M.Si selaku dosen wali yang senantiasa memberikan bimbingan kepada penulis selama penyusunan skripsi ini serta kepada rekan-rekan yang telah memberikan dorongan.

Akhir kata penulis mengharapkan, semoga skripsi ini dapat memberikan sesuatu yang bermanfaat bagi pembaca pada umumnya dan jurusan matematika pada khususnya.

Surabaya, Agustus 2005

Penulis

Dian Andriani, 2005. *Data Squashing untuk Model Regresi Logistik.* Skripsi ini dibawah bimbingan Drs. Eto Wuryanto, DEA dan Rimuljo Hendradi, S.Si, M.Si. Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam. Universitas Airlangga.

ABSTRAK

Skripsi ini bertujuan untuk mereduksi data yang sangat besar menjadi data yang lebih kecil dengan metode *squashing* dan menerapkannya pada model regresi logistik. Untuk mendapatkan data *squashing,* data dikelompokkan dengan metode kuantil. Kemudian dihitung nilai pengamatan baru serta pembobot untuk tiap kelompok. Dalam mengestimasi parameter model regresi logistik (β) digunakan metode *Maximum Likelihood Estimation* (MLE). Secara analitik untuk mengestimasi parameter model regresi logistik dengan metode MLE tidak didapatkan penyelesaian dari parameter β karena berbentuk fungsi implisit, sehingga diperlukan suatu metode numerik yaitu Newton-Raphson. Nilai awal parameter yang digunakan dalam metode Newton-Raphson adalah estimator parameter regresi linier.

Untuk mempermudah perhitungan digunakan program S-plus. Setelah dilakukan penerapan pada data bangkitan sebanyak 100000 data, nilai estimator untuk model regresi logistik diperoleh $\hat{\beta}_0 = 0.01794032$ dan $\hat{\beta}_1 = -0.04712208$. Sedangkan untuk nilai estimator pada data *squashing* dengan 2000 kelompok adalah $\hat{\beta}_0 = 0.008707499$ dan $\hat{\beta}_1 = -0.0006972267$, untuk data *squashing* dengan 5000 kelompok adalah sebesar $\hat{\beta}_0 = 0.009077274$ dan $\hat{\beta}_1 = -0.001817088$, untuk data *squashing* dengan 10000 kelompok adalah sebesar $\hat{\beta}_0 = 0.009540798$ dan $\hat{\beta}_1 = -0.00381976$, dan untuk data *squashing* dengan 20000 kelompok adalah sebesar $\hat{\beta}_0 = 0.0117504$ dan $\hat{\beta}_1 = -0.009408836$.

Kata kunci : Maximum Likelihood; Newton-Raphson; *Squashing*; Model Regresi Logistik

Dian Andriani, 2005. *Data Squashing for Logistic Regression Model*. This *Skripsi* under guidence of Drs. Eto Wuryanto, DEA and Rimuljo Hendradi, S.Si, M.Si. Departement of Mathematics, Faculty of Mathematics and Natural Science Airlangga University.

ABSTRACT

The purpose of this *Skripsi* is to reduce the large data to a smaller one with *squashing* method and apply it to the logistic regression model. Data *squashing* is obtained by grouping the large data in regions with quantile method. Then, calculated pseudo data point and weight for each regions. Estimation of parameter β in logistic regression model use *Maximum Likelihood Estimation* (MLE) method. In fact, the result of the MLE method is an implicit function, so a numeric method is needed that is the Newton-Raphson method. But this method must be use the initial value of parameter, in this case estimator of linear regression is used.

Implementation of logistic regression model use S-plus program. Generated data is used in this implementation. The estimator values of logistic regression for generated data are $\hat{\beta}_0 = 0.01794032$ and $\hat{\beta}_1 = -0.04712208$. While the estimator values for data *squashing* with 2000 regions are $\hat{\beta}_0 = 0.008707499$ and $\hat{\beta}_1 = -0.0006972267$, the estimator values for data *squashing* with 5000 regions are $\hat{\beta}_0 = 0.009077274$ and $\hat{\beta}_1 = -0.001817088$, the estimator values for data *squashing* with 10000 regions are $\hat{\beta}_0 = 0.009540798$ and $\hat{\beta}_1 = -0.00381976$, and the estimator values for data *squashing* with 20000 regions are $\hat{\beta}_0 = 0.0117504$ and $\hat{\beta}_1 = -0.009408836$.

Keyword : Maximum Likelihood; Newton-Raphson; Squashing; Logistic Regression Model

DAFTAR ISI

DAFTAR TABEL

DAFTAR LAMPIRAN

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi yang semakin pesat menghasilkan data besar yang memuat ribuan bahkan jutaan pengamatan dan jumlahnya akan mengalami kenaikan seiring bertambahnya waktu. Contoh data tersebut adalah data telekomunikasi. Adakalanya data-data tersebut dikumpulkan tanpa maksud atau alasan tertentu. Oleh karena itu, perlu dilakukan suatu teknik pengekstrasian informasi untuk membantu pembuatan keputusan, perencanaan strategis, dan monitoring sistem.

Dalam pengolahan data telekomunikasi yang berkaitan dengan identifikasi dini kepada sikap pelanggan telepon dapat dilakukan dengan model regresi logistik. Model ini dapat digunakan untuk mengetahui seorang pelanggan akan pindah ke provider lain atau tidak. Atau lebih tepatnya, dengan model regresi logistik ini dapat diketahui berapa peluang seorang pelanggan akan pindah ke provider lain.

Karena data telekomunikasi pada dasarnya berkaitan dengan jumlah data yang sangat besar dan adanya kenaikan jumlah data sejalan dengan bergeraknya waktu, maka analisis data dengan model regresi sulit dilaksanakan. Oleh karena itu, digunakan penelitian terbaru oleh **DuMouchel et.al (1999)** yang mengusulkan pendekatan baru yang disebut metode *squashing*. Ide dari metode *squashing* adalah mendapatkan nilai-nilai pengamatan baru (*pseudo data point*) sedemikian hingga nilai momen dari data hasil metode *squashing* (data *squashing*) sama dengan nilai yang dihasilkan oleh data induk.

Berdasarkan kesulitan yang terjadi pada pengolahan data dengan regresi logistik untuk data berukuran besar serta melihat sifat data *squashing* yang cukup menjanjikan maka sangat menarik untuk dilakukan penelitian yang berkaitan dengan data *squashing* dan model regresi logistik. Permasalahan yang timbul adalah memperoleh sub sampel yang dapat diolah dengan model regresi logistik secara mudah dan menghasilkan akurasi sesuai keinginan dalam proses pengambilan keputusan (*inference*). Namun dalam penulisan skripsi ini untuk penerapan model regresi logistik tidak menggunakan data sekunder tetapi menggunakan data bangkitan.

1.2 Rumusan Masalah

Permasalahan yang akan dibahas dalam skripsi ini adalah bagaimana mendapatkan data *squashing* dan mengestimasi parameter model regresi logistik serta menerapkan data *squashing* pada model regresi logistik?

1.3 Tujuan Penelitian

1. Mendapatkan sub sampel dengan metode *squashing* (data *squashing*).
2. Estimasi parameter model regresi logistik.
3. Menerapkan data *squashing* pada model regresi logistik.

1.4 Manfaat Penelitian

Skripsi ini diharapkan dapat memberikan manfaat sebagai berikut :

1. Menambah perbendaharaan ilmu khususnya statistika.
2. Memperluas wawasan tentang regresi logistik dan data *squashing.*

1.5 Batasan Masalah

Permasalahan yang akan dibahas dibatasi hanya pada regresi logistik berskala nominal khususnya biner untuk variabel respon, sedangkan skala rasio untuk satu variabel prediktor.

BAB II

TINJAUAN PUSTAKA

2.1 Analisis Regresi

Sir Francis Galton (1822-1911), seorang antropolog dan ahli meteorologi terkenal dari Inggris yang memperkenalkan istilah regresi dalam pidatonya di depan *Section H of the British Association* di Aberden, 1885, yang dimuat dalam majalah *Nature*, dan juga dalam sebuah makalah "*Regression towards mediocrity in hereditary stature*", yang dimuat dalam Journal of the *Anthropological Institute*,1885.

(Draper dan Smith, 1992)

Analisis regresi merupakan teknik statistik untuk menyelidiki dan membuat model hubungan di antara variabel-variabel.

(Montgomery dan Peck,1992)

Tujuan dari analisis regresi yaitu untuk mendapatkan model terbaik yang menggambarkan hubungan antara variabel respon (variabel tak bebas / variabel dependent) dan variabel bebas (variabel prediktor / variabel independent).

(Hosmer dan Lemeshow, 1989)

Yang dimaksud dengan variabel prediktor ialah variabel yang nilainya dapat ditentukan atau yang nilainya dapat diamati namun tidak dapat dikendalikan. Variabel respon ialah variabel yang nilainya dipengaruhi oleh perubahan-perubahan variabel-variabel prediktor.

(Draper dan Smith, 1992)

2.2 Model Regresi Logistik

Regresi logistik merupakan analisis regresi yang digunakan untuk menggambarkan hubungan antara variabel respon kategorik dengan beberapa variabel bebas. Regresi logistik berawal dari adanya respon biner yaitu 0 dan 1.

(Kleinbaum, 1994)

Jika variabel respon biner, dengan probabilitas π dan $1-\pi$, maka Y variabel random Bernoulli dengan E (Y| x) = π dengan fungsi kepadatan probabilitas (fkp) yang dinyatakan sebagai berikut :

$$f(Y) = \pi^{Y}(1-\pi)^{1-Y} \qquad (2.1)$$

dengan Y = 0,1

sehingga model regresi logistik dapat dinyatakan sebagai berikut :

$$Y = E(Y \mid x) + \varepsilon \qquad (2.2)$$

Karena distribusi dari ε bergantung pada distribusi Bernoulli dari respon Y, maka model regresi logistik lebih baik dinyatakan sebagai berikut :

$$E(Y \mid x) = \pi(x) = \frac{\exp(\beta_0 + \beta_1 x_1 + ... + \beta_k x_k)}{1 + \exp(\beta_0 + \beta_1 x_1 + ... + \beta_k x_k)} \qquad (2.3)$$

Model regresi logistik dari persamaan (2.3) dapat juga dituliskan :

$$E(Y \mid x) = \pi(x) = [1 + \exp(-\beta_0 - \beta_1 x_1 - ... - \beta_k x_k)]^{-1} \qquad (2.4)$$

Karena $E(Y \mid x)$ dapat dinyatakan dengan π, maka dapat dibuat transformasi sebagai berikut :

$$\text{logit}[\pi(x)] = \log_e\left(\frac{\pi(x)}{1 - \pi(x)}\right) \qquad (2.5)$$

atau dari persamaan (2.4) diperoleh :

$$\text{logit}[\pi(x)] = \beta_0 + \beta_1 x_1 + ... + \beta_k x_k \qquad (2.6)$$

dengan $-\infty < \text{logit}[\pi(x)] < \infty$ dan $-\infty < x < \infty$

Transformasi di atas disebut transformasi *logit* dari nilai probabilitas bersyarat kejadian sukses π.

(Neter et.al, 1996)

Jika $x_i = (x_{i0}, x_{i1}, ..., x_{ij})$ menyatakan nilai ke-i dari j variabel bebas untuk i=1, 2, ..., N dan x_{i0} = 1, maka model regrasi logistik dari persamaan (2.3) dapat dinyatakan sebagai berikut :

$$\pi(x_i) = \frac{\exp(\sum_{j=0}^{p} \beta_j x_{ij})}{[1 + \exp(\sum_{j=0}^{p} \beta_j x_{ij})]} \qquad (2.7)$$

Seperti pada persamaan (2.6), maka untuk $x_i = (x_{i0}, x_{i1}, ..., x_{ij})$ didapat fungsi logit sebagai berikut :

$$\text{logit}[\pi(x)] = \sum_j \beta_j x_{ij} \tag{2.8}$$

(Agresti, 1990)

2.3 Maximum Likelihood Estimation (MLE)

Definisi 2.1 : Misalkan X_1, X_2, ..., X_n adalah variabel random dari suatu distribusi dengan fungsi kepadatan probabilitas (fkp) $f(x,\theta)$, untuk $\theta \in \Omega$ dan Ω adalah ruang parameter. Fkp bersama antara X_1, X_2, ..., X_n adalah $f(x_1,\theta)f(x_2,\theta)...f(x_n,\theta)$. Jika fkp bersama tersebut dinyatakan sebagai fungsi terhadap θ maka dinamakan fungsi likelihood yang dinyatakan L atau ditulis :

$$L(\theta; x_1, x_2, ..., x_n) = f(x_1,\theta)f(x_2,\theta)...f(x_n,\theta) \tag{2.9}$$

dengan $\theta \in \Omega$, Ω adalah ruang parameter.

Definisi 2.2 : Jika statistik $\hat{\theta} = t(X_1, X_2, ..., X_n)$ memaksimumkan fungsi likelihood $L(\theta; x_1, x_2, ..., x_n)$; $\theta \in \Omega$, maka statistik $\hat{\theta} = t(X_1, X_2, ..., X_n)$ dinamakan maximum likelihood estimator dari θ.

(Hogg dan Craig, 1995)

2.4 Metode Newton-Raphson

Misalkan $g_1(u_1,u_2,u_3) = 0$, $g_2(u_1,u_2,u_3) = 0$ dan $g_3(u_1,u_2,u_3) = 0$ adalah tiga persamaan dengan u_1,u_2 dan u_3 yang tidak diketahui. Langkah-langkah dalam metode Newton-Raphson, sebagai berikut :

1. Asumsikan $\mathbf{u}^0 = \begin{bmatrix} u_1^{\,0} \\ u_2^{\,0} \\ u_3^{\,0} \end{bmatrix}$ diketahui sebagai solusi awal atau solusi perkiraan

   dari sistem tiga persamaan nonlinier dengan tiga variabel yang tidak diketahui:

$$\begin{cases} g_1(u_1,u_2,u_3)=0 \\ g_2(u_1,u_2,u_3)=0 \\ g_3(u_1,u_2,u_3)=0 \end{cases} \tag{2.10}$$

2. Menentukan jacobian tiga persamaan (2.10) tersebut.

3. Dengan ekspansi Taylor, diperoleh :

$$\underbrace{\begin{bmatrix} \frac{\partial g_1(\mathbf{u}^k)}{\partial u_1} & \frac{\partial g_1(\mathbf{u}^k)}{\partial u_2} & \frac{\partial g_1(\mathbf{u}^k)}{\partial u_3} \\ \frac{\partial g_2(\mathbf{u}^k)}{\partial u_1} & \frac{\partial g_2(\mathbf{u}^k)}{\partial u_2} & \frac{\partial g_2(\mathbf{u}^k)}{\partial u_3} \\ \frac{\partial g_3(\mathbf{u}^k)}{\partial u_1} & \frac{\partial g_3(\mathbf{u}^k)}{\partial u_2} & \frac{\partial g_3(\mathbf{u}^k)}{\partial u_3} \end{bmatrix}}^{\text{Jacobian} J(\mathbf{u}^k)} \overbrace{\begin{bmatrix} h_1^{\,k} \\ h_2^{\,k} \\ h_3^{\,k} \end{bmatrix}}^{\mathbf{h}^k} = \overbrace{-\begin{bmatrix} g_1(\mathbf{u}^k) \\ g_2(\mathbf{u}^k) \\ g_3(\mathbf{u}^k) \end{bmatrix}}^{-\,g(\mathbf{u}^k)} \tag{2.11}$$

   kemudian mencari nilai $\mathbf{h}^k$ dengan :

$$\mathbf{h}^k = -\left[J(\mathbf{u}^k)\right]^{-1} g(\mathbf{u}^k) \tag{2.12}$$

4. Misal perkiraan $\mathbf{u}^k$ diketahui, dimana $\mathbf{u}^k = \{ \mathbf{u}_1^k, \mathbf{u}_2^k, \mathbf{u}_3^k \}$. Untuk $\mathbf{u}^{k+1}$, dilakukan iterasi dimulai dengan k = 0 dan k bertambah satu tiap kali untuk barisan iterasi $\mathbf{u}^1, \mathbf{u}^2, \mathbf{u}^3, \ldots$ sehingga $\mathbf{u}^{k+1}$ dengan

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \mathbf{h}^k \tag{2.13}$$

sebagai perkiraan yang lebih baik dari perkiraan sebelumnya.

5. Menghentikan proses iterasi ketika diperoleh akurasi yang dikehendaki.

(Anonim)

2.5 Invers Moore-Penrose

Salah satu bagian yang penting dalam matriks adalah invers matriks. Misalkan diberikan matriks persegi A. Jika terdapat matriks persegi A^{-1} yang memenuhi hubungan $A^{-1}A = A\,A^{-1} = I$, maka A^{-1} disebut invers dari A.

Suatu matriks persegi mempunyai invers jika matriks tersebut nonsingular. Jika matriks tersebut persegi panjang atau singular, maka tidak ada matriks A^{-1} yang memenuhi kondisi diatas, oleh karena itu didefinisikan invers Moore-Penrose.

Misalkan diberikan sebarang matriks A yang mempunyai elemen-elemen riil atau kompleks terdapat matriks X yang memenuhi empat persamaan matriks $AXA = A$, $XAX = X$, $(AX)^* = AX$, $(XA)^* = XA$ yang kemudian matriks X ini dikenal sebagai invers semu (*pseudoinverse*) atau invers Moore-Penrose dan dilambangkan A^+.

(Anggraeni, R, 2002)

2.6 Metode *Squashing*

Metode *squashing* merupakan suatu metode yang mereduksi data (mengurangi jumlah data) yang berukuran besar menjadi data yang lebih kecil sehingga data tersebut dapat lebih mudah untuk diolah dan mempresentasikan data induk yang berukuran besar atau disebut juga data lengkap baik untuk keperluan penarikan kesimpulan (inferensi) maupun prediksi.

Misalkan data lengkap dapat dituliskan $(A_1, A_2, ..., A_C, X_1, X_2, ..., X_Q)$ dengan $A_1, A_2, ..., A_C$ adalah variabel kategori dan $X_1, X_2, ..., X_Q$ adalah variabel kuantitatif.

Jika $A = A_{ic}$ dan $X = X_{ij}$ di mana i = 1,2, ..., N

j = 1, 2, ..., Q

c = 1, 2, ..., C

sehingga data lengkap tersebut dapat dituliskan dalam bentuk matriks sebagai berikut :

$$A_{NxC} = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1C} \\ A_{21} & A_{22} & \cdots & A_{2C} \\ \cdots & \cdots & \cdots & \cdots \\ A_{N1} & A_{N2} & \cdots & A_{NC} \end{bmatrix} \quad ; \quad X_{NxQ} = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1Q} \\ X_{21} & X_{22} & \cdots & X_{2Q} \\ \cdots & \cdots & \cdots & \cdots \\ X_{N1} & X_{N2} & \cdots & X_{NQ} \end{bmatrix}$$

Jika data *squashing* mempunyai baris M yang jumlahnya jauh lebih kecil dari N, maka dapat dimisalkan :

$B = B_{ic}$ dan $Y = Y_{ij}$ di mana i = 1,2, ..., M

j = 1, 2, ..., Q

c = 1, 2, ..., C

Jika w_i merupakan pembobot untuk setiap i, di mana $\sum_{i=1}^{M} w_i = N$, maka data *squashing* dapat dituliskan dalam bentuk matriks sebagai berikut :

$$B_{MxC} = \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1C} \\ B_{21} & B_{22} & \cdots & B_{2C} \\ \cdots & \cdots & \cdots & \cdots \\ B_{M1} & B_{M2} & \cdots & B_{MC} \end{bmatrix} \quad ; \quad Y_{MxQ} = \begin{bmatrix} Y_{11} & Y_{12} & \cdots & Y_{1Q} \\ Y_{21} & Y_{22} & \cdots & Y_{2Q} \\ \cdots & \cdots & \cdots & \cdots \\ Y_{M1} & Y_{M2} & \cdots & Y_{MQ} \end{bmatrix}$$

Teknik *squashing* terdiri dari tiga tahap seperti berikut :

1. Mengelompokkan data ke dalam beberapa daerah atau partisi.
2. Menghitung nilai momen dari seluruh data yang ada di setiap partisi.
3. Menghitung nilai pengamatan baru (*pseudo data point*) dan pembobot dari setiap partisi sedemikian hingga nilai momen dari data bangkitan yang terboboti sama dengan nilai momen data induk.

(DuMouchel, et.al., 1999)

2.7 Teorema Taylor

Teorema 2.1 : Misalkan fungsi satu variabel f mempunyai turunan order ke-$k+1$ yang kontinu pada interval tertutup I = $[p,q]$. Untuk bilangan $x \in (p,q)$ dan $X \in [p,q]$, Teorema Taylor menyatakan bahwa :

$$f(X) = f(x) + f'(x)(X-x) + \frac{f''}{2!}(X-x)^2 + \ldots + \frac{f^{(k)}(x)}{k!}(X-x)^k + R_k$$

dengan suku sisa

$$R_k = \int_x^X \frac{(X-t)^k}{k!} f^{(k+1)}(t)\, dt$$

Suku sisa tersebut memenuhi :

$$\lim_{X \to x} \frac{R_k}{(X-x)^k} = 0$$

Bentuk ini mengatakan bahwa suku sisa R_k lebih kecil dibandingkan $(X-x)^k$ jika X cukup dekat ke x.

Bukti :

Mulai dengan suku sisa, kemudian dengan integral parsial reduksi turunan terhadap f tersebut, yaitu :

$$R_k = \int_x^X \frac{(X-t)^k}{k!} f^{(k+1)}(t)\, dt = \int_x^X \frac{(X-t)^k}{k!}\, df^{(k)}(t)$$

$$= \frac{(X-t)^k}{k!} f^{(k)}(t) \Big|_{t=x}^{t=X} + \int_x^X f^{(k)}(t) \frac{(X-t)^{k-1}}{(k-1)!}\, dt$$

$$= -\frac{(X-x)^k}{k!} f^{(k)}(x) + \int_x^X f^{(k)}(t) \frac{(X-t)^{k-1}}{(k-1)!}\, dt$$

Demikian seterusnya, turunkan order turunan dari f, sehingga akhirnya diperoleh :

$$R_k = -\frac{(X-x)^k}{k!} f^{(k)}(x) - \ldots - \frac{(X-x)^2}{2!} f''(x) - (X-x)f'(x) + \int_x^X f'(t)dt$$

$$= -\frac{(X-x)^k}{k!} f^{(k)}(x) - \ldots - \frac{(X-x)^2}{2!} f''(x) - (X-x)f'(x) + f(X) - f(x)$$

$$f(X)=f(x)+f'(x)(X-x)+\frac{f''}{2!}(X-x)^2+\ldots+\frac{f^{(k)}(x)}{k!}(X-x)^k+R_k$$

(Budhi, 2001)

Keunikan dari teorema di atas menyatakan bahwa f yang dapat dinyatakan dalam deret pangkat pada x adalah merupakan suatu deret

$$f(x)+f'(x)(X-x)+\frac{f''}{2!}(X-x)^2+\ldots+\frac{f^{(k)}(x)}{k!}(X-x)^k+\ldots$$

dan disebut deret Taylor.

Oleh karena itu, suatu fungsi dapat dinyatakan dalam bentuk deret pangkat

$$\sum_{k=0}^{\infty} g_k (X-x)^k$$

dalam interval tertutup I= $[p,q]$ dan untuk bilangan $x \in (p,q)$ serta $X \in [p,q]$, jika

$$f(X)=\sum_{k=0}^{\infty} g_k (X-x)^k, \quad \text{dengan } -\Re < X-x < \Re$$

untuk setiap X dalam I dan koefisien g_k memenuhi :

$$g_k=\frac{f^{(k)}(x)}{k!} \quad , \quad \text{untuk } k=0,\,1,\,2,\,\ldots$$

(Bradley, 1995)

2.8 Smooth Function

Smooth function adalah fungsi yang terdiferensial tak hingga kali, yang mempunyai derivatif pada semua order yang berhingga. Fungsi C^1 merupakan fungsi kontinu yang mempunyai derivatif, dikatakan pula sebagai fungsi yang terdiferensial secara kontinu. Sedangkan, suatu fungsi dikatakan fungsi C^n untuk $n \geq 1$ adalah fungsi kontinu yang terdiferensial n kali, dengan derivatif ke-n yang kontinu.

2.9 Metode Kuantil

Nilai-nilai yang dibawahnya terdapat sejumlah pecahan atau persentase tertentu dari seluruh pengamatan disebut dengan fraktil atau kuantil. Yang termasuk fraktil atau kuantil adalah persentil, desil dan kuartil. Berikut ini adalah definisi tentang persentil, desil dan kuartil:

2.9.1 Persentil

Definisi 2.3 : Persentil adalah nilai-nilai yang digunakan untuk membagi segugus pengamatan menjadi 100 bagian yang sama. Nilai-nilai itu dilambangkan dengan $P_1, P_2, ..., P_{99}$ bersifat bahwa 1% dari seluruh data terletak dibawah P_1, 2% terletak dibawah $P_2, ...,$ dan 99 % terletak dibawah P_{99}.

2.9.2 Desil

Definisi 2.4 : Desil adalah nilai-nilai yang digunakan untuk membagi segugus pengamatan menjadi 10 bagian yang sama. Nilai-nilai itu dilambangkan dengan $D_1, D_2, \ldots, D_9$ mempunyai sifat bahwa 10 % data jatuh dibawah D_1, 20 % jatuh dibawah $D_2, \ldots$, dan 90 % jatuh dibawah D_9.

2.9.3 Kuartil

Definisi 2.5 : Kuartil adalah nilai-nilai yang digunakan untuk membagi segugus pengamatan menjadi 4 bagian yang sama. Nilai-nilai itu dilambangkan dengan Q_1 , Q_2 dan Q_3 mempunyai sifat bahwa 25 % data jatuh dibawah Q_1, 50 % data jatuh dibawah Q_2 , dan 75 % data jatuh dibawah Q_3.

(Walpole,1995)

2.10 Mean Square Error

Meskipun mungkin belum maksimal, ukuran kebaikan atau kualitas estimator $t(X_1, \ldots, X_n)$ dari parameter θ cukup berguna, hal tersebut dinamakan *Mean Square Error* dari estimator.

Definisi 2.6 : Jika $T = t(X_1, X_2, \ldots, X_n)$ merupakan estimator dari parameter θ, maka $E_\theta\,[(T - \theta)^2]$ didefinisikan sebagai *mean square error* (MSE) dari estimator $T = t(X_1, X_2, \ldots, X_n)$.

(Graybill, et.al, 1963)

Rumus *Mean Square Error* (MSE) didefinsikan sebagai berikut :

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\{\pi(x_i) - \hat{\pi}(x_i)\}^2$$

2.11 S - Plus

Dalam (**Everitt, 1994**) disebutkan bahwa S-Plus adalah suatu paket program yang memungkinkan membuat program sendiri walaupun didalamnya sudah tersedia banyak program internal yang siap digunakan. Kelebihan dari paket program ini adalah baik program internal maupun program yang pernah dibuat digunakan sebagai sub program dari program yang akan dibuat.

Beberapa perintah internal yang digunakan dalam S – Plus

a. function(...)

   function(...) digunakan untuk menunjukkan fungsi yang akan digunakan dalam program

   Bentuknya adalah : function(...)

b. length(...)

   length(...) merupakan perintah untuk menunjukkan banyaknya data

   Bentuknya adalah : length(...)

c. sort(...)

   Untuk mengurutkan data dari terkecil sampai ke yang terbesar.

   Bentuknya adalah : sort (...)

d. y[order(x)]

   Untuk mengurutkan data y berdasarkan nilai x

Bentuknya adalah : ...[order(...)]

e. matrix(a, b, c)

Untuk membentuk sebuah matrik yang anggotanya a dengan jumlah baris sebanyak b dan jumlah kolom sebanyak c.

Bentuknya adalah : matrix (..., ..., ...)

f. rep(a,b)

Untuk membentuk sebuah vektor yang anggotanya a sebanyak b

Bentuknya adalah : rep(..., ...)

g. for (i in 1:n)

Untuk melakukan perulangan sebanyak n kali.

Bentuknya adalah : for (… in …:…)

h. abs(…)

Untuk membuat harga mutlak dari suatu bilangan

Bentuknya adalah : abs(…)

i. sum(…)

Untuk menjumlahkan semua bilangan anggota dari suatu vektor.

Bentuknya adalah : sum(…)

j. runif(n,m,s)

Untuk menggenerate data berdistribusi uniform sebanyak n, dengan nilai minimum m dan nilai maksimum s.

Bentuknya adalah : runif(..., ..., ...)

BAB III

METODE PENULISAN

1. Mendapatkan sub sampel dengan metode *squashing* (data *squashing*):
   a. Bangkitkan data dari model regresi logistik
   b. Mengelompokkan data dari model regresi logistik
   c. Menghitung momen (*mean*)dari tiap kelompok
   d. Hitung *pseudo data point* dan pembobotnya
   e. Mendapatkan data *squashing* dengan mengalikan *pseudo data point* dan pembobotnya.
2. Estimasi parameter model regresi logistik
   a. Menentukan fungsi likelihood pada model regresi logistik
   b. Me-ln-kan fungsi likelihood
   c. Mendiferensialkan fungsi ln likelihood terhadap parameter-parameter pada model regresi logistik
   d. Mendapatkan estimator dari parameter-parameter model regresi logistik
   e. Jika persamaan implisit, pendugaan parameter menggunakan iterasi Newton-Raphson
3. Menyusun algoritma berdasarkan langkah-langkah yang telah dibuat
4. Menerapkan algoritma yang telah disusun ke dalam program komputer (dengan software S-plus)
5. Menerapkan program pada data bangkitan

BAB IV

HASIL DAN PEMBAHASAN

4.1 Data *Squashing*

4.1.1 Pengertian Data *Squashing*

Suatu data lengkap dapat dinyatakan (A, X) dan bentuknya berupa suatu matriks seperti yang telah dijelaskan dalam sub-bab 2.6, maka model dari data lengkap yang diperoleh dari model probabilitas dapat diasumsikan sebagai berikut :

$$f(a_1,...,a_C,x_1,...,x_Q \mid \theta)$$

$$= P(A_1 = a_1,...,A_C = a_C, X_1 = x_1,...,X_Q = x_Q \mid \theta) \qquad (4.1)$$

dengan θ adalah parameter yang diestimasi dari data, A adalah variabel kategori, X adalah variabel kuantitatif, sedangkan a dan x masing-masing adalah nilai dari variabel A dan X.

Data *squashing* yang dinyatakan (B, Y, w) dan bentuknya berupa suatu matriks seperti yang telah dijelaskan dalam sub-bab 2.6, diharapkan dapat memberikan informasi yang sama dengan data lengkap yang memuat informasi mengenai θ. Artinya kedua model yaitu data lengkap dan data *squashing* akan mempunyai fungsi likelihood yang sama dengan syarat persamaan log likelihood sebagai berikut :

$$\sum_{i=1}^{M} w_i \log(f(B_{i1},...,B_{iC},Y_{i1},...,Y_{iQ} \mid \theta))$$

$$= P(A_1 = a_1,...,A_C = a_C, X_1 = x_1,...,X_Q = x_Q \mid \theta) \qquad (4.2)$$

yang berlaku untuk setiap nilai dari A dan θ. Model probabilitas $f(A,X,\theta)$ merupakan *relatively smooth function* dari $(X_1,...,X_Q)$ sehingga $f(A_1,...,A_C,X_1,...,X_Q)$ dapat dinyatakan dengan deret Taylor di sekitar titik $x=(x_1,...,x_Q)$, sehingga $\log(f(A_1,A_2,...,A_C,X_1,X_2,...,X_Q\mid\theta))$ di sekitar $x=(x_1,...,x_Q)$ dapat dituliskan sebagai berikut :

$$\log(f(A_1,A_2,...,A_C,X_1,X_2,...,X_Q\mid\theta))$$

$$= f(x_1,x_2,...,x_Q)+(X_1-x_1)\frac{\partial f}{\partial X_1}(x_1,x_2,...,x_Q)+(X_2-x_2)\frac{\partial f}{\partial X_2}(x_1,x_2,...,x_Q)+...+$$

$$(X_Q-x_Q)\frac{\partial f}{\partial X_Q}(x_1,x_2,...,x_Q)+\frac{1}{2!}[(X_1-x_1)^2\frac{\partial^2 f}{\partial X_1^{\,2}}(x_1,x_2,...,x_Q)+$$

$$(X_2-x_2)^2\frac{\partial^2 f}{\partial X_2^{\,2}}(x_1,x_2,...,x_Q)+...+(X_Q-x_Q)^2\frac{\partial^2 f}{\partial X_Q^{\,2}}(x_1,x_2,...,x_Q)+$$

$$2(X_1-x_1)(X_2-x_2)\frac{\partial^2 f}{\partial X_1\partial X_2}(x_1,x_2,...,x_Q)+...+2(X_1-x_1)(X_Q-x_Q)\frac{\partial^2 f}{\partial X_1\partial X_Q}(x_1,x_2,...,x_Q)$$

$$+...+2(X_{Q-1}-x_{Q-1})(X_Q-x_Q)\frac{\partial^2 f}{\partial X_{Q-1}\partial X_Q}(x_1,x_2,..x_Q)]+$$

$$\frac{1}{3!}[(X_1-x_1)^3\frac{\partial^3 f}{\partial X_1^{\,3}}(x_1,x_2,...,x_Q)+...+(X_Q-x_Q)^3\frac{\partial^3 f}{\partial X_Q^{\,3}}(x_1,x_2,...,x_Q)+$$

$$3(X_1-x_1)^2(X_2-x_2)\frac{\partial^3 f}{\partial X_1^{\,2}X_2}(x_1,x_2,...,x_Q)+...+3(X_1-x_1)^2(X_Q-x_Q)\frac{\partial^3 f}{\partial X_1^{\,2}\partial X_Q}(x_1,x_2,...,x_Q)$$

$$+...+3(X_{Q-1}-x_{Q-1})^2(X_Q-x_Q)\frac{\partial^3 f}{\partial X_{Q-1}^{\,2}\partial X_Q}(x_1,x_2,...,x_Q)]+...+$$

$$\frac{1}{k!}[(X_1 - x_1)^k \frac{\partial^k f}{\partial X_1^{\ k}} + \ldots + (X_Q - x_Q)^k \frac{\partial^k f}{\partial X_Q^{\ k}}(x_1, x_2, \ldots, x_Q) +$$

$$k(X_1 - x_1)^{k-1}(X_2 - x_2)\frac{\partial^k f}{\partial X_1^{\ k-1}\partial X_2}(x_1, x_2, \ldots, x_Q) + R_n$$

$$= g_1(X_1 - x_1)^0 (X_2 - x_2)^0 \ldots (X_Q - x_Q)^0 + g_2(X_1 - x_1)^1 (X_2 - x_2)^0 \ldots (X_Q - x_Q)^0$$

$$+ g_3(X_1 - x_1)^0 (X_2 - x_2)^1 \ldots (X_Q - x_Q)^0 + \ldots + g_{Q+1}(X_1 - x_1)^0 (X_2 - x_2)^0 \ldots$$

$$(X_Q - x_Q)^1 + R_n \quad , \quad n \to \infty$$

$$\approx \sum_{k=1}^{K} g_k \prod_{j=1}^{Q} (X_j - x_j)^{p_{kj}} \tag{4.3}$$

dengan K merupakan banyaknya suku deret Taylor dan $(p_{k1}, \ldots, p_{kQ})$ merupakan semua vektor nonnegative yang memenuhi

$$\sum_j p_{kj} \leq d \quad , \qquad d = \text{derajat taksiran.}$$

Vektor $(p_{k1}, \ldots, p_{kQ})$ dimulai dari vektor nol dan satu dan dilanjutkan hingga k, yaitu $\mathbf{p}_1 = (0,\ldots,0)$, $\mathbf{p}_2 = (1,0,\ldots,0)$, ..., $\mathbf{p}_{Q+1} = (0,\ldots,0,1)$, $\mathbf{p}_{Q+2} = (2,0,\ldots,0)$ dan seterusnya.

Misalkan terdapat R daerah dan data lengkap memuat N_r titik pada daerah r, di mana $\sum_{r=1}^{R} N_r = N$. Selanjutnya, data *squashing* juga dimisalkan memuat M_r titik pada daerah r, sehingga $\sum_{i=1}^{M_r} w_i = N_r$.

Dari persamaan (4.2) dan (4.3) diperoleh :

$$\sum_{i=1}^{M_r} w_i \sum_{k=1}^{K} g_k \prod_{j=1}^{Q} \left(Y_{ij} - x_j\right)^{p_{kj}} = \sum_{i=1}^{N_r} \sum_{k=1}^{K} g_k \prod_{j=1}^{Q} \left(X_{ij} - x_j\right)^{p_{kj}} \tag{4.4}$$

$r = 1, \ldots, R$

Jika g_k dihitung secara terpisah, maka diperoleh persamaan :

$$\sum_{i=1}^{M_r} w_i \prod_{j=1}^{Q} \left(Y_{ij} - x_j\right)^{p_{kj}} = \sum_{i=1}^{N_r} \prod_{j=1}^{Q} \left(X_{ij} - x_j\right)^{p_{kj}} \tag{4.5}$$

$r = 1, \ldots, R \, ; k = 1, \ldots, K$

4.1.2 Pengelompokkan Data

Pengelompokkan data merupakan salah satu tahap yang digunakan dalam teknik *squashing*. Untuk masalah data *squashing* pada model regresi logistik, pengelompokkan data menggunakan metode kuantil. Dalam hal ini data dikelompokkan ke dalam beberapa daerah atau partisi, sehingga data akan terbentuk menjadi beberapa kelompok seperti yang telah dijelaskan dalam sub-bab 2.6.

Kuantil merupakan ukuran-ukuran yang menjelaskan atau menunjukkan lokasi sebagian data relatif terhadap keseluruhan data, dimana nilai-nilai yang ada dibawahnya terdapat sejumlah pecahan atau persentase tertentu dari seluruh pengamatan.

4.1.3 Perhitungan Momen

Setelah data dikelompokkan, langkah selanjutnya adalah menghitung momen dari tiap kelompok, yaitu nilai *mean*.

$$\frac{1}{N_r}\sum_{i=1}^{N_r} X_i = \overline{X}_r \tag{4.6}$$

dengan $r = 1, 2, \ldots, R$

4.1.4 *Pseudo Data Points* dan Pembobot

Dari persamaan (4.5), jika X_{ij} adalah data centering dan diasumsikan x_j adalah *mean* dari X_{ij} sehingga $x_j = 0$, akan didapat persamaan

$$\sum_{i=1}^{M_r} w_i \prod_{j=1}^{Q} Y_{ij}^{p_{kj}} = z_k \quad , \quad k = 1,2,\ldots,K \tag{4.7}$$

dengan : $z_k = \sum_{i=1}^{N_r}\prod_{j=1}^{Q} X_{ij}^{p_{kj}}$

Dari persamaan (4.7) tidak digunakan pembobot negatif dan nilai variabel di luar daerah dari variabel data lengkap, karena diinginkan hasil dari data *squashing* akan sama dengan data lengkap. Namun, nilai dari Y tidak dibatasi pada nilai yang berhubungan dengan X atau dapat dituliskan sebagai berikut :

$$w_i \geq 0 \,; \min_i X_{ij} \leq Y_{ij} \leq \max_i X_{ij} \tag{4.8}$$

Untuk mengetahui nilai dari (Y, w), dengan meminimumkan persamaan :

$$S(Y,w)=\sum_{k=1}^{K}u_k\left(z_k-\sum_{i=1}^{M_r}w_i\prod_{j=1}^{Q}Y_{ij}^{\ p_{kj}}\right)^2 \tag{4.9}$$

dengan u_k merupakan *optimization weigths* untuk $k=1,2,...,K$ dan menurut **DuMouchel et.al (1999)** nilai $u_k = 1000$.

Jika persamaan (4.9) didiferensialkan terhadap Y dengan $a=1,...,M_r$ dan $c=1,...,Q$ diperoleh :

$$\frac{\partial S(Y,w)}{\partial Y_{ac}}=\sum_{k=1}^{K}2u_k\left(z_k-\sum_{i=1}^{Mr}w_i\prod_{j=1}^{Q}Y_{ij}^{\ pkj}\right)\left(-w_a\frac{p_{kc}}{Y_{ac}}\prod_{j=1}^{Q}Y_{aj}^{\ pkj}\right) \tag{4.10}$$

$$=0$$

Dan jika didiferensialkan terhadap w dengan $a=1,...,M_r$ didapat :

$$\frac{\partial S(Y,w)}{\partial w_a}=\sum_{k=1}^{K}2u_k\left(z_k-\sum_{i=1}^{M_r}w_i\prod_{j=1}^{Q}Y_{ij}^{\ p_{kj}}\right)\left(-\prod_{j=1}^{Q}Y_{aj}^{\ p_{kj}}\right) \tag{4.11}$$

$$=0$$

Turunan parsial pada persamaan (4.10) terhadap Y adalah :

jika $a=1,...,M_r$ dan $c=1,...,Q$ diperoleh :

$$\frac{\partial^2 S(Y,w)}{\partial Y_{ac}^{\ 2}}=\sum_{k=1}^{K}2u_k\left[\left(-w_a\frac{p_{kc}}{Y_{ac}}\prod_{j=1}^{Q}Y_{aj}^{\ p_{kj}}\right)^2+\left(-w_a\frac{p_{kc}(p_{kc}-1)}{Y_{ac}^{\ 2}}\prod_{j=1}^{Q}Y_{aj}^{\ p_{kj}}\right)\left(z_k-\sum_{i=1}^{M_r}w_i\prod_{j=1}^{Q}Y_{ij}^{\ p_{kj}}\right)\right]$$

jika $a=b$ untuk $a,b=1,...,M_r$ $c\neq d$ untuk $c,d=1,...,Q$ diperoleh :

$$\frac{\partial^2 S(Y,w)}{\partial Y_{ac}\partial Y_{bd}}=\sum_{k=1}^{K}2u_k\left[\left(\frac{p_{kc}p_{kd}}{Y_{ac}Y_{bd}}\left(-w_a\prod_{j=1}^{Q}Y_{aj}^{\ p_{kj}}\right)^2\right)+\left(-w_b\frac{p_{kd}}{Y_{bd}}\prod_{j=1}^{Q}Y_{bj}^{\ p_{kj}}\right)\left(z_k-\sum_{i=1}^{M_r}w_i\prod_{j=1}^{Q}Y_{ij}^{\ p_{kj}}\right)\right]$$

jika $a \neq b$ untuk $a,b = 1,\ldots,M_r$ dan $c \neq d$ untuk $c,d = 1,\ldots,Q$ diperoleh :

$$\frac{\partial^2 S(Y,w)}{\partial Y_{ac}\partial Y_{bd}} = \sum_{k=1}^{K} 2u_k \left[\left(-w_b \frac{p_{kd}}{Y_{bd}} \prod_{j=1}^{Q} Y_{bj}^{\ p_{kj}}\right)\left(-w_a \frac{p_{kc}}{Y_{ac}} \prod_{j=1}^{Q} Y_{aj}^{\ p_{kj}}\right)\right]$$

Turunan parsial pada persamaan (4.10) terhadap w adalah :

jika $a = 1,\ldots,M_r$ dan $c = 1,\ldots,Q$

$$\frac{\partial^2 S(Y,w)}{\partial Y_{ac}\partial w_a} = \sum_{k=1}^{K} 2u_k \left[w_a \frac{p_{kc}}{Y_{ac}} \left(\prod_{j=1}^{Q} Y_{aj}^{\ p_{kj}}\right)^2 + \left(-\frac{p_{kc}}{Y_{ac}} \prod_{j=1}^{Q} Y_{aj}^{\ p_{kj}}\right)\left(z_k - \sum_{i=1}^{M_r} w_i \prod_{j=1}^{Q} Y_{ij}^{\ p_{kj}}\right)\right]$$

jika $a \neq b$ untuk $a,b = 1,\ldots,M_r$ dan $c = 1,\ldots,Q$

$$\frac{\partial^2 S(Y,w)}{\partial Y_{ac}\partial w_b} = \sum_{k=1}^{K} 2u_k \left[\left(-\prod_{j=1}^{Q} Y_{bj}^{\ p_{kj}}\right)\left(-w_a \frac{p_{ka}}{Y_{ac}} \prod_{j=1}^{Q} Y_{aj}^{\ p_{kj}}\right)\right]$$

Sedangkan turunan parsial terhadap Y terhadap persamaan (4.11) adalah:

jika $a = b$ untuk $a,b = 1,\ldots,M_r$ dan $c = 1,\ldots,Q$

$$\frac{\partial^2 S(Y,w)}{\partial w_a \partial Y_{bc}} = \sum_{k=1}^{K} 2u_k \left[w_b \frac{p_{kc}}{Y_{bc}} \left(\prod_{j=0}^{Q} Y_{bj}^{\ p_{kj}}\right)^2 + \left(-\frac{p_{kc}}{Y_{bc}} \prod_{j=1}^{Q} Y_{bj}^{\ p_{kj}}\right)\left(z_k - \sum_{i=1}^{M_r} w_i \prod_{j=1}^{Q} Y_{ij}^{\ p_{kj}}\right)\right]$$

jika $a \neq b$ untuk $a,b = 1,\ldots,M_r$ dan $c = 1,\ldots,Q$

$$\frac{\partial^2 S(Y,w)}{\partial w_a \partial Y_{bc}} = \sum_{k=1}^{K} 2u_k \left[\left(-w_b \frac{p_{kc}}{Y_{bc}} \prod_{j=1}^{Q} Y_{bj}^{\ p_{kj}}\right)\left(-\prod_{j=1}^{Q} Y_{aj}\right)\right]$$

Sedangkan turunan parsial terhadap w terhadap persamaan (4.11) adalah:

$$\frac{\partial^2 S(Y,w)}{\partial w_a \partial w_b} = \sum_{k=1}^{K} 2u_k \left(-\prod_{j=1}^{Q} Y_{aj}^{\ p_{kj}}\right)\left(-\prod_{j=1}^{Q} Y_{bj}^{\ p_{kj}}\right)$$

Karena dari persamaan (4.10) dan (4.11) merupakan persamaan implisit, maka menurut (**DuMouchel et.al.,1999**) digunakan metode Newton-Raphson untuk meminimumkan (4.9).

Berikut ini merupakan langkah-langkah metode Newton-Raphson (**Anonim**) yang telah dijelaskan pada sub-bab 2.4 :

Langkah I :

Menentukan nilai awal parameter-parameter Y_{ij} dan w_i yang dapat ditulis dengan

$$u^0 = \begin{pmatrix} Y_{ij}^{\ 0} \\ w_i^{\ 0} \end{pmatrix} = \begin{pmatrix} Y_{11}^{\ 0} \\ \vdots \\ Y_{M_rQ}^{\ \ 0} \\ w_1^{\ 0} \\ \vdots \\ w_{M_r}^{\ \ 0} \end{pmatrix}$$

Langkah II :

Menentukan fungsi (4.10) dan (4.11) dalam bentuk matriks, dengan

$$u^k = \begin{pmatrix} Y_{ij}^{\ k} \\ w_i^{\ k} \end{pmatrix} = \begin{pmatrix} Y_{11}^{\ k} \\ \vdots \\ Y_{M_rQ}^{\ \ k} \\ w_1^{\ k} \\ \vdots \\ w_{M_r}^{\ \ k} \end{pmatrix};\ k = 0, 1, 2, \ldots, n\ \text{, yaitu}$$

$$A\left(u^k\right) = \begin{pmatrix} R_{ij}\left(u^k\right) \\ P_i\left(u^k\right) \end{pmatrix}$$

dimana $R_{ij}(u^k)$ adalah fungsi dari (4.10) dan $P_i\left(u^k\right)$ adalah fungsi dari (4.11) untuk i = 1, …, M_r , j = 1, …, Q

Langkah III :

Menentukan matriks jacobian dari fungsi (4.10) dan (4.11) yaitu

$$J(u^k)=\begin{pmatrix} \frac{\partial R_{11}(u^k)}{\partial Y_{11}} & \cdots & \frac{\partial R_{11}(u^k)}{\partial Y_{M,Q}} & \frac{\partial R_{11}(u^k)}{\partial w_1} & \cdots & \frac{\partial R_{11}(u^k)}{\partial w_{M_r}} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial R_{M,Q}(u^k)}{\partial Y_{11}} & \cdots & \frac{\partial R_{M,Q}(u^k)}{\partial Y_{M,Q}} & \frac{\partial R_{M,Q}(u^k)}{\partial w} & \cdots & \frac{\partial R_{M,Q}(u^k)}{\partial w_{M_r}} \\ \frac{\partial P_1(u^k)}{\partial Y_{11}} & \cdots & \frac{\partial P_1(u^k)}{\partial Y_{M,Q}} & \frac{\partial P_1(u^k)}{\partial w_1} & \cdots & \frac{\partial P_1(u^k)}{\partial w_{M_r}} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial P_{M_r}(u^k)}{\partial Y_{11}} & \cdots & \frac{\partial P_{M_r}(u^k)}{\partial Y_{M,Q}} & \frac{\partial P_{M_r}(u^k)}{\partial w_1} & \cdots & \frac{\partial P_{M_r}(u^k)}{\partial w_{M_r}} \end{pmatrix}$$

dengan :

$$\frac{\partial R_{11}(u^k)}{\partial Y_{11}}=\sum_{k=1}^{K}2u_k\left[\left(-w_1\frac{p_{k1}}{Y_{11}}\prod_{j=1}^{Q}Y_{1j}^{\,p_{kj}}\right)^2+\left(-w_1\frac{p_{k1}(p_{k1}-1)}{Y_{11}^{\,2}}\prod_{j=1}^{Q}Y_{1j}^{\,p_{kj}}\right)\left(z_k-\sum_{i=1}^{M_r}w_i\prod_{j=1}^{Q}Y_{ij}^{\,p_{kj}}\right)\right]$$

$$\frac{\partial R_{11}(u^k)}{\partial Y_{M,Q}}=\sum_{k=1}^{K}2u_k\left[\left(-w_{M_r}\frac{p_{kQ}}{Y_{M,Q}}\prod_{j=1}^{Q}Y_{M,j}^{\,p_{kj}}\right)\left(-w_1\frac{p_{k1}}{Y_{11}}\prod_{j=1}^{Q}Y_{1j}^{\,p_{kj}}\right)\right]$$

$$\frac{\partial R_{11}(u^k)}{\partial w_1}=\sum_{k=1}^{K}2u_k\left[w_1\frac{p_{k1}}{Y_{11}}\left(-\prod_{j=1}^{Q}Y_{1j}^{\,p_{kj}}\right)^2+\left(-\frac{p_{k1}}{Y_{11}}\prod_{j=1}^{Q}Y_{1j}^{\,p_{kj}}\right)\left(z_k-\sum_{i=1}^{M_r}w_i\prod_{j=1}^{Q}Y_{ij}^{\,p_{kj}}\right)\right]$$

$$\frac{\partial R_{11}(u^k)}{\partial w_{M_r}}=\sum_{k=1}^{K}2u_k\left[\left(-\prod_{j=1}^{Q}Y_{M,j}^{\,p_{kj}}\right)\left(-w_1\frac{p_{k1}}{Y_{11}}\prod_{j=1}^{Q}Y_{1j}^{\,p_{kj}}\right)\right]$$

$$\frac{\partial R_{M,Q}(u^k)}{\partial Y_{11}}=\sum_{k=1}^{K}2u_k\left[\left(-w_1\frac{p_{k1}}{Y_{11}}\prod_{j=1}^{Q}Y_{1j}^{\,p_{kj}}\right)\left(-w_{M_r}\frac{p_{kq}}{Y_{M,Q}}\prod_{j=1}^{Q}Y_{M,j}^{\,p_{kj}}\right)\right]$$

$$\frac{\partial R_{M,Q}(u^k)}{\partial Y_{M,Q}}=\sum_{k=1}^{K}2u_k\left[\left(-w_{M_r}\frac{p_{kQ}}{Y_{MrQ}}\prod_{j=1}^{Q}Y_{M,j}^{\,p_{kj}}\right)^2+\left(-w_{M_r}\frac{p_{kQ}(p_{kQ}-1)}{Y_{M,Q}^{\,2}}\prod_{j=1}^{Q}Y_{M,j}^{\,p_{kj}}\right)\left(z_k-\sum_{i=1}^{M_r}w_i\prod_{j=1}^{Q}Y_{ij}^{\,p_{kj}}\right)\right]$$

$$\frac{\partial R_{M,Q}(u^k)}{\partial w_1}=\sum_{k=1}^{K}2u_k\left[\left(-\prod_{j=1}^{Q}Y_{1j}{}^{p_{kj}}\right)\left(-w_{M_r}\frac{p_{kQ}}{Y_{M,Q}}\prod_{j=1}^{Q}Y_{M_r j}{}^{p_{kj}}\right)\right]$$

$$\frac{\partial R_{M,Q}(u^k)}{\partial w_{M_r}}=\sum_{k=1}^{K}2u_k\left[w_{M_r}\frac{p_{kQ}}{Y_{M,Q}}\left(-\prod_{j=1}^{Q}Y_{M_r j}{}^{p_{kj}}\right)^2+\left(-\frac{p_{kQ}}{Y_{M,Q}}\prod_{j=1}^{Q}Y_{M_r j}{}^{p_{kj}}\right)\left(z_k-\sum_{i=1}^{M_r}w_i\prod_{j=1}^{Q}Y_{ij}{}^{p_{kj}}\right)\right]$$

$$\frac{\partial P_1(u^k)}{\partial Y_{11}}=\sum_{k=1}^{K}2u_k\left[w_1\frac{p_{k1}}{Y_{11}}\left(-\prod_{j=1}^{Q}Y_{1j}{}^{p_{kj}}\right)^2+\left(-\frac{p_{k1}}{Y_{11}}\prod_{j=1}^{Q}Y_{1j}{}^{p_{kj}}\right)\left(z_k-\sum_{i=1}^{M_r}w_i\prod_{j=1}^{Q}Y_{ij}{}^{p_{kj}}\right)\right]$$

$$\frac{\partial P_1(u^k)}{\partial Y_{M,Q}}=\sum_{k=1}^{K}2u_k\left[\left(-w_{M_r}\frac{p_{k1}}{Y_{M,Q}}\prod_{j=1}^{Q}Y_{M_r j}{}^{p_{kj}}\right)\left(-\prod_{j=1}^{Q}Y_{1j}{}^{p_{kj}}\right)\right]$$

$$\frac{\partial P_1(u^k)}{\partial w_1}=\sum_{k=1}^{K}2u_k\left(-\prod_{j=1}^{Q}Y_{1j}{}^{p_{kj}}\right)^2$$

$$\frac{\partial P_1(u^k)}{\partial w_{M_r}}=\sum_{k=1}^{K}2u_k\left(-\prod_{j=1}^{Q}Y_{M_r j}{}^{p_{kj}}\right)\left(-\prod_{j=1}^{Q}Y_{1j}{}^{p_{kj}}\right)$$

$$\frac{\partial P_{M_r}(u^k)}{\partial Y_{11}}=\sum_{k=1}^{K}2u_k\left[\left(-w_1\frac{p_{k1}}{Y_{11}}\prod_{j=1}^{Q}Y_{1j}{}^{p_{kj}}\right)\left(-\prod_{j=1}^{Q}Y_{M_r j}{}^{p_{kj}}\right)\right]$$

$$\frac{\partial P_{M_r}(u^k)}{\partial Y_{M,Q}}=\sum_{k=1}^{K}2u_k\left[w_{M_r}\frac{p_{kQ}}{Y_{M,Q}}\left(-\prod_{j=1}^{Q}Y_{M_r j}{}^{p_{kj}}\right)^2+\left(-\frac{p_{kQ}}{Y_{M,Q}}\prod_{j=1}^{Q}Y_{M_r j}{}^{p_{kj}}\right)\left(z_k-\sum_{i=1}^{M_r}w_i\prod_{j=1}^{Q}Y_{ij}{}^{p_{kj}}\right)\right]$$

$$\frac{\partial P_{M_r}(u^k)}{\partial w_1}=\sum_{k=1}^{K}2u_k\left(-\prod_{j=1}^{Q}Y_{1j}{}^{p_{kj}}\right)\left(-\prod_{j=1}^{Q}Y_{M_r j}{}^{p_{kj}}\right)$$

$$\frac{\partial P_{M_r}(u^k)}{\partial w_{M_r}}=\sum_{k=1}^{K}2u_k\left(\prod_{j=1}^{Q}Y_{M_r j}{}^{p_{kj}}\right)^2$$

Langkah IV :

Mencari nilai koreksi (h^k) ,yaitu

$$h^k = -\left(J(u^k)\right)^{-1} A(u^k) = \begin{pmatrix} h_{11}{}^k \\ \vdots \\ h_{M,Q}{}^k \\ h_1{}^k \\ \vdots \\ h_{M,Q}{}^k \end{pmatrix}$$

dengan $\left(J(u^k)\right)^{-1}$ adalah invers dari $\left(J(u^k)\right)$ untuk k = 0, 1, 2,...,n . Jika matriks $\left(J(u^k)\right)$ singular, maka digunakan invers Moore-Penrose yang dijelaskan pada **(Anggraeni, R, 2002).**

Langkah V :

Menentukan $u^{k+1} = u^k + h^k$ atau $\begin{pmatrix} Y_{11}{}^{k+1} \\ \vdots \\ Y_{M,Q}{}^{k+1} \\ w_1{}^{k+1} \\ \vdots \\ w_{M_r}{}^{k+1} \end{pmatrix} = \begin{pmatrix} Y_{11}{}^k \\ \vdots \\ Y_{M,Q}{}^k \\ w_1{}^k \\ \vdots \\ w_{M,Q}{}^k \end{pmatrix} + \begin{pmatrix} h_{11}{}^k \\ \vdots \\ h_{M,Q}{}^k \\ h_1{}^k \\ \vdots \\ h_{M,Q}{}^k \end{pmatrix}$ dimana

u^{k+1} merupakan nilai estimator yang akan dicari.

Langkah VI :

Melakukan pengulangan dari langkah II sampai langkah V hingga nilai koreksi $h = \left|u^{k+1} - u^k\right|$ < error, dengan error = 0.01. Sehingga diperoleh nilai estimator parameter $\hat{u} = \begin{pmatrix} \hat{Y}_{11}^{\,k+1} \\ \vdots \\ \hat{Y}_{M,Q}^{\,k+1} \\ \hat{w}_1^{\,k+1} \\ \vdots \\ \hat{w}_{M_r}^{\,k+1} \end{pmatrix}$.

3.1.5 Kasus Khusus

Dari persamaan (4.5) diharapkan momen yang terboboti dari data *squashing* sama dengan momen yang tidak terboboti dari data lengkap. Persamaan (4.5) jika digunakan untuk data skalar dengan diberikan $X_1,\ldots,X_N$ dan $p_k = k-1$ dan $x = 0$ menjadi :

$$\sum_{i=1}^{M_r} w_i Y_i^{k-1} = \sum_{i=1}^{N_r} X_i^{\,k-1}, \qquad k = 1,\ldots,K \ , \ r = 1,\ldots,R \tag{4.12}$$

Dari (4.12) untuk k =1 dan r = 1, diperoleh :

$$\sum_{i=1}^{M_1} w_i \, Y_i^{1-1} = \sum_{i=1}^{N_1} X_i^{\,1-1}$$

atau dapat ditulis :

$$\sum_{i=1}^{M_1} w_i = \sum_{i=1}^{N_1} 1 = N_1$$

Untuk menghitung nilai pengamatan baru (Y_i), dari persamaan (4.12), dimisalkan untuk setiap kelompok diambil satu data *squashing* yaitu $M_r = 1$ akan didapatkan :

$$w_1 Y_1 = \sum_{i=1}^{N_1} X_i \tag{4.13}$$

Karena $\sum_{i=1}^{M_1} w_i = N_1$ dan $M_r = 1$, maka $w_1 = N_1$ dan $Y_1 = \frac{\sum_{i=1}^{N_1} X_i}{N_1} = \overline{X}_1$

Dengan cara yang sama, untuk $r = 2, 3, ..., R$ didapatkan :

$$w_r = N_r \text{ dan } \quad Y_r = \overline{X}_r$$

Sehingga secara umum persamaan (4.12) dapat dituliskan sebagai berikut :

$$w_1 Y_1 + w_2 Y_2 + \ldots + w_R Y_R = \sum_{i=1}^{N_1} X_i + \sum_{i=2}^{N_2} X_i + \ldots + \sum_{i=1}^{N_R} X_i \tag{4.14}$$

Dan didapatkan :

$$w_r = N_r \quad \text{dan} \quad Y_r = \frac{\sum_{i=1}^{N_r} X_i}{N_r} = \overline{X}_r \quad , \qquad r = 1, ..., R \tag{4.15}$$

4.2 Nilai Awal Parameter Model Regresi Logistik

Dalam mengestimasi parameter, terdapat hal penting yang sangat mempengaruhi nilai estimator parameter, yaitu penentuan nilai awal dari parameter β_j. Dalam skripsi ini, penentuan nilai awal parameter diperoleh dengan regresi linier, yang akan dijadikan sebagai nilai awal parameter pada metode Newton-Raphson.

4.3 Estimasi Parameter Model Regresi Logistik

Karena Y berdistribusi Bernoulli, maka pdf dapat didefinisikan sebagai berikut :

$$f(Y_i) = \pi^{Y_i}(1-\pi)^{1-Y_i} \tag{4.16}$$

dengan $f(1) = \pi$ dan $f(0) = 1-\pi$

Sehingga fungsi likelihood dapat diberikan sebagai berikut :

$$L(\beta) = \prod_{i=1}^{N} \pi(x_i)^{Y_i}\,[1-\pi(x_i)]^{1-Y_i}$$

$$= \left\{\prod_{i=1}^{N}[1-\pi(x_i)]\right\}\left\{\prod_{i=1}^{N}\left(\frac{\pi(x_i)}{1-\pi(x_i)}\right)^{Y_i}\right\}$$

$$= \left\{\prod_{i=1}^{N}[1-\pi(x_i)]\right\}\left\{\prod_{i=1}^{N}\exp\left[\ln\left(\frac{\pi(x_i)}{1-\pi(x_i)}\right)^{Y_i}\right]\right\}$$

$$= \left\{\prod_{i=1}^{N}[1-\pi(x_i)]\right\}\exp\left[\sum_{i=1}^{N}Y_i\ln\left(\frac{\pi(x_i)}{1-\pi(x_i)}\right)\right] \tag{4.17}$$

Dengan menggunakan persamaan (2.8), fungsi likelihood dari persamaan (4.17) adalah sebagai berikut :

$$L(\beta) = \left\{\prod_{i=1}^{N}[1-\pi(x_i)]\right\}\exp\left[\sum_{i=1}^{N}Y_i\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)\right]$$

$$= \left\{\prod_{i=1}^{N}[1-\pi(x_i)]\right\}\exp\left[\sum_{j=0}^{p}\left(\sum_{i=1}^{N}Y_i x_{ij}\right)\beta_j\right] \tag{4.18}$$

Dari persamaan (2.7) diperoleh

$$[1-\pi(x_i)] = \left[1+\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)\right]^{-1} \tag{4.19}$$

Sehingga didapat fungsi ln likelihood sebagai berikut

$$\ln L(\beta) = \sum_{j=0}^{p}\left(\sum_{i=1}^{N} Y_i x_{ij}\right)\beta_j - \sum_{i=1}^{N}\ln\left[1+\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)\right] \tag{4.20}$$

Selanjutnya dari fungsi ln likelihood dideferensialkan terhadap parameter β_a (a = 0, 1, ...p) diperoleh :

$$\frac{\partial \ln L(\beta)}{\partial \beta_a} = \sum_{i=1}^{N} Y_i x_{ia} - \sum_{i=1}^{N} x_{ia}\left[\frac{\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)}{1+\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)}\right]$$

$$\frac{\partial \ln L(\beta)}{\partial \beta_a} = \sum_{i=1}^{N} x_{ia}\left[Y_i - \frac{\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)}{1+\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)}\right] \tag{4.21}$$

$$\frac{\partial \ln L(\beta)}{\partial \beta_a} = \sum_{i=1}^{N} x_{ia}[Y_i - \pi(x_i)]$$

$$= 0$$

Sedangkan turunan parsial terhadap β_a dan β_b terhadap fungsi ln likelihood adalah :

$$\frac{\partial^2 \ln L(\beta)}{\partial \beta_a \partial \beta_b} = -\sum_{i=1}^{N}\left[\frac{x_{ia}x_{ib}\left[\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)\right]\left[1+\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)\right] - x_{ia}x_{ib}\left[\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)\right]}{\left[1+\exp\left(\sum_{j=0}^{k}\beta_j x_{ij}\right)\right]^2}\right]$$

$$\frac{\partial^2 \ln L(\beta)}{\partial\beta_a \partial\beta_b} = -\sum_{i=1}^{N} x_{ia}x_{ib}\left[\frac{\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)}{1+\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)} - \left(\frac{\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)}{1+\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)}\right)^2\right]$$

$$\frac{\partial^2 \ln L(\beta)}{\partial\beta_a \partial\beta_b} = -\sum_{i=1}^{N} x_{ia}x_{ib}\pi(x_i)(1-\pi(x_i)) \tag{4.22}$$

dengan a,b = 0, 1, ..., p

Dari persamaan (4.21) hasil metode MLE tidak diperoleh penyelesaian dari parameter β_j , yang disebabkan persamaan (4.21) tersebut berbentuk fungsi implisit dimana pada persamaan tersebut terdapat parameter β_j. Oleh karena itu, diperlukan suatu metode numerik untuk dapat menyelesaikannya, yaitu metode Newton-Raphson dari p fungsi dengan p variabel yang tidak diketahui.

4.4 Mean Square Error

Mean Square Error (MSE) di sini digunakan sebagai ukuran kebaikan dari estimator $\hat{\beta}$ untuk masing-masing kelompok dari data *squashing*.

Dari definisi (2.6) diperoleh rumus *Mean Square Error* (MSE) sebagai berikut :

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\{\pi(x_i) - \hat{\pi}(x_i)\}^2 \tag{4.23}$$

dengan $\pi(x_i) = \dfrac{1}{1+\exp(-\beta_0 - \beta_1 x_i)}$ untuk estimator dari data awal

$\hat{\pi}(x_i) = \dfrac{1}{1+\exp\left(-\hat{\beta}_0 - \hat{\beta}_1 x_i\right)}$ untuk estimator dari data *squashing*

4.5 Algoritma

Algoritma di buat berdasarkan teori-teori yang telah dibahas di sub-bab sebelumnya. Pada pembahasan skripsi ini akan dibuat tiga algoritma sebagai berikut :

1. Algoritma untuk menentukan Data *Squashing*

1. Bangkitkan data model regresi logistik dengan variabel respon berdistribusi uniform (0,1) yang telah dibulatkan dan variabel prediktor berdistribusi uniform (0,1).
2. Urutkan data berdasarkan variabel respon, sehingga didapat kelompok data dengan variabel respon nol dan kelompok data dengan variabel respon satu.
3. Urutkan variabel prediktor pada masing-masing variabel respon.
4. Kelompokkan masing-masing kelompok data variabel respon ke dalam beberapa daerah dengan metode kuantil dengan jumlah anggota kelompok yang berbeda.
5. Hitung nilai *mean* dan jumlah anggota dari tiap kelompok yang masing-masing merupakan nilai pengamatan baru (*pseudo data point*) dan pembobot.
6. Kalikan nilai pembobot dengan *pseudo data point.*

2. Algoritma Penentuan Estimator β pada Model Regresi Logistik

1. Menentukan nilai awal parameter-parameter β_j yang diperoleh dengan regresi linier dan dapat ditulis dengan

$$u^0 = \left(\beta_j{}^0\right) = \begin{pmatrix} \beta_0{}^0 \\ \beta_1{}^0 \\ \vdots \\ \beta_p{}^0 \end{pmatrix} \tag{4.24}$$

2. Menentukan fungsi (4.21) dalam bentuk matriks, dengan

$$u^k = \left(\beta_j{}^k\right) = \begin{pmatrix} \beta_0{}^k \\ \beta_1{}^k \\ \vdots \\ \beta_p{}^k \end{pmatrix} \text{ ; k = 0, 1, 2,...,n dan j = 0, 1, ..., p , yaitu}$$

$$A\left(u^k\right) = \left(A_j\left(u^k\right)\right)$$

$$= \begin{pmatrix} A_0\left(u^k\right) \\ A_1\left(u^k\right) \\ \vdots \\ A_p\left(u^k\right) \end{pmatrix} \tag{4.25}$$

dimana $A_j(u^k)$ adalah fungsi dari (4.21) untuk j = 0, ..., p

3. Menentukan matriks jacobian dari fungsi (4.21) yaitu

$$J(u^k) = \begin{bmatrix} \dfrac{\partial A_0\left(u^k\right)}{\partial \beta_0} & \dfrac{\partial A_0\left(u^k\right)}{\partial \beta_1} & \cdots & \dfrac{\partial A_0\left(u^k\right)}{\partial \beta_p} \\ \dfrac{\partial A_1\left(u^k\right)}{\partial \beta_0} & \dfrac{\partial A_1\left(u^k\right)}{\partial \beta_1} & \cdots & \dfrac{\partial A_1\left(u^k\right)}{\partial \beta_p} \\ \cdots & \cdots & \cdots & \cdots \\ \dfrac{\partial A_p\left(u^k\right)}{\partial \beta_0} & \dfrac{\partial A_p\left(u^k\right)}{\partial \beta_1} & \cdots & \dfrac{\partial A_p\left(u^k\right)}{\partial \beta_p} \end{bmatrix} \tag{4.26}$$

dengan :

$$\frac{\partial A_0(u^k)}{\partial \beta_0} = -\sum_{i=1}^{N} x_{i0}x_{i0}\left[\frac{\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)}{1+\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)} - \left(\frac{\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)}{1+\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)}\right)^2\right]$$

$$\frac{\partial A_0(u^k)}{\partial \beta_1} = -\sum_{i=1}^{N} x_{i0}x_{i1}\left[\frac{\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)}{1+\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)} - \left(\frac{\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)}{1+\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)}\right)^2\right]$$

$$\frac{\partial A_0(u^k)}{\partial \beta_p} = -\sum_{i=1}^{N} x_{i0}x_{ip}\left[\frac{\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)}{1+\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)} - \left(\frac{\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)}{1+\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)}\right)^2\right]$$

$$\frac{\partial A_1(u^k)}{\partial \beta_0} = -\sum_{i=1}^{N} x_{i1}x_{i0}\left[\frac{\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)}{1+\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)} - \left(\frac{\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)}{1+\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)}\right)^2\right]$$

$$\frac{\partial A_1(u^k)}{\partial \beta_1} = -\sum_{i=1}^{N} x_{i1}x_{i1}\left[\frac{\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)}{1+\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)} - \left(\frac{\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)}{1+\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)}\right)^2\right]$$

$$\frac{\partial A_1(u^k)}{\partial \beta_p} = -\sum_{i=1}^{N} x_{i1}x_{ip}\left[\frac{\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)}{1+\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)} - \left(\frac{\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)}{1+\exp\left(\sum_{j=0}^{p}\beta_j x_{ij}\right)}\right)^2\right]$$

$$\frac{\partial A_p(u^k)}{\partial \beta_0} = -\sum_{i=1}^{N} x_{ip} x_{i0} \left[\frac{\exp\left(\sum_{j=0}^{p} \beta_j x_{ij}\right)}{1+\exp\left(\sum_{j=0}^{p} \beta_j x_{ij}\right)} - \left(\frac{\exp\left(\sum_{j=0}^{p} \beta_j x_{ij}\right)}{1+\exp\left(\sum_{j=0}^{p} \beta_j x_{ij}\right)} \right)^2 \right]$$

$$\frac{\partial A_p(u^k)}{\partial \beta_1} = -\sum_{i=1}^{N} x_{ip} x_{i1} \left[\frac{\exp\left(\sum_{j=0}^{p} \beta_j x_{ij}\right)}{1+\exp\left(\sum_{j=0}^{p} \beta_j x_{ij}\right)} - \left(\frac{\exp\left(\sum_{j=0}^{p} \beta_j x_{ij}\right)}{1+\exp\left(\sum_{j=0}^{p} \beta_j x_{ij}\right)} \right)^2 \right]$$

$$\frac{\partial A_p(u^k)}{\partial \beta_p} = -\sum_{i=1}^{N} x_{ip} x_{ip} \left[\frac{\exp\left(\sum_{j=0}^{p} \beta_j x_{ij}\right)}{1+\exp\left(\sum_{j=0}^{p} \beta_j x_{ij}\right)} - \left(\frac{\exp\left(\sum_{j=0}^{p} \beta_j x_{ij}\right)}{1+\exp\left(\sum_{j=0}^{p} \beta_j x_{ij}\right)} \right)^2 \right]$$

4. Mencari nilai koreksi (h^k),yaitu

$$h^k = -\left(J(u^k)\right)^{-1} A\,(u^k) = \begin{pmatrix} h_0^{\,k} \\ h_1^{\,k} \\ \cdots \\ h_p^{\,k} \end{pmatrix} \tag{4.27}$$

dengan $\left(J(u^k)\right)^{-1}$ adalah invers dari $\left(J(u^k)\right)$ untuk k = 0, 1, 2,...,n . Jika matriks $\left(J(u^k)\right)$ singular, maka digunakan invers Moore-Penrose yang dijelaskan pada **(Anggraeni, R, 2002).**

5. Menentukan $u^{k+1} = u^k + h^k$ atau $\begin{pmatrix} \beta_1^{\,k+1} \\ \beta_2^{\,k+1} \\ \cdots \\ \beta_p^{\,k+1} \end{pmatrix} = \begin{pmatrix} \beta_1^{\,k} \\ \beta_2^{\,k} \\ \cdots \\ \beta_p^{\,k} \end{pmatrix} + \begin{pmatrix} h_1^{\,k} \\ h_2^{\,k} \\ \cdots \\ h_p^{\,k} \end{pmatrix}$ dimana u^{k+1} merupakan nilai estimator yang akan dicari.

6. Melakukan pengulangan dari langkah II sampai langkah V hingga nilai koreksi $h = \left|u^{k+1} - u^{k}\right| <$ error, dengan error = 0.01. Sehingga diperoleh nilai estimator parameter $\hat{u} = \begin{pmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \\ \cdots \\ \hat{\beta}_p \end{pmatrix}$

3. Algoritma untuk menentukan nilai Mean Square Error (MSE)

1. Masukkan data bangkitan dengan variabel respon Y berdistribusi uniform (0,1) yang telah dibulatkan dan variabel prediktor x berdistribusi uniform (0,1).

2. Masukkan nilai estimasi ($\hat{\beta}$) data awal yang diperoleh dari hasil output program **loop** ke dalam rumus π sebagai berikut :

$$\pi(x_i) = \frac{1}{1 + \exp(-\beta_0 - \beta_1 x_i)}$$

3. Masukkan nilai estimasi ($\hat{\beta}$) masing-masing data *squashing* dengan jumlah kelompok yang berbeda yang diperoleh dari output program **loop** ke dalam rumus $\hat{\pi}$ sebagai berikut :

$$\hat{\pi}(x_i) = \frac{1}{1 + \exp\left(-\hat{\beta}_0 - \hat{\beta}_1 x_i\right)}$$

4. Masukkan nilai dari $\pi(x_i)$ dan $\hat{\pi}(x_i)$ ke dalam rumus MSE seperti pada persamaan (4.23)

5. Bandingkan nilai MSE untuk jumlah data *squashing* yang berbeda

4.6 Implementasi Algoritma ke Program Komputer

Program komputer dibuat dengan menggunakan paket program S-plus. Algoritma yang telah tersusun digunakan pada data bangkitan yang selanjutnya data tersebut akan direduksi dan didapatkan estimator dari model regresi logistiknya. Berikut ini adalah garis besar program yang digunakan untuk mendukung algoritma yang dibuat, secara rincinya dapat dilihat pada Lampiran 1. Adapun programnya adalah :

1. **Program untuk Mendapatkan Data *Squashing***

   Program **squashing** untuk mengurutkan data berdasarkan variabel respon dan mendapatkan data *squashing* dengan variabel respon nol dan variabel respon satu.

2. **Program Estimasi Parameter Model Regresi Logistik**

   Program berikut tersusun atas beberapa program yang dapat dilihat pada lampiran 1. Adapun programnya yaitu :

   2.1 Program **duga (Purwaningsih, A)** untuk menentukan nilai awal parameter model regresi logistik dengan regresi linier.

   2.2 Program **inversMP (Wuryanto, E)** untuk mencari invers dari suatu matriks yang singular.

   2.3 Program **mat** untuk menghitung turunan pertama terhadap parameter-parameter dari fungsi ln likelihood model regresi logistik dalam bentuk matriks.

2.4 Program **matrixJ** untuk menghitung turunan parsial terhadap parameter-parameter dari fungsi ln likelihood model regresi logistik dalam bentuk matriks.

2.5 Program **loop** untuk menentukan nilai estimator parameter model regresi logistik.

3. Program untuk Mendapatkan Nilai Mean Square Error (MSE)

Program MSE untuk menentukan nilai Mean Square Error bagi masing-masing data *squashing*.

4.7 Data

Untuk uji coba program yang telah dibuat digunakan data bangkitan dengan menggunakan paket program S-Plus. Data yang dibangkitkan adalah data berskala biner sebagai variabel respon. Data tersebut dibangkitkan dengan distribusi uniform (0,1) yang merupakan hasil pembulatan, misalnya untuk data ≤ 0.5 dibulatkan menjadi nol, sedangkan untuk data > 0.5 dibulatkan menjadi satu.

Sedangkan untuk variabel prediktor berskala rasio dan dibangkitkan hanya satu variabel prediktor. Data untuk variabel prediktor juga berasal dari distribusi uniform (0,1).

4.8 Analisis Data

Dari data bangkitan (**xy**), yaitu data untuk variabel respon biner dan satu variabel prediktor yang berskala rasio serta satu variabel prediktor yang mempunyai nilai satu yang masing-masing mempunyai jumlah yang sama yaitu 100000 data, data **xy** diurutkan berdasarkan variabel respon biner, sehingga data **xy** terbagi ke dalam dua kelompok yaitu kelompok variabel prediktor dengan variabel respon nol dan kelompok variabel prediktor dengan variabel respon satu. Setelah data terbagi ke dalam dua kelompok berdasarkan variabel respon biner, masing-masing dari kelompok tersebut diurutkan berdasarkan variabel respon prediktor yang berskala rasio.

Untuk langkah selanjutnya, ditentukan terlebih dahulu jumlah kelompok yang diinginkan dan dalam penulisan ini, penulis mengambil contoh beberapa jumlah kelompok yang berbeda, yaitu 2000, 5000, 10000, dan 20000 kelompok. Dengan program **squashing** pada lampiran 1-1 pengelompokkan data dilakukan dengan metode kuantil berdasarkan variabel prediktor berskala rasio, sehingga dapat diketahui anggota dari masing-masing kelompok.

Langkah selanjutnya adalah melakukan reduksi data dengan metode *squashing* menggunakan program **squashing** sehingga diperoleh jumlah anggota untuk tiap kelompok yang jumlah anggotanya berbeda antar kelompok satu dengan kelompok yang lain dan *mean* untuk tiap kelompok yang masing-masing merupakan pembobot dan *pseudo data points.*

Dari program **squashing** juga dapat diketahui bahwa $\sum_{i=1}^{50} w_i Y_i = \sum_{i=1}^{N} X_i$ dengan w adalah pembobot dan Y adalah *pseudo data points*, dan hasilnya seperti terlihat pada tabel 4.1 dan secara rinci dapat dilihat pada lampiran 2-1.

Tabel 4.1 Hasil penjumlahan dari data awal dan data *squashing*

Data	Jumlah data untuk respon nol	Jumlah data untuk respon satu
Data awal	25146.23	24810.39
Data squashing	25146.23	24810.39

dengan data awal adalah $\sum_{i=1}^{N_i} X_i$ dan data *squashing* adalah $\sum_{i=1}^{50} w_i Y_i$

Selain itu, dari program **squashing** juga didapatkan nilai momen dari data awal dan masing-masing data *squashing* untuk jumlah kelompok yang berbeda. Nilai momen yang dimaksud di sini adalah nilai *mean* dan dapat dilihat pada tabel 4.2, secara rinci dapat dilihat pada lampiran 2-1.

Tabel 4.2 Nilai *mean* untuk data awal dan masing-masing data *squashing*.

Data	Nilai mean untuk respon nol	Nilai mean untuk respon satu
Data awal	0.5015203	0.4976011
Data *squashing* 2000 kelompok	12.57311	12.4052
Data *squashing* 5000 kelompok	5.029246	4.962079
Data *squashing* 10000 kelompok	2.514623	2.481039
Data *squashing* 20000 kelompok	1.257311	1.24052

Ternyata dari tabel 4.2 di atas terlihat bahwa semakin banyak jumlah kelompok yang dipilih untuk data *squashing*, maka nilai *mean* akan makin mendekati nilai *mean* data awal.

Untuk estimasi parameter model regresi logistik digunakan data awal dan data *squashing* dengan jumlah kelompok yang telah ditentukan sebelumnya. Dan dengan menerapkan program **loop** pada lampiran 1-2 didapatkan hasil estimator pada tabel 4.3 dan secara rinci dapat dilihat pada lampiran 2-2.

Tabel 4.3 Estimator $\hat{\beta}$ untuk data awal dan data *squashing*

Data	$\hat{\beta}_0$	$\hat{\beta}_1$
Data awal	0.01794032	-0.04712208
Data *squashing* 2000 kelompok	0.008707499	-0.0006972267
Data *squashing* 5000 kelompok	0.009077274	-0.001817088
Data *squashing* 10000 kelompok	0.009540798	-0.00381976
Data *squashing* 20000 kelompok	0.0117504	-0.009408836

Dari tabel 4.3, terlihat bahwa estimator $\hat{\beta}$ dari data *squashing* dengan jumlah kelompok lebih banyak yaitu 20000 kelompok semakin mendekati nilai estimator $\hat{\beta}$ dari data awal.

Setelah diperoleh estimator dari $\hat{\beta}$, maka didapat persamaan regresi logistik untuk data awal sebesar dan data *squashing* sebagai berikut :

$$\hat{\pi}(x_i)_{awal} = \frac{1}{1+\exp(-0.01794032+0.04712208x_1)}$$

$$\hat{\pi}(x_i)_{squashing,2000kelompok} = \frac{1}{1+\exp(-0.008707499+0.0006972267x_1)}$$

$$\hat{\pi}(x_i)_{squashing,5000kelompok} = \frac{1}{1+\exp(-0.009077274+0.001817088x_1)}$$

$$\hat{\pi}(x_i)_{squashing,10000kelompok} = \frac{1}{1+\exp(-0.009540798+0.00381976x_1)}$$

$$\hat{\pi}(x_i)_{squashing,20000kelompok} = \frac{1}{1+\exp(-0.0117504+0.009408836x_1)}$$

Untuk ukuran kebaikan dari estimator $\hat{\beta}$, digunakan indikator *Mean Square Error* dengan menggunakan program **MSE**, dan hasilnya terlihat pada tabel 4.4 dan secara rinci dapat dilihat pada lampiran 2-3.

Tabel 4.4 Hasil *Mean Square Error* antara data awal dengan data *squashing*

Data	MSE
Data *squashing* 2000 kelompok	0.00002338151
Data *squashing* 5000 kelompok	0.00002251884
Data *squashing* 10000 kelompok	0.0000206903
Data *squashing* 20000 kelompok	0.00001739431

Dari tabel 4.4, ternyata yang mempunyai nilai MSE paling kecil atau minimum adalah data *squashing* dengan 20000 kelompok. Hal ini berarti, dengan bertambahnya jumlah kelompok akan menghasilkan MSE makin kecil.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan pembahasan pada penulisan ini , dapat ditarik kesimpulan bahwa :

1. Dengan metode *squashing*, data awal yang besar dapat direduksi menjadi data yang lebih kecil (data *squashing*).
2. Untuk mendapatkan sub sampel (data *squashing*), data awal dikelompokkan terlebih dahulu dengan metode kuantil, kemudian dihitung jumlah anggota dan *mean* dari tiap kelompok yang masing-masing merupakan pembobot dan *pseudo data point*. Data *squashing* merupakan perkalian antara pembobot dengan *pseudo data point.*
3. Nilai momen (*mean*) yang mendekati data awal adalah nilai *mean* dari data *squashing* dengan lebih banyak kelompok.
4. Nilai estimator parameter $\hat{\beta}$ adalah sebagai berikut :

Data	$\hat{\beta}_0$	$\hat{\beta}_1$
Data awal	0.01794032	-0.04712208
Data *squashing* 2000 kelompok	0.008707499	-0.0006972267
Data *squashing* 5000 kelompok	0.009077274	-0.001817088
Data *squashing* 10000 kelompok	0.009540798	-0.00381976
Data *squashing* 20000 kelompok	0.0117504	-0.009408836

5. Dari nilai estimator diatas, didapat persamaan regresi logistik :

$$\hat{\pi}(x_i)_{awal} = \frac{1}{1+\exp(-0.01794032+0.04712208x_1)}$$

$$\hat{\pi}(x_i)_{squashing,2000kelompok} = \frac{1}{1+\exp(-0.008707499+0.0006972267x_1)}$$

$$\hat{\pi}(x_i)_{squashing,5000kelompok} = \frac{1}{1+\exp(-0.009077274+0.001817088x_1)}$$

$$\hat{\pi}(x_i)_{squashing,10000kelompok} = \frac{1}{1+\exp(-0.009540798+0.00381976x_1)}$$

$$\hat{\pi}(x_i)_{squashing,20000kelompok} = \frac{1}{1+\exp(-0.0117504+0.009408836x_1)}$$

6. Untuk ukuran kebaikan dengan metode MSE diperoleh hasil sebagai berikut:

Data	MSE
Data *squashing* 2000 kelompok	0.00002338151
Data *squashing* 5000 kelompok	0.00002251884
Data *squashing* 10000 kelompok	0.0000206903
Data *squashing* 20000 kelompok	0.00001739431

5.2 Saran

Estimasi parameter model regresi logistik untuk tulisan ini hanya menggunakan metode MLE, dan untuk pengembangan lebih lanjut dapat menggunakan metode estimasi lainnya.

Sedangkan penerapan estimasi parameter masih menggunakan satu variabel prediktor berskala rasio, sehingga diharapkan untuk dicoba menggunakan lebih dari satu variabel prediktor dengan skala yang lain.

DAFTAR PUSTAKA

Agresti, A., 1990, *Categorical Data Analysis*, John Wiley and sons, Inc., Canada.

Anggraeni, R. 2002. *Invers Moore-Penrose*, Skripsi, Matematika FMIPA Universitas Airlangga. Surabaya.

Anonim, *Solving A Systems of Nonlinear Equations : Course Notes for 136.212 Introductory Numerical Methods for engineers.* (http://130.179.64.208/intermath/136212/presentcourse/corrections/multiVariableNewtonRaphson.htm).

Anonim, *Smooth Function*, http://en.wikipedia.org/wiki/smooth_function, tgl akses 3 Agustus 2005.

Budhi, W. S., 2001, *Kalkulus Peubah Banyak dan Penggunaannya*, ITB, Bandung.

Bradley, G. L., 1995, *Calculus*, Prentice-Hall, Inc., New Jersey.

Draper, N. R., Smith, H., 1992, *Analisis Regresi Terapan*, Edisi kedua, PT Gramedia, Jakarta.

DuMouchel, W., Chris, V., Theodone, J., Corina, C., and Deryl, P., 1999, *Squashing Flat Files Flatter*, **http://citeseer.nj.nec.com/dumouchel99squasing.html**., tgl akses 19 Februari 2004.

Everitt, S. Brian, 1994, *A Handbook of Statistical Analyses Using S-Plus*, Chapman & Hall, London.

Graybill, F.A., Mood, A.M. and Boes, D.C., 1963, *Introduction To The Theory of Statistics*, Third Edition, Mc. Graw – Hill, Inc – Tokyo – Japan.

Hogg, R. V., and Craig, A. T., *Introduction to Mathematical Statistics*, Fifth Edition, Prentice Hall, Inc., New Jersey.

Hosmer, D. W., Lemeshow, 1989, *Applied Logistic Regression*, John Wiley and sons, Inc., New Jersey.

Kleinbaum, D. G., 1994, *Logistic Regression*, Springer, New York.

Montgomery, D. C., Peck, E. A., 1992, *Introduction to Linier Regression Analysis*, Second Edition, John Wiley and sons, Inc., Canada.

Neter, John., Kutner, Michael., Nachtshein, Christopher., and Wasserman, William., 1996, *Applied Linier Statistics Models*, Mc.Graw-Hill, Inc., Britain.

Walpole,R.E., 1995, *Pengantar Statistika*, Edisi Ketiga, Penerjemah : Ir. Bambang Sumantri, PT.Gramedia, Jakarta.

Lampiran 1. Program

1.1 Program untuk mendapatkan data squashing

```
> squashing
 function(xy, n)
 {
  xy <- as.matrix(xy)
  N <- nrow(xy)
  kol <- ncol(xy)
  yurut <- xy[order(xy[, 1]), 1:kol]
  hitung0 <- 0
  for(i in 1:N) {
     if(yurut[i, 1] == 0) {
        hitung0 <- hitung0 + 1
        hit0 <- hitung0
     }
  }
  x0 <- yurut[order(yurut[1:hit0, 3]), 1:kol]
  bar0 <- nrow(x0)
  x1 <- yurut[(hit0 + 1):N,  ][order(yurut[(hit0 + 1):N,
        3]), 1:kol]
  bar1 <- nrow(x1)
  Q0 <- quantile(x0[,3], probs = seq(0, 1, 1/(bar0 -
       1)))
  h0 <- Q0[-c(1,bar0)]
  w0 <- sample(h0, n - 1,F)
  w0 <- sort(w0)
  j<-0
  q0 <- rep(0, n)
  bobot0 <- rep(0, n)
  y0 <- rep(0, n)
  squashing0 <- matrix(0, n, 1)
  for(i in 1:n) {
     if(i == 1) {
        q0 <- x0[, 3][x0[, 3] <= w0[i]]
     }
     else if(i == n) {
        q0 <- x0[, 3][x0[, 3] > w0[i - 1]]
     }
     else {
        q0 <- x0[, 3][x0[, 3] > w0[i - 1] & x0[, 3] <=
            w0[i]]
     }
     if((length(q0))!=0){
        j<-j+1
        bobot0[j] <- length(q0)
        y0[j] <- sum(q0)/bobot0[j]
        squashing0[j] <- bobot0[j] * y0[j]
     }
```

```
    }
    Q1 <- quantile(x1[,3], probs = seq(0, 1, 1/(bar1 -
          1)))
    h1 <- Q1[-c(1,bar1)]
    w1 <- sample(h1, n - 1,F)
    w1 <- sort(w1)
    k<-0
    q1 <- rep(0, n)
    bobot1 <- rep(0, n)
    y1 <- rep(0, n)
    squashing1 <- matrix(0, n, 1)
    for(i in 1:n) {
       if(i == 1) {
          q1 <- x1[, 3][x1[, 3] <= w1[i]]
       }
       else if(i == n) {
          q1 <- x1[, 3][x1[, 3] > w1[i - 1]]
       }
       else {
          q1 <- x1[, 3][x1[, 3] > w1[i - 1] & x1[, 3] <=
                w1[i]]
       }
       if((length(q1))!=0){
          k<-k+1
          bobot1[k] <- length(q1)
          y1[k] <- sum(q1)/bobot1[k]
          squashing1[k] <- bobot1[k] * y1[k]
       }
    }
    jumlsq0 <- sum(squashing0)
    jumlawal0 <- sum(x0[, 3])
    meanawal0 <- mean(x0[,3])
    meansq0 <- mean(squashing0)
    jumlsq1 <- sum(squashing1)
    jumlawal1 <- sum(x1[, 3])
    meanawal1 <- mean(x1[,3])
    meansq1 <- mean(squashing1)
    xbar0 <- length(squashing0)
    xbar1 <- length(squashing1)
    sq <- rbind(squashing0, squashing1)
    bargab <- length(sq)
    x1 <- matrix(1, bargab, 1)
    y0 <- matrix(0, xbar0, 1)
    y1 <- matrix(1, xbar1, 1)
    y <- rbind(y0, y1)
    x <- cbind(x1, sq)
    datasquashing <- cbind(y, x)
    return(datasquashing,jumlawal0, jumlsq0, meanawal0,
     meansq0, jumlawal1, jumlsq1, meanawal1, meansq1)
}
```

1.2 Program Estimasi Parameter Model Regresi Logistik dengan Metode MLE

1.2.1 Program untuk menentukan nilai awal parameter dengan regresi linier

```
> duga
function(xy)
{
 xy <- as.matrix(xy)
 x <- xy[, 2:3]
 y <- xy[, 1]
 xtx <- t(x) %*% x
 xty <- t(x) %*% y
 beta <- solve(xtx) %*% xty
 beta <- round(beta, 3)
 beta0 <- beta[1]
 beta1 <- beta[2]
 return(beta0, beta1)
}
```

1.2.2 Program untuk mencari invers dari matriks yang singular

```
> inversMP
function(x, eps = 1e-005)
{
 x <- as.matrix(x)
 xsvd <- svd(x)
 diagx <- xsvd$d[xsvd$d > eps]
 if(length(diagx) == 1) {
    xplus <- as.matrix(xsvd$v[, 1] %*% t(
       as.matrix(xsvd$u[, 1]/diagx)))
 }
 else {
    xplus <- xsvd$v[, 1:length(diagx)] %*%
       diag(1/diagx) %*% t(xsvd$u[, 1:
       length(diagx)])
 }
 return(xplus)
}
```

1.2.3 Program untuk menghitung turunan pertama terhadap parameter dari fungsi log likelihood dalam bentuk matriks

```
> mat
function(xy, beta0, beta1)
{
 xy <- as.matrix(xy)
 x <- xy[, 2:3]
 y <- xy[, 1]
 xbar <- nrow(x)
 beta <- c(beta0, beta1)
 p <- length(beta)
 G <- matrix(0, p, 1)
 for(k in 1:p) {
    kerja <- 0
    for(i in 1:xbar) {
       hit <- exp( - sum(beta * x[i,  ]))
       kerja <- kerja + x[i, k] * (y[i] - (1/(1 + hit)))
       G[k, 1] <- kerja
    }
 }
 return(G)
}
```

1.2.4 Program matriks Jacobian

```
> j1.1
 function(xy, beta0, beta1)
 {
    xy <- as.matrix(xy)
    x <- xy[, 2:3]
    xbar <- nrow(x)
    beta <- c(beta0, beta1)
    kerja <- 0
    for(i in 1:xbar) {
       hit <- exp( - sum(beta * x[i,  ]))
                       kerja <- kerja + (x[i, 1] * x[i, 1]
             * (1/(1 + hit) - (1/(1 + hit)^2)))
    }
    J1.1 <-  - kerja
    return(J1.1)
 }
```

```
> j1.2
function(xy, beta0, beta1)
{
 xy <- as.matrix(xy)
 x <- xy[, 2:3]
 xbar <- nrow(x)
 beta <- c(beta0, beta1)
 kerja <- 0
 for(i in 1:xbar) {
    hit <- exp( - sum(beta * x[i,  ]))
    kerja <- kerja + (x[i, 1] * x[i, 2] * (1/(1 + hit)
             - (1/(1 + hit)^2)))
 }
 J1.2 <-  - kerja
 return(J1.2)
 }
```

```
> j2.1
function(xy, beta0, beta1)
{
 xy <- as.matrix(xy)
 x <- xy[, 2:3]
 xbar <- nrow(x)
 beta <- c(beta0, beta1)
 kerja <- 0
 for(i in 1:xbar) {
    hit <- exp( - sum(beta * x[i,  ]))
    kerja <- kerja + (x[i, 2] * x[i, 1] * (1/(1 + hit)
             - (1/(1 + hit)^2)))
 }
 J2.1 <-  - kerja
 return(J2.1)
}
```

```
> j2.2
function(xy, beta0, beta1)
{
 xy <- as.matrix(xy)
 x <- xy[, 2:3]
 xbar <- nrow(x)
 beta <- c(beta0, beta1)
 kerja <- 0
 for(i in 1:xbar) {
    hit <- exp( - sum(beta * x[i,  ]))
    kerja <- kerja + (x[i, 2] * x[i, 2] * (1/(1 + hit)
             - (1/(1 + hit)^2)))
 }
 J2.2 <-  - kerja
 return(J2.2)
}
```

```
> matrixJ
function(xy, beta0, beta1)
{
 xy <- as.matrix(xy)
 J <- matrix(0, 2, 2)
 J[1, 1] <- j1.1(xy, beta0, beta1)
 J[1, 2] <- j1.2(xy, beta0, beta1)
 J[2, 1] <- j2.1(xy, beta0, beta1)
 J[2, 2] <- j2.2(xy, beta0, beta1)
 return(J)
}
```

1.2.5 Program untuk menentukan nilai estimator model regresi logistik

```
> loop
function(xy, n)
{
 xy <- as.matrix(xy)
 sq <- squashing(xy,n)$datasquashing
 u <- matrix(0, 2, 1)
 G <- matrix(0, 2, 1)
 J <- matrix(0, 2, 2)
 h <- matrix(c(1, 1), 2, 1)
 beta <- duga(xy)
 u[1, 1] <- beta$beta0
 u[2, 1] <- beta$beta1
 m <- 1
 cat("estimasi untuk data awal :\n")
 repeat {
```

```
   cat("iterasi", m, "\n")
   J <- matrixJ(xy, u[1, 1], u[2, 1])
   G <- mat(xy, u[1, 1], u[2, 1])
   det <- prod(eigen(J)$values)
   if((abs(det) < 1e-005) | (abs(det) >
        100000000000)) {
      invj <- inversMP(J)
   }
   else {
      invj <- solve(J)
   }
   h <-  - invj %*% G
   u <- u + h
   beta0 <- u[1, 1]
   beta1 <- u[2, 1]
   error <- max(abs(h))
   if(error < 0.01) {
      break
   }
   m <- m + 1
   cat("error :", error, "\n")
}
betatopi0 <- beta0
betatopi1 <- beta1
usq <- matrix(0, 2, 1)
Gsq <- matrix(0, 2, 1)
Jsq <- matrix(0, 2, 2)
hsq <- matrix(c(1, 1), 2, 1)
betasq <- duga(sq)
usq[1, 1] <- betasq$beta0
usq[2, 1] <- betasq$beta1
m <- 1
cat("estimasi untuk data squashing :\n")
repeat {
   cat("iterasi", m, "\n")
   Jsq <- matrixJ(sq, usq[1, 1], usq[2, 1])
   Gsq <- mat(sq, usq[1, 1], usq[2, 1])
   detsq <- prod(eigen(Jsq)$values)
   if((abs(detsq) < 1e-005) | (abs(detsq) >
        100000000000)) {
      invjsq <- inversMP(Jsq)
   }
   else {
      invjsq <- solve(Jsq)
   }
   hsq <-  - invjsq %*% Gsq
   usq <- usq + hsq
   beta0sq <- usq[1, 1]
   beta1sq <- usq[2, 1]
   error <- max(abs(hsq))
   if(error < 0.01) {
```

```
        break
      }
      m <- m + 1
      cat("error :", error, "\n")
    }
    betatopi0sq <- beta0sq
    betatopi1sq <- beta1sq
    return(h, u, betatopi0, betatopi1,hsq , usq,
    betatopi0sq, betatopi1sq)
  }
```

1.3 Program untuk menentukan nilai Mean Square Error (MSE)

```
> MSE
  function(x, beta01, beta1, beta02, beta2, beta03,
   beta3,beta04,beta4,beta05,beta5)
  {
   x <- as.matrix(x)
   bar <- nrow(x)
   kol <- ncol(x)
   Y <- matrix(0, bar, 1)
   Ytopi1 <- matrix(0, bar, 1)
   Ytopi2 <- matrix(0, bar, 1)
   Ytopi3 <- matrix(0, bar, 1)
   Ytopi4 <- matrix(0, bar, 1)
   for(i in 1:bar) {
      Y[i] <- 1/(1 + exp( - beta01 - beta1 * x[i, 3]))
      Ytopi1[i] <- 1/(1 + exp( - beta02 - beta2 * x[i,
                 3]))
      Ytopi2[i] <- 1/(1 + exp( - beta03 - beta3 * x[i,
                 3]))
      Ytopi3[i] <- 1/(1 + exp( - beta04 - beta4 * x[i,
                 3]))
      Ytopi4[i] <- 1/(1 + exp( - beta05 - beta5 * x[i,
                 3]))
      }
   mse1 <- sum((Y - Ytopi1)^2)/bar
   mse2 <- sum((Y - Ytopi2)^2)/bar
   mse3 <- sum((Y - Ytopi3)^2)/bar
   mse4 <- sum((Y - Ytopi4)^2)/bar
   return(mse1, mse2,mse3,mse4)
  }
```

LAMPIRAN 2. Proses Analisis Data

2.1 Analisis Metode Squashing

2.1.1 Analisis Metode Squashing untuk data awal 100000 dengan 2000 kelompok

```
> squashing(xy100,2000)
  $jumlawal0:
  [1] 25146.23

  $jumlsq0:
  [1] 25146.23

  $meanawal0:
  [1] 0.5015203

  $meansq0:
  [1] 12.57311

  $jumlawal1:
  [1] 24810.39

  $jumlsq1:
  [1] 24810.39

  $meanawal1:
  [1] 0.4976011

  $meansq1:
  [1] 12.4052
```

2.1.2 Analisis Metode Squashing untuk data awal 100000 dengan 5000 kelompok

```
> squashing(xy100,5000)
  $jumlawal0:
  [1] 25146.23

  $jumlsq0:
  [1] 25146.23

  $meanawal0:
  [1] 0.5015203

  $meansq0:
  [1] 5.029246

  $jumlawal1:
  [1] 24810.39

  $jumlsq1:
  [1] 24810.39
```

```
$meanawal1:
[1] 0.4976011

$meansq1:
[1] 4.962079
```

2.1.3 Analisis Metode Squashing untuk data awal 100000 dengan 10000 kelompok

```
> squashing(xy100,10000)
$jumlawal0:
[1] 25146.23

$jumlsq0:
[1] 25146.23

$meanawal0:
[1] 0.5015203

$meansq0:
[1] 2.514623

$jumlawal1:
[1] 24810.39

$jumlsq1:
[1] 24810.39

$meanawal1:
[1] 0.4976011

$meansq1:
[1] 2.481039
```

2.1.4 Analisis Metode Squashing untuk data awal 100000 dengan 20000 kelompok

```
> squashing(xy100,20000)
$jumlawal0:
[1] 25146.23

$jumlsq0:
[1] 25146.23

$meanawal0:
[1] 0.5015203

$meansq0:
[1] 1.257311

$jumlawal1:
[1] 24810.39
```

```
$jumlsq1:
[1] 24810.39

$meanawal1:
[1] 0.4976011

$meansq1:
[1] 1.24052
```

2.2. Hasil Estimasi Parameter dengan Metode MLE

2.2.1 Estimasi Parameter untuk data awal 100000 dengan 2000 kelompok

```
> loop(xy100,2000)
 estimasi untuk data awal :
 iterasi 1
 iterasi 2
 iterasi 3
 estimasi untuk data squashing :
 iterasi 1
 iterasi 2
 iterasi 3

 $h:
                  [,1]
 [1,]  5.729644e-007
 [2,] -5.702110e-006

 $u:
               [,1]
 [1,]  0.01794032
 [2,] -0.04712208

 $betatopi0:
 [1] 0.01794032

 $betatopi1:
 [1] -0.04712208

 $hsq:
                  [,1]
 [1,] -3.999062e-007
 [2,] -1.049961e-007

 $usq:
                 [,1]
 [1,]  0.0087074993
 [2,] -0.0006972267

 $betatopi0sq:
 [1] 0.008707499

 $betatopi1sq:
 [1] -0.0006972267
```

2.2.2 Estimasi Parameter untuk data awal 100000 dengan 5000 kelompok

```
> loop(xy100,5000)
 estimasi untuk data awal :
 iterasi 1
 iterasi 2
 iterasi 3
 estimasi untuk data squashing :
 iterasi 1
 iterasi 2
 iterasi 3
 $h:
                [,1]
 [1,]  5.729644e-007
 [2,] -5.702110e-006

 $u:
             [,1]
 [1,]  0.01794032
 [2,] -0.04712208

 $betatopi0:
 [1] 0.01794032

 $betatopi1:
 [1] -0.04712208

 $hsq:
                [,1]
 [1,] -3.468504e-007
 [2,] -2.738675e-007

 $usq:
              [,1]
 [1,]  0.009077274
 [2,] -0.001817088

 $betatopi0sq:
 [1] 0.009077274

 $betatopi1sq:
 [1] -0.001817088
```

2.2.3 Estimasi Parameter untuk data awal 100000 dengan 10000 kelompok

```
> loop(xy100,10000)
 estimasi untuk data awal :
 iterasi 1
 iterasi 2
 iterasi 3
 estimasi untuk data squashing :
 iterasi 1
 iterasi 2
 iterasi 3
```

```
$h:
               [,1]
[1,]  5.729644e-007
[2,] -5.702110e-006

$u:
            [,1]
[1,]  0.01794032
[2,] -0.04712208

$betatopi0:
[1] 0.01794032


$betatopi1:
[1] -0.04712208

$hsq:
               [,1]
[1,] -3.835376e-007
[2,] -4.828925e-007

$usq:
             [,1]
[1,]  0.009540798
[2,] -0.003819760

$betatopi0sq:
[1] 0.009540798

$betatopi1sq:
[1] -0.00381976
```

2.2.4 Estimasi Parameter untuk data awal 100000 dengan 20000 kelompok

```
> loop(xy100,20000)
 estimasi untuk data awal :
 iterasi 1
 iterasi 2
 iterasi 3
 estimasi untuk data squashing :
 iterasi 1
 iterasi 2
 iterasi 3
 $h:
               [,1]
 [1,]  5.729644e-007
 [2,] -5.702110e-006

 $u:
            [,1]
 [1,]  0.01794032
 [2,] -0.04712208

 $betatopi0:
 [1] 0.01794032
```

```
$betatopi1:
[1] -0.04712208

$hsq:
               [,1]
[1,] -7.123774e-008
[2,] -1.255834e-006

$usq:
             [,1]
[1,]  0.011750396
[2,] -0.009408836


$betatopi0sq:
[1] 0.0117504

$betatopi1sq:
[1] -0.009408836
```

2.3 Hasil Mean Square Error

```
>MSE(xy100,0.01794032,-0.04712208,0.008707499,
-0.0006972267,0.009077274,-0.001817088,0.009540798,
-0.00381976,0.0117504,-0.009408836)
$mse1:
[1] 0.00002338151

$mse2:
[1] 0.00002251884

$mse3:
[1] 0.0000206903

$mse4:
[1] 0.00001739431
```