



**PERBANDINGAN METODE NEWTON RAPHSON
DAN POHON REDUKSI UNTUK Mencari ESTIMATOR
PADA DATA TERSENSOR BIVARIAT**

SKRIPSI

Oleh

**Selly Acita
NIM 131810101053**

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



**PERBANDINGAN METODE NEWTON RAPHSON
DAN POHON REDUKSI UNTUK MENCARI ESTIMATOR
PADA DATA TERSENSOR BIVARIAT**

SKRIPSI

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

Oleh

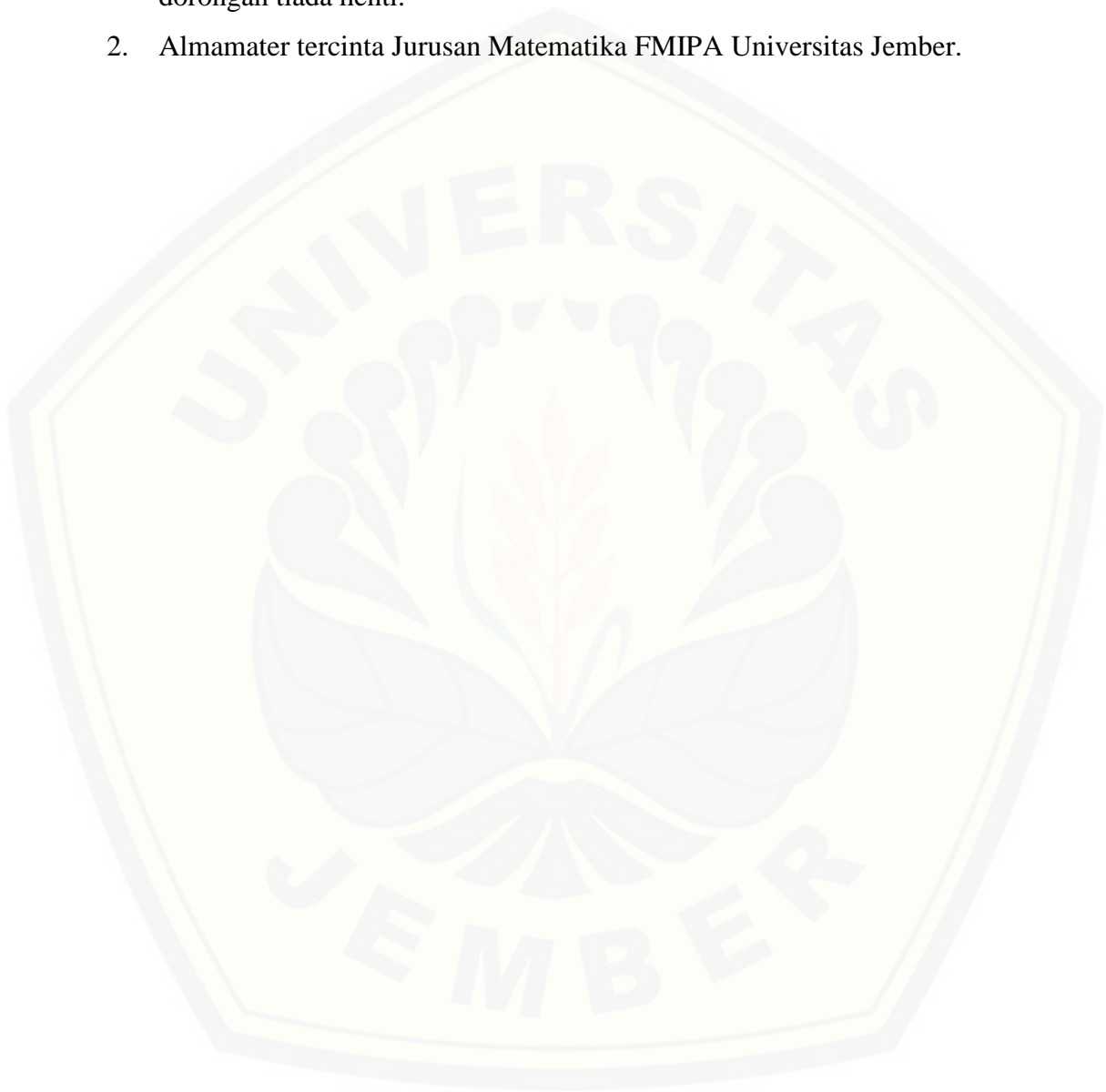
**Selly Acita
NIM 131810101053**

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

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Ayahanda, ibunda dan kakak yang telah memberikan semangat, motivasi dan dorongan tiada henti.
2. Almamater tercinta Jurusan Matematika FMIPA Universitas Jember.



MOTO

“Dibalik kesuksesan orang-orang besar, senantiasa ada amal ibadah yang mempesona”^{*)}



^{*)} dr. Gamal Albisaid

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

nama : Selly Acita

NIM : 131810101053

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Perbandingan Metode Newton Raphson dan Pohon Reduksi untuk Mencari Estimator pada Data Tersensor Bivariat” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi manapun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

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

Jember, Juni 2017

Yang menyatakan,

Selly Acita

NIM 131810101053

SKRIPSI

**PERBANDINGAN METODE NEWTON RAPHSON DAN
POHON REDUKSI UNTUK Mencari ESTIMATOR PADA
DATA TERSENSOR BIVARIAT**

Oleh

Selly Acita
NIM 131810101053

Pembimbing:

Dosen Pembimbing Utama : Dr. Mohamad Fatekurohman, S.Si., M.Si

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

PENGESAHAN

Skripsi berjudul “Perbandingan Metode Newton Raphson dan Pohon Reduksi untuk Mencari Estimator pada Data Tersensor Bivariat” telah diuji dan disahkan pada:

hari, tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas
Jember

Tim Penguji :

Ketua,

Anggota I,

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

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

Anggota II,

Anggota III,

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

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

Mengesahkan

Dekan,

Drs. Sujito, Ph.D
NIP 196102041987111001

RINGKASAN

Perbandingan Metode Newton Raphson dan Pohon Reduksi untuk Mencari Estimator pada Data Tersensor Bivariat; Selly Acita; 131810101053; 2017; 65 halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam.

Pada dunia kesehatan HIV dan AIDS merupakan dua diagnosa yang berbeda. HIV adalah virus yang menyebabkan infeksi, sedangkan AIDS adalah sebuah sindrom. Terinfeksi HIV akan membuat seseorang terjangkit AIDS. Proses dari HIV menjadi AIDS dapat dikatakan sebagai masa inkubasi, yaitu waktu dari penderita tertular/terinfeksi sampai tanda-tanda dan gejala awal penyakit muncul. Pada masa inkubasi atau mulai terjangkitnya HIV, jumlah sel CD-4 dalam tubuh perlahan-lahan akan berkurang sampai setengahnya. Ini berarti tubuh telah kehilangan setengah dari kekebalannya. Ketahanan hidup pasien HIV/AIDS dipengaruhi oleh banyak faktor. Adapun dalam ilmu statistika ketahanan hidup pasien dapat diprediksi dengan menggunakan analisis survival.

Data HIV/AIDS yang digunakan pada penelitian ini menggunakan tipe tersensor interval dalam bentuk himpunan data persegi. Pada data tersensor Gentlemen dan Vandal (2002) menjelaskan bahwa data tersensor dapat direpresentasikan sebagai graf irisan. Distribusi inkubasi tidak mengikuti distribusi apapun, sehingga distribusi massa inkubasi menggunakan metode nonparametrik. Maathuis (2005) menjelaskan bahwa perhitungan NPMLE terdiri dari dua tahapan, yaitu tahap reduksi, yaitu mencari *maximal intersection* dan tahap optimasi, sedangkan Bogaerts dan Lesaffre (2004) berpendapat bahwa dalam perhitungan NPMLE data tersensor interval bivariat terdiri dari dua bagian. Bagian pertama meliputi penetapan daerah irisan dan bagian kedua mencari *maximum likelihood*.

Pada penelitian Liu (2005) perhitungan NPMLE dilakukan menggunakan pohon reduksi dengan data univariat, dengan matriks hasil irisan pada kasus data univariat dapat digunakan langsung untuk memperoleh estimator dan bentuk matriks hasil irisan tersebut berbentuk bujur sangkar. Berbeda dengan data univariat, pada penelitian ini diperoleh matriks insidensi berbentuk segitiga atas

dan untuk memperoleh estimator digunakan matriks *clique* yang merupakan bentuk sederhana dari matriks insidensi.

Pada penelitian ini dilakukan perbandingan hasil NPMLE menggunakan metode Newton Raphson dan pohon reduksi dengan data tersensor interval bivariat. Penyelesaian perhitungan metode Newton Raphson dilakukan dengan menggunakan program MATLAB R2015b dan pemberian nilai awal $p_1=1, p_2=1, p_3=1, p_4=1, p_5=1, p_6=1, p_7=1, p_8=1, p_9=1, p_{10}=1, p_{11}=1, p_{12}=1$ dan $b=1$ dan galat 0,00001 menghasilkan nilai $p_1=0, p_2=0, p_3=0, p_4=0, p_5=0, p_6=0, p_7=1, p_8=0, p_9=0, p_{10}=0, p_{11}=0, p_{12}=0$ dan fungsi *likelihood* 0 sedangkan pada perhitungan pohon reduksi menghasilkan nilai $\hat{p} = \left(\frac{1}{12}, \frac{1}{12}, \frac{1}{12}, \frac{1}{12}, \frac{1}{12}, \frac{1}{12}, \frac{1}{12}, \frac{1}{12}, \frac{1}{12}, \frac{1}{12}, \frac{1}{12}, \frac{1}{12} \right)$ dan diperoleh fungsi *likelihood* $0,413 \times 10^{-8}$. Hasil perhitungan menggunakan pohon reduksi dianggap lebih baik dibandingkan metode Newton Raphson karena nilai fungsi *likelihood* pada pohon reduksi cenderung lebih besar tetapi tidak beda signifikan. Selain itu, perhitungan menggunakan pohon reduksi lebih baik dikarenakan pada metode Newton Raphson hanya p_7 yang memiliki peluang data pasien HIV/AIDS berurutan atau memiliki waktu masa inkubasi yang sama sedangkan pada parameter yang lain tidak berpeluang sama sekali karena menghasilkan nilai probabilitas nol. Hasil dari perhitungan metode Newton Raphson jika diinterpretasikan pada himpunan data persegi menghasilkan kesimpulan yang kurang tepat.

PRAKATA

Puji syukur kepada Allah SWT atas segala rahmat dan karuniaNya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “Perbandingan Metode Newton Raphson dan Pohon Reduksi untuk Mencari Estimator pada Data Tersensor Bivariat”. Skripsi ini disusun untuk memenuhi salah satu syarat untuk menyelesaikan pendidikan strata satu (S1) pada Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Penyusunan dan penyelesaian skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan ucapan terima kasih kepada:

1. Dr. Mohamad Fatekurohman, S.Si., M.Si dan Dian, S.Si., M.Si, selaku Dosen Pembimbing yang telah memberikan arahan dan dorongan serta telah meluangkan waktu, pikiran, tenaga dan perhatian selama penulisan skripsi ini;
2. Prof. Drs. I Made Tirta, M.Sc., Ph.D dan Dr. Alfian Futuhul Hadi, S.Si., M.Si selaku Dosen Penguji yang telah memberikan kritik dan saran sebagai penyempurna skripsi ini;
3. Seluruh dosen Fakultas MIPA Universitas Jember yang telah memberi banyak ilmu pengetahuan selama masa perkuliahan; staff dan karyawan atas segala bantuan yang telah diberikan selama menjadi mahasiswa Fakultas MIPA Universitas Jember;
4. Kedua orang tua tercinta yang telah mendoakan, membantu baik secara moril maupun materiil, serta telah membesarkan dan mengajarkan makna kehidupan; kakak tersayang yang telah menjadi contoh panutan dan inspirasi dan seluruh keluarga besar yang selalu menjadi penyemangat;
5. Sahabat sekaligus teman serumah “*Housemates*” Rizka, Famela dan Renova yang senantiasa selalu menemani dan memberikan semangat;
6. Teman seperjuangan D.Tiara dan Jihan Reni yang selalu menemani dari proses awal sampai tahap akhir; teman-teman ATLAS'13 yang selalu memberikan dukungan dan semangat;
7. Semua pihak yang telah membantu kelancaran skripsi ini.

Penulis juga menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat.

Jember, Juni 2017

Penulis



DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PERSEMBAHAN	ii
HALAMAN MOTTO	iii
HALAMAN PERNYATAAN.....	iv
HALAMAN PEMBIMBINGAN.....	v
HALAMAN PENGESAHAN.....	vi
RINGKASAN	vii
PRAKATA	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL	xiv
DAFTAR LAMPIRAN	xv
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah	3
1.4 Tujuan	3
1.5 Manfaat	3
BAB 2. TINJAUAN PUSTAKA.....	4
2.1 Analisis <i>Survival</i>	4
2.2 Data Tersensor	5
2.3 Representasi Geometri.....	7
2.4 Teori Graf	9
2.4.1 Definisi Graf	9
2.4.2 Graf Irisan	10
2.4.3 Graf <i>Clique</i>	11
2.4.4 <i>Maximal Clique</i>	12
2.5 <i>Maximum Likelihood Estimation (MLE)</i>.....	13

2.6	MLE pada Data Bivariat	17
2.7	Syarat Perlu dan Syarat Cukup Keoptimalan	18
2.8	Pohon Reduksi	19
2.9	Newton Rapshon	21
BAB 3.	METODE PENELITIAN	23
3.1	Data	23
3.2	Langkah-langkah Penelitian	23
BAB 4.	HASIL DAN PEMBAHASAN	25
4.1	Hasil	25
4.1.1	Data HIV /AIDS.....	25
4.1.2	Graf Irisan	26
4.1.3	Matriks insidensi	27
4.1.4	<i>Maximal Clique</i>	28
4.1.5	Pohon Reduksi	29
4.1.6	Fungsi <i>Likelihood</i>	30
4.1.7	Newton Raphson	36
4.2	Pembahasan	37
BAB 5.	KESIMPULAN DAN SARAN	39
5.1	Kesimpulan	39
5.2	Saran	39
DAFTAR PUSTAKA	40
LAMPIRAN	43

DAFTAR GAMBAR

	Halaman
2.1 <i>Maximal intersection</i> dan height map dari 6 persegi panjang.....	8
2.2 Memisahkan hubungan selama transformasi pengamatan persegi panjang menjadi <i>canonical rectangle</i>	8
2.3 Contoh <i>canonical rectagle</i>	9
2.4 Graf komplit	10
2.5 Graf irisan dari Gambar 2.3	10
2.6 Graf <i>clique</i> dari Gambar 2.3.....	12
2.7 (a) <i>cycle</i> genap, $n=8$. (b) <i>cycle</i> ganjil, $n=7$	15
2.8 Diagram alir metode Newton Raphson	22
3.1 Skema metode penelitian	24
4.1 Himpunan data persegi data tersensor interval bivariat	26
4.2 Graf irisan himpunan data persegi panjang.....	27
4.3 Graf <i>clique</i> dari data himpunan persegi	28
4.4 Perhitungan NPMLE dengan pohon reduksi.....	29

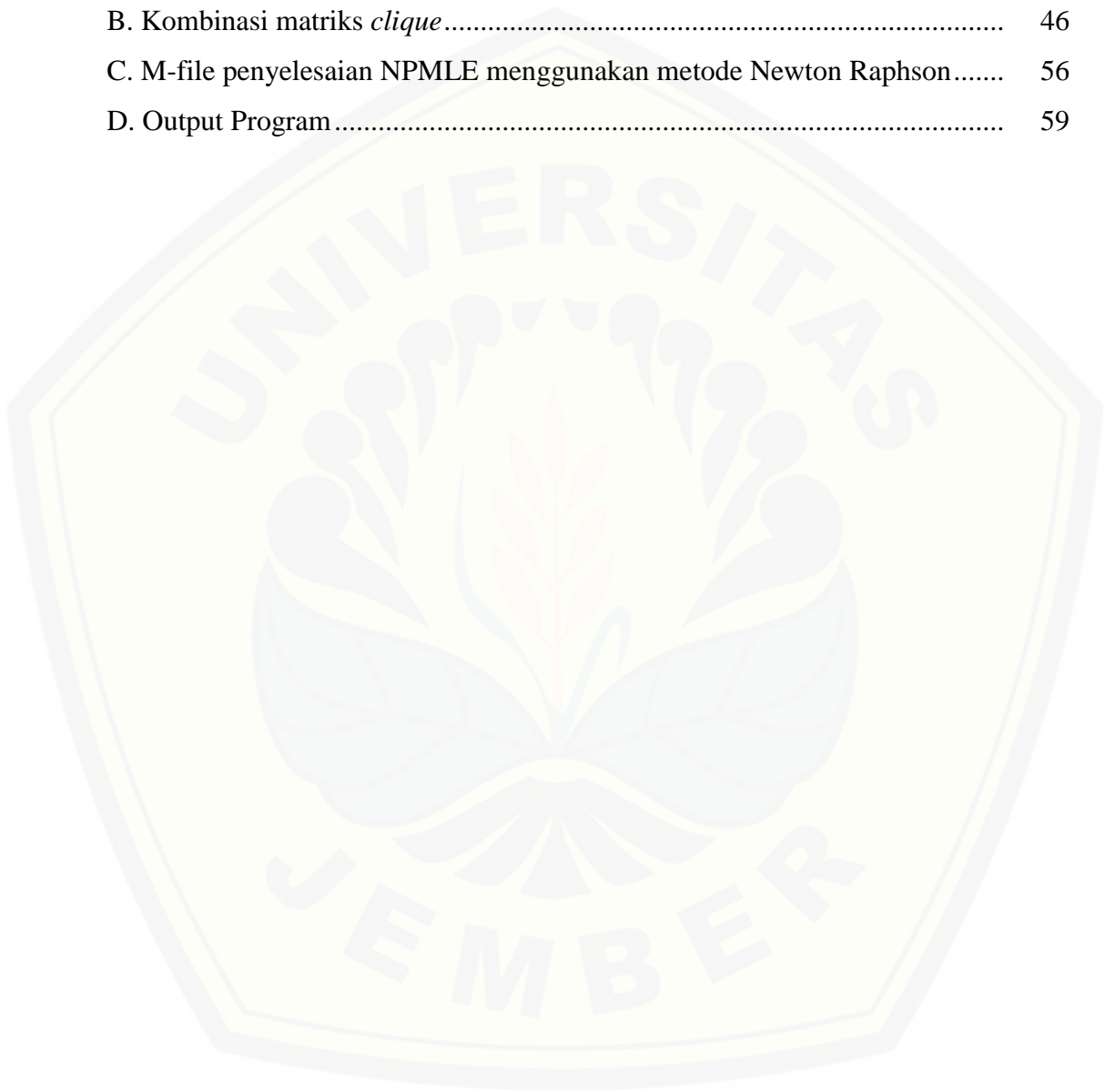
DAFTAR TABEL

	Halaman
4.1 Masa inkubasi pada pembentukan interval data.....	25



DAFTAR LAMPIRAN

	Halaman
A. Data pasien HIV/AIDS RSUD Balung	43
B. Kombinasi matriks <i>clique</i>	46
C. M-file penyelesaian NPMLE menggunakan metode Newton Raphson.....	56
D. Output Program.....	59



BAB 1. PENDAHULUAN

1.1 Latar Belakang

Pada dunia kesehatan HIV dan AIDS merupakan dua diagnosa yang berbeda. HIV adalah virus yang menyebabkan infeksi, sedangkan AIDS adalah sebuah sindrom. Terinfeksi HIV akan membuat seseorang terjangkit AIDS. Proses dari HIV menjadi AIDS dapat dikatakan sebagai masa inkubasi, yaitu waktu dari penderita tertular/terinfeksi sampai tanda-tanda dan gejala awal penyakit muncul. Pada masa inkubasi atau mulai terjangkitnya HIV, jumlah sel CD-4 dalam tubuh perlahan-lahan akan berkurang sampai setengahnya. Ini berarti tubuh telah kehilangan setengah dari kekebalannya. Ketahanan hidup pasien HIV/AIDS dipengaruhi oleh banyak faktor. Adapun dalam ilmu statistika ketahanan hidup pasien dapat diprediksi dengan menggunakan analisis survival.

Menurut Maathuis (2003), estimasi distribusi masa inkubasi merupakan permasalahan nontrivial, karena pada batas akhir waktu inkubasi keduanya tersensor. Waktu infeksi HIV merupakan tipe tersensor interval, dan waktu terjangkitnya AIDS dapat menggunakan tipe sensor kanan atau tersensor interval. Pada data tersensor Gentlemen dan Vandal (2002) menjelaskan bahwa data tersensor dapat direpresentasikan sebagai graf irisan. Distribusi masa inkubasi tidak mengikuti distribusi apapun, sehingga distribusi massa inkubasi menggunakan metode nonparametrik. Pada umumnya pendekatan nonparametrik tidak terikat oleh asumsi bentuk fungsi distribusi tertentu. Dalam kasus nyata hubungan antara variabel respon dengan variabel prediktor seringkali tidak diketahui bentuk fungsi distribusinya. Oleh karena itu pendekatan nonparametrik banyak sekali digunakan. Estimasi nonparametrik dilakukan berdasarkan data pengamatan dengan teknik tertentu.

Misalkan diberikan (X, Y) dan diberikan F_0 sebagai fungsi distribusi bersama. Jika menggunakan mekanisme sensor, (X, Y) saling bebas, maka (X, Y) tidak dapat diobservasi secara langsung. Jadi pengamatan tidak dilakukan langsung pada (x, y) , melainkan pada daerah persegi panjang $R \subset R^2$ yang mengandung (x, y) . Jadi data yang digunakan adalah himpunan data persegi

panjang R_1, \dots, R_n . Dari beberapa penelitian sebelumnya Maathuis (2005) menyimpulkan bahwa perhitungan *Nonparametric Maximum Likelihood Estimator* terdiri dari dua tahapan. Tahap pertama adalah tahap reduksi, yaitu mencari *maximal intersection* A_1, \dots, A_m . Hal tersebut mereduksi dimensi dan pada tahap kedua yaitu tahap optimasi diperlukan penyelesaian masalah optimasi. Pada tahap reduksi fungsi *likelihood* dapat diperoleh dari matriks *clique* untuk menentukan *Nonparametric Maximum Likelihood Estimator* (NPMLE), sedangkan Bogaerts dan Lesaffre (2004) berpendapat bahwa dalam perhitungan *Nonparametric Maximum Likelihood Estimator* data tersensor interval bivariat terdiri dari dua bagian. Bagian pertama meliputi penetapan daerah irisan dan bagian kedua mencari *maximum likelihood*. Namun dalam menentukan *maximum likelihood estimator*, pada persamaan *likelihood* terkadang sulit dilakukan secara analitik atau dengan kata lain tidak diperolehnya estimasi titik eksak, hal ini dikarenakan bentuk persamaan *likelihood* yang sangat kompleks. Untuk kasus yang demikian *maximum likelihood estimator* dapat ditempuh dengan cara iterasi numerik yaitu dengan metode Newton Raphson. Selain itu, untuk mencari estimator dapat menggunakan penyelesaian optimasi dengan kendala dan menggunakan pohon reduksi seperti pada penelitian Liu (2005) yaitu menghitung estimator menggunakan pohon reduksi pada data univariat, dengan matriks hasil irisan pada kasus data univariat dapat digunakan langsung untuk memperoleh estimator.

Berdasarkan penelitian yang telah dilakukan sebelumnya, maka pada penelitian kali ini akan dilakukan perhitungan estimator menggunakan pohon reduksi pada data bivariat yang hasilnya akan dibandingkan dengan pencarian estimator dengan menggunakan metode Newton Raphson. Variabel yang akan digunakan yaitu variabel X yang menyatakan interval waktu terinfeksi HIV dan variabel Y yang menyatakan interval waktu terjangkitnya AIDS.

1.2 Rumusan Masalah

Berdasarkan latar belakang, diperoleh rumusan masalah sebagai berikut :

- a. Bagaimana bentuk matriks insidensi pada graf irisan hasil irisan persegi panjang data bivariat?
- b. Bagaimana perbandingan hasil estimator dengan menggunakan metode Newton Raphson dan pohon reduksi?

1.3 Batasan Masalah

Data untuk penelitian ini dibatasi pada data waktu HIV/AIDS sampai Maret 2017 yang diambil dari RSUD Balung Jember. Data yang digunakan adalah data tersensor kanan dan interval.

1.4 Tujuan

Tujuan yang ingin dicapai dalam penulisan tugas ini adalah:

- a. Mengetahui bentuk matriks insidensi pada graf irisan hasil irisan persegi panjang data bivariat.
- b. Mengetahui perbandingan hasil estimator dengan metode Newton Rapshon dan pohon Reduksi.

1.5 Manfaat

Manfaat penelitian ini adalah untuk mengetahui keterkaitan antara analisis *survival* dan teori graf dan memberikan pemikiran baru tentang mengembangkan atau menyelesaikan suatu masalah khususnya dalam bidang matematika/statistika dengan metode lain guna mendapatkan hasil yang lebih baik. Serta mengetahui peranan matematika khususnya statistika dalam dunia kesehatan guna mengetahui waktu inkubasi yang sama sesuai pada penerapan HIV/AIDS.

BAB 2. TINJAUAN PUSTAKA

2.1 Analisis Survival

Analisis survival (*survival analysis*) atau analisis kelangsungan hidup atau analisis kesintasan bertujuan menaksir probabilitas kelangsungan hidup, kekambuhan, kematian, dan peristiwa-peristiwa lainnya sampai pada periode waktu tertentu. Ada sejumlah model telah dicoba untuk menghubungkan antara faktor risiko, kelangsungan hidup dan jangka waktu penaksiran. Pemilihan model perlu memerhatikan hal-hal berikut :

- a. Bentuk distribusi probabilitas kelangsungan hidup, apakah bersifat parametrik atau nonparametrik, sebab tiap penyakit dan keadaan-keadaan lainnya memiliki bentuk distribusi masing-masing.
- b. Apakah faktor risiko yang mendapat perhatian hanya sebuah (univariat) ataukah majemuk (multivariat).
- c. Ukuran sampel penelitian.
- d. Apakah data mencakup pengamatan tersensor atau tak tersensor. (Murti, 1997)

Dalam menentukan waktu survival, terdapat tiga elemen yang harus diperhatikan yaitu

- a. *Time origin or starting point* (titik awal) adalah waktu dimulainya suatu penelitian.
- b. *Ending event of interest* (kejadian akhir) adalah kejadian yang menjadi inti dari penelitian
- c. *Measurement scale for the passage of time* (skala ukuran untuk berlalunya waktu).

Perbedaan antara analisis survival dengan analisis statistik lainnya adalah adanya data tersensor. Data dikatakan tersensor jika pengamatan waktu survival hanya sebagian, tidak sampai waktu kejadian.

Dalam bidang kesehatan data diperoleh dari suatu pengamatan terhadap sekelompok atau beberapa kelompok individu dan dalam hal ini adalah pasien, yang diamati dan dicatat waktu terjadinya kegagalan dari setiap individu. Kegagalan yang dimaksudkan antara lain adalah kematian karena penyakit

tertentu, keadaan sakit yang terulang kembali setelah pengobatan atau munculnya penyakit baru. Apabila kegagalan yang diamati adalah terjadinya kematian pada pasien maka waktu survival yang dicatat antara lain sebagai berikut :

- a. Selisih waktu mulai dilakukannya pengamatan sampai terjadinya kematian dan data tersebut termasuk data tidak tersensor (*uncensored data*).
- b. Jika waktu kematiannya tidak diketahui, maka memakai selisih waktu mulai dilakukannya pengamatan sampai waktu terakhir penelitian dan data tersebut termasuk data tersensor (*censored data*).

Menurut Collet (2003), data survival tidak memenuhi syarat prosedur standar statistika yang digunakan pada analisis data. Alasan pertama karena data survival biasanya berdistribusi tidak simetris. Model histogram waktu survival pada sekelompok individu yang sama akan cenderung “*positively skewed*”, oleh karena itu histogram akan semakin miring ke kanan sesuai dengan interval waktu dengan jumlah pengamatan terbesar, sehingga tidak ada alasan untuk mengasumsikan bahwa data survival berdistribusi normal.

2.2 Data Tersensor

Perbedaan antara analisis survival dengan analisis statistik lainnya adalah adanya data tersensor. Data tersensor adalah data tercatat saat adanya informasi tentang waktu survival individual, tetapi tidak tahu persis waktu kejadian yang sebenarnya (Kleinbaum dan Klein, 2005). Sedangkan menurut Pyke dan Thompson (1986) data dikatakan tersensor jika pengamatan waktu survival hanya sebagian, tidak sampai *failure event*. Penyebab terjadinya data tersensor antara lain:

- a. *Loss to follow up*, terjadi bila obyek pindah, meninggal atau menolak untuk berpartisipasi.
- b. *Drop out*, terjadi bila perlakuan dihentikan karena alasan tertentu.
- c. *Termination*, terjadi bila masa penelitian berakhir sementara obyek yang diobservasi belum mencapai *failure event*.

Menurut Klein dan Moeschberger (1997), ada beberapa jenis penyensoran yaitu :

- a. sensor kanan (*right censoring*)
terjadi jika individu yang diamati masih tetap hidup pada saat waktu yang telah ditentukan.
- b. sensor kiri (*left censoring*)
terjadi jika semua informasi yang ingin diketahui dari seorang individu telah dapat diperoleh pada awal studi.
- c. sensor selang (*interval censoring*)
jika informasi yang dibutuhkan telah dapat diketahui pada kejadian peristiwa didalam selang pengamatan.
- d. sensor acak (*random censoring*)
terjadi apabila individu yang di amati meninggal karena sebab lain, bukan disebabkan dari tujuan utama penelitian.

Sedangkan menurut Lee dan Wang (2002), tipe penyensor terdiri dari:

- a. Tipe I
Tersensor tipe I yaitu dilakukan pengamatan terhadap objek-objek selama waktu tertentu yang telah ditentukan untuk mengakhiri semua n individu yang masuk pada waktu yang sama. Sering terdapat objek yang mengalami kejadian setelah masa pengamatan selesai dan sebagian lagi mengalami kejadian di luar waktu yang ditetapkan dalam penelitian. Sehingga individu tersebut belum mengalami kejadian hingga akhir periode penelitian, sedangkan waktu awal dari objek penelitian dapat diamati secara penuh.
- b. Tipe II
Tersensor tipe II adalah tipe penyensoran dimana sampel ke- r merupakan penelitian terkecil dalam sampel acak berukuran $n(1 \leq r \leq n)$. Total sampel berukuran n dengan n berlanjut sampai mengalami kejadian meninggal atau gagal maka percobaan akan dihentikan sampai r dari n unit penelitian yang mengalami kejadian. Semua objek yang masih hidup atau belum mengalami kejadian setelah diperoleh r kegagalan maka dikatakan tersensor. Semua unit penelitian masuk pada waktu yang sama.

c. Tipe III

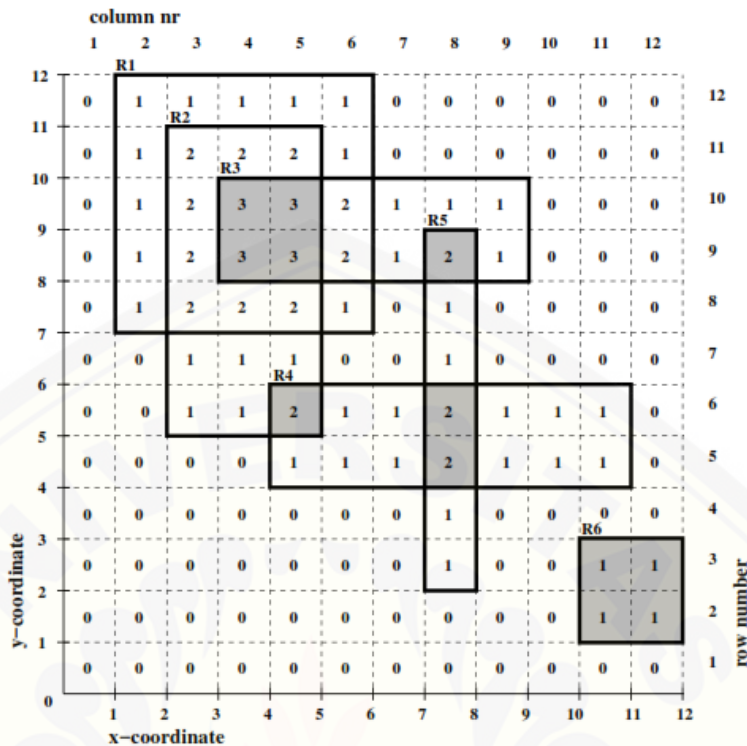
Tersensor tipe III yaitu individu atau objek masuk ke dalam penelitian pada waktu yang berlainan selama periode waktu tertentu. Beberapa objek memiliki kemungkinan gagal atau meninggal sebelum pengamatan berakhir sehingga waktu tahan hidupnya dapat diketahui secara pasti. Kemungkinan kedua adalah objek keluar sebelum pengamatan berakhir dan kemungkinan ketiga adalah objek tetap hidup hingga batas berakhirnya penelitian.

Pengamatan data HIV/AIDS akan menggunakan data tersensor interval dengan tipe 3. Dimana penelitian dihentikan apabila sudah terjangkit AIDS. Pada data tersensor interval, terdapat beberapa kendala terkait dengan estimasi *non-parametric* dalam konteks kegagalan waktu data tersensor interval. *Non-parametric estimators* untuk fungsi *survival* univariat pada data tersensor interval telah dijelaskan oleh beberapa penulis sebelumnya.

2.3 Representasi Geometri

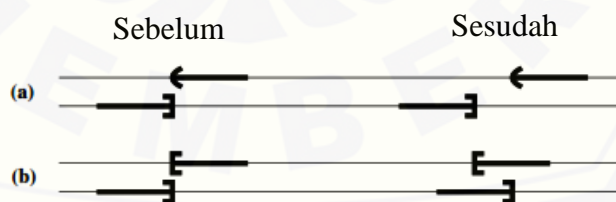
Maathuis (2003) menjelaskan bahwa setiap persegi panjang R_i akan direpresentasikan sebagai $(x_{1,i}, x_{2,i}, y_{1,i}, y_{2,i})$. Titik $(x_{1,i}, y_{1,i})$ adalah sudut kiri bawah dan $(x_{2,i}, y_{2,i})$ adalah sudut kanan atas persegi. $(x_{1,i}, x_{2,i})$ merupakan interval x dan $(y_{1,i}, y_{2,i})$ merupakan interval y dari R_i . Selain itu, digunakan variabel boolean $(C_{1,i}^x, C_{2,i}^x, C_{1,i}^y, C_{2,i}^y)$ untuk mengindikasikan apakah *endpoint* sudah tertutup. Untuk suatu kegagalan diasumsikan bahwa *endpoint* kiri terbuka dan *endpoint* kanan tertutup, jadi $(C_{1,i}^x, C_{2,i}^x, C_{1,i}^y, C_{2,i}^y) = (0, 1, 0, 1)$.

Transformasikan pengamatan persegi panjang R_1, \dots, R_n dalam *canonical rectangle* dengan susunan irisan yang sama. Disebut sebagai himpunan n *rectangle canonical* jika semua koordinat x berbeda dan semua koordinat y berbeda, dan jika nilai-nilai yang diambil dalam himpunan $\{1, 2, \dots, 2n\}$.



Gambar 2.1 *Maximal intersection* dan height map dari 6 persegi panjang.

Pada Gambar 2.1 dilakukan transformasi yang mempertimbangkan koordinat x dan koordinat y secara terpisah dan menggantinya dengan *order* statistik. Permasalahan hanya terletak pada ada kemungkinan hubungan dalam data. Oleh karena itu, perlu didefinisikan bagaimana memisahkan hubungan data tersebut. Berikut akan dijelaskan dasar pemisahan data dengan gambar dibawah ini.

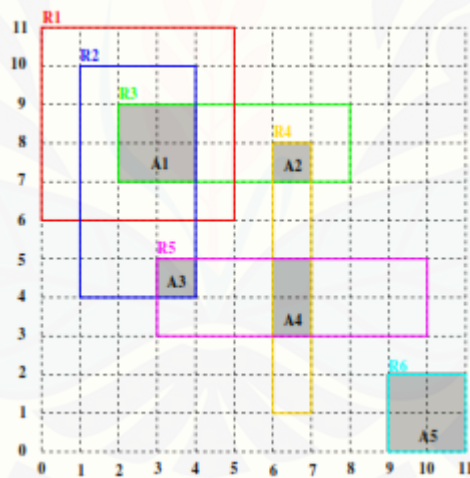


Gambar 2.2 Memisahkan hubungan selama transformasi pengamatan persegi panjang menjadi *canonical rectangle*.

Pada (a) terdapat *endpoint* kiri terbuka $x_{1,i}$ dan *endpoint* kanan tertutup $x_{2,j}$ dengan $x_{1,i} = x_{2,j}$ dan $i \neq j$. Maka interval x dari R_i dan R_j tidak tumpang tindih. Oleh karena itu, *endpoint* diurutkan sehingga interval kanonik yang sama tidak tumpang tindih, dengan $x_{2,j}$ lebih kecil. Pada (b) terdapat *endpoint* kiri tertutup

$x_{1,i}$ dan endpoint kanan tertutup $x_{2,j}$ dengan $x_{1,i} = x_{2,j}$ dan $i \neq j$. Pada interval x R_i dan R_j saling tumpang tindih. Oleh karena itu, *endpoint* diurutkan sehingga interval kanonik yang sama saling tumpang tindih, dengan $x_{1,i}$ lebih kecil. Dengan cara ini, dapat mempertimbangkan semua kemungkinan kombinasi dari *endpoint*. Dengan hasil perolehan pada tabel, telah ditemukan *code* sebuah algoritma untuk membandingkan *endpoint*.

Ada dua alasan untuk mentransformasikan pengamatan persegi panjang menjadi *canonical rectangle*. Pertama, mendahulukan pengerjaan apakah *endpoint* terbuka atau tertutup. Sebagai konsekuensi, tidak perlu memperhitungkan hubungan dan *endpoint* terbuka atau tertutup pada algoritma sebenarnya. Kedua, untuk menyederhanakan algoritma reduksi, karena nomor baris dan kolom pada *height map* secara langsung berhubungan dengan koordinat x dan y dari *canonical rectangle*.



Gambar 2.3 Contoh *canonical rectangle*

2.4 Teori Graf

2.4.1 Definisi Graf

Munir (2010) menjelaskan bahwa graf G didefinisikan sebagai pasangan himpunan (V, E) , ditulis dengan notasi $G = (V, E)$. Pada hal ini V adalah himpunan tidak kosong dari simpul-simpul (*vertice* atau *node*) dan E adalah himpunan sisi (*edge* atau *arcs*) yang menghubungkan sepasang simpul.

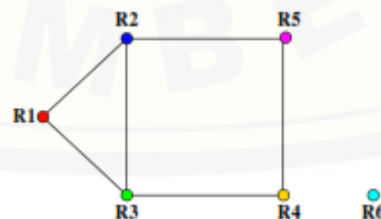
Graf lengkap adalah graf sederhana yang setiap simpulnya mempunyai sisi ke semua simpul lainnya. Graf lengkap dengan n buah simpul dilambangkan dengan K_n . Setiap simpul pada K_n berderajat $n-1$, sedangkan jumlah sisi pada graf lengkap yang terdiri dari n buah simpul adalah $n(n-1)/2$. Rumus ini diperoleh sebagai berikut: untuk 1 buah simpul terdapat $(n-1)$ buah sisi ke $(n-1)$ simpul lainnya, maka untuk n buah simpul terdapat $n(n-1)$ buah sisi. Karena setiap sisi terhitung dua kali untuk pasangan simpul yang bersisian dengannya, maka jumlah sisi seluruhnya dibagi dua, yaitu $n(n-1)/2$.



Gambar 2.4 Graf komplet

2.4.2 Graf Irisan

Suatu graf irisan terbentuk dari himpunan data R , yaitu setiap pengamatan R_j berkorespondensi dengan vertex v_j . Dua vertex v_j dan v_k terhubung oleh sebuah garis jika persegi panjang R_j dan R_k beririsan, dengan $j \neq k$. Menurut Fulkerson dan Gross (1965), sebarang graf terbatas dapat dilihat sebagai keluarga himpunan graf irisan (dalam banyak cara). Sedangkan menurut Marczewski (1945) menunjukkan bahwa kelas graf yang diperoleh sebagai graf irisan adalah kelas sederhana dari semua graf tak berarah.



Gambar 2.5 Graf irisan dari Gambar 2.3

Pada kasus ini simpul-simpul pada graf irisan merepresentasikan maximal intersection. *Maximal intersections* A_1, \dots, A_m diperoleh dari himpunan data persegi panjang R_1, \dots, R_n yang disebut juga sebagai *inner rectangle*. Terdapat

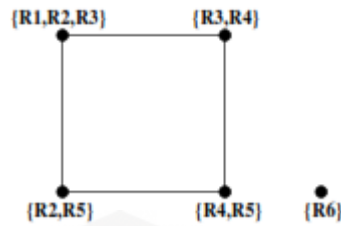
beberapa definisi yang sama untuk konsep *maximal intersection* pada literature. Wong dan Yu (1999) menggunakan berikut : adalah *maximal intersection* jika dan hanya jika merupakan irisan terbatas dari sehingga untuk setiap $A_j \cap R_i = \emptyset$ atau . Gentleman dan Vandal (2002) menggunakan teori graf *prespektif* : *maximal intersection* adalah representasi nyata dari dari *maximal clique* pada graf irisan dari pengamatan persegi panjang. Pada Gambar 2.1 persegi panjang berwarna abu-abu merupakan *maximum intersection*.

Pada graf irisan dapat dibentuk sebuah matriks insidensi yang merupakan representasi dari graf irisan. Matriks insidensi merupakan matriks bersisihan yang menyatakan kebersisihan *vertex* dan *edge*. Pada penelitian ini pembentukan matriks insidensi dimodifikasi guna mempermudah perhitungan estimator. Jika simpul dan sisi bersisihan maka bernilai 1 dan simpul dan sisi yang tidak bersisihan bernilai nol dengan sisi yang sudah dilewati tidak perlu dilewati lagi. Berikut adalah matriks insidensi dari graf irisan pada Gambar 2.5

$$\begin{matrix} \lfloor_0 & \rfloor_1 \end{matrix}$$

2.4.3 Graf *Clique*

Jika diketahui sebuah graf $G = (V, E)$ dan sebuah bilangan bulat k , sebuah *clique* adalah sebuah subset C dari V sehingga setiap pasangan dari titik dalam C saling dihubungkan oleh sebuah sisi. Roberts dan Spencer (1971) menjelaskan jika G adalah sebuah graf, notasi himpunan dari titik dan) adalah himpunan garis. Diperoleh G , misalkan adalah *clique*. Definisikan dengan $V(H) = \{K_1, K_2, \dots, K_n\}$ dan jika $i \neq j$ dan \emptyset . Maka H adalah graf *clique* dari G dan $H=K$.

Gambar 2.6 Graf *clique* dari Gambar 2.3

2.4.4 Maximal Clique

Setiap *maximal clique* pada graf irisan memiliki representasi nyata yang diberikan oleh irisan dari anggota *maximal clique*. Maka notasi representasi nyata dari *maximal clique* M_j dengan A_j , yaitu

$$A_j = \bigcap_{S \in M_j} S.$$

Misalkan diamati data w_1, \dots, w_n dengan himpunan yang sesuai dari pengamatan persegi panjang R_1, \dots, R_n dengan memperhitungkan graf irisan, maka untuk setiap persegi panjang dianggap sebagai sebuah simpul/titik dan dua simpul dihubungkan oleh sebuah garis jika dan hanya jika persegi panjang yang sesuai beririsan. Pada Gambar 2.5 dan Gambar 2.6 diperlihatkan graf irisan dan graf *clique* dari persegi panjang Gambar 2.3. Graf irisan memiliki 5 *maximal clique* : $M_1 = \{R_1, R_2, R_3\}$, $M_2 = \{R_3, R_4\}$, $M_3 = \{R_2, R_5\}$, dan $M_5 = \{R_6\}$. *Maximal clique* mewakili $A_1 = R_1 \cap R_2 \cap R_3$, $A_2 = R_3 \cap R_4$, $A_3 = R_2 \cap R_5$, $A_4 = R_4 \cap R_5$, dan $A_5 = R_6$. Hal tersebut merupakan gambaran *inner rectangle* dimana *Maximum Likelihood Estimation* (MLE) dapat menempatkan massa probabilitas. Matriks *clique* A dari graf irisan adalah :

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

dimana $a_{ij} = 1\{A_j \subset R_i\}$. Baris j dari A mengindikasikan persegi panjang berada dalam *maximal clique* M_j dan kolom i dari A mengindikasikan bahwa *maximal clique* memuat R_i .

Graf irisan dari keluarga persegi panjang disebut graf irisan persegi panjang. Sehingga graf irisan dapat dibentuk tidak hanya dari data, tetapi juga dari *maximal clique*. Pada graf, disebut graf clique jika simpul-simpulnya merepresentasikan *maximal clique*, dan 2 simpul M_j dan M_k dihubungkan oleh sebuah garis jika dan hanya jika M_j dan M_k memiliki irisan tak kosong yaitu jika ada paling sedikit 1 himpunan dalam \mathcal{F} yang merupakan elemen dari M_j dan M_k .

2.5 Maximum Likelihood Estimation (MLE)

Misalkan $\mathcal{R} = \{R_1, \dots, R_n\}$ dinotasikan sebagai himpunan persegi panjang, maka fungsi *likelihood* dan *log likelihood* dapat dituliskan dalam bentuk :

$$L(F) = \prod_{i=1}^n P_F(R_i), \quad \text{dengan} \quad P_F(R_i) = \sum_{j=1}^m \alpha_{ij} p_j \quad (2.1)$$

$$\ln(F) = \sum_{i=1}^n \log(P_F(R_i)), \quad (2.2)$$

Pada pengamatan data persegi panjang akan didapatkan *inner rectangle* yang saling bebas. Maka untuk setiap himpunan data terbatas terdapat sejumlah *inner rectangle* yang saling bebas dimana MLE menentukan massa probabilitas. Hal tersebut menunjukkan sebuah parameter reduksi. Misalkan α_j massa probabilitas yang menentukan *inner rectangle* A_j . Didefinikan $\alpha = (\alpha_1, \dots, \alpha_m)$, maka permasalahan optimasi dapat ditulis sebagai berikut :

$$\hat{\alpha} = \operatorname{argmax}_{\alpha \in R^m} \sum_{i=1}^n \log \left(\sum_{j=1}^m \alpha_j 1\{A_j \subset R_i\} \right) \quad (2.3)$$

sehingga $\alpha_j \geq 0$ untuk $j = 1, \dots, m$ dan $\sum_{j=1}^m \alpha_j = 1$. Ini adalah dimensi *convex* permasalahan optimasi. MLE harus menentukan massa probabilitas positif pada setiap persegi panjang R_i dan tidak dapat menentukan sembarang massa probabilitas diluar persegi panjang.

Dengan menggunakan matriks clique, didapatkan

$$[P(R_i)](\alpha) = \sum_{j=1}^m \alpha_j 1\{A_j \subset R_i\} = (A^T \alpha)_i. \quad (2.4)$$

sehingga didapat

$$\hat{\alpha} = \operatorname{argmax}_{\alpha \in \mathbb{R}^m} \sum_{i=1}^n \log((A^T \alpha)_i) \quad (2.5)$$

Maximum Likelihood Estimate (MLE) $\hat{\alpha}$ sepenuhnya ditentukan oleh matriks *clique*, atau ekuivalen dengan graf irisan. Oleh karena itu, harus ditentukan graf irisan apakah *Maximum Likelihood Estimate* (MLE) unik pada α atau tidak. Gentlemen dan Vandal (1999) menyatakan bahwa *chordless cycles* 4+ (*cycle* dengan panjang 4 dan lebih besar) dapat menyebabkan *non-unique*, karena melampaui batas massa probabilitas yang dapat dipindahkan disekitar *cycle* tanpa merubah *likelihood*. Contoh sederhana dari graf yang memiliki solusi *unique* adalah graf komplit. Graf komplit merupakan *maximal clique* itu sendiri. Oleh sebab itu mengandung 1 *inner rectangle* A_1 , dan MLE $\hat{\alpha}_1 = 1$.

Lemma 2.1 (Maathuis,2003)

Vektor massa $\hat{\alpha} = (\hat{\alpha}_1, \dots, \hat{\alpha}_m) \in K \cap A$ fungsi maximum log *likelihood* pada data tersensor bivariat didefinisikan jika dan hanya jika

$$\langle \alpha, \nabla \Phi_n(\hat{\alpha}) \rangle \geq 0 \text{ untuk semua } \alpha \in K \quad (2.6)$$

dan

$$\langle \alpha, \nabla \Phi_n(\hat{\alpha}) \rangle = 0; \quad (2.7)$$

atau, ekuivalen jika dan hanya jika

$$\sum_{i=1}^n \frac{[P(R_i)](\alpha)}{[P(R_i)](\hat{\alpha})} \leq n \sum_{j=1}^m \alpha_j \text{ untuk semua } \alpha \in K \quad (2.8)$$

dan

$$\sum_{j=1}^m \hat{\alpha}_j = 1; \quad (2.9)$$

atau, ekuivalen jika dan hanya jika

$$\frac{\partial \Phi_n(\hat{\alpha})}{\partial \alpha_j} \begin{cases} \geq 0 \text{ untuk semua } 1 \leq j \leq m \\ = 0 \text{ jika } \hat{\alpha}_j > 0 \end{cases}; \quad (2.10)$$

atau, ekuivalen jika dan hanya jika

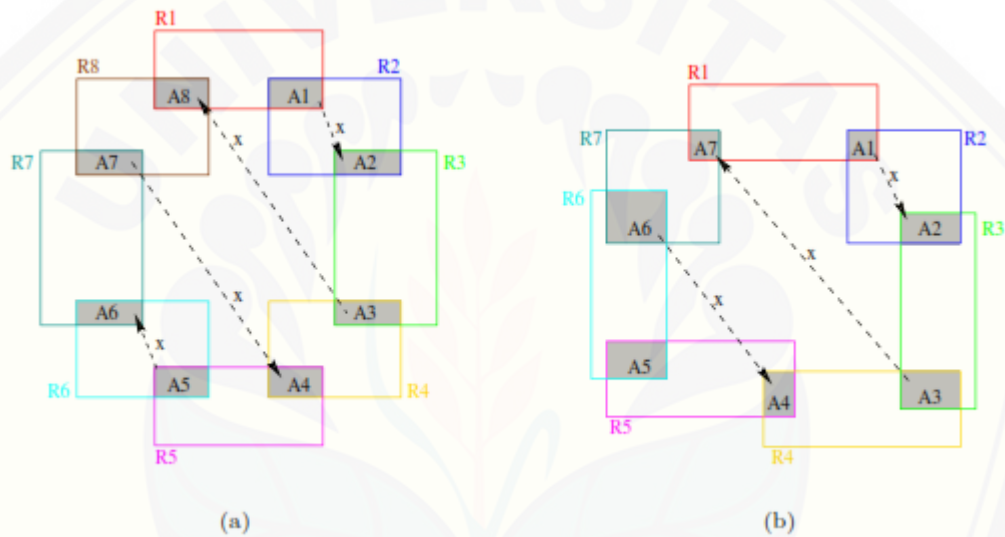
$$\sum_{i=1}^n \frac{1\{A_j \subset R_i\}}{[P(R_i)](\hat{\alpha})} \leq n \text{ untuk semua } j = 1, \dots, m \quad (2.11)$$

dan

$$\sum_{j=1}^m \hat{\alpha}_j = 1; \quad (2.12)$$

Lemma 2.2 (Maathuis,2003)

Sebuah graf irisan yang terdiri dari satu *cycle* genap memiliki solusi *non-unique*.



Gambar 2.7 (a) *cycle* genap, $n=8$. (b) *cycle* ganjil, $n=7$.

Misalkan *cycle* genap dengan ukuran n . Maka terdapat $m = n$ maximal clique dengan 2 ukuran : A_1, \dots, A_n . Tanpa kehilangan bentuk umum, dapat diasumsikan bahwa observasi persegi panjang dan *inner rectangle* adalah bilangan seperti pada Gambar 2.7 (a). Permasalahan memaksimalkan menjadi :

$$\begin{aligned} \max_{(\alpha_1, \dots, \alpha_n)} & \log(\alpha_1 + \alpha_2) + \log(\alpha_2 + \alpha_3) \\ & + \log(\alpha_3 + \alpha_4) + \dots + \log(\alpha_{n-1} + \alpha_n) + \log(\alpha_n + \alpha_1) \end{aligned}$$

sehingga $\alpha_j \geq 0, j = 1, \dots, n$ dan $\sum_{j=1}^m \alpha_j = 1$. Solusi berikut memenuhi dan kondisi cukup untuk optimal, mengingat kondisi 2.10 pada lemma 2.1.

$$\begin{aligned} \hat{\alpha}_1 = \hat{\alpha}_3 = \hat{\alpha}_5 = \dots = \hat{\alpha}_n - 1 = x, \\ \hat{\alpha}_2 = \hat{\alpha}_4 = \hat{\alpha}_6 = \dots = \hat{\alpha}_n = 2/n - x, \end{aligned}$$

Untuk $x \in [0, 2/n]$. Maka solusi pada α tidak *unique*.

Teorema 2.1 (Maathuis, 2007)

Maximum likelihood estimate dari α , $\hat{\alpha}$, adalah *unique* jika rank A sama dengan m .

Definisi 2.1 Rank Matriks (Anton, 2000)

Dimensi ruang baris dan ruang kolom tak nol dari matriks A disebut rank dari A dan dilambangkan dengan *Rank* A . Dimensi ruang nol di A disebut *nullitas* A dan dilambangkan *null* A .

Definisi 2.2 (Anton, 2004)

Jika A adalah suatu matriks berordo $m \times n$, maka

- Rank kolom $A = \dim C(A)$
- Rank baris $A = \dim R(A)$
- *Nullity* $A = \dim N(A)$

Teorema 2.3 (Anton, 2004)

- Rank baris $A =$ Rank kolom A
- Rank $A +$ *nullity* $A = n$

Lemma 2.3 (Maathuis, 2003)

Sebuah graf irisan yang terdiri dari satu *cycle* ganjil memiliki solusi yang *unique*.

$$\begin{aligned} \max_{(\alpha_1, \dots, \alpha_n)} & \log(\alpha_1 + \alpha_2) + \log(\alpha_2 + \alpha_3) \\ & + \log(\alpha_3 + \alpha_4) + \dots + \log(\alpha_{n-1} + \alpha_n) + \log(\alpha_n + \alpha_1) \end{aligned}$$

sehingga $\alpha_j \geq 0, j = 1, \dots, n$ dan $\sum_{j=1}^n \alpha_j = 1$.

Vektor $\hat{\alpha}_1 = \hat{\alpha}_2 = \hat{\alpha}_3 = \dots = \hat{\alpha}_n = 1/n$ syarat dan cukup untuk kondisi optimal. Rank matriks *clique* A sama dengan jumlah kolom, dan oleh karena itu berdasarkan teorema 2.1 maka α memiliki solusi yang *unique*.

Sirait dan Efendi (2013) menjelaskan prosedur dalam menentukan penaksir maksimum likelihood :

1. Menentukan fungsi *likelihood*, $L(F)$.
2. Melakukan transformasi logaritma natural dari $L(F)$.
3. Melakukan differensiasi $\ln L(F)$, dan kemudian menyamakan hasil derivatifnya dengan nol.
4. Menentukan akar persamaan *likelihood*.

2.6 MLE pada Data Bivariat

Pada penelitian ini, fungsi *likelihood* dirumuskan dengan memaksimalkan fungsi

$$L(p) = \prod_{i=1}^n \left[\sum_{j=1}^m \alpha_{ij} p_j \right] \quad (2.13)$$

dengan kendala

$$\begin{aligned} \sum_{i=1}^m p_j &= 1 \\ p_j &\geq 0 \end{aligned} \quad (2.14)$$

Permasalahan tersebut dapat dipandang sebagai permasalahan optimasi. Pada kasus ini akan digunakan metode pengali Lagrange dengan pembentukan fungsi Lagrange yang didefinisikan sebagai

$$L(x, \lambda) = f(\mathbf{X}) + \sum_{j=1}^m \lambda_j g_j(\mathbf{X}) \quad (2.15)$$

dengan cara tersebut, Luknanto (2000) menjelaskan syarat perlu agar persamaan optimasi mencapai titik minimum dapat dicari dengan syarat Kuhn-Tucker. Syarat ini perlu tetapi secara umum bukan merupakan syarat cukup untuk mencapai minimum. Tetapi untuk problema jenis konvex, syarat Kuhn-Tucker menjadi syarat perlu dan cukup sebuah minimum global.

Syarat Kuhn-Tucker untuk

Meminimumkan $f = f(\mathbf{X})$ dengan $\mathbf{X} = \{x_1, \dots, x_n\}$

Kendala $g_j(\mathbf{X}) \leq 0$ dengan $j = 1, \dots, m$

Dapat dinyatakan dalam dalam satu set pernyataan berikut:

$$\frac{\partial f}{\partial x_i} + \sum_{j=1}^m \lambda_j \frac{\partial g_j}{\partial x_i} = 0, \quad i = 1, 2, \dots, m \quad (2.16)$$

$$\lambda_j g_j = 0, \quad i = 1, 2, \dots, m \quad (2.17)$$

$$g_j \leq 0, \quad i = 1, 2, \dots, m \quad (2.18)$$

$$\lambda_j \geq 0, \quad i = 1, 2, \dots, m \quad (2.19)$$

Catatan:

- Jika permasalahannya adalah memaksimumkan, maka $\lambda_j \leq 0$ pada pers.(2.19).
- Jika kendalanya adalah $g_j \geq 0$, maka $\lambda_j \leq 0$ pada persamaan (2.19).
- Jika permasalahannya adalah memaksimumkan dan jika kendalanya adalah $g_j \geq 0$, maka $\lambda_j \geq 0$ pada persamaan (2.19).

2.7 Syarat Perlu dan Syarat Cukup Keoptimalan

Syarat perlu keoptimalan digunakan untuk mencari titik optimal x^* pada pendekatan analitis. Syarat perlu keoptimalan mengatakan bahwa:

Bila x^* adalah titik optimal dari $f(x)$ maka:

$$\nabla f(x^*) = 0$$

dengan $\nabla f(x)$ merupakan vektor gradien. x^* yang memenuhi persamaan diatas merupakan titik optimal.

Syarat cukup keoptimalan digunakan untuk menentukan apakah titik optimal yang didapatkan dari syarat perlu keoptimalan merupakan titik minimum atau titik maksimum. Syarat cukup keoptimalan yaitu:

1. Bila $\nabla f(x^*) = 0$ dan $H(x^*)$ definit positif maka x^* titik minimum.
2. Bila $\nabla f(x^*) = 0$ dan $H(x^*)$ definit negatif maka x^* titik minimum

Ada dua cara untuk menentukan apakah suatu matriks persegi merupakan matriks definit positif atau definit negatif atau tidak definit. Cara pertama adalah:

Definisi 2.3 (Anton, 1995)

Jika A matriks persegi $n \times n$, maka berlaku

1. A disebut definit positif $\leftrightarrow x^t Ax > 0$
2. A disebut definit negatif $\leftrightarrow x^t Ax < 0$
3. A disebut semi definit positif $\leftrightarrow x^t Ax \geq 0$
4. A disebut semi definit negatif $\leftrightarrow x^t Ax \leq 0$

Adapun cara kedua yang digunakan adalah sebagai berikut:

1. A disebut definit positif $\leftrightarrow \det(A_i) > 0$
2. A disebut definit negatif $\leftrightarrow (-1)^i \det(A_i) < 0$

3. A disebut semi definit positif $\leftrightarrow \det(A_i) \geq 0$
4. A disebut semi definit negatif $\leftrightarrow (-1)^i \det(A_i) \leq 0$

2.8 Pohon Reduksi

Pohon reduksi merupakan suatu pendekatan yang didasarkan pada data yang disensor dapat dibagi menjadi beberapa kelompok sehingga pengamatan dari kelompok yang berbeda tidak berpotongan, sehingga dapat menghitung NPMLE dengan data pada masing-masing kelompok secara terpisah. Liu (2005) menjelaskan dua prinsip sederhana dalam perhitungan NPMLE menggunakan pohon reduksi:

1. Pengamatan universal memiliki peranan dalam menemukan NPMLE dan dapat diabaikan untuk tujuan estimasi. Sehingga hanya mempertimbangan himpunan data dimana semua pengamatan universal dihapus.
2. Jika himpunan data \mathcal{R} dapat dibagi kedalam grup \mathcal{R}_g , $g = 1, \dots, G, G \geq 2$, maka NPMLE dapat ditemukan pada setiap grup \mathcal{R}_g secara terpisah dengan total massa grup $|\mathcal{R}_g|/|\mathcal{R}|$.

Dua prinsip diatas dapat digunakan langsung pada pembentukan pohon reduksi dari data. Simpul pada pohon adalah subset dari himpunan data asli \mathcal{R} , dengan akar dari pohon adalah himpunan \mathcal{R} itu sendiri. Berikut adalah beberapa kasus pada saat pembentukan pohon reduksi:

1. \mathcal{R} dapat dibagi
Maka dibentuk grup untuk \mathcal{R} 's, katakanlah $\mathcal{R}_1, \dots, \mathcal{R}_k$ sebagai anak dari \mathcal{R} .
2. \mathcal{R} memiliki pengamatan universal (oleh karena itu tidak dapat dibagi)
Maka $\mathcal{R}^{(r)}$ menjadi satu-satunya anak dari \mathcal{R} .
3. \mathcal{R} tidak mengandung pengamatan universal juga tidak dapat dibagi
Maka \mathcal{R} menjadi sebuah daun.

Penerapan pada setiap \mathcal{R}_j (kasus 1) dan $\mathcal{R}^{(r)}$ akan digunakan untuk mendapatkan pohon reduksi. Daun dari pohon reduksi tidak dapat dibagi dan bebas dari pengamatn universal. Selain itu, himpunan daun pada pohon reduksi adalah *unique*.

Alokasi massa NPMLE untuk setiap daun dapat diperoleh dalam proses pembentukan pohon reduksi. Jumlah massa yang ditetapkan pada akar adalah 1. Misalkan \mathcal{R}_* merupakan sebuah himpunan pengamatan dalam satu simpul pada setiap tingkat pada pohon reduksi. Massa tidak akan berubah antara \mathcal{R}_* dan $\mathcal{R}_*^{(r)}$. Ketika \mathcal{R}_* (atau $\mathcal{R}_*^{(r)}$ sebagai kasus) dapat dibagi, akar dari subpohon \mathcal{R}_* (atau $\mathcal{R}_*^{(r)}$) akan menetapkan massa proporsional untuk jumlah pengamatan disetiap simpul, dengan total massa yang sama dengan \mathcal{R}_* . Dari proses pembentukan pohon reduksi diketahui bahwa total massa NPMLE pada setiap daun selalu *unique*, bahkan ketika pembauran *nonunique* terjadi untuk himpunan data asli.

Untuk menemukan NPMLE yang diberikan data asli, hanya perlu fokus pada daun dari pohon reduksi. Misalkan $\mathcal{R}_1, \dots, \mathcal{R}_L$ merupakan himpunan data yang sesuai dengan daun dari pohon reduksi dengan massa m_1, \dots, m_L dan misalkan \hat{P}_k vektor probabilitas NPMLE untuk himpunan data $\mathcal{R}_k, k = 1, \dots, L$. Maka NPMLE yang diberikan oleh himpunan data asli adalah

$$\hat{P} = \begin{bmatrix} m_1 \hat{P}_1 \\ \vdots \\ m_L \hat{P}_L \end{bmatrix}$$

dengan \hat{P}_k adalah vektor probabilitas NPMLE untuk himpunan data \mathcal{R}_k masing-masing, $k = 1, \dots, L$.

Menurut Fatekurohman (2014) menjelaskan langkah – langkah algoritma pohon reduksi untuk data bivariat yang merupakan pengembangan dari algoritma pohon reduksi Liu (2005). Pengembangan tersebut dikembangkan dari kasus 3 Liu (2005): Jika \mathcal{R} tidak mempunyai pengamatan universal (dalam arti \mathcal{R} dapat dibagi) maka $\mathcal{R}^{(r)}$ merupakan daun dari \mathcal{R} , untuk data tersensor bivariat sangat jarang ditemukan pengamatan universal, sehingga untuk menghitung NPMLE diperlukan langkah penyederhanaan dari matriks insidensi ke matriks clique. Adapun langkah – langkah yang dilakukan adalah:

- Langkah awal (0), diberikan matriks *clique* sebagai akar.
- Langkah 1. Matriks *clique* memuat pengamatan universal atau tidak, jika memuat pengamatan universal maka massa probabilitas 0, jika tidak

maka massa probabilitas dapat dicari dari sisa matriks. (langkah ini akan berulang sampai semua massa probabilitas diperoleh).

- Langkah 2, untuk data tersensor interval bivariat, matriks yang digunakan adalah matriks *clique*.

2.9 Newton Raphson

Metode Newton Raphson merupakan salah satu metode iterasi yang digunakan untuk menyelesaikan permasalahan nonlinier. Secara umum metode Newton Raphson dikembangkan dari perluasan deret Taylor

$$f(x) = f(x_n) + f'(x_n)(x - x_n) + \frac{f''(x_n)}{2!}(x - x_n)^2 + \dots + \frac{f^n(x_n)}{n!}(x - x_n)^n.$$

Widana,dkk (2013) menjelaskan bahwa pada sistem persamaan nonlinier, semua persamaan harus disama dengankan nol. Dimana penyelesaiannya menggunakan perluasan metode Newton Raphson melalui ekspansi deret taylor pada masing-masing persamaan, dengan ekspansi deret taylor:

$$f(x_{i+1}) = f(x_i) + (x_{i+1} - x_i)f'(x_i) \quad (2.20)$$

maka persamaan 2.13 dapat disusun ulang menjadi

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \quad (2.21)$$

Iterasi dimulai dengan memberikan nilai awal untuk x . Iterasi akan berhenti apabila memenuhi pendekatan galat relatif $|x_{i+1} - x_i| < \varepsilon$, dengan x_{i+1} merupakan nilai x yang baru. Kekonvergenan ditentukan pada pemilihan tebakan awal. Tebakan yang terlalu jauh dari solusi akan menyebabkan iterasi divergen.

Apabila peubah atau parameternya berdimensi tinggi, maka fungsi turunan pertamanya berupa vektor T dan turunan keduanya merupakan matriks Hessian. Bentuk multivariat dari metode Newton Raphson yaitu:

$$x_{i+1} = x_i - H(x_i)^{-1}T(x_i) \quad (2.15)$$

Misalkan $x = (x_1, x_2, x_3, \dots, x_m)$, maka vektor T dan matrik Hessian H dapat di tunjukkan sebagai berikut :

$$T(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_m} \end{bmatrix} \text{ dan } H(x) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_m} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \dots & \frac{\partial^2 f}{\partial x_2 \partial x_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_m \partial x_1} & \frac{\partial^2 f}{\partial x_m \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_m^2} \end{bmatrix}$$

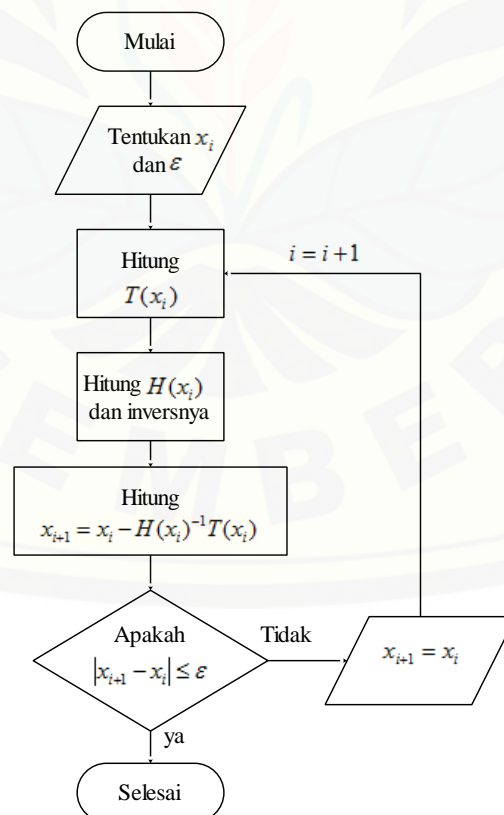
(Atkinson, 1989)

Algoritma metode Newton Raphson

1. Tentukan nilai titik awal x_i dan $\varepsilon > 0$
2. Hitung $T(x_i)$
3. Hitung $H(x_i)$ dan inversnya, $H(x_i)^{-1}$
4. Hitung $x_{i+1} = x_i - H(x_i)^{-1}T(x_i)$
5. Jika $|x_{i+1} - x_i| > \varepsilon$, ulangi langkah 2.

Jika $|x_{i+1} - x_i| \leq \varepsilon$, berhenti

Agar lebih mudah dalam memahami algoritma Newton raphson, dibawah ini diberikan diagram alir metode Newton raphson



Gambar 2.8 Diagram alir metode Newton Raphson

BAB 3. METODE PENELITIAN

3.1 Data

Data yang akan digunakan dalam penelitian ini adalah data tersensor interval bivariat. Data yang diambil adalah data HIV/AIDS dari RSUD Balung Jember. Pada kasus ini variabel yang akan digunakan adalah sebagai berikut :

a. Variabel Bebas (X)

Variabel bebas penelitian ini adalah interval waktu terinfeksi HIV

b. Variabel Terikat (Y)

Variabel terikat penelitian ini adalah interval waktu terjangkit AIDS

3.2 Langkah-langkah Penelitian

Langkah-langkah yang akan digunakan dalam menentukan NPMLE dengan menggunakan metode Newton Raphson dan pohon reduksi untuk data tersensor interval bivariat penyakit HIV/AIDS secara skematik dapat dilihat pada Gambar 3.1.

Penjelasan skema pada Gambar 3.1 untuk memperoleh hasil yang diinginkan adalah sebagai berikut :

a. Data Bivariat (Data HIV/AIDS)

Langkah awal yang dilakukan adalah mencari data bivariat HIV/AIDS. Pada penelitian ini digunakan data HIV/AIDS yang diambil dari RSUD Balung.

b. Graf Irisan

Pembentukan graf irisan dengan melakukan pengamatan pada irisan data persegi panjang yang membentuk *inner rectangle* untuk menentukan *maximal intersection*.

c. Matriks Insidensi

Dilakukan pembentukan matrik dari graf irisan.

d. *Maximal clique*

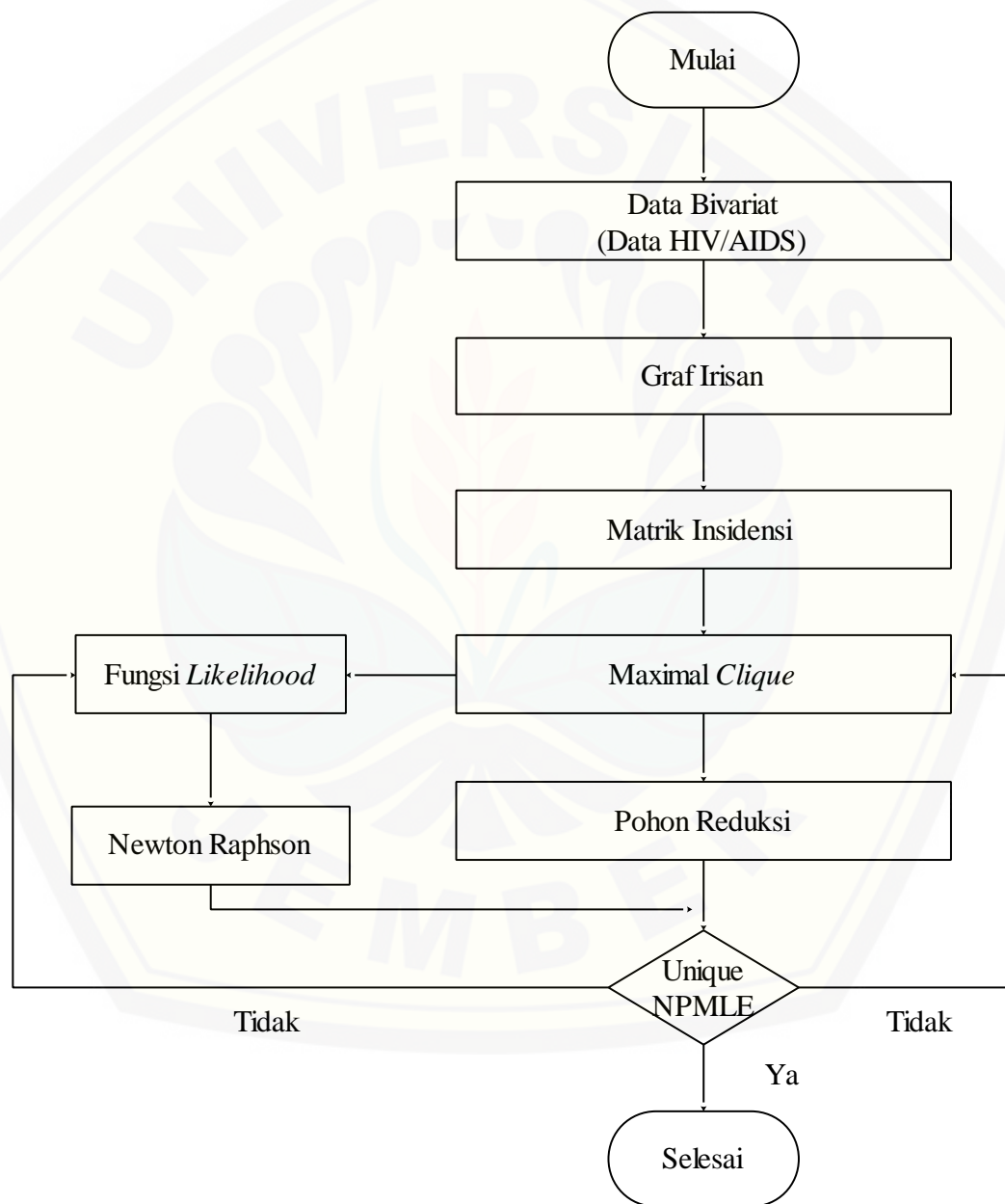
Menentukan jumlah *maximal clique* untuk mendapatkan matriks *clique*.

e. Fungsi *Likelihood* atau Pohon Reduksi

Setelah mendapatkan *maximal clique*, dapat menggunakan 2 cara untuk mencari NPMLE yaitu membentuk fungsi *likelihood* kemudian mencari penaksir dengan metode Newton Raphson atau melakukan perhitungan dengan membentuk pohon reduksi.

f. *Unique* NPMLE

g. Selesai



Gambar 3.1 Skema metode penelitian

BAB 5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dan pembahasan diperoleh kesimpulan sebagai berikut:

- a. Matriks insidensi yang diperoleh dari graf irisan hasil irisan persegi panjang berbentuk matriks segitiga atas. Matriks insidensi yang digunakan pada perhitungan estimator pada pohon reduksi berupa matriks *clique*, yaitu matriks penyederhanaan dari matriks insidensi.
- b. Perhitungan estimator yang menyatakan peluang waktu masa inkubasi pada masing-masing daerah irisan dengan menggunakan pohon reduksi menghasilkan nilai $\hat{p} = (\text{---})$ dengan nilai fungsi likelihood dan pada perhitungan menggunakan metode Newton Raphson dengan pemberian nilai awal $p_1=1, p_2=1, p_3=1, p_4=1, p_5=1, p_6=1, p_7=1, p_8=1, p_9=1, p_{10}=1, p_{11}=1$ dan $b=1$ dan galat 0,00001 menghasilkan nilai $p_1=0, p_2=0, p_3=0, p_4=0, p_5=0, p_6=0, p_7=1, p_8=0, p_9=0, p_{10}=0, p_{11}=0$ dan nilai fungsi *likelihood* 0. Berdasarkan nilai fungsi *likelihood* dapat disimpulkan bahwa perhitungan NPMLE dengan menggunakan pohon reduksi lebih baik dibandingkan metode Newton Raphson, karena pohon reduksi memiliki nilai fungsi *likelihood* yang lebih besar.

5.2 Saran

Pada penelitian ini digunakan data tersensor interval. Pada penelitian selanjutnya bisa menggunakan data tersensor lainnya atau data terpotong. Dalam melakukan perhitungan NPMLE digunakan metode Newton Raphson dan pohon reduksi. Untuk penelitian selanjutnya dapat digunakan metode lain untuk mencari estimator NPMLE.

DAFTAR PUSTAKA

- Anton, H. 2000. *Dasar-Dasar Aljabar Linier* . Batam: Interaksara.
- Anton, H. 2004 . *Aljabar Linier Elementer Edisi ke-8*. Jakarta: Erlangga.
- Atkinson, K.E. 1989. *An Introduction to Numerical Analysis*. Second Edition. New York: John Willey & Sons, Inc.
- Bogaerts, K. dan E. Lesaffre. 2004. A New, Fast Algorithm to Find the Regions of Possible Support for Bivariate Interval-Censored Data. *Journal of Computational and Graphical Statistics*. 13(2): 330-340.
- Fatekurohman, M., Subanar dan Danardono. 2014. *An Algorithm of Nonparametric Maximum Likelihood Estimation for Bivariate Censored Data*. Yogyakarta: Departmen of Mathematics Education, Yogyakarta State University.
- Fulkerson, D.R. dan O.A. Gross. 1965. Incidence Matrices and Interval Graphs. *Pacific J. Math*. 15(3): 835-855.
- Gentleman, R. dan A.C. Vandal. 1999. Computational Algorithm for Censored Data Problems Using Intersection Graphs. *J. Comp. Graphical Statist*.
- Gentleman, R. dan A.C. Vandal. 2002. Nonparametric estimation of the bivariate CDF for arbitrarily censored data. *The Canadian Journal of Statistics*. 30(4): 557-571.
- Kleinbum, D.G dan M. Klein. 2005. *Survival Analysis a Self-Learning Text Second Edition*. New York: Springer.
- Klein, J.P. dan M.L. Moeschberger. 1997. *Survival Analysis: Techniques For Censored and Tuncated Data*. New York: Springer-Verlag.

- Lee E.T. dan Wang, J.W. 2002. *Statistical Methods for Survival Data Analysis Third Edition*. New Jersey : John Wiley and Sons.
- Luknanto, D. 2000. *Pengantar Optimasi Nonlinier*. Yogyakarta: Universitas Gajah Mada.
- Liu, X. 2005. *Nonparametric estimation bivariate censored data a discrete of the approach*. A thesis submitted to McGill University in partial fulfilment of the requirements degree of Doctor of Philosophy Department of Mathematics and Statistics McGill University, Montreal, Canada.
- Maathuis, M. H. 2003. *Nonparametric Maximum Likelihood Estimation for Bivariate Censored Data*. Master's thesis. Delft: Delft University of Technology.
- Maathuis, M. H. 2005. Reduction Algorithm for NPMLE for the Distribution Function of Bivariate Interval Censored Data. *J Comp Graph Statist*. 14: 252-262.
- Maathuis, M. H. 2007. *Survival Analysis for Interval Censored Data*. Department of Statistics. Seattle: University of Wasingthon.
- Murti, B. 1997. *Prinsip dan metode riset epidemiologi*. Yogyakarta : Gajah mada University press.
- Munir, R. 2010. *Matematika Diskrit*. Bandung: Informatika Bandung.
- Pyke, D.A. dan J.N. Thompson. 1986. *Statistical Analysis of Survival and Removed Rate Experiments*, *Ecology*. 67(1): 240-245.
- Sirait, H. dan R. Efendi. 2013. *Penaksir Maksimum Likelihood dengan Metode Iterasi Newton Raphson*. Lampung : Program Studi Matematika FMIPA Universitas Riau.

Widana. I. N, N.N.R. Utami dan N. M. Asih. 2013. Perbandingan Solusi Sistem Persamaan Nonlinier Menggunakan Metode Newton Raphson dan Metode Jacobian, *E-jurnal Matematika*. 2(2): 11-17.

Wong. G. Y. dan Q, Yu. 1999. Generalized MLE of a Joint Distribution Function with Multivariate Interval-Censored Data, *Journal of Multivariate Analysis*. Vol **69**:155-166.



LAMPIRAN

Lampiran A. Data pasien HIV/AIDS RSUD Balung

No.	Nama	Umur	Jenis Kelamin	Awal Masuk	Stadium	INTERVAL HIV	MASA INKUBASI	INTERVAL AIDS
1	YNS	30th	L	26 April 2010	Stadium 3	1999-2002	2002-2010	2010-2011
2	AH	53th	L	07 Februari 2013	Stadium 3	2004-2007	2007-2013	2013-2014
3	SDH	33th	P	21 April 2011	Stadium 3	2000-2003	2003-2011	2011-2015
4	IBR	39th	L	20 Januari 2014	Stadium 3	2004-2007	2007-2014	2014-2015
5	SNA	36th	P	10 Agustus 2016	Stadium 3	2006-2009	2009-2016	2016-2017
6	AQ	31th	L	05 Februari 2014	Stadium 3	2003-2006	2006-2014	2014-2017
7	NR	36th	P	11 Juni 2012	Stadium 3	2002-2005	2005-2012	2012-2014
8	HNK	31th	P	25 Nopember 2011	Stadium 3	2000-2003	2003-2011	2011-2013
9	RSD	28th	P	20 Desember 2012	Stadium 3	2001-2004	2004-2012	2012-2013
10	NTY	54th	P	04 Agustus 2014	Stadium 3	2005-2008	2008-2014	2014-2015
11	SA	40th	L	12 Nopember 2012	Stadium 3	2002-2005	2005-2012	2012-2017
12	SK	43th	L	10 Maret 2012	Stadium 3	2002-2005	2005-2012	2012-2013
13	AH	31th	L	27 Juni 2011	Stadium 3	2000-2003	2003-2011	2011-2013
14	SDH	33th	P	21 April 2011	Stadium 3	2000-2003	2003-2011	2011-2014
15	HNK	31th	P	08 Desember 2011	Stadium 3	2000-2003	2003-2011	2011-2013
16	SM	61th	L	02 Maret 2016	Stadium 3	2008-2011	2011-2016	2016-2017
17	ALA	25th	L	23 Mei 2016	Stadium 3	2005-2008	2008-2016	2016-2017
18	SR	43th	L	13 Oktober 2016	Stadium 3	2006-2009	2009-2016	2016-2017
19	SD	64th	L	10 Februari 2016	Stadium 3	2008-2011	2011-2016	2016-2017
20	ST	48th	L	14 Juni 2016	Stadium 3	2006-2009	2009-2016	2016-2017

No.	Nama	Umur	Jenis Kelamin	Awal Masuk	Stadium	INTERVAL HIV	MASA INKUBASI	INTERVAL AIDS
21	TB	51th	L	21 Oktober 2016	Stadium 3	2007-2010	2010-2016	2016-2017
22	TS	48th	P	20 Oktober 2014	Stadium 3	2004-2007	2007-2014	2014-2015
23	WH	25th	P	18 Desember 2015	Stadium 3	2004-2007	2007-2015	2015-2017
24	YE	42th	P	26 Juli 2016	Stadium 3	2006-2009	2009-2016	2016-2017
25	SYF	26th	P	13 Juli 2016	Stadium 3	2005-2008	2008-2016	2016-2017
26	MN	27th	L	28 Oktober 2015	Stadium 3	2004-2007	2007-2015	2015-2016
27	SA	40th	L	12 Nopember 2012	Stadium 3	2002-2005	2005-2012	2012-2014
28	HR	41th	L	24 Nopember 2011	Stadium 3	2001-2004	2004-2011	2011-2017
29	RB	1th	L	31 Oktober 2016	Stadium 3	2012-2015	2015-2016	2016-2017
30	EDS	30th	L	13 Mei 2015	Stadium 3	1998-2001	2001-2014	2014-2017

Dua puluh satu data pasien HIV/AIDS

No.	Nama	Umur	Jenis Kelamin	Awal Masuk	Stadium	INTERVAL HIV	MASA INKUBASI	INTERVAL AIDS
1	YNS	30th	L	26 April 2010	Stadium 3	1999-2002	2002-2010	2010-2011
2	AH	53th	L	07 Februari 2013	Stadium 3	2004-2007	2007-2013	2013-2014
3	SDH	33th	P	21 April 2011	Stadium 3	2000-2003	2003-2011	2011-2015
4	IBR	39th	L	20 Januari 2014	Stadium 3	2004-2007	2007-2014	2014-2015
5	SNA	36th	P	10 Agustus 2016	Stadium 3	2006-2009	2009-2016	2016-2017
6	AQ	31th	L	05 Februari 2014	Stadium 3	2003-2006	2006-2014	2014-2017
7	NR	36th	P	11 Juni 2012	Stadium 3	2002-2005	2005-2012	2012-2014
8	HNK	31th	P	25 Nopember 2011	Stadium 3	2000-2003	2003-2011	2011-2013
9	RSD	28th	P	20 Desember 2012	Stadium 3	2001-2004	2004-2012	2012-2013
10	NTY	54th	P	04 Agustus 2014	Stadium 3	2005-2008	2008-2014	2014-2015
11	SA	40th	L	12 Nopember 2012	Stadium 3	2002-2005	2005-2012	2012-2017
12	SK	43th	L	10 Maret 2012	Stadium 3	2002-2005	2005-2012	2012-2013
13	SDH	33th	P	21 April 2011	Stadium 3	2000-2003	2003-2011	2011-2014
14	SM	61th	L	02 Maret 2016	Stadium 3	2008-2011	2011-2016	2016-2017
15	ALA	25th	L	23 Mei 2016	Stadium 3	2005-2008	2008-2016	2016-2017
16	TB	51th	L	21 Oktober 2016	Stadium 3	2007-2010	2010-2016	2016-2017
17	WH	25th	P	18 Desember 2015	Stadium 3	2004-2007	2007-2015	2015-2017
18	MN	27th	L	28 Oktober 2015	Stadium 3	2004-2007	2007-2015	2015-2016
19	HR	41th	L	24 Nopember 2011	Stadium 3	2001-2004	2004-2011	2011-2017
20	RB	1th	L	31 Oktober 2016	Stadium 3	2012-2015	2015-2016	2016-2017
21	EDS	30th	L	13 Mei 2015	Stadium 3	1998-2001	2001-2014	2014-2017

Lampiran C. M-file penyelesaian NPMLE menggunakan metode Newton Raphson

```

clc;
clear all;

z=0;
syms p1 p2 p3 p4 p5 p6 p7 p8 p9 p10 p11 p12 b L
f=input('Fungsi tujuan=');
h1=input('Fungsi kendala pertama=');
tol=0.00001;
disp('Maka diperoleh pengali lagrange sebagai berikut')
L=f+b*h1
dp1=diff(L,p1);
dp2=diff(L,p2);
dp3=diff(L,p3);
dp4=diff(L,p4);
dp5=diff(L,p5);
dp6=diff(L,p6);
dp7=diff(L,p7);
dp8=diff(L,p8);
dp9=diff(L,p9);
dp10=diff(L,p10);
dp11=diff(L,p11);
dp12=diff(L,p12);
db=diff(L,b);
disp('Diperoleh Vektor Gradien:')
G=[dp1; dp2; dp3; dp4; dp5; dp6; dp7; dp8; dp9; dp10;
dp11; dp12; db]
gradf1=[diff(dp1,'p1') diff(dp1,'p2') diff(dp1,'p3')
diff(dp1,'p4') diff(dp1,'p5') diff(dp1,'p6')
diff(dp1,'p7') diff(dp1,'p8') diff(dp1,'p9')
diff(dp1,'p10') diff(dp1,'p11') diff(dp1,'p12')
diff(dp1,'b')];
gradf2=[diff(dp2,'p1') diff(dp2,'p2') diff(dp2,'p3')
diff(dp2,'p4') diff(dp2,'p5') diff(dp2,'p6')
diff(dp2,'p7') diff(dp2,'p8') diff(dp2,'p9')
diff(dp2,'p10') diff(dp2,'p11') diff(dp2,'p12')
diff(dp2,'b')];
gradf3=[diff(dp3,'p1') diff(dp3,'p2') diff(dp3,'p3')
diff(dp3,'p4') diff(dp3,'p5') diff(dp3,'p6')
diff(dp3,'p7') diff(dp3,'p8') diff(dp3,'p9')
diff(dp3,'p10') diff(dp3,'p11') diff(dp3,'p12')
diff(dp3,'b')];
gradf4=[diff(dp4,'p1') diff(dp4,'p2') diff(dp4,'p3')
diff(dp4,'p4') diff(dp4,'p5') diff(dp4,'p6')

```

```
diff(dp4, 'p7') diff(dp4, 'p8') diff(dp4, 'p9')
diff(dp4, 'p10') diff(dp4, 'p11') diff(dp4, 'p12')
diff(dp4, 'b')];
gradf5=[diff(dp5, 'p1') diff(dp5, 'p2') diff(dp5, 'p3')
diff(dp5, 'p4') diff(dp5, 'p5') diff(dp5, 'p6')
diff(dp5, 'p7') diff(dp5, 'p8') diff(dp5, 'p9')
diff(dp5, 'p10') diff(dp5, 'p11') diff(dp5, 'p12')
diff(dp5, 'b')];
gradf6=[diff(dp6, 'p1') diff(dp6, 'p2') diff(dp6, 'p3')
diff(dp6, 'p4') diff(dp6, 'p5') diff(dp6, 'p6')
diff(dp6, 'p7') diff(dp6, 'p8') diff(dp6, 'p9')
diff(dp6, 'p10') diff(dp6, 'p11') diff(dp6, 'p12')
diff(dp6, 'b')];
gradf7=[diff(dp7, 'p1') diff(dp7, 'p2') diff(dp7, 'p3')
diff(dp7, 'p4') diff(dp7, 'p5') diff(dp7, 'p6')
diff(dp7, 'p7') diff(dp7, 'p8') diff(dp7, 'p9')
diff(dp7, 'p10') diff(dp7, 'p11') diff(dp7, 'p12')
diff(dp7, 'b')];
gradf8=[diff(dp8, 'p1') diff(dp8, 'p2') diff(dp8, 'p3')
diff(dp8, 'p4') diff(dp8, 'p5') diff(dp8, 'p6')
diff(dp8, 'p7') diff(dp8, 'p8') diff(dp8, 'p9')
diff(dp8, 'p10') diff(dp8, 'p11') diff(dp8, 'p12')
diff(dp8, 'b')];
gradf9=[diff(dp9, 'p1') diff(dp9, 'p2') diff(dp9, 'p3')
diff(dp9, 'p4') diff(dp9, 'p5') diff(dp9, 'p6')
diff(dp9, 'p7') diff(dp9, 'p8') diff(dp9, 'p9')
diff(dp9, 'p10') diff(dp9, 'p11') diff(dp9, 'p12')
diff(dp9, 'b')];
gradf10=[diff(dp10, 'p1') diff(dp10, 'p2')
diff(dp10, 'p3') diff(dp10, 'p4') diff(dp10, 'p5')
diff(dp10, 'p6') diff(dp10, 'p7') diff(dp10, 'p8')
diff(dp10, 'p9') diff(dp10, 'p10') diff(dp10, 'p11')
diff(dp10, 'p12') diff(dp10, 'b')];
gradf11=[diff(dp11, 'p1') diff(dp11, 'p2')
diff(dp11, 'p3') diff(dp11, 'p4') diff(dp11, 'p5')
diff(dp11, 'p6') diff(dp11, 'p7') diff(dp11, 'p8')
diff(dp11, 'p9') diff(dp11, 'p10') diff(dp11, 'p11')
diff(dp11, 'p12') diff(dp11, 'b')];
gradf12=[diff(dp12, 'p1') diff(dp12, 'p2')
diff(dp12, 'p3') diff(dp12, 'p4') diff(dp12, 'p5')
diff(dp12, 'p6') diff(dp12, 'p7') diff(dp12, 'p8')
diff(dp12, 'p9') diff(dp12, 'p10') diff(dp12, 'p11')
diff(dp12, 'p12') diff(dp12, 'b')];
gradf13=[diff(db, 'p1') diff(db, 'p2') diff(db, 'p3')
diff(db, 'p4') diff(db, 'p5') diff(db, 'p6') diff(db, 'p7')
diff(db, 'p8') diff(db, 'p9') diff(db, 'p10')
diff(db, 'p11') diff(db, 'p12') diff(db, 'b')];
```



```
disp('Diperoleh Matriks Hessian:')
H=[gradf1; gradf2; gradf3; gradf4; gradf5; gradf6;
gradf7; gradf8; gradf9; gradf10; gradf11; gradf12;
gradf13]
p1=input('Masukkan nilai awal p1=');
p2=input('Masukkan nilai awal p2=');
p3=input('Masukkan nilai awal p3=');
p4=input('Masukkan nilai awal p4=');
p5=input('Masukkan nilai awal p5=');
p6=input('Masukkan nilai awal p6=');
p7=input('Masukkan nilai awal p7=');
p8=input('Masukkan nilai awal p8=');
p9=input('Masukkan nilai awal p9=');
p10=input('Masukkan nilai awal p10=');
p11=input('Masukkan nilai awal p11=');
p12=input('Masukkan nilai awal p12=');
b=input('Masukkan nilai awal b=');
z0=[p1; p2; p3; p4; p5; p6; p7; p8; p9; p10; p11; p12;
b];
iterasi=0;
eps=1;
err=1;
while err>tol
    iterasi=iterasi+1
    disp('Vektor gradien dititik awal')
    T=eval(G)
    disp('Matriks Hessian dititik awal')
    R=eval(H)
    disp('Invers dari matriks Hessian dititik awal')
    P=inv(R)
    disp('Proses iterasi')
    Z=z0-(P*T)
    p1=Z(1,1)
    p2=Z(2,1)
    p3=Z(3,1)
    p4=Z(4,1)
    p5=Z(5,1)
    p6=Z(6,1)
    p7=Z(7,1)
    p8=Z(8,1)
    p9=Z(9,1)
    p10=Z(10,1)
    p11=Z(11,1)
    p12=Z(12,1)
    eps=(abs(Z-z0));
    err=norm(eps);
    z0=Z;
```

```

p1=z0(1); p2=z0(2); p3=z0(3); p4=z0(4); p5=z0(5);
p6=z0(6); p7=z0(7); p8=z0(8); p9=z0(9); p10=z0(10);
p11=z0(11); p12=z0(12); b=z0(13);
end

```

Lampiran D. Output Program

```

Fungsi tujuan=(p1+p2+p3+p6)*(p2+p3+p4+p6)*(p3+p4+p6+p8)*p4*p5*(p6+p8+p9+p10)*(p7+p11)*
(p1+p8+p9+p10)*(p1+p3+p9)*(p10+p11)*(p6+p8+p11)*p12
Fungsi kendala pertama=1-(p1+p2+p3+p4+p5+p6+p7+p8+p9+p10+p11+p12)
Maka diperoleh pengali lagrange sebagai berikut

```

L =

```

p4*p5*p12*(p7 + p11)*(p10 + p11)*(p1 + p3 + p9)*(p6 + p8 + p11)*(p1 + p2 + p3 + p6)*
(p2 + p3 + p4 + p6)*(p3 + p4 + p6 + p8)*(p1 + p8 + p9 + p10)*(p6 + p8 + p9 + p10) - b*
(p1 + p2 + p3 + p4 + p5 + p6 + p7 + p8 + p9 + p10 + p11 + p12 - 1)

```

Diperoleh Vektor Gradien:

G =

```

p4*p5*p12*(p7 + p11)*(p10 + p11)*(p1 + p3 + p9)*(p6 + p8 + p11)*(p1 + p2 + p3 + p6)*
(p2 + p3 + p4 + p6)*(p3 + p4 + p6 + p8)*(p6 + p8 + p9 + p10) - b + p4*p5*p12*(p7 +
p11)*(p10 + p11)*(p1 + p3 + p9)*(p6 + p8 + p11)*(p2 + p3 + p4 + p6)*(p3 + p4 + p6 +
p8)*(p1 + p8 + p9 + p10)*(p6 + p8 + p9 + p10) + p4*p5*p12*(p7 + p11)*(p10 + p11)*(p6 +
p8 + p11)*(p1 + p2 + p3 + p6)*(p2 + p3 + p4 + p6)*(p3 + p4 + p6 + p8)*(p1 + p8 + p9 +
p10)*(p6 + p8 + p9 + p10)

```

```

p4*p5*p12*(p7 + p11)*(p10 + p11)*(p1 + p3 + p9)*(p6 + p8 + p11)*(p1 + p2 + p3 + p6)*
(p3 + p4 + p6 + p8)*(p1 + p8 + p9 + p10)*(p6 + p8 + p9 + p10) - b + p4*p5*p12*(p7 +
p11)*(p10 + p11)*(p1 + p3 + p9)*(p6 + p8 + p11)*(p2 + p3 + p4 + p6)*(p3 + p4 + p6 +
p8)*(p1 + p8 + p9 + p10)*(p6 + p8 + p9 + p10)

```

```

p4*p5*p12*(p7 + p11)*(p10 + p11)*(p1 + p3 + p9)*(p6 + p8 + p11)*(p1 + p2 + p3 + p6)*
(p2 + p3 + p4 + p6)*(p1 + p8 + p9 + p10)*(p6 + p8 + p9 + p10) - b + p4*p5*p12*(p7 +
p11)*(p10 + p11)*(p1 + p3 + p9)*(p6 + p8 + p11)*(p1 + p2 + p3 + p6)*(p3 + p4 + p6 +
p8)*(p1 + p8 + p9 + p10)*(p6 + p8 + p9 + p10) + p4*p5*p12*(p7 + p11)*(p10 + p11)*(p1 +
p3 + p9)*(p6 + p8 + p11)*(p2 + p3 + p4 + p6)*(p3 + p4 + p6 + p8)*(p1 + p8 + p9 + p10)*
(p6 + p8 + p9 + p10) + p4*p5*p12*(p7 + p11)*(p10 + p11)*(p6 + p8 + p11)*(p1 + p2 + p3
+ p6)*(p2 + p3 + p4 + p6)*(p3 + p4 + p6 + p8)*(p1 + p8 + p9 + p10)*(p6 + p8 + p9 +
p10)

```

```

p4*p5*p12*(p7 + p11)*(p10 + p11)*(p1 + p3 + p9)*(p6 + p8 + p11)*(p1 + p2 + p3 + p6)*
(p2 + p3 + p4 + p6)*(p1 + p8 + p9 + p10)*(p6 + p8 + p9 + p10) - b + p4*p5*p12*(p7 +
p11)*(p10 + p11)*(p1 + p3 + p9)*(p6 + p8 + p11)*(p1 + p2 + p3 + p6)*(p3 + p4 + p6 +
p8)*(p1 + p8 + p9 + p10)*(p6 + p8 + p9 + p10) + p5*p12*(p7 + p11)*(p10 + p11)*(p1 + p3
+ p9)*(p6 + p8 + p11)*(p1 + p2 + p3 + p6)*(p2 + p3 + p4 + p6)*(p3 + p4 + p6 + p8)*(p1
+ p8 + p9 + p10)*(p6 + p8 + p9 + p10)

```

⋮

Masukkan nilai awal p1=1
 Masukkan nilai awal p2=1
 Masukkan nilai awal p3=1
 Masukkan nilai awal p4=1
 Masukkan nilai awal p5=1
 Masukkan nilai awal p6=1
 Masukkan nilai awal p7=1
 Masukkan nilai awal p8=1
 Masukkan nilai awal p9=1
 Masukkan nilai awal p10=1
 Masukkan nilai awal p11=1
 Masukkan nilai awal p12=1
 Masukkan nilai awal b=1

iterasi =

1

Vektor gradien dititik awal

T =

30719
 18431
 39935
 55295
 36863
 49151
 18431
 39935
 30719
 36863
 49151
 36863
 -11

Matriks Hessian dititik awal

R =

	16896	13056	26880	46080	30720	38656	15360	✓
30976	19200	28416	40960	30720	-1			
	13056	4608	15360	25344	18432	19968	9216	✓
19968	15360	18432	24576	18432	-1			
	26880	15360	32256	55296	39936	46336	19968	✓
40960	29184	39936	53248	39936	-1			
	46080	25344	55296	41472	55296	69120	27648	✓
57600	46080	55296	73728	55296	-1			
	30720	18432	39936	55296	0	49152	18432	✓
39936	30720	36864	49152	36864	-1			
	38656	19968	46336	69120	49152	52224	24576	✓
44544	38656	46848	61440	49152	-1			
	15360	9216	19968	27648	18432	24576	0	✓
19968	15360	18432	15360	18432	-1			
	30976	19968	40960	57600	39936	44544	19968	✓
32256	28672	35328	49152	39936	-1			
	19200	15360	29184	46080	30720	38656	15360	✓
28672	16896	26112	40960	30720	-1			
	28416	18432	39936	55296	36864	46848	18432	✓
35328	26112	23040	39936	36864	-1			
	40960	24576	53248	73728	49152	61440	15360	✓
49152	40960	39936	43008	49152	-1			
	30720	18432	39936	55296	36864	49152	18432	✓
39936	30720	36864	49152	0	-1			
	-1	-1	-1	-1	-1	-1	-1	✓
-1	-1	-1	-1	-1	0			

Invers dari matriks Hessian dititik awal

P =

```

1.0e+03 *
-0.0000  0.0000  0.0000 -0.0000  0.0000 -0.0000 -0.0000  0.0000 ✓
0.0000 -0.0000  0.0000  0.0000  0.0007
  0.0000 -0.0000 -0.0000 -0.0000 -0.0000  0.0000  0.0000 -0.0000 ✓
-0.0000  0.0000 -0.0000 -0.0000 -0.0017
  0.0000 -0.0000 -0.0000  0.0000  0.0000  0.0000  0.0000 -0.0000 ✓
0.0000 -0.0000 -0.0000  0.0000 -0.0002
-0.0000 -0.0000  0.0000 -0.0000  0.0000  0.0000  0.0000 -0.0000 ✓
-0.0000  0.0000 -0.0000  0.0000  0.0000  0.0000
  0.0000 -0.0000  0.0000  0.0000 -0.0000  0.0000  0.0000 -0.0000 ✓
0.0000  0.0000  0.0000  0.0000  0.0000
-0.0000  0.0000  0.0000  0.0000  0.0000 -0.0000 -0.0000  0.0000 ✓
0.0000 -0.0000  0.0000  0.0000  0.0013
-0.0000  0.0000  0.0000  0.0000  0.0000  0.0000 -0.0000 -0.0000 ✓
0.0000 -0.0000  0.0000  0.0000  0.0001
  0.0000 -0.0000 -0.0000 -0.0000 -0.0000  0.0000  0.0000 -0.0000 ✓
0.0000  0.0000 -0.0000 -0.0000 -0.0010
  0.0000 -0.0000  0.0000 -0.0000  0.0000  0.0000  0.0000  0.0000 ✓
-0.0000  0.0000 -0.0000  0.0000 -0.0004
-0.0000  0.0000 -0.0000  0.0000  0.0000  0.0000 -0.0000 -0.0000 ✓
0.0000 -0.0000  0.0000  0.0000  0.0003
  0.0000 -0.0000 -0.0000 -0.0000  0.0000  0.0000  0.0000 -0.0000 ✓
-0.0000  0.0000 -0.0000  0.0000 -0.0002
  0.0000 -0.0000  0.0000  0.0000  0.0000  0.0000  0.0000 -0.0000 ✓
0.0000  0.0000  0.0000 -0.0000  0.0000
  0.0007 -0.0017 -0.0002  0.0000  0.0000  0.0000  0.0013  0.0001 -0.0010 ✓
-0.0004  0.0003 -0.0002  0.0000  1.4799

```

Proses iterasi

Z =

```

1.0e+04 *
  0.0008
-0.0016
-0.0001
  0.0001
  0.0001
  0.0014
  0.0002
-0.0009
-0.0003
  0.0003
-0.0001
  0.0001
  1.4665

```

p1 =

```

  7.7080

```

p2 =

```

-15.6302

```

p3 =
-0.7786

p4 =
1.3431

p5 =
0.9453

p6 =
14.0730

p7 =
1.8735

p8 =
-9.2652

p9 =
-2.9002

p10 =
3.4647

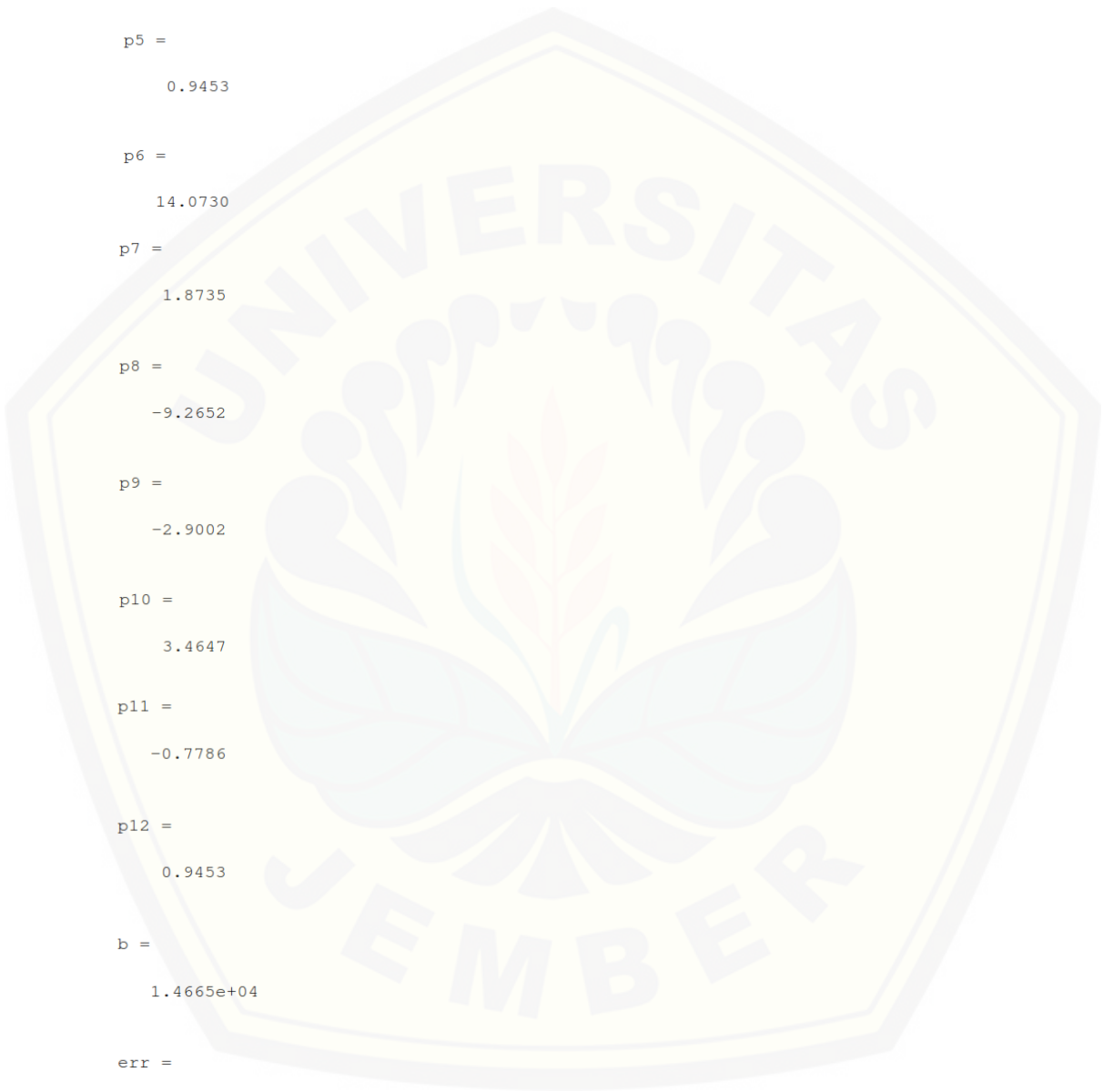
p11 =
-0.7786

p12 =
0.9453

b =
1.4665e+04

err =
1.4664e+04

:



```

iterasi =

    119

Vektor gradien dititik awal

T =

    1.0e-17 *
-0.0000
-0.0000
-0.0000
-0.0000
 0.0000
-0.0000
 0.0000
-0.0000
-0.0000
-0.0000
 0.0000
 0.0000
-0.3954

Matriks Hessian dititik awal

R =

-0.0000    0.0000    0.0000    0.0000   -0.0000    0.0000   -0.0000   -0.0000 ✓
-0.0000   -0.0000   -0.0000   -0.0000   -1.0000
 0.0000   -0.0000   -0.0000   -0.0000   -0.0000   -0.0000   -0.0000    0.0000 ✓
0.0000    0.0000   -0.0000   -0.0000   -1.0000
 0.0000   -0.0000   -0.0000   -0.0000   -0.0000   -0.0000   -0.0000    0.0000 ✓
0.0000    0.0000   -0.0000   -0.0000   -1.0000
 0.0000   -0.0000   -0.0000   -0.0000   -0.0000   -0.0000   -0.0000    0.0000 ✓
0.0000    0.0000   -0.0000   -0.0000   -1.0000
-0.0000   -0.0000   -0.0000   -0.0000    0        -0.0000    0.0000   -0.0000 ✓
-0.0000   -0.0000    0.0000    0.0000   -1.0000
 0.0000   -0.0000   -0.0000   -0.0000   -0.0000   -0.0000   -0.0000    0.0000 ✓
0.0000    0.0000   -0.0000   -0.0000   -1.0000
-0.0000   -0.0000   -0.0000   -0.0000    0.0000   -0.0000    0        -0.0000 ✓
-0.0000   -0.0000    0.0000    0.0000   -1.0000
-0.0000    0.0000    0.0000    0.0000   -0.0000    0.0000   -0.0000   -0.0000 ✓
-0.0000   -0.0000   -0.0000   -0.0000   -1.0000
-0.0000    0.0000    0.0000    0.0000   -0.0000    0.0000   -0.0000   -0.0000 ✓
-0.0000   -0.0000   -0.0000   -0.0000   -1.0000
-0.0000   -0.0000   -0.0000   -0.0000   -0.0000    0.0000   -0.0000   -0.0000 ✓
-0.0000   -0.0000    0.0000    0        -1.0000
-1.0000   -1.0000   -1.0000   -1.0000   -1.0000   -1.0000   -1.0000   -1.0000 ✓
-1.0000   -1.0000   -1.0000   -1.0000   -1.0000   -1.0000   -1.0000   -1.0000 ✓
-1.0000   -1.0000   -1.0000   -1.0000    0

Invers dari matriks Hessian dititik awal
Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND
= 1.193294e-48.
> In duabelas (line 65)

P =

    1.0e+46 *

```


-0.3857	0.1672	0.1660	-0.0125	0.0941	-0.3223	-0.3200	0.2269	✓
0.2633	-0.1099	0.1389	0.0941	0.0000				
0.1672	-0.3776	0.0502	-0.0017	-0.1866	0.3336	0.6542	-0.2087	✓
-0.2226	0.2710	-0.2925	-0.1866	-0.0000				
0.1660	0.0502	-0.3817	0.0258	-0.0080	0.3049	0.0011	-0.2426	✓
0.1308	-0.0537	0.0152	-0.0080	-0.0000				
-0.0125	-0.0017	0.0258	-0.0244	0.0175	-0.0007	-0.0318	0.0101	✓
-0.0052	0.0066	-0.0012	0.0175	0.0000				
0.0941	-0.1866	-0.0080	0.0175	-1.0217	0.1706	0.9290	-0.1100	✓
-0.0335	0.0430	-0.0080	0.1135	0.0000				
-0.3223	0.3336	0.3049	-0.0007	0.1706	-0.6438	-0.6070	0.4450	✓
0.0964	-0.2252	0.2778	0.1706	0.0000				
-0.3200	0.6542	0.0011	-0.0318	0.9290	-0.6070	-2.1976	0.3661	✓
0.3046	-0.3353	0.3077	0.9290	-0.0000				
0.2269	-0.2087	-0.2426	0.0101	-0.1100	0.4450	0.3661	-0.4247	✓
0.0459	0.1545	-0.1524	-0.1100	-0.0000				
0.2633	-0.2226	0.1308	-0.0052	-0.0335	0.0964	0.3046	0.0459	✓
-0.5720	0.2647	-0.2390	-0.0335	-0.0000				
-0.1099	0.2710	-0.0537	0.0066	0.0430	-0.2252	-0.3353	0.1545	✓
0.2647	-0.3118	0.2529	0.0430	0.0000				
0.1389	-0.2925	0.0152	-0.0012	-0.0080	0.2778	0.3077	-0.1524	✓
-0.2390	0.2529	-0.2915	-0.0080	-0.0000				
0.0941	-0.1866	-0.0080	0.0175	0.1135	0.1706	0.9290	-0.1100	✓
-0.0335	0.0430	-0.0080	-1.0217	0.0000				
0.0000	-0.0000	-0.0000	0.0000	0.0000	0.0000	-0.0000	-0.0000	✓
-0.0000	0.0000	-0.0000	0.0000	0.0000				

Proses iterasi

Z =

0.0000
 -0.0000
 -0.0000
 0.0000
 0.0000
 0.0000
 1.0000
 -0.0000
 -0.0000
 0.0000
 -0.0000
 0.0000
 -0.0000

p1 =

2.1057e-05

p2 =

-4.1764e-05

p3 =

-1.7874e-06

```
p4 =  
    3.9236e-06  
  
p5 =  
    2.5415e-05  
  
p6 =  
    3.8190e-05  
  
p7 =  
    1.0000  
p8 =  
   -2.4631e-05  
p9 =  
   -7.4984e-06  
p10 =  
    9.6346e-06  
p11 =  
   -1.7874e-06  
p12 =  
    2.5415e-05  
b =  
   -7.0251e-57  
err =  
    9.8423e-06  
>>
```