TESIS SS14 2501

METODE *FAST DOUBLE BOOTSTRAP* PADA REGRESI SPASIAL DATA PANEL DENGAN *SPATIAL FIXED EFFECT*

(Studi Kasus : Pemodelan Penduduk Miskin di Provinsi NTB)

NORA MUHTASIB
1313 201 709

DOSEN PEMBIMBING:
Dr. Sutikno, S.Si., M.Si.

PROGRAM MAGISTER
JURUSAN STATISTIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2015

TESIS – SS14 2501

FAST DOUBLE BOOTSTRAP METHOD ON SPATIAL PANEL DATA REGRESSION WITH *SPATIAL FIXED EFFECT*

(Case Study: Modelling of Poverty in NTB Province)

NORA MUHTASIB
1313 201 709

SUPERVISOR:
Dr. Sutikno, S.Si., M.Si.

PROGRAM OF MAGISTER
DEPARTMENT OF STATISTICS
FACULTY OF MATHEMATICS AND NATURAL SCIENCE
INSTITUTE OF TECHNOLOGY SEPULUH NOPEMBER
SURABAYA
2015

METODE *FAST DOUBLE BOOTSTRAP* PADA REGRESI SPASIAL DATA PANEL DENGAN *SPATIAL FIXED EFFECT*
(Studi Kasus : Pemodelan Persentase Penduduk Miskin di Provinsi Nusa Tenggara Barat)

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Sains (M.Si)
di
Institut Teknologi Sepuluh November
Oleh :

NORA MUHTASIB
NRP. 1313 201 709

Tanggal Ujian : 13 Februari 2015
Periode Wisuda : September 2015

Disetujui Oleh:

1. **Dr. Sutikno, S.Si, M.Si.** **(Pembimbing)**
   NIP. 19710313 199702 2 001

2. **Dr. Heru Margono, M.Sc.** **(Penguji)**
   NIP. 19610214 198312 1 001

3. **Prof. Dr. I Nyoman Budiantara, M.Si.** **(Penguji)**
   NIP. 19650603 198903 1 003

4. **Dr. Purhadi, M.Sc.** **(Penguji)**
   NIP. 19620204 198701 1 001

Direktur Pascasarjana

Prof. Dr. Ir. Adi Soeprijanto, MT.
NIP. 19640405 199002 1 001

METODE *FAST DOUBLE BOOTSTRAP* PADA REGRESI SPASIAL DATA PANEL DENGAN *SPATIAL FIXED EFFECT*

Nama mahasiswa : Nora Muhtasib
NRP : 1313 201 709
Pembimbing : Dr. Sutikno, S.Si., M.Si

ABSTRAK

Analisis tentang kemiskinan seringkali memiliki keterkaitan spasial antara satu wilayah dan wilayah di sekitarnya. Analisis data yang mempertimbangkan unsur spasial tidak hanya diterapkan pada data *crosssection*, namun telah berkembang pada data panel. Penggunaan data panel dalam regresi spasial memiliki banyak keuntungan, namun pengujian *spatial dependency* dan estimasi parameter yang dihasilkan pada regresi spasial data panel akan menjadi tidak akurat ketika diterapkan pada wilayah dengan unit analisis yang kecil. Salah satu metode mengatasi permasalahan pada jumlah pengamatan yang kecil adalah metode resampling. Penelitian ini menggunakan metode resampling *fast double bootstrap* (FDB) dengan memodelkan presentase penduduk miskin di Provinsi NTB sebagai kasus. Hasil pengujian *spatial dependency* menyimpulkan bahwa terdapat dependensi spasial kemiskinan antar wilayah di NTB. Pemodelan *Spatial Autoregressive* data panel dengan pendekatan FDB mampu menjelaskan keragaman persentase penduduk miskin di NTB sebesar 99,46 persen dan memenuhi asumsi kenormalan residual. Variabel yang berpengaruh secara signifikan terhadap persentase penduduk miskin di NTB adalah PDRB perkapita, persentase penduduk berusia 10 tahun keatas yang tidak/belum pernah bersekolah dan persentase penduduk yang bekerja di sektor industri.

Kata Kunci : Model Kemiskinan, Regresi Spasial Data Panel, *Fast Double Bootstrap, Spatial Fixed Effect*

FAST DOUBLE BOOTSTRAP METHOD ON SPATIAL PANEL DATA REGRESSION WITH *SPATIAL FIXED EFFECT*

Name	: Nora Muhtasib
NRP	: 1313 201 709
Supervisor	: Dr. Sutikno, S.Si., M.Si

ABSTRACT

Poverty analysis often have a spatial relationship between the region and the surrounding territories. Data analysis that considering spatial element is not only applied to the crosssectional data, but has grown on panel data. The use of panel data in spatial regression has many advantages, but the spatial dependency testing and parameter estimation result will be inaccurate when applied to small unit of analysis. One method to overcome the problems in a small number of observations is a resampling method. This study uses the fast double bootstrap resampling method (FDB) by modeling the percentage of poor people in NTB province as a case. The test results suggest that there are spatial dependency on poverty between regions in NTB. Modeling spatial autoregressive panel data with FDB approach is able to explain the diversity of the percentage of poor people in NTB by 99.46 percent and satisfy the assumption of normality of the residuals. The variables that significantly affect the percentage of poor people in NTB is GDP per capita, the percentage of the population aged 10 years and older who do not / have not been schooling and the percentage of people who work in the industrial sector.

Keywords : Poverty Modeling, Spatial Panel Data Regression, Fast Double Bootstrap, Spatial Fixed Effect

KATA PENGANTAR

Segala puji hanya bagi Allah SWT, Dzat Yang Maha Kuasa, yang telah memberikan nikmat dan karunia-Nya kepada penulis sehingga tesis yang berjudul ”Metode *Fast Doule Bootstrap* pada Regresi Spasial Data Panel dengan *Spatial Fixed Effect*, Studi Kasus : Pemodelan Persentase Penduduk Miskin di Provinsi NTB” dapat terselesaikan.

Penulisan tesis ini dapat terselesaikan berkat bantuan dan dukungan dari berbagai pihak, oleh karena itu pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih dan penghargaan setinggi-tingginya kepada :

1. Istri tercinta Siskarossa Ika Oktora serta kedua putriku Zahra Shareta Mumtaz dan Raihana Salsabila Mumtaza yang telah mendoakan, memberi dukungan dan bersabar selama penulis menjalani masa studi.
2. Kedua orang tua, Bapak Ahmad Azri dan Ibu Siti Nuryati, serta mertua, Papa Endang Darmawan dan Mama Andriawati yang selalu memberikan doa restu. Adik-adikku Nurida Azkar, Hafidh Ahmad, Shinta dan Bramandika yang telah banyak membantu penulis.
3. Kepala BPS RI beserta jajarannya yang telah memberi kesempatan kepada penulis untuk melanjutkan pendidikan pada jenjang pascasarjana.
4. Bapak Dr. Sutikno, M.Si. selaku pembimbing yang telah meluangkan waktu memberikan arahan serta berbagai motivasi kepada penulis.
5. Bapak Dr. Heru Margono, M.Sc., Bapak Prof. Dr. I Nyoman Budiantara, M.Si. dan Bapak Dr. Purhadi, M.Sc. selaku penguji yang telah memberikan masukan, saran dan koreksi bagi kebaikan tesis ini.
6. Bapak Dr. Suhartono, M.Sc. selaku ketua Program Studi Pascasarjana Statistika FMIPA ITS.
7. Ibu Dra. Sri Mumpuni Retnaningsih, MT selaku dosen wali selama penulis menuntut ilmu serta seluruh Bapak/Ibu dosen pengajar yang telah memberikan berbagai ilmu yang bermanfaat.
8. Rekan-rekan angkatan 2013 Jurusan Statistika ITS khususnya S2 BPS yang telah bersama-sama selama menempuh pendidikan.

9. Semua pihak yang telah membantu dan tidak dapat disebutkan satu per satu.

Penulis menyadari bahwa tesis ini masih jauh dari kesempurnaan, sehingga penulis memohon maaf atas segala kekeliruan. Akhirnya, semoga tulisan ini dapat bermanfaat dan ilmu yang diperoleh mendapatkan keberkahan dari Allah SWT. Amiin.

Surabaya, Maret 2015

Penulis

DAFTAR ISI

Halaman

DAFTAR TABEL

DAFTAR GAMBAR

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Kemiskinan (*poverty*) adalah suatu kondisi kehidupan ketika sejumlah penduduk tidak mampu mendapatkan sumber daya yang cukup untuk memenuhi kebutuhan dasar minimum hidup layak, baik makanan maupun non makanan, serta hidup di bawah tingkat kebutuhan minimum tersebut (Todaro dan Smith, 2006). Kemiskinan merupakan masalah serius bagi negara-negara berkembang seperti Indonesia. Berbagai program pemerintah yang difokuskan pada upaya pengentasan kemiskinan selalu menjadi agenda tahunan pemerintah pusat maupun pemerintah daerah. Banyak studi telah dilakukan untuk mengetahui dan mempelajari faktor penyebab serta pola kemiskinan di berbagai wilayah di Indonesia. Crandall dan Weber (2004) menyatakan bahwa pengurangan kemiskinan di suatu tempat akan mempengaruhi dan dipengaruhi oleh tempat-tempat lain di sekitarnya, atau dapat dikatakan bahwa kemiskinan mengandung unsur spasial.

Analisis regresi spasial pertama kali diperkenalkan oleh Jean Paelinck pada tahun 1970 yang kemudian dikembangkan oleh Anselin (1988). Model regresi spasial pada permasalahan riel seringkali menghasilkan kesimpulan yang lebih masuk akal dan lebih realistis daripada model regresi klasik (Ren, Long, Zhang, dan Chen, 2014). Dalam regresi spasial, keberadaan hubungan antar pengamatan (*spatial dependency*) merupakan syarat mutlak penggunaan analisis ini. Metode pengujian keberadaan *spatial dependency*, diantaranya: Uji Moran's I, Lagrange Multiplier (LM), Likelihood Ratio (LR), dan Rao's Score. Uji Moran's I merupakan uji yang paling sering digunakan, karena uji ini tidak mengasumsikan hipotesis alternative, namun dapat menguji keterkaitan lag (*Spatial lag dependence*) maupun keterkaitan error (*Spatial error dependence*) (Ren dkk., 2014).

Pada awalnya penggunaan analisis spasial menggunakan data *cross section*, namun pada satu dekade terakhir penggunaan analisis spasial semakin berkembang pada data panel. Beberapa penelitian yang menggunakan analisis spasial data panel,

diantaranya: Baltagi (2005), Lesage dan Pace (2009), Elhorst (2010), Lee dan Yu (2010), serta Lottman (2012).

Data panel merupakan gabungan data *cross section* yang tersusun dalam runtun waktu yang berurutan, atau dengan kata lain merupakan gabungan data *cross section* dan data *time series*. Menurut Baltagi (2005) penggunaan data panel memiliki banyak keuntungan, antara lain: (1) data lebih komperhensif bila dibandingkan dengan data *cross section*, sehingga meningkatkan jumlah dan variasi data, (2) memiliki derajat bebas yang lebih besar, serta mampu menghindari masalah multikolinieritas, (3) penggunaan data panel mampu meminimalisasi bias dan lebih unggul dalam mempelajari perubahan dinamis, sehingga dapat meningkatkan efisiensi dalam penaksiran parameternya, (4) mampu mendeteksi pengaruh-pengaruh yang tidak dapat diobservasi pada data *cross section* murni maupun data *time series* murni.

Menurut data BPS tahun 2012, persentase penduduk miskin di Provinsi Nusa Tenggara Barat (NTB) berada pada peringkat ke-6 tertinggi di Indonesia. Persentase penduduk miskin di NTB pada tahun 2012 mencapai 18,02%, berkurang dari 19,73% pada tahun 2011. Angka ini lebih baik dari Provinsi Papua, Papua Barat, Maluku, Nusa Tenggara Timur dan Nanggroe Aceh Darussalam. NTB hanya terdiri dari 10 kabupaten/kota yaitu Kabupaten Lombok Barat, Lombok Tengah, Lombok Timur, Lombok Utara, Sumbawa, Sumbawa Barat, Bima, Dompu, Kota Mataram dan Kota Bima.

Kendala yang dihadapi dalam pemodelan spasial tentang kemiskinan di NTB adalah jumlah unit analisis kabupaten/kota yang kecil. Di samping itu, penggunaan data panel dengan series waktu tahun 2010 – 2012 juga belum mampu menghasilkan jumlah pengamatan yang cukup besar. Kondisi tersebut akan memunculkan permasalahan dalam pengujian keberadaan *spatial dependency* yang disebabkan sampel yang kecil maupun residual yang tidak berdistribusi normal. Menurut Kogan (2010), pada umumnya, statistik inferensi didasarkan pada asumsi distribusi normal yang asimptotik dengan mengacu pada Hukum Bilangan Besar dan Teorema Limit Pusat. Pada sampel kecil, statistik inferensi akan menghasilkan selang kepercayaan dan uji statistik yang tidak sesuai. Pada sampel kecil, keakuratan estimator yang dihasilkan dari metode *maximum likelihood estimator*

(MLE) maupun *ordinary least square* (OLS) kurang baik. Untuk mengatasi permasalahan tersebut dapat dilakukan dengan metode Bootstrap. Lynch (2003) mengatakan bahwa jika error pada sampel kecil tidak berdistribusi normal maka metode Bootstrap dapat menjadi pemecahan masalah yang baik dalam mengestimasi parameter.

Efron (1979) memperkenalkan metode bootstrap yang berbasis komputasi sebagai alternatif pemecahan masalah secara empiris. Metode ini terbukti lebih akurat dibandingkan dengan metode asimptotik pada kondisi sampel kecil dan distribusi parameter tidak diketahui. Prinsip dasar metode bootstrap adalah membangkitkan set data baru dari data asli sebanyak B replikasi. Beran (1988) mengembangkan metode double bootstrap dengan kinerja yang lebih baik dibandingkan metode bootstrap biasa. Prinsip dasar metode double bootstrap adalah dari set data hasil bootstrap tahap pertama sebanyak B_1 dilakukan resampling kembali sebanyak B_2 replikasi bootstrap tahap kedua. Kelemahan metode double bootstrap adalah waktu penghitungan yang dibutuhkan lebih lama karena harus menghitung sebanyak $B_1 + B_1 B_2$ nilai statistik uji.

Davidson dan MacKinnon (2001) mengasumsikan bahwa statistik uji pada set data bootstrap tahap pertama dan statistik uji pada set data bootstrap tahap kedua saling bebas, dengan demikian untuk setiap set data bootstrap tahap pertama cukup dilakukan satu kali replikasi pada bootstrap tahap kedua. Metode ini menghasilkan tingkat akurasi yang sama dengan metode double bootstrap namun membutuhkan waktu pengolahan yang jauh lebih singkat. Metode ini disebut dengan *fast double bootstrap* (FDB). Lin, dkk. (2007) mengembangkan *spatial bootstrap test* berdasarkan OLS residual dengan didasarkan pada statistik Moran's I untuk menguji korelasi spasial pada model. Studi ini menggunakan metode bootstrap pada data *cross section*. Ren, dkk. (2014) mengembangkan penggunaan bootstrap dengan metode *fast double bootstrap test* (FDB) untuk menghitung nilai Moran's I pada data panel dengan sampel kecil.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, permasalahan yang dapat dirumuskan dalam tesis ini adalah :

1. Bagaimana prosedur FDB yang diterapkan pada regresi spasial data panel dengan *spatial fixed effect*?
2. Variabel apa sajakah yang berdampak signifikan terhadap tingkat kemiskinan di NTB dengan menggunakan metode FDB untuk spasial data panel?
3. Seberapa besarkah pengaruh tingkat kemiskinan di satu kabupaten/kota dengan tingkat kemiskinan kabupaten/kota lainnya di NTB?.

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah diuraikan diatas, maka tujuan yang ingin dicapai adalah :

1. Mengkaji penggunaan prosedur FDB pada regresi spasial data panel dengan *spatial fixed effect*.
2. Mendapatkan model kemiskinan terbaik di NTB dengan metode FDB yang diterapkan pada regresi spasial data panel.

1.4 Manfaat Penelitian

Manfaat yang ingin dicapai penelitian ini antara lain:

1. Menambah wawasan keilmuan tentang model spasial data panel dengan jumlah pengamatan yang kecil.
2. Mengkaji faktor penyebab tingginya tingkat kemiskinan di NTB sehingga diharapkan dapat merumuskan pemecahan masalah.
3. Dapat dijadikan sebagai bahan pertimbangan bagi pemerintah daerah dalam mengambil kebijakan untuk mengurangi tingkat kemiskinan khususnya di NTB.

1.5 Batasan Penelitian

Data yang digunakan merupakan *balanced panel data* dan model yang digunakan diasumsikan mengikuti model spasial data panel dengan *spatial fixed effect*.

BAB 2
TINJAUAN PUSTAKA

Bagian awal bab ini membahas tentang model regresi spasial data panel dengan *spatial fixed effect* dan metode *fast double bootstrap* serta langkah-langkah pemodelan regresinya. Bagian akhir bab ini membahas tentang tingkat kemiskinan di NTB dan faktor-faktor yang diduga mempengaruhinya.

2.1 Model Regresi Spasial Data Panel

2.1.1 Model Regresi Spasial

Konsep data spasial terdiri atas dependensi spasial dan atau heterogenitas spasial (Anselin, 1988). Spasial dependensi yang dikembangkan oleh anselin didasarkan pada hukum I Tobler yang berbunyi: *"Everything is related to everything else, but near thing are more related than distant things"*. Segala sesuatu saling berhubungan satu dengan yang lainnya, tetapi sesuatu yang dekat lebih mempunyai pengaruh daripada sesuatu yang jauh. Konsep heterogenitas spasial berarti bahwa antar lokasi mempunyai perbedaan dalam struktur, sehingga memungkinkan pemodelan yang berbeda di tiap lokasi.

Dengan konsep spasial dependensi, Anselin (1988) mengembangkan suatu bentuk *general spatial model* pada data *spatial cross section* dengan menambahkan unsur spasial pada model *ordinary least squares* (OLS). Bentuk umum *general spatial model* merupakan gabungan antara *autoregressive* dan *moving average (SARMA)* yang dirumuskan seperti pada persamaan 2.1 berikut :

$$\begin{aligned} \mathbf{y} &= \rho\mathbf{W}_1\mathbf{y} + \mathbf{X}\boldsymbol{\beta} + \mathbf{u}, \\ \mathbf{u} &= \lambda\mathbf{W}_2\mathbf{u} + \boldsymbol{\varepsilon}, \\ \boldsymbol{\varepsilon} &\sim \mathrm{N}(\mathbf{0}, \sigma^2\mathbf{I}) \end{aligned} \tag{2.1}$$

$\mathbf{y}$ adalah vektor variabel dependen berukuran *n x 1*. ρ adalah koefisien variabel dependen spasial lag. $\mathbf{u}$ adalah vektor error berukuran *n x 1*. $\mathbf{W_1} = \mathbf{W_2} = \mathbf{W}$ adalah matriks penimbang spasial berukuran *n x n*. $\boldsymbol{\beta}$ adalah vektor parameter regresi

berukuran *(k+1) x 1*. **X** adalah matriks variabel independen berukuran *n x (k+1)*. λ merupakan koefisien autokorelasi spasial error.

Dari persamaan 2.1 diatas dapat diperoleh beberapa model turunan diantaranya :

a. Jika $\rho = 0$ dan $\lambda = 0$, maka bentuk persamaan umum diatas menjadi

$$\mathbf{y} = \mathbf{X\beta} + \boldsymbol{\varepsilon}, \tag{2.2}$$

yang merupakan model regresi OLS.

b. Jika $\rho \neq 0$ dan $\lambda = 0$, maka bentuk persamaan umum diatas menjadi

$$\mathbf{y} = \rho\mathbf{Wy} + \mathbf{X\beta} + \boldsymbol{\varepsilon}, \tag{2.3}$$

yaitu bentuk model *spatial autoregressive* (SAR) atau *spatial lag model* (SLM).

c. Jika $\rho = 0$ dan $\lambda \neq 0$, maka bentuk persamaan umum diatas menjadi

$\mathbf{y} = \mathbf{X\beta} + \mathbf{u}$, dimana

$$\mathbf{u} = \lambda\mathbf{Wu} + \boldsymbol{\varepsilon} \tag{2.4}$$

yaitu *spatial error model* (SEM).

d. Jika $\rho \neq 0$ dan $\lambda \neq 0$, maka bentuk persamaan umum diatas menjadi

$$\mathbf{y} = \rho\mathbf{Wy} + \mathbf{X\beta} + \mathbf{u},$$

dimana $\mathbf{u} = \lambda\mathbf{Wu} + \boldsymbol{\varepsilon}$ (2.5)

yang disebut dengan model *spatial autoregressive moving average* (SARMA).

2.1.2 Model Regresi Spasial Data Panel

Model regresi *ordinary least squares* dengan data panel tanpa adanya interaksi spasial (*common pooled* OLS) memiliki bentuk sebagai berikut :

$$y_{it} = \mathbf{x}_{it}\boldsymbol{\beta} + \varepsilon_{it} \tag{2.6}$$

atau secara umum dapat dituliskan dengan

$$\boldsymbol{y}_t = \mathbf{X}_t\boldsymbol{\beta} + \boldsymbol{\varepsilon}_t \tag{2.7}$$

$$\begin{pmatrix} y_{11} \\ y_{21} \\ \dots \\ y_{n1} \\ y_{12} \\ y_{22} \\ \dots \\ y_{n2} \\ y_{it} \\ y_{i+1,t} \\ \dots \\ y_{nt} \end{pmatrix} = \begin{pmatrix} 1 & x_{111} & x_{112} & \cdots & x_{11k} \\ 1 & x_{211} & x_{212} & \cdots & x_{21k} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n11} & x_{n12} & \cdots & x_{n1k} \\ 1 & x_{121} & x_{122} & \cdots & x_{12k} \\ 1 & x_{221} & x_{222} & \cdots & x_{22k} \\ \vdots & \vdots & & & \vdots \\ 1 & x_{n21} & x_{n22} & \cdots & x_{n2k} \\ 1 & x_{it1} & x_{it2} & & x_{itk} \\ 1 & x_{i+1,t,1} & x_{i+1t2} & & x_{i+1tk} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{nt1} & x_{nt2} & & x_{ntk} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \\ \\ \\ \\ \\ \\ \\ \\ \beta_k \end{pmatrix} + \begin{pmatrix} \varepsilon_{11} \\ \varepsilon_{21} \\ \vdots \\ \varepsilon_{n1} \\ \varepsilon_{12} \\ \varepsilon_{22} \\ \vdots \\ \varepsilon_{n1} \\ \varepsilon_{it} \\ \varepsilon_{i+1t} \\ \vdots \\ \varepsilon_{nt} \end{pmatrix}$$

dengan i adalah indeks untuk unit *cross section* (unit spasial), i = 1,2,...,n dan t adalah indeks untuk waktu (*time period*), t=1,2,...,T. $\mathbf{y}_{it}$ adalah variabel dependen pada pengamatan ke-i dan waktu ke-t yang merupakan bagian dari vektor $\boldsymbol{y}_t$ yang berukuran *nt*x1. $\mathbf{x}_{it}$ merupakan vektor baris variabel independen pada pengamatan ke-i dan waktu ke-t dengan ukuran 1x(*k*+1) yang merupakan bagian dari matriks $\boldsymbol{X}_t$ yang berukuran *nt*x(*k*+1). $\boldsymbol{\beta}$ adalah vektor parameter yang bersifat *fixed* yang tidak diketahui dan berukuran (*k*+1)x1. ε_{it} adalah error yang berdistribusi normal, bersifat identik dan independen dengan $E(\varepsilon_{it}) = 0$ dan $Var(\varepsilon_{it}) = \sigma^2$.

Baltagi (2005) menyatakan bahwa berdasarkan komponen *error*-nya, data panel dapat dituliskan mengikuti *one way model* yaitu dengan menambahkan efek spasial maupun *two way model* dengan menambahkan efek spasial dan efek waktu (*time period*). Dengan menambahkan efek spasial, maka persamaan 2.6 akan menjadi *one way model* sebagai berikut :

$$y_{it} = \mathbf{x}_{it}\boldsymbol{\beta} + u_i + \varepsilon_{it} \tag{2.8}$$

dan secara umum dapat dituliskan

$$\boldsymbol{y}_t = \mathbf{X}_t\boldsymbol{\beta} + \mathbf{u} + \boldsymbol{\varepsilon}_t \tag{2.9}$$

dengan u_i adalah efek spasial yang mengontrol semua ruang yang menyebabkan bias pada penelitian data *cross section*. Sedangkan penambahan efek spasial dan efek waktu akan membentuk model berikut

$$y_{it} = \mathbf{x}_{it}\boldsymbol{\beta} + u_i(optional) + \omega_t(optional) + \varepsilon_{it} \tag{2.10}$$

dengan ω_t adalah efek waktu yang mengontrol semua ruang yang menyebabkan bias pada penelitian data *time series*.

Elhorst (2003,2010) mengembangkan model regresi spasial data panel yang tidak jauh berbeda dengan model regresi spasial pada data *cross section*. Model ini dikembangkan dengan pendekatan mulai model OLS sampai pada model yang paling umum yaitu model *general nesting spatial* (GNS) (Vega dan Elhorst, 2013). Seperti halnya pada data *cross section*, Elhorst (2003) mengembangkan regresi data panel dengan adanya interaksi spasial. Model yang terbentuk dapat mengandung proses *autoregressive* pada error yang disebut *spatial error model*, maupun mengandung proses *autoregressive* pada variabel dependen yang disebut sebagai *spatial lag model* (SLM) atau *spatial autoregressive* (SAR). Model spasial lag (SLM) memperlihatkan bahwa variabel dependen di suatu lokasi memiliki keterkaitan terhadap variabel dependen tetangga (*neighboring*) dan suatu set karakteristik lokal (Elhorst, 2010a). Model SLM dituliskan sebagai berikut:

$$y_{it} = \rho \sum_{j=1}^{n} w_{ij}\, y_{jt} + \mathbf{x}_{it}\boldsymbol{\beta} + u_i + \varepsilon_{it} \tag{2.11}$$

dengan bentuk umum

$$\mathbf{y}_t = \rho\mathbf{W}\mathbf{y}_t + \mathbf{X}_t\boldsymbol{\beta} + \mathbf{u} + \boldsymbol{\varepsilon}_t \tag{2.12}$$

$$\begin{pmatrix} y_{11} \\ y_{21} \\ \cdots \\ y_{n1} \\ y_{12} \\ y_{22} \\ \cdots \\ y_{n2} \\ y_{it} \\ y_{i+1,t} \\ \cdots \\ y_{nt} \end{pmatrix} = \rho \begin{pmatrix} 0 & w_{12} & w_{13} & \cdots & w_{1n} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ w_{21} & 0 & w_{23} & & w_{2n} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ w_{31} & w_{32} & 0 & & \vdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & & & 0 & & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ w_{n1} & w_{n2} & \cdots & & 0 & \cdots & \cdots & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \vdots & 0 & \cdots & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \vdots & \vdots & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \vdots & \vdots & \vdots & 0 & w_{12} & w_{13} & \cdots & w_{1n} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & w_{21} & 0 & w_{23} & & w_{2n} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & w_{31} & w_{32} & 0 & & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \vdots & & & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & w_{n1} & w_{n2} & \cdots & & 0 \end{pmatrix} \begin{pmatrix} y_{11} \\ y_{21} \\ \cdots \\ y_{n1} \\ y_{12} \\ y_{22} \\ \cdots \\ y_{n2} \\ y_{jt} \\ y_{j+1,t} \\ \cdots \\ y_{nt} \end{pmatrix}$$

$$+ \begin{pmatrix} 1 & x_{111} & x_{112} & \cdots & x_{11k} \\ 1 & x_{211} & x_{212} & \cdots & x_{21k} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n11} & x_{n12} & \cdots & x_{n1k} \\ 1 & x_{121} & x_{122} & \cdots & x_{12k} \\ 1 & x_{221} & x_{222} & \cdots & x_{22k} \\ \vdots & \vdots & & & \vdots \\ 1 & x_{n21} & x_{n22} & \cdots & x_{n2k} \\ 1 & x_{it1} & x_{it2} & & x_{itk} \\ 1 & x_{i+1,t,1} & x_{i+1t2} & & x_{i+1tk} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{nt1} & x_{nt2} & & x_{ntk} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \\ \\ \\ \\ \beta_k \end{pmatrix} + \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \\ \\ \\ \\ u_n \end{pmatrix} + \begin{pmatrix} \varepsilon_{11} \\ \varepsilon_{21} \\ \vdots \\ \varepsilon_{n1} \\ \varepsilon_{12} \\ \varepsilon_{22} \\ \vdots \\ \varepsilon_{n1} \\ \varepsilon_{it} \\ \varepsilon_{i+1t} \\ \vdots \\ \varepsilon_{nt} \end{pmatrix}$$

dengan y_{it} adalah pengamatan ke-i dan waktu ke-t. ρ merupakan parameter interaksi spasial antar lokasi. Variabel $\sum_{j=1}^{n} w_{ij}\, y_{jt}$ menggambarkan dampak interaksi antara y_{it} dan y_{jt} dalam sebuah hubungan kebertetanggaan (*neighboring*), dimana w_{ij} adalah elemen pada baris ke-i, dan kolom ke-j di dalam matriks penimbang non-negatif **W** berukuran nxn yang menggambarkan susunan

kedekatan dalam suatu set karakteristik lokal sebagai unit analisis. $\mathbf{x}_{it}$ adalah vektor baris variabel prediktor pada pengamatan ke-i dan waktu ke-t yang berukuran 1x(k+1) yang merupakan bagian dari matriks $\boldsymbol{X}_t$ yang berukuran ntx(k+1). $\boldsymbol{\beta}$ adalah vektor parameter berukuran (k+1)x1. u_i adalah efek spasial dan ε_{it} adalah residual yang berdistribusi normal, bersifat identik dan independen dengan $E(\varepsilon_{it}) = 0$ dan $Var(\varepsilon_{it}) = \sigma^2$.

Pada *spatial error model* (SEM), error pada lokasi ke-i (ϕ_{it}) memiliki hubungan kebertetanggaan (*neighboring*) dengan error pada lokasi ke-j yang direpresentasikan dalam matriks penimbang **W** dan sebuah komponen ε_{it}. Secara umum bentuk modelnya adalah

$$y_{it} = \mathbf{x}_{it}\boldsymbol{\beta} + u_i + \phi_{it}$$

$$\phi_{it} = \lambda \sum_{j=1}^{n} w_{ij}\, \phi_{jt} + \varepsilon_{it} \qquad (2.13)$$

dengan λ yang disebut sebagai koefisien autokorelasi spasial error.

2.1.3 Estimasi Parameter Regresi Data Panel tanpa Interaksi Spasial

Untuk mengestimasi parameter regresi data panel tanpa interaksi spasial (*pooled OLS*) digunakan metode *maximum likelihood* (ML). Efek spasial u_i dapat diperlakukan sebagai efek tetap (*fixed effect*) maupun efek acak (*random effect*) yang identik dan independen serta berdistribusi normal dengan *mean* nol dan varians konstan (σ_u^2). Kemudian diasumsikan pula bahwa u_i dan ε_{it} saling bebas.

a. Model *fixed effect*

Jika u_i diasumsikan tetap, maka estimasi parameter dilakukan dengan langkah berikut. *Pertama*, u_i dieliminasi dengan metode *demeaning* dari variabel y dan x dengan transformasi sebagai berikut.

$$y_{it}^* = y_{it} - \frac{1}{T}\sum_{t=1}^{T} y_{it} \quad \text{dan} \quad \mathbf{x}_{it}^* = \mathbf{x}_{it} - \frac{1}{T}\sum_{t=1}^{T} \mathbf{x}_{it} \qquad (2.14)$$

Kedua, setelah mendapatkan nilai yang baru, persamaan regresinya menjadi $y_{it}^* = \mathbf{x}_{it}^*\boldsymbol{\beta} + \varepsilon_{it}^*$, sehingga dengan metode OLS diperoleh $\widehat{\boldsymbol{\beta}} = (\mathbf{X}^{*\prime}\mathbf{X}^*)^{-\mathbf{1}}(\mathbf{X}^{*\prime}\mathbf{y}^*)$ serta varians $\hat{\sigma}^2 = \frac{1}{(nT-n-K)}\left(\mathbf{y}^* - \mathbf{X}^*\widehat{\boldsymbol{\beta}}\right)'\left(\mathbf{y}^* - \mathbf{X}^*\widehat{\boldsymbol{\beta}}\right)$.

Estimasi persamaan *demeaned* juga dapat diturunkan dengan metode ML dengan fungsi *log-likelihood* sebagai berikut :

$$\ln L = -\frac{nT}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}\sum_{t=1}^{T}(y_{it}^* - \mathbf{x}_{it}^{*\prime}\boldsymbol{\beta})^2 \tag{2.15}$$

sehingga dengan metode ML diperoleh $\widehat{\boldsymbol{\beta}} = (\mathbf{X}^{*\prime}\mathbf{X}^*)^{-\mathbf{1}}(\mathbf{X}^{*\prime}\mathbf{y}^*)$ dan varians $\hat{\sigma}^2 = \frac{1}{nT}\left(\mathbf{y}^* - \mathbf{X}^*\widehat{\boldsymbol{\beta}}\right)'\left(\mathbf{y}^* - \mathbf{X}^*\widehat{\boldsymbol{\beta}}\right)$. Dengan demikian,

$$u_i = \frac{1}{T}\sum_{t=1}^{T}(y_{it} - \mathbf{x}_{it}{}'\boldsymbol{\beta})$$

dimana i=1,...,n (2.16)

Pada model regresi data panel tanpa interaksi spasial, u_i yang tetap diasumsikan memenuhi kondisi $\sum u_i = 0$.

b. Model *random effect*

Untuk mendapatkan estimasi parameter ML dari model *random effect* Breusch dalam Elhorst (2010a) menggunakan prosedur estimasi dua tahap. *Log likelihood* yang diperoleh adalah

$$\ln L = -\frac{nT}{2}\log(2\pi\sigma^2) + \frac{n}{2}\ln\theta^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{n}\sum_{t=1}^{T}(y_{it}^* - \mathbf{x}_{it}^{*\prime}\boldsymbol{\beta})^2 \tag{2.17}$$

dengan θ menunjukkan bobot yang melekat pada data *cross section* dengan $0 \leq \theta^2 = \frac{\sigma^2}{(T\sigma_u^2+\sigma^2)} \leq 1$ dan simbol * menunjukkan suatu transformasi variabel dependen pada θ, yaitu

$$y_{it}^* = y_{it} - (1-\theta)\frac{1}{T}\sum_{t=1}^{T} y_{it} \text{ dan } \mathbf{x}_{it}^* = \mathbf{x}_{it} - (1-\theta)\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_{it} \tag{2.18}$$

Jika $\theta = 0$, maka transformasi yang terbentuk akan menyederhanakan prosedur *demeaning* pada persamaan 2.11 sehingga kembali ke model efek tetap.

Dengan maksimasi *log likelihood* pada orde pertama diperoleh $\widehat{\boldsymbol{\beta}} = (\mathbf{X}^{*\prime}\mathbf{X}^{*})^{-\mathbf{1}}(\mathbf{X}^{*\prime}\mathbf{y}^{*})$ dan varians $\hat{\sigma}^2 = \frac{1}{nT}\left(\mathbf{y}^{*} - \mathbf{X}^{*}\widehat{\boldsymbol{\beta}}\right)'\left(\mathbf{y}^{*} - \mathbf{X}^{*}\widehat{\boldsymbol{\beta}}\right)$.

2.1.4 Estimasi Parameter Regresi Spasial Data Panel dengan s*patial fixed effect*

Estimasi parameter regresi spasial data panel ini mengasumsikan bahwa matriks **W** adalah konstan sepanjang waktu dan data yang digunakan adalah data panel seimbang (*balanced panel*).

a. Fixed Effect Spatial Lag Model

Estimator ML dapat diperoleh dengan menyelesaikan persamaan berikut.

$\mathbf{y} = \boldsymbol{\rho}\mathbf{W}_{\mathbf{nT}}\mathbf{y} + \mathbf{X}\boldsymbol{\beta} + \mathbf{u} + \boldsymbol{\varepsilon}$ dimana $\mathbf{W}_{\mathrm{nT}} = (\mathbf{I}_{\mathrm{T}} \otimes \mathbf{W})$.

$$\boldsymbol{\varepsilon} = \mathbf{y} - \boldsymbol{\rho}\mathbf{W}_{\mathbf{nT}}\mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{u} \tag{2.19}$$

persamaan fungsi *log likelihood* dengan asumsi efek spasial tetap

$$\ln L = -\frac{nT}{2}\ln(2\pi\sigma^2) + T\ln|\mathbf{I} - \rho\mathbf{W}_{\mathrm{nT}}| - \frac{1}{2\sigma^2}\sum_{\mathrm{i}=1}^{\mathrm{n}}\sum_{\mathrm{t}=1}^{\mathrm{T}}\left(\mathrm{y}_{\mathrm{it}} - \rho\sum_{\mathrm{j}=1}^{\mathrm{n}}\mathrm{w}_{\mathrm{ij}}\mathrm{y}_{\mathrm{jt}} - \mathbf{x}_{\mathrm{it}}{}'\boldsymbol{\beta} - \mathrm{u}_{\mathrm{i}}\right)^2 \tag{2.20}$$

(Anselin, 1988).

Turunan pertama dari persamaan 2.19 terhadap u_{i} disamakan dengan 0.

$$\frac{\partial \ln L}{\partial \mathrm{u}_{\mathrm{i}}} = \frac{1}{\sigma^2}\sum_{\mathrm{t}=1}^{\mathrm{T}}\left(\mathrm{y}_{\mathrm{it}} - \rho\sum_{\mathrm{j}=1}^{\mathrm{n}}\mathrm{w}_{\mathrm{ij}}\mathrm{y}_{\mathrm{jt}} - \mathbf{x}_{\mathrm{it}}{}'\boldsymbol{\beta} - \mathrm{u}_{\mathrm{i}}\right) = 0, i = 1, \dots, N$$

$$\sum_{\mathrm{t}=1}^{\mathrm{T}}\left(\mathrm{y}_{\mathrm{it}} - \rho\sum_{\mathrm{j}=1}^{\mathrm{n}}\mathrm{w}_{\mathrm{ij}}\mathrm{y}_{\mathrm{jt}} - \mathbf{x}_{\mathrm{it}}{}'\boldsymbol{\beta} - \mathrm{u}_{\mathrm{i}}\right) = 0$$

$$\mathrm{T}\mathrm{u}_{\mathrm{i}} = \sum_{\mathrm{t}=1}^{\mathrm{T}}\left(\mathrm{y}_{\mathrm{it}} - \rho\sum_{\mathrm{j}=1}^{\mathrm{n}}\mathrm{w}_{\mathrm{ij}}\mathrm{y}_{\mathrm{jt}} - \mathbf{x}_{\mathrm{it}}{}'\boldsymbol{\beta}\right)$$

$$u_i = \frac{1}{T}\sum_{t=1}^{T}\left(y_{it} - \rho\sum_{j=1}^{n} w_{ij}y_{jt} - \mathbf{x}_{it}{}'\boldsymbol{\beta}\right), i = 1, \dots, n$$

(2.21)

Substitusi nilai u_i ke dalam fungsi *log-likelihood* serta penyusunan ulang fungsi *concentrated log-likelihood* untuk $\boldsymbol{\beta}, \rho$ dan $\hat{\sigma}^2$ akan memunculkan persamaan 2.22

$$\begin{aligned}
\ln L = &-\frac{nT}{2}\ln(2\pi\sigma^2) + T\ln|\mathbf{I} - \rho\mathbf{W}_{nT}| \\
&-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\sum_{t=1}^{T}\left(y_{it} - \rho\sum_{j=1}^{n} w_{ij}y_{jt} - \mathbf{x}_{it}'\boldsymbol{\beta}\right. \\
&\qquad\left.-\frac{1}{T}\sum_{t=1}^{T}\left(y_{it} - \rho\sum_{j=1}^{n} w_{ij}y_{jt} - \mathbf{x}_{it}{}'\boldsymbol{\beta}\right)\right)^2 \\
= &-\frac{nT}{2}\ln(2\pi\sigma^2) + T\ln|\mathbf{I} - \rho\mathbf{W}_{nT}| \\
&-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\sum_{t=1}^{T}\left(y_{it}^* - \rho\left[\sum_{j=1}^{n} w_{ij}y_{jt}\right]^* - \mathbf{x}_{it}^*{}'\boldsymbol{\beta}\right)^2
\end{aligned}$$

(2.22)

tanda bintang menunjukkan prosedur *demeaning* yang telah dijelaskan pada persamaan 2.14.

Anselin dan Hudak (1992) menerangkan prosedur estimasi parameter dengan metode ML untuk memaksimasi fungsi *log likelihood* yang dikembangkan dari data *cross section* sebanyak *N* observasi ke data panel sebanyak *nxT* observasi. Dari susunan data *cross section* untuk $t = 1, \dots, T$ diperoleh vektor $\mathbf{y}^*$ berukuran *nT*x1, $\mathbf{W}_{nT}\mathbf{y}^*$ serta matriks $\mathbf{X}^*$berukuran *nTxk*, kesemuanya merupakan variabel yang telah melalui proses *demeaned*. Selanjutnya estimator $\mathbf{b}_0$ dan $\mathbf{b}_1$ dapat diperoleh dari regresi OLS variabel-variabel tersebut. Selain itu, $\mathbf{e}_0^*$ dan $\mathbf{e}_1^*$ sebagai error hasil estimasi tersebut juga dapat diperoleh, sehingga ρ diperoleh dengan memaksimasi fungsi *concentred log-likelihood* berikut

$$\ln L = C - \frac{nT}{2}\ln[(\mathbf{e}_0^* - \delta\mathbf{e}_1^*)'(\mathbf{e}_0^* - \rho\mathbf{e}_1^*)] + T\ln|\mathbf{I} - \rho\mathbf{W}| \qquad (2.23)$$

dengan C adalah konstanta yang tidak tergantung pada ρ. Estimator $\widehat{\boldsymbol{\beta}}$ dan $\hat{\sigma}^2$ dapat dihitung setelah estimasi numerik ρ. Dengan menyamakan turunan pertama persamaan 2.22 terhadap $\boldsymbol{\beta}$ dan σ^2, maka diperoleh

$$\frac{\partial \ln L(\beta, \rho, \sigma^2 | \mathbf{y}, \mathbf{X})}{\partial \boldsymbol{\beta}} =$$

$$\frac{\partial \left[-\frac{1}{2\sigma^2} (\mathbf{y}^* - \rho \mathbf{W}_{nT} \mathbf{y}^* + \mathbf{X}^* \boldsymbol{\beta})' (\mathbf{y}^* - \rho \mathbf{W}_{nT} \mathbf{y}^* + \mathbf{X}^* \boldsymbol{\beta}) \right]}{\partial \boldsymbol{\beta}} = 0$$

$$\frac{1}{\sigma^2} \left((\mathbf{X}^{*\prime} [\mathbf{y}^* - \rho \mathbf{W}_{nT} \mathbf{y}^*]) - (\mathbf{X}^{*\prime} \mathbf{X}^*) \boldsymbol{\beta} \right) = 0$$

$$\widehat{\boldsymbol{\beta}} = \mathbf{b}_0 - \rho \mathbf{b}_1 = (\mathbf{X}^{*\prime} \mathbf{X}^*)^{-1} (\mathbf{X}^{*\prime} [\mathbf{y}^* - \rho \mathbf{W}_{nT} \mathbf{y}^*])$$

$$\frac{\partial \ln L(\beta, \rho, \sigma^2 | \mathbf{y}, \mathbf{X})}{\partial \boldsymbol{\beta}} =$$

$$\frac{\partial \left[-\frac{nT}{2} \ln \sigma^2 - \frac{1}{2\sigma^2} (\mathbf{y}^* - \rho \mathbf{W}_{nT} \mathbf{y}^* + \mathbf{X}^* \boldsymbol{\beta})' (\mathbf{y}^* - \rho \mathbf{W}_{nT} \mathbf{y}^* + \mathbf{X}^* \boldsymbol{\beta}) \right]}{\partial \sigma^2} = 0$$

$$-\frac{nT}{2\sigma^2} + \frac{1}{2\sigma^4} [(\mathbf{y}^* - \rho \mathbf{W}_{nT} \mathbf{y}^* + \mathbf{X}^* \boldsymbol{\beta})' (\mathbf{y}^* - \rho \mathbf{W}_{nT} \mathbf{y}^* + \mathbf{X}^* \boldsymbol{\beta})] = 0$$

$$\frac{nT}{2\sigma^2} = \frac{1}{2\sigma^4} [(\mathbf{y}^* - \rho \mathbf{W}_{nT} \mathbf{y}^* + \mathbf{X}^* \boldsymbol{\beta})' (\mathbf{y}^* - \rho \mathbf{W}_{nT} \mathbf{y}^* + \mathbf{X}^* \boldsymbol{\beta})]$$

$$\frac{nT 2\sigma^4}{2\sigma^2} = [(\mathbf{y}^* - \rho \mathbf{W}_{nT} \mathbf{y}^* + \mathbf{X}^* \boldsymbol{\beta})' (\mathbf{y}^* - \rho \mathbf{W}_{nT} \mathbf{y}^* + \mathbf{X}^* \boldsymbol{\beta})]$$

$$\hat{\sigma}^2 = \frac{1}{nT} (\mathbf{y}^* - \rho \mathbf{W}_{nT} \mathbf{y}^* + \mathbf{X}^* \boldsymbol{\beta})' (\mathbf{y}^* - \rho \mathbf{W}_{nT} \mathbf{y}^* + \mathbf{X}^* \boldsymbol{\beta}).$$

Untuk mengganti variabel *demeaned* dengan variabel asli **Y** dan **X**, maka variabel $\mathbf{y}^* = \mathbf{Q}\mathbf{y}$, $(\mathbf{I}_T \otimes \mathbf{W})\mathbf{y}^* = \mathbf{Q}(\mathbf{I}_T \otimes \mathbf{W})\mathbf{y}$ dan $\mathbf{X}^* = \mathbf{Q}\mathbf{X}$, dimana **Q** menunjukkan operator *demeaned* dalam bentuk matriks, $\mathbf{Q} = \mathbf{I}_{nT} - \frac{1}{T} \boldsymbol{\iota}_T \boldsymbol{\iota}_T' \otimes \mathbf{I}_n$, dengan $\boldsymbol{\iota}_T$ adalah vektor satu yang menunjukkan panjang vektor ini. **Q** adalah matriks idempoten simetris sehingga estimator $\boldsymbol{\beta}$ dapat ditulis

$$\begin{aligned} \widehat{\boldsymbol{\beta}} &= (\mathbf{X}'\mathbf{Q}'\mathbf{Q}\mathbf{X})^{-1} \mathbf{X}'\mathbf{Q}'\mathbf{Q}[\mathbf{y} - \rho(\mathbf{I}_T \otimes \mathbf{W})]\mathbf{y} \\ &= (\mathbf{X}'\mathbf{Q}\mathbf{X})^{-1} \mathbf{X}'\mathbf{Q}[\mathbf{y} - \rho(\mathbf{I}_T \otimes \mathbf{W})]\mathbf{y} \end{aligned} \tag{2.24}$$

b. Fixed Effect Spatial Error Model

Menurut Anselin dan Hudak (1992), estimasi parameter β, λ dan σ^2 dapat diperoleh melalui tahapan berikut

$\mathbf{y} = \mathbf{X\beta} + [\mathbf{I} - \lambda\mathbf{W}_{nT}]^{-1}\boldsymbol{\varepsilon}$, dimana $\mathbf{W}_{nT} = (\mathbf{I}_T \otimes \mathbf{W})$.

$\boldsymbol{\varepsilon} = (\mathbf{I} - \lambda\mathbf{W}_{nT})(\mathbf{y} - \mathbf{X}'\boldsymbol{\beta})$, setelah dilakukan proses demeaning, maka $\mathbf{y}$ dan $\mathbf{X}$ menjadi $\mathbf{y}^*$dan $\mathbf{X}^*$.

$$J = \frac{\partial v}{\partial x} = |\mathbf{I} - \lambda\mathbf{W}_{nT}|$$

$$\ln L = -\frac{nT}{2}\ln(2\pi\sigma^2) + T\ln|\mathbf{I} - \lambda\mathbf{W}_{nT}| - \frac{1}{2\sigma^2}\sum_{i=1}^{n}\sum_{t=1}^{T}\Big(y_{it}^* - \lambda\big[\sum_{j=1}^{n} w_{ij}y_{jt}\big]^* - \Big(x_{it}^* - \lambda\big[\sum_{j=1}^{n} w_{ij}\mathbf{x}_{jt}\big]^*\Big)\boldsymbol{\beta}\Big)^2$$

$$= -\frac{nT}{2}\ln(2\pi) - \frac{nT}{2}\ln\sigma^2 + T\ln|\mathbf{I} - \lambda\mathbf{W}_{nT}| - \frac{1}{2\sigma^2}(\mathbf{y}^* - \mathbf{X}^*\beta)'(\mathbf{I} - \lambda\mathbf{W}_{nT})'(\mathbf{I} - \lambda\mathbf{W}_{nT})(\mathbf{y}^* - \mathbf{X}^*\beta)$$

estimator ML dari $\boldsymbol{\beta}$ dan σ^2diperoleh dari turunan pertama disamakan dengan nol, yaitu

$$\frac{\partial \ln L(\beta, \lambda, \sigma^2|\mathbf{y}, \mathbf{X})}{\partial \boldsymbol{\beta}} = \frac{\partial\left[-\frac{1}{2\sigma^2}(\mathbf{y}^* - \mathbf{X}^*\boldsymbol{\beta})'(\mathbf{I} - \lambda\mathbf{W}_{nT})'(\mathbf{I} - \lambda\mathbf{W}_{nT})(\mathbf{y}^* - \mathbf{X}^*\boldsymbol{\beta})\right]}{\partial \boldsymbol{\beta}} = 0$$

$$\frac{1}{\sigma^2}([\mathbf{X}^* - \lambda\mathbf{W}_{nT}\mathbf{X}^*]'[\mathbf{X}^* - \lambda\mathbf{W}_{nT}\mathbf{y}^*]) - ([\mathbf{X}^* - \lambda\mathbf{W}_{nT}\mathbf{X}^*]'[\mathbf{X}^* - \lambda\mathbf{W}_{nT}\mathbf{X}^*]\boldsymbol{\beta}) = 0$$

$$\widehat{\boldsymbol{\beta}} = ([\mathbf{X}^* - \lambda(\mathbf{I}_T \otimes \mathbf{W})\mathbf{X}^*]'[\mathbf{X}^* - \lambda(\mathbf{I}_T \otimes \mathbf{W})\mathbf{X}^*])^{-1} ([\mathbf{X}^* - \lambda(\mathbf{I}_T \otimes \mathbf{W})\mathbf{X}^*]'[\mathbf{X}^* - \lambda(\mathbf{I}_T \otimes \mathbf{W})\mathbf{y}^*]) \quad (2.25)$$

$$\frac{\partial \ln L(\beta, \lambda, \sigma^2|\mathbf{y}, \mathbf{X})}{\partial \sigma^2} = \frac{\partial\left[-\frac{nT}{2}\ln\sigma^2 - \frac{1}{2\sigma^2}(\mathbf{y}^* - \mathbf{X}^*\boldsymbol{\beta})'(\mathbf{I} - \lambda\mathbf{W}_{nT})'(\mathbf{I} - \lambda\mathbf{W}_{nT})(\mathbf{y}^* - \mathbf{X}^*\boldsymbol{\beta})\right]}{\partial \sigma^2} = 0$$

$$-\frac{nT}{2\sigma^2}+\frac{1}{2\sigma^4}[(\mathbf{y}^*-\mathbf{X}^*\boldsymbol{\beta})'(\mathbf{I}-\lambda\mathbf{W}_{\mathrm{nT}})'(\mathbf{I}-\lambda\mathbf{W}_{\mathrm{nT}})(\mathbf{y}^*-\mathbf{X}^*\boldsymbol{\beta})]=0$$

$$\frac{nT}{2\sigma^2}=\frac{1}{2\sigma^4}[(\mathbf{y}^*-\mathbf{X}^*\boldsymbol{\beta})'(\mathbf{I}-\lambda\mathbf{W}_{\mathrm{nT}})'(\mathbf{I}-\lambda\mathbf{W}_{\mathrm{nT}})(\mathbf{y}^*-\mathbf{X}^*\boldsymbol{\beta})]$$

$$\frac{nT2\sigma^4}{2\sigma^2}=[(\mathbf{y}^*-\mathbf{X}^*\boldsymbol{\beta})'(\mathbf{I}-\lambda\mathbf{W}_{\mathrm{nT}})'(\mathbf{I}-\lambda\mathbf{W}_{\mathrm{nT}})(\mathbf{y}^*-\mathbf{X}^*\boldsymbol{\beta})]$$

$$\hat{\sigma}^2=\frac{1}{nT}\left(\mathbf{y}^*-\lambda\mathbf{W}_{\mathrm{nT}}\mathbf{y}^*-[\mathbf{X}^*-\lambda\mathbf{W}_{\mathrm{nT}}\mathbf{X}^*]\widehat{\boldsymbol{\beta}}\right)'\left(\mathbf{y}^*-\lambda\mathbf{W}_{\mathrm{nT}}\mathbf{y}^*-[\mathbf{X}^*-\lambda\mathbf{W}_{\mathrm{nT}}\mathbf{X}^*]\widehat{\boldsymbol{\beta}}\right)$$

$$\hat{\sigma}^2=\frac{1}{nT}\mathbf{e}(\lambda)'\mathbf{e}(\lambda) \tag{2.26}$$

dengan $\mathbf{e}\ (\lambda)=\mathbf{y}^*-\lambda(\mathbf{I}_{\mathrm{T}}\otimes\mathbf{W})\mathbf{y}^*-[\mathbf{X}^*-\lambda(\mathbf{I}_{\mathrm{T}}\otimes\mathbf{W})\mathbf{X}^*]\widehat{\boldsymbol{\beta}}$

fungsi *concentrated log likelihood* berbentuk

$$\ln L=-\frac{nT}{2}\ln[\mathbf{e}(\lambda)'\mathbf{e}(\lambda)]+T\ln|\mathbf{I}_{\mathrm{n}}-\lambda\mathbf{W}| \tag{2.27}$$

karena persamaan diatas tidak *closed form* maka diperlukan proses iterasi.

Dengan demikian nilai estimasi spasial efek tetapnya adalah

$$\mathrm{u_i}=\frac{1}{T}\sum_{\mathrm{t=1}}^{\mathrm{T}}(\mathrm{y_{it}}-\mathbf{x}_{\mathrm{it}}'\boldsymbol{\beta})\ ,i=1,\dots,n \tag{2.28}$$

2.1.5 Pengujian Dependensi Spasial

Untuk mengetahui keberadaan efek spasial antar wilayah, perlu dilakukan pengujian dependensi spasial. Beberapa pengujian dependensi spasial yang seringkali dilakukan adalah uji statistik Moran's I dan Lagrange Multiplier (LM).

1. Moran's I

   Uji Moran's I pertama kali diperkenalkan oleh Moran pada tahun 1948. Uji Moran's I pada awalnya dikembangkan untuk mengetahui keberadaan dependensi spasial suatu variabel random. Cliff dan Ord (1981) mengembangkan pengujian ini pada residual suatu model regresi untuk mengetahui keberadaan dependensi spasial pada unit observasi model regresi tersebut. Arbia (2005) dalam Ren, dkk.(2014) mengembangkan uji Moran's I untuk data panel dengan persamaan

$$I_0 = \frac{\hat{\boldsymbol{\varepsilon}}'\mathbf{W}_{nT}\hat{\boldsymbol{\varepsilon}}}{\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}}} \tag{2.29}$$

$\mathbf{W}_{nT} = \mathbf{I}_T \otimes \mathbf{W}$. $\mathbf{W}$ merupakan matriks pembobot spasial yang tetap sepanjang waktu dan $\hat{\boldsymbol{\varepsilon}}$ merupakan set residual hasil regresi OLS data panel.

2. Uji LM

Anselin, Syabri dan Kho (2006) merumuskan LM lag dan LM error untuk menguji dependensi spasial pada data panel. Uji LM lag yang dikembangkan mengikuti persamaan

$$LM_\rho = \frac{\left(\hat{\boldsymbol{\varepsilon}}'\mathbf{W}_{nT}\mathbf{y}/\hat{\sigma}^2\right)^2}{J} \tag{2.30}$$

sedangkan persamaan untuk LM error adalah

$$LM_\lambda = \frac{\left(\hat{\boldsymbol{\varepsilon}}'\mathbf{W}_{nT}\hat{\boldsymbol{\varepsilon}}/\hat{\sigma}^2\right)^2}{TxT_w} \tag{2.31}$$

dengan

$$\hat{\sigma}^2 = \frac{\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}}}{nT}$$

$$J = \frac{\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}}}{nT\hat{\sigma}^2}\left[\left(\mathbf{W_{nT}X}\hat{\boldsymbol{\beta}}\right)'\left(\mathbf{I}_{nT} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}\right)\left(\mathbf{W_{nT}X}\hat{\boldsymbol{\beta}}\right) + TT_w\hat{\sigma}^2\right]$$

$$T_W = trace(\mathbf{WW} + \mathbf{W}'\mathbf{W})$$

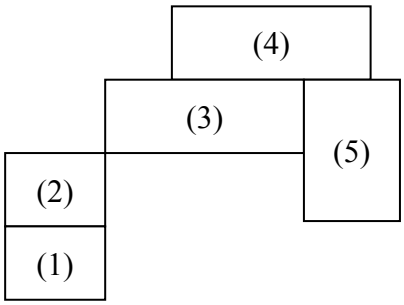
$\mathbf{W}$ merupakan matriks pembobot spasial yang tetap sepanjang waktu dan $\hat{\boldsymbol{\varepsilon}}$ merupakan set residual hasil regresi OLS data panel. $\hat{\boldsymbol{\beta}}$ adalah vektor koefisien regresi OLS data panel. Anselin dkk. (2006) juga mengembangkan robust LM lag dan robust LM error dengan persamaan

$$robust\ LM_\rho = \frac{\left(\hat{\boldsymbol{\varepsilon}}'\mathbf{W_{nT}y}/\hat{\sigma}^2 - \hat{\boldsymbol{\varepsilon}}'\mathbf{W_{nT}}\hat{\boldsymbol{\varepsilon}}/\hat{\sigma}^2\right)^2}{J - TT_w} \tag{2.32}$$

$$robust\ LM_{\lambda} = \frac{\left(\frac{\hat{\boldsymbol{\varepsilon}}'\mathbf{W_{nT}}\hat{\boldsymbol{\varepsilon}}}{\hat{\sigma}^2} - \left(\frac{TT_w}{J}\right)\left(\frac{\hat{\boldsymbol{\varepsilon}}'\mathbf{W_{nT}}\hat{\boldsymbol{\varepsilon}}}{\hat{\sigma}^2}\right)\right)^2}{TT_w\left(1 - \frac{TT_w}{J}\right)} \tag{2.33}$$

2.1.6 Matriks Penimbang Spasial

Matriks penimbang spasial merepresentasikan kedekatan suatu lokasi dengan lokasi yang lain. Lesage (1999) menjelaskan beberapa metode dalam mendefinisikan persinggungan (*contiguity*) dalam membentuk matriks penimbang spasial antara lain *Linear Contiguity, Rook Contiguity, Bishop Contiguity, Double Linear Contiguity, Double Rook Contiguity dan Queen Contiguity*.



Gambar 2.1 Ilustrasi persinggungan (*Contiguity*)

Sumber : Lesage (1999)

Dalam penelitian ini digunakan metode *Queen Contiguity*. *Queen Contiguity* (persinggungan sisi sudut), nilai $w_{ij} = 1$ untuk daerah yang bersisian (*common side*) maupun bertemu titik sudutnya (*common vertex*) dengan daerah yang menjadi perhatian, sedangkan $w_{ij} = 0$ untuk lainnya. Jika daerah 3 menjadi perhatian, maka $w_{23} = 1$, $w_{34} = 1$ dan $w_{35} = 1$ sedangkan yang lain bernilai 0. Sehingga matriks penimbang spasial yang terbentuk adalah

$$W_0 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

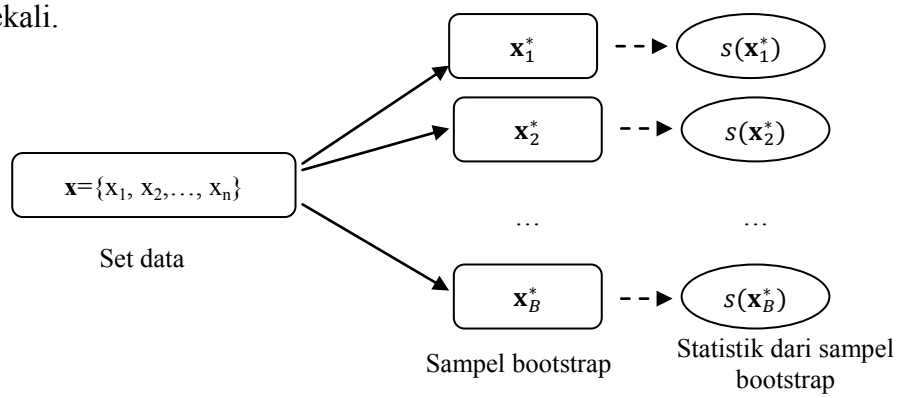
dimana baris dan kolom menyatakan daerah yang ada pada peta. Matriks penimbang spasial adalah matriks simetris dengan kaidah bahwa diagonal utama selalu bernilai nol. Dalam penggunaannya seringkali dilakukan normalisasi baris sehingga memperoleh jumlah setiap baris sama dengan satu.

$$W_{queen} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0,5 & 0 & 0,5 & 0 & 0 \\ 0 & 0,33 & 0 & 0,33 & 0,33 \\ 0 & 0 & 0,5 & 0 & 0,5 \\ 0 & 0 & 0,5 & 0,5 & 0 \end{bmatrix}$$

2.2 Bootstrap

Efron (1979) memperkenalkan metode berbasis komputasi yang merupakan salah satu teknik nonparametrik dan resampling untuk mengestimasi standar error $\hat{\tau}$ yang disebut dengan Bootsrap. Bootstrap adalah sebuah metode simulasi berdasarkan data sebagai alternatif dari metode eksak ketika distribusi sampling dari suatu statistik tidak diketahui atau sulit ditemukan.

Jika $\mathbf{x} = (x_1, x_2, \dots, x_n)$ adalah suatu vektor yang menyatakan suatu sampel data dari populasi dengan fungsi distribusi yang tidak diketahui yang memiliki statistik $s(\mathbf{x})$, maka simulasi bootstrap didasarkan pada set data baru $\mathbf{x}^* = x_1^*, x_2^*, \dots x_n^*$ yang merupakan sampel random yang diambil dengan pengembalian dari suatu populasi n obyek pengamatan. Dengan kata lain, set data bootstrap $\mathbf{x_1^*}, \mathbf{x_2^*}, \dots \mathbf{x_B^*}$ terdiri atas kombinasi set data asli $x_1, x_2, \dots, x_n$ dengan beberapa sampel yang muncul sekali, dua kali dan seterusnya atau bahkan tidak muncul sama sekali.



Gambar 2.2. Skema proses bootstrap

Algoritma bootstrap diawali dengan membangkitkan B buah sampel yang saling bebas masing-masing berukuran n, yaitu $\mathbf{x}_1^*, \mathbf{x}_2^*, \dots \mathbf{x}_B^*$, sehingga diperoleh statistik dari replikasi sebanyak B tersebut $s(\mathbf{x}_b^*)$ dengan $b = 1,2,..,B$. Jika $s(\mathbf{x})$ adalah rata-rata data pengamatan, maka $s(\mathbf{x}_b^*)$ adalah rata-rata data sampel bootstrap. Ilustrasi proses bootstrap disajikan pada Gambar 2.2.

Langkah-langkah untuk mengestimasi bootstrap standar error antara lain :

1. Menentukan sampel bebas bootstrap $\mathbf{x}_1^*, \mathbf{x}_2^*, \dots \mathbf{x}_B^*$ dimana masing-masing sampel terdiri atas n data yang diambil dari set data aslinya dengan pengembalian.
2. Mengevaluasi replikasi pada setiap sampel bootstrap yang terbentuk. $\hat{\tau}_b^* = s(\mathbf{x}_\mathrm{b}^*)\,, b = 1,2, \dots, B$ dimana $s(\mathbf{x}_\mathrm{b}^*)$ merupakan rata-rata set data hasil bootstrap dengan

$$s(\mathbf{x}_\mathrm{b}^*) \;= \bar{\mathrm{x}}^* = \frac{1}{n}\sum_{i=1}^{n} \mathrm{x}_\mathrm{i}^* \tag{2.34}$$

3. Mengestimasi standar error $\hat{\tau}$ sebanyak B replikasi.

$$se(\hat{\tau}_b^*) = \left\{\frac{\sum_{b=1}^{B}(\hat{\tau}_b^* - \hat{\tau}^*)^2}{(B-1)}\right\}^{1/2}, b = 1,2, \dots, B$$

$$\hat{\tau}^* = \frac{1}{B}\sum_{b=1}^{B} \hat{\tau}_b^* \tag{2.35}$$

Penduga selang τ diperoleh dengan pendekatan persentil. Setelah diperoleh $\hat{\tau}_b^*$ untuk setiap replikasi, lalu diurutkan sehingga $\hat{\tau}_1^* \leq \hat{\tau}_2^* \leq \cdots \leq \hat{\tau}_B^*$. Maka batas atas dan batas bawah selang kepercayaannya adalah

$$\left[\hat{\tau}_{low}^*, \hat{\tau}_{up}^*\right] = \left[\hat{\tau}_{B.(\alpha/2),}^*, \hat{\tau}_{B.(1-\alpha/2)}^*\right]$$

untuk B=1000 dan $\alpha = 5\%$, maka batas bawah selang kepercayaan adalah elemen ke-25 dan batas atas selang kepercayaan adalah elemen ke-975 dari barisan $\hat{\tau}_b^*$ yang telah diurutkan (Schmidheiny, 2010).

Pendekatan paling sederhana dalam uji hipotesis bootstrap adalah dengan menghitung taksiran *p-value*. Dengan $H_0: \hat{\tau} = \tau$, jika dari suatu himpunan $\mathbf{x} = (x_1, x_2, \dots, x_n)$ diperoleh statistik τ, maka suatu himpunan suatu data bootstrap ke-b yaitu $\mathbf{x}_b^*\,, b = 1,2, \dots, B$ yang diambil dari data asli dengan pengembalian berukuran n, memiliki statistik uji $\hat{\tau}_b^*, b = 1,2, \dots, B$. Dari nilai $\hat{\tau}$ dan $\hat{\tau}_1^*, \hat{\tau}_2^*, \dots, \hat{\tau}_B^*$ nilai bootstrap *p-value* adalah

$$\hat{p}^* = \frac{\#\{\hat{\tau}_b^* \geq \hat{\tau}, b = 1,2,\dots,B\}}{B}$$

atau (2.36)

$$\hat{p}^* = \frac{banyaknya\ (\hat{\tau}_b^* \geq \hat{\tau})}{B}$$

Jika nilai *p-value* lebih kurang dari taraf signifikansi α, maka H_0 ditolak.

2.3 Double Bootstrap

Beran (1988) memperkenalkan prosedur double bootstrap yang secara teori menghasilkan nilai *p-value* yang lebih akurat dibandingkan nilai *p-value* bootstrap biasa. Prosedur double bootstrap pada dasarnya dilakukan dengan membangkitkan kembali data baru dari set data bootstrap yang telah dibangkitkan sebelumnya. Dari set data bootstrap tahap pertama yang direplikasi sebanyak B_1 dari set data asli, dilakukan proses bootstrap kembali sebanyak B_2 replikasi, sehingga total jumlah statistik uji yang harus dihitung sebanyak $B_1 + B_1 B_2$.

Setelah mendapatkan set data bootsrap tahap pertama $\mathbf{x}_b^*$ dan nilai statistik uji $\hat{\tau}_b^*, b = 1,2,\dots,B_1$ serta taksiran nilai *p-value* bootstrap $\hat{p}^*$, selanjutnya untuk setiap sampel bootstrap tahap pertama dibangkitkan set data sampel bootstrap tahap kedua yang dunotasikan dengan $\mathbf{x}_{bj}^{**}$ dengan $b = 1,2,\dots,B_1$ dan $j = 1,2,\dots,B_2$. Dari set data sampel bootstrap tahap kedua dapat dihitung nilai statistik uji $\hat{\tau}_{bj}^{**}$, $j = 1,2,\dots,B_2$. Nilai bootstrap *p-value* tahap kedua dihitung dengan

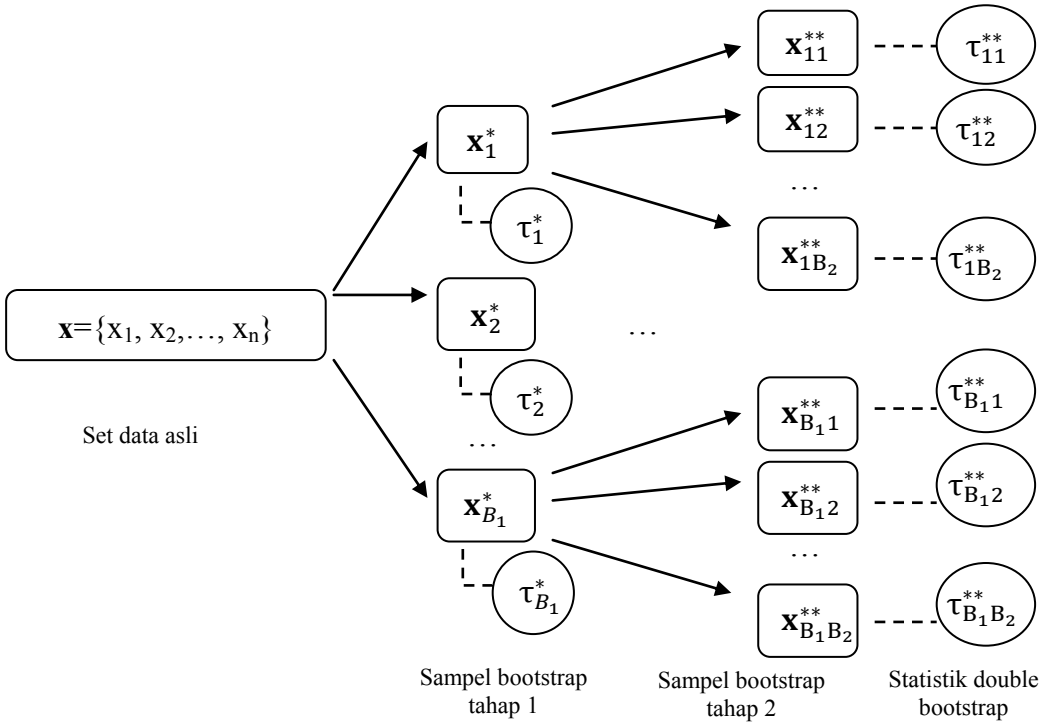
$$\hat{p}_b^{**} = \frac{banyaknya\left(\hat{\tau}_{bj}^{**} \geq \hat{\tau}_b^*\right)}{B_2} \tag{2.37}$$

Sedangkan nilai *p-value* double bootstrap dihitung dengan

$$\hat{p}^{**} = \frac{banyaknya\left(\hat{p}_b^{**} \geq \hat{p}^*\right)}{B_1} \tag{2.38}$$

Jika nilai *p-value* double bootstrap bernilai kurang dari taraf signifikansi α, maka H_0 ditolak. Prosedur ini memiliki kelemahan pada lamanya waktu penghitungan

karena harus membangkitkan set data sampel bootsrap tahap kedua sebanyak B_2 untuk setiap set data sampel bootstrap tahap pertama, sehingga total nilai statistik uji yang dihitung sebanyak $B_1 + B_1 B_2$. Hal ini penting mengingat distribusi τ_{bj}^{**} mungkin saja tidak independen terhadap τ_b^* (MacKinnon, 2006).



Gambar 2.3 Prosedur double bootstrap

2.4 Fast Double Bootstrap (FDB)

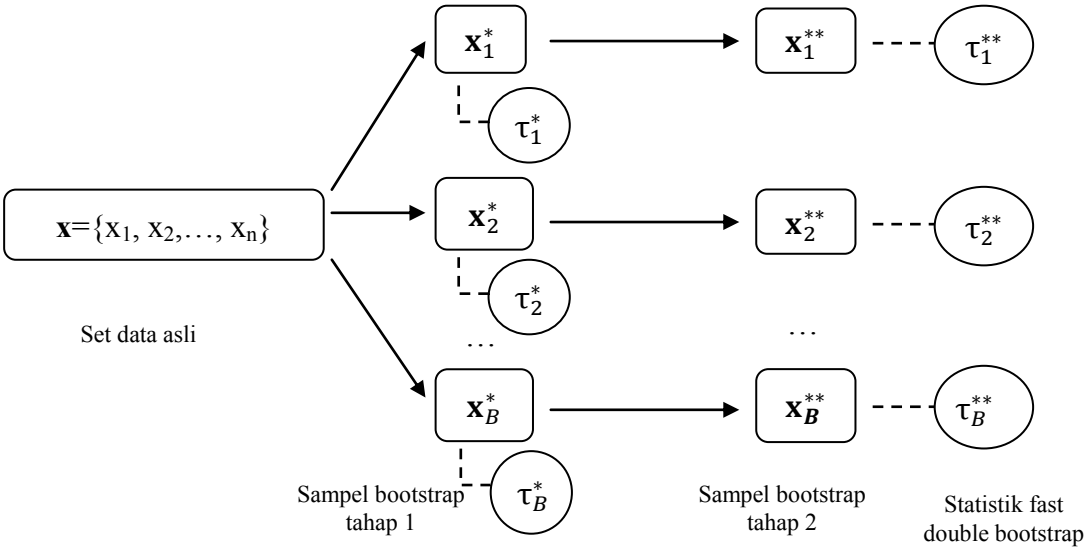
Davidson dan MacKinnon (2001) memperkenalkan prosedur Fast Double Bootstrap (FDB). Dengan mengasumsikan bahwa τ_{bj}^{**} independen terhadap τ_b^*, maka setiap data set sampel bootstrap tahap pertama hanya dibangkitkan sekali pada proses bootstrap tahap kedua. Hal ini mampu mengurangi waktu dan biaya dalam penerapan double bootstrap secara signifikan dengan tingkat akurasi yang sama.

Setelah mendapatkan set data bootsrap tahap pertama $\mathbf{x}_b^*$ dan nilai statistik uji $\tau_b^*, b = 1,2, \ldots, B$ serta taksiran nilai *p-value* bootstrap $\hat{p}^*$, selanjutnya untuk setiap sampel bootstrap tahap pertama dibangkitkan satu set data sampel bootstrap tahap kedua yang dinotasikan dengan $\mathbf{x}_b^{**}$ dengan $b = 1,2, \ldots, B$. Dari set data sampel bootstrap tahap kedua dapat dihitung nilai statistik uji $\tau_b^{**}, b = 1,2, \ldots, B$.

Dari hasil tersebut dapat dihitung $Q^{**}(1-\hat{p}^{*})$ yaitu kuantil ke-$(1-\hat{p}^{*})$ dari τ_b^{**} atau dapat dikatakan nilai τ_b^{**} pada urutan ke-$(1-\hat{p}^{*})B$. Nilai *p-value* FDB dihitung dengan persamaan

$$\hat{p}^{**} = \frac{banyaknya\left(\tau_b^{**} \geq Q^{**}(1-\hat{p}^{*})\right)}{B} \tag{2.34}$$

jika taksiran *p-value* FDB bernilai kurang dari taraf signifikansi α, maka H_0 ditolak.



Gambar 2.4 Prosedur fast double bootstrap

2.4.1 FDB Moran's I

Ren, dkk. (2014) mengembangkan penggunaan bootstrap dengan metode *Fast Double Bootstrap* (FDB) untuk menghitung nilai Moran's I pada data panel dengan sampel kecil. Penghitungan nilai Moran's I yang digunakan mengikuti persamaan 2.29. Dengan demikian, nilai Moran's I untuk setiap set data hasil replikasi diperoleh dengan persamaan

$$\mathrm{I}_{\mathrm{b}}^{**} = \frac{\hat{\boldsymbol{\varepsilon}}_{\mathrm{b}}^{**\prime}\mathbf{W}_{nT}\hat{\boldsymbol{\varepsilon}}_{\mathrm{b}}^{**}}{\hat{\boldsymbol{\varepsilon}}_{\mathrm{b}}^{**\prime}\hat{\boldsymbol{\varepsilon}}_{\mathrm{b}}^{**}} \tag{2.38}$$

dimana $b = 1,2,\ldots,B$. $\mathbf{W}_{nT} = (\mathbf{I}_{\mathrm{T}} \otimes \mathbf{W})$ merupakan matriks pembobot spasial yang tetap sepanjang waktu. $\hat{\boldsymbol{\varepsilon}}_{\mathrm{b}}^{**}$ merupakan set residual hasil resampling tahap kedua

dari residual hasil regresi OLS data panel ($\hat{\boldsymbol{\varepsilon}}$) sesuai dengan persamaan 2.8. B adalah banyaknya replikasi.

Nilai FDB moran's I (I^{**}) diperoleh dengan persamaan

$$\mathrm{I}^{**} = \frac{1}{B}\sum_{b=1}^{B} \mathrm{I}_b^{**} \tag{2.39}$$

dimana $b = 1,2,\dots,B$

Standar error I^{**} dihitung dengan persamaan

$$se(\mathrm{I}^{**}) = \left[\frac{1}{(B-1)}\sum_{b=1}^{B}(\mathrm{I}_b^{**} - \mathrm{I}^{**})^2\right]^{1/2} \tag{2.40}$$

dimana $b = 1,2,\dots,B$. Setelah mengurutkan nilai I_b^{**} di setiap replikasi sehingga $\mathrm{I}_1^{**} \leq \mathrm{I}_2^{**} \leq \cdots \leq \mathrm{I}_B^{**}$, maka diperoleh batas bawah dan batas atas selang kepercayaan $\left[\mathrm{I}_{low}^{**}, \mathrm{I}_{up}^{**}\right] = \left[\mathrm{I}_{B(\alpha/2)}^{**}, \mathrm{I}_{B(1-\alpha/2)}^{**}\right]$.

Nilai *p-value* FDB Moran's I diperoleh dari persamaan

$$\hat{p}_I^{**} = \frac{banyaknya\ (\mathrm{I}_b^{**} \geq Q^{**}(1-\hat{p}_I^{*}))}{B} \tag{2.41}$$

dengan $\hat{p}_I^{*}$adalah nilai *p-value* moran's I pada bootstrap tahap pertama.

$$\hat{p}_I^{*} = \frac{banyaknya\ (\mathrm{I}_b^{*} \geq \mathrm{I}_0)}{B}$$

$$\mathrm{I}_0 = \frac{\hat{\boldsymbol{\varepsilon}}'\mathbf{W}_{nT}\hat{\boldsymbol{\varepsilon}}}{\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}}} \tag{2.42}$$

dengan I_0 adalah nilai Moran's I data asli.

Hipotesis yang digunakan adalah

$\mathrm{H}_0 : \mathrm{I} = \mathrm{I}_0$

$\mathrm{H}_1 : \mathrm{I} \neq \mathrm{I}_0$

Jika nilai *p-value* FDB moran's I kurang dari taraf signifikansi α, maka H_0 ditolak.

2.4.2 FDB LM lag dan FDB LM error

Uji Lagrange Multiplier (LM) pada pengujian model spasial dengan metode *fast double bootstrap* dirumuskan sebagai berikut.

a. FDB LM lag

Uji LM lag dapat digunakan untuk menguji keberadaan dependensi spasial antar wilayah pada variabel dependen. Pengujian LM lag dengan pendekatan FDB dikembangkan dari statistik uji LM lag pada persamaan 2.30. Nilai FDB LM lag diperoleh dengan persamaan

$$\text{LM lag}_{fdb} = \text{LM}_{\rho}^{**} = \frac{1}{B}\sum_{b=1}^{B} \text{LM}_{\rho b}^{**}$$

$$\text{LM}_{\rho b}^{**} = \frac{(\hat{\boldsymbol{\varepsilon}}_{\text{b}}^{**\prime}\, \mathbf{W}_{nT}\mathbf{y}_{\text{b}}^{**}/\hat{\sigma}_b^{**2})^2}{J_b^{**}}$$

(2.43)

dimana $\text{LM}_{\rho b}^{**}$ merupakan nilai LM lag untuk setiap replikasi. $\hat{\boldsymbol{\varepsilon}}_{\text{b}}^{**}$ merupakan set residual hasil resampling tahap kedua dari regresi OLS data panel ($\hat{\boldsymbol{\varepsilon}}$).

$$\mathbf{y}_{\text{b}}^{**} = \mathbf{X}\widehat{\boldsymbol{\beta}} + \hat{\boldsymbol{\varepsilon}}_{\text{b}}^{**}$$

$$(\hat{\sigma}_b^{**})^2 = \frac{\hat{\boldsymbol{\varepsilon}}_{\text{b}}^{**\prime}\hat{\boldsymbol{\varepsilon}}_{\text{b}}^{**}}{nT}$$

$$J_b^{**} = \frac{1}{nT\hat{\sigma}_b^{**2}}\left[\left((\mathbf{I}_{\text{T}}\otimes\mathbf{W})\mathbf{X}\widehat{\boldsymbol{\beta}}\right)'\left(\mathbf{I}_{nT} - \mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\right)((\mathbf{I}_{\text{T}}\otimes\mathbf{W})\mathbf{X}\widehat{\boldsymbol{\beta}}) + TT_W\hat{\sigma}_b^{**2}\right]$$

$$T_W = trace(\mathbf{WW} + \mathbf{W}'\mathbf{W})$$

$$b = 1,2,\dots,B.$$

Standar error LM_{ρ}^{**} dapat diperoleh melalui persamaan

$$se(\text{LM}_{\rho}^{**}) = \left[\frac{1}{(B-1)}\sum_{b=1}^{B}(\text{LM}_{\rho b}^{**} - \text{LM}_{\rho}^{**})^2\right]^{1/2}$$

(2.44)

dimana $b = 1,2,\dots,B$. Setelah mengurutkan nilai $\mathrm{LM}_{\rho b}^{**}$ di setiap replikasi sehingga $\mathrm{LM}_{\rho 1}^{**} \leq \mathrm{LM}_{\rho 2}^{**} \leq \cdots \leq \mathrm{LM}_{\rho B}^{**}$, maka diperoleh batas bawah dan batas atas selang kepercayaan $\left[\mathrm{LM}_{\rho\, low}^{**}, \mathrm{LM}_{\rho\, up}^{**}\right] = \left[\mathrm{LM}_{\rho\, B(\alpha/2)}^{**}, \mathrm{LM}_{\rho\, B(1-\alpha/2)}^{**}\right]$.

Nilai *p-value* FDB LM lag dapat diperoleh dengan persamaan

$$\hat{p}_{LM_\rho}^{**} = \frac{banyaknya\ \left(\mathrm{LM}_{\rho b}^{**} \geq Q^{**}(1-\hat{p}_{LM_\rho}^{*})\right)}{B}$$

$$\hat{p}_{LM_\rho}^{*} = \frac{banyaknya\ \left(\mathrm{LM}_{\rho b}^{*} \geq \mathrm{LM}_{\rho_0}\right)}{B}$$

$$\mathrm{LM}_{\rho b}^{*} = \frac{(\hat{\boldsymbol{\varepsilon}}_{\mathrm{b}}^{*\prime}\, \mathbf{W}_{nT}\mathbf{y}_{\mathrm{b}}^{*}/\hat{\sigma}_b^{*2})^2}{J_b^{*}}$$

$$\mathrm{LM}_{\rho_0} = \frac{\left(\hat{\boldsymbol{\varepsilon}}'\, \mathbf{W}_{nT}\mathbf{y}/\hat{\sigma}^2\right)^2}{J}$$

(2.45)

dimana $b = 1,2,\dots,B$. LM_{ρ_0} adalah nilai LM lag data asli.

Hipotesis yang digunakan adalah

$\mathrm{H}_0 : \mathrm{LM}_\rho = \mathrm{LM}_{\rho_0}$

$\mathrm{H}_1 : \mathrm{LM}_\rho \neq \mathrm{LM}_{\rho_0}$

Jika nilai *p-value* FDB LM lag kurang dari taraf signifikansi α, maka H_0 ditolak.

b. FDB LM error

Uji LM error dapat digunakan untuk menguji keberadaan hubungan spasial residual antar wilayah. Pengujian LM error dengan pendekatan FDB dikembangkan dari statistik uji LM error pada persamaan 2.31. Nilai statistik uji FDB LM error yang dirumuskan sebagai berikut.

$$\text{LM error}_{fdb} = \mathrm{LM}_{\lambda}^{**} = \frac{1}{B}\sum_{b=1}^{B} \mathrm{LM}_{\lambda b}^{**}$$

$$\mathrm{LM}_{\lambda b}^{**} = \frac{(\hat{\boldsymbol{\varepsilon}}_{\mathrm{b}}^{**\prime}\, \mathbf{W}_{nT}\mathbf{y}_{\mathrm{b}}^{**}/\hat{\sigma}_b^{**2})^2}{TxT_W}$$

(2.46)

dimana $\mathrm{LM}_{\lambda b}^{**}$ merupakan nilai LM error untuk setiap replikasi.

$$(\hat{\sigma}_b^{**})^2 = \frac{\hat{\boldsymbol{\varepsilon}}_{\mathrm{b}}^{**\prime}\hat{\boldsymbol{\varepsilon}}_{\mathrm{b}}^{**}}{nT}$$

$$T_W = trace(\mathbf{WW} + \mathbf{W}'\mathbf{W})$$

$$b = 1,2,\dots,B$$

Standar error LM_λ^{**} diperoleh melalui persamaan

$$se(\mathrm{LM}_\lambda^{**}) = \left[\frac{1}{(B-1)}\sum_{b=1}^{B}(\mathrm{LM}_{\lambda b}^{**} - \mathrm{LM}_\lambda^{**})^2\right]^{1/2} \tag{2.47}$$

dimana $b = 1,2,\dots,B$. setelah mengurutkan nilai $\mathrm{LM}_{\lambda b}^{**}$ di setiap replikasi sehingga $\mathrm{LM}_{\lambda 1}^{**} \leq \mathrm{LM}_{\lambda 2}^{**} \leq \cdots \leq \mathrm{LM}_{\lambda B}^{**}$, maka diperoleh batas bawah dan batas atas selang kepercayaan $\left[\mathrm{LM}_{\lambda\, low}^{**}, \mathrm{LM}_{\lambda\, up}^{**}\right] = \left[\mathrm{LM}_{\lambda\, B(\alpha/2)}^{**}, \mathrm{LM}_{\lambda\, B(1-\alpha/2)}^{**}\right]$.

Nilai *p-value* FDB LM error dapat diperoleh dengan persamaan

$$\hat{p}_{LM_\lambda}^{**} = \frac{banyaknya\left(\mathrm{LM}_{\lambda b}^{**} \geq Q^{**}(1-\hat{p}_{LM_\lambda}^{*})\right)}{B}$$

$$\hat{p}_{LM_\rho}^{*} = \frac{banyaknya\left(\mathrm{LM}_{\lambda b}^{*} \geq \mathrm{LM}_{\lambda_0}\right)}{B}$$

$$\mathrm{LM}_{\lambda b}^{*} = \frac{(\hat{\boldsymbol{\varepsilon}}_{\mathrm{b}}^{*\prime}\,\mathbf{W}_{\boldsymbol{n}T}\hat{\boldsymbol{\varepsilon}}_{\mathrm{b}}^{*}/\hat{\sigma}_b^{*2})^2}{TxT_W}$$

$$\mathrm{LM}_{\lambda_0} = \frac{\left(\hat{\boldsymbol{\varepsilon}}'\,\mathbf{W}_{nT}\hat{\boldsymbol{\varepsilon}}/\hat{\sigma}^2\right)^2}{TxT_W} \tag{2.48}$$

dimana $b = 1,2,\dots,B$. LM_{λ_0} adalah nilai LM error data asli.

Hipotesis yang digunakan adalah

$\mathrm{H}_0 : \mathrm{LM}_\lambda = \mathrm{LM}_{\lambda_0}$

$\mathrm{H}_1 : \mathrm{LM}_\lambda \neq \mathrm{LM}_{\lambda_0}$

Jika nilai *p-value* FDB LM error kurang dari taraf signifikansi α, maka H_0 ditolak.

2.4.3 FDB SEM dan FDB SAR

a. FDB SEM

Monchuk, Hayes, Miranowski dan Lambert (2010) melakukan pemodelan spasial bootstrap dengan data *cross section* yang diterapkan pada *spatial error model*. Tahapan pemodelan FDB SEM mengikuti prosedur yang dilakukan pada spasial bootstrap tersebut.

Pemodelan SEM dengan *spatial fixed effect* pada persamaan 2.13 akan menghasilkan set data residual ($\hat{\boldsymbol{\varepsilon}}$). Bootstrap set data residual dilakukan sebanyak dua tahap untuk mendapatkan replikasi *fast double bootstrap* ($\hat{\boldsymbol{\varepsilon}}_{\mathrm{b}}^{**}$). Set data bootstrap residual tahap kedua digunakan untuk mengestimasi $\mathbf{y}_{\mathrm{b}}^{**}$ dengan persamaan

$$\mathbf{y}_{\mathrm{b}}^{**} = \mathbf{X}\widehat{\boldsymbol{\beta}} + (\mathrm{I}_{\mathrm{T}} \otimes \mathbf{u}) + \left[\mathbf{I}_{nT} - \hat{\lambda}(\mathbf{I}_{\mathrm{T}} \otimes \mathbf{W})\right]^{-1} \hat{\boldsymbol{\varepsilon}}_{\mathrm{b}}^{**} \tag{2.49}$$

dengan

$\mathbf{X}$: matriks variabel independen set data asli berukuran *nTx(k+1)*

$\widehat{\boldsymbol{\beta}}$: vektor estimasi parameter yang diperoleh dari SEM dengan *spatial fixed effect* berukuran *(k+1)x1*

$\mathbf{u}$: vektor *spatial fixed effect* berukuran *nx1*

$\mathbf{I}_{nT}$: matriks identitas berukuran *nTx1*

$\mathbf{I}_{\mathrm{T}}$: matriks identitas berukuran *Tx1*

$\hat{\lambda}$: koefisien autokorelasi spasial error

$\mathbf{W}$: matriks pembobot spasial berukuran *nxn*

$\mathbf{y}_{\mathrm{b}}^{**}$ yang telah diperoleh pada setiap set data replikasi dimodelkan dengan SEM untuk memperoleh estimasi parameter setiap set data replikasi. Estimasi parameter model SEM yang diperoleh dengan menggunakan persamaan 2.26 untuk setiap set data replikasi adalah $\hat{\beta}_{\mathrm{b}}^{**}$. **N**ilai estimasi parameter FDB SEM adalah

$$\hat{\beta}^{**} = \frac{1}{B}\sum_{b=1}^{B} \hat{\beta}_{b}^{**} \tag{2.50}$$

untuk $b = 1,2,\ldots,B$. Standar error $\widehat{\beta}^{**}$ dapat diperoleh dengan persamaan

$$se(\hat{\beta}^{**}) = \left[\frac{1}{(B-1)}\sum_{b=1}^{B}(\hat{\beta}_b^{**} - \hat{\beta}^{**})^2\right]^{1/2}$$

(2.51)

dimana $b = 1,2,\dots,B$. Nilai penduga koefisien autokorelasi spasial error setiap set data replikasi ($\hat{\lambda}_b^{**}$) diperoleh dari iterasi pada persamaan SEM setiap set data replikasi. Nilai penduga koefisien autokorelasi spasial error dengan pendekatan FDB mengikuti persamaan

$$\hat{\lambda}^{**} = \frac{\sum_{b=1}^{B}\hat{\lambda}_b^{**}}{B}$$

(2.52)

dengan standar error $\hat{\lambda}^{**}$ adalah

$$se(\hat{\boldsymbol{\lambda}}^{**}) = \left[\frac{1}{(B-1)}\sum_{b=1}^{B}(\hat{\lambda}_b^{**} - \hat{\lambda}^{**})^2\right]^{1/2}$$

(2.53)

dimana $b = 1,2,\dots,B$.

b. FDB SAR

Pemodelan SAR dengan *spatial fix effect* pada persamaan 2.11 akan menghasilkan set data residual ($\hat{\boldsymbol{\varepsilon}}$). Bootstrap set data residual dilakukan sebanyak dua tahap untuk mendapatkan replikasi *fast double bootstrap* ($\hat{\boldsymbol{\varepsilon}}_{\mathrm{b}}^{**}$). Set data bootstrap residual tahap kedua digunakan untuk mengestimasi $\mathbf{y}_{\mathrm{b}}^{**}$ dengan persamaan

$$\mathbf{y}_{\mathrm{b}}^{**} = [\mathbf{I}_{\mathrm{nT}} - \hat{\rho}(\mathbf{I}_{\mathrm{T}}\otimes\mathbf{W})]^{-1}\left(\mathbf{X}\hat{\boldsymbol{\beta}} + (\mathrm{I}_{\mathrm{T}}\otimes\mathbf{u}) + \hat{\boldsymbol{\varepsilon}}_{\mathrm{b}}^{**}\right).$$

(2.54)

dengan

$\mathbf{X}$: matriks variabel independen set data asli berukuran *nTx(k+1)*

$\hat{\boldsymbol{\beta}}$: vektor estimasi parameter yang diperoleh dari SEM dengan *spatial fixed effect* berukuran *(k+1)x1*

$\mathbf{u}$: vektor *spatial fixed effect* berukuran *nx1*

$\mathbf{I}_{NT}$: matriks identitas berukuran *nTx1*

$\mathbf{I}_T$: matriks identitas berukuran *Tx1*

$\hat{\rho}$: koefisien autokorelasi spasial lag

W : matriks pembobot spasial berukuran *nxn*

$\mathbf{y}_b^{**}$ yang telah diperoleh pada setiap set data replikasi dimodelkan dengan model SAR untuk memperoleh estimasi parameter setiap set data replikasi. Estimasi parameter model SAR yang diperoleh dengan menggunakan persamaan 2.24 untuk setiap set data replikasi adalah $\hat{\beta}_b^{**}$.

Nilai estimasi parameter FDB SEM didapatkan dengan persamaan

$$\hat{\beta}^{**} = \frac{1}{B}\sum_{b=1}^{B}\hat{\beta}_b^{**} \tag{2.55}$$

untuk $b = 1,2,\dots,B$. Standar error $\hat{\boldsymbol{\beta}}^{**}$ dapat diperoleh dengan persamaan

$$se(\hat{\beta}^{**}) = \left[\frac{1}{(B-1)}\sum_{b=1}^{B}(\hat{\beta}_b^{**} - \hat{\beta}^{**})^2\right]^{1/2} \tag{2.56}$$

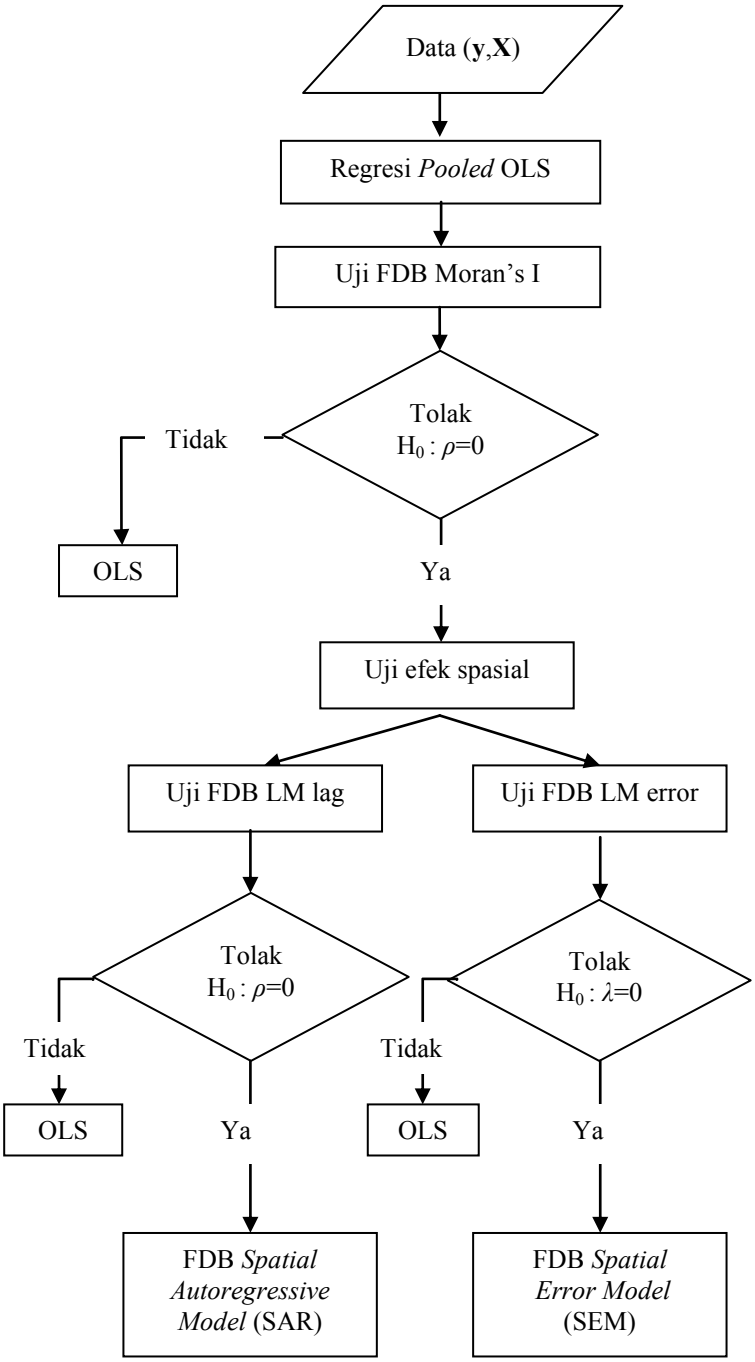
dimana $b = 1,2,\dots,B$. Nilai penduga koefisien autokorelasi spasial lag untuk setiap set data hasil replikasi ($\hat{\rho}_b^{**}$) diperoleh melalui proses iterasi persamaan model SAR pada setiap replikasi. Nilai penduga koefisien autokorelasi spasial lag dengan pendekatan FDB mengikuti persamaan

$$\hat{\rho}^{**} = \frac{\sum_{b=1}^{B}\hat{\rho}_b^{**}}{B} \tag{2.57}$$

dengan standar error $\hat{\rho}^{**}$ adalah

$$se(\hat{\rho}^{**}) = \left[\frac{1}{(B-1)}\sum_{b=1}^{B}(\hat{\rho}_b^{**} - \hat{\rho}^{**})^2\right]^{1/2} \tag{2.58}$$

dimana $b = 1,2,\dots,B$.



Gambar 2.5. Skema Prosedur Pemodelan FDB Regresi Spasial Data Panel

2.5 Tingkat Kemiskinan Daerah dan Faktor-faktor yang Mempengaruhinya

Kemiskinan didefinisikan sebagai suatu kondisi kehidupan dimana sejumlah penduduk tidak mampu mendapatkan sumber daya yang cukup untuk memenuhi kebutuhan dasar minimum untuk hidup layak (Todaro dan Smith, 2007).

Kemiskinan menurut terminologi dibagi menjadi kemiskinan relatif dan kemiskinan absolut. Kemiskinan relatif ditentukan berdasarkan ketidakmampuan untuk mencapai standar kehidupan yang ditetapkan masyarakat setempat sehingga proses penentuannya sangat subjektif. Ketika suatu negara menjadi lebih kaya, maka negara tersebut cenderung merevisi garis kemiskinannya. Sedangkan kemiskinan absolut ditentukan berdasarkan ketidakmampuan untuk mencukupi kebutuhan pokok minimum yang seringkali diterjemahkan dalam bentuk uang. Garis kemiskinan absolut sangat penting jika seseorang akan menilai efek kebijakan anti kemiskinan antar waktu (BPS, 2005).

Berbagai penelitian telah banyak dilakukan untuk mengetahui penyebab dan faktor-faktor yang terkait dengan kemiskinan. Bank Dunia (2009) menyebutkan bahwa kemiskinan dipengaruhi oleh beberapa karakteristik utama, antara lain karakteristik regional, karakteristik komunitas, karakteristik rumah tangga dan karakteristik individu. Kondisi geografis, kondisi iklim dan cuaca, keterisolasian suatu wilayah, pemerintahan daerah yang baik digambarkan sebagai faktor yang termasuk dalam karakteristik regional. Ketersediaan infrastruktrur (air bersih, listrik dan akses jalan) yang memadai, serta pelayanan pendidikan dan kesehatan yang baik kepada masyarakat termasuk dalam karakteristik komunitas. Karakteristik rumah tangga dan individu digambarkan dalam aspek demografi, ekonomi dan sosial. Beberapa faktor yang termasuk didalam aspek demografi antara lain angka ketergantungan, struktur umur penduduk, banyaknya orang dalam sebuah rumah tangga, serta jenis kelamin kepala rumah tangga. Aspek ekonomi digambarkan oleh status ketenagakerjaan, kepemilikan property dan jam kerja, sedangkan aspek social digambarkan oleh kondisi kesehatan dan pendidikan.

Niskanen (1996) menemukan fakta bahwa kemiskinan di Amerika Serikat semakin berkurang seiring meningkatnya tingkat pendidikan dan pendapatan perkapita penduduk. Cameron (2000) dalam Prasetyo (2013) menyimpulkan bahwa pengurangan kemiskinan di Jawa diasosiasikan dengan meningkatnya pencapaian pendidikan dan peningkatan pendapatan dari tenaga kerja terdidik dan pendapatan yang didapat pekerja di luar pertanian (sektor industri).

Islam (2003) melakukan penelitian di 23 negara berkembang dan menyimpulkan bahwa kemiskinan dapat berkurang seiring dengan peningkatan

pendidikan (menurunnya persentase buta huruf) dan peningkatan persentase tenaga kerja di sektor industri. Kemiskinan akan meningkat seiring dengan meningkatnya rasio ketergantungan dan meningkatnya persentase tenaga kerja di sektor pertanian. Senada dengan Knowles (2002) yang menyatakan bahwa meningkatnya rasio ketergantungan akan meningkatkan proporsi penduduk yang hidup dalam kemiskinan.

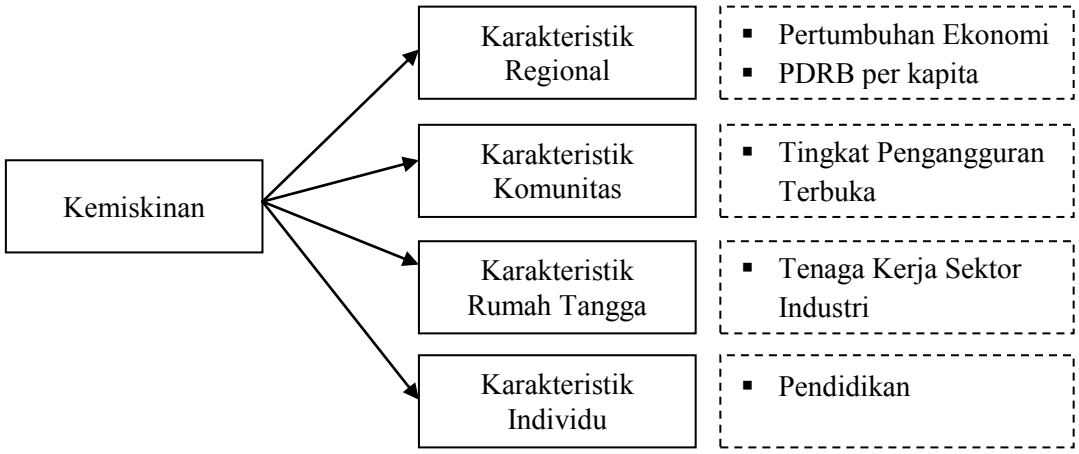
Siregar dan Wahyuniarti (2008) melakukan penelitian untuk mengetahui dampak pertumbuhan ekonomi terhadap penurunan jumlah penduduk miskin di Indonesia. Hasil penelitian tersebut menyimpulkan bahwa pertumbuhan ekonomi berpengaruh signifikan terhadap penurunan jumlah penduduk miskin walaupun dalam magnitude yang relatif kecil. Pertumbuhan ekonomi yang berkualitas dan berkeadilan adalah syarat keharusan dalam penurunan jumlah penduduk miskin. Variabel yang berpengaruh signifikan dan memiliki dampak perubahan yang besar adalah sektor pendidikan.

Astuti (2008) menggunakan analisis regresi data panel untuk memodelkan kemiskinan di Daerah Istimewa Yogyakarta (DIY) dengan Produk Domestik Regional Bruto (PDRB) per kapita, tingkat pendidikan dan tingkat pengangguran sebagai variabel prediktor. Susiati (2012) meneliti tentang hubungan IPM, PDRB per kapita, persentase rumah tangga pengguna air bersih serta belanja publik yang digunakan pemerintah terhadap tingkat kemiskinan kabupaten/kota di DIY.

2.6 Kerangka Konsep Penelitian

Berdasarkan pendekatan yang digunakan oleh Bank Dunia (2009) pada faktor yang mempengaruhi kemiskinan, maka kerangka konsep yang digunakan dalam penelitian ini dirumuskan seperti Gambar 2.5. Karakteristik regional penduduk di NTB dicirikan oleh pertumbuhan ekonomi dan PDRB perkapita. Kedua variabel ini menggambarkan kondisi perekonomian regional masyarakat NTB pada tataran makroekonomi. Karakteristik komunitas di NTB dicirikan oleh tingkat pengangguran terbuka (TPT). Variabel ini menggambarkan kondisi komunitas masyarakat dari aspek ketenagakerjaan. Karakteristik rumah tangga di NTB dicirikan oleh tenaga kerja di sektor industri. Variabel tersebut menggambarkan kemampuan perekonomian yang ditopang oleh pekerja di sektor industri.

Karakteristik individu digambarkan oleh variabel tingkat pendidikan masyarakat terutama penduduk 10 tahun keatas yang belum/tidak pernah bersekolah.

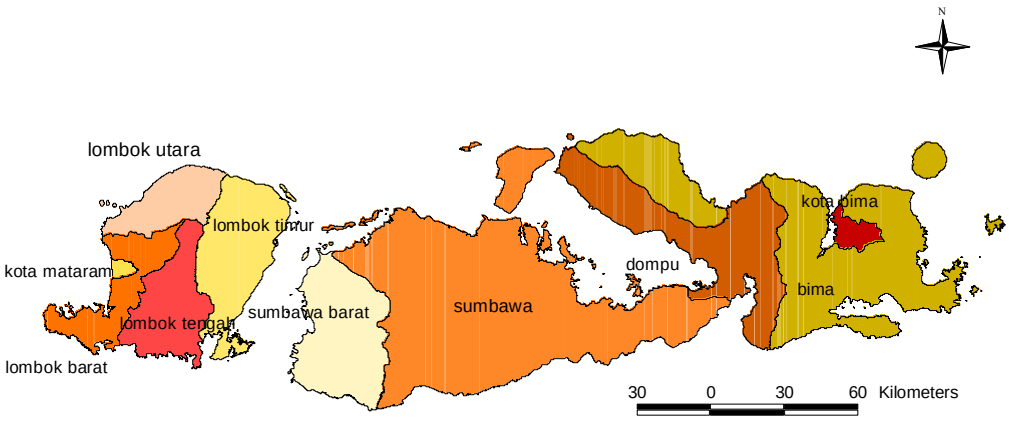


Gambar 2.5. Kerangka Konsep Penelitian

BAB 3
METODOLOGI PENELITIAN

3.1 Sumber Data

Data yang digunakan adalah data sekunder yang berasal dari Survei Sosial Ekonomi Nasional (Susenas) di NTB tahun 2010 - 2012. Wilayah penelitian adalah Provinsi NTB yang terdiri atas sepuluh kabupaten/kota, seperti tertera pada Gambar 3.1.



Gambar 3.1. Wilayah administrasi Provinsi NTB

3.2 Variabel Penelitian

Variabel dependen (Y) yang digunakan adalah persentase penduduk miskin di kabupaten/kota di NTB. Variabel independen (X_i) dirinci sebagai berikut.

Tabel 3.1 Variabel Independen yang Digunakan dalam Pemodelan

Variabel X_i	**Keterangan**
X_1 (Pertumbuhan Ekonomi)	Pertumbuhan ekonomi di setiap kabupaten/kota di NTB.
X_2 (Perkapita)	Pendapatan perkapita masyarakat di setiap kabupaten/kota di NTB.
X_3 (Pendidikan)	Persentase penduduk berumur 10 tahun keatas yang belum/tidak pernah sekolah di setiap kabupaten/kota di NTB.
X_4 (TPT)	Persentase tingkat pengangguran terbuka di setiap kabupaten/kota di NTB.
X_5 (TKI)	Persentase Penduduk yang bekerja di sektor industri di setiap kabupaten/kota di NTB.

Definisi operasional variabel-variabel diatas dijelaskan sebagai berikut :

Tingkat kemiskinan adalah persentase penduduk miskin di setiap kabupaten/kota di NTB. Angka yang digunakan adalah kemiskinan makro yang bersumber dari data Survei Sosial Ekonomi Nasional (Susenas).

Pertumbuhan ekonomi adalah angka laju pertumbuhan ekonomi setiap kabupaten/kota di NTB.

Pendapatan perkapita adalah jumlah nilai tambah bruto yang dihasilkan dalam satu tahun dibagi dengan jumlah penduduk pada tahun tersebut setiap kabupaten/kota di NTB.

Tingkat pendidikan adalah persentase penduduk berumur 10 tahun keatas yang belum/tidak pernah bersekolah di setiap kabupaten/kota di NTB.

Tingkat pengangguran terbuka adalah persentase penduduk usia 15 tahun keatas dengan status pengangguran dari keseluruhan jumlah angkatan kerja di setiap kabupaten/kota di NTB.

Tenaga Kerja Sektor Industri adalah persentase angkatan kerja yang memiliki pekerjaan utama di sektor industri.

3.3 Metode dan Tahapan Penelitian

Metode dan tahapan penelitian yang akan dilakukan untuk mencapai tujuan penelitian adalah sebagai berikut:

1) Uji Dependensi Spasial dan Uji Identifikasi Model.

   Uji yang digunakan untuk mengetahui keterkaitan spasial antar lokasi adalah FDB moran's I sedangkan uji untuk mengidentifikasi model yang diperoleh adalah uji LM dengan langkah-langkah sebagai berikut:

   a. FDB moran's I

   - Melakukan regresi dengan menggunakan set data asli X dan y untuk mendapatkan residual regresi *pooled* OLS.
   - Menentukan matriks penimbang spasial (W) dan W_{nT}.
   - Menghitung nilai moran's I set data asli (I_0).
   - Resampling tahap pertama data residual sebanyak B replikasi ($\hat{\boldsymbol{\varepsilon}}_{\mathbf{b}}^{*}$), menentukan nilai moran's I pada masing-masing replikasi (I_b^*) serta memperoleh nilai bootstrap *p-value* tahap pertama ($\hat{p}_I^*$).

- Dari masing-masing sampel data bootstrap residual tahap pertama yang terbentuk dilakukan proses *bootstrapping* tahap kedua sebanyak satu kali, sehingga terbentuk sebanyak B set data residual bootstrap tahap kedua.
- Menghitung nilai moran's I pada setiap set data dari hasil bootstrap tahap kedua ($\mathrm{I}_{\mathrm{b}}^{**}$).
- Menghitung nilai *p-value fast double bootstrap* moran's I menggunakan kuintil ke-$(1-\hat{p}_I^*)$ dengan memanfaatkan bootstrap *p-value* moran's I tahap pertama.

b. FDB LM lag

Uji identifikasi adanya keterkaitan variabel dependen antar lokasi menggunakan FDB LM lag dengan langkah-langkah sebagai berikut :

- Melakukan regresi dengan menggunakan X dan y untuk mendapatkan residual regresi OLS data panel.
- Menentukan matriks penimbang spasial (W) dan W_{nT}.
- Menghitung LM lag set data asli (LM_{ρ_0})
- Resampling tahap pertama residual set data asli sebanyak B replikasi sehingga diperoleh $\hat{\varepsilon}_{\mathrm{b}}^*$, dan nilai $\mathrm{y}_{\mathrm{b}}^*$ pada masing-masing replikasi.
- Menghitung $\hat{\sigma}_b^{*2}$ untuk mendapatkan J_b^* dan menghitung LM lag dari set data bootstrap tahap pertama (LM_ρ^*) serta memperoleh nilai bootstrap *p-value* tahap pertama ($\hat{p}_{LM_\rho}^*$).
- Resampling kembali masing-masing bootstrap residual tahap pertama sebanyak satu kali sehingga diperoleh $\hat{\varepsilon}_{\mathrm{b}}^{**}$, dan nilai $\mathrm{y}_{\mathrm{b}}^{**}$ pada masing-masing replikasi bootstrap tahap kedua.
- Menghitung $\hat{\sigma}_b^{**2}$ untuk mendapatkan J_b^{**} dan menghitung LM lag dari set data bootstrap tahap kedua (LM_ρ^{**}).
- menghitung nilai *p-value* FDB LM lag ($\hat{p}_{LM_\rho}^{**}$).

c. FDB LM error

Uji identifikasi adanya keterkaitan error antar lokasi menggunakan FDB LM error dengan langkah-langkah sebagai berikut :

- Melakukan regresi dengan menggunakan set data asli X dan y untuk mendapatkan residual regresi OLS data panel.

- Menentukan matriks penimbang spasial (W) dan W_{nT}.
- Menghitung T_W
- Menghitung LM error set data asli (LM_{λ_0})
- Resampling tahap pertama residual set data asli sebanyak B replikasi sehingga diperoleh $\hat{\varepsilon}_b^*$ pada masing-masing replikasi.
- Menghitung $\hat{\sigma}_b^{*2}$ dan menghitung LM error dari set data bootstrap tahap pertama (LM_λ^*) serta memperoleh nilai bootstrap *p-value* tahap pertama ($\hat{p}_{LM_\lambda}^*$).
- Resampling kembali masing-masing bootstrap residual tahap pertama sebanyak satu kali sehingga diperoleh $\hat{\varepsilon}_b^{**}$pada masing-masing replikasi bootstrap tahap kedua.
- Menghitung $\hat{\sigma}_b^{**2}$ dan menghitung LM error dari set data bootstrap tahap kedua (LM_ρ^{**}).
- menghitung nilai *p-value* FDB LM error ($\hat{p}_{LM_\lambda}^{**}$).

2) Mendapatkan Model Regresi Spasial Data Panel dengan *spatial fixed effect*.

a. FDB SAR *spatial fixed effect*

- Dengan data asli dilakukan regresi spasial data panel dengan *spatial fix effect* untuk mendapatkan estimasi parameter $\hat{\rho}_0$dan $\hat{\boldsymbol{\beta}}_0$ menggunakan metode maksimum likelihood.
- Menentukan residualnya.
- Melakukan resampling residual tahap pertama sebanyak B replikasi ($\hat{\boldsymbol{\varepsilon}}_{\mathbf{b}}^*$).
- Melakukan resampling residual tahap kedua sebanyak B replikasi ($\hat{\boldsymbol{\varepsilon}}_{\mathbf{b}}^{**}$).
- Menghitung y_b^{**} pada setiap replikasi.
- Dengan menggunakan $\mathbf{y}_{\mathbf{b}}^{**}$ dan *fixed* **X,** diperoleh $\hat{\beta}_{\mathbf{b}}^{**}$ dan $\hat{\rho}_b^{**}$.
- Menghitung FDB *p-value* $\hat{\beta}^{**}$ dan $\hat{\rho}^{**}$.

b. FDB SEM *spatial fixed effect*

- Dengan data asli dilakukan regresi spasial data panel dengan *spatial fixed effect* untuk mendapatkan estimasi parameter $\hat{\lambda}_0$ dan $\hat{\boldsymbol{\beta}}_0$ menggunakan metode maksimum likelihood.

- Menentukan residualnya.
- Resampling residual tahap pertama sebanyak B replikasi ($\hat{\boldsymbol{\varepsilon}}_{\mathbf{b}}^{*}$).
- Melakukan resampling residual tahap kedua sebanyak B replikasi ($\hat{\boldsymbol{\varepsilon}}_{\mathbf{b}}^{**}$).
- Menghitung y_b^{**} pada setiap replikasi.
- Dengan menggunakan $\mathbf{y}_{\mathbf{b}}^{**}$ dan *fixed* **X,** diperoleh $\hat{\beta}_{\mathbf{b}}^{**}$ dan $\hat{\lambda}_b^{**}$.
- Menghitung nilai $\hat{\beta}^{**}$ dan $\hat{\lambda}^{**}$.
- Menghitung FDB *p-value* $\hat{\beta}^{**}$ dan $\hat{\lambda}^{**}$.

BAB 4
ANALISIS DAN PEMBAHASAN

Pada bagian awal bab ini akan dijelaskan algoritma dan pemrograman uji dependensi spasial dengan FDB moran's I dan FDB LM lag maupun FDB LM error. Algoritma untuk pemodelan regresi spasial data panel dengan FDB SAR maupun FDB SEM akan dilakukan pada bagian selanjutnya. Bagian akhir bab ini menjelaskan tentang hasil uji spasial serta pemodelan kemiskinan di NTB menggunakan regresi spasial data panel dengan pendekatan FDB.

4.1 Penyusunan Algoritma dan Program

4.1.1 Penyusunan Algoritma

1. FDB Moran's I

   Penyusuan algoritma FDB Moran's I didasarkan pada formula penghitungan indeks Moran's I pada persamaan 2.29 dengan pendekatan resampling FDB. Tahapan pengujian dependensi spasial dengan FDB Moran's I dirincikan sebagai berikut.

   a. Melakukan regresi OLS data panel untuk mendapatkan residual.

      Input : $\mathbf{y}$ dan $\mathbf{X}$

      Output : $\hat{\boldsymbol{\varepsilon}}$ dan $\hat{\boldsymbol{\beta}}$

      Algoritma :

      $$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

      $$\widehat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}(\mathbf{X}'\mathbf{Y})$$

      $$\hat{\boldsymbol{\varepsilon}} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}$$

      ```
      results=ols(y,x)
      ```

   b. Menentukan matriks penimbang spasial ($\mathbf{W}_{nT}$).

      Input : $\mathbf{W}$ dan $\mathbf{I}_{\mathrm{T}}$

      Output : $\mathbf{W}_{nT}$

      Algoritma : $\mathbf{W}_{nT} = (\mathbf{I}_{\mathrm{T}} \otimes \mathbf{W})$

      ```
      Ww=kron(It,W)
      ```

   c. Menghitung nilai Moran's I set residual data asli.

      Input : $\hat{\boldsymbol{\varepsilon}}$ dan $\mathbf{W}_{nT}$

Output : I_0

Algoritma : $I_0 = \frac{\hat{\boldsymbol{\varepsilon}}'\mathbf{W}_{nT}\hat{\boldsymbol{\varepsilon}}}{\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}}}$

```
% Moran's I data asli
epe0=e'*e;
ewe0=e'*Ww*e;
mi0=ewe0/epe0;
```

d. Resampling residual tahap pertama sebanyak B replikasi dan menghitung nilai morans'I tiap replikasi tahap pertama.

Input : $\hat{\boldsymbol{\varepsilon}}$, B dan $\mathbf{W}_{nT}$

Output : I_b^*

Algoritma : for i=1:B

$$\hat{\boldsymbol{\varepsilon}}_b^* = randsample\ (\hat{\boldsymbol{\varepsilon}}, nobs, true)$$

$$I_b^* = \frac{\hat{\boldsymbol{\varepsilon}}_b^{*\prime}\mathbf{W}_{nT}\hat{\boldsymbol{\varepsilon}}_b^*}{\hat{\boldsymbol{\varepsilon}}_b^{*\prime}\hat{\boldsymbol{\varepsilon}}_b^*}$$

```
% Bootstrap tahap pertama
for i=1:B;
    er=randsample(e,nobs,true);
    epe1(i)=er'*er;
    ewe1(i)=er'*Ww*er;
     mir1(i)=ewe1(i)/epe1(i);
```

e. Resampling residual tahap kedua sebanyak B replikasi dan menghitung nilai morans'I tiap replikasi tahap kedua.

Input : $\hat{\boldsymbol{\varepsilon}}_b^*$, B dan $\mathbf{W}_{nT}$

Output : I_b^{**}

Algoritma : for i=1:B

$$\hat{\boldsymbol{\varepsilon}}_b^{**} = randsample\ (\hat{\boldsymbol{\varepsilon}}_b^*, nobs, true)$$

$$I_b^{**} = \frac{\hat{\boldsymbol{\varepsilon}}_b^{**\prime}\mathbf{W}_{nT}\hat{\boldsymbol{\varepsilon}}_b^{**}}{\hat{\boldsymbol{\varepsilon}}_b^{**\prime}\hat{\boldsymbol{\varepsilon}}_b^{**}}$$

```
% Bootstrap tahap kedua
for i=1:B;
    err=randsample(er,nobs,true);
    epe(i)=err'*err;
    ewe(i)=err'*Ww*err;
    mir(i)=ewe(i)/epe(i);
   mi=sort(mir);
```

f. Menghitung *p-value*, bias, standar error dan selang kepercayaan statistik uji FDB morans'I.

2. FDB LM Lag

Penyusunan algoritma FDB LM lag didasarkan pada formula yang digunakan untuk mendapatkan LM lag pada persamaan 2.30 dengan menambahkan resampling FDB. Tahapan yang dilakukan dalam algoritma FDB LM lag dirincikan sebagai berikut.

a. Melakukan regresi data panel OLS untuk mendapatkan residual ($\hat{\boldsymbol{\varepsilon}}$) dan estimasi parameter koefisien regresi ($\widehat{\boldsymbol{\beta}}$).

Input : $\mathbf{y}$ dan $\mathbf{X}$

Output : $\hat{\boldsymbol{\varepsilon}}$ dan $\widehat{\boldsymbol{\beta}}$

Algoritma : $\hat{\boldsymbol{\varepsilon}} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}$

$$\widehat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}(\mathbf{X}'\mathbf{y})$$

```
results=ols(ywith,xwith)
```

b. Menentukan matriks penimbang spasial ($\mathbf{W}_{nT}$).

Input : $\mathbf{W}$ dan $\mathbf{I}_{\mathrm{T}}$

Output : $\mathbf{W}_{nT}$

Algoritma : $\mathbf{W}_{nT} = (\mathbf{I}_{\mathrm{T}} \otimes \mathbf{W})$

```
Ww=kron(It,W)
```

c. Menghitung nilai M dan Tr untuk mendapatkan nilai J serta menghitung nilai LM lag.

Input : $\mathbf{W}_{nT}$, $\hat{\boldsymbol{\varepsilon}}$, dan $\mathbf{X}$

Output : LM_{ρ_0}

Algoritma :

$$\hat{\sigma}^2 = \frac{\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}}}{nT}$$

$$\mathbf{M} = \left(\mathbf{I}_{nT} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\right)$$

$$J = \frac{1}{nT\hat{\sigma}^2}\left[\left(\mathbf{W}_{nT}\mathbf{X}\widehat{\boldsymbol{\beta}}\right)'\mathbf{M}(\mathbf{W}_{nT}\mathbf{X}\widehat{\boldsymbol{\beta}}) + \mathrm{T}T_W\hat{\sigma}^2\right]$$

$$T_W = trace(\mathbf{WW} + \mathbf{W}'\mathbf{W})$$

$$\mathrm{LM}_{\rho_0} = \frac{\left(\hat{\boldsymbol{\varepsilon}}'\,\mathbf{W}_{nT}\mathbf{y}/\hat{\sigma}^2\right)^2}{J}$$

```
sige=e'*e/nobs;
WXB2=0;EWE=0;EWY=0;
for t=1:T
    t1=(t-1)*N+1;t2=t*N;
    WXB=W*xwith(t1:t2,:)*b;

M=eye(N)-xwith(t1:t2,:)*inv(xwith(t1:t2,:)'*xwi
th(t1:t2,:))*xwith(t1:t2,:)';
    WXB2=WXB2+WXB'*M*WXB;
    EWE=EWE+e(t1:t2,1)'*W*e(t1:t2,1);
    EWY=EWY+e(t1:t2,1)'*W*ywith(t1:t2,1);
end
Ttr=T*tr;
J=(WXB2+Ttr*sige)/sige;
```

d. Resampling residual tahap pertama untuk menghitung nilai estimasi y setiap replikasi tahap pertama.

Input : $\hat{\boldsymbol{\varepsilon}}_{\mathrm{b}}^{*}$, $\mathbf{X}$ dan $\widehat{\boldsymbol{\beta}}$

Output : $\mathbf{y}_{\mathrm{b}}^{*}$

Algoritma :

for i=1:B

$$\hat{\boldsymbol{\varepsilon}}_{\mathbf{b}}^{*} = randsample\ (\hat{\boldsymbol{\varepsilon}}, nobs, true)$$

$$\mathbf{y}_{\mathrm{b}}^{*} = \mathbf{X}\widehat{\boldsymbol{\beta}} + \hat{\boldsymbol{\varepsilon}}_{\mathrm{b}}^{*}$$

e. Menghitung nilai LM lag tiap replikasi tahap pertama.

Input : $\hat{\boldsymbol{\varepsilon}}_{\mathrm{b}}^{*}$

Output : $\mathrm{LM}_{\rho b}^{*} = \frac{\left(\hat{\boldsymbol{\varepsilon}}_{\mathrm{b}}^{*\prime}\mathbf{W}_{nT}\mathbf{y}_{\mathrm{b}}^{*}/\hat{\sigma}_{b}^{*2}\right)^{2}}{J_{b}^{*}}$

Algoritma :

$$(\hat{\sigma}_{b}^{*})^{2} = \frac{\hat{\boldsymbol{\varepsilon}}_{\mathrm{b}}^{*\prime}\hat{\boldsymbol{\varepsilon}}_{\mathrm{b}}^{*}}{nT}$$

$$\mathbf{M} = \left(\mathbf{I}_{nT} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\right)$$

$$J_{b}^{*} = \frac{1}{nT\hat{\sigma}_{b}^{*2}}\left[\left(\mathbf{W}_{nT}\mathbf{X}\widehat{\boldsymbol{\beta}}\right)'\mathbf{M}(\mathbf{W}_{nT}\mathbf{X}\widehat{\boldsymbol{\beta}}) + TT_{W}\hat{\sigma}_{b}^{*2}\right]$$

$$T_{W} = trace(\mathbf{WW} + \mathbf{W}'\mathbf{W})$$

```
for i=1:B
    er=randsample(e,nobs,true);
    yr=xwith*b+er;
    sige1=er'*er/nobs;
    WXB2=0;EWE=0;EWY=0;
    for t=1:T
        t1=(t-1)*N+1;t2=t*N;
        WXB=W*xwith(t1:t2,:)*b;
```

```
M=eye(N)-xwith(t1:t2,:)*inv(xwith(t1:t2,:)'*xwith(t
1:t2,:))*xwith(t1:t2,:)';
        WXB2=WXB2+WXB'*M*WXB;
        EWE=EWE+er(t1:t2,1)'*W*er(t1:t2,1);
        EWY=EWY+er(t1:t2,1)'*W*yr(t1:t2,1);
    end
    Ttr=T*tr;
    J1=(WXB2+Ttr*sige1)/sige1;
    LMlag1(i)=(EWY/sige1)^2/J1;
    robustLMlag1(i)=((EWY-EWE)/sige1)^2/(J1-Ttr);
end
```

f. Resampling residual tahap kedua untuk menghitung nilai estimasi y setiap replikasi tahap kedua.

Input : $\hat{\boldsymbol{\varepsilon}}_{\mathbf{b}}^{**}$, $\mathbf{X}$ dan $\widehat{\boldsymbol{\beta}}$

Output : $\mathbf{y}_{\mathbf{b}}^{**}$

Algoritma :

for i=1:B

$$\hat{\boldsymbol{\varepsilon}}_{\mathbf{b}}^{**} = randsample\ (\hat{\boldsymbol{\varepsilon}}_{\mathbf{b}}^{*}, nobs, true)$$

$$\mathbf{y}_{\mathbf{b}}^{**} = \mathbf{X}\widehat{\boldsymbol{\beta}} + \hat{\boldsymbol{\varepsilon}}_{\mathbf{b}}^{**}$$

g. Menghitung nilai LM lag tiap replikasi tahap kedua.

Input : $\hat{\boldsymbol{\varepsilon}}_{\mathbf{b}}^{**}$

Output : $\mathrm{LM}_{\rho b}^{**} = \frac{\left(\hat{\boldsymbol{\varepsilon}}_{\mathbf{b}}^{**\prime}\mathbf{W}_{nT}\mathbf{y}_{\mathbf{b}}^{**}/\hat{\sigma}_{b}^{**2}\right)^2}{J_b^{**}}$

Algoritma :

$$(\hat{\sigma}_b^{**})^2 = \frac{\hat{\boldsymbol{\varepsilon}}_{\mathbf{b}}^{**\prime}\hat{\boldsymbol{\varepsilon}}_{\mathbf{b}}^{**}}{nT}$$

$$\mathbf{M} = \left(\mathbf{I}_{nT} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\right)$$

$$J_b^{**} = \frac{1}{nT\hat{\sigma}_b^{**2}}\left[\left(\mathbf{W}_{nT}\mathbf{X}\widehat{\boldsymbol{\beta}}\right)'\mathbf{M}(\mathbf{W}_{nT}\mathbf{X}\widehat{\boldsymbol{\beta}}) + TT_W\hat{\sigma}_b^{**2}\right]$$

$$T_W = trace(\mathbf{WW} + \mathbf{W}'\mathbf{W})$$

```
for i=1:B
    err=randsample(er,nobs,true);
    yrr=xwith*b+err;
    sige2=err'*err/nobs;
    WXB2=0;EWE=0;EWY=0;
    for t=1:T
        t1=(t-1)*N+1;t2=t*N;
        WXB=W*xwith(t1:t2,:)*b;

M=eye(N)-xwith(t1:t2,:)*inv(xwith(t1:t2,:)'*xwith(t
1:t2,:))*xwith(t1:t2,:)';
        WXB2=WXB2+WXB'*M*WXB;
```

```
        EWE=EWE+err(t1:t2,1)'*W*err(t1:t2,1);
        EWY=EWY+err(t1:t2,1)'*W*yrr(t1:t2,1);
    end
    Ttr=T*tr;
    J2=(WXB2+Ttr*sige2)/sige2;
    LMlag2(i)=(EWY/sige2)^2/J2;
    robustLMlag2(i)=((EWY-EWE)/sige2)^2/(J2-Ttr);
    lml=sort(LMlag2);
    rlml=sort(robustLMlag2);
end
```

h. Menghitung *p-value*, bias, standar error dan selang kepercayaan statistik uji FDB LM lag.

3. FDB LM Error

Penyusunan algoritma FDB LM lag didasarkan pada formula yang digunakan untuk mendapatkan LM lag pada persamaan 2.31 dengan menambahkan resampling FDB. Tahapan yang dilakukan dalam algoritma FDB LM lag dirincikan sebagai berikut.

a. Melakukan regresi data panel OLS untuk mendapatkan residual.

Input : $\mathbf{y}$ dan $\mathbf{X}$

Output : $\hat{\boldsymbol{\varepsilon}}$

Algoritma : $\hat{\boldsymbol{\varepsilon}} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}$

```
results=ols(ywith,xwith)
```

b. Menentukan matriks penimbang spasial ($\mathbf{W}_{nT}$).

Input : $\mathbf{W}$ dan $\mathbf{I}_{\mathrm{T}}$

Output : $\mathbf{W}_{nT}$

Algoritma : $\mathbf{W}_{nT} = (\mathbf{I}_{\mathrm{T}} \otimes \mathbf{W})$

```
Ww=kron(It,W)
```

c. Menghitung nilai T_W.

Input : $\mathbf{W}$

Output : T_W

Algoritma : $T_W = trace(\mathbf{WW} + \mathbf{W}'\mathbf{W})$

```
sige=e'*e/nobs;
WXB2=0;EWE=0;EWY=0;
for t=1:T
    t1=(t-1)*N+1;t2=t*N;
    WXB=W*xwith(t1:t2,:)*b;

M=eye(N)-xwith(t1:t2,:)*inv(xwith(t1:t2,:)'*xwi
th(t1:t2,:))*xwith(t1:t2,:)';
```

```
    WXB2=WXB2+WXB'*M*WXB;
    EWE=EWE+e(t1:t2,1)'*W*e(t1:t2,1);
    EWY=EWY+e(t1:t2,1)'*W*ywith(t1:t2,1);
end
Ttr=T*tr;
```

d. Menghitung nilai $\hat{\sigma}^2$ untuk mendapatkan LM error.

Input : $\hat{\boldsymbol{\varepsilon}}$, T_W, n, T dan $\mathbf{W}_{nT}$

Output : $\hat{\sigma}^2$ dan LM_{λ_0}

Algoritma :

$$\hat{\sigma}^2 = \frac{\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}}}{nT}$$

$$\mathrm{LM}_{\lambda_0} = \frac{(\hat{\boldsymbol{\varepsilon}}'\,\mathbf{W}_{nT}\hat{\boldsymbol{\varepsilon}}/\hat{\sigma}^2)^2}{TxT_W}$$

e. Resampling residual tahap pertama untuk menghitung nilai $(\hat{\sigma}_b^*)^2$ dan LM error tiap replikasi tahap pertama.

Input : $\hat{\boldsymbol{\varepsilon}}$, T_W, n, T dan $\mathbf{W}_{\boldsymbol{nT}}$

Output : $(\hat{\sigma}_b^*)^2$ dan $\mathrm{LM}_{\lambda b}^*$

Algoritma :

for i=1:B

$\hat{\boldsymbol{\varepsilon}}_{\mathrm{b}}^* = randsample\ (\hat{\boldsymbol{\varepsilon}}, nobs, true)$

$$(\hat{\sigma}_b^*)^2 = \frac{\hat{\boldsymbol{\varepsilon}}_{\mathrm{b}}^{*\prime}\hat{\boldsymbol{\varepsilon}}_{\mathrm{b}}^*}{\mathrm{nT}}$$

$$\mathrm{LM}_{\lambda b}^* = \frac{\left(\hat{\boldsymbol{\varepsilon}}_{\mathrm{b}}^{*\prime}\,\mathbf{W}_{\boldsymbol{nT}}\hat{\boldsymbol{\varepsilon}}_{\mathrm{b}}^*/\hat{\sigma}_b^{*2}\right)^2}{\mathrm{T}xT_W}$$

End

```
for i=1:B
    er=randsample(e,nobs,true);
    yr=xwith*b+er;
    sige1=er'*er/nobs;
    WXB2=0;EWE=0;EWY=0;
    for t=1:T
        t1=(t-1)*N+1;t2=t*N;
        WXB=W*xwith(t1:t2,:)*b;

M=eye(N)-xwith(t1:t2,:)*inv(xwith(t1:t2,:)'*xwith(t
1:t2,:))*xwith(t1:t2,:)';
        WXB2=WXB2+WXB'*M*WXB;
        EWE=EWE+er(t1:t2,1)'*W*er(t1:t2,1);
        EWY=EWY+er(t1:t2,1)'*W*yr(t1:t2,1);
    end
    Ttr=T*tr;
    J1=(WXB2+Ttr*sige1)/sige1;
    LMerror1(i)=(EWE/sige1)^2/Ttr;
```

```
robustLMerror1(i)=((EWE-(Ttr/J1)*EWY)/sige1)^2/(Ttr
*(1-Ttr/J1));
end
```

f. Resampling residual tahap kedua untuk menghitung nilai LM error tiap replikasi tahap kedua.

Input : $\hat{\boldsymbol{\varepsilon}}_{\mathrm{b}}^{*}$, T_W, n, T dan $\mathbf{W}_{\boldsymbol{nT}}$

Output : $(\hat{\sigma}_b^{**})^2$ dan $\mathrm{LM}_{\lambda b}^{**}$

Algoritma :

for i=1:B

$$\hat{\boldsymbol{\varepsilon}}_{\mathrm{b}}^{**} = randsample\ (\hat{\boldsymbol{\varepsilon}}_{\mathrm{b}}^{*}, nobs, true)$$

$$(\hat{\sigma}_b^{**})^2 = \frac{\hat{\boldsymbol{\varepsilon}}_{\mathrm{b}}^{**\prime}\hat{\boldsymbol{\varepsilon}}_{\mathrm{b}}^{**}}{\mathrm{nT}}$$

$$\mathrm{LM}_{\lambda b}^{**} = \frac{\left(\hat{\boldsymbol{\varepsilon}}_{\mathrm{b}}^{**\prime}\,\mathbf{W}_{\boldsymbol{nT}}\hat{\boldsymbol{\varepsilon}}_{\mathrm{b}}^{**}/\hat{\sigma}_b^{**2}\right)^2}{\mathrm{T}xT_W}$$

End

```
for i=1:B
    err=randsample(er,nobs,true);
    yrr=xwith*b+err;
    sige2=err'*err/nobs;
    WXB2=0;EWE=0;EWY=0;
    for t=1:T
        t1=(t-1)*N+1;t2=t*N;
        WXB=W*xwith(t1:t2,:)*b;

M=eye(N)-xwith(t1:t2,:)*inv(xwith(t1:t2,:)'*xwith(t
1:t2,:))*xwith(t1:t2,:)';
        WXB2=WXB2+WXB'*M*WXB;
        EWE=EWE+err(t1:t2,1)'*W*err(t1:t2,1);
        EWY=EWY+err(t1:t2,1)'*W*yrr(t1:t2,1);
    end
    Ttr=T*tr;
    LMerror2(i)=(EWE/sige2)^2/Ttr;
robustLMerror2(i)=((EWE-(Ttr/J2)*EWY)/sige2)^2/(Ttr
*(1-Ttr/J2));
    lme=sort(LMerror2);
    rlme=sort(robustLMerror2);
end
```

g. Menghitung *p-value*, bias, standar error dan selang kepercayaan statistik uji FDB LM error.

Algoritma yang digunakan dalam pemodelan regresi spasial data panel dengan pendekatan FDB dijelaskan sebagai berikut.

4. FDB SAR

Penyusunan algoritma FDB SAR didasarkan pada model SAR serta estimasi parameter model SAR pada persamaan 2.19 sampai dengan persamaan 2.24. Tahapan yang dilakukan diterapkan dalam algoritma sebagai berikut.

a. Melakukan regresi SAR data panel dengan *spatial fixed effect* untuk mendapatkan residual, nilai estimasi parameter koefisien regresi dengan persamaan 2.24 serta iterasi koefisien autokorelasi spasial lag.

Input : $\mathbf{y}$, $\mathbf{X}$ dan $\mathbf{W}$

Output : $\mathbf{Q}$, $\widehat{\boldsymbol{\beta}}$, $\hat{\rho}_0$ dan $\hat{\boldsymbol{\varepsilon}}$

Algoritma :

$$\mathbf{Q} = \mathbf{I}_{nT} - \frac{1}{T}\boldsymbol{\iota}_T\boldsymbol{\iota}_T'\otimes\mathbf{I}_n$$

$$\widehat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{Q}\mathbf{X})^{-1}\mathbf{X}'\mathbf{Q}\left[\mathbf{y} - \hat{\rho}_0(\mathbf{I}_T\otimes\mathbf{W})\right]\mathbf{y}$$

```
results=sar_panel_FE(y,x,W,T,info)
```

b. Melakukan resampling residual tahap pertama sebanyak B replikasi.

Input : $\hat{\boldsymbol{\varepsilon}}$

Output : $\hat{\boldsymbol{\varepsilon}}_b^*$

Algoritma :

for i=1:B

$\hat{\boldsymbol{\varepsilon}}_b^* = randsample\ (\hat{\boldsymbol{\varepsilon}}, nobs, true)$

end

c. Melakukan resampling residual tahap kedua sebanyak B replikasi.

Input : $\hat{\boldsymbol{\varepsilon}}_b^*$

Output : $\hat{\boldsymbol{\varepsilon}}_b^{**}$

Algoritma :

for i=1:B

$\hat{\boldsymbol{\varepsilon}}_b^{**} = randsample\ (\hat{\boldsymbol{\varepsilon}}_b^*, nobs, true)$

end

```
for i=1:B;
    ur=randsample(u,nobs,true);
    urr=randsample(ur,nobs,true);
```

d. Menghitung nilai $\mathbf{y}$ estimasi pada setiap replikasi.

Input : $\hat{\boldsymbol{\varepsilon}}_b^{**}$, $\mathbf{X}$, $\widehat{\boldsymbol{\beta}}$, $\hat{\rho}_0$ dan $\mathbf{W}_{\mathbf{nT}}$

Output : $\mathbf{y}_b^{**}$

Algoritma :

for i=1:B

$$\mathbf{y}_{\mathrm{b}}^{**} = [\mathbf{I}_{\mathrm{nT}} - \hat{\rho}(\mathbf{I}_{\mathrm{T}} \otimes \mathbf{W})]^{-1}\left(\mathbf{X}\widehat{\boldsymbol{\beta}} + (\mathrm{I}_{\mathrm{t}} \otimes \mathbf{u}) + \hat{\boldsymbol{\varepsilon}}_{\mathrm{b}}^{**}\right)$$

end

e. Dengan menggunakan vektor $\mathbf{y}_{\mathrm{b}}^{**}$ dan matriks **X,** nilai estimasi parameter koefisien regresi setiap set data replikasi ($\widehat{\boldsymbol{\beta}}_{\mathrm{b}}^{**}$) dihitung dengan mengikuti persamaan 2.24, sedangkan estimasi parameter koefisien autokorelasi spasial lag setiap set data replikasi diperoleh dari proses iterasi.

f. Menghitung koefisien regresi ($\widehat{\boldsymbol{\beta}}_i^{**}, i = 0,1,2, \dots, k$) dan autokorelasi spasial lag ($\hat{\rho}^{**}$) dengan pendekatan FDB.

Input : $\widehat{\boldsymbol{\beta}}_b^{**}$ dan $\hat{\rho}_b^{**}$

Output : $\widehat{\boldsymbol{\beta}}_i^{**}$ dan $\hat{\rho}^{**}$

Algoritma :

$$\widehat{\boldsymbol{\beta}}_i^{**} = \frac{1}{B}\sum_{b=1}^{B} \hat{\beta}_b^{**}$$

$$\hat{\rho}^{**} = \frac{\sum_{b=1}^{B} \hat{\rho}_b^{**}}{B}$$

g. Menghitung *p-value,* bias, standar error, dan selang kepercayaan $\widehat{\boldsymbol{\beta}}_i^{**}$ dan $\hat{\rho}^{**}$, dan koefisien determinasi model FDB SAR.

5. FDB SEM

Penyusunan algoritma FDB SEM didasarkan pada model SEM serta estimasi parameter model SEM pada persamaan 2.25 sampai dengan persamaan 2.26. Tahapan yang dilakukan diterapkan dalam algoritma sebagai berikut.

a. Melakukan regresi SEM data panel dengan *spatial fixed effect* untuk mendapatkan residual, nilai estimasi parameter koefisien regresi dengan persamaan 2.26 serta iterasi koefisien autokorelasi spasial error.

Input : **y**, **X** dan **W**

Output : $\widehat{\boldsymbol{\beta}}$**,** $\hat{\lambda}_0$ dan $\hat{\boldsymbol{\varepsilon}}$

Algoritma :

$$\widehat{\boldsymbol{\beta}} = \left(\left[\mathbf{X}^* - \hat{\lambda}_0(\mathbf{I}_{\mathrm{T}} \otimes \mathbf{W})\mathbf{X}^*\right]'\left[\mathbf{X}^* - \hat{\lambda}_0(\mathbf{I}_{\mathrm{T}} \otimes \mathbf{W})\mathbf{X}^*\right]\right)^{-1}$$

$$\left[\mathbf{X}^* - \hat{\lambda}_0(\mathbf{I}_\mathrm{T}\otimes\mathbf{W})\mathbf{X}^*\right]'\left[\mathbf{X}^* - \hat{\lambda}_0(\mathbf{I}_\mathrm{T}\otimes\mathbf{W})\mathbf{y}^*\right]$$

```
results=sar_panel_FE(y,x,W,T,info)
```

b. Melakukan resampling residual tahap pertama sebanyak B replikasi.

Input : $\hat{\boldsymbol{\varepsilon}}$

Output : $\hat{\boldsymbol{\varepsilon}}_\mathrm{b}^*$

Algoritma :

for i=1:B

$\hat{\boldsymbol{\varepsilon}}_\mathrm{b}^* = randsample\ (\hat{\boldsymbol{\varepsilon}}, nobs, true)$

end

c. Melakukan resampling residual tahap kedua sebanyak B replikasi.

Input : $\hat{\boldsymbol{\varepsilon}}_\mathrm{b}^*$

Output : $\hat{\boldsymbol{\varepsilon}}_\mathrm{b}^{**}$

Algoritma :

for i=1:B

$\hat{\boldsymbol{\varepsilon}}_\mathrm{b}^{**} = randsample\ (\hat{\boldsymbol{\varepsilon}}_\mathrm{b}^*, nobs, true)$

end

```
for i=1:B;
    ur=randsample(u,nobs,true);
    urr=randsample(ur,nobs,true);
```

d. Menghitung nilai **y** estimasi pada setiap replikasi.

Input : $\hat{\boldsymbol{\varepsilon}}_\mathrm{b}^{**}$, **X**, $\hat{\boldsymbol{\beta}}$, $\hat{\lambda}_0$ dan $\mathbf{W_{nT}}$

Output : $\mathbf{y}_\mathrm{b}^{**}$

Algoritma :

for i=1:B

$\mathbf{y}_\mathrm{b}^{**} = \mathbf{X}\hat{\boldsymbol{\beta}} + (\mathrm{I_T}\otimes\mathbf{u}) + \left[\mathbf{I}_{nT} - \hat{\lambda}(\mathbf{I}_\mathrm{T}\otimes\mathbf{W})\right]^{-1}\hat{\boldsymbol{\varepsilon}}_\mathrm{b}^{**}$

end

e. Dengan menggunakan vektor $\mathbf{y}_\mathrm{b}^{**}$ dan matriks **X,** nilai estimasi parameter koefisien regresi setiap set data replikasi ($\hat{\beta}_\mathrm{b}^{**}$) dihitung dengan mengikuti persamaan 2.24, sedangkan estimasi parameter koefisien autokorelasi spasial error setiap set data replikasi diperoleh dari proses iterasi.

f. Menghitung nilai estimasi parameter koefisien autokorelasi spasial error dan koefisien regresi dengan pendekatan FDB.

Input : $\hat{\beta}_b^{**}$ dan $\hat{\lambda}_b^{**}$

Output : $\hat{\beta}_i^{**}$ dan $\hat{\lambda}^{**}$

Algoritma :

$$\hat{\beta}_i^{**} = \frac{1}{B}\sum_{b=1}^{B}\hat{\beta}_b^{**}$$

$$\hat{\lambda}^{**} = \frac{\sum_{b=1}^{B}\hat{\lambda}_b^{**}}{B}$$

g. Menghitung *p-value,* bias, standar error, dan selang kepercayaan $\hat{\beta}_i^{**}$ dan $\hat{\lambda}^{**}$, serta koefisien determinasi FDB SEM.

4.1.2 Penyusunan Program

Program aplikasi yang disusun meliputi matlab code untuk pengujian dependensi spasial yaitu FDB Moran's I, FDB Lagrange Multiplier, serta estimasi parameter pada pemodelan FDB *Spatial Autoregressive Model* (SAR) data panel maupun FDB *Spatial Error Model* dengan asumsi *spatial fixed effect*. Program aplikasi disusun dengan menambahkan *syntax* untuk resampling *fast double bootstrap* (FDB) pada *function* spasial ekonometrika data panel yang telah dibuat oleh LeSage (1999) dan dikembangkan oleh Elhorst (2008). Program aplikasi selengkapnya ditampilkan pada Lampiran 9, 10, 11, 12, 13 dan 14.

4.2 Pemodelan Kemiskinan di NTB

4.2.1 Gambaran Persentase Penduduk Miskin di NTB dan Variabel yang Mempengaruhinya

a. Persentase Penduduk Miskin

Gambaran kemiskinan di setiap kabupaten/kota di NTB secara umum disajikan pada Tabel 4.1.

Tabel 4.1 Jumlah dan Persentase Penduduk Miskin, Indeks Kedalaman dan Indeks Keparahan Kemiskinan Kabupaten/Kota di NTB tahun 2010 – 2012.

Tahun	Kabupaten/Kota	Jumlah (ribu jiwa)	Persentase (persen)	Indeks Kedalaman	Indeks Keparahan
2010	Lombok Barat	129,8	21,59	3,35	0,78

	Lombok Tengah	171,4	19,92	3,28	0,85
	Lombok Timur	263,7	23,82	3,57	0,81
	Sumbawa	90,5	21,74	4,46	1,32
	Dompu	43,7	19,89	2,92	0,81
	Bima	85,2	19,41	4,20	1,42
	Sumbawa Barat	25,1	21,81	4,66	1,62
	Lombok Utara	86,3	43,12	9,33	2,73
	Kota Mataram	58,3	14,44	2,36	0,60
	Kota Bima	18,3	12,81	2,46	0,69
2011	Lombok Barat	119,6	19,70	2,99	0,75
	Lombok Tengah	158,0	18,14	2,68	0,63
	Lombok Timur	243,1	21,71	3,40	0,76
	Sumbawa	83,4	19,82	4,38	1,41
	Dompu	40,3	18,17	2,99	0,70
	Bima	78,5	17,66	2,49	0,52
	Sumbawa Barat	23,1	19,88	5,63	2,32
	Lombok Utara	79,5	39,27	8,07	2,36
	Kota Mataram	53,7	13,18	2,41	0,64
	Kota Bima	16,9	11,69	1,70	0,42
2012	Lombok Barat	110,5	17,91	3,10	0,83
	Lombok Tengah	146,0	16,71	2,71	0,68
	Lombok Timur	224,7	20,07	3,25	0,77
	Sumbawa	77,1	18,25	3,87	1,23
	Dompu	37,2	16,57	1,95	0,55
	Bima	72,6	16,22	1,94	0,39
	Sumbawa Barat	21,4	17,60	3,73	1,14
	Lombok Utara	73,5	35,97	7,41	2,23
	Kota Mataram	49,6	11,87	2,09	0,56
	Kota Bima	15,6	10,54	1,75	0,48

Sumber : Badan Pusat Statistik Provinsi NTB

Secara umum persentase penduduk miskin antar kabupaten/kota di NTB hampir tidak terlalu jauh berbeda kecuali persentase penduduk miskin di Kabupaten Lombok Utara yang terlihat sangat tinggi dan berbeda dengan rata-rata persentase penduduk miskin di kabupaten/kota lainnya. Kabupaten Lombok Utara merupakan wilayah pemekaran dari Kabupaten Lombok Barat pada tahun 2008.

(2010)

(2011)

(2012)

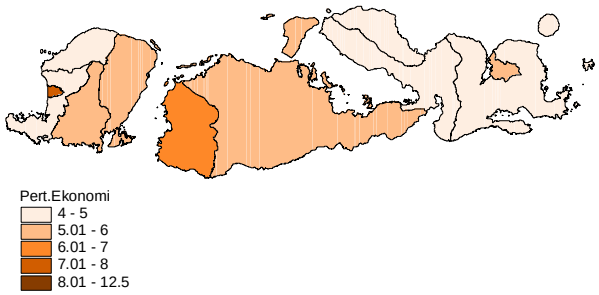
Gambar 4.2 Persentase penduduk miskin di NTB tahun 2010 - 2012

Persentase penduduk miskin di kabupaten/kota di NTB terus berkurang selama periode tahun 2010 sampai dengan 2012. Gambaran persentase penduduk miskin di kabupaten/kota di NTB antara tahun 2010-2012 disajikan pada Gambar 4.2.

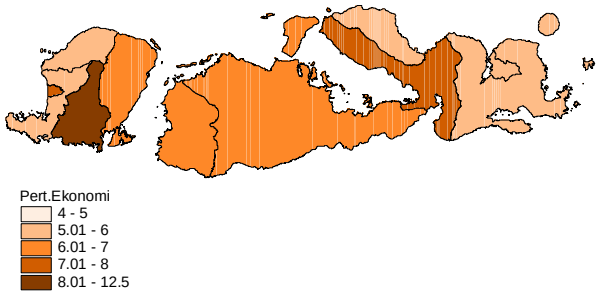
b. Laju Pertumbuhan Ekonomi

Laju pertumbuhan ekonomi kabupaten/kota di NTB antara tahun 2010 – 2012 cenderung meningkat. Peningkatan laju pertumbuhan ekonomi pada periode tersebut tidak terlalu berbeda antar tahun, kecuali Kabupaten Lombok Tengah dan

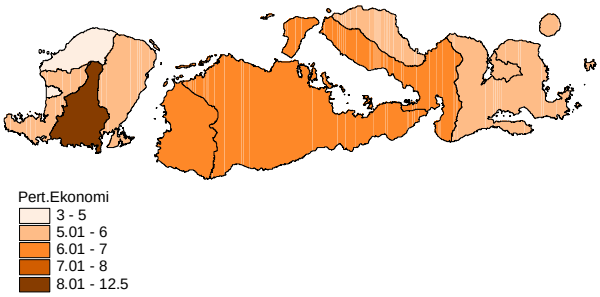
Kota Mataram di tahun 2012. Keduanya mengalami percepatan dan perlambatan laju pertumbuhan ekonomi secara signifikan di tahun 2012.



(2010)



(2011)



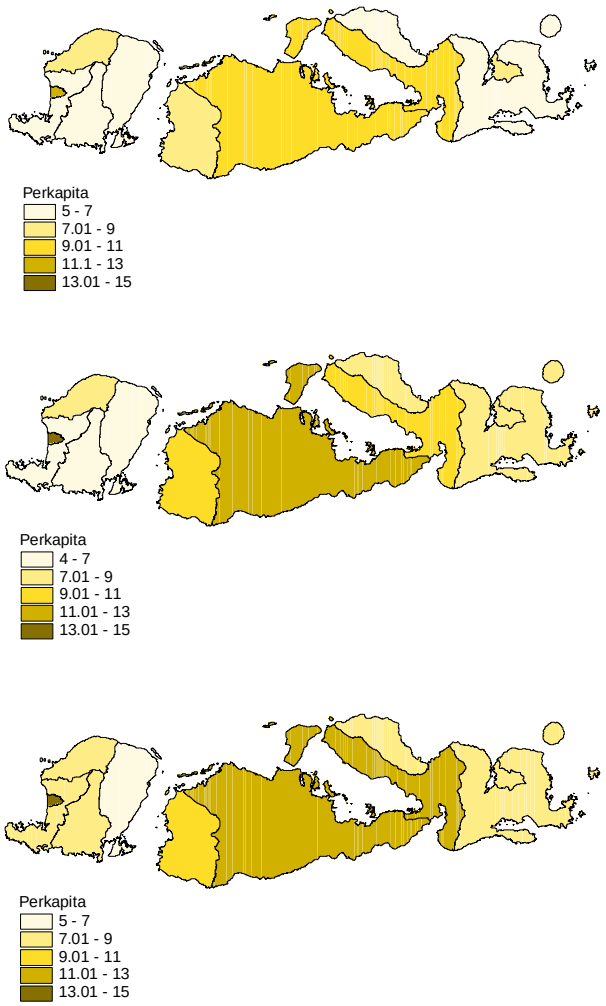
(2012)

Gambar 4.3 Laju pertumbuhan ekonomi kabupaten/kota di NTB tahun 2010 - 2012

c. PDRB Perkapita

PDRB perkapita masyarakat NTB secara umum terus mengalami peningkatan dari tahun ke tahun. Hal ini merupakan gambaran bahwa laju pertumbuhan ekonomi masyarakat NTB lebih tinggi dari laju pertumbuhan penduduknya. PDRB perkapita tertinggi berada di Kota Mataram yaitu sebesar Rp.

14.630.000,00, sedangkan PDRB perkapita terendah berada di Kabupaten Lombok Timur sebesar Rp. 6.940.000,00.

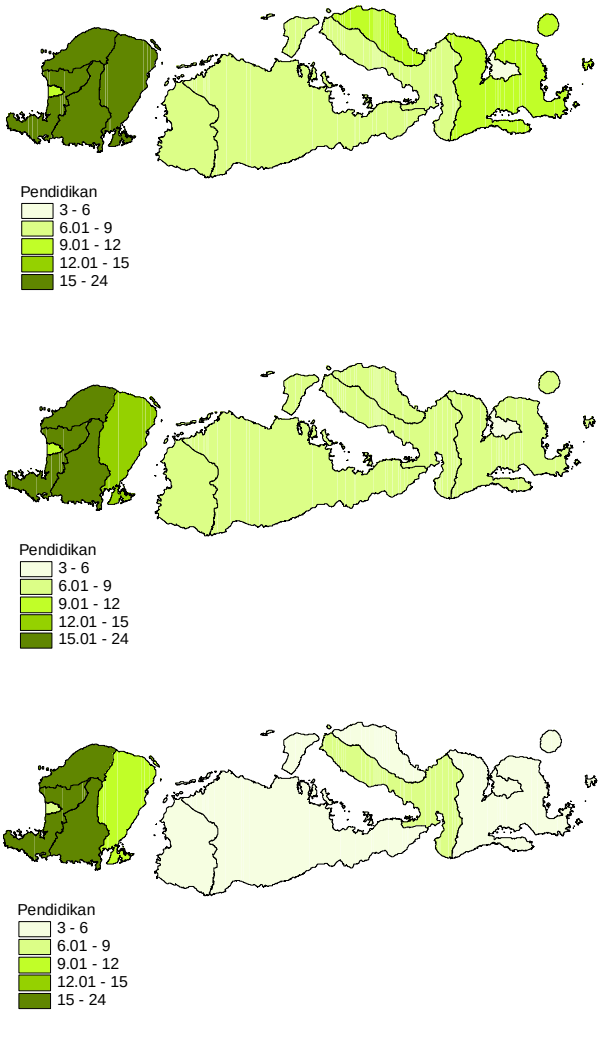


Gambar 4.4 PDRB perkapita Kabupaten/kota di NTB tahun 2010 - 2012

d. Tingkat Pendidikan

Persentase penduduk berusia 10 tahun keatas yang belum/ tidak pernah bersekolah di NTB masih cukup tinggi pada tahun 2012. Persentase penduduk berusia 10 tahun keatas yang belum/ tidak pernah bersekolah tertinggi berada di Kabupaten Lombok Utara (20,03 persen). Kabupaten Lombok Tengah (19,32

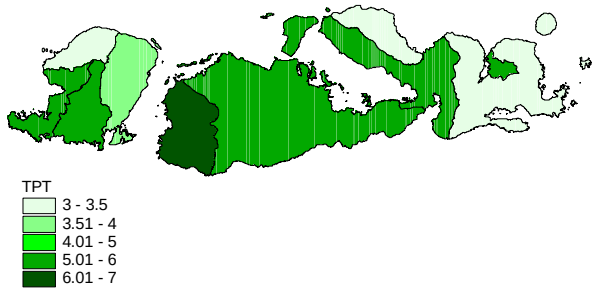
persen) merupakan kabupaten dengan persentase tertinggi kedua dengan selisih yang sangat kecil dengan Kabupaten Lombok Utara.



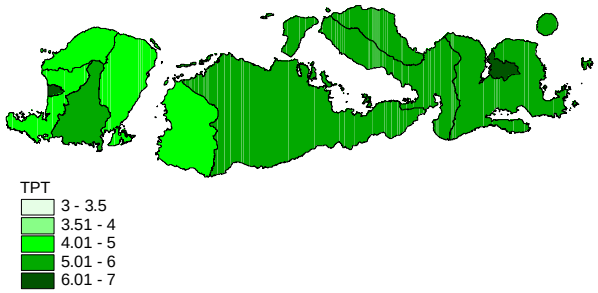
Gambar 4.4 Persentase penduduk berusia 10 tahun ke atas yang belum/ tidak pernah bersekolah per kabupaten/kota di NTB tahun 2010 - 2012

e. Tingkat Pengangguran Terbuka (TPT)

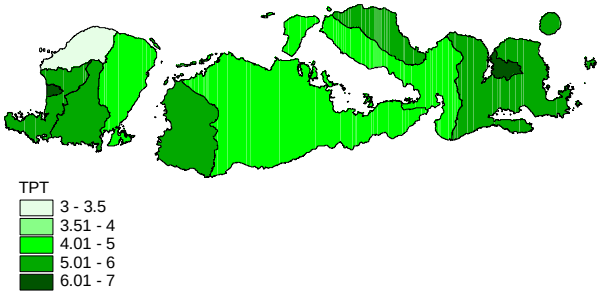
Kabupaten/kota dengan tingkat pengangguran terbuka (TPT) tertinggi pada tahun 2012 adalah Kota Mataram (6,54 persen), sedangkan kabupaten dengan TPT terendah adalah Kabupaten Dompu (4,78 persen).



(2010)



(2011)

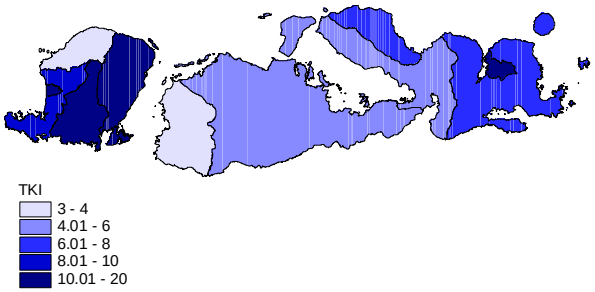


(2012)

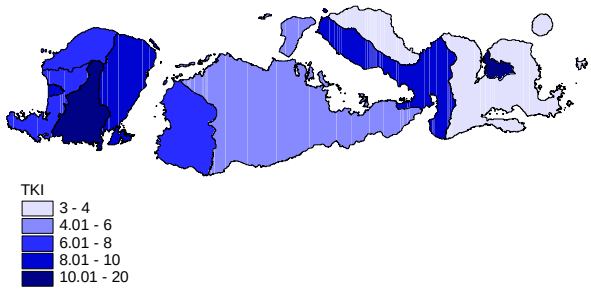
Gambar 4.5 Tingkat Pengangguran Terbuka (TPT) di NTB tahun 2010 – 2012

f. Tenaga Kerja di Sektor Industri

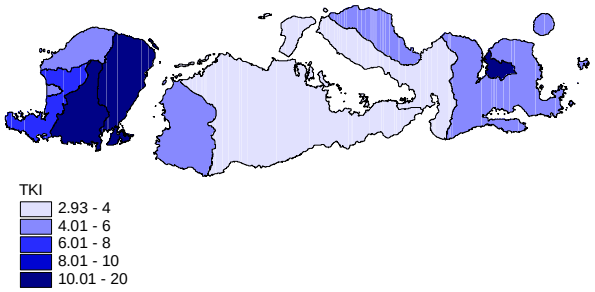
Persentase tenaga kerja di sektor industri secara umum terus mengalami peningkatan dari tahun ke tahun. Hal ini merupakan gambaran pergeseran budaya kerja masyarakat NTB dari sektor pertanian ke sektor industri. Gambaran persentase tenaga kerja di sektor industri dari tahun 2010-2012 disajikan pada Gambar 4.6.



(2010)



(2011)



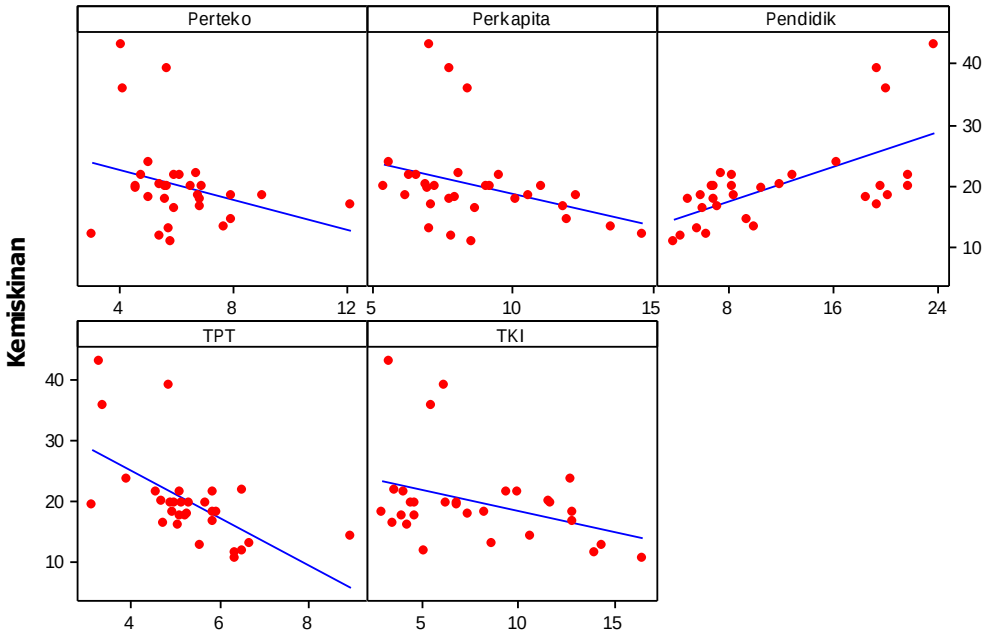
(2012)

Gambar 4.6 Persentase penduduk yang bekerja di sektor industri di NTB tahun 2010 – 2012

g. Pola hubungan setiap variabel independen dengan variabel depanden

Pola hubungan yang terjadi pada setiap variabel independen dengan variabel dependen ditampilkan pada Gambar 4.7. Berdasarkan Gambar 4.7 dapat diketahui bahwa laju pertumbuhan ekonomi, PDRB perkapita, TPT dan persentase penduduk yang bekerja di sektor industri memiliki hubungan yang berkebalikan dengan persentase penduduk miskin di NTB. Semakin tinggi keempat variabel tersebut akan semakin mengurangi persentase penduduk miskin. Persentase penduduk berusia 10 tahun ke atas yang belum/tidak pernah bersekolah, memiliki hubungan

yang searah dengan persentase penduduk miskin. Semakin banyak penduduk yang tidak bersekolah maka persentase penduduk miskin cenderung bertambah.



Gambar 4.7 Pola hubungan persentase penduduk miskin dengan variabel yang mempengaruhinya.

4.2.2 Pemodelan Regresi OLS

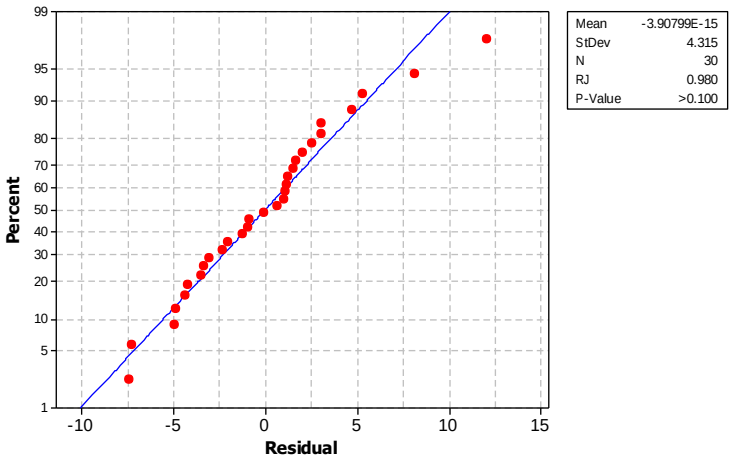
Analisis awal pemodelan kemiskinan di NTB dilakukan dengan regresi OLS data panel.

Tabel 4.2 Nilai Koefisien Regresi, Standar Error, Statistik Uji t dan *p-value* Pemodelan Persentase Penduduk Miskin di NTB dengan Regresi OLS Data Panel.

Parameter	Koefisien	Std. Error	t-stat	p-value	R^2
Konstanta	30,976	6,9300	4,47	0,000	0,664
β_1	-0,3563	0,5732	-0,62	0,540	
β_2	-0,3077	0,5925	-0,52	0,608	
β_3	0,6236	0,1684	3,70	0,001	
β_4	-1,4210	1,1120	-1,28	0,214	
β_5	-0,7367	0,3030	-2,43	0,023	

Berdasarkan hasil regresi OLS data panel diketahui bahwa variabel yang berpengaruh secara signifikan terhadap persentase penduduk miskin di NTB pada α = 0,10 adalah tingkat pendidikan (X_3) dan persentase tenaga kerja di sektor industri (X_5). Nilai koefisien determinasi (R^2) yang diperoleh dari model regresi OLS data panel sebesar 0,664. Hal ini berarti bahwa kelima variabel independen tersebut mampu menjelaskan keragaman persentase penduduk miskin di NTB sebesar 66,4 persen.

Berdasarkan gambar 4.8 dapat diketahui bahwa asumsi normalitas residual relatif telah terpenuhi walaupun ukuran pengamatan cukup kecil.



Gambar 4.8 Plot normalitas residual regresi OLS data panel

4.2.3 Pemodelan Regresi Spasial Data Panel

a. Pengujian Dependensi Spasial

Hasil pengujian dependensi spasial dengan statistik uji Moran's I menghasilkan nilai sebesar 0,2345 dengan *p-value* sebesar 0,0971. Hal ini menunjukkan adanya dependensi spasial persentase penduduk miskin antar kabupaten/kota di NTB. Nilai statistik uji LM lag dan robust LM lag yang diperoleh masing-masing sebesar 1,5216 dan 5,1627 dengan *p-value* sebesar 0,217 dan 0,023. Nilai statistik uji LM error dan robust LM error yang diperoleh masing-masing sebesar 0,0370 dan 3,6781 dengan *p-value* sebesar 0,847 dan 0,0550. Hasil pengujian tersebut menunjukkan kecenderungan model regresi spasial persentase

penduduk miskin di NTB merupakan model SAR. Analisis dengan menggunakan model SARMA tidak dilakukan karena keterbatasan program pengolahan.

Tabel 4.3 Nilai Statistik Uji dan Nilai *p-value* pada Identifikasi Model Regresi Spasial Persentase Penduduk Miskin di NTB

Statistik Uji	**Nilai**	***p-value***
Moran's I	0,2345	0,0971
LM lag	1,5216	0,2170
Robust LM lag	5,1627	0,0230
LM error	0,0370	0,8470
Robust LM error	3,6781	0,0550

b. Pemodelan Regresi Spasial Data Panel

Pemodelan regresi spasial persentase penduduk miskin di NTB menghasilkan estimasi parameter yang ditampilkan pada Tabel 4.4. Variabel yang secara signifikan mempengaruhi persentase penduduk miskin di NTB pada $\alpha = 0,10$ adalah PDRB perkapita (X_2), tingkat pendidikan (X_3) dan persentase tenaga kerja di sektor industri (X_5).

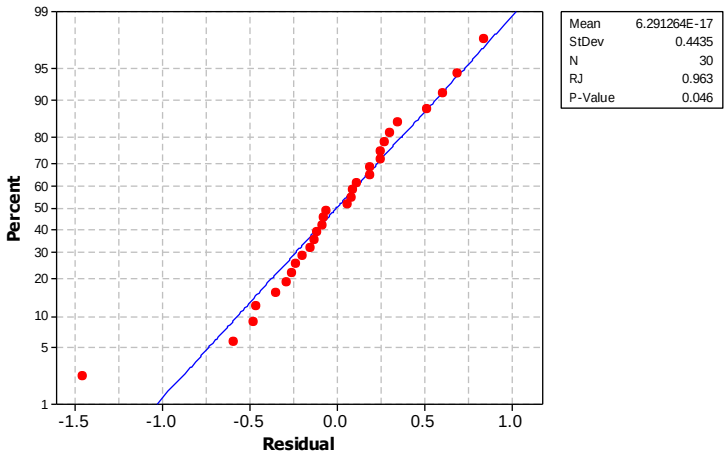
Tabel 4.4 Nilai Koefisien Regresi, Standar Error, Statistik Uji t dan p-value Pemodelan Persentase Penduduk Miskin di NTB dengan Regresi Spasial Data Panel.

Parameter	**Koefisien**	**Std. Error**	**t-stat**	**p-value**	**R^2**
Rho	0,1700	0,046	3,6630	0,000	0,9964
Konstanta	12,5322	50,492	0,2482	0,804	
β_1	0,1336	0,099	1,3490	0,177	**R^2 (corr^2)**
β_2	-0,6150	0,231	-2,6582	0,008	0,9311
β_3	0,4709	0,161	2,9308	0,003	
β_4	0,2331	0,234	0,9980	0,318	
β_5	-0,2642	0,073	-3,6330	0,000	

Nilai koefisien determinasi (R^2) yang dihasilkan sebesar 0,9964. Hal ini menunjukkan bahwa kelima variabel independen dapat menerangkan 99,64 persen

keragaman persentase penduduk miskin di NTB. Nilai R^2 terkoreksi sebesar 0,9311 atau terdapat selisih 0,0653 dengan nilai R^2. Dengan demikian dapat disimpulkan bahwa pengaruh *spatial fixed effect* pada model hanya sebesar 6,53 persen.

Plot residual regresi spasial data panel menunjukkan bahwa asumsi residual berdistribusi normal tidak terpenuhi.



Gambar 4.9 Plot normalitas residual SAR data panel

4.2.3 Pemodelan Regresi Spasial Data Panel dengan pendekatan FDB

a. Pengujian Dependensi Spasial dengan pendekatan FDB

Hasil pengujian dependensi spasial dengan pendekatan FDB ditampilkan pada Tabel 4.5. Nilai FDB Moran's I yang diperoleh sebesar 0,0834 dengan *p-value* 0,1150. Hal ini berarti bahwa FDB Moran's I tidak berbeda secara signifikan dengan Nilai Moran's I data asli atau $I^{**} = I_0$. Hasil tersebut menunjukkan bahwa terdapat dependensi spasial persentase penduduk miskin antar wilayah di NTB.

Tabel 4.5 Nilai Statistik Uji dan Nilai *p-value* pada Identifikasi Model Regresi Spasial Persentase Penduduk Miskin di NTB dengan Pendekatan FDB

Statistik Uji	**Nilai**	***p-value***
FDB Moran's I	0,0834	0,1150
FDB LM lag	5,6243	0,5930
FDB LM error	0,7631	0,8180

Nilai FDB LM lag dan FDB LM error juga menghasilkan kesimpulan yang tidak berbeda secara signifikan dengan LM lag dan LM error data asli.

b. Pemodelan Regresi Spasial Data Panel dengan Pendekatan FDB

Pemodelan SAR data panel dengan pendekatan FDB pada persentase penduduk miskin di NTB menghasilkan estimasi parameter regresi seperti ditunjukkan pada Tabel 4.5 berikut. Replikasi dilakukan sebanyak 200 kali.

Tabel 4.6 Nilai Koefisien Regresi, Standar Error, Statistik Uji t dan p-value Pemodelan Persentase Penduduk Miskin di NTB dengan Regresi Spasial Data Panel menggunakan pendekatan FDB (B=200).

Parameter	Koefisien	Std. Error	Bias	p-value	R^2
Rho	0,1479	0,0495	-0,0181	0,0028	0,9946
Konstanta	13,7683	4,2978	1,1264	0,0014	
β_1	0,1044	0,1009	-0,0245	0,3006	**R^2 ($Corr^2$)**
β_2	-0,6517	0,2286	-0,0341	0,0044	0,9310
β_3	0,4866	0,1587	0,0084	0,0022	
β_4	0,2338	0,2349	-0,0051	0,3194	
β_5	-0,2606	0,0738	0,0039	0,0004	

Nilai koefisien determinasi (R^2) yang diperoleh menunjukkan bahwa 99,46 persen keragaman persentase penduduk miskin di NTB dapat dijelaskan oleh kelima variabel independen menggunakan regresi spasial data panel dengan pendekatan FDB. Nilai Bias dan standar error yang diperoleh cukup rendah pada setiap parameter. Variabel yang secara signifikan mempengaruhi variabel dependen antara lain PDRB perkapita (X_2), tingkat pendidikan (X_3) dan persentase tenaga kerja di sektor industri (X_5). Model SAR data panel dengan pendekatan FDB yang diperoleh adalah

$$\hat{y}_{it} = 0{,}1479 \sum_{\substack{i=1 \\ i \neq j}}^{n} \mathrm{w_{ij} y_j} + \mathrm{u_i} - 0{,}6517\ \mathrm{X_{2t}} + 0{,}4866\ \mathrm{X_{3t}} - 0{,}2606\ \mathrm{X_{5t}}\ .$$

Nilai koefisien autokorelasi spasial lag (ρ) signifikan pada α=0,10. Hal ini mengindikasikan bahwa pengurangan persentase penduduk miskin di NTB memiliki keterkaitan spasial antar kabupaten/kota.

Model SAR Kabupaten Lombok Barat adalah

$$\hat{y}_{lombok\ barat\ ke-t} = 0,0493\, y_{kota\ mataram\ ke-t} + 0,0493\, y_{lomboktengah\ ke-t} + 0,0493\, y_{lombok\ utara\ ke-t}$$
$$-9,480 - 0,6517 x_{2\ lombok\ barat\ ke-t} + 0,4866 x_{3\ lombok\ barat\ ke-t} - 0,2606 x_{5\ lombok\ barat\ ke-t}$$

persentase penduduk miskin di Kabupaten Lombok Tengah, Lombok Utara dan Kota Mataram memiliki peran 0,0493 persen terhadap persentase penduduk miskin di Kabupaten Lombok Barat. Pengurangan kemiskinan pada ketiga wilayah yang berdekatan dengan Kabupaten Lombok Barat tersebut akan berdampak pada pengurangan persentase penduduk miskin di Kabupaten Lombok Barat masing-masing sebesar 0,0493 persen. Peningkatan PDRB perkapita 1 juta rupiah berpeluang menurunkan persentase penduduk miskin sebesar 0,6517 persen dengan asumsi variabel yang lain konstan. Pengurangan persentase penduduk belum/tidak pernah bersekolah sebesar 1 persen di Kabupaten Lombok Barat dapat mengurangi persentase penduduk miskin sebesar 0,4866 persen. peningkatan persentase penduduk yang bekerja di sektor industri di Kabupaten Lombok Barat dapat mengurangi persentase penduduk miskin sebesar 0,2606 dengan asumsi variabel yang lain konstan.

Model SAR Kabupaten Sumbawa adalah

$$\hat{y}_{sumbawa\ ke-t} = 0,0739\, y_{sumbawabarat\ ke-t} + 0,0739\, y_{dompu\ ke-t} + 3,377$$
$$-0,6517 x_{2\ sumbawa\ ke-t} + 0,4866 x_{3\ sumbawa\ ke-t} - 0,2606 x_{5\ sumbawa\ ke-t}$$

persentase penduduk miskin di Sumabawa Barat dan Dompu memiliki peran 0,0739 persen terhadap persentase penduduk miskin di Kabupaten Sumbawa. Pengurangan kemiskinan pada kedua wilayah yang berdekatan dengan Kabupaten Sumbawa tersebut akan berdampak pada pengurangan persentase penduduk miskin di Kabupaten Sumbawa masing-masing sebesar 0,0739 persen. Peningkatan PDRB perkapita 1 juta rupiah berpeluang menurunkan persentase penduduk miskin sebesar 0,6517 persen dengan asumsi variabel yang lain konstan. Pengurangan persentase penduduk belum/tidak pernah bersekolah sebesar 1 persen di Kabupaten

Sumbawa dapat mengurangi persentase penduduk miskin sebesar 0,4866 persen. peningkatan persentase penduduk yang bekerja di sektor industri di Kabupaten Sumbawa dapat mengurangi persentase penduduk miskin sebesar 0,2606 dengan asumsi variabel yang lain konstan.

Model SAR Kota Bima adalah

$$\hat{y}_{kotabima\ ke-t} = 0,1479\, y_{bima\ ke-t} + 0,520 - 0,6517 x_{2\ kotabima\ ke-t} + 0,4866 x_{3\ kotabima\ ke-t} - 0,2606 x_{5\ kotabima\ ke-t}$$

Pengurangan persentase penduduk miskin di Kabupaten Bima berpengaruh sebesar 0,1479 persen terhadap pengurangan persentase penduduk miskin di Kota Bima. Peningkatan PDRB perkapita 1 juta rupiah berpeluang menurunkan persentase penduduk miskin sebesar 0,6517 persen dengan asumsi variabel yang lain konstan. Pengurangan persentase penduduk belum/tidak pernah bersekolah sebesar 1 persen di Kota Bima dapat mengurangi persentase penduduk miskin sebesar 0,4866 persen. peningkatan persentase penduduk yang bekerja di sektor industri di Kota Bima dapat mengurangi persentase penduduk miskin sebesar 0,2606 dengan asumsi variabel yang lain konstan.

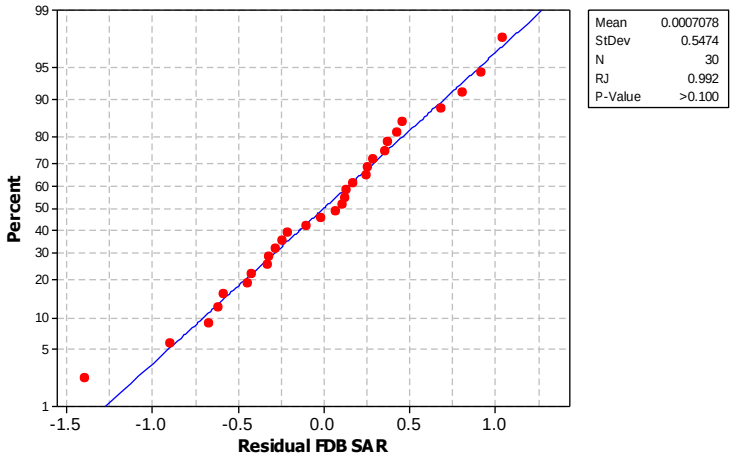
Tabel 4.7 Nilai dan p-value *Spatial Fixed Effect* (SFE) pada Pemodelan Persentase Penduduk Miskin di NTB dengan Regresi Spasial Data Panel menggunakan pendekatan FDB (B=200).

No.	Kabupaten/Kota	Nilai SFE	p-value
1	Lombok Barat	-9,480	0,893
2	Lombok Tengah	-12,968	0,884
3	Lombok Timur	-1,699	0,979
4	Sumbawa	3,377	0,919
5	Dompu	1,383	0,971
6	Bima	0,886	0,980
7	Sumbawa Barat	5,513	0,877
8	Lombok Utara	11,429	0,863
9	Kota Mataram	1,040	0,982
10	Kota Bima	0,520	0,993

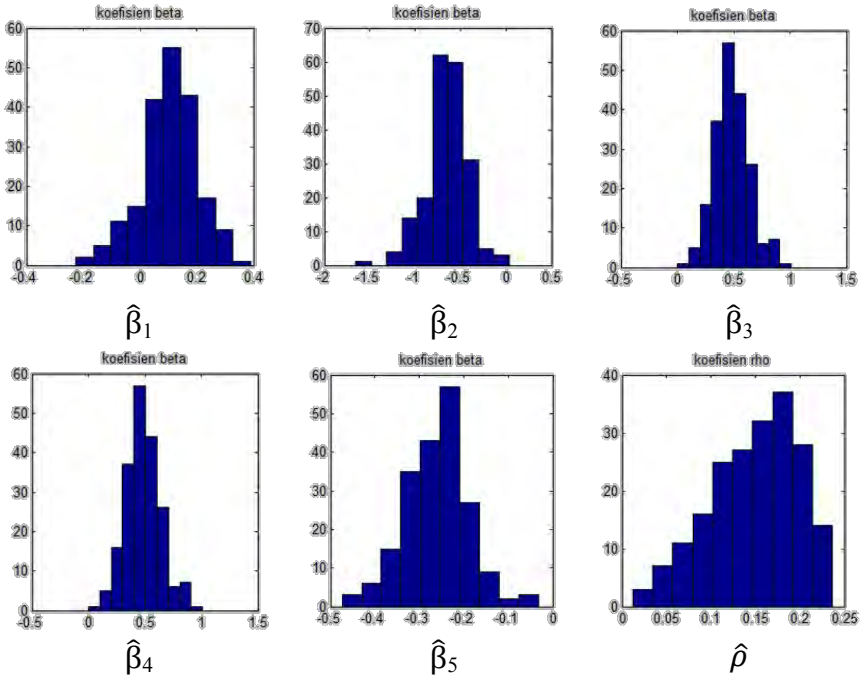
Nilai R^2 terkoreksi yang diperoleh sebesar 0,9310 atau terdapat selisih 0,0636 dari nilai R^2. Berdasarkan hasil tersebut dapat disimpulkan bahwa secara

umum *spatial fixed effect* hanya berpengaruh 6,36 persen pada model. Nilai *p-value* secara keseluruhan menunjukkan nilai yang tidak signifikan. Dengan demikian dapat disimpulkan bahwa tidak terdapat perbedaan signifikan yang ditimbulkan oleh adanya perbedaan efek lokasi di setiap kabupaten/kota.

Berdasarkan Gambar 4.11 dapat diketahui bahwa residual yang diperoleh dari pemodelan SAR dengan pendekatan FDB memenuhi asumsi kenormalan.



Gambar 4.10 Plot normalitas residual SAR data panel dengan pendekatan FDB.



Gambar 4.11 Histogram estimasi parameter SAR dengan pendekatan FDB

Berdasarkan Gambar 4.11 nilai estimasi parameter yang diperoleh dengan pendekatan FDB memiliki standar error yang relatif kecil dan mendekati distribusi normal (*limiting normal distribution*).

BAB 5
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan analisis data yang telah dilakukan pada bab sebelumnya, maka dapat diperoleh beberapa kesimpulan, antara lain :

1. Pengujian dependensi spasial dengan pendekatan FDB Moran's I, FDB LM lag dan FDB LM error menghasilkan nilai yang tidak berbeda secara signifikan dengan uji statistik Moran's I, LM lag dan LM error. Berdasarkan pengujian tersebut dapat diketahui adanya dependensi spasial persentase penduduk miskin antar kabupaten/kota di NTB dan model yang sesuai adalah model SAR data panel.
2. Pemodelan SAR data panel menghasilkan koefisien determinasi yang tinggi (99,64 persen), namun tidak memenuhi asumsi kenormalan residual. Pemodelan SAR data panel dengan pendekatan FDB menghasilkan koefisien determinasi sedikit lebih rendah (99,46 persen), namun asumsi kenormalan residual dapat terpenuhi. Pemodelan SAR data panel dengan pendekatan FDB dapat menerangkan keragaman persentase penduduk miskin di NTB sebesar 99,46 persen dengan pengaruh *spatial fixed effect* sebesar 6,31 persen.
3. Variabel yang mempengaruhi persentase penduduk miskin di NTB secara signifikan adalah PDRB perkapita, persentase penduduk berumur 10 tahun ke atas yang tidak/belum pernah bersekolah dan persentase penduduk yang bekerja di sektor industri.

5.2 Saran

Berdasarkan hasil yang diperoleh, maka beberapa saran untuk penelitian selanjutnya, antara lain :

1. Pada penelitian ini hasil yang diperoleh dengan pendekatan FDB belum menunjukkan nilai yang lebih baik. Oleh karena itu, perlu dikembangkan uji statistik dependensi spasial dengan pendekatan FDB yang mempertimbangkan keberadaan data outlier.

2. Penelitian ini masih terbatas pada asumsi *spatial fixed effect*. Penelitian selanjutnya diharapkan dapat mengembangkan model regresi spasial data panel dengan pendekatan FDB pada asumsi *spatial random effect*.

DAFTAR PUSTAKA

Anselin, L. (1988). *Spatial Econometrics: Methods and Models*. Dordrecht: Kluwer Academic Publisher.

Anselin, L. & Hudak, S. (1992). "Spatial econometrics in practice: A review of software options".*Regional Science and Urban Economics* 22,509-536.

Anselin, L., Syabri, I., Kho, Y. (2006). Geoda: an Introduction to Spatial Data Analysis. *Geographical Analysis.* Vol.38 (1).Pages 5-22.

Astusti, M. B. (2010). *Faktor-faktor yang Mempengaruhi Tingkat Kemiskinan di Provinsi DIY 2002-2008*. Thesis. Universitas Gadjah Mada, Yogyakarta.

Badan Pusat Statistik Provinsi Nusa Tenggara Barat (2010), *Nusa Tenggara Barat Dalam Angka 2010*, BPS. NTB.

_____. (2011), *Nusa Tenggara Barat Dalam Angka 2011*, BPS. NTB

_____. (2012), *Nusa Tenggara Barat Dalam Angka 2012*, BPS. NTB

Baltagi, B.H. (2005). "*Econometrics Analysis of Panel Data" 3rd edition,* John Wiley & Sons Ltd. Chicester, England.

Beran, R. (1988). *Prepivoting Test Statistics: A Bootstrap View of Asymptotic efinements*. J.Am. Stat. Assoc. 83,687-697.

Crandall, M.S. & Weber, B.A. (2004),"Local Social and Economic Conditions, Spatial Concentration of Poverty, and Poverty Dynamics", *Poverty, Policy and Place: Spatial Analysis of Poverty Dynamics, American Journal Agricultural Economics*, 86.5:1276-1281.

Davidson, R. dan MacKinnon, J.G. (2001). *Improving the Reliability of Bootstrap Test with The Fast Double Bootstrap*. Comput. Stat. Data Anal. 51, 3259-3281.

Efron, B. & Thibsirani, R. (1993), *An Introduction to the Bootstrap*. Capital City Press, Chapman & Hall, New York.

Efron, B. (1979), "Bootstrap Methods : Another Look at the Jacknife", *The Annals of Statistics*, Vol. 7 No.1, 1-26

Elhorst, J.P. (2003). "Specification and Estimation of Spatial Panel Data Models", *International Regional Science Review*, 26(3), 244-268.

_____ . (2010a). "Spatial Panel Data Models". *In Handbook of Applied Spatial Analysis, eds. M.M. Fischer and A. Getis*, 377-407, Berlin: Springer.

_____ . (2012). “Matlab Software for Spatial Panels”. *International Regional Science Review* DOI:101177/0160017612452429.

Gujarati, D. N. (2003). *Basic Econometrics.* Fourth Edition. McGrawHill Singapore.

Iradian, Garbis. (2005). *Inequality, Poverty, and Growth: Cross Country Evidence*. IMF Working Paper. Middle East and Central Asia Departement.

Islam, Rizwatul. (2003). *The Nexus of Economic Growth, Employment and Poverty Reduction An Empirical Analysis*. Report on Seminar on Accelerating Growth and Poverty Reduction in Bangladesh. ILO, Geneva.

Kogan, L. (2010). *Small-Sample Inference and Bootstrap*, MIT, Sloan, Fall 2010.

Knowles, J.C. (2002). *A Look at Poverty in The Developing Countries of Asia*. Asia-Pacific Population & Policy, No. 52, January 2000.

Lee, L.F., & Yu, J. (2010a). ”Estimation of Spatial Autoregressive Panel Data Models with Fixed Effects”, *Journal of Econometrics*.

Lee, L.F., & Yu, J. (2010b). *Some Recent Developments in Spatial Panel Data Models. Regional Science and Urban Economics* 40,255-271.

LeSage, J.P. (1999). The Theory and Practice of Spatial Econometrics. University of Toledo.

LeSage, J.P. & Pace, R.K. (2009). Introduction to Spatial Econometrics. Boca Raton. US: CRC Press Taylor & Francis Group.

Lin, K.-P., Long, Z., Wu Mei (2007). ”Bootstrap Test Statistics for Spatial Econometric Models”, *Journal of Econometrics*.

Lin, K.-P., Long, Z., Ou, B. (2009). ”Properties of Bootstrap Moran’s I for Diagnostic Testing A Spatial Autoregressive Linear Regression Model”, *Journal of Econometrics*.

Lynch, S.M. (2003). *Alternative Estimation Strategies*, Soc 504, Princeton University.

MacKinnon, J.G. (2006). Bootstrap Methods in Econometrics. *Journal of Economics*. Vol. 82. Pages S2-S18.

Marsono (2013). *Pemodelan Pengangguran Terbuka di Indonesia dengan Pendekatan Ekonometrika Spasial Data Panel*. Thesis. Institut Teknologi Sepuluh Nopember, Surabaya.

Monchuk, D.C.,Hayes, D.J.,Miranowski, J.A. & Lambert, D.M. (2010). *Inference Based on Alternative Bootstrapping Methods in Spatial Models with an Applicaton to Country Income Growth in the United States*. Working Paper 10-WP 507. May 2010, Centre for Agricultural and Rural Development, Iowa State University.

Niskanen, William A. (1996). *Welfare and Culture of Poverty*. The Cato Jurnal, Vol. 16, No. 1.

Nurlita, M.R. (2006). *Pengujian Hipotesis untuk Masalah Dua Sampel Saling Bebas Menggunakan Fast Double Bootstrap Nonparametrik*. Skripsi. Universitas Islam Bandung.

Prasetyo, S. (2013). Studi Faktor Penyebab Kemiskinan dan Mekanisma Penanggulangan Kemiskinan di Indonesia.

Ren, T., Long, Z., Zhang, R., Chen, Q. (2014). *Moran's I Test of Spatial Panel Data Model – Based on Bootstrap Method*. Shenzhen Polytechnic.

Rusmasari, A. (2011). *Pemodelan Regresi Spasial dengan Pendekatan Residual Bootstrap*. Thesis. Institut Teknologi Sepuluh Nopember, Surabaya.

Schmidheiny, K. (2010), *The Bootstrap, Short Guide to Microeconometrics*, Fall 2010, Universitat Pompeu Fabra.

Suharto, E. (2011), *Robust Lagrange Multiplier pada Pemodelan Regresi Spasial Dependensi*. Thesis, Institut Teknologi Sepuluh Nopember.

Susiati, D. (2012). *Analisis Faktor-faktor yang Mempengaruhi Tingkat Kemiskinan Kabupaten/kota di Provinsi DIY Tahun 2004-2010*. Thesis. Universitas Gadjah Mada, Yogyakarta.

Todaro, M.P. & Smith, S.C. (2006). *Pembangunan Ekonomi*. Terjemahan Hasris Munandar. Jakarta: Erlangga.

Vega, S.H. & Elhorst, J.P. (2013). "On Spatial Econometric Models, Spillover Effect, and W". *Paper presented in Ph.D course spatial econometrics*. April 15-19, 2013. Paris.

World Bank. (2009). *Handbook of Poverty and Inequality*.

Yang, Z.L. (2011). *LM Test of Spatial Dependence Based on Bootstrap Critcal Values*. Working Paper. School of Economics, Singapore Management University.

LAMPIRAN

Lampiran 1. Data yang Digunakan

Tahun	Kabupaten/Kota	Y	X_1	X_2	X_3	X_4	X_5
2012	Lombok Barat	17.91	5.03	7.97	18.41	5.3	7.37
	Lombok Tengah	16.71	12.16	7.10	19.32	5.85	12.89
	Lombok Timur	20.07	5.4	6.94	11.76	4.69	11.63
	Sumbawa	18.25	6.8	12.30	5.82	4.97	2.93
	Dompu	16.57	6.82	11.82	7.03	4.75	3.44
	Bima	16.22	5.9	8.71	5.86	5.08	4.24
	Sumbawa Barat	17.60	6.82	10.11	4.85	5.25	4.6
	Lombok Utara	35.97	4.13	8.41	20.03	3.38	5.43
	Kota Mataram	11.87	3.02	14.63	6.18	6.53	5.1
	Kota Bima	10.54	5.82	8.55	3.74	6.36	16.47
2011	Lombok Barat	19.70	5.58	7.25	19.50	4.89	6.81
	Lombok Tengah	18.14	9.05	6.21	20.10	5.94	12.87
	Lombok Timur	21.71	6.12	6.32	12.85	4.59	9.95
	Sumbawa	19.82	6.9	11.05	6.59	5.17	4.42
	Dompu	18.17	7.98	10.56	8.27	5.87	8.3
	Bima	17.66	5.63	7.80	6.83	5.13	3.9
	Sumbawa Barat	19.88	6.53	9.19	6.70	4.99	6.24
	Lombok Utara	39.27	5.69	7.76	19.28	4.85	6.15
	Kota Mataram	13.18	7.67	13.53	9.92	6.7	8.66
	Kota Bima	11.69	5.38	7.82	4.30	6.36	13.99
2010	Lombok Barat	21.59	4.78	6.58	21.65	5.12	9.43
	Lombok Tengah	19.92	5.69	5.41	21.65	5.69	11.68
	Lombok Timur	23.82	5.01	5.62	16.17	3.93	12.78
	Sumbawa	21.74	5.92	9.54	8.12	5.88	4.05
	Dompu	19.89	4.57	9.06	8.15	5.31	4.59
	Bima	19.41	4.55	7.00	10.38	3.14	6.80
	Sumbawa Barat	21.81	6.73	8.10	7.33	6.54	3.55
	Lombok Utara	43.12	4.04	7.03	23.55	3.29	3.27
	Kota Mataram	14.44	7.95	11.99	9.33	8.96	10.66
	Kota Bima	12.81	5.74	7.02	5.44	5.56	14.38

Lampiran 2. Matriks Korelasi

	Kemiskinan	Perteko	Perkapita	Pendidik	TPT
Perteko	-0.286				
	0.125				
Perkapita	-0.331	0.065			
	0.074	0.733			
Pendidik	0.618	-0.009	-0.553		
	0.000	0.962	0.002		
TPT	-0.600	0.415	0.437	-0.351	
	0.000	0.023	0.016	0.057	
TKI	-0.368	0.258	-0.418	0.128	0.292
	0.045	0.169	0.021	0.500	0.117

Cell Contents: Pearson correlation
P-Value

Lampiran 3. Output Regresi OLS Data Panel

Predictor	Coef	SE Coef	T	P	VIF
Constant	30.976	6.930	4.47	0.000	
Perteko	-0.3563	0.5732	-0.62	0.540	1.258
Perkapita	-0.3077	0.5925	-0.52	0.608	2.504
Pendidik	0.6236	0.1684	3.70	0.001	1.497
TPT	-1.421	1.112	-1.28	0.214	2.110
TKI	-0.7367	0.3030	-2.43	0.023	1.856

S = 4.74273 R-Sq = 66.4% R-Sq(adj) = 59.3%

Analysis of Variance

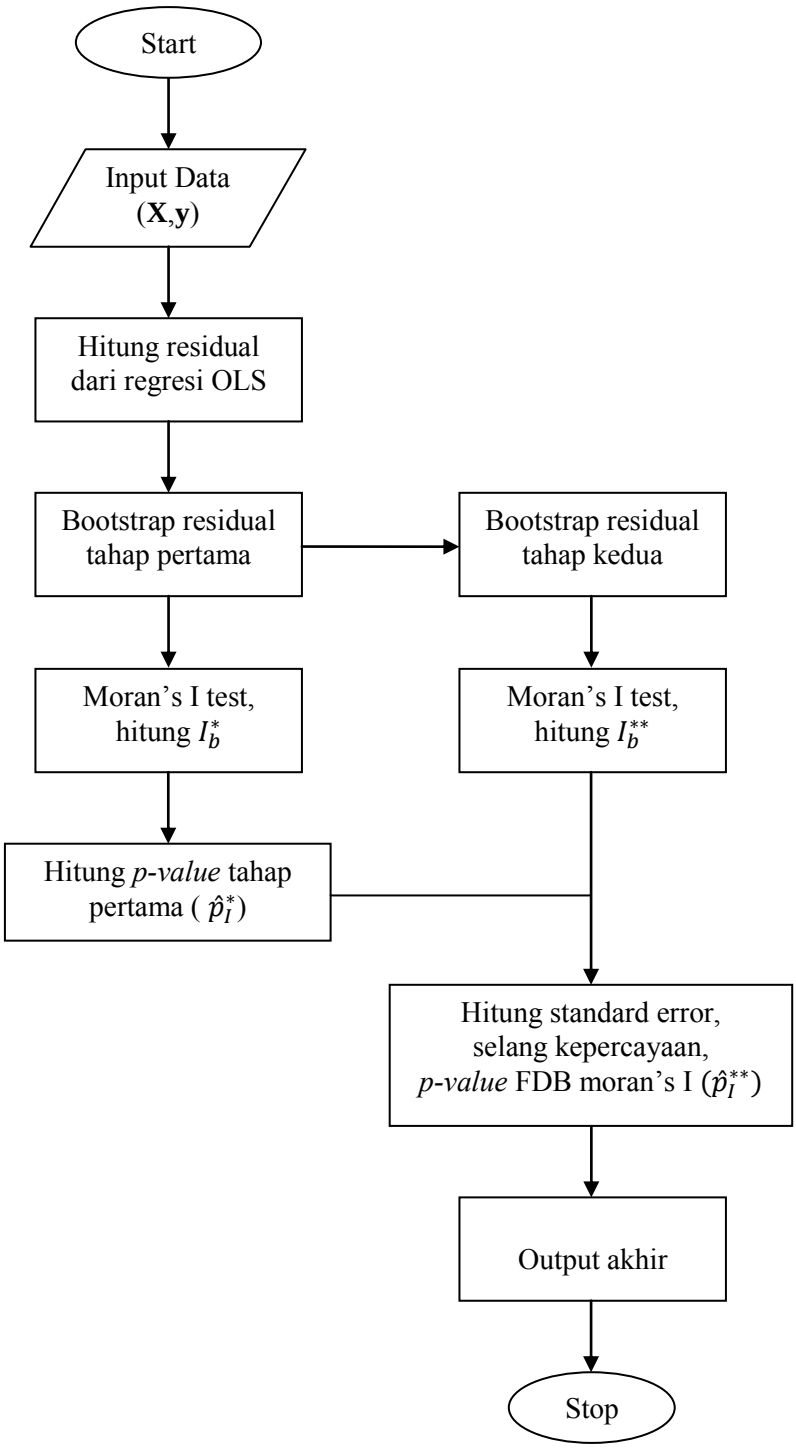
Source	DF	SS	MS	F	P
Regression	5	1064.46	212.89	9.46	0.000
Residual Error	24	539.84	22.49		
Total	29	1604.30			

Source	DF	Seq SS
Perteko	1	131.23
Perkapita	1	157.54
Pendidik	1	451.81
TPT	1	190.95
TKI	1	132.94

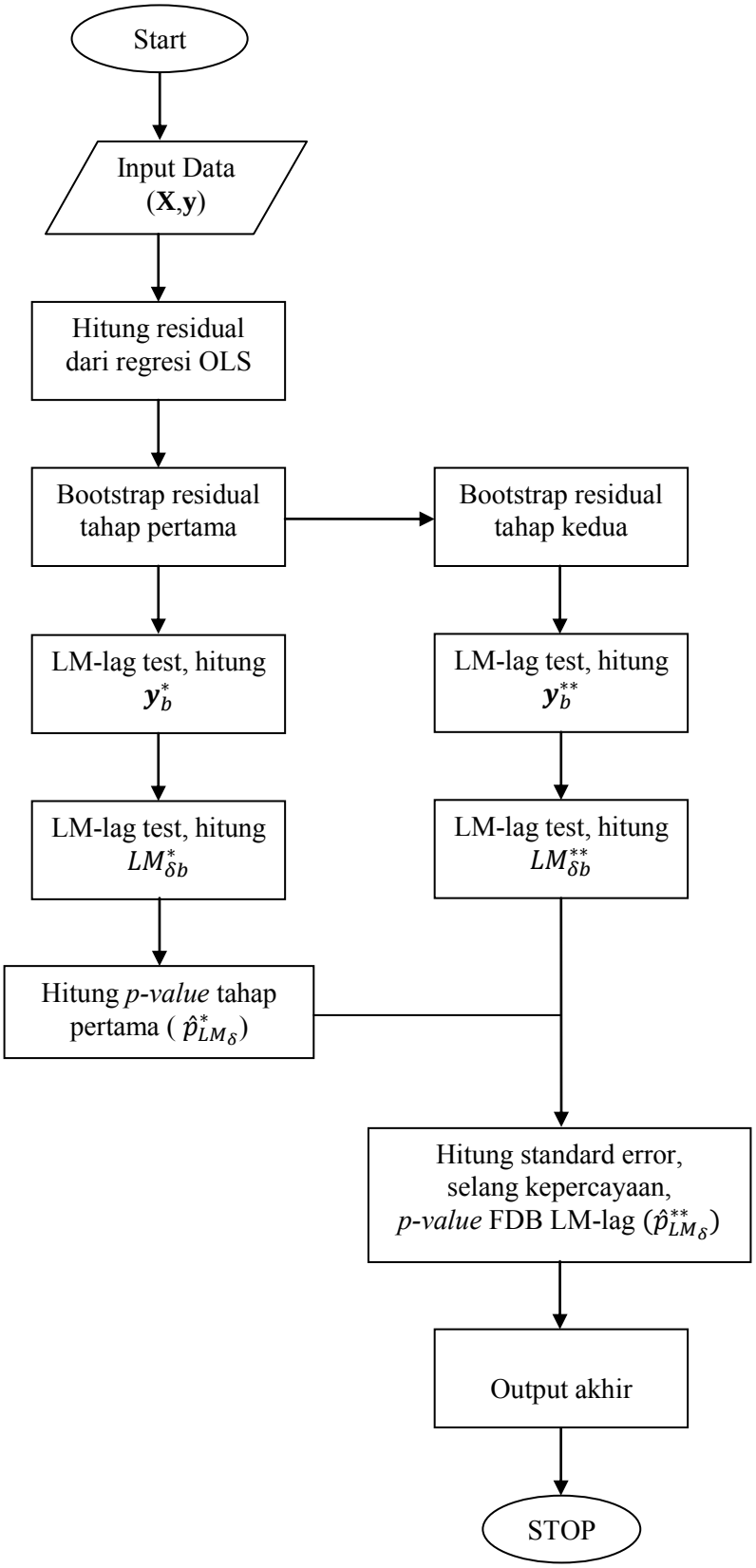
Lampiran 4. Matriks Penimbang Spasial *Queen Contiguity*

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	0.00	0.33	0.00	0.00	0.00	0.00	0.00	0.33	0.33	0.00
2	0.33	0.00	0.33	0.00	0.00	0.00	0.00	0.33	0.00	0.00
3	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00
4	0.00	0.00	0.00	0.00	0.50	0.00	0.50	0.00	0.00	0.00
5	0.00	0.00	0.00	0.50	0.00	0.50	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00	0.50
7	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.33	0.33	0.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00

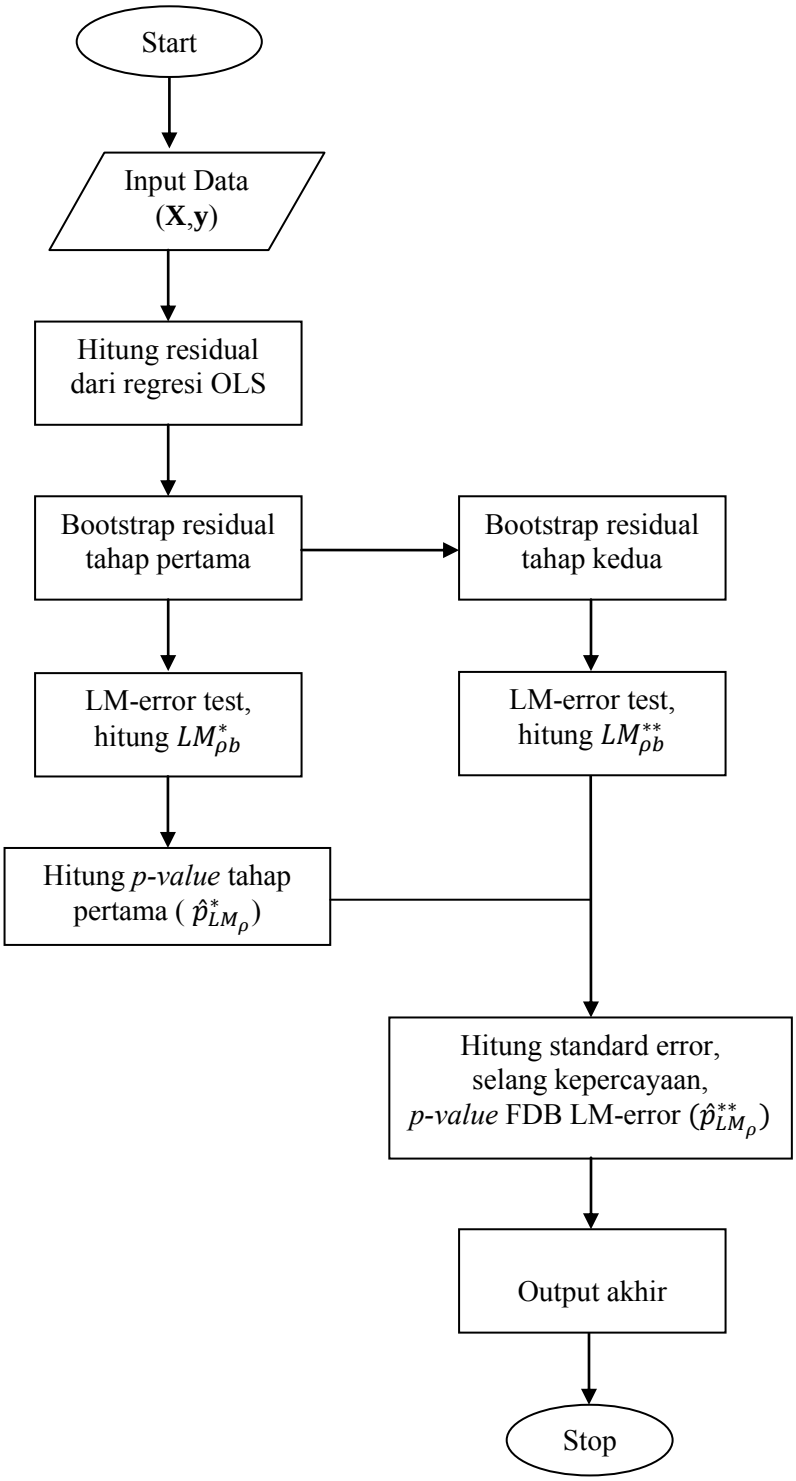
Lampiran 5. *Flowchart* FDB Moran's I



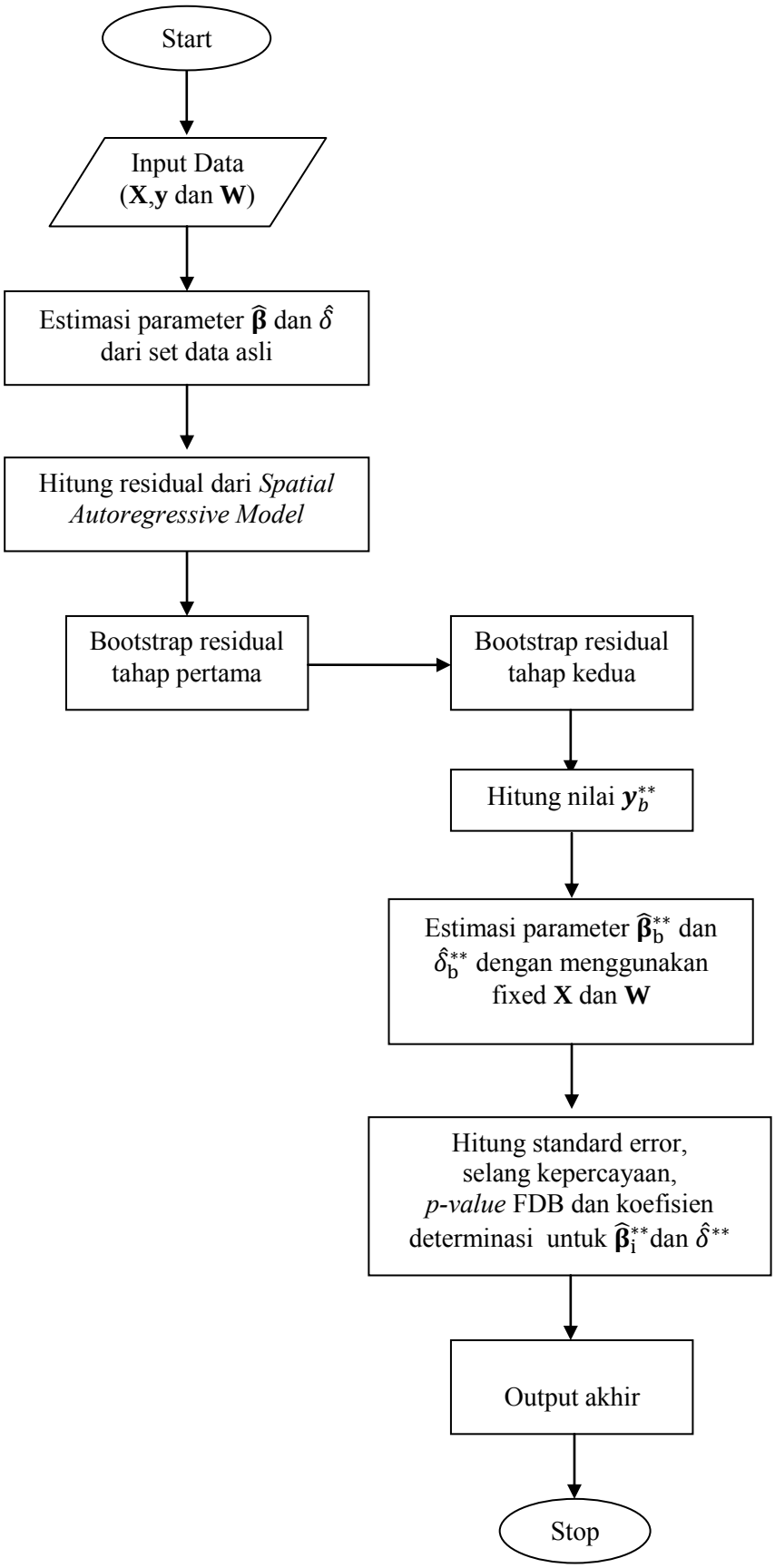
Lampiran 6. *Flowchart* FDB LM lag



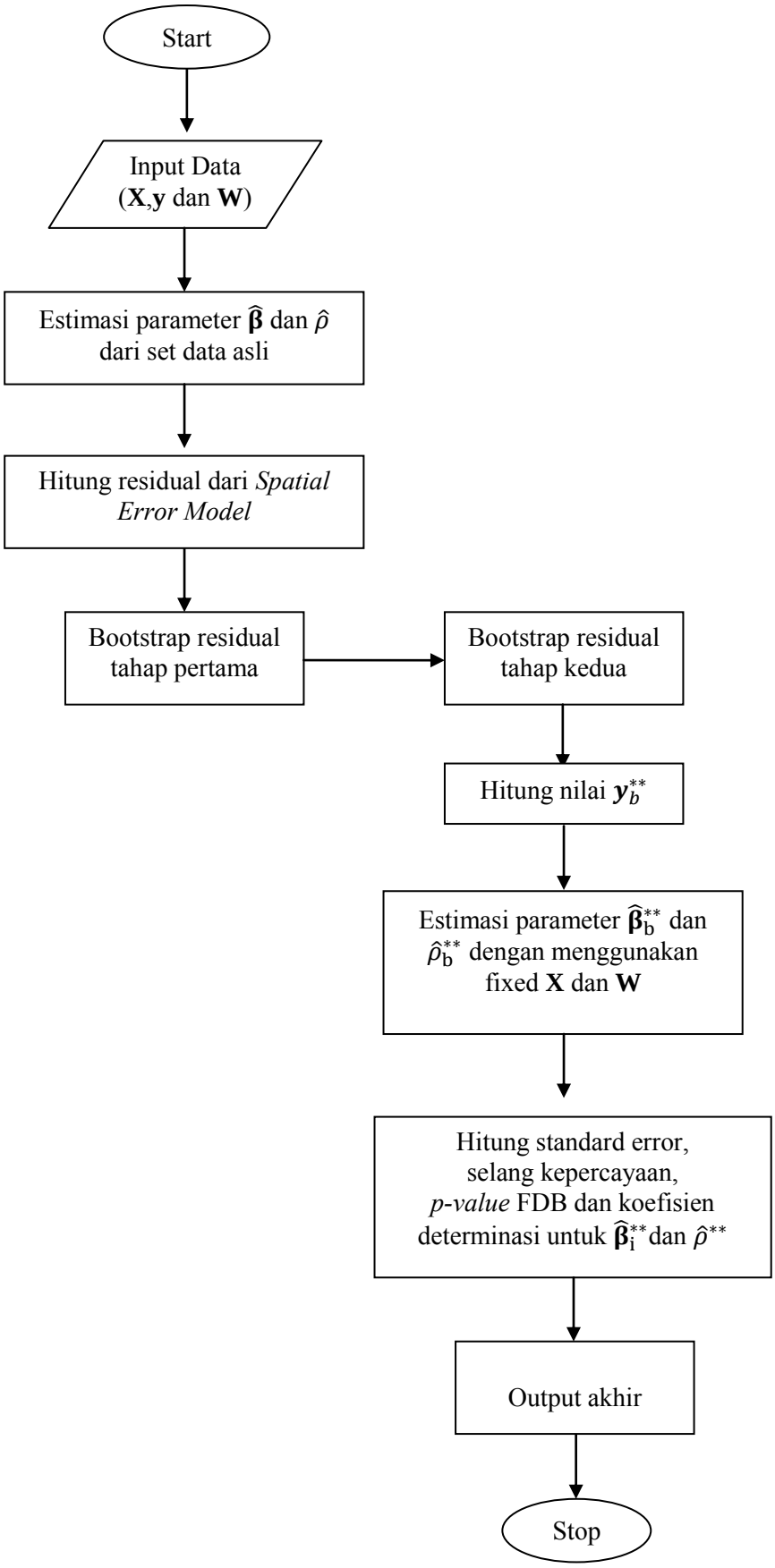
Lampiran 7. *Flowchart* FDB LM error



Lampiran 8. *Flowchart* FDB *Spatial Autoregressive* (SAR)



Lampiran 9. *Flowchart* FDB *Spatial Error Model* (SEM)



Lampiran 10. *Matlab Code* FDB Moran's I

```
function result=moran_fdb(y,x,W,B,alpha);

if nargin~=5
error('input salah');
end

T=3;
N=10;

[nobs K]=size(x);

% Regresi OLS data panel untuk mendapatkan residual
results=ols(y,x);

e=results.resid;
W=normw(W);
It=eye(T,T);
Ww=kron(It,W);

% Moran's I data asli
epe0=e'*e;
ewe0=e'*Ww*e;
mi0=ewe0/epe0;

% Bootstrap tahap pertama
for i=1:B;
    er=randsample(e,nobs,true);
    epe1(i)=er'*er;
    ewe1(i)=er'*Ww*er;
    mir1(i)=ewe1(i)/epe1(i);

if mir1(i)>mi0
p(i)=1;
else p(i)=0;
end
end
npval=sum(p)
pval_boot=npval/B;

q=B*(1-pval_boot);
q=round(q);

% Bootstrap tahap kedua
for i=1:B;
    err=randsample(er,nobs,true);
    epe(i)=err'*err;
    ewe(i)=err'*Ww*err;
    mir(i)=ewe(i)/epe(i);
    mi=sort(mir);
end
for i=1:B
if mi(i)>mi(q)
pp(i)=1;
else pp(i)=0;
end
end

mi_fdb=mean(mi);                            % FDB moran's I
smi_fdb=std(mi);                            % standard error FDB moran's I

mimin=min(mi);
mimax=max(mi);
lower=unifinv(alpha/2,mimin,mimax);         % batas bawah moran's I
upper=unifinv(1.0-(alpha/2),mimin,mimax);   % batas atas moran's I

nppval=sum(pp);
pval_fdb=nppval/B;                          % p-value FDB moran's I
hist(mi);                                   % histogram
```

```
M=eye(nobs)-x*(inv(x'*x))*x';
tmw=trace(M*Ww);

meani=tmw/(nobs-K);
vari0=trace((M*Ww)*(M*Ww'))+trace((M*Ww)*(M*Ww))+tmw*tmw;
vari1=vari0/((nobs-K)*(nobs-K+2));
vari=vari1-(meani*meani);
mis=(mi0-meani)/sqrt(vari);
prob=norm_prb(mis);                          % p-value data asli
bias=mi0-mi_fdb;                             % bias

% Output
result.meth='FDB moran I';
result.nvar=K;
result.nobs=nobs;
result.replikasi=B;
result.moranI=mi0;
result.fdb_moranI=mi_fdb;
result.se_fdb=smi_fdb;
result.pval_mi0=prob;
result.pval_fdb=pval_fdb;
result.lower=lower;
result.upper=upper;
result.bias=bias;
```

Lampiran 11. *Matlab Code* LMsarsem_panel

```
function LMsarsem_panel(results,W,y,x)
% PURPOSE: Computes (robust) LM tests for spatial lag and spatial error
% model of a panel data model
% ---------------------------------------------------------------------------
% Usage: LM=LMsarsem_panel(results,W)
% where: results = a structure returned by a spatial panel regression
%              W = spatial weights matrix (standardized)
%              y = dependent variable vector
%              x = independent variables matrix
% ---------------------------------------------------------------------------
% RETURNS: print of lm tests and probabilities
% ---------------------------------------------------------------------------
% Note: probabilitities smaller than 0.05 point to signifance of spatial
%       lag or spatial error
% ---------------------------------------------------------------------------
% Written by: J.Paul Elhorst summer 2008
% University of Groningen
% Department of Economics
% 9700AV Groningen
% the Netherlands
% j.p.elhorst@rug.nl
%
% REFERENCE:
% Elhorst JP (2009) Spatial Panel Data Models. In Fischer MM, Getis A (Eds.)
% Handbook of Applied Spatial Analysis, Ch. C.2. Springer: Berlin Heidelberg New York.
%
tr=trace((W'+W)*W);
[N junk]=size(W);
[nobs junk]=size(x);
T=nobs/N;
beta=results.beta;
res=results.resid;
sige=res'*res/nobs;
WXB2=0;EWE=0;EWY=0;
for t=1:T
    t1=(t-1)*N+1;t2=t*N;
    WXB=W*x(t1:t2,:)*beta;
    M=eye(N)-x(t1:t2,:)*inv(x(t1:t2,:)'*x(t1:t2,:))*x(t1:t2,:)';
    WXB2=WXB2+WXB'*M*WXB;
    EWE=EWE+res(t1:t2,1)'*W*res(t1:t2,1);
    EWY=EWY+res(t1:t2,1)'*W*y(t1:t2,1);
end
Ttr=T*tr;
J=(WXB2+Ttr*sige)/sige;
LMerror=(EWE/sige)^2/Ttr;
LMlag=(EWY/sige)^2/J;
robustLMerror=((EWE-(Ttr/J)*EWY)/sige)^2/(Ttr*(1-Ttr/J));
robustLMlag=((EWY-EWE)/sige)^2/(J-Ttr);
fprintf(1,'LM test no spatial lag, probability          = %9.4f,%9.3f
\n',LMlag,1-chis_prb(LMlag,1));
fprintf(1,'robust LM test no spatial lag, probability   = %9.4f,%9.3f
\n',robustLMlag,1-chis_prb(robustLMlag,1));
fprintf(1,'LM test no spatial error, probability        = %9.4f,%9.3f
\n',LMerror,1-chis_prb(LMerror,1));
fprintf(1,'robust LM test no spatial error, probability = %9.4f,%9.3f
\n',robustLMerror,1-chis_prb(robustLMerror,1));
```

Lampiran 12. *Matlab Code* FDB LM lag dan FDB LM error

```
function result = LMsarsem_fdb(y,x,W,B)

T=3;
N=10;

W=normw(W);
model=1;
[ywith,xwith,meanny,meannx,meanty,meantx]=demean(y,x,N,T,model);

% Regresi OLS data panel untuk mendapatkan residual
results=ols(ywith,xwith);
b=results.beta;
e=results.resid;

tr=trace((W'+W)*W);
[N junk]=size(W);
[nobs junk]=size(xwith);

% Nilai LM data
sige=e'*e/nobs;
WXB2=0;EWE=0;EWY=0;
for t=1:T
    t1=(t-1)*N+1;t2=t*N;
    WXB=W*xwith(t1:t2,:)*b;
    M=eye(N)-xwith(t1:t2,:)*inv(xwith(t1:t2,:)'*xwith(t1:t2,:))*xwith(t1:t2,:)';
    WXB2=WXB2+WXB'*M*WXB;
    EWE=EWE+e(t1:t2,1)'*W*e(t1:t2,1);
    EWY=EWY+e(t1:t2,1)'*W*ywith(t1:t2,1);
end
Ttr=T*tr;
J=(WXB2+Ttr*sige)/sige;
LMerror=(EWE/sige)^2/Ttr;
LMlag=(EWY/sige)^2/J;
robustLMerror=((EWE-(Ttr/J)*EWY)/sige)^2/(Ttr*(1-Ttr/J));
robustLMlag=((EWY-EWE)/sige)^2/(J-Ttr);

% Nilai LM bootstrap tahap pertama
for i=1:B
    er=randsample(e,nobs,true);
    yr=xwith*b+er;
    sige1=er'*er/nobs;
    WXB2=0;EWE=0;EWY=0;
    for t=1:T
        t1=(t-1)*N+1;t2=t*N;
        WXB=W*xwith(t1:t2,:)*b;

M=eye(N)-xwith(t1:t2,:)*inv(xwith(t1:t2,:)'*xwith(t1:t2,:))*xwith(t1:t2,:)';
        WXB2=WXB2+WXB'*M*WXB;
        EWE=EWE+er(t1:t2,1)'*W*er(t1:t2,1);
        EWY=EWY+er(t1:t2,1)'*W*yr(t1:t2,1);
    end
    Ttr=T*tr;
    J1=(WXB2+Ttr*sige1)/sige1;
    LMerror1(i)=(EWE/sige1)^2/Ttr;
    LMlag1(i)=(EWY/sige1)^2/J1;
    robustLMerror1(i)=((EWE-(Ttr/J1)*EWY)/sige1)^2/(Ttr*(1-Ttr/J1));
    robustLMlag1(i)=((EWY-EWE)/sige1)^2/(J1-Ttr);
end

% p-value LM bootstrap tahap pertama
for i=1:B
if LMerror1(i)>LMerror
pe(i)=1;
else pe(i)=0;
end

if LMlag1(i)>LMlag
pl(i)=1;
else pl(i)=0;
```

```
end

if robustLMerror1(i)>robustLMerror
pre(i)=1;
else pre(i)=0;
end

if robustLMlag1(i)>robustLMlag
prl(i)=1;
else prl(i)=0;
end
end

npvale=sum(pe)
pvale=npvale/B;
npvall=sum(pl)
pvall=npvall/B;
npvalre=sum(pre)
pvalre=npvalre/B;
npvalrl=sum(prl)
pvalrl=npvalrl/B;

% kuintil
q=B*(1-pvale);
q=round(q);
r=B*(1-pvall);
r=round(r);
s=B*(1-pvalre);
s=round(s);
u=B*(1-pvalrl);
u=round(u);

% Nilai LM bootstrap tahap kedua
for i=1:B
    err=randsample(er,nobs,true);
    yrr=xwith*b+err;
    sige2=err'*err/nobs;
    WXB2=0;EWE=0;EWY=0;
    for t=1:T
        t1=(t-1)*N+1;t2=t*N;
        WXB=W*xwith(t1:t2,:)*b;

M=eye(N)-xwith(t1:t2,:)*inv(xwith(t1:t2,:)'*xwith(t1:t2,:))*xwith(t1:t2,:)';
        WXB2=WXB2+WXB'*M*WXB;
        EWE=EWE+err(t1:t2,1)'*W*err(t1:t2,1);
        EWY=EWY+err(t1:t2,1)'*W*yrr(t1:t2,1);
    end
    Ttr=T*tr;
    J2=(WXB2+Ttr*sige2)/sige2;
    LMerror2(i)=(EWE/sige2)^2/Ttr;
    LMlag2(i)=(EWY/sige2)^2/J2;
    robustLMerror2(i)=((EWE-(Ttr/J2)*EWY)/sige2)^2/(Ttr*(1-Ttr/J2));
    robustLMlag2(i)=((EWY-EWE)/sige2)^2/(J2-Ttr);
    lme=sort(LMerror2);
    lml=sort(LMlag2);
    rlme=sort(robustLMerror2);
    rlml=sort(robustLMlag2);
end

% Nilai FDB LM
fdb_lme=mean(lme);
fdb_lml=mean(lml);
fdb_roblme=mean(rlme);
fdb_roblml=mean(rlml);
LM=[fdb_lml fdb_lme fdb_roblml fdb_roblme]';

% standard error FDB LM
slme=std(lme);
slml=std(lml);
srlme=std(rlme);
srlml=std(rlml);
std_err=[slml slme srlml srlme]';
```

```
% p-value FDB LM
for i=1:B
if lme(i)>lme(q)
ppe(i)=1;
else ppe(i)=0;
end;

if lml(i)>lml(r)
ppl(i)=1;
else ppl(i)=0;
end;

if rlme(i)>rlme(s)
ppre(i)=1;
else ppre(i)=0;
end;

if rlml(i)>rlml(u)
pprl(i)=1;
else pprl(i)=0;
end;
end

tstatLM=LM/std_err;

nplme=sum(ppe)
plme=nplme/B;
nplml=sum(ppl)
plml=nplml/B;
nplmre=sum(ppre)
plmre=nplmre/B;
nplmrl=sum(pprl)
plmrl=nplmrl/B;
p_value=[plml plme nplmrl nplmre]';

% Output
result.LM=LM;
result.std_err=std_err;
result.tstatLM=tstatLM;
result.p_value=p_value;
```

Lampiran 13. *Matlab Code* SAR Panel *Spatial Fixed Effect*

```
function results = sar_panel_FE(y,x,W,T,info)
% PURPOSE: computes spatial lag model estimates for spatial panels
%          (N regions*T time periods) with spatial fixed effects (u)
%          and/or time period fixed effects (v)
%       y = p*W*y + X*b + u (optional) + v(optional) + e, using sparse matrix algorithms
% Supply data sorted first by time and then by spatial units, so first region 1,
% region 2, et cetera, in the first year, then region 1, region 2, et
% cetera in the second year, and so on
% sar_panel_FE computes y and x in deviation of the spatial and/or time means
% ---------------------------------------------------
%  USAGE: results = sar_panel_FE(y,x,W,T,info)
%  where:  y = dependent variable vector
%          x = independent variables matrix
%          W = spatial weights matrix (standardized)
%          T = number of points in time
%       info = an (optional) structure variable with input options:
%      info.model = 0 pooled model without fixed effects (default, x may contain an
intercept)
%                  = 1 spatial fixed effects (x may not contain an intercept)
%                  = 2 time period fixed effects (x may not contain an intercept)
%            = 3 spatial and time period fixed effects (x may not contain an intercept)
%      info.fe    = report fixed effects and their t-values in prt_sp (default=0=not
reported; info.fe=1=report)
%      info.Nhes  = N =< Nhes asymptotic variance matrix is computed using analytical
formulas,
%                   N > Nhes asymptotic variance matrix is computed using numerical
formulas
%                     (Default NHes=500)
%       info.rmin  = (optional) minimum value of rho to use in search
%       info.rmax  = (optional) maximum value of rho to use in search
%       info.convg = (optional) convergence criterion (default = 1e-8)
%       info.maxit = (optional) maximum # of iterations (default = 500)
%       info.lflag = 0 for full lndet computation (default = 1, fastest)
%                  = 1 for MC lndet approximation (fast for very large problems)
%                  = 2 for Spline lndet approximation (medium speed)
%       info.order = order to use with info.lflag = 1 option (default = 50)
%       info.iter  = iterations to use with info.lflag = 1 option (default = 30)
%      info.lndet = a matrix returned by sar containing log-determinant information
to save time
% ---------------------------------------------------
%  RETURNS: a structure
%         results.meth  = 'psar' if infomodel=0
%                       = 'sarsfe' if info.model=1
%                       = 'sartfe' if info.model=2
%                       = 'sarstfe' if info.model=3
%         results.beta  = bhat
%         results.rho   = rho (p above)
%      results.cov   = asymptotic variance-covariance matrix of the parameters b(eta)
and rho
%         results.tstat = asymp t-stat (last entry is rho=spatial autoregressive
coefficient)
%         results.yhat  = [inv(y-p*W)]*[x*b+fixed effects] (according to prediction
formula)
%         results.resid = y-p*W*y-x*b
%         results.sige  = (y-p*W*y-x*b)'*(y-p*W*y-x*b)/n
%         results.rsqr  = rsquared
%         results.corr2 = goodness-of-fit between actual and fitted values
%         results.sfe   = spatial fixed effects (if info.model=1 or 3)
%         results.tfe   = time period fixed effects (if info.model=2 or 3)
%         results.tsfe  = t-values spatial fixed effects (if info.model=1 or 3)
%         results.ttfe  = t-values time period fixed effects (if info.model=2 or 3)
%         results.con   = intercept
%         results.con   = t-value intercept
%         results.lik   = log likelihood
%         results.nobs  = # of observations
%         results.nvar  = # of explanatory variables in x
%         results.tnvar = nvar + W*y + # fixed effects
%         results.iter  = # of iterations taken
%         results.rmax  = 1/max eigenvalue of W (or rmax if input)
%         results.rmin  = 1/min eigenvalue of W (or rmin if input)
```

```
%         results.lflag = lflag from input
%         results.fe    = fe from input
%         results.liter = info.iter option from input
%         results.order = info.order option from input
%         results.limit = matrix of [rho lower95,logdet approx, upper95] intervals
%                         for the case of lflag = 1
%         results.time1 = time for log determinant calcluation
%         results.time2 = time for eigenvalue calculation
%         results.time3 = time for hessian or information matrix calculation
%         results.time4 = time for optimization
%         results.time  = total time taken
%         results.lndet = a matrix containing log-determinant information
%                           (for use in later function calls to save time)
% ---------------------------------------------------
%  NOTES: if you use lflag = 1 or 2, info.rmin will be set = -1
%                                    info.rmax will be set = 1
%         For number of spatial units < 500 you should use lflag = 0 to get
%         exact results,
%         Fixed effects and their t-values are calculated as the deviation
%         from the mean intercept
% ----------------------------------------------------
%
% Updated by: J.Paul Elhorst summer 2008
% University of Groningen
% Department of Economics
% 9700AV Groningen
% the Netherlands
% j.p.elhorst@rug.nl
%
% REFERENCES:
% Elhorst JP (2003) Specification and Estimation of Spatial Panel Data Models,
% International Regional Science Review 26: 244-268.
% Elhorst JP (2009) Spatial Panel Data Models. In Fischer MM, Getis A (Eds.)
% Handbook of Applied Spatial Analysis, Ch. C.2. Springer: Berlin Heidelberg New York.

% This function is partly based on James. P LeSage's function SAR

time1 = 0;
time2 = 0;
time3 = 0;
time4 = 0;

timet = clock; % start the clock for overall timing

W=sparse(W);

% if we have no options, invoke defaults
if nargin == 4
    info.lflag = 1;
    info.model=0;
    info.Nhes=500;
    fprintf(1,'default: pooled model without fixed effects \n');
end;

fe=0;
model=0;
Nhes=500;

fields = fieldnames(info);
nf = length(fields);
if nf > 0
    for i=1:nf
        if strcmp(fields{i},'model') model = info.model;
        elseif strcmp(fields{i},'fe') fe = info.fe;
        elseif strcmp(fields{i},'Nhes') Nhes = info.Nhes;
        end
    end
end
if model==0
    results.meth='psar';
elseif model==1
    results.meth='sarsfe';
```

```
elseif model==2
    results.meth='sartfe';
elseif model==3
    results.meth='sarstfe';
else
    error('sar_panel: wrong input number of info.model');
end

% check size of user inputs for comformability
[nobs nvar] = size(x);
[N Ncol] = size(W);
if N ~= Ncol
error('sar: wrong size weight matrix W');
elseif N ~= nobs/T
error('sar: wrong size weight matrix W or matrix x');
end;
[nchk junk] = size(y);
if nchk ~= nobs
error('sar: wrong size vector y or matrix x');
end;

if (fe==1 & model==0 ) error('info.fe=1, but cannot compute fixed effects if info.model
is set to 0 or not specified'); end

% parse input options
%[rmin,rmax,convg,maxit,detval,ldetflag,eflag,order,miter,options] =
sar_parse(info); % function of LeSage
[rmin,rmax,convg,maxit,detval,ldetflag,eflag,order,miter,options,ndraw,sflag,p,cfl
ag] = sar_parse(info);

% compute eigenvalues or limits
[rmin,rmax,time2] = sar_eigs(eflag,W,rmin,rmax,N); % function of LeSage

% do log-det calculations
[detval,time1] = sar_lndet(ldetflag,W,rmin,rmax,detval,order,miter); % function of
LeSage

for t=1:T
    t1=1+(t-1)*N;t2=t*N;
    Wy(t1:t2,1)=W*y(t1:t2,1);
end

% demeaning of the y and x variables, depending on (info.)model

if (model==1 | model==3);
meanny=zeros(N,1);
meannwy=zeros(N,1);
meannx=zeros(N,nvar);
for i=1:N
    ym=zeros(T,1);
    wym=zeros(T,1);
    xm=zeros(T,nvar);
    for t=1:T
        ym(t)=y(i+(t-1)*N,1);
        wym(t)=Wy(i+(t-1)*N,1);
        xm(t,:)=x(i+(t-1)*N,:);
    end
    meanny(i)=mean(ym);
    meannwy(i)=mean(wym);
    meannx(i,:)=mean(xm);
end
clear ym wym xm;
end % if statement

if ( model==2 | model==3)
meanty=zeros(T,1);
meantwy=zeros(T,1);
meantx=zeros(T,nvar);
for i=1:T
    t1=1+(i-1)*N;t2=i*N;
    ym=y([t1:t2],1);
    wym=Wy([t1:t2],1);
```

```
    xm=x([t1:t2],:);
    meanty(i)=mean(ym);
    meantwy(i)=mean(wym);
    meantx(i,:)=mean(xm);
end
clear ym wym xm;
end % if statement

en=ones(T,1);
et=ones(N,1);
ent=ones(nobs,1);

if model==1
    ywith=y-kron(en,meanny);
    wywith=Wy-kron(en,meannwy);
    xwith=x-kron(en,meannx);
elseif model==2
    ywith=y-kron(meanty,et);
    wywith=Wy-kron(meantwy,et);
    xwith=x-kron(meantx,et);
elseif model==3
    ywith=y-kron(en,meanny)-kron(meanty,et)+kron(ent,mean(y));
    wywith=Wy-kron(en,meannwy)-kron(meantwy,et)+kron(ent,mean(Wy));
    xwith=x-kron(en,meannx)-kron(meantx,et)+kron(ent,mean(x));
else
    ywith=y;
    wywith=Wy;
    xwith=x;
end % if statement

% step 1) do regressions
t0 = clock;
          AI = xwith'*xwith;
          b0 = AI\(xwith'*ywith);
          bd = AI\(xwith'*wywith);
          e0 = ywith - xwith*b0;
          ed = wywith - xwith*bd;
          epe0 = e0'*e0;
          eped = ed'*ed;
          epe0d = ed'*e0;

% step 2) maximize concentrated likelihood function;
    options = optimset('fminbnd');
    [p,liktmp,exitflag,output] =
fminbnd('f_sarpanel',rmin,rmax,options,detval,epe0,eped,epe0d,N,T);

time4 = etime(clock,t0);

if exitflag == 0
fprintf(1,'sar: convergence concentrated likelihood function not obtained in %4d
iterations \n',output.iterations);
end;
results.iter = 1;

% step 3) find b,sige maximum likelihood estimates
results.beta = b0 - p*bd;
results.rho = p;
bhat = results.beta;
results.sige = (1/nobs)*(e0-p*ed)'*(e0-p*ed);
sige = results.sige;

% step 4) find fixed effects and their t-values
if model==1
    intercept=mean(y)-mean(Wy)*results.rho-mean(x)*results.beta;
    results.con=intercept;
    results.sfe=meanny-meannwy*results.rho-meannx*results.beta-kron(et,intercept);
    xhat=x*results.beta+kron(en,results.sfe)+kron(ent,intercept);

results.tsfe=results.sfe./sqrt(sige/T*ones(N,1)+diag(sige*meannx*(xwith'*xwith)*me
annx'));
    results.tcon=results.con/sqrt(sige/nobs+sige*mean(x)*(xwith'*xwith)*mean(x)');
    tnvar=nvar+N;
```

```
elseif model==2
    intercept=mean(y)-mean(Wy)*results.rho-mean(x)*results.beta;
    results.con=intercept;
    results.tfe=meanty-meantwy*results.rho-meantx*results.beta-kron(en,intercept);
    xhat=x*results.beta+kron(results.tfe,et)+kron(ent,intercept);

results.ttfe=results.tfe./sqrt(sige/N*ones(T,1)+diag(sige*meantx*(xwith'*xwith)*me
antx'));
    results.tcon=results.con/sqrt(sige/nobs+sige*mean(x)*(xwith'*xwith)*mean(x)');
    tnvar=nvar+T;
elseif model==3
    intercept=mean(y)-mean(Wy)*results.rho-mean(x)*results.beta;
    results.con=intercept;
    results.sfe=meanny-meannwy*results.rho-meannx*results.beta-kron(et,intercept);
    results.tfe=meanty-meantwy*results.rho-meantx*results.beta-kron(en,intercept);

results.tsfe=results.sfe./sqrt(sige/T*ones(N,1)+diag(sige*meannx*(xwith'*xwith)*me
annx'));

results.ttfe=results.tfe./sqrt(sige/N*ones(T,1)+diag(sige*meantx*(xwith'*xwith)*me
antx'));
    results.tcon=results.con/sqrt(sige/nobs+sige*mean(x)*(xwith'*xwith)*mean(x)');

xhat=x*results.beta+kron(en,results.sfe)+kron(results.tfe,et)+kron(ent,intercept);
    tnvar=nvar+T;
else
    xhat=x*results.beta;
    tnvar=nvar;
end

% r-squared and corr-squared between actual and fitted values
results.tnvar=tnvar;
results.resid = y - p*Wy - xhat;
yme=y-mean(y);
rsqr2=yme'*yme;
rsqr1 = results.resid'*results.resid;
results.rsqr=1.0-rsqr1/rsqr2; %rsquared

yhat=zeros(nobs,1);
ywithhat=zeros(nobs,1);
for t=1:T
    t1=1+(t-1)*N;t2=t*N;
    ywithhat(t1:t2,1)=(speye(N) - p*W)\xwith(t1:t2,:)*results.beta;
    yhat(t1:t2,1)=(speye(N) - p*W)\xhat(t1:t2,1);
end
res1=ywith-mean(ywith);
res2=ywithhat-mean(ywith);
rsq1=res1'*res2;
rsq2=res1'*res1;
rsq3=res2'*res2;
results.corr2=rsq1^2/(rsq2*rsq3); %corr2
results.yhat=yhat;

parm = [results.beta
        results.rho
        results.sige];

results.lik = f2_sarpanel(parm,ywith,xwith,W,detval,T); %Elhorst

% Determination variance-covariance matrix
if N <= Nhes % Analytically
t0 = clock;
B = speye(N) - p*W;
BI = inv(B); WB = W*BI;
pterm = trace(WB*WB + WB'*WB);
xpx = zeros(nvar+2,nvar+2);
% bhat,bhat
xpx(1:nvar,1:nvar) = (1/sige)*(xwith'*xwith);
% bhat,rho
ysum=zeros(nvar,1);
for t=1:T
    t1=1+(t-1)*N;t2=t*N;
```

```
    ysum=ysum+(1/sige)*xwith(t1:t2,:)'*WB*xwith(t1:t2,:)*bhat;
end
xpx(1:nvar,nvar+1) = ysum;
xpx(nvar+1,1:nvar) = xpx(1:nvar,nvar+1)';
% rho,rho
ysom=0;
for t=1:T
    t1=1+(t-1)*N;t2=t*N;
    ysom=ysom+(1/sige)*bhat'*xwith(t1:t2,:)'*WB'*WB*xwith(t1:t2,:)*bhat + pterm;
end
xpx(nvar+1,nvar+1) = ysom;
% sige, sige
xpx(nvar+2,nvar+2) = nobs/(2*sige*sige);
% rho,sige
xpx(nvar+1,nvar+2) = (T/sige)*trace(WB);
xpx(nvar+2,nvar+1) = xpx(nvar+1,nvar+2);
xpxi = xpx\eye(size(xpx));
results.cov=xpxi(1:nvar+1,1:nvar+1);
tmp = diag(xpxi(1:nvar+1,1:nvar+1));
bvec = [results.beta
        results.rho];
tmp = bvec./(sqrt(tmp));
results.tstat = tmp;
time3 = etime(clock,t0);

else  % asymptotic t-stats using numerical hessian
t0 = clock;
dhessn = hessian('f2_sarpanel',parm,ywith,xwith,W,detval,T); %Elhorst
hessi = invpd(-dhessn);
results.cov=hessi(1:nvar+1,1:nvar+1);
tvar = abs(diag(hessi));
tmp = [results.beta
       results.rho];
results.tstat = tmp./sqrt(tvar(1:end-1,1));
time3 = etime(clock,t0);

end; % end of t-stat calculations

% return stuff
results.nobs  = nobs;
results.nvar  = nvar;
results.rmax  = rmax;
results.rmin  = rmin;
results.lflag = ldetflag;
results.order = order;
results.miter = miter;
results.fe    = fe;
results.time  = etime(clock,timet);
results.time1 = time1;
results.time2 = time2;
results.time3 = time3;
results.time4 = time4;
results.lndet = detval;
results.N     = N;
results.T     = T;
results.model = model;


function
[rmin,rmax,convg,maxit,detval,ldetflag,eflag,order,iter,options,ndraw,sflag,p,cfla
g] = sar_parse(info)
% PURPOSE: parses input arguments for sar model
% ---------------------------------------------------
%  USAGE: [rmin,rmax,convg,maxit,detval,ldetflag,eflag,order,iter,options] =
sar_parse(info)
% where info contains the structure variable with inputs
% and the outputs are either user-inputs or default values
% ---------------------------------------------------

% set defaults
options = zeros(1,18); % optimization options for fminbnd
options(1) = 0;
```

```
options(2) = 1.e-6;
options(14) = 500;

eflag = 0;      % default to not computing eigenvalues
ldetflag = 1;   % default to 1999 Pace and Barry MC determinant approx
order = 50;     % there are parameters used by the MC det approx
iter = 30;      % defaults based on Pace and Barry recommendation
rmin = -1;      % use -1,1 rho interval as default
rmax = 1;
detval = 0;     % just a flag
convg = 0.0001;
maxit = 500;
ndraw = 1000;
sflag = 0;
p = 0;
cflag = 0;

fields = fieldnames(info);
nf = length(fields);
if nf > 0

 for i=1:nf
    if strcmp(fields{i},'rmin')
        rmin = info.rmin;  eflag = 0;
    elseif strcmp(fields{i},'rmax')
        rmax = info.rmax; eflag = 0;
    elseif strcmp(fields{i},'p')
        p = info.p;
    elseif strcmp(fields{i},'cflag')
        cflag = info.cflag;
    elseif strcmp(fields{i},'convg')
        options(2) = info.convg;
    elseif strcmp(fields{i},'maxit')
        options(14) = info.maxit;
    elseif strcmp(fields{i},'lndet')
    detval = info.lndet;
    ldetflag = -1;
    eflag = 0;
    rmin = detval(1,1);
    nr = length(detval);
    rmax = detval(nr,1);
    elseif strcmp(fields{i},'lflag')
        tst = info.lflag;
        if tst == 0,
        ldetflag = 0; % compute full lndet, no approximation
        elseif tst == 1,
        ldetflag = 1; % use Pace-Barry approximation
        elseif tst == 2,
        ldetflag = 2; % use spline interpolation approximation
        else
        error('sar: unrecognizable lflag value on input');
        end;
    elseif strcmp(fields{i},'order')
        order = info.order;
    elseif strcmp(fields{i},'eig')
        eflag = info.eig;
    elseif strcmp(fields{i},'iter')
        iter = info.iter;
     elseif strcmp(fields{i},'ndraw')
        ndraw = info.ndraw;
     elseif strcmp(fields{i},'sflag')
        sflag = info.sflag;
    end;
 end;

else, % the user has input a blank info structure
      % so we use the defaults
end;

function [rmin,rmax,time2] = sar_eigs(eflag,W,rmin,rmax,n);
% PURPOSE: compute the eigenvalues for the weight matrix
% ---------------------------------------------------
```

```
%  USAGE: [rmin,rmax,time2] = far_eigs(eflag,W,rmin,rmax,W)
% where eflag is an input flag, W is the weight matrix
%       rmin,rmax may be used as default outputs
% and the outputs are either user-inputs or default values
% ---------------------------------------------------


if eflag == 1 % do eigenvalue calculations
t0 = clock;
opt.tol = 1e-3; opt.disp = 0;
lambda = eigs(sparse(W),speye(n),1,'SR',opt);
rmin = real(1/lambda);
rmax = 1.0;
time2 = etime(clock,t0);
else % use rmin,rmax arguments from input or defaults -1,1
time2 = 0;
end;


function [detval,time1] = sar_lndet(ldetflag,W,rmin,rmax,detval,order,iter);
% PURPOSE: compute the log determinant |I_n - rho*W|
% using the user-selected (or default) method
% ---------------------------------------------------
%  USAGE: detval = far_lndet(lflag,W,rmin,rmax)
% where eflag,rmin,rmax,W contains input flags
% and the outputs are either user-inputs or default values
% ---------------------------------------------------


% do lndet approximation calculations if needed
if ldetflag == 0 % no approximation
t0 = clock;
out = lndetfull(W,rmin,rmax);
time1 = etime(clock,t0);
tt=rmin:.001:rmax; % interpolate a finer grid
outi = interp1(out.rho,out.lndet,tt','spline');
detval = [tt' outi];

elseif ldetflag == 1 % use Pace and Barry, 1999 MC approximation

t0 = clock;
out = lndetmc(order,iter,W,rmin,rmax);
time1 = etime(clock,t0);
results.limit = [out.rho out.lo95 out.lndet out.up95];
tt=rmin:.001:rmax; % interpolate a finer grid
outi = interp1(out.rho,out.lndet,tt','spline');
detval = [tt' outi];

elseif ldetflag == 2 % use Pace and Barry, 1998 spline interpolation

t0 = clock;
out = lndetint(W,rmin,rmax);
time1 = etime(clock,t0);
tt=rmin:.001:rmax; % interpolate a finer grid
outi = interp1(out.rho,out.lndet,tt','spline');
detval = [tt' outi];

elseif ldetflag == -1 % the user fed down a detval matrix
    time1 = 0;
        % check to see if this is right
        if detval == 0
            error('sar: wrong lndet input argument');
        end;
        [n1,n2] = size(detval);
        if n2 ~= 2
            error('sar: wrong sized lndet input argument');
        elseif n1 == 1
            error('sar: wrong sized lndet input argument');
        end;
end;
```

```
function H = hessian(f,x,varargin)
% PURPOSE: Computes finite difference Hessian
% -------------------------------------------------------
% Usage:  H = hessian(func,x,varargin)
% Where: func = function name, fval = func(x,varargin)
%           x = vector of parameters (n x 1)
%    varargin = optional arguments passed to the function
% -------------------------------------------------------
% RETURNS:
%           H = finite differnce hessian
% -------------------------------------------------------

% Code from:
% COMPECON toolbox [www4.ncsu.edu/~pfackler]
% documentation modified to fit the format of the Ecoometrics Toolbox
% by James P. LeSage, Dept of Economics
% University of Toledo
% 2801 W. Bancroft St,
% Toledo, OH 43606
% jlesage@spatial-econometrics.com

eps = 1e-6;

n = size(x,1);
fx = feval(f,x,varargin{:});

% Compute the stepsize (h)
h = eps.^(1/3)*max(abs(x),1e-2);
xh = x+h;
h = xh-x;
ee = sparse(1:n,1:n,h,n,n);

% Compute forward step
g = zeros(n,1);
for i=1:n
  g(i) = feval(f,x+ee(:,i),varargin{:});
end

H=h*h';
% Compute "double" forward step
for i=1:n
for j=i:n
  H(i,j) = (feval(f,x+ee(:,i)+ee(:,j),varargin{:})-g(i)-g(j)+fx)/H(i,j);
  H(j,i) = H(i,j);
end
end
```

Lampiran 14. *Matlab Code* FDB SAR Panel *Spatial Fixed Effect*

```
function result = sar_fdb(y,x,W,B,alpha)

% FDB SAR Panel Spatial Fixed Effect
% y = variabel dependen
% x = variabel independen
% W = matriks penimbang spasial
% B = banyaknya replikasi
% ----------------------------------------------------

format short

model=1

[nobs K]=size(x);
T=3;
N=10


% Model SAR Panel untuk mendapatkan residual
info.model=1;
results=sar_panel_FE(y,x,W,T,info);

u=results.resid;
r=results.rho;
b=results.beta;
c=results.con;
FE=results.sfe;

It=eye(T,T);
I=eye(nobs,nobs);
Ww=kron(It,W);
Wi=inv([I-r*Ww]);
XB=x*b;
ent=ones(nobs,1);
cent=kron(ent,c);
FE1=ones(T,1);
FE2=kron(FE1,FE);

Bo=zeros(B,K);                      % hasil estimasi koefisien beta
Ro=zeros(B,1);                      % hasil estimasi koefisien rho
Co=zeros(B,1);                      % hasil estimasi intersep

% Resampling
for i=1:B;
    ur=randsample(u,nobs,true);
    urr=randsample(ur,nobs,true);
    yr=Wi*cent+Wi*XB+Wi*urr+Wi*FE2;
    bres=sar_panel_FE(yr,x,W,T,info);
    br=bres.beta;                   % estimasi beta setiap replikasi
    rr=bres.rho;                    % estimasi rho setiap replikasi
    cr=bres.con;                    % estimasi intersep setiap replikasi
    Bo(i,:)=br';
    Ro(i,:)=rr;
    Co(i,:)=cr;
end;
%koefisien
Br=mean(Bo);                        % beta
Rr=mean(Ro);                        % rho
Cr=mean(Co);                        % intersep

%standar error
Bse=std(Bo);
Rse=std(Ro);
Cse=std(Co);

coeff=[Rr Cr Br]';
rse=[Rse Cse Bse]';

%p-value
```

```
zval_rho=Rr/Rse;
zval_const=Cr/Cse;
for l=1:K
zval_beta(l)=Br(l)/Bse(l);
end
zvalue=[zval_rho zval_const zval_beta]';
prob=norm_prb(zvalue);

%selang kepercayaan
stat_beta=sort(Bo);
stat_rho=sort(Ro);
stat_con=sort(Co);

statmin_beta=min(stat_beta);
statmax_beta=max(stat_beta);
lower_beta=unifinv(alpha/2,statmin_beta,statmax_beta);
upper_beta=unifinv(1-(alpha/2),statmin_beta,statmax_beta);

statmin_rho=min(stat_rho);
statmax_rho=max(stat_rho);
lower_rho=unifinv(alpha/2,statmin_rho,statmax_rho);
upper_rho=unifinv(1-(alpha/2),statmin_rho,statmax_rho);

statmin_con=min(stat_con);
statmax_con=max(stat_con);
lower_con=unifinv(alpha/2,statmin_con,statmax_con);
upper_con=unifinv(1-(alpha/2),statmin_con,statmax_con);

lower=[lower_rho lower_con lower_beta]';
upper=[upper_rho upper_con upper_beta]';

%bias
bias.rho=Rr-r;
bias.con=Cr-c;
bias.beta=Br'-b;
bias=[bias.rho bias.con bias.beta']';

for t=1:T
    t1=1+(t-1)*N;t2=t*N;
    Wy(t1:t2,1)=W*y(t1:t2,1);
end

%R-square dan corr2
ent=ones(nobs,1);
ymy=y-mean(y);
xhat=x*Br'+FE2+kron(ent,Cr);
residual=y-Rr*Wy-xhat;
R1=residual'*residual;
R2=ymy'*ymy;
Rsquare=1.0-(R1/R2);
tsfe=results.tsfe;
psfe=norm_prb(tsfe);
corr2=results.corr2;

%Histogram
m=K+2;
for l=1:K
subplot(2,4,l)
hist(Bo(:,l));
title('koefisien beta')
end

subplot(2,4,K+1)
hist(Ro);
title('koefisien rho');

% Output
result.meth='FDB SAR_FE';
result.nvar=K;
result.nobs=nobs;
result.replikasi=B;
result.coeff=coeff;
```

```
result.rse=rse;
result.prob=prob;
result.lower=lower;
result.upper=upper;
result.bias=bias;
result.sfe=FE2;
result.tsfe=tsfe;
result.psfe=psfe;
result.residual=residual;
result.Rsquare=Rsquare;
result.corr2=corr2;
```

Lampiran 15. *Matlab Code* SEM Panel *Spatial Fixed Effect*

```
function results = sem_panel_FE(y,x,W,T,info)
% PURPOSE: computes spatial error model estimates for spatial panels
%          (N regions*T time periods) with spatial fixed effects (u) and/or
%          time period fixed effects (v)
%       y = XB + u (optional) + v (optional) + s,  s = p*W*s + e, using sparse algorithms
% Supply data sorted first by time and then by spatial units, so first region 1,
% region 2, et cetera, in the first year, then region 1, region 2, et
% cetera in the second year, and so on
% sem_panel_FE computes y and x in deviation of the spatial and/or time means
% ---------------------------------------------------
%  USAGE: results = sem_panel_FE(y,x,W,T,info)
%  where: y = dependent variable vector
%         x = independent variables matrix
%         W = spatial weights matrix (standardized)
%         T = number of points in time
%       info = an (optional) structure variable with input options:
%      info.model = 0 pooled model without fixed effects (default, x may contain an
intercept)
%                  = 1 spatial fixed effects (x may not contain an intercept)
%                  = 2 time period fixed effects (x may not contain an intercept)
%            = 3 spatial and time period fixed effects (x may not contain an intercept)
%      info.fe    = report fixed effects and their t-values in prt_sp (default=0=not
reported; info.fe=1=report)
%       info.rmin  = (optional) minimum value of rho to use in search
%       info.rmax  = (optional) maximum value of rho to use in search
%       info.convg = (optional) convergence criterion (default = 1e-4)
%       info.maxit = (optional) maximum # of iterations (default = 500)
%       info.lflag = 0 for full lndet computation (default = 1, fastest)
%                  = 1 for MC lndet approximation (fast for very large problems)
%                  = 2 for Spline lndet approximation (medium speed)
%       info.order = order to use with info.lflag = 1 option (default = 50)
%       info.iter  = iterations to use with info.lflag = 1 option (default = 30)
%      info.lndet = a matrix returned by sem containing log-determinant information
to save time
% ---------------------------------------------------
%  RETURNS: a structure
%         results.meth  = 'psem' if infomodel=0
%                       = 'semsfe' if info.model=1
%                       = 'semtfe' if info.model=2
%                       = 'semstfe' if info.model=3
%         results.beta  = bhat
%         results.rho   = rho (p above)
%      results.cov   = asymptotic variance-covariance matrix of the parameters b(eta)
and rho
%         results.tstat = asymp t-stats (last entry is rho=spatial autocorrelation
coefficient)
%         results.yhat  = x*b+fixed effects (according to prediction formula)
%         results.resid = y-x*b
%         results.sige  = e'(I-p*W)'*(I-p*W)*e/nobs
%         results.rsqr  = rsquared
%         results.corr2 = goodness-of-fit between actual and fitted values
%         results.sfe   = spatial fixed effects (if info.model=1 or 3)
%         results.tfe   = time period fixed effects (if info.model=2 or 3)
%         results.tsfe  = t-values spatial fixed effects (if info.model=1 or 3)
%         results.ttfe  = t-values time period fixed effects (if info.model=2 or 3)
%         results.con   = intercept
%         results.con   = t-value intercept
%         results.lik   = log likelihood
%         results.nobs  = # of observations
%         results.nvar  = # of explanatory variables in x
%         results.tnvar = nvar + # fixed effects
%         results.iter  = # of iterations taken
%         results.rmax  = 1/max eigenvalue of W (or rmax if input)
%         results.rmin  = 1/min eigenvalue of W (or rmin if input)
%         results.lflag = lflag from input
%         results.liter = info.iter option from input
%         results.order = info.order option from input
%         results.limit = matrix of [rho lower95,logdet approx, upper95] intervals
%                         for the case of lflag = 1
```

```
%         results.time1 = time for log determinant calcluation
%         results.time2 = time for eigenvalue calculation
%         results.time3 = time for hessian or information matrix calculation
%         results.time4 = time for optimization
%         results.time  = total time taken
%         results.lndet = a matrix containing log-determinant information
%                         (for use in later function calls to save time)
%  ---------------------------------------------------
%  NOTES: if you use lflag = 1 or 2, info.rmin will be set = -1
%                                    info.rmax will be set = 1
%         For number of spatial units < 500 you should use lflag = 0 to get
%         exact results,
%         Fixed effects and their t-values are calculated as the deviation
%         from the mean intercept
% ---------------------------------------------------
%
% Updated by: J.Paul Elhorst summer 2008
% University of Groningen
% Department of Economics
% 9700AV Groningen
% the Netherlands
% j.p.elhorst@rug.nl
%
% REFERENCES:
% Elhorst JP (2003) Specification and Estimation of Spatial Panel Data Models,
% International Regional Science Review 26: 244-268.
% Elhorst JP (2009) Spatial Panel Data Models. In Fischer MM, Getis A (Eds.)
% Handbook of Applied Spatial Analysis, Ch. C.2. Springer: Berlin Heidelberg New York.

% This function is partly based on James. P LeSage's function SEM

time1 = 0;
time2 = 0;
time3 = 0;

timet = clock; % start the clock for overall timing

W=sparse(W);

% if we have no options, invoke defaults
if nargin == 4
    info.lflag = 1;
    info.model = 0;
    info.Nhes=500;
    fprintf(1,'default: pooled model without fixed effects \n');
end;

fe=0;
model=0;
Nhes=500;

fields = fieldnames(info);
nf = length(fields);
if nf > 0
    for i=1:nf
        if strcmp(fields{i},'model') model = info.model;
        elseif strcmp(fields{i},'fe') fe = info.fe;
        elseif strcmp(fields{i},'Nhes') Nhes = info.Nhes;
        end
    end
end
if model==0
    results.meth='psem';
elseif model==1
    results.meth='semsfe';
elseif model==2
    results.meth='semtfe';
elseif model==3
    results.meth='semstfe';
else
    error('sem_panel: wrong input number of info.model');
end
```

```
% check size of user inputs for comformability
[nobs nvar] = size(x);
[N Ncol] = size(W);
if N ~= Ncol
error('sem: wrong size weight matrix W');
elseif N ~= nobs/T
error('sem: wrong size weight matrix W or matrix x');
end;
[nchk junk] = size(y);
if nchk ~= nobs
error('sem: wrong size vector y or matrix x');
end;

if (fe==1 & model==0 ) error('info.fe=1, but cannot compute fixed effects if info.model
is set to 0 or not specified'); end

results.nobs = nobs;
results.nvar = nvar;

% parse input options
[rmin,rmax,convg,maxit,detval,ldetflag,eflag,order,miter,options] =
sem_parse(info); %function of LeSage

% compute eigenvalues or limits
[rmin,rmax,time2] = sem_eigs(eflag,W,rmin,rmax,N); %function of LeSage

results.rmin = rmin;
results.rmax = rmax;
results.lflag = ldetflag;
results.miter = miter;
results.order = order;

% do log-det calculations
[detval,time1] = sem_lndet(ldetflag,W,rmin,rmax,detval,order,miter); % function of
LeSage

% demeaning of the y and x variables, depending on (info.)model

if (model==1 | model==3);
meanny=zeros(N,1);
meannx=zeros(N,nvar);
for i=1:N
    ym=zeros(T,1);
    xm=zeros(T,nvar);
    for t=1:T
        ym(t)=y(i+(t-1)*N,1);
        xm(t,:)=x(i+(t-1)*N,:);
    end
    meanny(i)=mean(ym);
    meannx(i,:)=mean(xm);
end
clear ym xm;
end % if statement

if ( model==2 | model==3)
meanty=zeros(T,1);
meantx=zeros(T,nvar);
for i=1:T
    t1=1+(i-1)*N;t2=i*N;
    ym=y([t1:t2],1);
    xm=x([t1:t2],:);
    meanty(i)=mean(ym);
    meantx(i,:)=mean(xm);
end
clear ym xm;
end % if statement

en=ones(T,1);
et=ones(N,1);
ent=ones(nobs,1);
```

```
if model==1
    ywith=y-kron(en,meanny);
    xwith=x-kron(en,meannx);
elseif model==2
    ywith=y-kron(meanty,et);
    xwith=x-kron(meantx,et);
elseif model==3
    ywith=y-kron(en,meanny)-kron(meanty,et)+kron(ent,mean(y));
    xwith=x-kron(en,meannx)-kron(meantx,et)+kron(ent,mean(x));
else
    ywith=y;
    xwith=x;
end % if statement

t0 = clock;

for t=1:T
    t1=1+(t-1)*N;t2=t*N;
    Wx([t1:t2],:)= sparse(W)*xwith([t1:t2],:);
    Wy([t1:t2],1)= sparse(W)*ywith([t1:t2],1);
end

options = optimset('MaxIter',maxit);
rho = 0.5;
converge = 1;
criteria = 1e-4;
iter = 1;

% Two-stage iterative procedure to find the ML estimates
while (converge > criteria) & (iter < maxit)

xs = xwith - rho*Wx;
ys = ywith - rho*Wy;
b = (xs'*xs)\(xs'*ys);
e = (ywith - xwith*b);
rold = rho;
[rho,like,exitflag,output] = fminbnd('f_sempanel',rmin,rmax,options,e,W,detval,T);
converge = abs(rold - rho);
iter = iter + 1;
end;

res=ys-xs*b;
sige=res'*res/nobs;
time4 = etime(clock,t0);

if exitflag == maxit
fprintf(1,'\n sem: convergence not obtained in %4d iterations \n',output.iterations);
end;
% return results
results.iter = output.iterations;
results.beta = b;
results.rho  = rho;
results.sige = sige;

% step 4) find fixed effects and their t-values
if model==1
    intercept=mean(y)-mean(x)*results.beta;
    results.con=intercept;
    results.sfe=meanny-meannx*results.beta-kron(et,intercept);
    xhat=x*results.beta+kron(en,results.sfe)+kron(ent,intercept);

results.tsfe=results.sfe./sqrt(sige/T*ones(N,1)+diag(sige*meannx*(xwith'*xwith)*me
annx'));
    results.tcon=results.con/sqrt(sige/nobs+sige*mean(x)*(xwith'*xwith)*mean(x)');
    tnvar=nvar+N;
elseif model==2
    intercept=mean(y)-mean(x)*results.beta;
    results.con=intercept;
    results.tfe=meanty-meantx*results.beta-kron(en,intercept);
    xhat=x*results.beta+kron(results.tfe,et)+kron(ent,intercept);
```

```
results.ttfe=results.tfe./sqrt(sige/N*ones(T,1)+diag(sige*meantx*(xwith'*xwith)*me
antx'));
    results.tcon=results.con/sqrt(sige/nobs+sige*mean(x)*(xwith'*xwith)*mean(x)');
    tnvar=nvar+T;
elseif model==3
    intercept=mean(y)-mean(x)*results.beta;
    results.con=intercept;
    results.sfe=meanny-meannx*results.beta-kron(et,intercept);
    results.tfe=meanty-meantx*results.beta-kron(en,intercept);

results.tsfe=results.sfe./sqrt(sige/T*ones(N,1)+diag(sige*meannx*(xwith'*xwith)*me
annx'));

results.ttfe=results.tfe./sqrt(sige/N*ones(T,1)+diag(sige*meantx*(xwith'*xwith)*me
antx'));
    results.tcon=results.con/sqrt(sige/nobs+sige*mean(x)*(xwith'*xwith)*mean(x)');

xhat=x*results.beta+kron(en,results.sfe)+kron(results.tfe,et)+kron(ent,intercept);
    tnvar=nvar+N+T-1;
else
    xhat=x*results.beta;
    tnvar=nvar;
end
results.tnvar=tnvar;
results.resid = y - xhat;
yme=y-mean(y);
rsqr2=yme'*yme;
rsqr1 = results.resid'*results.resid;
results.rsqr=1.0-rsqr1/rsqr2; %rsquared

yhat=xhat;
ywithhat=xwith*results.beta;
res1=ywith-mean(ywith);
res2=ywithhat-mean(ywith);
rsq1=res1'*res2;
rsq2=res1'*res1;
rsq3=res2'*res2;
results.corr2=rsq1^2/(rsq2*rsq3); %corr2
results.yhat=yhat;

parm = [results.beta
        results.rho
        results.sige];

results.lik = f2_sempanel(parm,ywith,xwith,W,detval,T); %Elhorst

% Determination variance-covariance matrix
if N <= Nhes % Analytically

t0 = clock;
B = speye(N) - rho*W;
BI = inv(B); WB = W*BI;
pterm = trace(WB*WB + WB'*WB);
xpx = zeros(nvar+2,nvar+2);
% beta, beta
xpx(1:nvar,1:nvar) = (1/sige)*xs'*xs;
% rho, rho
xpx(nvar+1,nvar+1) = T*pterm;
% sige, sige
xpx(nvar+2,nvar+2) = nobs/(2*sige*sige);
% rho, sige
xpx(nvar+1,nvar+2) = (T/sige)*trace(WB);
xpx(nvar+2,nvar+1) = xpx(nvar+1,nvar+2);
xpxi=xpx\eye(size(xpx));
results.cov=xpxi(1:nvar+1,1:nvar+1);
tmp = diag(xpxi);
bvec = [results.beta
        results.rho];
results.tstat = bvec./(sqrt(tmp(1:nvar+1,1)));
time3 = etime(clock,t0);
```

```
else % asymptotic t-stats using numerical hessian

t0 = clock;
hessn = hessian('f2_sempanel',parm,ywith,xwith,W,detval,T); %Elhorst

if hessn(nvar+2,nvar+2) == 0
 hessn(nvar+2,nvar+2) = 1/sige;  % this is a hack for very large models that
end;                             % should not affect inference in these cases

xpxi = invpd(-hessn);
results.cov=xpxi(1:nvar+1,1:nvar+1);
tmp = diag(xpxi(1:nvar+1,1:nvar+1));
zip = find(tmp <= 0);
 if length(zip) > 0
 tmp(zip,1) = 1;
 fprintf(1,'sem: negative or zero variance from numerical hessian \n');
 fprintf(1,'sem: replacing t-stat with 0 \n');
 end;
 bvec = [results.beta
        results.rho];
 results.tstat = bvec./sqrt(tmp);
 if length(zip) ~= 0
 results.tstat(zip,1) = 0;
 end;
time3 = etime(clock,t0);

end; % end of t-stat calculations

results.lndet = detval;
results.time = etime(clock,timet);
results.time1 = time1;
results.time2 = time2;
results.time3 = time3;
results.time4 = time4;
results.fe    = fe;
results.N     = N;
results.T     = T;
results.model = model;


function [rmin,rmax,convg,maxit,detval,ldetflag,eflag,order,iter,options] =
sem_parse(info)
% PURPOSE: parses input arguments for far, far_g models
% ---------------------------------------------------
%  USAGE: [rmin,rmax,convg,maxit,detval,ldetflag,eflag,order,iter] = far_parse(info)
% where info contains the structure variable with inputs
% and the outputs are either user-inputs or default values
% ---------------------------------------------------

% set defaults
options = optimset('fminbnd');
options.MaxIter = 500;

eflag = 1;      % default to not computing eigenvalues
ldetflag = 1;   % default to 1999 Pace and Barry MC determinant approx
order = 50;     % there are parameters used by the MC det approx
iter = 30;      % defaults based on Pace and Barry recommendation
rmin = -0.99;      % use -1,1 rho interval as default
rmax = 0.99;
detval = 0;     % just a flag
convg = 0.0001;
maxit = 500;

fields = fieldnames(info);
nf = length(fields);
if nf > 0

 for i=1:nf
    if strcmp(fields{i},'rmin')
        rmin = info.rmin;  eflag = 1;
    elseif strcmp(fields{i},'rmax')
        rmax = info.rmax;  eflag = 1;
```

```
    elseif strcmp(fields{i},'eigs')
        eflag = info.eigs; % flag for compute the eigenvalues
    elseif strcmp(fields{i},'convg')
       options.TolFun = info.convg;
    elseif strcmp(fields{i},'maxit')
        options.MaxIter = info.maxit;
    elseif strcmp(fields{i},'lndet')
    detval = info.lndet;
    ldetflag = -1;
    eflag = 1;
    rmin = detval(1,1);
    nr = length(detval);
    rmax = detval(nr,1);
    elseif strcmp(fields{i},'lflag')
        tst = info.lflag;
        if tst == 0,
        ldetflag = 0;
        elseif tst == 1,
        ldetflag = 1;
        elseif tst == 2,
        ldetflag = 2;
        else
        error('sar: unrecognizable lflag value on input');
        end;
    elseif strcmp(fields{i},'order')
        order = info.order;
    elseif strcmp(fields{i},'iter')
        iter = info.iter;
    end;
 end;

else, % the user has input a blank info structure
      % so we use the defaults
end;

function [rmin,rmax,time2] = sem_eigs(eflag,W,rmin,rmax,n);
% PURPOSE: compute the eigenvalues for the weight matrix
% ---------------------------------------------------
%  USAGE: [rmin,rmax,time2] = far_eigs(eflag,W,rmin,rmax,W)
% where eflag is an input flag, W is the weight matrix
%       rmin,rmax may be used as default outputs
% and the outputs are either user-inputs or default values
% ---------------------------------------------------


if eflag == 0
t0 = clock;
opt.tol = 1e-3; opt.disp = 0;
lambda = eigs(sparse(W),speye(n),1,'SR',opt);
rmin = 1/lambda;
rmax = 1;
time2 = etime(clock,t0);
else
time2 = 0;
end;


function [detval,time1] = sem_lndet(ldetflag,W,rmin,rmax,detval,order,iter);
% PURPOSE: compute the log determinant |I_n - rho*W|
% using the user-selected (or default) method
% ---------------------------------------------------
%  USAGE: detval = far_lndet(lflag,W,rmin,rmax)
% where eflag,rmin,rmax,W contains input flags
% and the outputs are either user-inputs or default values
% ---------------------------------------------------


% do lndet approximation calculations if needed
if ldetflag == 0 % no approximation
t0 = clock;
out = lndetfull(W,rmin,rmax);
time1 = etime(clock,t0);
```

```
tt=rmin:.001:rmax; % interpolate a finer grid
outi = interp1(out.rho,out.lndet,tt','spline');
detval = [tt' outi];

elseif ldetflag == 1 % use Pace and Barry, 1999 MC approximation

t0 = clock;
out = lndetmc(order,iter,W,rmin,rmax);
time1 = etime(clock,t0);
results.limit = [out.rho out.lo95 out.lndet out.up95];
tt=rmin:.001:rmax; % interpolate a finer grid
outi = interp1(out.rho,out.lndet,tt','spline');
detval = [tt' outi];

elseif ldetflag == 2 % use Pace and Barry, 1998 spline interpolation

t0 = clock;
out = lndetint(W,rmin,rmax);
time1 = etime(clock,t0);
tt=rmin:.001:rmax; % interpolate a finer grid
outi = interp1(out.rho,out.lndet,tt','spline');
detval = [tt' outi];

elseif ldetflag == -1 % the user fed down a detval matrix
    time1 = 0;
        % check to see if this is right
        if detval == 0
            error('sem: wrong lndet input argument');
        end;
        [n1,n2] = size(detval);
        if n2 ~= 2
            error('sem: wrong sized lndet input argument');
        elseif n1 == 1
            error('sem: wrong sized lndet input argument');
        end;
end;
```

Lampiran 16. *Matlab Code* FDB SEM Panel *Spatial Fixed Effect*

```
function result = sem_fdb(y,x,W,B,alpha)

% FDB SEM Panel Spatial Fixed Effect
% y = variabel dependen
% x = variabel independen
% W = matriks penimbang spasial
% B = banyaknya replikasi
% ---------------------------------------------------

format short

model=1

[nobs K]=size(x);
T=3;
N=10


% Model SEM Panel untuk mendapatkan residual
info.model=1;
results=sem_panel_FE(y,x,W,T,info);

u=results.resid;
r=results.rho;
b=results.beta;
c=results.con;
FE=results.sfe;

It=eye(T,T);
I=eye(nobs,nobs);
Ww=kron(It,W);
Wi=inv([I-r*Ww]);
XB=x*b;
ent=ones(nobs,1);
cent=kron(ent,c);
FE1=ones(T,1);
FE2=kron(FE1,FE);

Bo=zeros(B,K);                      % hasil estimasi koefisien beta
Ro=zeros(B,1);                      % hasil estimasi koefisien lambda
Co=zeros(B,1);                      % hasil estimasi intersep

% Resampling
for i=1:B;
    ur=randsample(u,nobs,true);
    urr=randsample(ur,nobs,true);
    yr=cent+XB+Wi*urr+FE2;
    bres=sem_panel_FE(yr,x,W,T,info);
    br=bres.beta;                   % estimasi beta setiap replikasi
    rr=bres.rho;                    % estimasi lambda setiap replikasi
    cr=bres.con;                    % estimasi intersep setiap replikasi
    Bo(i,:)=br';
    Ro(i,:)=rr;
    Co(i,:)=cr;
end;
%koefisien
Br=mean(Bo);                        % beta
Rr=mean(Ro);                        % lambda
Cr=mean(Co);                        % intersep

%standar error
Bse=std(Bo);
Rse=std(Ro);
Cse=std(Co);

coeff=[Rr Cr Br]';
rse=[Rse Cse Bse]';

%p-value
```

```
zval_rho=Rr/Rse;
zval_const=Cr/Cse;
for l=1:K
zval_beta(l)=Br(l)/Bse(l);
end
zvalue=[zval_rho zval_const zval_beta]';
prob=norm_prb(zvalue);

%selang kepercayaan
stat_beta=sort(Bo);
stat_rho=sort(Ro);
stat_con=sort(Co);

statmin_beta=min(stat_beta);
statmax_beta=max(stat_beta);
lower_beta=unifinv(alpha/2,statmin_beta,statmax_beta);
upper_beta=unifinv(1-(alpha/2),statmin_beta,statmax_beta);

statmin_rho=min(stat_rho);
statmax_rho=max(stat_rho);
lower_rho=unifinv(alpha/2,statmin_rho,statmax_rho);
upper_rho=unifinv(1-(alpha/2),statmin_rho,statmax_rho);

statmin_con=min(stat_con);
statmax_con=max(stat_con);
lower_con=unifinv(alpha/2,statmin_con,statmax_con);
upper_con=unifinv(1-(alpha/2),statmin_con,statmax_con);

lower=[lower_rho lower_con lower_beta]';
upper=[upper_rho upper_con upper_beta]';

%bias
bias.rho=Rr-r;
bias.con=Cr-c;
bias.beta=Br'-b;
bias=[bias.rho bias.con bias.beta']';

for t=1:T
    t1=1+(t-1)*N;t2=t*N;
    Wy(t1:t2,1)=W*y(t1:t2,1);
end

%R-square dan corr2
ent=ones(nobs,1);
ymy=y-mean(y);
xhat=x*Br'+FE2+kron(ent,Cr);
residual=y-xhat;
R1=residual'*residual;
R2=ymy'*ymy;
Rsquare=1.0-(R1/R2);
tsfe=results.tsfe;
psfe=norm_prb(tsfe);
corr2=results.corr2;

%Histogram
m=K+2;
for l=1:K
subplot(2,4,l)
hist(Bo(:,l));
title('koefisien beta')
end

subplot(2,4,K+1)
hist(Ro);
title('koefisien lambda');

% Output
result.meth='FDB SAR_FE';
result.nvar=K;
result.nobs=nobs;
result.replikasi=B;
result.coeff=coeff;
```

```
result.rse=rse;
result.prob=prob;
result.lower=lower;
result.upper=upper;
result.bias=bias;
result.sfe=FE2;
result.tsfe=tsfe;
result.psfe=psfe;
result.residual=residual;
result.Rsquare=Rsquare;
result.corr2=corr2;
```

BIOGRAFI PENULIS

Penulis dilahirkan di Bantul, Daerah Istimewa Yogyakarta pada tanggal 23 Agustus 1984. Putra pertama dari tiga bersaudara pasangan Ahmad Azri dan Siti Nuryati. Penulis saat ini telah berkeluarga dengan beristrikan Siskarossa Ika Oktora dan telah dikaruniai dua orang putri, yaitu Zahra Shareta Mumtaz dan Raihana Salsabila Mumtaza.

Riwayat pendidikan penulis adalah SD Negeri Jejeran II (lulus tahun 1995), SMP Negeri I Pleret (1995-1998), SMU Negeri 8 Yogyakarta (1998-2001), Sekolah Tinggi Ilmu Statistik (STIS) Jakarta (2001-2005). Setelah menamatkan pendidikan D IV di STIS, penulis ditugaskan di BPS Kabupaten Kepulauan Aru, Provinsi Maluku. Jabatan terakhir di BPS Kabupaten Kepulauan Aru adalah Kasi Neraca Wilayah dan Analisis Statistik. Pada tahun 2013, penulis mendapatkan kesempatan untuk melanjutkan pendidikan S2 di Jurusan Statistik FMIPA ITS. Penulis dapat dihubungi melalui email: nora.muh@gmail.com.