PENERAPAN METODE BOOTSTRAP DALAM MENENTUKAN *CONFIDENCE INTERVAL* PARAMETER REGRESI

SKRIPSI

Oleh
Betha Pungkasaning Putri
NIM 141810101056

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2019

PENERAPAN METODE BOOTSTRAP DALAM MENENTUKAN *CONFIDENCE INTERVAL* PARAMETER REGRESI

SKRIPSI

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

Oleh
Betha Pungkasaning Putri
NIM 141810101056

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2019

PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Ibunda Riyatik dan Ayahanda Sujoko yang telah membesarkan, mendidik, mendoakan dan memberikan cinta kasih sayang kepada putrinya;
2. Segenap keluarga besarku yang selalu mendukung dan mendoakanku;
3. Guru-guru sejak taman kanak-kanak hingga perguruan tinggi yang telah memberikan ilmu dan membimbing dengan kesabaran;
4. Almamater Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

MOTTO

“Musuh yang paling berbahaya di atas dunia ini adalah penakut dan bimbang.
Teman yang paling setia hanyalah keberanian dan keyakinan yang teguh”
(Andrew Jackson)

“Aku akan perintahkan diriku dan mengatakan bahwa aku mampu!
Aku akan mengalahkan keraguan, rasa takut, perasaan minder
dan menukarnya dengan keberanian”
(Merry Riana)

PERNYATAAN

Saya yang bertanda tangan di bawah ini :

nama : Betha Pungkasaning Putri

NIM : 141810101056

menyatakan dengan sesungguhnya bahwa skripsi yang berjudul "Penerapan Metode Bootstrap dalam Menentukan *Confidence Interval* Parameter Regresi" adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya dan belum pernah diajukan pada institusi manapun serta bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

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

Jember, Januari 2019
Yang menyatakan,

Betha Pungkasaning Putri
NIM 141810101056

SKRIPSI

PENERAPAN METODE BOOTSTRAP DALAM MENENTUKAN *CONFIDENCE INTERVAL* PARAMETER REGRESI

Oleh
Betha Pungkasaning Putri
NIM 141810101056

Pembimbing :

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

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

PENGESAHAN

Skripsi berjudul "Penerapan Metode Bootstrap dalam Menentukan *Confidence Interval* Parameter Regresi" telah duji dan disahkan pada :

hari, tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember

Tim Penguji:

Ketua,	Anggota I,
Dr. Alfian Futuhul Hadi, S.Si., M.Si.	Dian Anggraeni, S.Si., M.Si.
NIP. 197407192000121001	NIP. 198202162006042002
Anggota II,	Anggota III,
Dr. Mohamat Fatekurohman, S.Si., M.Si.	Prof. Drs. I Made Tirta, M.Sc., Ph.D
NIP. 196906061998031001	NIP. 195912201985031002

Mengesahkan
Dekan,

Drs. Sujito, Ph.D.
NIP. 196102041987111001

RINGKASAN

Penerapan Metode Bootstrap dalam Menentukan *Confidence Interval* Parameter Regresi; Betha Pungkasaning Putri; 141810101056; 2019; 39 Halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Metode Bootstrap adalah metode berbasis *resampling* data sampel dengan syarat pengembalian pada datanya dalam menyelesaikan statistik ukuran suatu sampel dengan harapan sampel tersebut mewakili data populasi sebenarnya. Metode Bootstrap dapat digunakan dalam model regresi tanpa diketahui bentuk distribusi dari model regresi tersebut. Metode Bootstrap dalam model regresi terdiri dari dua jenis yaitu metode Bootstrap observasi dan metode Bootstrap residual. Metode Bootstrap observasi yang di-*resampling* adalah variabel bebas dan variabel terikatnya secara berpasangan, sedangkan metode Bootstrap residual yang di-*resampling* adalah nilai residual dari model regresinya.

Penelitian ini dilakukan dengan membandingkan metode Bootstrap observasi dan Bootstrap residual yang akan dipilih dan diterapkan pada data IHK dan tingkat inflasi di Indonesia khususnya kota-kota di Pulau Jawa dan Pulau Sumatra. Data yang digunakan dalam proses membandingkan metode Bootstrap observasi dan Bootstrap residual merupakan data bangkitan. Data tersebut merupakan data yang dibuat peneliti dengan X yang digunakan yaitu `x = rnorm(49,4,2)`, nilai residual `e = rnorm(49)`, parameter regresi $\beta_0 = 1{,}2$ dan $\beta_1 = 2$ sehingga didapatkan fungsi untuk Y yaitu `y = 1.2 + 2*x + e`. Langkah selanjutnya dilakukan *resampling* Bootstrap sebanyak $\boldsymbol{B} = 100, 500, 1000, 3000$ kali untuk mengetahui *Confidence Interval* normal dan *percentile* data bangkitan tersebut. Setelah mendapatkan *Confidence Interval*, kemudian menghitung lebar selangnya. Lebar selang tersebut digunakan untuk mengetahui metode Bootstrap yang digunakan dalam penelitian ini.

Dari data hasil analisis dapat disimpulkan bahwa metode Bootstrap residual yang dipilih dan digunakan untuk menentukan *Confidence Interval* pada

data IHK dan tingkat inflasi di kota-kota Pulau Jawa dan Pulau Sumatra pada bulan Januari sampai bulan Desember 2018. Hal ini dapat ditunjukkan pada nilai *Confidence Interval* Bootstrap residual yang menghasilkan selang lebih sempit dibandingkan metode Bootstrap observasi. *Confidence Interval* normal dan *percentile* ditampilkan dalam tiap bulan dengan replikasi $\boldsymbol{B} = 100$, nilai $\alpha = 0{,}05$ dan data sebanyak 49 tiap bulannya. Dari *Confidence Interval* bulan Januari sampai bulan Desember dapat diketahui bahwa nilai parameter β_1 yang signifikan terjadi pada bulan Maret karena memiliki selang yang nilainya positif.

PRAKATA

Puji syukur kehadirat Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul "Penerapan Metode Bootstrap dalam Menentukan *Confidence Interval* Parameter Regresi". Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan studi pada Program Studi Matematika (S1) dan mencapai gelar Sarjana Sains pada Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Penyusunan skripsi ini tidak terlepas dari perhatian, bimbingan, motivasi dan petunjuk dari beberapa pihak, baik secara langsung maupun tidak langsung. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Bapak Dr. Alfian Futuhul Hadi, S.Si., M.Si., selaku Dosen Pembimbing Utama dan Ibu Dian Anggraeni, S.Si., M.Si., selaku Dosen Pembimbing Anggota yang dengan penuh kesabaran membimbing, mengarahkan dan memberikan saran dalam penyusunan skripsi ini;
2. Bapak Dr. Mohamat Fatekurohman, S.Si., M.Si. dan Prof. Drs. I Made Tirta, M.Sc., Ph.D, selaku Dosen Penguji yang telah memberikan kritik dan saran yang membangun dalam penyusunan skripsi ini;
3. Dosen dan Karyawan Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember;
4. Ibunda Riyatik dan Ayahanda Sujoko yang telah memberikan doa dan kasih sayang;
5. Kakakku Leny Maya Sari yang memberikan motivasi, memberikan saran dan mendoakan;
6. Sahabatku Aldelweis Yosmaning Genovani, Yauma Ayu Arista, Delinda Ratna Safitri, Maya Lutviana Aulia dan Rila Nickytha Dewi yang telah memberikan canda tawa, keceriaan dan dukungannya dari jauh;
7. Sahabat dan teman-temanku Rhemayzita Nur Istiqlaliyah, Ema Fahma Farikha, Binar Aulia Setyawan, Dwi Anugrah Wibisono dan Fedora Adi Brata yang telah membantu, mendoakan dan memberikan saran;

8. Teman-teman Matematika 2014 yang telah menemani selama menjadi mahasiswa dan berbagi canda tawa;
9. Serta semua pihak yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari bahwa penelitian ini masih jauh dari sempurna. Oleh karena itu penulis mengharapkan kritik dan saran demi kesempurnaan penelitian selanjutnya. Semoga skripsi ini dapat bermanfaat bagi kita semua.

Jember, Januari 2019

Penulis

DAFTAR ISI

DAFTAR GAMBAR

Halaman

DAFTAR TABEL

Halaman

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Statistika merupakan cabang ilmu matematika yang mempelajari bagaimana merencanakan, mengumpulkan, menganalisis, menginterpretasi dan mempresentasikan data (Turmudi & Harini, 2008). Ilmu statistik saat ini sudah berkembang pada berbagai bidang, salah satunya adalah bidang ekonomi. Statistik ekonomi meliputi pengumpulan data, pemrosesan, dan penyajian data ekonomi dalam bentuk grafik dan tabel, misalnya data mengenai perubahan harga-harga barang yang berlaku dari waktu ke waktu tidak menunjukkan adanya konsistensi. Perubahan harga biasanya terjadi terhadap barang yang secara umum diperlukan. Presentase perubahannya tidaklah sama dari satu barang dengan barang yang lainnya.

Dalam banyak keperluan dan perhitungannya, dibuat besaran yang dapat menggambarkan tingkat perubahan harga secara umum. Besaran tersebut dapat dinyatakan dalam indeks harga. Indeks harga yang secara umum digunakan untuk menyatakan tingkat harga dari barang yang selalu diperlukan oleh konsumen disebut indeks harga konsumen (IHK). IHK memberikan informasi mengenai perkembangan rata-rata perubahan harga sekelompok tetap barang/jasa yang pada umumnya dikonsumsi oleh rumah tangga dalam suatu kurun waktu tertentu. IHK dari waktu ke waktu menggambarkan tingkat kenaikan (inflasi) atau tingkat penurunan (deflasi) harga barang/jasa kebutuhan rumah tangga sehari-hari. Inflasi merupakan kecenderungan naiknya harga barang yang berlangsung secara terus menerus.

Dalam hal ini, peneliti menggunakan model regresi. Model regresi yang digunakan adalah model regresi linier yang diterapkan pada data IHK dan tingkat inflasi di Indonesia khususnya kota-kota di Pulau Jawa dan Pulau Sumatra. Regresi linier merupakan metode statistik yang digunakan untuk mengetahui pengaruh antara satu atau beberapa variabel terhadap variabel lain. Variabel yang mempengaruhi sering disebut variabel bebas, variabel *independent* atau variabel penjelas. Variabel yang dipengaruhi sering disebut dengan variabel terikat atau variabel *dependent*. Menurut Sembiring (1995), metode yang sering digunakan dalam menyelesaikan masalah estimasi koefisien regresi adalah metode kuadrat terkecil, karena estimator yang didapat dari metode ini merupakan estimator yang tak bias. Kelemahan metode kuadrat terkecil diantaranya adalah dibutuhkan sampel yang berukuran besar dan perhitungan yang rumit.

Peneliti sering kali menghadapi suatu masalah karena memperoleh jumlah sampel yang kecil dalam suatu pemodelan. Data yang sedikit menyebabkan sulitnya analisis data terutama untuk menentukan bentuk distribusi data dan estimasi parameternya, karena tidak cukup untuk melakukan analisis parametrik pada data yang kecil. Estimasi adalah keterampilan untuk menentukan sesuatu tanpa melakukan proses perhitungan secara eksak. Salah satu metode yang dapat digunakan untuk melakukan analisis data dengan kendala sampel data yang kecil adalah dengan menggunakan metode Bootstrap.

Metode Bootstrap adalah metode berbasis *resampling* data sampel dengan syarat pengembalian pada datanya dalam menyelesaikan statistik ukuran suatu sampel dengan harapan sampel tersebut mewakili data populasi sebenarnya. Ukuran *resampling* biasanya diambil secara ribuan kali agar dapat mewakili data populasinya. Metode Bootstrap dapat digunakan dalam model regresi tanpa diketahui bentuk distribusi dari model regresi tersebut. Metode Bootstrap dalam model regresi terdiri dari dua jenis yaitu metode Bootstrap observasi dan metode Bootstrap residual. Metode Bootstrap observasi yang di-*resampling* adalah variabel bebas dan variabel terikatnya, sedangkan dalam metode Bootstrap residual yang di-*resampling* adalah nilai residual dari model regresinya.

Salah satu kajian Bootstrap adalah membangun selang kepercayaan (*Confidence Interval*). Hasil yang didapatkan dari penerapan metode Bootstrap dalam model regresi linier digunakan untuk membangun *Confidence Interval* suatu data yang diambil. *Confidence Interval* Bootstrap untuk parameter regresi diberikan dalam interval pendekatan normal dan interval *percentile*. Efron dan Tisbhirani (1993) telah menerangkan bahwa metode Bootstrap persentil adalah suatu metode berbasis komputer yang sangat potensial untuk dipergunakan pada masalah-masalah ketidakbiasan dan keakurasian, khususnya dalam menentukan interval. Interval yang dihasilkan tidak memerlukan bantuan distribusi.

Pada penelitian ini sebelum menentukan *Confidence Interval* parameter regresi, terlebih dahulu dilakukan proses perbandingan antara metode Bootstrap observasi dan metode Bootstrap residual. Proses perbandingan dilakukan menggunakan data bangkitan. Metode Bootstrap yang mempunyai selang lebih sempit akan dipilih dan selanjutnya digunakan untuk menentukan *Confidence Interval* parameter regresi pada data IHK dan tingkat inflasi. Proses penelitian ini akan menggunakan program R.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, dapat dibuat rumusan masalah sebagai berikut :

1. Bagaimana perbandingan antara metode Bootstrap observasi dan metode Bootstrap residual untuk menentukan *Confidence Interval* ?
2. Bagaimana menentukan *Confidence Interval* estimasi parameter untuk data IHK dan tingkat inflasi menggunakan metode Bootstrap ?

1.3 Tujuan Penelitian

Tujuan yang ingin dicapai berdasarkan rumusan masalah di atas adalah :

1. Mendapatkan hasil perbandingan antara metode Bootstrap observasi dan metode Bootstrap residual untuk menentukan *Confidence Interval*.
2. Mendapatkan *Confidence Interval* estimasi parameter untuk data IHK dan tingkat inflasi menggunakan metode Bootstrap.

1.4 Manfaat Penelitian

Manfaat yang dapat diambil dari penelitian ini adalah dapat menambah wawasan dan pengetahuan tentang penerapan ilmu statistika khususnya metode Bootstrap yang dibandingkan antara Bootstrap observasi dan Bootstrap residual. Hasil dari perbandingan kemudian dapat digunakan untuk menentukan *Confidence Interval* untuk data IHK dan tingkat inflasi.

BAB 2. TINJAUAN PUSTAKA

2.1 Indeks Harga Konsumen (IHK) dan Inflasi

2.1.1 Indeks Harga Konsumen (IHK)

Pengertian IHK adalah angka indeks yang menggambarkan perubahan harga barang dan jasa yang dikonsumsi oleh masyarakat secara umum pada suatu periode tertentu dengan periode waktu yang telah ditetapkan, maka dari itu IHK merupakan indikator penting terhadap pasar keuangan (Listyowati, 2013).

Menurut BPS (2018), IHK adalah suatu indeks yang menghitung rata-rata perubahan harga dari suatu paket barang dan jasa yang dikonsumsi oleh rumah tangga dalam kurun waktu tertentu. IHK merupakan indikator yang digunakan untuk mengukur tingkat inflasi. Perubahan IHK dari waktu ke waktu menggambarkan tingkat kenaikan (inflasi) atau tingkat penurunan (deflasi) dari barang dan jasa.

2.1.2 Inflasi

Dalam ilmu ekonomi, inflasi adalah suatu proses meningkatnya harga-harga secara umum dan terus-menerus (kontinu) berkaitan dengan mekanisme pasar yang dapat disebabkan oleh berbagai faktor, antara lain konsumsi masyarakat yang meningkat, berlebihnya likuiditas di pasar yang memicu konsumsi atau spekulasi, termasuk juga akibat adanya ketidaklancaran distribusi barang (Suparmoko, 2000).

Menurut istilah dalam BPS (2018), inflasi adalah kecenderungan naiknya harga barang dan jasa secara umum yang berlangsung secara terus menerus, jika harga barang dan jasa di dalam negeri meningkat, maka inflasi mengalami kenaikan. Naiknya harga barang dan jasa tersebut menyebabkan turunnya nilai uang. Dengan demikian, inflasi dapat juga diartikan sebagai penurunan nilai uang terhadap nilai barang dan jasa secara umum.

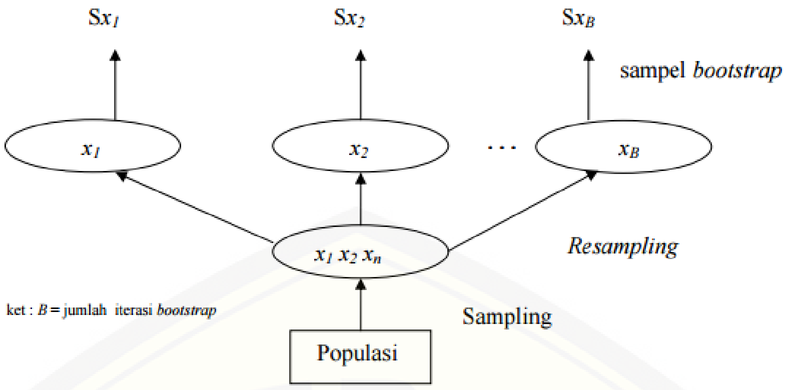
2.2 Metode Bootstrap

Bootstrap adalah suatu metode yang dapat bekerja tanpa membutuhkan asumsi distribusi karena sampel asli digunakan sebagai populasi. Bootstrap merupakan teknik *resampling* nonparametrik yang bertujuan untuk menentukan estimasi standar *error* dan *Confidence Interval* dari parameter populasi seperti mean, rasio, median, proporsi, koefisien korelasi atau koefisien regresi tanpa menggunakan asumsi distribusi. Bootstrap dapat digunakan untuk mengatasi permasalahan dalam statistika baik masalah data yang sedikit, data yang menyimpang dari asumsinya maupun data yang tidak memiliki asumsi dalam distribusinya (Sahinler & Topuz, 2007).

Bootstrap diperkenalkan pertama kali oleh Efron tahun 1979. Metode Bootstrap digunakan untuk mencari distribusi sampling dari suatu estimator dengan prosedur *resampling* dengan pengembalian dari data asli. Metode Bootstrap dilakukan dengan mengambil sampel dari sampel asli dengan ukuran sama dengan ukuran sampel asli dan dilakukan dengan pengembalian. Kedudukan sampel asli dalam metode Bootstrap dipandang sebagai populasi. Metode penyampelan ini biasa disebut dengan *resampling* Bootstrap. Bootstrap juga sering digunakan untuk mengestimasi *Confidence Interval* dari suatu parameter populasi yang tidak diketahui.

Menurut Walpole (1992) sampel adalah suatu himpunan bagian dari populasi. Istilah sampel asli digunakan untuk menyebut himpunan bagian pertama diambil dari populasi, sebelum dilakukan *resampling* yaitu proses pengambilan sampel kembali dari sampel yang telah kita ambil dari populasi, sedangkan istilah sampel Bootstrap (*resample*) digunakan untuk menyebut sampel yang telah kita *resampling* dari sampel asli. Sampel asli dilambangkan dengan $x = \{x_1, x_2, \dots, x_n\}$ dan sampel Bootstrap dilambangkan dengan $x^* = \{x_1{}^*, x_2{}^*, \dots, x_B{}^*\}$.

Proses sampling Bootstrap dilakukan dengan menggunakan bantuan program komputer, mengingat besarnya jumlah *resampling* yang bisa mencapai ribuan kali sehingga sangatlah sulit untuk melakukan perhitungan secara manual. Berikut adalah skema dari *resample* Bootstrap :



Gambar 2.1 Skema *Resample* (Sumber : Efron & Tisbhirani, 1993)

Kelebihan metode Bootstrap yaitu tidak memerlukan asumsi ketat mengenai distribusi yang harus dipenuhi, dapat digunakan untuk ukuran data yang relatif kecil, dan metode Bootstrap lebih baik dari pada metode tradisional misalnya metode Klasik dan Bayes, karena metode Bootstrap menghasilkan interval yang lebih sempit. Kelemahan metode Bootstrap yaitu metode Bootstrap biasanya memerlukan data bangkitan Bootstrap atau *resample* (sampel diperoleh dari pengacakan saling bebas dan dengan pemulihan dari distribusi empiris). Data bangkitan Bootstrap ini sulit dicari karena memerlukan beberapa kali penghitungan yang cukup rumit dan membutuhkan ketelitian yang tinggi.

Menurut Saraswati (2009) langkah-langkah dalam prosedur Bootstrap adalah :

1. Membangun distribusi empiris $\hat{F}_n(x)$ dari suatu sampel dengan menempatkan probabilitas $\frac{1}{n}$ artinya peluang terambilnya setiap sampel secara acak dari populasi memiliki kemungkinan yang sama pada setiap x_i dengan $i = 1, 2, \dots, n$.
2. Mengambil sampel random berukuran n dengan pengembalian dari fungsi distribusi empiris $\hat{F}_n(x)$ sebanyak $\boldsymbol{B}$. Hal ini dinamakan sebagai *resample* dan disebut ${x^*}_b$.
3. Menghitung statistik $\hat{\theta}$ yang diinginkan dari *resample* yang disebut ${\hat{\theta}^*}_b$ sebanyak $\boldsymbol{B}$ kali.

4. Membentuk distribusi empiris dari $\hat{\theta}^*{}_b$ dengan probabilitas masing-masing $\frac{1}{B}$ pada setiap $\hat{\theta}^*{}_1, \hat{\theta}^*{}_2, \dots, \hat{\theta}^*{}_B$. Distribusi ini adalah estimator distribusi $\hat{\theta}$ yang disebut $\hat{F}(\hat{\theta}^*)$.

Jika $\hat{\theta}$ merupakan mean (rata-rata) hasil *resample*, maka dapat ditentukan rata-rata dan variansi Bootstrapnya yaitu :

$$\bar{\theta}^* = \frac{\sum_{b=1}^{B} \hat{\theta}^*{}_b}{B} \tag{2.1}$$

dan

$$\hat{V}^* = \frac{\sum_{b=1}^{B} \left(\hat{\theta}^*{}_b - \bar{\theta}^*\right)^2}{B-1} \tag{2.2}$$

2.3 Regresi Linier

Analisis regresi merupakan salah satu analisis statistik yang sering digunakan untuk menganalisis hubungan antara dua variabel atau lebih. Menurut Drapper (1998) analisis regresi merupakan metode analisis yang dapat digunakan untuk menganalisis data dan mengambil kesimpulan yang bermakna tentang hubungan ketergantungan variabel terhadap variabel lainnya. Hubungan yang didapat pada umumnya dinyatakan dalam bentuk persamaan matematika yang menyatakan hubungan antara variabel bebas (*independent*) X dan variabel tak bebas/ terikat (*dependent*) Y, misalkan hubungan antara variabel Y_i dengan variabel X_i untuk subjek $i = 1, 2, \dots, n$ ditentukan oleh

$$\left.\begin{array}{ccc} Y_1 & = & \beta_0 + \beta_1 X_1 + \epsilon_1 \\ \vdots & \vdots & \vdots \\ Y_i & = & \beta_0 + \beta_1 X_i + \epsilon_i \\ \vdots & \vdots & \vdots \\ Y_n & = & \beta_0 + \beta_1 X_n + \epsilon_n \end{array}\right\} \tag{2.3}$$

dengan :

1. X_i adalah variabel tetap yang tidak bersifat acak (lebih lanjut diasumsikan X_i diukur tanpa kesalahan).
2. ϵ_i yaitu komponen kesalahan/ *error*, adalah berdistribusi identik dan independen normal dengan nilai tengah 0 dan varian konstan (misal σ^2).
3. Kesalahan individu satu dengan yang lainnya saling bebas yaitu untuk $i \neq i'$ maka $\epsilon_i || \epsilon_i{}'$ atau korelasi ϵ_i dengan $\epsilon_i{}'$ adalah 0.

Parameter yang menjadi kepentingan utama dalam regresi linier sederhana adalah komponen dari koefisien regresi $\beta = \binom{\beta_0}{\beta_1}$. Parameter lain yang juga perlu diestimasi adalah komponen ragam σ^2. Salah satu metode estimasi parameter dalam model regresi adalah metode kuadrat terkecil. Metode kuadrat terkecil merupakan suatu metode yang digunakan untuk menaksir parameter regresi dengan cara meminimumkan jumlah kuadrat kesalahan (*error*) dari model regresi yang terbentuk, sehingga diperoleh estimasi kuadrat terkecil dari β pada regresi linier adalah $\hat{\beta} = (X'X)^{-1}X'Y$.

Langkah-langkah untuk mengestimasi parameter dengan menggunakan metode kuadrat terkecil yaitu :

1. Mengubah model linier menjadi eksplisit terhadap galat/ *error*, karena yang akan diminimumkan adalah kuadrat galat. Berdasarkan persamaan (2.3) diperoleh rumusan galat :

$$\epsilon_i = Y_i - (\beta_0 + \beta_1 X_i) \qquad (2.4)$$

2. Mengkuadratkan *error* yang diperoleh serta menjumlahkannya untuk seluruh pasangan data, diperoleh bentuk jumlah kuadrat :

$$Q = \sum_{i=1}^{n} {\epsilon_i}^2 = \sum_{i=1}^{n} [Y_i - (\beta_0 + \beta_1 X_i)]^2 = \sum_{i=1}^{n} \left[Y_i - \sum_{j=0}^{1} \beta_j X_{ij}\right]^2 \qquad (2.5)$$

dalam hal ini $X_{i0} = 1$ dan $X_{i1} = X_i$.

3. Menurunkan bentuk kuadrat yang diperoleh terhadap parameter. Estimasi dengan metode kuadrat terkecil diproses dengan mencari minimum Q terhadap β_j. Minimum dari Q diperoleh dengan mencari turunan pertama maupun kedua

$$\frac{\partial Q}{\partial \beta_0} = -2 \sum_{i=1}^{n} [Y_i - (\beta_0 + \beta_1 X_i)]$$

$$\frac{\partial Q}{\partial \beta_1} = -2 \sum_{i=1}^{n} [Y_i - (\beta_0 + \beta_1 X_i)] X_i$$

4. Menyusun persamaan normal yang diperoleh dari sistem persamaan $\frac{\partial Q}{\partial \beta_j} = 0$, diperoleh persamaan normal :

$$\left.\begin{aligned}\sum_{i=1}^{n}[Y_i-(\beta_0+\beta_1X_i)] &= 0\\ \sum_{i=1}^{n}[Y_i-(\beta_0+\beta_1X_i)]X_i &= 0\end{aligned}\right\} \tag{2.6}$$

Persamaan (2.6) selanjutnya dapat disederhanakan menjadi :

$$\sum_{i=1}^{n} Y_i - n\beta_0 - \beta_1 \sum_{i=1}^{n} X_i = 0 \tag{2.7}$$

$$\sum_{i=1}^{n} X_iY_i - \beta_0 \sum_{i=1}^{n} X_i - \beta_1 \sum_{i=1}^{n} {X_i}^2 = 0 \tag{2.8}$$

5. Dari persamaan (2.7) diperoleh :

$$\hat{\beta}_0 = \frac{1}{n}\sum_{i=1}^{n} Y_i - \beta_1 \frac{1}{n}\sum_{i=1}^{n} X_i$$

$$= \bar{Y} - \beta_1\bar{X} \tag{2.9}$$

Hasil persamaan (2.9) selanjutnya disubtitusikan pada persamaan (2.8) sehingga diperoleh :

$$\hat{\beta}_1 = \frac{n\sum_{i=1}^{n} X_iY_i - \sum_{i=1}^{n} X_i \sum_{i=1}^{n} Y_i}{n\sum_{i=1}^{n} {X_i}^2 - \left(\sum_{i=1}^{n} X_i\right)^2} \tag{2.10}$$

Sebagaimana telah disampaikan bahwa metode kuadrat terkecil belum memanfaatkan informasi distribusi dari ϵ_i, oleh karena itu apabila σ^2 tidak diketahui tidak ada cara khusus dengan metode kuadrat terkecil untuk mengestimasi σ^2 namun σ^2 biasa diestimasi dari rata-rata kuadrat deviasi data terhadap garis regresi yang diperoleh dari $\hat{\beta}_j$. Derajat kebebasan yang dimiliki oleh deviasi ini adalah $n-k$ dengan k banyaknya penduga β_j, maka untuk model dengan dua parameter β_0 dan β_1 yaitu :

$$\hat{\sigma}^2 = {s_e}^2 = \frac{1}{n-2}\sum_{i=1}^{n}\left[Y_i - \left(\hat{\beta}_0 + \hat{\beta}_1X_i\right)\right]^2 \tag{2.11}$$

Apabila σ^2 tidak diketahui tetapi diganti dengan $\hat{\sigma}^2 = {s_e}^2$ maka $var(\hat{\beta}_j)$ dinotasikan dengan $s^2(\hat{\beta}_j)$ dengan $j = 0{,}1$ menjadi :

$$s^2(\hat{\beta}_0) = \left[\frac{1}{n} + \frac{\left(1/n\sum_{i=1}^{n} X_i\right)^2}{\sum_{i=1}^{n} {X_i}^2 - 1/n\left(\sum_{i=1}^{n} X_i\right)^2}\right]{s_e}^2 \tag{2.12}$$

dan

$$s^2(\hat{\beta}_1) = \frac{{s_e}^2}{\sum_{i=1}^{n} {X_i}^2 - 1/n\left(\sum_{i=1}^{n} X_i\right)^2} \tag{2.13}$$

(Tirta, 2009).

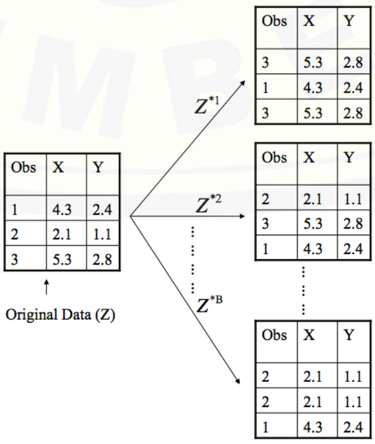
Dalam Rumtiasih dan Suparman (2015), Bootstrap dapat diterapkan dalam model regresi. Metode Bootstrap terdiri dari dua jenis yaitu metode Bootstrap observasi dan metode Bootstrap residual. Metode Bootstrap observasi dalam mengestimasi parameter regresi dilakukan dengan cara menggandengkan antara variabel bebas dengan variabel responnya, setelah itu dilakukan *resampling* Bootstrap sebanyak ***B*** kali dengan teknik pengembalian. Apabila proses *resampling* telah selesai dilakukan selanjutnya dapat dihitung nilai ragam masing-masing parameter regresi. Tingkat keakurasian metode ini dapat dilihat dengan menggunakan pendugaan *Confidence Interval* normal dan *Confidence Interval persentile*.

Adapun langkah-langkah metode Bootstrap observasi yaitu :

1. Mengkontruksi sampel dari data berpasangan (Y_i, X_i) dengan $i = 1, 2, \dots, n$ secara acak dengan peluang $\frac{1}{n}$ artinya peluang terambilnya setiap sampel secara acak dari populasi memiliki kemungkinan yang sama. Data yang diperoleh merupakan data asli yang berasal dari populasi.
2. Misalkan data hasil pengacakan tersebut dinyatakan dalam (Y^*, X^*) maka didapatkan $Y^* = \beta X^* + \epsilon$.
3. Berdasarkan model tersebut akan dihitung estimasi parameter β yakni dengan nilai estimasi parameter ${\beta^*}_j$ dengan $j = 0, 1$.
4. Supaya menghasilkan estimasi parameter yang lebih baik atau mendekati nilai sebenarnya, ulangi langkah-langkah sebelumnya sebanyak ***B*** kali

(Fitri, 2016).

Berikut skema contoh data menggunakan Bootstrap observasi :



Gambar 2.2 Skema Bootstrap Observasi (Sumber : James *et al*., 2009)

Pendekatan Bootstrap jika diulang lebih dari satu kali akan memberikan hasil yang berbeda. Perbedaan hasil ini disebabkan oleh simulasi yang dilakukan, jika dapat dilakukan dengan menggunakan semua kemungkinan sampel yaitu n^n maka hasilnya akan sama.

Prinsip Bootstrap residual adalah mencocokkan model regresi dan memperoleh residual sebanyak n. Prosedur Bootstrap residual sama dengan prosedur Bootstrap pada umumnya, hanya saja dalam Bootstrap residual nilai residualnya yang di-*resampling*. Misalkan fungsi regresi sampel asli yaitu :

$$Y_i = \beta_0 + \beta_1 X_i \tag{2.14}$$

untuk $i = 1, 2, 3, \dots, n$ dan estimasinya :

$$\widehat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_i \tag{2.15}$$

Maka prosedur untuk Bootstrap residual adalah :

1. Menghitung $\hat{\beta}_0$ dan $\hat{\beta}_1$ dari sampel asli untuk mendapatkan nilai residual.

$$\epsilon_i = Y_i - \widehat{Y}_i \tag{2.16}$$

2. Mengambil sampel Bootstrap berukuran n dari residual secara random dengan pengembalian. Jika

$$\epsilon_i = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

maka sampel bootstrap ke- b $(b = 1, 2, 3, \dots, B)$ dapat dituliskan

$$\epsilon_i^{*(b)} = \begin{bmatrix} \epsilon_1^{*(b)} \\ \epsilon_2^{*(b)} \\ \vdots \\ \epsilon_n^{*(b)} \end{bmatrix} \tag{2.17}$$

sehingga

$$\epsilon_i^{*(1)} = \begin{bmatrix} \epsilon_1^{*(1)} \\ \epsilon_2^{*(1)} \\ \vdots \\ \epsilon_n^{*(1)} \end{bmatrix}, \epsilon_i^{*(2)} = \begin{bmatrix} \epsilon_1^{*(2)} \\ \epsilon_2^{*(2)} \\ \vdots \\ \epsilon_n^{*(2)} \end{bmatrix}, \dots, \epsilon_i^{*(B)} = \begin{bmatrix} \epsilon_1^{*(B)} \\ \epsilon_2^{*(B)} \\ \vdots \\ \epsilon_n^{*(B)} \end{bmatrix}.$$

3. Menghitung nilai Bootstrap untuk Y_i dengan menambah $\epsilon_i^{*(b)}$ sehingga diperoleh

$$Y_i^{*(b)} = \hat{\beta}_0 + \hat{\beta}_1 X_i + \epsilon_i^{*(b)} \tag{2.18}$$

4. Menghitung koefisien regresi untuk sampel Bootstrap ${Y_i}^{*(b)}$ dengan X_i diperoleh $\hat{\beta}^{*(b)} = (X'X)^{-1}X'{Y_i}^{*(b)}$.
5. Mengulangi proses di atas sebanyak $\boldsymbol{B}$ kali diperoleh $\hat{\beta}^{*(1)}, \hat{\beta}^{*(2)}, \dots, \hat{\beta}^{*(B)}$.

2.4 *Confidence Interval* Bootstrap

Confidence Interval adalah rentang antara dua nilai dimana nilai suatu sampel mean tepat berada di tengah-tengahnya. *Confidence Interval* pada dasarnya menunjukkan tingkat keterpercayaan sejauh mana statistik sampel dapat mengestimasi dengan benar parameter populasi dan/ atau sejauh mana pengambilan keputusan mengenai hasil uji hipotesis nol diyakini kebenarannya (Sembiring, 1995).

Estimasi *Confidence Interval* untuk parameter $\hat{\beta}_j$ diperoleh dengan melihat nilai t atau z yang membatasi *presentase* atau luas daerah dari kurva fungsi kepadatannya. Penduga selang β_j untuk taraf kepercayaan $(1-\alpha)100\%$ atau taraf signifikansi $\alpha\ x\ 100\%$ jika n cukup besar :

$$\hat{\beta}_j - z_{\alpha/2}\sqrt{var(\hat{\beta}_j)} \leq \beta_j \leq \hat{\beta}_j + z_{\alpha/2}\sqrt{var(\hat{\beta}_j)} \tag{2.19}$$

Keterangan :

$\hat{\beta}_j$ = estimasi koefisien regresi dengan $j = 0{,}1$

$z_{\alpha/2}$ = harga z pada luasan kurva normal yang sesuai dengan tingkat konfidensi

$var(\hat{\beta}_j)$ = parameter varians

(Tirta, 2009).

Estimasi *Confidence Interval* Bootstrap untuk parameter regresi diberikan dalam interval pendekatan normal dan *percentile*. *Confidence Interval* Bootstrap dengan pendekatan normal sebenarnya analog dengan *Confidence Interval* standar. Menurut Bennett (2009) pemanfaatan metode Bootstrap dalam mengkonstruksi interval ini adalah untuk menentukan standar *error* dari estimator. Standar *error* Bootstrap $Se_B\left(\hat{\beta}^*{}_j\right)$ diperoleh dari akar varians. *Confidence Interval* Bootstrap pendekatan normal $(1-\alpha)100\%$ untuk β_j yaitu :

$$\hat{\beta}_j + z_{\alpha/2} Se_B\left(\hat{\beta}^*{}_j\right) < \beta_j < \hat{\beta}_j + z_{1-\alpha/2} Se_B\left(\hat{\beta}^*{}_j\right) \tag{2.20}$$

Confidence Interval Bootstrap *persentile* didasarkan pada distribusi estimator Bootstrap untuk setiap j dibentuk distribusi empiris untuk $\hat{\beta}^{*(1)}{}_j, \hat{\beta}^{*(2)}{}_j, \dots, \hat{\beta}^{*(B)}{}_j$ misalkan $\hat{F}^*$. Berdasarkan distribusi ini dapat dihitung nilai *persentile* yang merupakan ide dasar konstruksi *Confidence Interval* Bootstrap *persentile*. *Confidence Interval* Bootstrap *persentile* $(1-\alpha)100\%$ untuk β_j yaitu :

$$\left(\left(\hat{F}^*\right)^{-1}{}_{(\alpha/2)}, \left(\hat{F}^*\right)^{-1}{}_{(1-\alpha/2)}\right) \tag{2.21}$$

dengan $\left(\hat{F}^*\right)^{-1}{}_{(\alpha/2)}$ adalah persentil $100\%(\alpha/2)$ dan $\left(\hat{F}^*\right)^{-1}{}_{(1-\alpha/2)}$ merupakan persentil ke $100\%(1-\alpha/2)$ dari distribusi $\hat{F}^*$.

BAB 3. METODE PENELITIAN

3.1 Data Penelitian

Data yang digunakan adalah data Indeks Harga Konsumen (IHK) dan tingkat inflasi pada bulan Januari sampai bulan Desember 2018 di Kota-kota Pulau Jawa dan Pulau Sumatra yang terdapat pada www.bps.go.id.

3.2 Identifikasi Variabel

Variabel yang digunakan dalam penelitian ini yaitu :

Y : variabel terikat yaitu tingkat inflasi bulanan

X : variabel bebas yaitu indeks harga konsumen

β_0 : parameter *intercept*

β_1 : parameter koefisien regresi variabel bebas X

ϵ : residual

3.3 Langkah-langkah Penelitian

Langkah-langkah yang akan dilakukan dalam penelitian ini adalah :

1. Studi Literatur

   Studi literatur dilakukan untuk mendapatkan informasi dari buku, jurnal dan skripsi yang membahas mengenai *Confidence Interval* Bootstrap. Informasi yang didapat dari studi literatur ini akan digunakan sebagai acuan untuk menyelesaikan penelitian ini.
2. Pengumpulan Data

   Penelitian ini menggunakan data IHK dan tingkat inflasi pada bulan Januari sampai bulan Desember 2018 di kota-kota Pulau Jawa dan Pulau Sumatra.
3. Simulasi Data Bangkitan

   Pada proses ini menggunakan data bangkitan yang ditentukan oleh peneliti dengan membangkitkan variabel bebas `x = rnorm(49,4,2)`, residual ϵ dengan `e = rnorm(49)` serta memasukkan nilai parameter $\beta_0 = 1{,}2$ dan $\beta_i = 2$ sehingga didapatkan variabel terikat `y = 1.2 + 2*x + e`.

4. Pengolahan Data

   Setelah proses membangkitkan data selesai maka dilakukan pengolahan data yang dilakukan menggunakan metode Bootstrap. Menentukan (Y^*, X^*) untuk metode Bootstrap observasi, nilai residual $\epsilon_i{}^*$ untuk metode Bootstrap residual serta estimasi parameter koefisien regresi $\hat{\beta}^*$ dengan menggunakan program R.

5. Perbandingan Metode Bootstrap

   Membandingkan antara metode Bootstrap observasi dan metode Bootstrap residual dengan mengevaluasi nilai intervalnya.

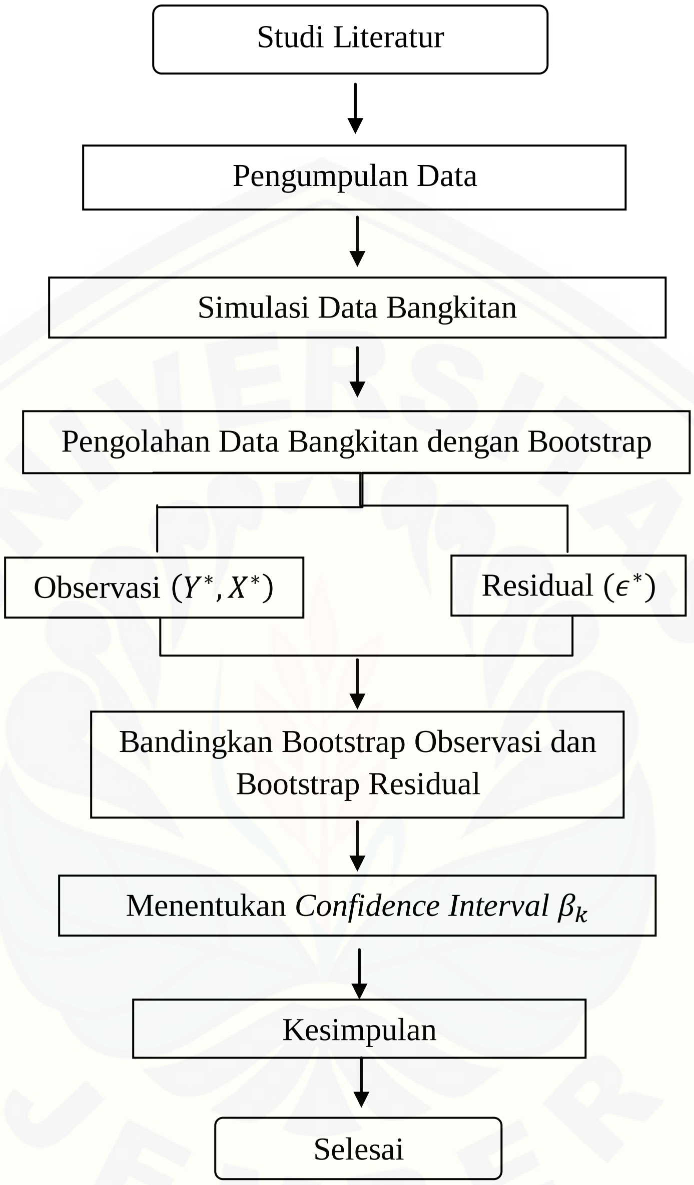
6. Menentukan *Confidence Interval*

   Confidence Interval β_k untuk data IHK dan tingkat inflasi ditentukan dengan menggunakan pendekatan normal dan interval *persentile* dengan penerapan metode Bootstrap hasil perbandingan pada point 5.

7. Kesimpulan

   Interpretasi akhir dilakukan untuk mengetahui apakah hasil pengolahan data sesuai dengan yang diharapkan yang kemudian akan ditarik kesimpulan dari hasil.

Skema dari langkah-langkah penelitian dapat dilihat pada Gambar 3.1 :



Gambar 3.1 *Flowchart* Metode Penelitian

BAB 5. PENUTUP

5.1 Kesimpulan

Berdasarkan hasil analisis data dan pembahasan maka dapat disimpulkan bahwa :

a. Diantara dua metode Bootstrap yang dibandingkan lebar selangnya yaitu metode Bootstrap residual dan Bootstrap observasi untuk semua parameter normal dan *percentile* dengan memberikan nilai α dan σ yang sama dan replikasi $\boldsymbol{B} = 100, 500, 1000, 3000$ kali, maka dapat disimpulkan bahwa metode Bootstrap residual yang digunakan untuk menentukan *Confidence Interval* pada data IHK dan tingkat inflasi bulan Januari sampai bulan Desember 2018 di kota-kota Pulau Jawa dan Pulau Sumatra dalam penelitian ini, karena selangnya yang cukup sempit.

b. *Confidence Interval* untuk data IHK dan tingkat inflasi di kota-kota Pulau Jawa dan Sumatra pada bulan Januari sampai bulan Desember 2018 diperoleh dengan menerapkan metode Bootstrap residual. *Confidence Interval* normal dan *percentile* ditampilkan dalam tiap bulan dengan replikasi $\boldsymbol{B} = 100$, nilai $\alpha = 0{,}05$ dan data sebanyak 49 tiap bulannya. Dari *Confidence Interval* bulan Januari sampai bulan Desember dapat diketahui bahwa nilai parameter β_1 yang signifikan terjadi pada bulan Maret karena memiliki selang yang nilainya positif.

5.2 Saran

Saran untuk penelitian selanjutnya yaitu diharapkan dapat menggunakan data perbandingan IHK dan tingkat inflasi baru dengan jumlah data yang lebih banyak untuk model regresi, misalkan data perbandingan IHK dan tingkat inflasi seluruh Indonesia atau menggunakan data dengan variabel bebas yang lebih dari satu serta melakukan *resampling* dengan $\boldsymbol{B}$ yang bervariasi.

DAFTAR PUSTAKA

Bennett, P. J. 2009. *Introduction to the Bootstrap and Robust Statistics*. Winter Term, PSY711/712.

BPS. 2018. *Perkembangan Indeks Harga Konsumen/ Inflasi*. www.bps.go.id. [Diakses pada 7 Juli 2018].

Drapper, NR. 1998. *Applied Regression Analysis*. Canada : A Willey Interscience Publication.

Efron, B & Tibshirani, R. 1993. *An Introduction to the Bootstrap*. New York : Chapman & Hall.

Fitri, D. A. 2016. Estimasi Parameter Model Regresi Linier Berganda dengan Teknik Bootstrap. *Jurnal Matematika UNAND*. Vol. 3 No. 3.

James, G. *et al*. 2009. *An Introduction to Statistical Learning with Applications in R*. USA : Department of Statistics Stanford University.

Listyowati & Brodjol, S. SU. 2013. Pemodelan Indeks Harga Konsumen (IHK) Umum Berdasarkan IHK Sektor Bahan Makanan dan IHK Sektor Makanan Jadi, Minuman/ Rokok. *Jurnal Sains dan Seni Pomits* Vol. 2, No. 2.

Rumtiasih, H & Suparman. 2015. Metode Bootstrap dalam Inferensi Model Regresi Polinomial. *Jurnal Konvergensi*. Vol 5, No 1.

Sahinler, S. & Topuz, D. 2007. Bootstrap and Jackknife Resampling Algorithms for Estimation of Regression Parameters. *JAQM*, no. 2, 2, 188-199.

Saraswati. 2009. *Estimasi Densitas Kernel Epanechnikov Rata-rata Resample Bootstrap untuk Penentuan Waktu Panen Optimal Tanaman Rami*. FMIPA, Universitas Sebelas Maret Surakarta.

Sembiring. RK. 1995. *Analisis Regresi*. ITB : Bandung.

Suparmoko. 2000. *Keuangan Negara: Teori dan Praktek*. BPFE-Yogyakarta. Hal 4,44-45.

Tirta, I. M. 2009. *Analisis Regresi dengan R*. Jember : Jember University Press.

Turmudi & Harini, S. 2008. *Metode Statistika*. Malang : UIN-Malang Press.

Walpole, R. E. 1992. *Pengantar Statistika*. edisi ke-3. Jakarta : Gramedia Pustaka Utama.

LAMPIRAN

A. Data Penelitian

a) Perbandingan Indeks dan Tingkat Inflasi Januari 2018 Kota-kota di Pulau Jawa dan Pulau Sumatra

	Kota	IHK	Tingkat Inflasi (%)		Kota	IHK	Tingkat Inflasi (%)
1	Jakarta	131,53	0,43	27	Meulaboh	131,63	-0,14
2	Bogor	132,75	0,67	28	Banda Aceh	125,35	-0,33
3	Sukabumi	131,25	0,79	29	Lhokseumawe	128,94	0,32
4	Bandung	130,68	0,83	30	Sibolga	138,34	1,28
5	Cirebon	127,72	1,01	31	Pematangsiantar	136,89	0,54
6	Bekasi	127,96	0,94	32	Medan	138,14	0,71
7	Depok	130,12	0,68	33	Padangsidimpuan	130,51	0,28
8	Tasikmalaya	130,55	1,00	34	Padang	136,88	0,43
9	Cilacap	135,22	1,33	35	Bukittinggi	128,98	0,75
10	Purwokerto	129,70	1,29	36	Tembilahan	136,38	0,70
11	Kudus	138,03	1,00	37	Pekanbaru	133,95	0,59
12	Surakarta	126,91	0,55	38	Dumai	134,30	0,36
13	Semarang	130,17	0,81	39	Bungo	129,96	0,25
14	Tegal	128,90	1,15	40	Jambi	131,81	0,91
15	Yogyakarta	129,10	0,55	41	Palembang	129,29	0,60
16	Jember	127,59	0,56	42	Lubuklinggau	129,82	0,88
17	Banyuwangi	127,26	0,70	43	Bengkulu	141,22	0,99
18	Sumenep	128,01	0,64	44	Bandar Lampung	133,17	1,42
19	Kediri	126,95	0,14	45	Metro	137,85	0,48
20	Malang	132,00	0,69	46	Tanjung Pandan	140,17	0,44
21	Probolinggo	127,37	0,29	47	Pangkalpinang	138,69	1,27
22	Madiun	129,41	0,62	48	Batam	133,51	0,99
23	Surabaya	132,09	0,63	49	Tanjung Pinang	130,49	0,18
24	Tangerang	138,34	0,04				
25	Cilegon	138,42	0,41				
26	Serang	141,17	0,91				

b) Perbandingan Indeks dan Tingkat Inflasi Februari 2018 Kota-kota di Pulau Jawa dan Pulau Sumatra

	Kota	IHK	Tingkat Inflasi (%)		Kota	IHK	Tingkat Inflasi (%)
1	Jakarta	132,02	0,37	27	Meulaboh	131,30	-0,25
2	Bogor	133,21	0,35	28	Banda Aceh	125,09	-0,21
3	Sukabumi	131,53	0,21	29	Lhokseumawe	128,26	-0,53
4	Bandung	130,97	0,22	30	Sibolga	137,65	-0,50
5	Cirebon	128,48	0,60	31	Pematangsiantar	136,10	-0,58
6	Bekasi	128,72	0,59	32	Medan	136,82	-0,96
7	Depok	130,50	0,29	33	Padangsidimpuan	129,75	-0,58
8	Tasikmalaya	130,83	0,21	34	Padang	136,76	-0,09
9	Cilacap	135,73	0,38	35	Bukittinggi	128,69	-0,22
10	Purwokerto	129,76	0,05	36	Tembilahan	135,87	-0,37
11	Kudus	138,81	0,57	37	Pekanbaru	133,59	-0,27
12	Surakarta	127,53	0,49	38	Dumai	133,98	-0,24
13	Semarang	130,65	0,37	39	Bungo	129,79	-0,13
14	Tegal	128,97	0,05	40	Jambi	130,71	-0,83
15	Yogyakarta	129,04	-0,05	41	Palembang	129,21	-0,06
16	Jember	127,82	0,18	42	Lubuklinggau	129,79	-0,02
17	Banyuwangi	127,47	0,17	43	Bengkulu	140,80	-0,30
18	Sumenep	128,11	0,08	44	Bandar Lampung	133,25	0,06
19	Kediri	127,28	0,26	45	Metro	138,11	0,19
20	Malang	132,22	0,17	46	Tanjung Pandan	139,77	-0,29
21	Probolinggo	127,76	0,31	47	Pangkalpinang	137,54	-0,83
22	Madiun	129,73	0,25	48	Batam	133,43	-0,06
23	Surabaya	132,27	0,14	49	Tanjung Pinang	130,68	0,15
24	Tangerang	138,71	0,27				
25	Cilegon	138,76	0,25				
26	Serang	141,38	0,15				

c) Perbandingan Indeks dan Tingkat Inflasi Maret 2018 Kota-kota di Pulau Jawa dan Pulau Sumatra

	Kota	IHK	Tingkat Inflasi (%)		Kota	IHK	Tingkat Inflasi (%)
1	Jakarta	132,14	0,09	27	Meulaboh	131,57	0,21
2	Bogor	133,48	0,20	28	Banda Aceh	124,99	-0,08
3	Sukabumi	131,70	0,13	29	Lhokseumawe	127,94	-0,25
4	Bandung	131,24	0,21	30	Sibolga	138,74	0,79
5	Cirebon	128,11	-0,29	31	Pematangsiantar	136,33	0,17

6	Bekasi	129,57	0,66	32	Medan	137,66	0,61
7	Depok	130,68	0,14	33	Padangsidimpuan	130,18	0,33
8	Tasikmalaya	130,96	0,10	34	Padang	137,19	0,31
9	Cilacap	135,58	-0,11	35	Bukittinggi	129,05	0,28
10	Purwokerto	129,19	-0,44	36	Tembilahan	137,75	1,38
11	Kudus	138,90	0,06	37	Pekanbaru	134,34	0,56
12	Surakarta	127,76	0,18	38	Dumai	134,05	0,05
13	Semarang	130,71	0,05	39	Bungo	130,20	0,32
14	Tegal	128,62	-0,27	40	Jambi	131,53	0,63
15	Yogyakarta	129,23	0,15	41	Palembang	129,72	0,39
16	Jember	127,72	-0,08	42	Lubuklinggau	130,18	0,30
17	Banyuwangi	127,62	0,12	43	Bengkulu	141,32	0,37
18	Sumenep	128,12	0,01	44	Bandar Lampung	133,40	0,11
19	Kediri	127,41	0,10	45	Metro	138,34	0,17
20	Malang	132,38	0,12	46	Tanjung Pandan	139,70	-0,05
21	Probolinggo	127,59	-0,13	47	Pangkalpinang	137,92	0,28
22	Madiun	129,76	0,02	48	Batam	133,79	0,27
23	Surabaya	132,35	0,06	49	Tanjung Pinang	130,45	-0,18
24	Tangerang	139,38	0,48				
25	Cilegon	138,85	0,06				
26	Serang	141,71	0,23				

d) Perbandingan Indeks dan Tingkat Inflasi April 2018 Kota-kota di Pulau Jawa dan Pulau Sumatra

	Kota	IHK	Tingkat Inflasi (%)		Kota	IHK	Tingkat Inflasi (%)
1	Jakarta	132,22	0,06	27	Meulaboh	130,88	-0,52
2	Bogor	133,74	0,19	28	Banda Aceh	124,87	-0,10
3	Sukabumi	131,74	0,03	29	Lhokseumawe	127,37	-0,45
4	Bandung	131,59	0,27	30	Sibolga	137,85	-0,64
5	Cirebon	128,01	-0,08	31	Pematangsiantar	137,10	0,56
6	Bekasi	129,10	-0,36	32	Medan	137,65	-0,01
7	Depok	130,48	-0,15	33	Padangsidimpuan	131,17	0,76
8	Tasikmalaya	130,83	-0,10	34	Padang	137,20	0,01
9	Cilacap	135,43	-0,11	35	Bukittinggi	129,21	0,12
10	Purwokerto	129,27	0,06	36	Tembilahan	137,98	0,17
11	Kudus	138,91	0,01	37	Pekanbaru	134,61	0,20
12	Surakarta	127,73	-0,02	38	Dumai	134,24	0,14
13	Semarang	130,74	0,02	39	Bungo	130,44	0,18
14	Tegal	128,61	-0,01	40	Jambi	131,81	0,21
15	Yogyakarta	129,36	0,10	41	Palembang	130,10	0,29
16	Jember	128,23	0,40	42	Lubuklinggau	130,56	0,29

17	Banyuwangi	127,67	0,04	43	Bengkulu	141,69	0,26
18	Sumenep	128,09	-0,02	44	Bandar Lampung	133,39	-0,01
19	Kediri	127,59	0,14	45	Metro	138,18	-0,12
20	Malang	132,57	0,14	46	Tanjung Pandan	139,32	-0,27
21	Probolinggo	127,86	0,21	47	Pangkalpinang	139,31	1,01
22	Madiun	130,04	0,22	48	Batam	133,36	-0,32
23	Surabaya	132,61	0,20	49	Tanjung Pinang	130,28	-0,13
24	Tangerang	139,93	0,39				
25	Cilegon	138,82	-0,02				
26	Serang	141,94	0,16				

e) Perbandingan Indeks dan Tingkat Inflasi Mei 2018 Kota-kota di Pulau Jawa dan Pulau Sumatra

	Kota	IHK	Tingkat Inflasi (%)		Kota	IHK	Tingkat Inflasi (%)
1	Jakarta	132,82	0,45	27	Meulaboh	131,62	0,57
2	Bogor	133,79	0,04	28	Banda Aceh	125,77	0,72
3	Sukabumi	131,99	0,19	29	Lhokseumawe	128,25	0,69
4	Bandung	131,88	0,22	30	Sibolga	137,75	-0,07
5	Cirebon	128,21	0,16	31	Pematangsiantar	137,09	-0,01
6	Bekasi	129,81	0,55	32	Medan	136,46	-0,86
7	Depok	130,65	0,13	33	Padangsidimpuan	130,45	-0,55
8	Tasikmalaya	131,38	0,42	34	Padang	137,83	0,46
9	Cilacap	135,32	-0,08	35	Bukittinggi	128,71	-0,39
10	Purwokerto	129,28	0,01	36	Tembilahan	138,30	0,23
11	Kudus	139,11	0,14	37	Pekanbaru	134,58	-0,02
12	Surakarta	127,78	0,04	38	Dumai	134,45	0,16
13	Semarang	130,62	-0,09	39	Bungo	130,63	0,15
14	Tegal	128,92	0,24	40	Jambi	131,66	-0,11
15	Yogyakarta	129,46	0,08	41	Palembang	130,30	0,15
16	Jember	128,55	0,25	42	Lubuklinggau	130,65	0,07
17	Banyuwangi	127,83	0,13	43	Bengkulu	142,15	0,32
18	Sumenep	128,48	0,30	44	Bandar Lampung	133,32	-0,05
19	Kediri	127,37	-0,17	45	Metro	137,73	-0,33
20	Malang	132,96	0,29	46	Tanjung Pandan	139,76	0,32
21	Probolinggo	127,98	0,09	47	Pangkalpinang	137,93	-0,99
22	Madiun	130,19	0,12	48	Batam	133,50	0,10
23	Surabaya	132,83	0,17	49	Tanjung Pinang	130,94	0,51
24	Tangerang	139,95	0,01				
25	Cilegon	139,47	0,47				
26	Serang	142,17	0,16				

f) Perbandingan Indeks dan Tingkat Inflasi Juni 2018 Kota-kota di Pulau Jawa dan Pulau Sumatra

	Kota	IHK	Tingkat Inflasi (%)		Kota	IHK	Tingkat Inflasi (%)
1	Jakarta	133,46	0,48	27	Meulaboh	131,88	0,20
2	Bogor	134,66	0,65	28	Banda Aceh	126,83	0,84
3	Sukabumi	132,58	0,45	29	Lhokseumawe	129,66	1,10
4	Bandung	132,51	0,48	30	Sibolga	138,15	0,29
5	Cirebon	129,09	0,69	31	Pematangsiantar	137,23	0,10
6	Bekasi	130,03	0,17	32	Medan	136,47	0,01
7	Depok	131,55	0,69	33	Padangsidimpuan	130,95	0,38
8	Tasikmalaya	132,15	0,59	34	Padang	138,37	0,39
9	Cilacap	136,35	0,76	35	Bukittinggi	128,97	0,20
10	Purwokerto	130,53	0,97	36	Tembilahan	138,45	0,11
11	Kudus	139,55	0,32	37	Pekanbaru	134,60	0,01
12	Surakarta	128,86	0,85	38	Dumai	135,33	0,65
13	Semarang	131,45	0,64	39	Bungo	131,04	0,31
14	Tegal	130,17	0,97	40	Jambi	133,52	1,41
15	Yogyakarta	130,05	0,46	41	Palembang	131,15	0,65
16	Jember	129,50	0,74	42	Lubuklinggau	130,90	0,19
17	Banyuwangi	128,47	0,50	43	Bengkulu	143,30	0,81
18	Sumenep	129,56	0,84	44	Bandar Lampung	134,62	0,98
19	Kediri	127,92	0,43	45	Metro	138,44	0,52
20	Malang	133,29	0,25	46	Tanjung Pandan	141,55	1,28
21	Probolinggo	128,92	0,73	47	Pangkalpinang	140,44	1,82
22	Madiun	131,14	0,73	48	Batam	135,22	1,29
23	Surabaya	133,33	0,38	49	Tanjung Pinang	131,26	0,24
24	Tangerang	140,34	0,28				
25	Cilegon	140,46	0,71				
26	Serang	142,91	0,52				

g) Perbandingan Indeks dan Tingkat Inflasi Juli 2018 Kota-kota di Pulau Jawa dan Pulau Sumatra

	Kota	IHK	Tingkat Inflasi (%)		Kota	IHK	Tingkat Inflasi (%)
1	Jakarta	133,46	0,48	27	Meulaboh	132,81	0,71
2	Bogor	134,66	0,65	28	Banda Aceh	126,93	0,08
3	Sukabumi	132,58	0,45	29	Lhokseumawe	130,10	0,34
4	Bandung	132,51	0,48	30	Sibolga	139,00	0,62
5	Cirebon	129,09	0,69	31	Pematangsiantar	137,71	0,35

6	Bekasi	130,03	0,17	32	Medan	137,14	0,49
7	Depok	131,55	0,69	33	Padangsidimpuan	131,64	0,53
8	Tasikmalaya	132,15	0,59	34	Padang	139,23	0,62
9	Cilacap	136,35	0,76	35	Bukittinggi	129,08	0,09
10	Purwokerto	130,53	0,97	36	Tembilahan	138,76	0,22
11	Kudus	139,55	0,32	37	Pekanbaru	135,13	0,39
12	Surakarta	128,86	0,85	38	Dumai	135,19	-0,10
13	Semarang	131,45	0,64	39	Bungo	131,20	0,12
14	Tegal	130,17	0,97	40	Jambi	132,51	-0,76
15	Yogyakarta	130,05	0,46	41	Palembang	131,14	-0,01
16	Jember	129,50	0,74	42	Lubuklinggau	131,27	0,28
17	Banyuwangi	128,47	0,50	43	Bengkulu	144,55	0,87
18	Sumenep	129,56	0,84	44	Bandar Lampung	134,56	-0,04
19	Kediri	127,92	0,43	45	Metro	138,37	-0,05
20	Malang	133,29	0,25	46	Tanjung Pandan	142,75	0,85
21	Probolinggo	128,92	0,73	47	Pangkalpinang	140,92	0,34
22	Madiun	131,14	0,73	48	Batam	135,54	0,24
23	Surabaya	133,33	0,38	49	Tanjung Pinang	131,85	0,45
24	Tangerang	140,34	0,28				
25	Cilegon	140,46	0,71				
26	Serang	142,91	0,52				

h) Perbandingan Indeks dan Tingkat Inflasi Agustus 2018 Kota-kota di Pulau Jawa dan Pulau Sumatra

	Kota	IHK	Tingkat Inflasi (%)		Kota	IHK	Tingkat Inflasi (%)
1	Jakarta	133,85	0,03	27	Meulaboh	133,11	0,23
2	Bogor	135,17	-0,10	28	Banda Aceh	127,56	0,50
3	Sukabumi	133,17	-0,10	29	Lhokseumawe	129,88	-0,17
4	Bandung	132,71	-0,02	30	Sibolga	138,48	-0,37
5	Cirebon	128,94	-0,32	31	Pematangsiantar	137,80	0,07
6	Bekasi	130,95	-0,15	32	Medan	137,15	0,01
7	Depok	132,14	0,42	33	Padangsidimpuan	131,65	0,01
8	Tasikmalaya	131,84	-0,37	34	Padang	138,67	-0,40
9	Cilacap	136,30	-0,12	35	Bukittinggi	128,91	-0,13
10	Purwokerto	130,41	-0,17	36	Tembilahan	138,03	-0,53
11	Kudus	139,54	-0,11	37	Pekanbaru	135,38	0,19
12	Surakarta	128,22	-0,58	38	Dumai	134,73	-0,34
13	Semarang	131,45	-0,11	39	Bungo	131,24	0,03
14	Tegal	129,96	-0,22	40	Jambi	132,61	0,08
15	Yogyakarta	130,44	-0,26	41	Palembang	130,92	-0,17
16	Jember	129,38	-0,01	42	Lubuklinggau	131,36	0,07

17	Banyuwangi	128,45	-0,05	43	Bengkulu	141,95	-1,80
18	Sumenep	129,39	-0,19	44	Bandar Lampung	134,63	0,05
19	Kediri	127,91	-0,10	45	Metro	138,65	0,20
20	Malang	133,64	0,05	46	Tanjung Pandan	143,46	0,50
21	Probolinggo	128,55	-0,35	47	Pangkalpinang	139,47	-1,03
22	Madiun	131,25	-0,08	48	Batam	134,65	-0,66
23	Surabaya	133,68	0,23	49	Tanjung Pinang	132,15	0,23
24	Tangerang	141,55	0,30				
25	Cilegon	140,49	-0,22				
26	Serang	143,81	0,07				

i) Perbandingan Indeks dan Tingkat Inflasi September 2018 Kota-kota di Pulau Jawa dan Pulau Sumatra

	Kota	IHK	Tingkat Inflasi (%)		Kota	IHK	Tingkat Inflasi (%)
1	Jakarta	133,68	-0,13	27	Meulaboh	132,57	-0,41
2	Bogor	134,82	-0,26	28	Banda Aceh	126,60	-0,75
3	Sukabumi	132,77	-0,30	29	Lhokseumawe	128,78	-0,85
4	Bandung	132,39	-0,24	30	Sibolga	139,02	0,39
5	Cirebon	128,59	-0,27	31	Pematangsiantar	137,47	-0,24
6	Bekasi	130,86	-0,07	32	Medan	137,28	0,09
7	Depok	131,95	-0,14	33	Padangsidimpuan	131,70	0,04
8	Tasikmalaya	131,49	-0,27	34	Padang	138,18	-0,35
9	Cilacap	136,12	-0,13	35	Bukittinggi	129,04	0,10
10	Purwokerto	130,30	-0,08	36	Tembilahan	136,99	-0,75
11	Kudus	139,44	-0,07	37	Pekanbaru	135,10	-0,21
12	Surakarta	127,98	-0,19	38	Dumai	134,38	-0,26
13	Semarang	131,57	0,09	39	Bungo	131,25	0,01
14	Tegal	129,95	-0,01	40	Jambi	131,91	-0,53
15	Yogyakarta	130,29	-0,11	41	Palembang	130,39	-0,40
16	Jember	129,32	-0,05	42	Lubuklinggau	130,98	-0,29
17	Banyuwangi	127,82	-0,49	43	Bengkulu	142,79	0,59
18	Sumenep	129,41	0,02	44	Bandar Lampung	134,36	-0,20
19	Kediri	128,17	0,20	45	Metro	138,39	-0,19
20	Malang	133,22	-0,31	46	Tanjung Pandan	141,85	-1,12
21	Probolinggo	128,14	-0,32	47	Pangkalpinang	139,54	0,05
22	Madiun	131,09	-0,12	48	Batam	134,53	-0,09
23	Surabaya	133,88	0,15	49	Tanjung Pinang	131,98	-0,13
24	Tangerang	141,64	0,06				
25	Cilegon	140,30	-0,14				
26	Serang	143,51	-0,21				

j) Perbandingan Indeks dan Tingkat Inflasi Oktober 2018 Kota-kota di Pulau Jawa dan Pulau Sumatra

	Kota	IHK	Tingkat Inflasi (%)		Kota	IHK	Tingkat Inflasi (%)
1	Jakarta	134,05	0,28	27	Meulaboh	132,35	-0,17
2	Bogor	135,14	0,24	28	Banda Aceh	127,01	0,32
3	Sukabumi	132,93	0,12	29	Lhokseumawe	129,43	0,50
4	Bandung	133,05	0,50	30	Sibolga	140,75	1,24
5	Cirebon	128,75	0,12	31	Pematangsiantar	138,57	0,80
6	Bekasi	131,07	0,16	32	Medan	139,26	1,44
7	Depok	132,38	0,33	33	Padangsidimpuan	131,84	0,11
8	Tasikmalaya	131,56	0,05	34	Padang	139,28	0,80
9	Cilacap	136,68	0,41	35	Bukittinggi	130,23	0,92
10	Purwokerto	130,76	0,35	36	Tembilahan	136,94	-0,04
11	Kudus	139,84	0,29	37	Pekanbaru	135,72	0,46
12	Surakarta	128,29	0,24	38	Dumai	135,05	0,50
13	Semarang	131,94	0,28	39	Bungo	131,92	0,51
14	Tegal	130,40	0,35	40	Jambi	133,07	0,88
15	Yogyakarta	130,46	0,13	41	Palembang	130,57	0,14
16	Jember	129,63	0,24	42	Lubuklinggau	131,02	0,03
17	Banyuwangi	127,93	0,09	43	Bengkulu	141,73	-0,74
18	Sumenep	129,80	0,30	44	Bandar Lampung	134,39	0,02
19	Kediri	128,38	0,16	45	Metro	138,69	0,22
20	Malang	133,62	0,30	46	Tanjung Pandan	142,70	0,60
21	Probolinggo	128,39	0,20	47	Pangkalpinang	139,07	-0,34
22	Madiun	131,32	0,18	48	Batam	134,71	0,13
23	Surabaya	134,08	0,15	49	Tanjung Pinang	132,36	0,29
24	Tangerang	141,63	-0,01				
25	Cilegon	140,32	0,01				
26	Serang	143,59	0,06				

k) Perbandingan Indeks dan Tingkat Inflasi November 2018 Kota-kota di Pulau Jawa dan Pulau Sumatra

	Kota	IHK	Tingkat Inflasi (%)		Kota	IHK	Tingkat Inflasi (%)
1	Jakarta	134,45	0,30	27	Meulaboh	132,45	0,08
2	Bogor	135,67	0,39	28	Banda Aceh	128,18	0,92
3	Sukabumi	133,36	0,32	29	Lhokseumawe	129,80	0,29
4	Bandung	133,53	0,36	30	Sibolga	140,36	-0,28
5	Cirebon	129,23	0,37	31	Pematangsiantar	138,56	-0,01

6	Bekasi	131,35	0,21	32	Medan	138,37	-0,64
7	Depok	132,64	0,20	33	Padangsidimpuan	132,50	0,50
8	Tasikmalaya	131,90	0,26	34	Padang	139,55	0,19
9	Cilacap	137,11	0,31	35	Bukittinggi	131,31	0,83
10	Purwokerto	131,18	0,32	36	Tembilahan	138,04	0,80
11	Kudus	140,25	0,29	37	Pekanbaru	136,29	0,42
12	Surakarta	128,57	0,22	38	Dumai	136,00	0,70
13	Semarang	132,22	0,21	39	Bungo	132,62	0,53
14	Tegal	130,74	0,26	40	Jambi	133,27	0,15
15	Yogyakarta	131,06	0,46	41	Palembang	130,84	0,21
16	Jember	129,98	0,27	42	Lubuklinggau	131,35	0,25
17	Banyuwangi	128,26	0,26	43	Bengkulu	142,01	0,20
18	Sumenep	130,11	0,24	44	Bandar Lampung	134,73	0,25
19	Kediri	128,89	0,40	45	Metro	139,06	0,27
20	Malang	134,12	0,37	46	Tanjung Pandan	142,16	-0,38
21	Probolinggo	128,84	0,35	47	Pangkalpinang	139,06	-0,01
22	Madiun	131,76	0,34	48	Batam	135,40	0,51
23	Surabaya	134,36	0,21	49	Tanjung Pinang	132,22	-0,11
24	Tangerang	142,18	0,39				
25	Cilegon	140,81	0,35				
26	Serang	144,26	0,47				

l) Perbandingan Indeks dan Tingkat Inflasi Desember 2018 Kota-kota di Pulau Jawa dan Pulau Sumatra

	Kota	IHK	Tingkat Inflasi (%)		Kota	IHK	Tingkat Inflasi (%)
1	Jakarta	135,25	0,60	27	Meulaboh	133,08	0,48
2	Bogor	136,73	0,78	28	Banda Aceh	128,20	0,02
3	Sukabumi	134,06	0,52	29	Lhokseumawe	131,16	1,05
4	Bandung	134,48	0,71	30	Sibolga	140,50	0,10
5	Cirebon	129,98	0,58	31	Pematangsiantar	139,09	0,38
6	Bekasi	132,13	0,59	32	Medan	138,53	0,12
7	Depok	132,93	0,22	33	Padangsidimpuan	133,04	0,41
8	Tasikmalaya	132,23	0,25	34	Padang	139,77	0,16
9	Cilacap	137,73	0,45	35	Bukittinggi	131,85	0,41
10	Purwokerto	131,87	0,53	36	Tembilahan	139,00	0,70
11	Kudus	140,92	0,48	37	Pekanbaru	136,54	0,18
12	Surakarta	129,30	0,57	38	Dumai	136,30	0,22
13	Semarang	132,70	0,36	39	Bungo	132,83	0,16
14	Tegal	131,35	0,47	40	Jambi	134,57	0,98
15	Yogyakarta	131,81	0,57	41	Palembang	132,09	0,96
16	Jember	130,62	0,49	42	Lubuklinggau	131,80	0,34

17	Banyuwangi	128,96	0,55	43	Bengkulu	143,13	0,79
18	Sumenep	130,78	0,51	44	Bandar Lampung	135,15	0,31
19	Kediri	129,77	0,29	45	Metro	139,44	0,27
20	Malang	134,99	0,65	46	Tanjung Pandan	143,35	0,84
21	Probolinggo	129,77	0,72	47	Pangkalpinang	141,67	1,88
22	Madiun	132,09	0,25	48	Batam	137,03	1,20
23	Surabaya	135,24	0,65	49	Tanjung Pinang	133,34	0,85
24	Tangerang	143,08	0,63				
25	Cilegon	141,62	0,58				
26	Serang	145,19	0,64				

B. *Script* dan *Output* Program R Data Bangkitan

```
> #Bangkitkan data
> set.seed(1)
> x = rnorm(49, 4, 2)
> set.seed(2)
> e = rnorm(49)
> set.seed(3)
> y = 1.2 + 2*x + e
> #Menentukan Confidence Interval
> f.bang = function (y,x,B,a){
+ n = length(y)
+ b0 = matrix(coef(lm(y~x)), nrow = 1, ncol = 2)
+ yfit = b0[,1] + b0[,2]*x
+ e = seq(1:n)
+ for(i in 1:n) {
+ e[i] = y[i] - yfit[i]}
+ estar = matrix(0, nrow = B, ncol = n)
+ ystar = matrix(0, nrow = B, ncol = n)
+ b.r = matrix(0, nrow = B, ncol = 2)
+ for (i in 1:B) {
+ estar[i,] = sample(e, n, replace = T)
+ ystar[i,] = b0[,1] + b0[,2]*x + estar[i,]
+ b.r[i,] = coef(lm(ystar[i,]~x))}
+ bboot = apply(b.r,2,mean)
+ lo0_r = b0[,1] + qnorm(a/2)*sqrt(var(b.r[,1]))
+ lo1_r = b0[,2] + qnorm(a/2)*sqrt(var(b.r[,2]))
+ up0_r = b0[,1] + qnorm(1-a/2)*sqrt(var(b.r[,1]))
+ up1_r = b0[,2] + qnorm(1-a/2)*sqrt(var(b.r[,2]))
+ int1_r = matrix(c(lo0_r, up0_r, lo1_r, up1_r),
+ nrow = 2, byrow = T)
+ lop0_r = quantile(b.r[,1], a/2)
+ lop1_r = quantile(b.r[,2], a/2)
+ upp0_r = quantile(b.r[,1], 1-a/2)
+ upp1_r = quantile(b.r[,2], 1-a/2)
+ int2_r = matrix(c(lop0_r, upp0_r, lop1_r, upp1_r),
+ nrow = 2, byrow = T)
```

```
+ data = data.frame(x,y)
+ b.o = matrix(0, nrow = B, ncol = 2)
+ for (i in 1:B){
+ data.obs = sample(1:nrow(data), n, replace = T)
+ obs = data[data.obs,]
+ b.o[i,] = coef(lm(obs$y~obs$x))}
+ b1boot = apply(b.o,2,mean)
+ lo0_o = b0[,1] + qnorm(a/2)*sqrt(var(b.o[,1]))
+ lo1_o = b0[,2] + qnorm(a/2)*sqrt(var(b.o[,2]))
+ up0_o = b0[,1] + qnorm(1-a/2)*sqrt(var(b.o[,1]))
+ up1_o = b0[,2] + qnorm(1-a/2)*sqrt(var(b.o[,2]))
+ int1_o = matrix(c(lo0_o, up0_o, lo1_o, up1_o),
+ nrow = 2, byrow = T)
+ lop0_o = quantile(b.o[,1], a/2)
+ lop1_o = quantile(b.o[,2], a/2)
+ upp0_o = quantile(b.o[,1], 1-a/2)
+ upp1_o = quantile(b.o[,2], 1-a/2)
+ int2_o = matrix(c(lop0_o, upp0_o, lop1_o, upp1_o),
+ nrow = 2, byrow = T)
+ list(est.boot_r = bboot, int.normal_r = int1_r,
+ int.pct_r = int2_r, est.boot_o = b1boot,
+ int.normal_o = int1_o, int.pct_o = int2_o)}
> #Bootstrap - 100
> set.seed(100)
> f.bang(y,x,100,0.05)
```

```
$est.boot_r                    $est.boot_o
[1] 0.9578797 2.0817368        [1] 0.8963212 2.0928241

$int.normal_r                  $int.normal_o
          [,1]     [,2]                  [,1]     [,2]
[1,] 0.1786758 1.827947        [1,] 0.02489533 1.981728
[2,] 1.8894253 2.250512        [2,] 1.86907162 2.270866

$int.pct_r                     $int.pct_o
           [,1]     [,2]                 [,1]     [,2]
[1,] 0.04285087 1.658629       [1,] -0.144493 1.760407
[2,] 1.93881115 2.279137       [2,]  1.912977 2.313714
```

```
> #Bootstrap - 500
> set.seed(500)
> f.bang(y,x,500,0.05)
```

```
$est.boot_r                    $est.boot_o
[1] 0.9763194 2.0748097        [1] 0.9949021 2.0750329

$int.normal_r                  $int.normal_o
          [,1]     [,2]                 [,1]     [,2]
[1,] 0.1350196 1.871603        [1,] 0.0296197 1.977003
[2,] 1.8741041 2.265833        [2,] 1.8610383 2.278899

$int.pct_r                     $int.pct_o
           [,1]     [,2]                  [,1]     [,2]
[1,] 0.07751839 1.819320       [1,] 0.06700264 1.910858
[2,] 1.88875533 2.280418       [2,] 1.86354784 2.265903
```

```
> #Bootstrap - 1000
> set.seed(1000)
> f.bang(y,x,1000,0.05)
$est.boot_r
[1] 1.004059 2.070376

$int.normal_r
          [,1]     [,2]
[1,] 0.1520895 1.854533
[2,] 1.8834602 2.256477

$int.pct_r
          [,1]     [,2]
[1,] 0.1345273 1.823904
[2,] 1.8909342 2.255950
```

```
$est.boot_o
[1] 0.976572 2.075287

$int.normal_o
           [,1]     [,2]
[1,] 0.00128502 2.005338
[2,] 1.85853308 2.281404

$int.pct_o
           [,1]     [,2]
[1,] 0.01333235 1.956484
[2,] 1.87142551 2.287152
```

```
> #Bootstrap - 3000
> set.seed(3000)
> f.bang(y,x,3000,0.05)
$est.boot_r
[1] 1.001853 2.070168

$int.normal_r
          [,1]     [,2]
[1,] 0.1671317 1.839491
[2,] 1.8840876 2.255850

$int.pct_r
          [,1]     [,2]
[1,] 0.1478577 1.861131
[2,] 1.8786893 2.257095
```

```
$est.boot_o
[1] 0.9784355 2.0736612

$int.normal_o
           [,1]     [,2]
[1,] 0.01999327 1.986630
[2,] 1.85899830 2.280939

$int.pct_o
          [,1]     [,2]
[1,] 0.0247076 1.956388
[2,] 1.8698215 2.285043
```

C. *Script* dan *Output* Program R Data IHK dan Inflasi untuk Menentukan *Confidence Interval* Bootstrap

a. Januari

```
> data1 = read.csv(file.choose(), header = T, sep = ",")
> model1 = lm(Inflasi~IHK, data = data1)
> summary(model1)

Call:
lm(formula = Inflasi ~ IHK, data = data1)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.86093 -0.27944 -0.00199  0.22736  0.74000 
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.85846    1.65171  -1.125    0.266
IHK          0.01906    0.01247   1.528    0.133

Residual standard error: 0.3711 on 47 degrees of freedom
Multiple R-squared:  0.04734,   Adjusted R-squared:  0.02707
F-statistic: 2.336 on 1 and 47 DF,  p-value: 0.1331
> x1 = data1[,1]
> y1 = data1[,2]
> freg.j = function (y1,x1,B,a){
+ n = length(y1)
+ b01 = matrix(coef(lm(y1~x1)), nrow = 1, ncol = 2)
+ y1fit = b01[,1] + b01[,2]*x1
+ e1 = seq(1:n)
+ for(i in 1:n) {
+ e1[i] = y1[i] - y1fit[i]}
+ e1star = matrix(0, nrow = B, ncol = n)
+ y1star = matrix(0, nrow = B, ncol = n)
+ b1.r = matrix(0, nrow = B, ncol = 2)
+ for (i in 1:B) {
+ e1star[i,] = sample(e1, n, replace = T)
+ y1star[i,] = b01[,1] + b01[,2]*x1 + e1star[i,]
+ b1.r[i,] = coef(lm(y1star[i,]~x1))}
+ b1boot = apply(b1.r,2,mean)
+ lo0_r1 = b01[,1] + qnorm(a/2)*sqrt(var(b1.r[,1]))
+ lo1_r1 = b01[,2] + qnorm(a/2)*sqrt(var(b1.r[,2]))
+ up0_r1 = b01[,1] + qnorm(1-a/2)*sqrt(var(b1.r[,1]))
+ up1_r1 = b01[,2] + qnorm(1-a/2)*sqrt(var(b1.r[,2]))
+ int1_r1 = matrix(c(lo0_r1, up0_r1, lo1_r1, up1_r1),
+ nrow = 2, byrow = T)
+ lop0_r1 = quantile(b1.r[,1], a/2)
+ lop1_r1 = quantile(b1.r[,2], a/2)
+ upp0_r1 = quantile(b1.r[,1], 1-a/2)
+ upp1_r1 = quantile(b1.r[,2], 1-a/2)
+ int2_r1 = matrix(c(lop0_r1, upp0_r1, lop1_r1, upp1_r1),
+ nrow = 2, byrow = T)
> list(est.boot_r1 = b1boot, int.normal_r1 = int1_r1,
+ int.pct_r1 = int2_r1)}
> #Bootstrap - 100
> set.seed(101)
> freg.j(y1,x1,100,0.05)
$est.boot_r1
[1] -1.54489636  0.01674256

$int.normal_r1
             [,1]       [,2]
[1,] -5.310950689 1.59403327
[2,] -0.007017616 0.04514114

$int.pct_r1
             [,1]       [,2]
[1,] -4.769210657 1.75231692
[2,] -0.007986675 0.04131918
```

b. Februari

```
> data2 = read.csv(file.choose(), header = T, sep = ",")
> model2 = lm(Inflasi~IHK, data = data2)
> summary(model2)

Call:
lm(formula = Inflasi ~ IHK, data = data2)

Residuals:
     Min       1Q   Median       3Q      Max
-0.87404 -0.20880  0.07569  0.25411  0.69433

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.55201    1.69915   1.502     0.14
IHK         -0.01928    0.01283  -1.503     0.14

Residual standard error: 0.3698 on 47 degrees of freedom
Multiple R-squared:  0.04583,   Adjusted R-squared:  0.02553
F-statistic: 2.258 on 1 and 47 DF,  p-value: 0.1396

> x2 = data2[,1]
> y2 = data2[,2]
> freg.f = function (y2,x2,B,a){
+ n = length(y2)
+ b02 = matrix(coef(lm(y2~x2)), nrow = 1, ncol = 2)
+ y2fit = b02[,1] + b02[,2]*x2
+ e2 = seq(1:n)
+ for(i in 1:n) {
+ e2[i] = y2[i] - y2fit[i]}
+ e2star = matrix(0, nrow = B, ncol = n)
+ y2star = matrix(0, nrow = B, ncol = n)
+ b2.r = matrix(0, nrow = B, ncol = 2)
+ for (i in 1:B) {
+ e2star[i,] = sample(e2, n, replace = T)
+ y2star[i,] = b02[,1] + b02[,2]*x2 + e2star[i,]
+ b2.r[i,] = coef(lm(y2star[i,]~x2))}
+ b2boot = apply(b2.r,2,mean)
+ lo0_r2 = b02[,1] + qnorm(a/2)*sqrt(var(b2.r[,1]))
+ lo1_r2 = b02[,2] + qnorm(a/2)*sqrt(var(b2.r[,2]))
+ up0_r2 = b02[,1] + qnorm(1-a/2)*sqrt(var(b2.r[,1]))
+ up1_r2 = b02[,2] + qnorm(1-a/2)*sqrt(var(b2.r[,2]))
+ int1_r2 = matrix(c(lo0_r2, up0_r2, lo1_r2, up1_r2),
+ nrow = 2, byrow = T)
+ lop0_r2 = quantile(b2.r[,1], a/2)
+ lop1_r2 = quantile(b2.r[,2], a/2)
+ upp0_r2 = quantile(b2.r[,1], 1-a/2)
+ upp1_r2 = quantile(b2.r[,2], 1-a/2)
+ int2_r2 = matrix(c(lop0_r2, upp0_r2, lop1_r2, upp1_r2),
+ nrow = 2, byrow = T)
> list(est.boot_r2 = b2boot, int.normal_r2 = int1_r2,
+ int.pct_r2 = int2_r2)}
```

```
> #Bootstrap - 100
> set.seed(201)
> freg.f(y2,x2,100,0.05)
$est.boot_r2
[1]  2.62265267 -0.01978848

$int.normal_r2
            [,1]        [,2]
[1,] -0.92213599 6.026164864
[2,] -0.04540934 0.006848086

$int.pct_r2
            [,1]        [,2]
[1,] -0.78629321 5.571252817
[2,] -0.04224472 0.005735406
```

c. Maret

```
> data3 = read.csv(file.choose(), header = T, sep = ",")
> model3 = lm(Inflasi~IHK, data = data3)
> summary(model3)

Call:
lm(formula = Inflasi ~ IHK, data = data3)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.52087 -0.16427 -0.03855  0.10236  1.04984 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)   
(Intercept) -3.681594   1.252729  -2.939  0.00509 **
IHK          0.029123   0.009443   3.084  0.00341 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2825 on 47 degrees of freedom
Multiple R-squared:  0.1683,	Adjusted R-squared:  0.1506 
F-statistic: 9.512 on 1 and 47 DF,  p-value: 0.003414

> x3 = data3[,1]
> y3 = data3[,2]
> freg.m = function (y3,x3,B,a){
+ n = length(y3)
+ b03 = matrix(coef(lm(y3~x3)), nrow = 1, ncol = 2)
+ y3fit = b03[,1] + b03[,2]*x3
+ e3 = seq(1:n)
+ for(i in 1:n) {
+ e3[i] = y3[i] - y3fit[i]}
+ e3star = matrix(0, nrow = B, ncol = n)
+ y3star = matrix(0, nrow = B, ncol = n)
+ b3.r = matrix(0, nrow = B, ncol = 2)
+ for (i in 1:B) {
+ e3star[i,] = sample(e3, n, replace = T)
+ y3star[i,] = b03[,1] + b03[,2]*x3 + e3star[i,]
+ b3.r[i,] = coef(lm(y3star[i,]~x3))}
+ b3boot = apply(b3.r,2,mean)
```

```
+ lo0_r3 = b03[,1] + qnorm(a/2)*sqrt(var(b3.r[,1]))
+ lo1_r3 = b03[,2] + qnorm(a/2)*sqrt(var(b3.r[,2]))
+ up0_r3 = b03[,1] + qnorm(1-a/2)*sqrt(var(b3.r[,1]))
+ up1_r3 = b03[,2] + qnorm(1-a/2)*sqrt(var(b3.r[,2]))
+ int1_r3 = matrix(c(lo0_r3, up0_r3, lo1_r3, up1_r3),
+ nrow = 2, byrow = T)
+ lop0_r3 = quantile(b3.r[,1], a/2)
+ lop1_r3 = quantile(b3.r[,2], a/2)
+ upp0_r3 = quantile(b3.r[,1], 1-a/2)
+ upp1_r3 = quantile(b3.r[,2], 1-a/2)
+ int2_r3 = matrix(c(lop0_r3, upp0_r3, lop1_r3, upp1_r3),
+ nrow = 2, byrow = T)
> list(est.boot_r3 = b3boot, int.normal_r3 = int1_r3,
+ int.pct_r3 = int2_r3)}
> #Bootstrap - 100
> set.seed(301)
> freg.m(y3,x3,100,0.05)
$est.boot_r3
[1] -3.65513877  0.02890555

$int.normal_r3                       $int.pct_r3
            [,1]        [,2]                     [,1]        [,2]
[1,] -6.04558245 -1.31760456         [1,] -5.92548042 -1.24231157
[2,]  0.01131976  0.04692718         [2,]  0.01060133  0.04603024
```

d. April

```
> data4 = read.csv(file.choose(), header = T, sep = ",")
> model4 = lm(Inflasi~IHK, data = data4)
> summary(model4)

Call:
lm(formula = Inflasi ~ IHK, data = data4)

Residuals:
     Min       1Q   Median       3Q      Max
-0.76247 -0.12329 -0.00421  0.13099  0.87263

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.284121   1.267124  -1.013    0.316
IHK          0.010204   0.009545   1.069    0.291

Residual standard error: 0.2885 on 47 degrees of freedom
Multiple R-squared:  0.02374,   Adjusted R-squared:  0.002968
F-statistic: 1.143 on 1 and 47 DF,  p-value: 0.2905

> x4 = data4[,1]
> y4 = data4[,2]
> freg.a = function (y4,x4,B,a){
+ n = length(y4)
+ b04 = matrix(coef(lm(y4~x4)), nrow = 1, ncol = 2)
+ y4fit = b04[,1] + b04[,2]*x4
```

```
+ e4 = seq(1:n)
+ for(i in 1:n) {
+ e4[i] = y4[i] - y4fit[i]}
+ e4star = matrix(0, nrow = B, ncol = n)
+ y4star = matrix(0, nrow = B, ncol = n)
+ b4.r = matrix(0, nrow = B, ncol = 2)
+ for (i in 1:B) {
+ e4star[i,] = sample(e4, n, replace = T)
+ y4star[i,] = b04[,1] + b04[,2]*x4 + e4star[i,]
+ b4.r[i,] = coef(lm(y4star[i,]~x4))}
+ b4boot = apply(b4.r,2,mean)
+ lo0_r4 = b04[,1] + qnorm(a/2)*sqrt(var(b4.r[,1]))
+ lo1_r4 = b04[,2] + qnorm(a/2)*sqrt(var(b4.r[,2]))
+ up0_r4 = b04[,1] + qnorm(1-a/2)*sqrt(var(b4.r[,1]))
+ up1_r4 = b04[,2] + qnorm(1-a/2)*sqrt(var(b4.r[,2]))
+ int1_r4 = matrix(c(lo0_r4, up0_r4, lo1_r4, up1_r4),
+ nrow = 2, byrow = T)
+ lop0_r4 = quantile(b4.r[,1], a/2)
+ lop1_r4 = quantile(b4.r[,2], a/2)
+ upp0_r4 = quantile(b4.r[,1], 1-a/2)
+ upp1_r4 = quantile(b4.r[,2], 1-a/2)
+ int2_r4 = matrix(c(lop0_r4, upp0_r4, lop1_r4, upp1_r4),
+ nrow = 2, byrow = T)
> list(est.boot_r4 = b4boot, int.normal_r4 = int1_r4,
+ int.pct_r4 = int2_r4)}
> #Bootstrap - 100
> set.seed(401)
> freg.a(y4,x4,100,0.05)
$est.boot_r4
[1] -1.176538230  0.009326184

$int.normal_r4
             [,1]       [,2]
[1,] -3.900271017 1.33202880
[2,] -0.009455595 0.02986322

$int.pct_r4
            [,1]       [,2]
[1,] -3.45077735 1.43557719
[2,] -0.01037787 0.02629518
```

e. Mei

```
> data5 = read.csv(file.choose(), header = T, sep = ",")
> model5 = lm(Inflasi~IHK, data = data5)
> summary(model5)
```

```
Call:
lm(formula = Inflasi ~ IHK, data = data5)

Residuals:
     Min       1Q   Median       3Q      Max
-1.03500 -0.11729 -0.00145  0.18167  0.52164

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.80275    1.49133   1.209    0.233
IHK         -0.01274    0.01122  -1.136    0.262

Residual standard error: 0.3324 on 47 degrees of freedom
Multiple R-squared:  0.02671,   Adjusted R-squared:  0.005999
F-statistic:  1.29 on 1 and 47 DF,  p-value: 0.2619
```

```
> x5 = data5[,1]
> y5 = data5[,2]
> freg.me = function (y5,x5,B,a){
+ n = length(y5)
+ b05 = matrix(coef(lm(y5~x5)), nrow = 1, ncol = 2)
+ y5fit = b05[,1] + b05[,2]*x5
+ e5 = seq(1:n)
+ for(i in 1:n) {
+ e5[i] = y5[i] - y5fit[i]}
+ e5star = matrix(0, nrow = B, ncol = n)
+ y5star = matrix(0, nrow = B, ncol = n)
+ b5.r = matrix(0, nrow = B, ncol = 2)
+ for (i in 1:B) {
+ e5star[i,] = sample(e5, n, replace = T)
+ y5star[i,] = b05[,1] + b05[,2]*x5 + e5star[i,]
+ b5.r[i,] = coef(lm(y5star[i,]~x5))}
+ b5boot = apply(b5.r,2,mean)
+ lo0_r5 = b05[,1] + qnorm(a/2)*sqrt(var(b5.r[,1]))
+ lo1_r5 = b05[,2] + qnorm(a/2)*sqrt(var(b5.r[,2]))
+ up0_r5 = b05[,1] + qnorm(1-a/2)*sqrt(var(b5.r[,1]))
+ up1_r5 = b05[,2] + qnorm(1-a/2)*sqrt(var(b5.r[,2]))
+ int1_r5 = matrix(c(lo0_r5, up0_r5, lo1_r5, up1_r5),
+ nrow = 2, byrow = T)
+ lop0_r5 = quantile(b5.r[,1], a/2)
+ lop1_r5 = quantile(b5.r[,2], a/2)
+ upp0_r5 = quantile(b5.r[,1], 1-a/2)
+ upp1_r5 = quantile(b5.r[,2], 1-a/2)
+ int2_r5 = matrix(c(lop0_r5, upp0_r5, lop1_r5, upp1_r5),
+ nrow = 2, byrow = T)
> list(est.boot_r5 = b5boot, int.normal_r5 = int1_r5,
+ int.pct_r5 = int2_r5)}
```

```
> #Bootstrap - 100
> set.seed(501)
> freg.me(y5,x5,100,0.05)
$est.boot_r5
[1]  1.70345320 -0.01195279

$int.normal_r5
             [,1]       [,2]
[1,] -1.22422062 4.8297248
[2,] -0.03551945 0.0100318

$int.pct_r5
             [,1]        [,2]
[1,] -0.95442408 4.995182868
[2,] -0.03687175 0.008202952
```

f. Juni

```
> data6 = read.csv(file.choose(), header = T, sep = ",")
> model6 = lm(Inflasi~IHK, data = data6)
> summary(model6)
Call:
lm(formula = Inflasi ~ IHK, data = data6)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.59618 -0.30034 -0.06941  0.16586  1.19556 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.021212   1.707640  -0.012    0.990
IHK          0.004597   0.012774   0.360    0.721

Residual standard error: 0.3805 on 47 degrees of freedom
Multiple R-squared:  0.002748,  Adjusted R-squared:  -0.01847 
F-statistic: 0.1295 on 1 and 47 DF,  p-value: 0.7205
> x6 = data6[,1]
> y6 = data6[,2]
> freg.jn = function (y6,x6,B,a){
+ n = length(y6)
+ b06 = matrix(coef(lm(y6~x6)), nrow = 1, ncol = 2)
+ y6fit = b06[,1] + b06[,2]*x6
+ e6 = seq(1:n)
+ for(i in 1:n) {
+ e6[i] = y6[i] - y6fit[i]}
+ e6star = matrix(0, nrow = B, ncol = n)
+ y6star = matrix(0, nrow = B, ncol = n)
+ b6.r = matrix(0, nrow = B, ncol = 2)
+ for (i in 1:B) {
+ e6star[i,] = sample(e6, n, replace = T)
+ y6star[i,] = b06[,1] + b06[,2]*x6 + e6star[i,]
+ b6.r[i,] = coef(lm(y6star[i,]~x6))}
+ b6boot = apply(b6.r,2,mean)
```

```
+ lo0_r6 = b06[,1] + qnorm(a/2)*sqrt(var(b6.r[,1]))
+ lo1_r6 = b06[,2] + qnorm(a/2)*sqrt(var(b6.r[,2]))
+ up0_r6 = b06[,1] + qnorm(1-a/2)*sqrt(var(b6.r[,1]))
+ up1_r6 = b06[,2] + qnorm(1-a/2)*sqrt(var(b6.r[,2]))
+ int1_r6 = matrix(c(lo0_r6, up0_r6, lo1_r6, up1_r6),
+ nrow = 2, byrow = T)
+ lop0_r6 = quantile(b6.r[,1], a/2)
+ lop1_r6 = quantile(b6.r[,2], a/2)
+ upp0_r6 = quantile(b6.r[,1], 1-a/2)
+ upp1_r6 = quantile(b6.r[,2], 1-a/2)
+ int2_r6 = matrix(c(lop0_r6, upp0_r6, lop1_r6, upp1_r6),
+ nrow = 2, byrow = T)
> list(est.boot_r6 = b6boot, int.normal_r6 = int1_r6,
+ int.pct_r6 = int2_r6)}
> #Bootstrap - 100
> set.seed(601)
> freg.jn(y6,x6,100,0.05)
$est.boot_r6
[1] -0.427376823  0.007662549

$int.normal_r6
            [,1]       [,2]
[1,] -3.30364958 3.26122514
[2,] -0.01999011 0.02918475

$int.pct_r6
            [,1]       [,2]
[1,] -3.28292521 2.85790312
[2,] -0.01709662 0.02893511
```

g. Juli

```
> data7 = read.csv(file.choose(), header = T, sep = ",")
> model7 = lm(Inflasi~IHK, data = data7)
> summary(model7)

Call:
lm(formula = Inflasi ~ IHK, data = data7)

Residuals:
     Min       1Q   Median       3Q      Max
-1.20491 -0.17845  0.03345  0.25100  0.52913

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.215793   1.431324   0.151    0.881
IHK         0.001729   0.010692   0.162    0.872

Residual standard error: 0.3301 on 47 degrees of freedom
Multiple R-squared:  0.0005562, Adjusted R-squared:  -0.02071
F-statistic: 0.02615 on 1 and 47 DF,  p-value: 0.8722

> x7 = data7[,1]
> y7 = data7[,2]
> freg.jl = function (y7,x7,B,a){
+ n = length(y7)
+ b07 = matrix(coef(lm(y7~x7)), nrow = 1, ncol = 2)
+ y7fit = b07[,1] + b07[,2]*x7
```

```
+ e7 = seq(1:n)
+ for(i in 1:n) {
+ e7[i] = y7[i] - y7fit[i]}
+ e7star = matrix(0, nrow = B, ncol = n)
+ y7star = matrix(0, nrow = B, ncol = n)
+ b7.r = matrix(0, nrow = B, ncol = 2)
+ for (i in 1:B) {
+ e7star[i,] = sample(e7, n, replace = T)
+ y7star[i,] = b07[,1] + b07[,2]*x7 + e7star[i,]
+ b7.r[i,] = coef(lm(y7star[i,]~x7))}
+ b7boot = apply(b7.r,2,mean)
+ lo0_r7 = b07[,1] + qnorm(a/2)*sqrt(var(b7.r[,1]))
+ lo1_r7 = b07[,2] + qnorm(a/2)*sqrt(var(b7.r[,2]))
+ up0_r7 = b07[,1] + qnorm(1-a/2)*sqrt(var(b7.r[,1]))
+ up1_r7 = b07[,2] + qnorm(1-a/2)*sqrt(var(b7.r[,2]))
+ int1_r7 = matrix(c(lo0_r7, up0_r7, lo1_r7, up1_r7),
+ nrow = 2, byrow = T)
+ lop0_r7 = quantile(b7.r[,1], a/2)
+ lop1_r7 = quantile(b7.r[,2], a/2)
+ upp0_r7 = quantile(b7.r[,1], 1-a/2)
+ upp1_r7 = quantile(b7.r[,2], 1-a/2)
+ int2_r7 = matrix(c(lop0_r7, upp0_r7, lop1_r7, upp1_r7),
+ nrow = 2, byrow = T)
> list(est.boot_r7 = b7boot, int.normal_r7 = int1_r7,
+ int.pct_r7 = int2_r7)}
> #Bootstrap - 100
> set.seed(701)
> freg.jl(y7,x7,100,0.05)
$est.boot_r7
[1] 0.227564036 0.001635997

$int.normal_r7
             [,1]       [,2]
[1,] -2.62475219 3.05633866
[2,] -0.01951012 0.02296826

$int.pct_r7
             [,1]       [,2]
[1,] -2.69180488 2.71446933
[2,] -0.01721316 0.02301541
```

h. Agustus

```
> data8 = read.csv(file.choose(), header = T, sep = ",")
> model8 = lm(Inflasi~IHK, data = data8)
> summary(model8)

Call:
lm(formula = Inflasi ~ IHK, data = data8)

Residuals:
     Min       1Q   Median       3Q      Max
-1.58149 -0.14681  0.03003  0.17979  0.73681
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.50189    1.67216   0.898    0.374
IHK         -0.01212    0.01249  -0.970    0.337

Residual standard error: 0.378 on 47 degrees of freedom
Multiple R-squared:  0.01963,   Adjusted R-squared:  -0.001225
F-statistic: 0.9413 on 1 and 47 DF,  p-value: 0.3369
> x8 = data8[,1]
> y8 = data8[,2]
> freg.ag = function (y8,x8,B,a){
+ n = length(y8)
+ b08 = matrix(coef(lm(y8~x8)), nrow = 1, ncol = 2)
+ y8fit = b08[,1] + b08[,2]*x8
+ e8 = seq(1:n)
+ for(i in 1:n) {
+ e8[i] = y8[i] - y8fit[i]}
+ e8star = matrix(0, nrow = B, ncol = n)
+ y8star = matrix(0, nrow = B, ncol = n)
+ b8.r = matrix(0, nrow = B, ncol = 2)
+ for (i in 1:B) {
+ e8star[i,] = sample(e8, n, replace = T)
+ y8star[i,] = b08[,1] + b08[,2]*x8 + e8star[i,]
+ b8.r[i,] = coef(lm(y8star[i,]~x8))}
+ b8boot = apply(b8.r,2,mean)
+ lo0_r8 = b08[,1] + qnorm(a/2)*sqrt(var(b8.r[,1]))
+ lo1_r8 = b08[,2] + qnorm(a/2)*sqrt(var(b8.r[,2]))
+ up0_r8 = b08[,1] + qnorm(1-a/2)*sqrt(var(b8.r[,1]))
+ up1_r8 = b08[,2] + qnorm(1-a/2)*sqrt(var(b8.r[,2]))
+ int1_r8 = matrix(c(lo0_r8, up0_r8, lo1_r8, up1_r8),
+ nrow = 2, byrow = T)
+ lop0_r8 = quantile(b8.r[,1], a/2)
+ lop1_r8 = quantile(b8.r[,2], a/2)
+ upp0_r8 = quantile(b8.r[,1], 1-a/2)
+ upp1_r8 = quantile(b8.r[,2], 1-a/2)
+ int2_r8 = matrix(c(lop0_r8, upp0_r8, lop1_r8, upp1_r8),
+ nrow = 2, byrow = T)
> list(est.boot_r8 = b8boot, int.normal_r8 = int1_r8,
+ int.pct_r8 = int2_r8)}
> #Bootstrap - 100
> set.seed(801)
> freg.ag(y8,x8,100,0.05)
$est.boot_r8
[1]  1.51030759 -0.01217487

$int.normal_r8                    $int.pct_r8
            [,1]       [,2]                   [,1]        [,2]
[1,] -1.76167532 4.76545408      [1,] -1.22266564 5.424107399
[2,] -0.03661271 0.01237321      [2,] -0.04174711 0.008019849
```

i. September

```
> data9 = read.csv(file.choose(), header = T, sep = ",")
> model9 = lm(Inflasi~IHK, data = data9)
> summary(model9)
Call:
lm(formula = Inflasi ~ IHK, data = data9)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.01897 -0.09656  0.02651  0.17235  0.68192 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.474553   1.292299  -1.141    0.260
IHK          0.009683   0.009672   1.001    0.322

Residual standard error: 0.2947 on 47 degrees of freedom
Multiple R-squared:  0.02088,   Adjusted R-squared:  4.878e-05 
F-statistic: 1.002 on 1 and 47 DF,  p-value: 0.3219
> x9 = data9[,1]
> y9 = data9[,2]
> freg.s = function (y9,x9,B,a){
+ n = length(y9)
+ b09 = matrix(coef(lm(y9~x9)), nrow = 1, ncol = 2)
+ y9fit = b09[,1] + b09[,2]*x9
+ e9 = seq(1:n)
+ for(i in 1:n) {
+ e9[i] = y9[i] - y9fit[i]}
+ e9star = matrix(0, nrow = B, ncol = n)
+ y9star = matrix(0, nrow = B, ncol = n)
+ b9.r = matrix(0, nrow = B, ncol = 2)
+ for (i in 1:B) {
+ e9star[i,] = sample(e9, n, replace = T)
+ y9star[i,] = b09[,1] + b09[,2]*x9 + e9star[i,]
+ b9.r[i,] = coef(lm(y9star[i,]~x9))}
+ b9boot = apply(b9.r,2,mean)
+ lo0_r9 = b09[,1] + qnorm(a/2)*sqrt(var(b9.r[,1]))
+ lo1_r9 = b09[,2] + qnorm(a/2)*sqrt(var(b9.r[,2]))
+ up0_r9 = b09[,1] + qnorm(1-a/2)*sqrt(var(b9.r[,1]))
+ up1_r9 = b09[,2] + qnorm(1-a/2)*sqrt(var(b9.r[,2]))
+ int1_r9 = matrix(c(lo0_r9, up0_r9, lo1_r9, up1_r9),
+ nrow = 2, byrow = T)
+ lop0_r9 = quantile(b9.r[,1], a/2)
+ lop1_r9 = quantile(b9.r[,2], a/2)
+ upp0_r9 = quantile(b9.r[,1], 1-a/2)
+ upp1_r9 = quantile(b9.r[,2], 1-a/2)
+ int2_r9 = matrix(c(lop0_r9, upp0_r9, lop1_r9, upp1_r9),
+ nrow = 2, byrow = T)
> list(est.boot_r9 = b9boot, int.normal_r9 = int1_r9,
+ int.pct_r9 = int2_r9)}
```

```
> #Bootstrap - 100
> set.seed(901)
> freg.s(y9,x9,100,0.05)
$est.boot_r9
[1] -1.71459700  0.01151431

$int.normal_r9                       $int.pct_r9
             [,1]        [,2]                     [,1]        [,2]
[1,] -3.886235612 0.93713040   [1,] -4.098420849 0.63973859
[2,] -0.008419587 0.02778548   [2,] -0.006107502 0.02924213
```

j. Oktober

```
> data10 = read.csv(file.choose(), header = T, sep = ",")
> model10 = lm(Inflasi~IHK, data = data10)
> summary(model10)

Call:
lm(formula = Inflasi ~ IHK, data = data10)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.07090 -0.16336 -0.03069  0.10638  1.12244 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.434745   1.602477  -0.271    0.787
IHK          0.005402   0.011959   0.452    0.654

Residual standard error: 0.3656 on 47 degrees of freedom
Multiple R-squared:  0.004323,  Adjusted R-squared:  -0.01686 
F-statistic: 0.2041 on 1 and 47 DF,  p-value: 0.6535
> x10 = data10[,1]
> y10 = data10[,2]
> freg.o = function (y10,x10,B,a){
+ n = length(y10)
+ b010 = matrix(coef(lm(y10~x10)), nrow = 1, ncol = 2)
+ y10fit = b010[,1] + b010[,2]*x10
+ e10 = seq(1:n)
+ for(i in 1:n) {
+ e10[i] = y10[i] - y10fit[i]}
+ e10star = matrix(0, nrow = B, ncol = n)
+ y10star = matrix(0, nrow = B, ncol = n)
+ b10.r = matrix(0, nrow = B, ncol = 2)
+ for (i in 1:B) {
+ e10star[i,] = sample(e10, n, replace = T)
+ y10star[i,] = b010[,1] + b010[,2]*x10 + e10star[i,]
+ b10.r[i,] = coef(lm(y10star[i,]~x10))}
+ b10boot = apply(b10.r,2,mean)
```

```
+ lo0_r10 = b010[,1] + qnorm(a/2)*sqrt(var(b10.r[,1]))
+ lo1_r10 = b010[,2] + qnorm(a/2)*sqrt(var(b10.r[,2]))
+ up0_r10 = b010[,1] + qnorm(1-a/2)*sqrt(var(b10.r[,1]))
+ up1_r10 = b010[,2] + qnorm(1-a/2)*sqrt(var(b10.r[,2]))
+ int1_r10 = matrix(c(lo0_r10, up0_r10, lo1_r10, up1_r10),
+ nrow = 2, byrow = T)
+ lop0_r10 = quantile(b10.r[,1], a/2)
+ lop1_r10 = quantile(b10.r[,2], a/2)
+ upp0_r10 = quantile(b10.r[,1], 1-a/2)
+ upp1_r10 = quantile(b10.r[,2], 1-a/2)
+ int2_r10 = matrix(c(lop0_r10, upp0_r10, lop1_r10, upp1_r10),
+ nrow = 2, byrow = T)
> list(est.boot_r10 = b10boot, int.normal_r10 = int1_r10,
+ int.pct_r10 = int2_r10)}
> #Bootstrap - 100
> set.seed(1001)
> freg.o(y10,x10,100,0.05)
$est.boot_r10
[1] -0.544452043  0.006217851

$int.normal_r10
            [,1]      [,2]
[1,] -3.43771672 2.5682263
[2,] -0.01700507 0.0278094

$int.pct_r10
            [,1]       [,2]
[1,] -3.25672787 1.85559913
[2,] -0.01168185 0.02613983
```

k. November

```
> data11 = read.csv(file.choose(), header = T, sep = ",")
> model11 = lm(Inflasi~IHK, data = data11)
> summary(model11)

Call:
lm(formula = Inflasi ~ IHK, data = data11)

Residuals:
     Min       1Q   Median       3Q      Max
-0.85164 -0.11388 -0.00131  0.12248  0.58268
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.594486   1.182894   2.193   0.0333 *
IHK         -0.017221   0.008803  -1.956   0.0564 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2629 on 47 degrees of freedom
Multiple R-squared:  0.07529,   Adjusted R-squared:  0.05562
F-statistic: 3.827 on 1 and 47 DF,  p-value: 0.05639
> x11 = data11[,1]
> y11 = data11[,2]
> freg.n = function (y11,x11,B,a){
+ n = length(y11)
+ b011 = matrix(coef(lm(y11~x11)), nrow = 1, ncol = 2)
+ y11fit = b011[,1] + b011[,2]*x11
```

```
+ e11 = seq(1:n)
+ for(i in 1:n) {
+ e11[i] = y11[i] - y11fit[i]}
+ e11star = matrix(0, nrow = B, ncol = n)
+ y11star = matrix(0, nrow = B, ncol = n)
+ b11.r = matrix(0, nrow = B, ncol = 2)
+ for (i in 1:B) {
+ e11star[i,] = sample(e11, n, replace = T)
+ y11star[i,] = b011[,1] + b011[,2]*x11 + e11star[i,]
+ b11.r[i,] = coef(lm(y11star[i,]~x11))}
+ b11boot = apply(b11.r,2,mean)
+ lo0_r11 = b011[,1] + qnorm(a/2)*sqrt(var(b11.r[,1]))
+ lo1_r11 = b011[,2] + qnorm(a/2)*sqrt(var(b11.r[,2]))
+ up0_r11 = b011[,1] + qnorm(1-a/2)*sqrt(var(b11.r[,1]))
+ up1_r11 = b011[,2] + qnorm(1-a/2)*sqrt(var(b11.r[,2]))
+ int1_r11 = matrix(c(lo0_r11, up0_r11, lo1_r11, up1_r11),
+ nrow = 2, byrow = T)
+ lop0_r11 = quantile(b11.r[,1], a/2)
+ lop1_r11 = quantile(b11.r[,2], a/2)
+ upp0_r11 = quantile(b11.r[,1], 1-a/2)
+ upp1_r11 = quantile(b11.r[,2], 1-a/2)
+ int2_r11 = matrix(c(lop0_r11, upp0_r11, lop1_r11, upp1_r11),
+ nrow = 2, byrow = T)
> list(est.boot_r11 = b11boot, int.normal_r11 = int1_r11,
+ int.pct_r11 = int2_r11)}
> #Bootstrap - 100
> set.seed(1101)
> freg.n(y11,x11,100,0.05)
$est.boot_r11
[1]  2.5461133 -0.0168753

$int.normal_r11
            [,1]         [,2]
[1,]  0.74832452  4.440646896
[2,] -0.03102639 -0.003415332

$int.pct_r11
            [,1]         [,2]
[1,]  0.68594063  4.192925688
[2,] -0.02931878 -0.003309975
```

l. Desember

```
> data12 = read.csv(file.choose(), header = T, sep = ",")
> model12 = lm(Inflasi~IHK, data = data12)
> summary(model12)
Call:
lm(formula = Inflasi ~ IHK, data = data12)

Residuals:
     Min       1Q   Median       3Q      Max
-0.52028 -0.21949 -0.01536  0.11484  1.24246
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.45151    1.43271  -1.013    0.316
IHK          0.01475    0.01060   1.391    0.171

Residual standard error: 0.3223 on 47 degrees of freedom
Multiple R-squared:  0.03952,   Adjusted R-squared:  0.01908
F-statistic: 1.934 on 1 and 47 DF,  p-value: 0.1709
> x12 = data12[,1]
> y12 = data12[,2]
> freg.d = function (y12,x12,B,a){
+ n = length(y12)
+ b012 = matrix(coef(lm(y12~x12)), nrow = 1, ncol = 2)
+ y12fit = b012[,1] + b012[,2]*x12
+ e12 = seq(1:n)
+ for(i in 1:n) {
+ e12[i] = y12[i] - y12fit[i]}
+ e12star = matrix(0, nrow = B, ncol = n)
+ y12star = matrix(0, nrow = B, ncol = n)
+ b12.r = matrix(0, nrow = B, ncol = 2)
+ for (i in 1:B) {
+ e12star[i,] = sample(e12, n, replace = T)
+ y12star[i,] = b012[,1] + b012[,2]*x11 + e12star[i,]
+ b12.r[i,] = coef(lm(y12star[i,]~x12))}
+ b12boot = apply(b12.r,2,mean)
+ lo0_r12 = b012[,1] + qnorm(a/2)*sqrt(var(b12.r[,1]))
+ lo1_r12 = b012[,2] + qnorm(a/2)*sqrt(var(b12.r[,2]))
+ up0_r12 = b012[,1] + qnorm(1-a/2)*sqrt(var(b12.r[,1]))
+ up1_r12 = b012[,2] + qnorm(1-a/2)*sqrt(var(b12.r[,2]))
+ int1_r12 = matrix(c(lo0_r12, up0_r12, lo1_r12, up1_r12),
+ nrow = 2, byrow = T)
+ lop0_r12 = quantile(b12.r[,1], a/2)
+ lop1_r12 = quantile(b12.r[,2], a/2)
+ upp0_r12 = quantile(b12.r[,1], 1-a/2)
+ upp1_r12 = quantile(b12.r[,2], 1-a/2)
+ int2_r12 = matrix(c(lop0_r12, upp0_r12, lop1_r12, upp1_r12),
+ nrow = 2, byrow = T)
> list(est.boot_r12 = b12boot, int.normal_r12 = int1_r12,
+ int.pct_r12 = int2_r12)}
> #Bootstrap - 100
> set.seed(1201)
> freg.d(y12,x12,100,0.05)
$est.boot_r12
[1] -1.50393781  0.01504352

$int.normal_r12
             [,1]       [,2]
[1,] -3.823046321 0.92001750
[2,] -0.002851855 0.03234365

$int.pct_r12
             [,1]       [,2]
[1,] -4.018331372 0.79120216
[2,] -0.001701043 0.03374474
```