PENERAPAN METODE ENSEMBLE KALMAN FILTER PADA MODEL PERTUMBUHAN LOGISTIK MENGGUNAKAN FUNGSI POPULASI PARABOLIK

SKRIPSI

Oleh
Ayu Wulandari
NIM 151810101023

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2019

PENERAPAN METODE ENSEMBLE KALMAN FILTER PADA MODEL PERTUMBUHAN LOGISTIK MENGGUNAKAN FUNGSI POPULASI PARABOLIK

SKRIPSI

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

Oleh
Ayu Wulandari
NIM 151810101023

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2019

PERSEMBAHAN

Dengan segala kerendahan hati dan puji syukur yang tak terhingga kepada Allah Subhanahu Wa Ta'ala. Skripsi ini saya persembahkan untuk :

1. Alm. Ayahanda Musa dan Ibunda Musyarofah tercinta yang selalu memberi saya semangat tanpa batas dan selalu memberikan doa, restu, kasih sayang dan pengorbanan yang sangat luar biasa di setiap langkah saya;
2. Adik-adik saya tersayang Faizal Akbar dan Fanina Ulin Nuha yang selalu mendoakan dan memotivasi saya dalam setiap rintangan menjalani hidup ini;
3. Guru-guru taman kanak Sabilal Muta'alimin, MI Tharbiyatussibyan, SMPN 04 Genteng, SMAN 01 Glenmore, serta seluruh Dosen Jurusan Matematika yang telah mendidik dan membimbing dengan penuh kesabaran;
4. Almamater tercinta Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

MOTTO

"Fabiayyi 'ala irobbikuma tukadziban"
Maka nikmat Tuhanmu manakah yang kamu dustakan
(Q.S: Ar-Rahman Ayat 16)

Ketakutanmu terhadap kegagalan, itulah yang membuatmu gagal
(Habib Syeh Bin Abdul Qadir Assegaf)

"Khoirunnas anfa'uhum linnas (sebaik-baiknya manusia adalah yang bisa memberi manfaat untuk orang lain)". (H.R Bukhari)[1]

[1] Kutipan Inspirasi Iman (25-04-203) Ustadz Felix Siauw dan Ahmad Fuadi (Penulis Novel Best Seller Negeri 5 Menara)

PERNYATAAN

Saya yang bertanda tangan di bawah ini

Nama : Ayu Wulandari

NIM : 151810101023

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul ”Penerapan Metode *Ensemble Kalman Filter* Pada Model Pertumbuhan Logistik Menggunakan Fungsi Populasi Parabolik” adalah benar-benar hasil karya ilmiah sendiri, kecuali jika dalam pengutipan substansi disebutkan 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, Juli 2019

Yang menyatakan,

Ayu Wulandari

NIM 151810101023

SKRIPSI

PENERAPAN METODE ENSEMBLE KALMAN FILTER PADA MODEL PERTUMBUHAN LOGISTIK MENGGUNAKAN FUNGSI POPULASI PARABOLIK

Oleh
Ayu Wulandari
151810101023

Pembimbing

Dosen Pembimbing Utama : Kosala Dwidja Purnomo, S.Si., M.Si.

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

PENGESAHAN

Skripsi berjudul "Penerapan Metode *Ensemble Kalman Filter* Pada Model Pertumbuhan Logistik Menggunakan Fungsi Populasi Parabolik", telah diuji dan disahkan pada:

hari, tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Tim Penguji:

Ketua,	Sekretaris,
Kosala Dwidja Purnomo, S.Si., M.Si. NIP.196908281998021001	Dian Anggraeni, S.Si., M.Si. NIP.198202162006042002
Penguji I,	Penguji II,
Dr. Firdaus Ubaidillah, S.Si., M.Si. NIP. 197006061998031003	Dr. Alfian Futuhul Hadi, S.Si, M.Si NIP. 197407192000121001

Mengesahkan
Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Jember

Drs. Sujito, Ph.D.
NIP. 196102041987111001

RINGKASAN

Penerapan Metode *Ensemble Kalman Filter* Pada Model Pertumbuhan Logistik Menggunakan Fungsi Populasi Parabolik; Ayu Wulandari 151810101023; 2019: 46 halaman; Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Jember.

Populasi merupakan kumpulan organisme dari spesies yang sama dan hidup secara bersamaan dengan bertumbuh dan berkembang biak. Pada pertumbuhan makhluk hidup di suatu populasi didefinisikan perubahan ukuran populasi pada periode waktu tertentu. Populasi mempunyai beberapa pola pertambahan yang disebut sebagai bentuk pertumbuhan populasi. Terdapat pertumbuhan populasi eksponensial apabila lingkungan tidak terbatas seperti ruang hidup, ketersediaan makanan, dan organisme lainnya yang tidak melakukan pembatasan maka populasi suatu spesies akan meningkat secara tidak terbatas. Selain itu terdapat pula Pertumbuhan logistik merupakan model pertumbuhan populasi dengan kapasitas daya tampung (*carriying capacity*). Daya tampung (*carriying capacity*) merupakan batas atas dari pertumbuhan suatu populasi. Tingkat pertumbuhan dihitung menggunakan persamaan diferensial dengan dua faktor yaitu jumlah populasi dan satuan waktu. Untuk mengetahui hubungan tersebut dilakukan prediksi dan estimasi. Salah satu estimasi data yaitu EnKF yang merupakan untuk estimasi data linier maupun non linier. Tujuan dari penelitian ini adalah estimasi pertumbuhan logistik dengan metode *Ensemble Kalman Filter* menggunakan fungsi populasi parabolik terbuka ke atas.

Beberapa langkah untuk memperoleh hasil estimasi, yaitu menentukan model persamaan diferensial dan solusi numerik, dimana solusi numerik dicari dengan metode Runge-Kutta orde empat. Pendiskritan dilakukan dengan metode hingga maju. Model diskrit yang telah diperoleh selanjutnya akan diubah menjadi sistem stokastik. Perubahan sistem stokastik ditunjukkan dengan adanya penambahan *noise*. Setelah itu mengimplementasi metode EnKF pada model pertumbuhan logistik. Banyaknya *ensemble* yang dibangkitkan pada langkah ini

dengan ukuran yang berbeda diantaranya 100, 200, 300, 400, 500, dan 1000. Pengambilan jumlah *ensemble* tersebut guna untuk melihat masing-masing ukuran *ensemble* yang cocok dalam mengestimasi pertumbuahn logistik. Analisis dilakukan dengan melihat dan mengevaluasi ketepatan hasil estimasi dari metode EnKF yang kemudian dibandingkan dengan solusi numeriknya.

Hasil simulasi menunjukkan bahwa fungsi populasi parabolik terbuka ke atas bisa diterapkan di model pertumbuhan logistik menggunakan metode *Ensemble Kalman Filter*. Ini terlihat dari grafik yang dihasilkan dalam proses estimasi yang pergerakan potret fasenya membentuk seperti solusi analitik dari bentuk model pertumbuhan logistik. Pada pengestimasian pertumbuhan logistik dengan metode *Ensemble Kalman Filter* menggunakan fungsi populasi parabolik terbuka ke atas, pengambilan jumlah $N_e = 100$ memberikan hasil estimasi yang baik. Ini terlihat dari hasil nilai rata-rata selisih mutlak yang relatif kecil sebesar 1,8086 dan juga adanya nilai rata-rata norm kovariansi *error* yang mempunyai *error* paling kecil diantara pengambilan *ensemble* 200, 300, 400, 500, dan 1000 yaitu sebesar 6,9277 x 10^{-5}.

PRAKATA

Puji Syukur kehadirat Allah SWT atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Skripsi yang berjudul “Penerapan Metode *Ensemble Kalman Filter* Pada Model Pertumbuhan Logistik Menggunakan Fungsi Populasi Parabolik”.Skripsi ini disusun untuk memenuhi salah satu syarat dalam menyelesaikan pendidikan strata satu (S1) pada Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak, oleh karena itu penulis ingin menyampaikan ucapan terima kasih kepada:

1. Kosala Dwidja Purnomo, S.Si., M.Si. selaku Dosen Pembimbing Utama dan Dian Anggraeni, S.Si., M.Si selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, serta perhatian dalam penulisan skripsi ini;
2. Dr. Firdaus Ubaidillah, S.Si., M.Si dan Dr. Alfian Futuhul Hadi, S.Si., M.Si selaku dosen penguji yang telah memberikan kritik dan saran demi perbaikan tugas akhir ini;
3. Seluruh dosen dan staf karyawan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember;
4. Seluruh teman seperjuangan SIGMA’15 Jurusan Matematika Universitas Jember yang telah banyak membantu selama perkuliahan;
5. Kakek Taprani dan Almh. Nenek Saninten yang selalu mendoakan kesuksesan saya tanpa henti;
6. Kaka sepupu Anas Nurhidayah dan Taqrub Iyyaka yang selalu memberi semangat kuliah saya agar lebih baik;
7. Teman-teman KKN 243 Agel khususnya Dini, Devi, Diah, dan Hadi yang telah menjadi saudara di jember ini;
8. Teman-teman Kos 39 Squad Kalimantan yang telah menjadi keluarga kedua selama berada di Jember;
9. Teman-teman BHAMBLO yaitu Indy, Tika, Nanda, dan Yuli yang selalu memberi dukungan, tenaga dan semangat tanpa henti;

10. Teman-teman satu bidang skripsi Kalman Filter Rory Ronella Agustin yang selalu memberi dukungan dan semangat tanpa henti;
11. Teman seperjuangan skripsi Dyakza Hadi Pramestika dan Ellenda Alkhori;
12. Pengurus HIMATIKA "Geokompstat" masa bakti 2017 yang selalu senantiasa memberi dukungan;
13. Pengurus Dewan Perwakilan Mahasiswa (DPM) masa bakti 2018 yang selalu senantiasa memberi dukungan;
14. Kakak tingkat dan adek tingkat serta alumni, massay, kak mia, mas fathoni kak afidah yang membantu dan memotivasi penulis;
15. Semua pihak yang telah memberikan sumbangan tenaga, semangat, dan pikiran yang tidak dapat disebutkan satu persatu oleh penulis.

Penulisan skripsi ini masih jauh dari kesempurnaan, sehingga diharapkan adanya kritik dan saran untuk perbaikan selanjutnya. Penulis berharap skripsi ini bermanfaat bagi semua.

Jember, Juli 2019 Penulis

DAFTAR ISI

Halaman

DAFTAR TABEL

Halaman

DAFTAR GAMBAR

Halaman

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Populasi merupakan kumpulan organisme dari spesies yang sama dan hidup secara bersamaan dengan bertumbuh dan berkembang biak. Populasi dapat mengalami perubahan baik jumlah populasinya yang bertambah ataupun menurun. Pertumbuhan sendiri adalah proses bertambahnya volume dan jumlah sel sehingga ukuran tubuh makhluk hidup tersebut bertambah besar. Pada pertumbuhan makhluk hidup di suatu populasi didefinisikan perubahan ukuran populasi pada periode waktu tertentu. Populasi mempunyai beberapa pola pertambahan yang disebut sebagai bentuk pertumbuhan populasi. Terdapat pertumbuhan populasi eksponensial apabila lingkungan tidak terbatas seperti ruang hidup, ketersediaan makanan, dan organisme lainnya yang tidak melakukan pembatasan maka populasi suatu spesies akan meningkat secara tidak terbatas (Sunarsih dan Hidayati, 2010). Selain itu terdapat pula yang dikenal dengan pertumbuhan logistik. Pertumbuhan logistik merupakan model pertumbuhan populasi dengan kapasitas daya tampung (*carriying capacity*). Daya tampung (*carriying capacity*) merupakan batas atas dari pertumbuhan suatu populasi. Proses perubahan populasi tersebut berlangsung secara diskrit, sehingga untuk pengukurannya dilakukan di selang waktu tertentu. Tingkat pertumbuhan dihitung menggunakan persamaan diferensial dengan dua faktor yaitu jumlah populasi dan satuan waktu. Untuk mengetahui hubungan tersebut maka dilakukan prediksi dan estimasi. Prediksi tersebut digunakan untuk mengoreksi data sedangkan estimasi untuk mengestimasi data dengan hasil yang mendekati kejadian sebenarnya. Ada beberapa estimator atau metode estimasi data, diantaranya metode *Kalman Filter.*

Kalman Filter (KF) adalah salah satu metode untuk mengestimasi suatu masalah dengan menggunakan sistem keadaan dan model pengukuran yang diperkenalkan oleh R.E. Kalman (1960). Algoritma Kalman Filter sendiri hanya dapat digunakan dalam masalah linier saja. Namun pada kenyataannya permasalahan yang timbul tidak hanya sistem linier melainkan juga non linier.

Dikembangkan algoritma yang dapat diimplementasikan pada masalah tersebut. Algoritma yang telah dikembangkan diantaranya *Extended Kalman Filter* (EKF) dan *Ensemble Kalman Filter* (EnKF). Perbedaan antara kedua metode tersebut adalah *Extended Kalman Filter* (EKF) untuk sistem non linier dan cara perhitungannya perlu dilinierisasi apabila sistem tidak linier lalu pendiskritan sistem apabila sistem berbentuk kontinu. *Ensemble Kalman Filter* (EnKF) bisa mengestimasi model sistem linier maupun non linier. Lalu untuk perhitungannya dengan cara membangkitkan sejumlah *ensemble* tertentu untuk menghitung *mean* dan kovarian *error*. Beberapa contoh artikel yang membahas permasalahan yang diselesaikan dengan metode Kalman Filter yaitu aplikasi metode EnKF dalam mengestimasi populasi plankton di laut menghasilkan solusi bahwa ada sebagian variabel tertentu penggunaan EKF lebih cocok mengestimasi daripada metode EnKF (Purnomo, 2008). Terdapat algoritma Kalman Filter dalam *The Groundwater Pollution Estimation by The Ensemble Kalman Filter* menghasilkan bahwa metode EnKF lebih akurat dibanding Kalman Filter (Erna dkk. 2011).

Penerapan *Ensemble Kalman Filter* pada pertumbuhan logistik sebelumnya telah diterapkan oleh Fitriani (2012) yang telah melakukan penelitian pada kurva parabolik terbuka ke bawah. Penelitian ini menghasilkan bahwa metode EnKF dapat diterapkan pada model pertumbuhan logistik dengan langkah-langkah metode EnKF itu sendiri dan persamaan Bernoulli dengan solusi analitik dari model pertumbuhan logistik. Solusi analitik dari model pertumbuhan logistik tersebut bisa membuktikan bahwa hasil dari kovariansi errornya sangat kecil sehingga terbukti bahwa metode EnKF bisa juga diterapkan pada model pertumbuhan logistik.

Pada tulisan ini akan dilakukan estimasi pertumbuhan logistik dengan metode *Ensemble Kalman Filter* menggunakan fungsi populasi parabolik terbuka ke atas (warna hijau). Fungsi populasi parabolik terbuka ke atas tersebut lalu digunakan sebagai acuan untuk mencari solusi numerik. Pendiskritisasian dan penambahan noise dilakukan untuk menghitung gangguan sistem yang ada. Setelah itu dilakukan penerapan *Ensemble Kalman Filter* pada pertumbuhan logistik dengan fungsi parabolik terbuka ke atas. Untuk memudahkan simulasi pada komputasi dalam proses estimasi digunakan program matlab. Program matlab yang digunakan versi Matlab R2015b. Sehingga dari simulasi Matlab R2015b dapat diketahui apakah bisa diterapkan EnKF juga pada model pertumbuhan logistik.

1.2 Perumusan Masalah

Permasalahan yang akan dibahas dalam penelitian ini adalah apakah metode *Ensemble Kalman Filter* dapat diterapkan untuk mengestimasi model pertumbuhan logistik menggunakan fungsi populasi parabolik terbuka ke atas.

1.3 Tujuan

Tujuan yang ingin dicapai dalam penelitian ini adalah mengetahui hasil metode *Ensemble Kalman Filter* dalam penerapan model pertumbuhan logistik menggunakan fungsi populasi parabolik terbuka ke atas.

1.4 Manfaat

Manfaat yang diperoleh dalam penelitian ini adalah memberikan informasi penyelesaian penerapan metode *Ensemble Kalman Filter* pada model pertumbuhan logistik menggunakan fungsi populasi parabolik terbuka ke atas.

BAB 2. TINJAUAN PUSTAKA

Pada bab ini diuraikan beberapa hal yang berhubungan dengan permasalahan yang akan dibahas, yaitu estimasi pertumbuhan logistik dengan metode *Ensemble Kalman Filter* (EnKF) menggunakan fungsi populasi parabolik terbuka ke atas. Berikut beberapa teori dasar yang berhubungan dengan estimasi pertumbuhan logistik. Pertama kajian tentang pertumbuhan logistik serta persamaan diferensial dan untuk mencari solusi numerik digunakan metode Runge-Kutta orde empat. Kemudian untuk melakukan diskritisasi diperlukan pembahasan tentang metode beda hingga. Selanjutnya untuk prediksi dan estimasi diuraikan mengenai kedua metode yaitu KF dan EnKF. Terakhir, untuk mengetahui tingkat keakuratan dari kedua metode tersebut dibutuhkan pembahasan tentang norm matriks kovariansi error.

2.1 Model Pertumbuhan Logistik

Pada model pertumbuhan logistik terdapat ketergantungan antara jumlah populasi pada waktu yang berturut-turut. Misal notasi untuk banyaknya populasi pada waktu t adalah $N(t)$, sehingga untuk laju perubahan populasi terhadap waktu t adalah $dN(t)/dt$. Jika laju perubahan populasi sebanding dengan banyaknya populasi yang ada, maka:

$$\frac{dN(t)}{dt} = kN(t) \tag{2.1}$$

atau

$$\frac{dN(t)/dt}{N(t)} = k \tag{2.2}$$

dengan k merupakan konstanta dan disebut laju reproduksi.

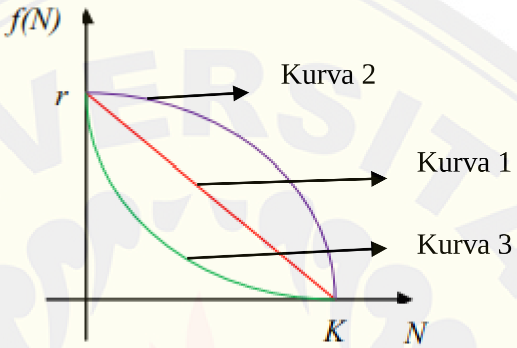
Nilai k dapat diwujudkan sebagai suatu fungsi apabila k tidak konstan, misal $f(N)$. Sehingga untuk model Persamaan (2.2) diubah menjadi:

$$\frac{dN(t)/dt}{N(t)} = f(N(t)) \tag{2.3}$$

atau

$$\frac{dN(t)}{dt} = f(N(t))N(t) \tag{2.4}$$

Pada Persamaan (2.1) merupakan persamaan pertumbuhan eksponensial. Kemudian Persamaan (2.1) tersebut dapat ditentukan model baru yaitu persamaan Pertumbuhan logistik (2.4). Penghambatan pertumbuhan populasi dijelaskan secara matematika dengan menambahkan variabel yang menjelaskan pengaruh kepadatan ke dalam persamaan eksponensial.



Gambar 2. 1 Grafik Hubungan $f(N)$ dan N

Keterangan dimisalkan:

Kurva linier (berwarna merah) : kurva 1

Kurva terbuka ke bawah (berwarna ungu) : kurva 2

Kurva terbuka ke atas (berwarna hijau) : kurva 3

Berdasarkan Gambar 2.1 asumsi yang dapat dibuat dari Persamaan (2.4) adalah $f(N(t))$ kurva 1, yaitu $f\big(N(t)\big) = c_1 N(t) + c_2$. Apabila menggunakan kondisi $f(0) = r$, dimana r adalah laju pertumbuhan dan $f(K) = 0$ dimana K merupakan *carriying capacity*. Sehingga didapat $c_2 = r$ dan $c_1 = -r/K$. Jadi diperoleh bentuk $f\big(N(t)\big) = r - \frac{r}{K}N(t)$. Oleh karena itu, persamaan pertumbuhan logistik dengan $f\big(N(t)\big)$ linier adalah:

$$\frac{dN(t)}{dt} = \left(r - \frac{r}{K}N(t)\right)N(t) \tag{2.5}$$

Dari Persamaan (2.5) mempunyai solusi analitik, yaitu:

$$N(t) = \frac{K}{1+e^{-rt}CK} \tag{2.6}$$

C adalah konstanta, Persamaan (2.6) merupakan solusi analitik dari kuva berbentuk linier pada Gambar 2.1.

Pada tulisan sebelumnya telah melakukan penelitian juga berdasarkan Gambar 2.1 mengembangkan asumsi $f(N(t))$ kurva 1 menjadi $f(N(t))$ kurva 2 dengan asumsi yaitu $f\big(N(t)\big) = aN(t)^2 + b$. Jika menggunakan kondisi $f(0) = r$ dan $f(K) = 0$, maka dapat menemukan nilai dari konstanta $a = -\frac{r}{K^2}$ dan $b = r$. Sehingga diperoleh $f\big(N(t)\big) = -\frac{r}{K^2}(N(t))^2 + r$. Oleh karena itu persamaan pertumbuhan logistik adalah:

$$\frac{dN(t)}{dt} = \left(-\frac{r}{K^2}\big(N(t)\big)^2 + r\right)N(t) \tag{2.7}$$

dari Persamaan (2.7) mempunyai solusi analitik, yaitu:

$$N(t) = \sqrt{\frac{K^2}{1+2re^{-2rt}C}} \tag{2.8}$$

C adalah konstanta sembarang (Fitriani, 2013).

Pada penelitian ini untuk asumsi $f(N(t))$ di kurva 1 dan $f\big(N(t)\big)$ bentuk parabolik di kurva 2 akan dikembangkan lagi menjadi bentuk parabolik kurva 3 seperti Gambar 2.1. Asumsi untuk bentuk parabolik kurva 3 yaitu :

$$f\big(N(t)\big) = a(N(t) - b)^2 \tag{2.9}$$

dimana a dan b adalah konstanta. Dalam hal ini, model pertumbuhan logistik yang akan dikaji adalah model pertumbuhan logistik dengan fungsi parabolik yang asumsinya $f\big(N(t)\big)$ adalah fungsi parabolik dalam bentuk Persamaan (2.9).

2.2 Persamaan Diferensial

Persamaan diferensial adalah suatu persamaan yang melibatkan suatu fungsi yang dicari variabel terikat dan turunannya. Persamaan diferensial dibagi menjadi dua yaitu persamaan diferensial biasa dan persamaan diferensial parsial. Persamaan diferensial biasa (PDB) adalah persamaan diferensial yang hanya mempunyai satu peubah bebas, sedangkan persamaan diferensial parsial (PDP) adalah persamaan diferensial yang mempunyai lebih dari satu peubah bebas (Zill, 2012). Berikut contoh persamaan diferensial:

$$\frac{dy}{dx} = x + y$$

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 6xye^{x+y} \qquad (2.10)$$

Persamaan (2.10) merupakan persamaan diferensial parsial orde 2. Orde atau tingkat persamaan diferensial adalah orde dari turunan tertinggi yang muncul dalam persamaan tersebut (Hidayat, 2006).

2.3 Metode Runge-Kutta Orde Empat

Metode Runge-Kutta merupakan salah satu subbab dari solusi persamaan diferensial biasa. Metode Runge Kutta adalah metode yang memberikan ketelitian yang besar dan tidak memerlukan turunan fungsi. Metode ini berusaha memperoleh derajat ketelitian yang lebih tinggi dengan mengevaluasi fungsi $f(t,x)$. Metode Runge-Kutta mempunyai bentuk umum seperti berikut:

$$x_{i+1} = x_i + (a_1k_1 + a_2k_2 + a_3k_3 + \cdots + a_nk_n)h$$

dimana nilai x_i adalah nilai lama dan x_{i+1} nilai baru yang akan dicari sepanjang interval h, sedangkan $a_1, a_2, \dots, a_n$ adalah tetapan, dan k adalah:

$$k_1 = f(t_i, x_i)$$

$$k_2 = f(t_i + p_ih, x_i + q_{11}k_1h)$$

$$k_3 = f(t_i + p_ih, x_i + q_{21}k_1h + q_{22}k_2h)$$

$$\vdots$$

$$k_n = f(t_i + p_{n-1}h, x_i + q_{n-1,1}k_1h + q_{n-1,2}k_2h + \cdots + q_{n-1,1-1}k_{n-1}h)$$

nilai p dan q adalah konstanta, nilai k menunjukkan hubungan berurutan. Nilai k_1 muncul di k_2, lalu k_1 dan k_2 akan muncul di k_3 dan seterusnya.

Metode Runge-Kutta orde empat mempunyai tingkat ketelitian solusinya tinggi dibandingkan metode Runge-Kutta orde sebelumnya. Metode Runge-Kutta orde empat sering digunakan untuk menyelesaiakn masalah persamaan diferensial. Metode Runge-Kutta orde empat mempunyai bentuk umum sebagai berikut.

$$x_{i+1} = x_i + \tfrac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)h \tag{2.11}$$

dengan

$$k_1 = f(t_i, x_i)$$
$$k_2 = f\left(t_i + \frac{1}{2}h, x_i + \frac{1}{2}hk_1\right)$$
$$k_3 = f\left(t_i + \frac{1}{2}h, x_i + \frac{1}{2}hk_2\right)$$
$$k_4 = f(t_i + h, x_i + hk_3)$$

Metode Runge-Kutta orde empat ini mempunyai tingkat ketelitian solusi yang lebih tinggi dari pada metode Runge-Kutta orde dua dan orde tiga. Metode Runge-Kutta orde empat juga mudah diprogram, stabil, kecil kesalahan pembulatan (Triatmodjo, 2002).

2.4 Metode Beda Hingga

Apabila dalam suatu persamaan dasar tidak diperoleh penyelesaian analitik maka digunakan pendekatan numerik yaitu dengan metode beda hingga. Untuk menggunakan metode beda hingga, domain dari persamaan dasar tersebut harus didiskritkan. Deret Taylor merupakan dasar untuk penyelesaian masalah numerik terutama pada persamaan diferensial. Diferensial numerik sendiri untuk memperkirakan bentuk diferensial kontinyu menjadi bentuk diskrit. Berikut Persamaan deret Taylor:

$$f(x_{k+1}) = f(x_k) + \Delta x f'(x_k) + \tfrac{\Delta x^2}{2!} f''(x_k) + \cdots \tag{2.12}$$

$$f(x_{k-1}) = f(x_k) - \Delta x f'(x_k) - \tfrac{\Delta x^2}{2!} f''(x_k) - \cdots \tag{2.13}$$

Beberapa skema numerik dari metode beda hingga, yaitu:

a. Beda Hingga Maju

Dari Persamaan (2.12) diperoleh:

$$f(x_{k+1}) = f(x_k) + \Delta x f'(x_k) + \tfrac{\Delta x^2}{2!} f''(x_k) + \cdots$$

$$f(x_{k+1}) = f(x_k) + \Delta x f'(x_k) + O(\Delta x)$$

atau

$$\frac{\partial f}{\partial x} = f'(x_k) = \frac{f(x_{k+1}) - f(x_k)}{\Delta x} - O(\Delta x)$$

$$\frac{\partial f}{\partial x} \approx \frac{f(x_{k+1}) - f(x_k)}{\Delta x} \qquad (2.14)$$

Persamaan diferensial (2.14) disebut diferensial beda hingga maju karena menggunakan data di titik x_k dan x_{k+1}.

b. Beda Hingga Mundur

Dari Persamaan (2.13) diperoleh:

$$f(x_{k-1}) = f(x_k) - \Delta x f'(x_k) + \frac{\Delta x^2}{2!} f''(x_k) - \cdots$$

$$f(x_{k-1}) = f(x_k) - \Delta x f'(x_k) - O(\Delta x)$$

atau

$$\frac{\partial f}{\partial x} = f'(x_k) = \frac{f(x_k) - f(x_{k-1})}{\Delta x} - O(\Delta x)$$

$$\frac{\partial f}{\partial x} \approx \frac{f(x_k) - f(x_{k-1})}{\Delta x} \qquad (2.15)$$

Sedangkan Persamaan (2.15) disebut persamaan diferensial beda hingga mundur karena menggunakan data di titik x_k dan x_{k-1}.

c. Beda Hingga Pusat

Apabila Persamaan (2.14) dikurangi Persamaan (2.15) maka diperoleh:

$$f(x_{k+1}) - f(x_{k-1}) = 2f'(x_k)\Delta x + f'''(x_k)\frac{\Delta x^3}{3!} + \cdots$$

atau

$$\frac{\partial f}{\partial x} = f'(x_k) = \frac{f(x_{k+1}) - f(x_{k-1})}{2\Delta x} - O(\Delta x)$$

$$\frac{\partial f}{\partial x} \approx \frac{f(x_{k+1}) - f(x_{k-1})}{2\Delta x} \tag{2.16}$$

Untuk Persamaan (2.16) disebut persamaan beda hingga pusat karena menggunakan data pada titik x_{k+1} dan x_{k-1} (Triatmodjo, 1992).

Diantara beberapa skema proses beda hingga, digunakan metode beda hingga maju. Metode beda hingga maju dinilai cocok untuk permasalahan model persamaan logistik. Karena data yang diketahui adalah data pada saat k, sedangkan data yang akan dicari adalah data saat $k + 1$.

2.5 Metode Kalman Filter

Kalman Filter (KF) adalah suatu metode estimasi variabel keadaan dari sistem dinamik stokastik linier diskrit yang meminimumkan kovariansi error. Metode KF pertama kali diperkenalkan oleh Rudolp E. Kalman pada tahun 1960 melalui papernya yang sangat terkenal mengenai suatu penyelesaian rekursif pada masalah filtering data diskrit yang linier (Welc & Bishop, 2006). KF merupakan metode pendekatan teknis untuk mengestimasi fungsi parameter dalam peramalan deret berkala (*time series*). Keunggulan Kalman Filter yaitu kemampuannya dalam mengestimasi suatu keadaan berdasarkan data yang minim. Data pengukuran terbaru menjadi bagian penting dari Kalman Filter karena data mutakhir akan mengoreksi hasil prediksi. Sehingga hasil estimasi selalu mendekati hasil atau kondisi yang sebenarnya (Masduqi dan Apriliani, 2008).

Proses estimasi metode Kalman Filter terdapat dua tahapan, yaitu tahap prediksi (*time update*) dan tahap koreksi (*measurement update*). Pada tahap prediksi dipengaruhi oleh dinamika sistem dengan memprediksi variabel keadaan dan tingkat akurasinya dihitung dengan menggunakan persamaan kovariansi error atau norm kovariansi error.

Pada tahap koreksi, hasil estimasi variabel keadaan yang diperoleh pada tahap sebelumnya (tahap prediksi) dikoreksi menggunakan model pengukuran untuk memperbaiki estimasi sesudahnya. Salah satu bagian dari tahap ini yaitu menentukan matriks Kalman Gain yang digunakan untuk meminimumkan kovarinasi error (Susanto, 2008).

Untuk menggambarkan suatu sistem dengan noise dan data pengukuran, maka dapat dibentuk persamaan linier stokastik waktu diskrit, sebagai berikut:

$$x_{k+1} = A_k x_k + B_k u_k + w_k \tag{2.17}$$

dengan pengukuran $z_k \in \mathcal{R}^p$ yang memenuhi

$$z_k = H_k x_k + v_k \tag{2.18}$$

dengan kondisi awal $x_0 = \bar{x}_0$; $x_0 \backsim N(\bar{x}_0, P_{x_0}); w_k \backsim N(0, Q_k); v_k \backsim N(0, R_k)$

Keterangan:

x_0 = inisialisasi dari sistem

x_{k+1} = variabel keadaan pada waktu *k+1* dan berdimensi

x_k = variabel keadaan pada waktu k yang nilai estimasi awalnya $\bar{x}_0$ dan kovariansi awal P_{x_0}, $x_k \in \mathcal{R}^n$

u_k = vektor masukan deterministik pada waktu k, $u_k \in \mathcal{R}^m$

w_k = *noise* pada sistem dengan *mean* $\bar{w}_k = 0$ dengan kovariansi Q_k

z_k = variabel pengukuran, $z_k \in \mathcal{R}^p$

v_k = *noise* pada pengukuran dengan mean $v_k = 0$ dengan kovariansi R_k

A_k, B_k, H_k = matriks-matriks dengan nilai elemen-elemennya adalah koefisien variabel masing-masing

Variabel $w_k \backsim N(0, Q_k)$ dan $v_k \backsim N(0, R_k)$ ini diasumsikan white (berdistribusi normal dengan mean nol), tidak berkorelasi satu sama lain maupun dengan nilai estimasi awal $\bar{x}_0$. Tahap prediksi dan koreksi dilakukan secara rekursif dengan cara meminimumkan kovariansi kesalahan estimasi $(x_k - \hat{x}_k)$, dimana x_k merupakan variabel keadaan yang sebenarnya dan $\hat{x}_k$ merupakan penaksiran dari variabel keadaan.

Misal diberikan kondisi awal $P_0 = P_{x_0}$ dan $\hat{x}_0 = \bar{x}_0$ kemudian akan diberikan estimasi untuk setiap waktu $k \geq 0$ dengan,

$\hat{x}_{\bar{k}+1}$ = estimasi sebelum pengukuran z_k dimasukkan pada waktu $k+1$

$\hat{P}^-_{x_0+1}$ = kovariansi error estimasi pengukuran sebelum z_k dimasukkan pada waktu $k+1$

$\hat{x}_{k+1}$ = estimasi setelah pengukuran z_k dimasukkan pada waktu $k+1$

$\hat{P}_{x_0+1}$ = kovariansi error estimasi pengukuran setelah z_k dimasukkan pada waktu $k+1$

Berikut adalah persamaan algoritma dari metode Kalman Filter selengkapnya dapat dilihat pada Tabel 2.1 yaitu:

Tabel 2. 1 Algoritma Kalman Filter

Model Sistem dan Model Pengukuran
$x_{k+1} = A_k x_k + B_k u_k + w_k$ $z_k = H_k x_k + v_k$

Lanjutan Tabel

$x_0 \backsim N(\bar{x}_0, P_{x_0}); w_k \backsim N(0, Q_k); v_k \backsim N(0, R_k)$
Inisialisasi
$\hat{x}_0 = \bar{x}_0$ $P_0 = P_{x_0}$
Tahap Prediksi
Estimasi : $\hat{x}_{k+1}^{-} = A_k \hat{x} + B_k u_k$ Kovariansi *Error* : $P_{k+1}^{-} = A_k P_k A_k^T + Q_k$
Tahap Koreksi
Kalman Gain: $K_{k+1} = P_{k+1}^{-} H_{k+1}^T (H_{k+1} P_{k+1}^{-} H_{k+1}^T + R_{k+1})^{-1}$ Estimasi: $\hat{x}_{k+1} = \hat{x}_{k+1}^{-} + K_{k+1}(Z_{k+1} - H_{k+1}\hat{x}_{k+1}^{-})$ Kovariansi Error : $P_{k+1} = [I - K_{k+1} H_{k+1}]\, P_{k+1}^{-}$

Pada Tabel 2.1 menunjukkan algoritma Kalman Filter yang terdiri dari empat bagian, yakni bagian pertama mendefinisikan model sistem dan pengukuran, kemudian pada bagian kedua mendefinisikan nilai awal, selanjutnya bagian ketiga adalah algoritma tahap prediksi, dan bagian keempat adalah algoritma tahap koreksi (Purnomo, 2008).

2.6 Metode *Ensemble Kalman Filter*

Metode *Ensemble Kalman Filter* (EnKF) adalah metode estimasi modifikasi dari algoritma Kalman Filter yang dapat digunakan untuk mengestimasi model sistem linier maupun nonlinier. Metode EnKF pertama kali dikembangkan oleh G. Evensen (1992-1993) pada saat mencoba mengimplementasikan metode EKF ternyata menyebabkan kovariansi errornya membesar menuju tak hingga. Pada tahun 1994 metode EnKF diperkenalkan oleh Evensen dengan membangkitkan atau menggunakan sejumlah *ensemble* pada tahap prediksi untuk mengestimasi kovariansi errornya.

Secara umum untuk Algoritma EnKF juga terdiri dari dua tahapan yaitu tahap prediksi (*time update*) dan tahap koreksi (*measurement update*). Pada metode EnKF sebelum memasuki tahap prediksi dihitung terlebih dahulu mean *ensemble*-nya, yaitu:

$$\hat{x}_k^- = \frac{1}{N_e}\sum_{i=1}^{N_e}(x_{k,i}) \tag{2.19}$$

dimana N_e adalah banyaknya *ensemble* yang dibangkitkan sedangkan $x_{k,i}$ merupakan nilai *ensemble* yang dibangkitkan.

Proses estimasi pada metode EnKF dengan proses estimasi KF terdapat perbedaan. Bentuk umum sistem dinamik nonlinier pada metode EnKF adalah:

$$x_{k+1} = f(u_k, x_k) + w_k \tag{2.20}$$

dengan pengukuran linier $y_k \in \mathcal{R}^p$ yaitu:

$$y_k = H_k x_k + v_k \tag{2.21}$$

$$x_0 \backsim N(\bar{x}_0, P_{x_0}); w_k \backsim N(0, Q_k); v_k \backsim N(0, R_k)$$

misalkan akan dibangkitkan sejumlah ensemble (N_e) untuk memperoleh nilai rata-rata (*mean*) :

$$x_{0,i} = [x_{0,1}, x_{0,2}, x_{0,3}, \dots \dots, x_{0,N_e}]$$

Mean ensemble yang diperoleh dari Persamaan (2.19) digunakan untuk menghitung estimasi $\hat{x}_k^-$ pada tahap prediksi (*time update*) dan $\hat{x}_k$ pada tahap koreksi (*measurement update*). Sedangkan pada tahap prediksi untuk menghitung kovariansi error P_k^- menggunakan

$$P_k = \frac{1}{N_e - 1} \sum_{i=1}^{N_e} (\hat{x}_{k,i}^- - \hat{x}_k^-)(\hat{x}_{k,i}^- - \hat{x}_k^-)^T \qquad (2.22)$$

Pada metode *Ensemble Kalman Filter*, *noise* pada tahap prediksi (w_k) dan noise pada tahap koreksi (v_k) dibangkitkan dalam bentuk *ensemble*. Berikut algoritma metode EnKF dalam mengestimasi penyelesaian model pada persamaan (2.20) dan (2.21) dapat dilihat pada Tabel 2.2

Tabel 2. 2 Algoritma Ensemble Kalman Filter (EnKF)

Model Sistem dan Model Pengukuran
$x_{k+1} = f(u_k, x_k) + w_k, w_k \backsim N(0, Q_k)$ $y_k = H_k x_k + v_k, v_k \backsim N(0, R_k)$
Inisialisasi
Bangkitkan N_e ensemble sesuai estimasi awal $\bar{x}_0$ $x_{0,i} = [x_{0,1}, x_{0,2}, x_{0,3}, \dots \dots, x_{0,N_e}]$ Tentukan nilai awal $\hat{x}_0 = \frac{1}{N_e} \sum_{i=1}^{N_e} (x_{0,i})$
Tahap Prediksi
$\hat{x}_{k,i}^- = f(u_{k-1}, x_{k-1}) + w_{k,i}, w_{k,i} \backsim N(0, Q_k)$

Estimasi:

$$\hat{x}_k^- = \frac{1}{N_e}\sum_{i=1}^{N_e} \hat{x}_{k,i}^-$$

Kovariansi Error:

$$P_k^- = \frac{1}{N_e - 1}\sum_{i=1}^{N_e} (\hat{x}_{k,i}^- - \hat{x}_k^-)(\hat{x}_{k,i}^- - \hat{x}_k^-)^T$$

Tahap Koreksi

$$y_{k,i} = y_k + v_{k,i}\,, v_{k,i} \backsim N(0, R_k)$$

Kalman Gain:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}$$

Estimasi:

$$\hat{x}_{k,i} = \hat{x}_{k,i}^- + K_k(y_{k,i} - H_k \hat{x}_{k,i}^-)$$

$$\hat{x}_k = \frac{1}{N_e}\sum_{i=1}^{N_e} \hat{x}_{k,i}$$

Kovariansi Error:

$$P_k = [I - K_k H_k]\, P_k^-$$

Keterangan:

$\hat{x}_k^-$: estimasi variabel keadaan pada tahap prediksi

P_k^- : kovariansi error pada tahap prediksi

(Purnomo, 2008).

2.7 Norm Kovariansi Error

Norm matriks kovariansi error berkaitan dengan kesimpulan baik tidaknya metode KF digunakan dalam pengestimasian. Hal ini sebagai tolak ukur dalam mengestimasi suatu model pada metode Kalman Filter. Apabila semakin kecil norm kovariansi error, maka hasil estimasi mempunyai tingkat keakuratan yang semakin tinggi.

Norm matriks pada himpunan S memuat matriks-matriks berukuran n x n dengan notasi ||.|| atau bisa disebut juga panjang atau besar yang merupakan fungsi bernilai real yang terdefinisi pada S dan memenuhi:

(i) $\|A\| \geq 0$,

(ii) $\|A\| = 0$, jika dan hanya jika $A = 0$

(iii) $\|\alpha A\| = |\alpha|\ \|A\|$, untuk semua bilangan real α

(iv) $\|A+B\| \leq \|A\|+\|B\|$

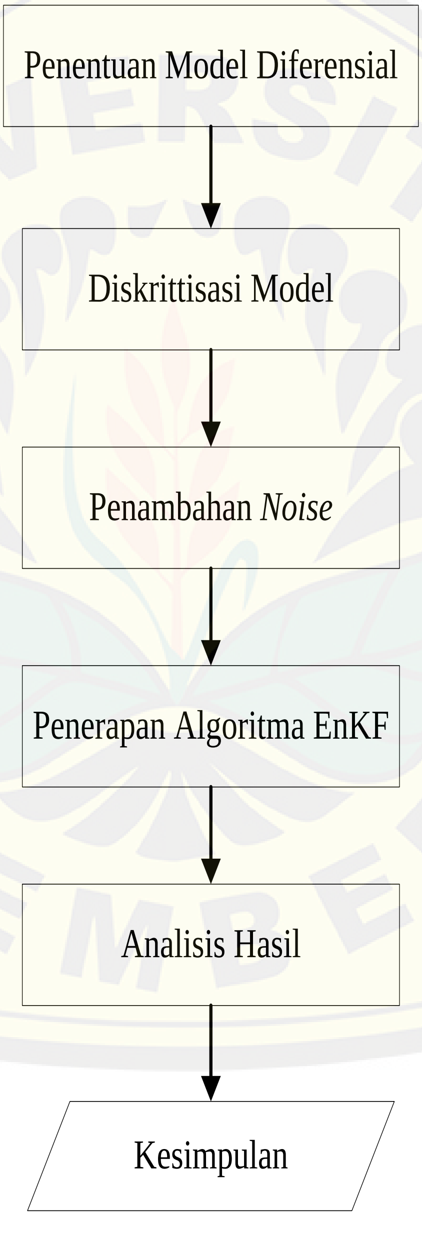
(v) $\|AB\| \leq \|A\|\ \|B\|$,

untuk semua $A, B \in S$ (Purnomo, 2008).

Secara umum matriks kovariansi error merupakan hubungan antara error dari masing-masing variabel dalam estimasi. Besaran matriks kovariansi error dinyatakan dengan nilai normnya. Dalam permasalahan ini, norm kovariansi matriks didefinisikan sesuai norm matriks yang digunakan dalam Matlab R2015b. Bentuk matriks kovariansi error didefinisikan sebagai norm *(P)*, maka $\|P\|$ merupakan besarnya matriks kovariansi error.

BAB 3. METODE PENELITIAN

Pada bab ini akan dijelaskan langkah-langkah yang digunakan untuk mendapatkan estimasi yang tepat. Berikut beberapa langkah yang akan diuraikan dalam mengestimasi pertumbuhan logistik, diantaranya adalah:



Gambar 3. 1 Skema Langkah-langkah Penelitian

3.1 Penentuan Model Persamaan Diferensial dan Solusi Numerik

Permasalahan yang akan dibahas dalam model pertumbuhan logistik ini adalah model pertumbuhan logistik dengan fungsi parabolik. Berdasarkan persamaan (2.4) dan (2.9) diperoleh persamaan diferensial untuk pertumbuhan logistik yaitu:

$$\frac{dN(t)}{dt} = a(N(t) - b)^2 N(t) \tag{3.1}$$

dimana untuk a dan b adalah konstanta dengan kondisi awal $N(t) = N_0$.
Setelah mendapatkan model persamaan diferensial maka selanjutnya mencari solusi numerik dari persamaan pertumbuhan logistik tersebut. Solusi numerik yang dicari menggunakan metode Runge-Kutta orde empat.

3.2 Melakukan Pendiskritan pada Model Pertumbuhan Logistik

Model populasi pada pertumbuhan logistik yang dipandang dalam masalah ini adalah masih berbentuk kontinu. Sehingga perlu dilakukan pendiskritan terhadap model tersebut guna mendapatkan model sistem diskrit. Untuk mendapatkan model sistem diskrit pada variabel keadaan N (jumlah populasi) terhadap waktu akan digunakan metode beda hingga dengan skema pendiskritan berupa beda hingga maju. Berdasarkan persamaan model pertumbuhan logistik (3.1) akan didiskritisasi dengan metode beda hingga maju, dengan panjang dari grid waktu adalah Δt, maka diperoleh:

$$N \approx N_k \tag{3.2}$$

$$\frac{dN(t)}{dt} \approx \frac{N_{k+1} - N_k}{\Delta t} \tag{3.3}$$

3.3 Penambahan *Noise*

Model diskrit yang telah diperoleh dari langkah sebelumnya selanjutnya akan diubah menjadi sistem stokastik. Perubahan sistem stokastik ditunjukkan dengan adanya penambahan noise. Penambahan noise dapat dilakukan dengan cara membangkitkan sejumlah bilangan *random* atau acak dari komputer melalui program Matlab. *Noise* yang dibangkitkan dalam langkah ini diasumsikan berdistribusi normal dan mempunyai *mean* nol karena merupakan *white noise*. Secara umum variansi *noise* diantaranya *noise* sistem (Q_k) dan *noise* pengukuran (R_k). Pada permasalahan ini variansi dari noise tersebut diasumsikan bernilai konstan.

3.4 Mengimplementasikan Metode *Ensemble Kalman Filter*

Sistem diskrit stokastik yang diperoleh dari langkah sebelumnya akan diimplementasikan pada metode *Ensemble Kalman Filter* pada Tabel 2.2. Dalam hal ini, variabel yang diestimasi pada pertumbuhan logistik fungsi parabolik adalah jumlah populasi (N). Beberapa input parameter yang mempengaruhi jumlah populasi diantaranya laju pertumbuhan (r) dan *carriying capacity* (K), dan besaran yang diinput adalah *time step* (Δt). Dalam prosesnya metode EnKF selalu terdapat *ensemble* yang dibangkitkan. Banyaknya *ensemble* yang dibangkitkan pada langkah ini dengan ukuran yang berbeda diantaranya 100, 200, 300, 400, 500, dan 1000. Pengambilan jumlah ensemble tersebut guna untuk melihat masing-masing ukuran ensemble yang cocok dalam mengestimasi pertumbuhan logistik.

3.5 Pembahasan Hasil Simulasi

Pada saat *output* hasil simulasi pada program Matlab R2015b dilakukanlah analisis terhadap output tersebut. Analisis dilakukan dengan melihat dan mengevaluasi ketepatan hasil estimasi dari metode EnKF yang kemudian dibandingkan dengan solusi numeriknya. Setelah mengevaluasi dapat diketahui apakah hasil estimasi tersebut mendekati solusi numeriknya atau tidak. Analisis yang dilakukan juga mengevaluasi rata-rata norm kovariansi *error* dan nilai rata-rata *error*. Dalam hal ini nilai error disini dilihat dari mutlak selisih diantara nilai numerik dengan nilai nilai estimasi yang relatif kecil.

Setelah menganalisis hasil, kemudian ditarik kesimpulan untuk hasil estimasi menggunakan metode tersebut. Penarikan kesimpulan dapat dilihat dari nilai *norm* kovariansi errornya. Jika nilai kovariansi error dan nilai rata-rata error yang semakin kecil, maka membuktikan bahwa metode EnKF dengan kurva parabolik terbuka ke atas pada Gambar 2.1 cocok atau tepat untuk estimasi pertumbuhan logistik. Tetapi apabila nilai kovariansi error dan nilai rata-rata error semakin besar, maka metode tersebut tidak cocok atau tidak tepat untuk mengestimasi variabel pada model pertumbuhan logistik.

BAB 5. KESIMPULAN DAN SARAN

Pada bab ini diperoleh kesimpulan dari hasil analisis dan pembahasan yang diperoleh dari penerapan metode *Ensemble Kalman Filter* pada model pertumbuhan logistik menggunakan fungsi populasi parabolik terbuka ke atas, serta diberikan saran yang dapat dilakukan sebagai kelanjutan skripsi ini.

5.1 Kesimpulan

Kesimpulan yang diperoleh berdasarkan hasil dan pembahasan adalah sebagai berikut:

1. Sesuai dengan hasil simulasi bahwa fungsi populasi parabolik terbuka ke atas bisa diterapkan di model pertumbuhan logistik menggunakan metode *Ensemble Kalman Filter*. Ini terlihat dari grafik yang dihasilkan dalam proses estimasi yang pergerakan potret fasenya membentuk seperti solusi analitik dari bentuk model pertumbuhan logistik.
2. Pada pengestimasian pertumbuhan logistik dengan metode *Ensemble Kalman Filter* menggunakan fungsi populasi parabolik terbuka ke atas, pengambilan jumlah $N_e = 100$ memberikan hasil estimasi yang baik. Ini terlihat dari hasil nilai rata-rata selisih mutlak yang relatif kecil sebesar 1,8086 dan juga adanya nilai rata-rata norm kovariansi *error* yang mempunyai *error* paling kecil diantara pengambilan *ensemble* 200, 300, 400, 500, dan 1000 yaitu sebesar 6,9277 x 10^{-5} .

5.2 Saran

Agar mendapatkan hasil yang lebih baik lagi dari hasil penelitian ini yang menggunakan kurva parabolik terbuka ke atas, penulis menyarankan:

1. Diperlukan kajian lebih lanjut tentang metode lain yang lebih cocok sehingga dapat digunakan pada proses estimasi model pertumbuhan logistik.
2. Penelitian selanjutnya diharapkan dapat menunjukkan variasi bentuk kurva untuk pengestimasian model pertumbuhan logistik.

DAFTAR PUSTAKA

Erna, A., Sanjoyo, A., dan Dieky, A. 2011. The Ground Water Pollution Estimation by Ensemble Kalman Filter. *Canadian Journal on Science and Engineering Mathematics*, Vol 2 No. 2:60-63.

Fitriani, V.N dan K.D.Purnomo. 2013. Solution Estimation of Logistic Growth Model with Ensemble Kalman Filter Method. *Jurnal Ilmu Dasar*. 14(2):85-90.

Hidayat, R. 2006. *Persamaan Diferensial Parsial*. Jember: UPT Penerbitan Universitas Jember

Masduqi, A. dan Apriliani, E. 2008. Estimation of Surabaya River Water Quality Using Kalman Filter Algorithm. *The Journal for Technology and Science*, 19 (3):87-91.

Purnomo, K. D. 2008. Estimasi Populasi Plankton dengan Ensemble Kalman Filter. *Jurnal Ilmu Dasar* 9(1).

Sunarsih & Hidayati, F. N. 2010. Model Pertumbuhan Logistik Predator dan Prey Pada Populasi Prey dan Solusi Kesetimbangan. *Jurnal Sains dan Matematika (JSM)*, Vol 18 No. 1:7-12.

Susanto, E. 2008. *Minimum Varians untuk Sistem Multi Input Multi Output (Mimo)*. Bandung: Departemen Teknik Elektro ITB.

Triatmodjo, B. 1992. *Metode Numerik*. Yogyakarta: Jurusan Teknil Sipil Fakultas Teknik Universitas Gadjah Mada.

Triatmodjo, B. 2002. *Metode Numerik Dilengkapi dengan Program Komputer* . Yogyakarta: Beta Offset

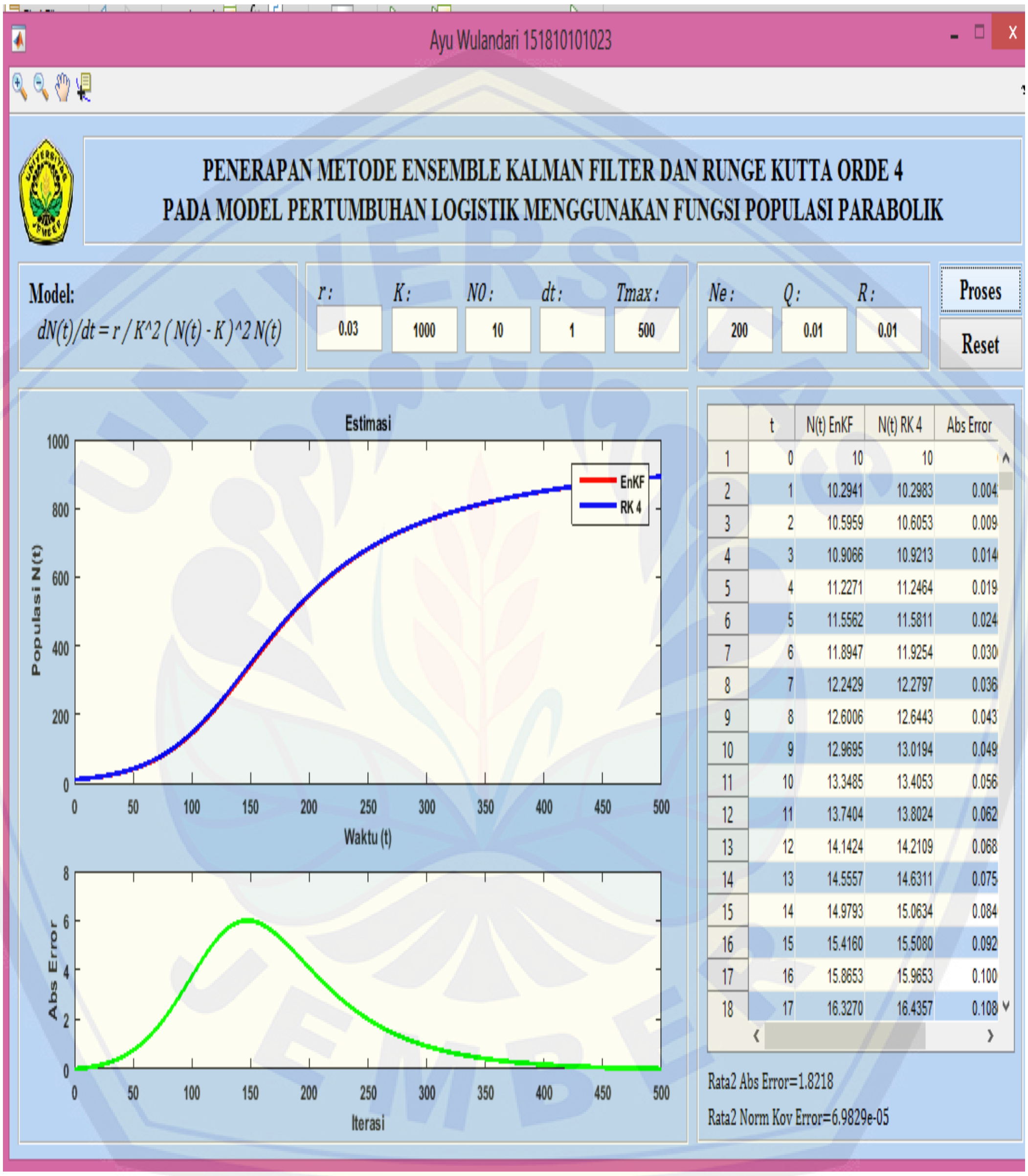
Welch, G. & Bishop, G. 2006. *An Introduction to the Kalman Filter*. Chapel Hill: Department of Computer Science University of North Carolina.

Ziil, D. G. 2012. *A First Course in Differential Equations with Modelling Application.* USA: Brooks/Cole Cengange Learning.

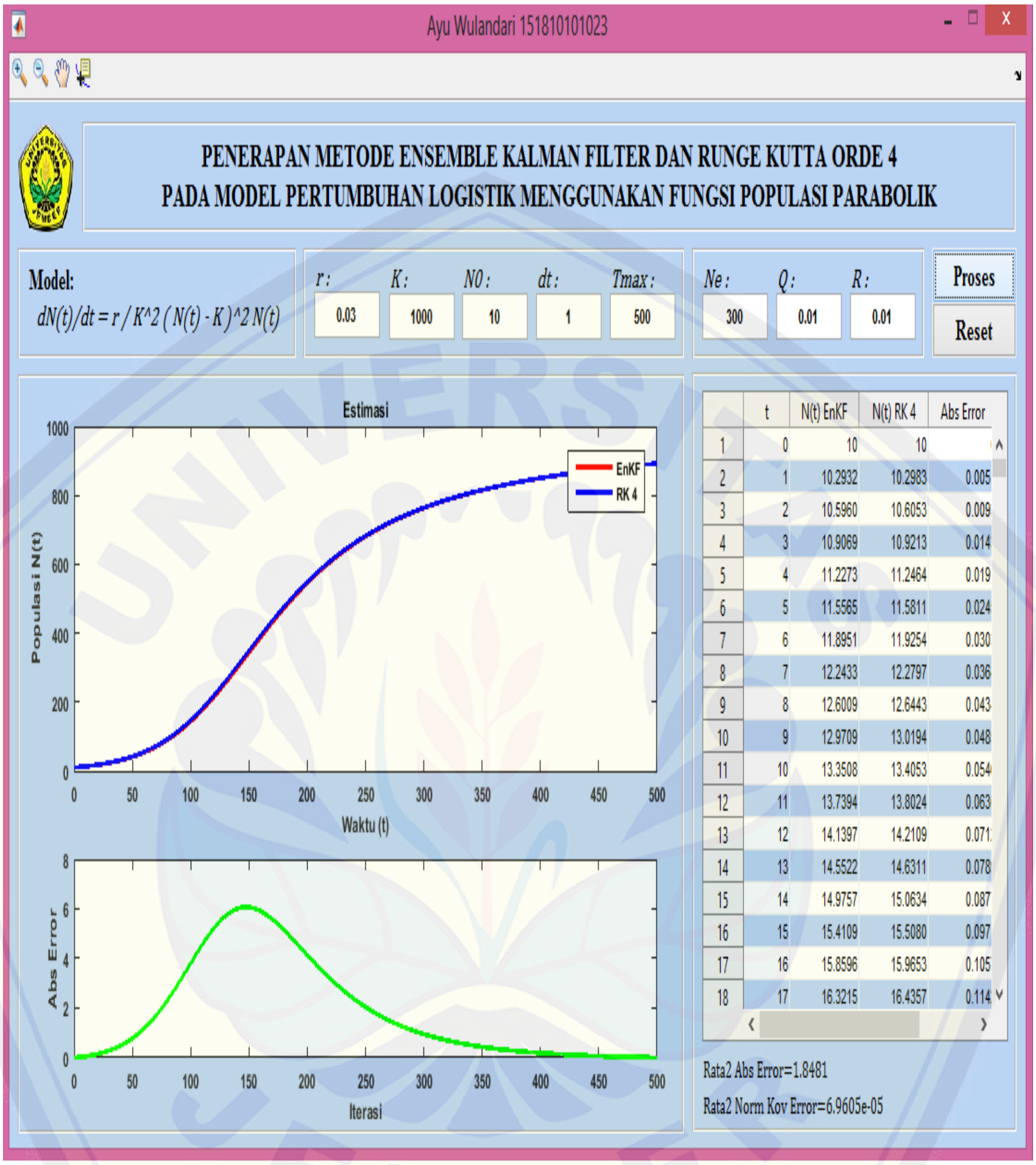
LAMPIRAN

A. Output Program

A.1 $N_e = 200$

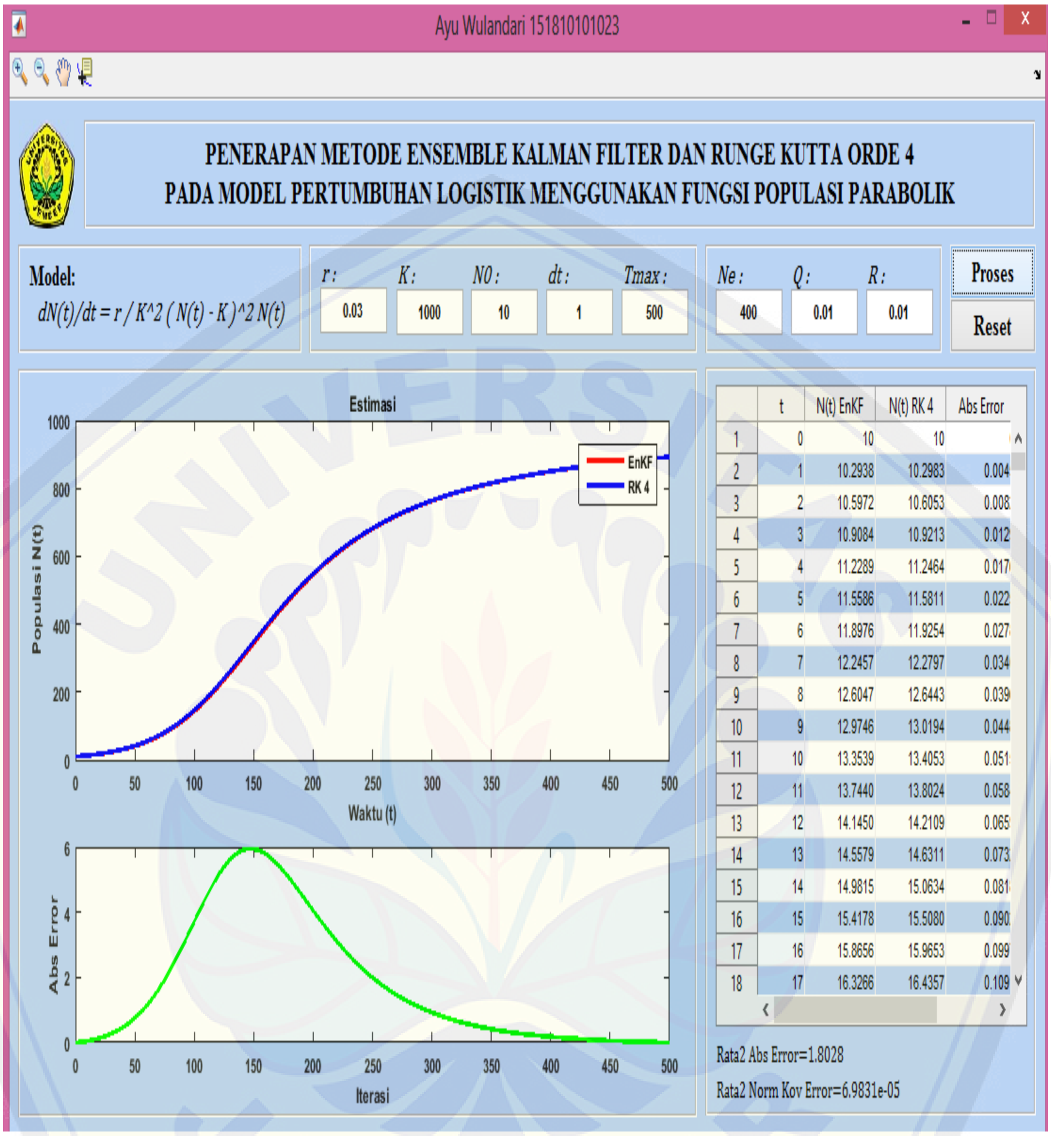


A.2 $N_e = 300$

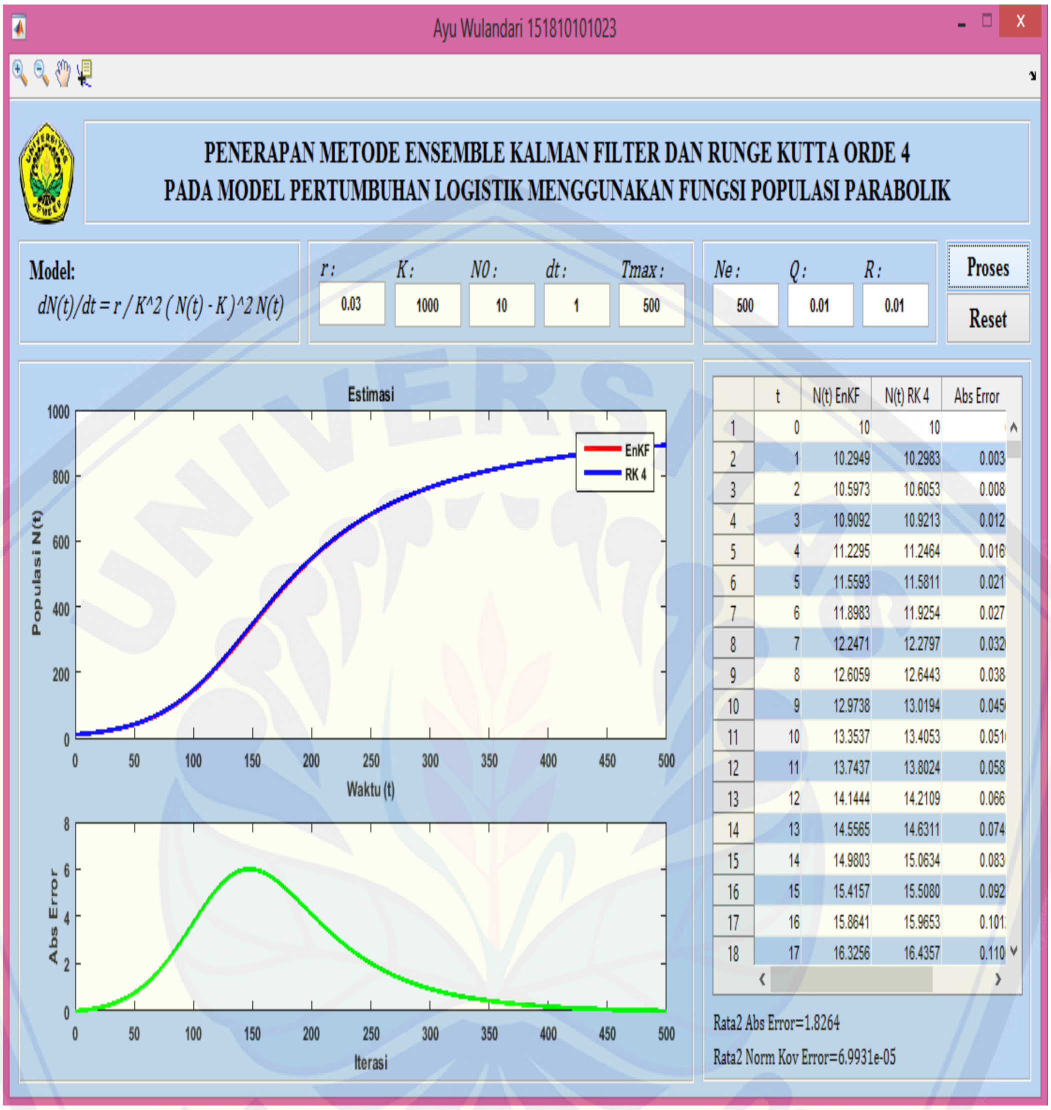


	t	N(t) EnKF	N(t) RK 4	Abs Error
1	0	10	10	
2	1	10.2932	10.2983	0.005
3	2	10.5960	10.6053	0.009
4	3	10.9069	10.9213	0.014
5	4	11.2273	11.2464	0.019
6	5	11.5565	11.5811	0.024
7	6	11.8951	11.9254	0.030
8	7	12.2433	12.2797	0.036
9	8	12.6009	12.6443	0.043
10	9	12.9709	13.0194	0.048
11	10	13.3508	13.4053	0.054
12	11	13.7394	13.8024	0.063
13	12	14.1397	14.2109	0.071
14	13	14.5522	14.6311	0.078
15	14	14.9757	15.0634	0.087
16	15	15.4109	15.5080	0.097
17	16	15.8596	15.9653	0.105
18	17	16.3215	16.4357	0.114

A.3 $N_e = 400$


Ayu Wulandari 151810101023
PENERAPAN METODE ENSEMBLE KALMAN FILTER DAN RUNGE KUTTA ORDE 4
PADA MODEL PERTUMBUHAN LOGISTIK MENGGUNAKAN FUNGSI POPULASI PARABOLIK
Model:
dN(t)/dt = r / K^2 (N(t) - K)^2 N(t)
r : 0.03
K : 1000
N0 : 10
dt : 1
Tmax : 500
Ne : 400
Q : 0.01
R : 0.01
Proses
Reset
Estimasi
EnKF
RK 4
Populasi N(t)
Waktu (t)
Abs Error
Iterasi
t | N(t) EnKF | N(t) RK 4 | Abs Error
1 | 0 | 10 | 10 |
2 | 1 | 10.2938 | 10.2983 | 0.004
3 | 2 | 10.5972 | 10.6053 | 0.008
4 | 3 | 10.9084 | 10.9213 | 0.012
5 | 4 | 11.2289 | 11.2464 | 0.017
6 | 5 | 11.5586 | 11.5811 | 0.022
7 | 6 | 11.8976 | 11.9254 | 0.027
8 | 7 | 12.2457 | 12.2797 | 0.034
9 | 8 | 12.6047 | 12.6443 | 0.039
10 | 9 | 12.9746 | 13.0194 | 0.044
11 | 10 | 13.3539 | 13.4053 | 0.051
12 | 11 | 13.7440 | 13.8024 | 0.058
13 | 12 | 14.1450 | 14.2109 | 0.065
14 | 13 | 14.5579 | 14.6311 | 0.073
15 | 14 | 14.9815 | 15.0634 | 0.081
16 | 15 | 15.4178 | 15.5080 | 0.090
17 | 16 | 15.8656 | 15.9653 | 0.099
18 | 17 | 16.3266 | 16.4357 | 0.109
Rata2 Abs Error=1.8028
Rata2 Norm Kov Error=6.9831e-05

A.4 $N_e = 500$


Ayu Wulandari 151810101023
PENERAPAN METODE ENSEMBLE KALMAN FILTER DAN RUNGE KUTTA ORDE 4
PADA MODEL PERTUMBUHAN LOGISTIK MENGGUNAKAN FUNGSI POPULASI PARABOLIK
Model:
dN(t)/dt = r / K^2 (N(t) - K)^2 N(t)
r : 0.03
K : 1000
N0 : 10
dt : 1
Tmax : 500
Ne : 500
Q : 0.01
R : 0.01
Proses
Reset
Estimasi
EnKF
RK 4
Populasi N(t)
Waktu (t)
Abs Error
Iterasi
t | N(t) EnKF | N(t) RK 4 | Abs Error
1 | 0 | 10 | 10 |
2 | 1 | 10.2949 | 10.2983 | 0.003
3 | 2 | 10.5973 | 10.6053 | 0.008
4 | 3 | 10.9092 | 10.9213 | 0.012
5 | 4 | 11.2295 | 11.2464 | 0.016
6 | 5 | 11.5593 | 11.5811 | 0.021
7 | 6 | 11.8983 | 11.9254 | 0.027
8 | 7 | 12.2471 | 12.2797 | 0.032
9 | 8 | 12.6059 | 12.6443 | 0.038
10 | 9 | 12.9738 | 13.0194 | 0.045
11 | 10 | 13.3537 | 13.4053 | 0.051
12 | 11 | 13.7437 | 13.8024 | 0.058
13 | 12 | 14.1444 | 14.2109 | 0.066
14 | 13 | 14.5565 | 14.6311 | 0.074
15 | 14 | 14.9803 | 15.0634 | 0.083
16 | 15 | 15.4157 | 15.5080 | 0.092
17 | 16 | 15.8641 | 15.9653 | 0.101
18 | 17 | 16.3256 | 16.4357 | 0.110
Rata2 Abs Error=1.8264
Rata2 Norm Kov Error=6.9931e-05

B. Skrip Program

```
function varargout = ProgramFix(varargin)

% PROGRAMFIX MATLAB code for ProgramFix.fig
%      PROGRAMFIX, by itself, creates a new PROGRAMFIX or raises
the existing
%      singleton*.
%
%      H = PROGRAMFIX returns the handle to a new PROGRAMFIX or
the handle to
%      the existing singleton*.
%
%      PROGRAMFIX('CALLBACK',hObject,eventData,handles,...) calls
the local
%      function named CALLBACK in PROGRAMFIX.M with the given
input arguments.
%
%      PROGRAMFIX('Property','Value',...) creates a new PROGRAMFIX
or raises the
%      existing singleton*.  Starting from the left, property
value pairs are
%      applied to the GUI before ProgramFix_OpeningFcn gets
called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to ProgramFix_OpeningFcn via
varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help ProgramFix

% Last Modified by GUIDE v2.5 08-Apr-2019 08:58:01
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @ProgramFix_OpeningFcn, ...
                   'gui_OutputFcn',  @ProgramFix_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
```

```
% --- Executes just before ProgramFix is made visible.
function ProgramFix_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to ProgramFix (see VARARGIN)

% Choose default command line output for ProgramFix
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
clc;
movegui(gcf,'center');
axes(handles.axes3);
imshow(imread('unej.jpg'));
set(handles.edit1,'string','');
set(handles.edit2,'string','');
set(handles.edit3,'string','');
set(handles.edit4,'string','');
set(handles.edit5,'string','');
set(handles.edit6,'string','');
set(handles.edit7,'string','');
set(handles.edit8,'string','');
cla(handles.axes1,'reset');
axes(handles.axes1);
xlabel('Waktu (t)'); ylabel('Populasi N(t)'); title('Estimasi');
set(handles.axes1,'FontSize',8,'FontWeight','bold');
cla(handles.axes2,'reset');
axes(handles.axes2);
xlabel('Iterasi'); ylabel('Abs Error');
set(handles.axes2,'FontSize',8,'FontWeight','bold');
set(handles.uitable1,'data',[],'rowname',1:4);
set(handles.text13,'string','Rata2 Abs Error=');
set(handles.text14,'string','Rata2 Norm Kov Error=');
% UIWAIT makes ProgramFix wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = ProgramFix_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;
```

```
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
clc;
cla(handles.axes1,'reset');
axes(handles.axes1);
xlabel('Waktu (t)'); ylabel('Populasi N(t)'); title('Estimasi');
set(handles.axes1,'FontSize',8,'FontWeight','bold');
cla(handles.axes2,'reset');
axes(handles.axes2);
xlabel('Waktu (t)'); ylabel('Abs Error');
set(handles.axes2,'FontSize',8,'FontWeight','bold');
set(handles.uitable1,'data',[],'rowname',1:4);
set(handles.text13,'string','Rata2 Abs Error=');
set(handles.text14,'string','Rata2 Norm Kov Error=');


%-----------------------------------------------------------------
---------
r=str2num(get(handles.edit1,'string'));
K=str2num(get(handles.edit2,'string'));
N0=str2num(get(handles.edit3,'string'));
dt=str2num(get(handles.edit4,'string'));
Tmax=str2num(get(handles.edit5,'string'));
Ne=str2num(get(handles.edit6,'string'));
Q=str2num(get(handles.edit7,'string'));
R=str2num(get(handles.edit8,'string'));



%-----------------------------------------------------------------
---------
X=repmat(N0,1,Ne);
Xhat(1)=mean(X,2);
Pk=Q; %kovariansi error
Npk(1)=Pk; %norm kovariansi error


%-----------------------------------------------------------------
---------
Nrk(1)=N0;

Tt=0:dt:Tmax;


% ----------------------------------------------------------------
---------
for t=1:length(Tt)-1
    %Prediksi ---------------------------------------------------
---------
    W=normrnd(0,Q,1,Ne); %noise estimasi
    X=repmat((r/K^2*(Xhat(t)-K)^2*Xhat(t))*dt+Xhat(t),1,Ne)+W;
    Xhat(t+1)=mean(X,2); %N(t+1) prediksi
```

```
    for i=1:Ne
        Pk=Pk+(X(i)-Xhat(t+1))*(X(i)-Xhat(t+1))';
    end
    Pk=Pk/(Ne-1); %kovariansi error

    %Koreksi ------------------------------------------------------
---------
    V=normrnd(0,Q,1,Ne); %noise pengukuran
    Vbar=mean(V,2);
    Z=repmat(ones(1)*Xhat(t+1),1,Ne)+V;
    Rk=0;
    for i=1:Ne
        Rk=Rk+(V(i)-Vbar)*(V(i)-Vbar)';
    end
    Rk=Rk/(Ne-1);
    Kk=Pk*ones(1)'*(ones(1)*Pk*ones(1)'+Rk)^-1;
    for i=1:Ne
        X(i)=X(i)+Kk*(Z(i)-ones(1)*X(i));
    end
    Xhat(t+1)=mean(X,2); %N(t+1) koreksi
    Pk=(eye(1)-Kk*ones(1))*Pk; %eye (matriks identitas )
    Npk(t+1)=norm(Pk);


%RungeKutta ---------------------------------------------------------
-----
    k1=r/K^2*(Nrk(t)-K)^2*Nrk(t);
    k2=r/K^2*(Nrk(t)+dt/2*k1-K)^2*(Nrk(t)+dt/2*k1);
    k3=r/K^2*(Nrk(t)+dt/2*k2-K)^2*(Nrk(t)+dt/2*k2);
    k4=r/K^2*(Nrk(t)+dt*k3-K)^2*(Nrk(t)+dt*k3);
    Nrk(t+1)=Nrk(t)+dt*(k1+2*k2+2*k3+k4)/6;


%AbsError ----------------------------------------------------------
-----
    AbsE=abs(Xhat-Nrk);


%Plotting ----------------------------------------------------------
-----
    axes(handles.axes1)

plot(0:dt:Tt(t+1),Xhat,'r',0:dt:Tt(t+1),Nrk,'b','Linewidth',2);
    xlabel('Waktu (t)'); ylabel('Populasi N(t)');
title('Estimasi');
    set(handles.axes1,'FontSize',8,'FontWeight','bold');
    axes(handles.axes2)
    plot(0:t,AbsE,'g','LineWidth',2);
    xlabel('Iterasi'); ylabel('Abs Error');
    set(handles.axes2,'FontSize',8,'FontWeight','bold');

    %Tabel --------------------------------------------------------
---------
```

```
set(handles.uitable1,'data',[(0:dt:Tt(t+1))',Xhat',Nrk',AbsE',Npk'
],'rowname','numbered');
    pause(0.01);
end


axes(handles.axes1)
plot(0:dt:Tt(t+1),Xhat,'r',0:dt:Tt(t+1),Nrk,'b','Linewidth',2);
legend('EnKF','RK 4');
xlabel('Waktu (t)'); ylabel('Populasi N(t)'); title('Estimasi');
set(handles.axes1,'FontSize',8,'FontWeight','bold');
set(handles.text13,'string',['Rata2 Abs Error='
num2str(mean(AbsE))]);
set(handles.text14,'string',['Rata2 Norm Kov Error='
num2str(mean(Npk))]);

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
clc;
movegui(gcf,'center');
set(handles.edit1,'string','');
set(handles.edit2,'string','');
set(handles.edit3,'string','');
set(handles.edit4,'string','');
set(handles.edit5,'string','');
set(handles.edit6,'string','');
set(handles.edit7,'string','');
set(handles.edit8,'string','');
cla(handles.axes1,'reset');
axes(handles.axes1);
xlabel('Waktu (t)'); ylabel('Populasi N(t)'); title('Estimasi');
set(handles.axes1,'FontSize',8,'FontWeight','bold');
cla(handles.axes2,'reset');
axes(handles.axes2);
xlabel('Iterasi'); ylabel('Abs Error');
set(handles.axes2,'FontSize',8,'FontWeight','bold');
set(handles.uitable1,'data',[],'rowname',1:4);
set(handles.text13,'string','Rata2 Abs Error=');
set(handles.text14,'string','Rata2 Norm Kov Error=');


function edit6_Callback(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of edit6 as text
%        str2double(get(hObject,'String')) returns contents of
edit6 as a double
```

```
% --- Executes during object creation, after setting all
properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function edit7_Callback(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)


% Hints: get(hObject,'String') returns contents of edit7 as text
%        str2double(get(hObject,'String')) returns contents of
edit7 as a double


% --- Executes during object creation, after setting all
properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called


% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)
% hObject    handle to edit8 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit8 as text
%        str2double(get(hObject,'String')) returns contents of
edit8 as a double
```

```
% --- Executes during object creation, after setting all
properties.
function edit8_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit8 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)


% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of
edit1 as a double

% --- Executes during object creation, after setting all
properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)


% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of
edit2 as a double
```

```
% --- Executes during object creation, after setting all
properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called


% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of
edit3 as a double


% --- Executes during object creation, after setting all
properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%        str2double(get(hObject,'String')) returns contents of
edit4 as a double
```

```
% --- Executes during object creation, after setting all
properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text
%        str2double(get(hObject,'String')) returns contents of
edit5 as a double


% --- Executes during object creation, after setting all
properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```