



**PERBANDINGAN METODE *EXTENDED KALMAN FILTER* DAN
UNSCENTED KALMAN FILTER PADA ESTIMASI MODEL
PREDATOR-PREY LOTKA-VOLTERRA**

SKRIPSI

Oleh
Abd. Muis Hadiano
NIM 081810101005

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2015**



**PERBANDINGAN METODE *EXTENDED KALMAN FILTER* DAN
UNSCENTED KALMAN FILTER PADA ESTIMASI MODEL
*PREDATOR-PREY LOTKA-VOLTERRA***

SKRIPSI

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

Oleh
Abd. Muis Hadiano
NIM 081810101005

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2015**

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Ibunda Masrudah Luluk AM, Ayahanda H. Abdul Chozin, adik tercinta Sitta Yulia MM dan Ahmad Syaihuddin TH, serta keluarga lainnya yang telah mendoakan dan memberi kasih sayang serta pengorbanan yang tidak bisa saya sebutkan satu persatu,
2. guru-guru taman kanak-kanak RA Perwanida, MI Sabilul Huda, SMPN 3 Darul Ulum Peterongan, SMAN 2 Jombang, dan dosen-dosen Universitas Jember.
3. Almamater Jurusan Matematika FMIPA Universitas Jember.

MOTTO

يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنْكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ

Artinya : “Allah akan meninggikan orang-orang yang beriman diantara kamu dan orang-orang yang diberi ilmu pengetahuan beberapa derajat”.
(Terjemahan Q.S. Al-Mujaadilah: 11)*)

إِنَّ مَعَ الْعُسْرِ يُسْرًا

Sesungguhnya sesudah kesulitan itu ada kemudahan
(Terjemahan Q.S. Alam Nasyrah: 6)**)

*) dan **) Departemen Agama Republik Indonesia. 1998. *Al Qur'an dan Terjemahannya*. Semarang: PT Kumudasmoro Grafindo.

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

nama : Abdul Muis Hadianto

NIM : 081810101005

menyatakan dengan ini sesungguhnya bahwa karya tulis ilmiah yang berjudul “Perbandingan Metode *Extended Kalman Filter* dan *Unscented Kalman Filter* pada Estimasi Model *Predator-Prey* Lotka-Volterra” 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 dikemudian hari pernyataan ini tidak benar.

Jember, Juni 2015

Yang menyatakan,

Abdul Muis Hadianto

NIM 081810101005

SKRIPSI

**PERBANDINGAN METODE *EXTENDED KALMAN FILTER* DAN
UNSCENTED KALMAN FILTER PADA ESTIMASI MODEL
*PREDATOR-PREY LOTKA-VOLTERRA***

Oleh

Abdul Muis Hadiano

NIM 081810101005

Pembimbing

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

Dosen Pembimbing Anggota : Drs. Rusli Hidayat, M.Sc.

PENGESAHAN

Skripsi berjudul “Perbandingan Metode *Extended Kalman Filter* dan *Unscented Kalman Filter* pada Estimasi Model *Predator-Prey* Lotka-Volterra” telah diuji dan disahkan pada:

hari, tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas
Jember.

Tim Penguji:

Dosen Pembimbing Utama,

Dosen Pembimbing Anggota,

Kosala Dwidja Purnomo, S.Si, M.Si.
NIP 196908281998021001

Drs. Rusli Hidayat, S.Si, M.Sc.
NIP 196610121993031001

Penguji I,

Penguji II,

Kusbudiono, S.Si, M.Si.
NIP 197704302005011001

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

Mengesahkan

Dekan,

Prof. Drs. Kusno, DEA, Ph.D.

NIP 196101081986021001

RINGKASAN

Perbandingan Metode *Extended Kalman Filter* dan *Unscented Kalman Filter* pada Estimasi Model *Predator-Prey* Lotka-Volterra; Abdul Muis Hadianto; 081810101005; 40 halaman; jurusan Matematika Fakultas MIPA Universitas Jember

Salah satu bentuk interaksi antara makhluk hidup adalah predasi atau saling memangsa. Hubungan ini terjadi antara mangsa (*prey*) dan pemangsa (*predator*). Model Lotka-Volterra merupakan tipe dari model *Predator-Prey* yang membahas masalah pemangsaan (*predation*) dan dirumuskan oleh Alfred J. Lotka (1925) dan Vito Volterra (1926). Model tersebut dapat diimplementasikan pada suatu permasalahan di bidang pertanian atau perkebunan, diantaranya untuk mengontrol populasi hama tanaman.

Salah satu cara untuk menyelesaikan model tersebut adalah dengan metode *Kalman Filter* (KF). *Kalman Filter* adalah metode yang pertama kali diperkenalkan oleh Rudolph E. Kalman (1960) untuk mengestimasi suatu sistem keadaan dan model pengukuran. Metode ini hanya dapat diimplementasikan pada model linier saja, sedangkan masalah yang ada tidak hanya berupa model linier tetapi juga non-linier. Salah satu algoritma yang dapat diimplementasikan dalam model non-linier adalah *Extended Kalman Filter* (EKF) dan *Unscented Kalman Filter* (UKF) yang merupakan modifikasi dari algoritma Filter Kalman.

Metode Kalman filter digunakan untuk mengestimasi model tersebut karena persamaan tersebut yang bersifat dinamik yang artinya bergantung pada waktu. Dan keuntungan dari KF sendiri yang apa bila didapat data baru, maka tidak perlu perubahan model karena model tersebut masih bisa digunakan lagi dan apabila ada gangguan diluar sistem maka gangguan itu tidak akan mempengaruhi model karena

gangguan tersebut merupakan *noise*. Data baru yang didapat bisa digunakan untuk menyempurnakan estimasi selanjutnya.

Tujuan dari penulisan tugas akhir ini adalah untuk mendapatkan hasil estimasi dari jumlah populasi mangsa dan pemangsa model *Predator-Prey* dengan menggunakan metode *Extended Kalman Filter* (EKF) dan *Unscented Kalman Filter* (UKF) dan mengetahui metode mana yang lebih baik untuk mengestimasi jumlah populasi mangsa dan pemangsa.

Untuk memperoleh hasil estimasi dan mengetahui metode yang terbaik, dilakukan beberapa langkah, yaitu diskritisasi dengan menggunakan metode beda hingga maju, lalu dilanjutkan dengan penambahan faktor stokastik berupa *noise* (*noise* sistem dan *noise* pengukuran) dengan membangkitkan suatu bilangan acak dari komputer. Setelah itu mengimplementasikan algoritma *Extended Kalman Filter* dan *Unscented Kalman Filter*. Langkah selanjutnya adalah menganalisis hasil simulasi. Hasil yang akan dianalisis yaitu populasi mangsa (Kutu Hijau) dan populasi pemangsa (Kumbang Merah). Analisis yang dilakukan yaitu dengan membandingkan nilai *norm* kovariansi *error* dari kedua metode.

Hasil dari simulasi, grafik populasi mangsa (Kutu Hijau) dan pemangsa (Kumbang Merah) dengan metode EKF lebih mendekati nilai numeriknya dibandingkan dengan metode UKF. Dari hasil simulasi juga dapat dilihat bahwa grafik mengalami perulangan setiap sekitar 40 hari, ini menunjukkan siklus hidup dari mangsa (Kutu Hijau) dan pemangsa (Kumbang Merah). Nilai *norm* kovariansi populasi mangsa (Kutu Hijau) dengan metode EKF sebesar 0,00071374 sedangkan dengan metode UKF sebesar 1,1238. Ini menunjukkan bahwa untuk estimasi populasi mangsa (Kutu Hijau) metode EKF lebih baik dibandingkan metode UKF. Nilai *norm* kovariansi populasi pemangsa (Kumbang Merah) dengan metode EKF sebesar 0,00071374 sedangkan dengan metode UKF sebesar 0,0031978. Ini menunjukkan bahwa untuk estimasi populasi pemangsa (Kumbang Merah), metode EKF lebih baik dibandingkan metode UKF. Dengan demikian dapat disimpulkan bahwa metode EKF

lebih baik daripada metode UKF dalam mengestimasi populasi mangsa dan populasi pemangsa pada model *Predator-Prey* Lotka-Volterra.

PRAKATA

Puji syukur kepada Allah SWT atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Perbandingan Metode *Extended Kalman Filter* dan *Unscented Kalman Filter* pada Estimasi Model *Predator-Prey* Lotka-Volterra”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan strata satu (S1) pada Jurusan Matematika Fakultas MIPA Universitas Jember.

Dalam menyelesaikan skripsi ini, penulis telah banyak mendapat bantuan dan dorongan dari berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Bapak Kosala Dwidja Purnomo, S.Si, M.Si., selaku Dosen Pembimbing Utama dan Bapak Drs. Rusli Hidayat, S.Si., M.Sc., selaku Dosen Pembimbing Anggota, yang telah meluangkan waktu, pikiran dan perhatian dalam penyusunan skripsi ini;
2. Ibu Agustina Pradjaningsih, S.Si, M.Si., dan Bapak Dr. Alfian Futuhul Hadi S.Si, M.Si., selaku Dosen Penguji yang telah memberi kritik dan saran dalam menyusun skripsi ini;
3. Bapak Kusbudiono, S.Si, M.Si., selaku Dosen Penguji sidang yang telah memberi kritik dan saran dalam menyusun skripsi ini;
4. Sitta Yulia MM, Ahmad Syaihuiddin TH, Conan Family (bayu, dayvis, indah, priani, yesi), arif, laily, semua Laskar Maliner 2008 serta semua teman-teman Jurusan Matematika FMIPA Universitas Jember yang telah memberikan dukungan positif selama penyusunan skripsi ini;
5. teman-teman kost “48c” (bisri, sandy, yuli, resa, estu, pandu, okky, ali, topek) yang selalu menemani kegalauan selama menyusun skripsi ini;
6. semua pihak yang tidak dapat disebutkan satu-persatu.

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

Jember, Juni 2015

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PERSEMBAHAN	ii
HALAMAN MOTTO	iii
HALAMAN PERNYATAAN	iv
HALAMAN PEMBIMBING	v
HALAMAN PENGESAHAN	vi
RINGKASAN	vii
PRAKATA	x
DAFTAR ISI	xii
DAFTAR TABEL	xiv
DAFTAR GAMBAR	xv
DAFTAR LAMPIRAN	xvi
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	3
1.3 Tujuan	3
1.4 Manfaat	4
BAB 2. TINJAUAN PUSTAKA	5
2.1 Model <i>Predator-Prey</i>	5
2.2 Penambahan Faktor Stokastik	7
2.3 Kalman Filter	8
2.4 Metode <i>Extended Kalman Filter (EKF)</i>	10
2.5 Metode <i>Unscented Kalman Filter (UKF)</i>	11
2.6 Diskritisasi Model	15
2.7 Norm Matriks Kovariansi <i>Error</i>	16

BAB 3. METODE PENELITIAN.....	18
BAB 4. HASIL DAN PEMBAHASAN.....	22
4.1 Diskritisasi Model <i>Predator-Prey</i> Lotka-Volterra.....	22
4.2 Penambahan Faktor Stokastik.....	23
4.3 Implementasi Algoritma EKF untuk Model Predator-Prey Lotka-Volterra.....	24
4.4 Implementasi Algoritma UKF untuk Model Predator-Prey Lotka-Volterra.....	25
4.5 Simulasi Program.....	28
4.5.1 Estimasi dengan Metode EKF.....	28
4.5.2 Estimasi dengan Metode UKF.....	32
4.6 Analisis Hasil Simulasi.....	37
BAB5. PENUTUP.....	41
DAFTAR PUSTAKA.....	42

DAFTAR TABEL

	Halaman
Tabel 4.1 Perbandingan Norm Kovariansi <i>Error</i> Mangsa.....	38
Tabel 4.2 Perbandingan Norm Kovariansi <i>Error</i> Pemangsa	40

DAFTAR GAMBAR

	Halaman
3.1 Skema metode penelitian	18
4.1 Hasil Estimasi Mangsa pada Metode EKF	29
4.2 Norm Kovariansi <i>Error</i> Mangsa pada Metode EKF.....	29
4.3 Hasil Estimasi Populasi Pemangsa pada Metode EKF	30
4.4 Norm Kovariansi <i>error</i> Pemangsa pada Metode EKF.....	31
4.5 Hasil Estimasi Populasi Mangsa (Kutu Hijau) dan Pemangsa (Kumbang Merah) pada Metode EKF.....	32
4.6 Hasil Estimasi Populasi Mangsa pada Metode UKF	33
4.7 Norm Kovariansi <i>Error</i> Mangsa Pada Metode UKF	33
4.8 Hasil Estimasi Populasi Pemangsa pada Metode UKF.....	34
4.9 Norm Kovariansi <i>error</i> Pemangsa pada Metode UKF	35
4.10 Hasil Estimasi Populasi Mangsa (Kutu Hijau) dan Pemangsa (Kumbang Merah) pada Metode UKF	36
4.11 Hasil Estimasi Populasi Mangsa pada Metode EKF dan UKF.....	37
4.12 Norm Kovariansi <i>error</i> Mangsa pada Metode EKF dan UKF.....	38
4.13 Hasil Estimasi Populasi Pemangsa pada Metode EKF dan Metode UKF	39
4.14 Norm Kovariansi <i>error</i> Pemangsa pada Metode EKF dan Metode UKF.....	40

DAFTAR LAMPIRAN

	Halaman
A. Skrip Program Metode EKF	43
B. Skrip Program Metode UKF	44

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Semua makhluk hidup yang ada di bumi tidak akan bisa hidup sendiri, karena semua makhluk membutuhkan makhluk lain untuk berinteraksi. Salah satu bentuk interaksi antara makhluk hidup adalah predasi atau saling memangsa. Hubungan ini terjadi antara mangsa (*prey*) dan pemangsa (*predator*). Hubungan ini sangatlah erat kaitannya, tanpa mangsa maka pemangsa tidak akan bisa bertahan hidup karena tidak adanya sumber makanan untuk melangsungkan hidupnya. Sedangkan jika tidak ada pemangsa, maka tidak akan ada pengatur populasi dari mangsa sehingga akan terjadi pelonjakan populasi mangsa.

Predasi dari dua populasi ini sangat penting untuk kelangsungan kehidupan manusia yang tergantung pada keseimbangan lingkungannya. Model Lotka-Volterra merupakan tipe dari model *Predator-Prey* yang membahas masalah pemangsaan (*predation*) dan dirumuskan oleh Alfred J. Lotka (1925) dan Vito Volterra (1926). Model tersebut dapat diimplementasikan pada suatu permasalahan di bidang pertanian atau perkebunan, diantaranya untuk mengontrol populasi hama tanaman. Salah satu cara untuk menyelesaikan model tersebut adalah dengan metode *Kalman Filter*. *Kalman Filter* adalah metode yang pertama kali diperkenalkan oleh Rudolph E. Kalman (1960) untuk mengestimasi suatu sistem keadaan dan model pengukuran. Metode ini hanya dapat diimplementasikan pada model linier. Namun pada kenyataannya, masalah yang ada tidaklah selalu berupa model linier tetapi juga berupa model non-linier. Sehingga perlu dikembangkan sebuah algoritma yang dapat diimplementasikan untuk model non-linier. Salah satu algoritma yang bisa digunakan adalah *Extended Kalman Filter* (EKF) yang merupakan modifikasi dari algoritma Filter Kalman. Dalam *Extended Kalman Filter*, sistem perlu dilinierisasi (apabila

sistem tidak linier), dan pendiskritan sistem (apabila sistem kontinu), dan beberapa tahapan lain (Ichwan, 2010).

Ichwan (2010) mengaplikasikan metode EKF untuk mengestimasi posisi dan kecepatan kapal selam pada model dinamika gerak kapal selam yang menghasilkan waktu respon yang cepat dan memiliki nilai *error* hasil estimasi yang relatif kecil, sehingga sangat membantu dalam mengoptimalkan kinerja dari kapal selam. Kemudian Wicaksono (2010) menerapkan algoritma EKF untuk mendeteksi waktu terjadinya kerak pada alat penukar panas dan hasilnya adalah kerak telah timbul atau terakumulasi saat memasuki waktu ke-400 satuan waktu. Hal tersebut terdeteksi dengan adanya penurunan nilai estimasi dari kualitas perpindahan panas pada fluida panas dan fluida dingin yang drastis dan juga perubahan pola pergerakan temperatur. Angga (2010) telah meneliti aplikasi model pemangsa-mangsa Lotka-Volterra dalam tujuannya untuk menerapkan formulasi model matematis analitis siklus fauna di beberapa area perkebunan wilayah Jember. Sedangkan Prayudi (2012) mengaplikasikan metode EKF untuk mengestimasi jumlah populasi mangsa dan jumlah populasi pemangsa pada model *Predator-Prey* dimana tingkat kepercayaan yang dihasilkan cukup baik yang ditunjukkan dengan hasil norm kovariansi *error* yang cukup kecil dan estimasinya mendekati nilai riilnya.

Selain metode EKF diatas, dikembangkan juga algoritma lain yang bisa digunakan untuk sistem dinamik non-linier yaitu *Unscented Kalman Filter* (UKF). *Unscented Kalman Filter* adalah salah satu metode untuk mengestimasi keadaan sistem nonlinier yang pertama kali diperkenalkan oleh Julier Simon J dan Uhlmann Jeffery K (2002). UKF disebut juga sebagai *Sigma Point Filter*. Karena ide utama dari UKF adalah untuk mengaproksimasi mean dan kovarian dari suatu himpunan titik-titik yang disebut sebagai *sigma point* yang didapatkan melalui *Unscented Transformation* pada suatu variabel stokastik awal yang mengalami transformasi.

Gumilar (2011) telah mengestimasi posisi peluru kendali pada lintasan menggunakan *Unscented Kalman Filter* sehingga bergerak sesuai lintasan yang diberikan dimana pergerakan peluru kendali sangat dipengaruhi oleh input sudut

serangnya. Juwariya (2013) mengaplikasikan metode EKF dan UKF untuk mengestimasi posisi misil di setiap rentang waktu pada lintasan yang telah ditentukan untuk bisa menuju target dengan tepat sasaran serta metode manakah yang tepat diantara metode EKF dan UKF dalam mengestimasi posisi misil dengan model yang telah ditentukan. Hasilnya posisi akhir estimasi dari posisi horisontal dan posisi vertikal terhadap target, EKF lebih mendekati target dibanding posisi akhir dari UKF. Hasbullah (2011) menerapkan *Unscented Kalman Filter* untuk mengestimasi keadaan pada persamaan air dangkal. Algoritma adaptive rank UKF dengan pengurangan rank sampai dengan 10 yang disimulasikan pada persamaan air dangkal dengan dimensi 20 (rank 20) ternyata memberikan hasil estimasi yang baik, pemotongan rank berpengaruh pada besarnya error dan waktu komputasinya.

Dari tulisan yang telah diuraikan diatas, penulis tertarik untuk melakukan penelitian tentang metode terbaik diantara metode *Extended Kalman Filter* (EKF) dan *Unscented Kalman Filter* (UKF) untuk mengestimasi jumlah populasi mangsa dan jumlah populasi pemangsa pada model *Predator-Prey*.

1.2 Rumusan Masalah

Rumusan masalah yang dibahas dalam tugas akhir ini adalah metode manakah yang lebih tepat diantara metode *Extended Kalman Filter* (EKF) dan *Unscented Kalman Filter* (UKF) untuk mengestimasi jumlah populasi mangsa dan pemangsa pada model *Predator-Prey*. Dalam hal ini yang akan diestimasi adalah populasi kumbang merah (pemangsa) dan kutu hijau (mangsa). Model yang digunakan dalam skripsi ini adalah model *Predator-Prey* Lotka-Volterra yang diambil dari Pradjaningsih & Hidayat (2011).

1.3 Tujuan

Tujuan dari penulisan tugas akhir ini adalah untuk mendapatkan hasil estimasi dari jumlah populasi mangsa dan pemangsa model *Predator-Prey* dengan menggunakan metode *Extended Kalman Filter* (EKF) dan *Unscented Kalman Filter*

(UKF) dan mengetahui metode mana yang lebih baik untuk mengestimasi jumlah populasi mangsa dan pemangsa.

1.4 Manfaat

Manfaat dari penulisan tugas akhir ini adalah memberikan informasi metode mana yang lebih baik diantara *Extended Kalman Filter* (EKF) dan *Unscented Kalman Filter* (UKF) untuk mengestimasi jumlah populasi mangsa dan pemangsa model *Predator-Prey* Lotka-Volterra dan dapat dijadikan pertimbangan dalam memilih metode yang akan digunakan dalam mengestimasi jumlah populasi mangsa dan pemangsa.

BAB 2. TINJAUAN PUSTAKA

Dalam bab ini akan diberikan dasar-dasar teori yang digunakan untuk mengestimasi jumlah populasi mangsa dan pemangsa pada model *Predator-Prey*. Teori tersebut meliputi model mangsa pemangsa (*Predator-Prey*), metode *Kalman Filter* (KF), metode *Extended Kalman Filter* (EKF), *Unscented Kalman Filter* (UKF). Teori ini akan menjadi dasar dalam pembahasan selanjutnya.

2.1 Model *Predator-Prey*

Model *Predator-Prey* adalah model yang membahas dua spesies yaitu pemangsa (*predator*) dan mangsa (*prey*). Ada beberapa jenis model *Predator-Prey* diantaranya:

1. Model *Predator-Prey* Lotka-Volterra

Model Lotka-Volterra merupakan tipe dari model *Predator-Prey* yang membahas masalah pemangsaan (*predation*) dan dirumuskan oleh Alfred J. Lotka (1925) dan Vito Volterra (1926). Berdasarkan model tersebut, dapat diketahui bahwa kedua spesies saling mempengaruhi secara signifikan. Khususnya jika terdapat spesies mangsa yang berlimpah, maka jumlah populasi pemangsa juga akan meningkat. Namun sebaliknya, kalau pertumbuhan spesies mangsa lambat maka populasi pemangsa akan banyak yang mati karena kekurangan makanan.

Untuk memodelkan interaksi tersebut, pertama harus diperhatikan tingkat pertumbuhan mangsa dan pemangsa jika tidak ada interaksi. Suatu spesies mangsa dapat tumbuh mengikuti pola eksponensial apabila diasumsikan tidak ada sekelompok pemangsa. Dalam hal ini, pertumbuhan spesies mangsa setiap saat dinotasikan dengan $x(t)$, yaitu

$$\frac{dx}{dt} = ax \quad (2.1)$$

dimana x menyatakan jumlah populasi mangsa, $a > 0$ adalah konstanta pertumbuhan, dan t adalah waktu (dalam hari). Sehingga solusi dari persamaan differensial diatas adalah $x(t) = x(0)e^{at}$, sehingga populasi akan tumbuh terus tanpa batas. Dalam hal ini diasumsikan bahwa persediaan makanannya cukup tak terbatas untuk spesies mangsa sehingga pertumbuhannya tak terbatas yang berarti tak ada spesies yang mati. Seperti halnya pada mangsa, pertumbuhan spesies pemangsa setiap saat dinotasikan dengan $y(t)$, yaitu

$$\frac{dy}{dt} = -bx \quad (2.2)$$

dimana y menyatakan jumlah populasi pemangsa dan b merupakan konstanta penurunan. Penurunan ini bisa terjadi karena tanpa adanya mangsa maka pemangsa akan mati karena kelaparan.

Selanjutnya akan disusun suatu model yang membahas hubungan interaksi antara mangsa dan pemangsa. Hubungan tersebut didasarkan fakta bahwa spesies mangsa akan dimakan oleh spesies pemangsa. Dimana pada akhirnya akan diperoleh model hubungan mangsa pemangsa berikut

$$\frac{dx}{dt} = ax - \alpha xy \quad (2.3)$$

$$\frac{dy}{dt} = -by + \beta xy \quad (2.4)$$

dimana α dan β adalah koefisien interaksi. Dalam hal ini α memberikan aksi penurunan dalam spesies mangsa karena populasi mangsa akan menjadi mangsa dari populasi pemangsa, sedangkan β memberikan aksi kenaikan dalam spesies pemangsa karena pemangsa telah mempunyai persediaan makanan. Sistem *otonomus* dan *nonlinier* pada persamaan (2.3) dan (2.4) dinamakan persamaan *Predator-Prey* Lotka-Volterra (Pradjaningsih dan Hidayat, 2011).

2. Model *Predator-Prey* Leslie-Gower Dengan Pemanenan

Model *Predator-Prey* Leslie-Gower adalah contoh lain dari persamaan yang membahas tentang hubungan pemangsa dan mangsa. Model ini terdiri dari populasi *predator* dan populasi *prey* yang dinyatakan sebagai

$$\frac{dP}{dt} = \left(r_2 - a_2 \frac{P}{H} \right) P - c_2 P$$

$$\frac{dH}{dt} = (r_1 - a_1 P - b_1 H) H - c_1$$

Dimana : r_1 = laju pertumbuhan *prey*,

r_2 = laju pertumbuhan *predator*,

a_1 = laju pemangsaan *predator* terhadap *prey*,

b_1 = koefisien interaksi antar *prey*,

c_1 = laju pemanenan *prey*,

c_2 = laju pemanenan *predator*,

dengan $0 < c_i < r_i$, $i = 1, 2$ (Huo et al, 2011).

2.2 Penambahan Faktor Stokastik

Model suatu sistem matematika dapat diklasifikasi menjadi beberapa hal. Salah satunya berdasarkan fenomena dari parameter-parameternya. Berdasarkan jenis ini, model matematika dibagi menjadi dua yaitu model deterministik dan model stokastik. Model deterministik adalah model yang memiliki parameter yang sudah pasti. Dalam hal ini parameter yang diamati dianggap tetap dan tidak memiliki sebaran sehingga hubungan yang diperoleh merupakan hubungan matematika yang bersifat fungsional seperti $y = f(x)$. Sedangkan model stokastik adalah model yang memiliki parameter yang tidak pasti. Dalam hal ini, hubungan yang diperoleh selain mengandung komponen fungsional juga mengandung adanya galat berupa peubah acak yang berdistribusi dengan sebaran tertentu. Misalnya fungsi $y = f(x) + e$ dengan $f(x)$ adalah komponen fungsional dan e adalah peubah acak yang berdistribusi normal (Tirta, 2009).

Suatu model deterministik dapat diubah menjadi model stokastik dengan cara menambahkan faktor stokastik (ketidakpastian). Misalkan diberikan model deterministik:

$$x_k = f(x_{k-1}, u_{k-1}) \quad (2.5)$$

Yang kemudian ditambahkan faktor stokastik dalam bentuk *noise* yang memiliki sebaran normal yaitu w_k . Sehingga persamaan (2.5) menjadi:

$$x_k = f(x_{k-1}, u_{k-1}) + w_k \quad (2.6)$$

Membangkitkan noise sistem dan noise pengukuran menggunakan Matlab 7.8.0 atau R2009a diekspresikan dengan formula $w_k = \text{normrnd}(0, \text{sqrt}(Q), 2, 1)$ yang menggambarkan kalau w_k adalah *noise* pengukuran diamana mean = 0, kovarian Q dan bentuk matriksnya 2 x 1 dan membangkitkan *noise* pengukuran dengan menggunakan formula $v_k = \text{normrnd}(0, \text{sqrt}(R), 2, 1)$, v_k adalah noise pengukuran diamana mean = 0, kovarian R dan bentuk matriksnya 2 x 1 .

2.3 Kalman Filter

Kalman Filter adalah filter prediksi rekursif berdasarkan pada penggunaan teknik ruang keadaan dan algoritma rekursif yang digunakan untuk mengestimasi keadaan sistem dinamik (Kleinbauer, 2004). Keunggulan Filter Kalman adalah kemampuan mengestimasi suatu keadaan berdasarkan data yang minim (Masduki dan Apriliani, 2008).

Metode Kalman Filter digunakan untuk mengestimasi variabel keadaan dari sistem dinamik stokastik linier diskrit yang secara umum dituliskan sebagai berikut

$$x_{k+1} = A_k x_k + B_k u_k + G_k w_k \quad (2.7)$$

yang meminimumkan kovarian kesalahan estimasi dengan pengukuran yang memenuhi

$$z_k = H_k x_k + v_k \quad (2.8)$$

$$x_0 \sim N(\bar{x}_0, P_{x_0}); w_k \sim N(0, Q); v_k \sim N(0, R) \quad (2.9)$$

Dimana A_k , B_k , G_k dan H_k adalah matriks koefisien yang berukuran bersesuaian, x_k adalah variabel keadaan saat waktu k, u_k adalah vektor masukan, z_k adalah vektor pengukuran, v_k adalah *noise* pengukuran yang bersebaran normal dengan mean $\bar{v}_k = 0$ dan kovariansi R, dan w_k adalah *noise* sistem yang bersebaran normal dengan mean $\bar{w}_k = 0$ dan kovariansi Q. Dalam hal ini variabel $w_k \sim N(0, Q)$ dan $v_k \sim N(0, R)$ diasumsikan tidak berkorelasi satu sama lain dan berdistribusi normal atau biasa disebut *white noise* (Ichwan, 2010).

Dalam proses estimasi Kalman Filter ada dua buah tahapan, yaitu tahap prediksi atau lebih dikenal dengan *time update* dan tahap koreksi atau lebih dikenal dengan *measurement update*. Dimana tahap prediksi dipengaruhi oleh model dinamika sistem sedangkan tahap koreksi dipengaruhi oleh informasi dari pengukuran. Kedua tahapan ini akan diulang terus-menerus sampai waktu yang ditentukan k.

Algoritma Kalman Filter secara umum adalah sebagai berikut.

Model sistem dan model pengukuran:

$$\begin{aligned}x_{k+1} &= A_k x_k + B_k u_k + G_k w_k \\z_k &= H_k x_k + v_k \\x_0 &\sim N(\bar{x}_0, P_{x_0}); w_k \sim (0, Q); v_k \sim N(0, R)\end{aligned}$$

Tahap inisialisasi:

$$\begin{aligned}\hat{x}_0 &= \bar{x}_0 \\P_0 &= P_{x_0}\end{aligned}$$

Tahap prediksi (*time update*)

$$\begin{aligned}\text{Estimasi:} \quad \hat{x}_{k+1}^- &= \mathbf{A}_k \hat{x}_k + \mathbf{B}_k \mathbf{u}_k \\ \text{Kovariansi error:} \quad P_k^- &= \mathbf{A}_k P_k \mathbf{A}_k^T + \mathbf{G}_k Q_k \mathbf{G}_k^T\end{aligned}$$

Tahap koreksi (*measurement update*)

$$\begin{aligned}\text{Kalman Gain:} \quad K_{k+1} &= P_{k+1}^- \mathbf{H}_{k+1}^T (\mathbf{H}_{k+1} P_{k+1}^- \mathbf{H}_{k+1}^T + R_{k+1})^{-1} \\ \text{Estimasi:} \quad \hat{x}_{k+1} &= \hat{x}_{k+1}^- + K_{k+1} (\mathbf{z}_{k+1} - \mathbf{H}_{k+1} \hat{x}_{k+1}^-) \\ \text{Kovariansi error:} \quad P_{k+1} &= [\mathbf{I} - K_{k+1} \mathbf{H}_{k+1}] P_{k+1}^-\end{aligned}$$

(Miftahuddin, 2011)

Nilai estimasi pada tahap Koreksi bergantung pada residual. Koefisien pembobotan pada residual itu yang dinamakan *Kalman Gain*. Tingkat ketelitian dihitung menggunakan persamaan kovarian *error* yang melibatkan matriks *Kalman Gain*. Matriks ini yang digunakan untuk meminimumkan nilai kovarian *error* (Welch & Bishop, 2006).

2.4 Metode *Extended Kalman Filter* (EKF)

Algoritma Kalman Filter dikembangkan untuk estimasi dalam bentuk rekursif dan linier. Tapi dalam kenyataannya masalah yang dihadapi tidak selalu berbentuk linier, karena itu dikembangkan algoritma yang bisa untuk mengestimasi masalah nonlinier. Salah satu algoritma itu adalah algoritma *Extended Kalman Filter* (EKF).

Misalkan diberikan sebuah model sistem dinamik stokastik nonlinier:

$$x_{k+1} = f(x_k, u_k) + w_k \quad (2.10)$$

dengan pengukuran nonlinier $z_k \in \mathfrak{R}^P$ yang memenuhi

$$z_k = h(x_k) + v_k \quad (2.11)$$

$$x_0 \sim N(\bar{x}_0, P_{x_0}); w_k \sim N(0, Q); v_k \sim N(0, R)$$

dalam hal ini variabel $w_k \sim N(0, Q)$ dan $v_k \sim N(0, R)$ diasumsikan tidak berkorelasi satu sama lain dan berdistribusi normal karena merupakan *white noise*.

Sebelum mengestimasi model diatas (2.10) dan (2.11) terlebih dahulu dilakukan linierisasi dengan mendefinisikan:

$$x_{k+1}^* = f(\hat{x}_k, u_k) \quad (2.12)$$

$$z_{k+1}^* = f(x_{k+1}^*) \quad (2.13)$$

$$\mathbf{A} = [A_{i,j}] = \left[\frac{\partial f_i}{\partial x_j}(\hat{x}_k, u_k) \right] \quad (2.14)$$

$$\mathbf{H} = [H_{i,j}] = \left[\frac{\partial h_i}{\partial x_j}(x_{k+1}^*) \right] \quad (2.15)$$

dimana \mathbf{A} dan \mathbf{H} adalah matriks Jacobi yang diperoleh dari penurunan f dan h terhadap x . Berdasarkan persamaan (2.12)–(2.15) dan pengertian deret Taylor, maka persamaan (2.10) dan (2.11) diaproksimasi ke bentuk linier menjadi:

$$x_{k+1} \approx x_{k+1}^* + \mathbf{A}(x_k - \hat{x}_k) + w_k \quad (2.16)$$

$$z_{k+1} \approx z_{k+1}^* + \mathbf{H}(x_{k+1} - x_{k+1}^*) + v_{k+1} \quad (2.17)$$

Persamaan (2.16) dan (2.17) sudah berbentuk persamaan linier, sehingga bisa digunakan dalam metode Kalman Filter. Modifikasi inilah yang disebut dengan metode *Extended Kalman Filter* (EKF) (Purnomo, 2008).

Algoritma *Extended Kalman Filter* (EKF) adalah sebagai berikut:

Model sistem dan model pengukuran:

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

$$z_k = Hx_k + v_k$$

$$x_0 \sim N(\bar{x}_0, P_0); w_k \sim N(0, Q); v_k \sim N(0, R)$$

Inisialisasi:

$$P_0 = P_{x_0}, \hat{x}_0 = \bar{x}_0$$

Tahap prediksi (*time update*):

$$\text{Kovariansi error: } P_{k+1}^- = AP_k A^T + Q \quad (2.18)$$

$$\text{dengan } A = \left[\frac{\partial f_i}{\partial x_j}(\hat{x}_k, u_k) \right]$$

$$\text{Estimasi: } \hat{x}_{k+1}^- = f(\hat{x}_k, u_k)$$

Tahap koreksi (*measurement update*):

$$\text{Kalman Gain: } K_{k+1} = P_{k+1}^- H^T (HP_{k+1}^- H^T + R_{k+1})^{-1} \quad (2.19)$$

$$\text{dengan } H = \left[\frac{\partial h_i}{\partial x_j}(x_{k+1}^*) \right]$$

$$\text{Kovariansi error: } P_{k+1} = [I - K_{k+1}H]P_{k+1}^- \quad (2.20)$$

$$\text{Estimasi: } \hat{x}_{k+1} = \hat{x}_{k+1}^- + K_{k+1}(z_{k+1} - H\hat{x}_{k+1}^-) \quad (2.21)$$

(Ichwan, 2010).

2.5 Metode *Unscented Kalman Filter*(UKF)

Unscented Kalman Filter (UKF) adalah estimator nonlinear yang secara khusus sesuai untuk masalah sistem nonlinear kompleks. Dalam proses estimasinya, UKF diawali dengan *unscented transformation* sebelum diimplementasikan ke dalam algoritma UKF itu sendiri. Dalam hal ini transformasi yang digunakan adalah *Augmented Unscented Transformation*. *Augmented Unscented Transformation* adalah Transformasi *Unscented* yang menggunakan variabel *state* yang diperbesar (*Augmented state*). Yuanxin *et al.* (2005) menyatakan bahwa *Augmented Unscented*

Transformation lebih baik dari *non-augmented Unscented Transformation* berdasar nilai MSE (*Mean Square Error*) yang lebih rendah.

Misalkan diberikan sebuah model sistem stokastik nonlinier dan model pengukuran:

$$x_{k+1} = f(x_k, u_x) + w_k \quad (2.22)$$

$$z_k = H(x_k, k) + v_k \quad (2.23)$$

Pada persamaan (2.22) dan (2.23) akan didapatkan suatu sistem yang diperbesar berdimensi $L = L_x + L_w + L_v$ dimana L_x adalah dimensi sistem, L_w adalah dimensi *noise* sistem, dan L_v adalah dimensi *noise* pengukuran.

$$x_k^{aug} = [x_k \quad w_k \quad v_k]^T \quad (2.24)$$

Dengan kovariannya sebagai berikut

$$P_k^{aug} = \begin{bmatrix} P_k^x & 0 & 0 \\ 0 & R_k^w & 0 \\ 0 & 0 & R_k^v \end{bmatrix} \quad (2.25)$$

Dengan x_k berdimensi L_x dan memiliki mean \bar{x} serta kovarian P_x , *noise* sistem w_k berdimensi L_w dengan kovariansi R_w , fungsi pengukuran z_k , matrik pengukuran H , dan *noise* pengukuran v_k berdimensi L_v dengan kovariansi P_v . Sistem yang didekati dengan *Unscented Transformation* membentuk matrik titik sigma sebanyak $2L+1$. Titik sigma tersebut terbentuk dari gabungan variabel *state* yang diperbesar (*augmented state*) dengan *noise* sistem dan *noise* pengukuran.

Unscented Transformation dari variable keadaan x_k^{aug} berdimensi L didekati dengan $2L+1$ titik sigma χ_i berdasarkan persamaan seperti dibawah ini:

$$X_k^0 = \bar{x}_{k-1}^{aug} \quad (2.26)$$

$$\chi_i = \bar{x}_{k-1}^{aug} + \left(\sqrt{(L + \lambda) P_{k-1}^i} \right), i = 1, \dots, L \quad (2.27)$$

$$\chi_i = \bar{x}_{k-1}^{aug} - \left(\sqrt{(L + \lambda) P_{k-1}^i} \right), i = L + 1, \dots, 2L \quad (2.28)$$

Dengan $\lambda = \varepsilon^2(L + \kappa) - L$ adalah parameter penskalaan. Sebuah konstanta ε yang digunakan untuk menentukan sebaran dari titik *sigma* di sekeliling x , dimana

$0 < \varepsilon < 1$. Konstanta κ adalah penskalaan tambahan, dimana nilai $\kappa \geq 0$. Nilai yang paling sering digunakan yaitu $\kappa = 0$ (Hasbullah, 2011).

Misalkan diberikan $y_k = f(x_k)$, dengan variabel keadaan yang telah diperbesar:

$$x_k^{aug} = [x_1 \quad x_2 \quad \dots \quad x_L]^T$$

Jika dinyatakan dalam bentuk matriks, titik-titik *sigma* diatas dapat dituliskan sebagai berikut :

$$X = [X_0 \quad X_1 \quad \dots X_L \quad X_{L+1} \quad X_{L+2} \quad \dots \quad X_{2L}]^T$$

maka penyebaran vektor *sigma* y_k adalah:

$$Y_i = f(X_i), \quad i = 0, \dots, 2L$$

Selanjutnya nilai *mean* dari y_k didekati dengan sampel dari *mean* yang terboboti dari titik sigma sebagai berikut,

$$\bar{y} \approx \sum_{i=0}^{2L} N_k^{(i)} Y_i \quad (2.29)$$

Demikian juga dengan kovarian y_k ,

$$P_y \approx \sum_{i=0}^{2L} N_c^{(i)} (Y_i - \bar{y}) (Y_i - \bar{y})^T \quad (2.30)$$

dengan bobot N_i diberikan oleh:

$$N_k^{(0)} = \frac{\lambda}{L + \lambda}$$

$$N_c^{(0)} = \frac{\lambda}{L + \lambda} + 1 - \varepsilon^2 + \delta$$

$$N_k^{(i)} = N_c^{(i)} = \frac{1}{2(L + \lambda)}$$

$$i = 1, \dots, 2L$$

(Wan & Merwe, 2000).

Algoritma *Unscented Kalman Filter* secara lengkap dituliskan sebagai berikut:

Model Sistem:

$$x_k = f(x_{k-1}, u_k) + w_k$$

Model Pengukuran:

$$z_k = \mathbf{H}(x_k, k) + v_k$$

Inisialisasi:

$$\hat{x}_0 = \bar{x}_0$$

$$P_{x_0} = P_{\bar{x}_0}$$

$$\hat{x}_0^a = \begin{bmatrix} \hat{x}_0 \\ 0 \\ 0 \end{bmatrix}$$

$$P_0^a = E[(x_0^a - \hat{x}_0^a)(x_0^a - \hat{x}_0^a)^T] = \begin{bmatrix} P_0^x & 0 & 0 \\ 0 & R_0^w & 0 \\ 0 & 0 & R_0^v \end{bmatrix}$$

Tahap Prediksi (*Time Update*)

a. Membangun matriks *sigma points*:

$$\mathbf{X} = \begin{bmatrix} \bar{x}_{k-1}^{aug} & \bar{x}_{k-1}^{aug} + \left(\sqrt{(L + \lambda)P_{k-1}^i} \right) \bar{x}_{k-1}^{aug} - \left(\sqrt{(L + \lambda)P_{k-1}^i} \right) \end{bmatrix}$$

dimana:

$$\lambda = \varepsilon^2(L + \kappa) - L$$

b. *Forecast*

Forecast untuk tiap-tiap *sigma point* dengan

$$X_k^f = f(X_{k-1}^x, X_{k-1}^w)$$

Forecast untuk vektor keadaan dan kovariannya

$$\bar{x}_k^f = E[f(X_k)] = \sum_{i=0}^{2L} N_k^{(i)} X_k^{(i)f}$$

$$P_k^{xf} = \sum_{i=0}^{2L} N_c^{(i)} \left[(X_k^{(i)f} - \bar{x}_k^f)(X_k^{(i)f} - \bar{x}_k^f)^T \right]$$

Tahap Koreksi (*Measurement Update*):

- a. Memetakan *sigma point* dalam ruang *forecast* pengukuran

$$Z_k^i = \mathbf{H}(X_k^x, X_k^v)$$

- b. Menghitung kovarian untuk mendapatkan kalman gain

$$z_k^f = \sum_{i=0}^{2L} N_i^{(m)} Z_k^i$$

$$P_k^{zz} = \sum_{i=0}^{2L} N_i^{(c)} [(Z_k^i - z_k^f)(Z_k^i - z_k^f)^T]$$

$$P_k^{xz} = \sum_{i=0}^{2L} N_i^{(c)} [(X_k^i - \bar{x}_k^f)(Z_k^i - z_k^f)^T]$$

$$K_k = P_k^{xz} (P_k^{zz})^{-1}$$

- c. Menghitung mean dan kovarian analisis

$$\bar{x}_k^a = \bar{x}_k^f + K_k (z_k - z_k^f)$$

$$P_k^a = P_k^{xf} - K_k P_k^{zz} K_k^T \quad (2.31)$$

- d. Membentuk *state* dan kovarian yang diperbesar untuk iterasi berikutnya.

(Wan & Merwe, 2000).

2.6 Diskritisasi Model

Persamaan model sistem dinamik *Predator-Prey* Lotka-Volterra diatas merupakan persamaan yang kontinyu. Untuk mengestimasi suatu sistem dinamik, diperlukan model persamaan dalam bentuk diskrit. Oleh karena itu diperlukan diskritisasi model. Proses diskritisasi dapat dilakukan dengan beberapa cara dan salah satunya adalah dengan menggunakan metode beda hingga maju. Misalkan jika $u = u(x)$ diekspansi menurut deret Taylor, maka:

$$u(x+h) = u(x) + h \frac{\partial}{\partial x} u(x) + \frac{h^2}{2!} \frac{\partial^2}{\partial x^2} u(x) + \dots \quad (2.32)$$

Maka diperoleh suatu sistem

$$u(x+h) - u(x) = h \frac{\partial}{\partial x} u(x) + O(h^2) \quad (2.33)$$

$$\frac{u(x+h) - u(x)}{h} \approx \frac{\partial}{\partial x} u(x) \quad (2.34)$$

Persamaan di atas disebut persamaan beda hingga maju. Jika menggunakan notasi beda hingga dengan $u(x = ih)$ dimana $h = \Delta x$ dan $i = 1, 2, 3, \dots$ maka diperoleh persamaan

$$\frac{\partial u}{\partial x} \approx \frac{u_{i+1} - u_i}{\Delta x}. \quad (2.35)$$

(Triatmodjo, 2002).

2.7 Norm Matriks Kovariansi Error

Dalam proses estimasi *Kalman filter* terdapat istilah *norm kovariansi error*. *Norm kovariansi error* itu sendiri berguna untuk mengevaluasi ketelitian hasil estimasi dari *kalman filter* dapat dilihat dari bentuk *norm kovariansi error*nya. Dengan demikian, *norm kovariansi error* digunakan sebagai tolak ukur dalam pengestimasi suatu model pada metode *Kalman Filter*.

Norm matriks pada himpunan S yang memuat matriks-matriks berukuran $n \times n$, yaitu dituliskan dengan notasi $\| \cdot \|$ atau biasanya sering disebut panjang/besar yang merupakan fungsi bernilai real dan positif (Autar K. Kaw dalam Purnomo, 2008). Norm matriks didefinisikan sebagai:

$$\|x\| = \sqrt{\sum_{i=1}^n \sum_{j=1}^n a_{ij}^2}$$

Kovarian dari dua variable acak x_1 dan x_2 didefinisikan sebagai:

$$cov(x_1, x_2) = E[(x_1 - \hat{x}_1)(x_2 - \hat{x}_2)]$$

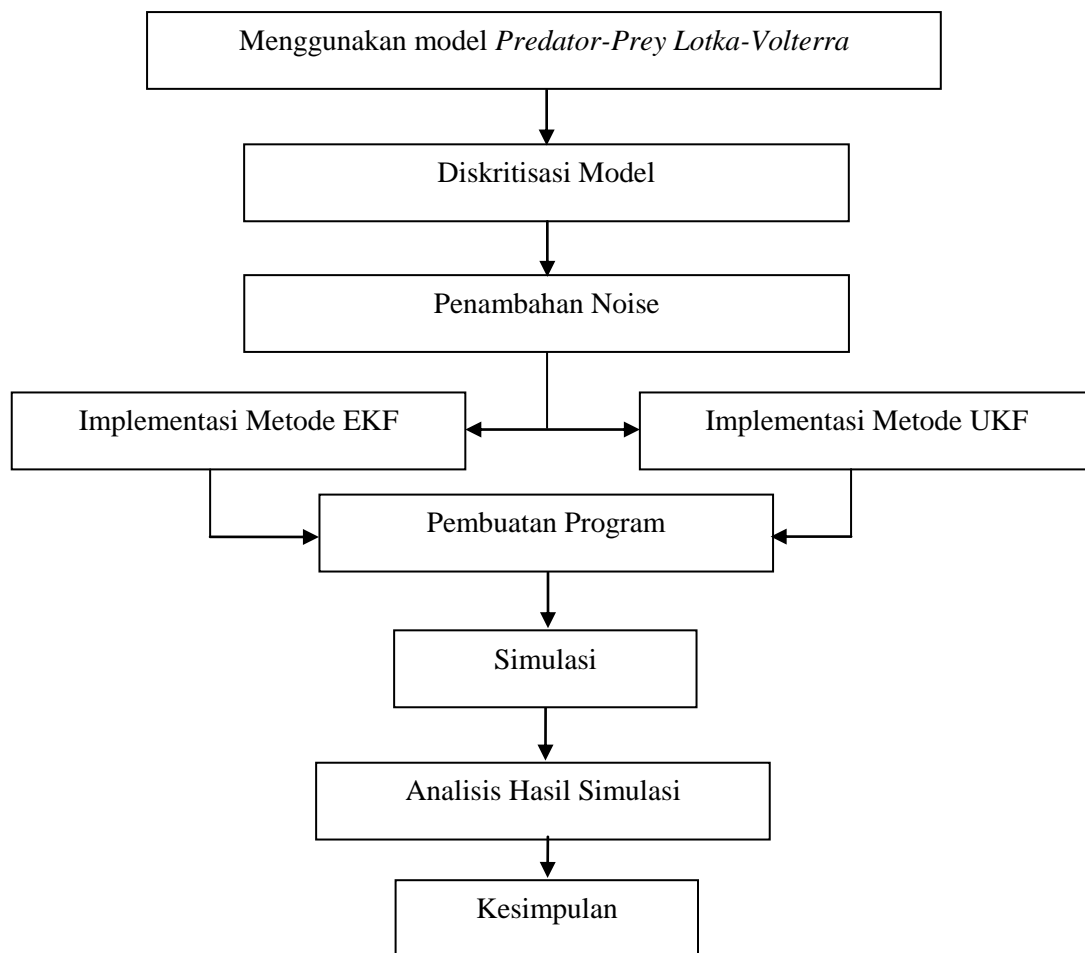
sedangkan kovarian dari vektor kolom $x = [x_1 \dots x_n]^T$ didefinisikan sebagai:

$$\text{cov}(x) = E[(x - \hat{x})(x - \hat{x})^T]$$

Yang merupakan matriks simetri $n \times n$ dan definit positif jika tidak ada kebergantungan linier dari komponen x . Kovarian matriks adalah elemen diagonal merupakan varian yang menunjukkan sebaran x , elemen selain pada diagonal merupakan kovarian yang juga menunjukkan korelasi atau hubungan antar variabel x (*independent/dependent*) (Kleeman dalam Fitriani, 2012). Dari definisi diatas dapat mendefinisikan *norm* dari matriks kovariansi *error*. Jadi, matriks kovariansi *error* merupakan hubungan antara *error* dari masing-masing variabel dalam proses estimasi. Kovariansi *error* didefinisikan sebagai P_k^a , sehingga norm kovariansi *error* dapat dituliskan sebagai $\|P_k^a\|$ yang merupakan besarnya matriks kovariansi *error*. Nilai dari $\|P_k^a\|$ sangat mempengaruhi hasil estimasi. Semakin besar nilainya maka semakin besar korelasi antara *error* variabelnya maka *error* yang dihasilkan akan semakin besar juga sehingga hasil estimasinya semakin jauh dari *realnya*, dan begitu pula sebaliknya.

BAB 3. METODE PENELITIAN

Bab ini menjelaskan tentang prosedur yang digunakan untuk menyelesaikan rumusan masalah yang akan dikaji dalam penelitian ini. Berikut adalah prosedur yang digunakan yaitu:



Gambar 3.1 Skema metode penelitian

- a. Menggunakan model *Predator-Prey Lotka-Volterra* yang bersumber dari Prajaningsih & Hidayat (2011).

$$\frac{dx}{dt} = ax - \alpha xy$$

$$\frac{dy}{dt} = -by + \beta xy$$

- Dimana: a : konstanta pertumbuhan
 b : konstanta penurunan
 x : jumlah populasi mangsa
 y : jumlah populasi pemangsa
 α dan β : koefisien interaksi

Model sistem persamaan yang dipakai diatas berupa model matematika klasik tentang interaksi dua spesies yang khusus membahas masalah *predation*, dimana bentuk modelnya adalah nonlinier dan kontinu. Masukan awal yang digunakan adalah rata-rata populasi mangsa dan rata-rata populasi pemangsa.

- b. Melakukan pendiskritan terhadap model Predator-Prey

Model sistem persamaan yang dibahas diatas masih berbentuk model kontinu, sehingga perlu dilakukan pendiskritan agar model bisa digunakan dalam algoritma EKF maupun UKF. Perubahan variabel keadaan x dan y terhadap waktu diaproksimasi dengan skema beda hingga maju. Selanjutnya akan menghasilkan model yang berbentuk diskrit.

$$f_{k+1} = \begin{bmatrix} M_{k+1} \\ N_{k+1} \end{bmatrix} = \begin{bmatrix} M_k + \Delta t(aM_k - \alpha M_k N_k) \\ N_k + \Delta t(-bN_k + \beta M_k N_k) \end{bmatrix}$$

- c. Penambahan faktor stokastik

Model diskrit pada persamaan diatas masih dalam bentuk deterministik sehingga belum dapat digunakan pada algoritma EKF dan UKF. Sehingga harus diubah ke dalam bentuk stokastik dengan cara menambahkan faktor stokastik berupa *noise*. Secara umum, *noise* tersebut disimbolkan dengan w_k dan v_k dimana kedua simbol

tersebut menunjukkan *noise* sistem dan *noise* pengukuran. Penambahan *noise* ini dilakukan dengan membangkitkan sejumlah bilangan acak dari komputer. *Noise* yang dibangkitkan diasumsikan memiliki sebaran normal dan *mean* nol karena merupakan *white noise*. Sedangkan variansi dari *noise* ini diasumsikan konstan sebesar Q_k dan R_k . Membangkitkan *noise* sistem dan *noise* pengukuran menggunakan Matlab 7.8.0 atau R2009a diekspresikan dengan formula $w_k = \text{normrnd}(0, \text{sqrt}(Q), 2, 1)$ dan membangkitkan *noise* pengukuran dengan menggunakan formula $v_k = \text{normrnd}(0, \text{sqrt}(R), 2, 1)$.

d. Mengimplementasikan algoritma EKF dan UKF

Model sistem yang diperoleh dari tahap sebelumnya diimplementasikan pada algoritma EKF dan UKF. Untuk EKF, sebelum diimplementasikan perlu adanya linearisasi dengan matriks Jacobi sehingga membentuk persamaan (2.14) dan (2.15). Berbeda dengan UKF yang tidak memerlukan linearisasi pada model dinamik *Predator-Prey* Lotka-Volterra, akan tetapi diawali dengan *unscented transformation* untuk mencari matriks titik sigma χ_k^{aug} yang sesuai dari variabel keadaan tambahan $x_k^{aug} = [x_k \ w_k \ v_k]^T$ berdasar persamaan (2.25) dan (2.26). Variabel keadaan yang diestimasi adalah jumlah populasi mangsa (x) serta jumlah populasi pemangsa (y). Dimana input nilai awal dari populasi mangsa (kutu hijau) $x = 216$ ekor, nilai awal dari populasi pemangsa (kumbang merah) $y = 6$ ekor, konstanta $a = 0,40$; $b = 0,12$; $\alpha = 0,11$; dan $\beta = 0,0032$ (Angga, 2010). Kemudian hasil yang diperoleh dari penggunaan algoritma EKF disimulasikan dengan program Matlab 7.8.0 atau R2009a.

e. Pembuatan Program

Software yang akan digunakan dalam pembuatan program adalah software MATLAB 7.8.0. Prosedur dalam pembuatan simulasi dari model *Predator-Prey* Lotka-Volterra adalah sebagai berikut:

1. Input nilai parameter
2. Proses

- a. Membuat subprogram untuk metode EKF
- b. Membuat subprogram untuk metode UKF

3. Output

Output yang dihasilkan dari simulasi ini berupa grafik jumlah populasi pemangsa dan mangsa, dan *norm kovariansi error* dari kedua metode yang dibandingkan.

f. Simulasi Program

Dalam langkah ini nilai-nilai parameter yang mempengaruhi jumlah populasi pemangsa dan mangsa diinputkan ke dalam program yang telah dibuat sebelumnya. Hal ini bertujuan untuk mengetahui grafik dari jumlah populasi pemangsa dan mangsa yang diimplementasikan dalam metode EKF dan UKF yang hasilnya akan dibandingkan untuk mengetahui keakuratan kedua metode tersebut dengan membandingkan nilai *normnya*. Metode dengan nilai *norm* lebih kecil adalah metode yang lebih baik dan akurat.

g. Analisis hasil simulasi

Hasil yang diperoleh dari simulasi, selanjutnya akan digunakan untuk menganalisis kedua metode tersebut dengan membandingkan hasil *norm kovariansi error* dari masing-masing metode. Formula yang digunakan dilakukan dengan membandingkan nilai *norm kovariansi error* dari metode EKF dan UKF adalah $\text{norm}()$. Metode terbaik akan memiliki *norm kovariansi error* yang lebih kecil.

BAB 4. HASIL DAN PEMBAHASAN

Model *Predator-Prey* Lotka-Volterra mempunyai bentuk berupa sistem nonlinier dan kontinu. Oleh karena itu, harus dilakukan diskritisasi menggunakan metode beda hingga maju sehingga didapatkan model *Predator-Prey* Lotka-Volterra dengan bentuk sistem nonlinier dan diskrit. Setelah didiskritisasi model masih dalam bentuk deterministik harus diubah menjadi model stokastik dengan cara menambahkan *noise* pada sistem dan pengukuran. Pada bagian akhir akan dibahas tentang hasil simulasi estimasi dengan metode EKF dan metode UKF.

4.1 Diskritisasi Model *Predator-Prey* Lotka-Volterra

Model *Predator-Prey* Lotka-Volterra mempunyai bentuk berupa sistem nonlinier ditunjukkan pada persamaan (2.3)-(2.4) sehingga harus kita lakukan diskritisasi agar dapat diimplementasikan ke dalam algoritma EKF dan UKF. Diskritisasi menggunakan metode beda hingga maju seperti pada persamaan (2.35). Jika populasi mangsa (Kutu Hijau) dimisalkan M dan pemangsa (Kumbang Merah) N , maka diperoleh aproksimasi variabel keadaan sistem terhadap waktu yaitu:

$$\dot{M} = \frac{dM}{dt} \approx \frac{M_{k+1} - M_k}{\Delta t} \quad (4.1)$$

$$\dot{N} = \frac{dN}{dt} \approx \frac{N_{k+1} - N_k}{\Delta t} \quad (4.2)$$

Persamaan (4.1) disubstitusikan ke persamaan (2.3), maka dihasilkan sebuah persamaan diskrit berikut

$$\frac{M_{k+1} - M_k}{\Delta t} = aM_k - \alpha M_k N_k \quad (4.3)$$

atau

$$M_{k+1} = M_k + \Delta t(aM_k - \alpha M_k N_k) \quad (4.4)$$

Begitu juga dengan persamaan (4.2), setelah disubstitusikan ke dalam persamaan (2.4) akan diperoleh

$$\frac{N_{k+1} - N_k}{\Delta t} = -bN_k + \beta M_k N_k \quad (4.5)$$

atau

$$N_{k+1} = N_k + \Delta t(-bN_k + \beta M_k N_k) \quad (4.6)$$

Persamaan (4.4) dan (4.6) dapat juga ditulis dengan notasi berikut

$$f_{k+1} = \begin{bmatrix} M_{k+1} \\ N_{k+1} \end{bmatrix} = \begin{bmatrix} M_k + \Delta t(\alpha M_k - \alpha M_k N_k) \\ N_k + \Delta t(-bN_k + \beta M_k N_k) \end{bmatrix} \quad (4.7)$$

Model diskrit pada persamaan (4.7) secara umum dapat dituliskan ke dalam bentuk fungsi nonlinier

$$x_{k+1} = f(x_k, u_k) \quad (4.8)$$

dimana $x_{k+1} = \begin{bmatrix} M_{k+1} \\ N_{k+1} \end{bmatrix}$

$$x_k = \begin{bmatrix} M_k \\ N_k \end{bmatrix}$$

$$u_k = 0$$

$$f(x_k, u_k) = f(x_k) = f_k$$

dengan u_k adalah vektor masukan deterministik. Model pengukuran yang digunakan dalam hal ini diasumsikan berbentuk linier, yaitu:

$$z_k = \mathbf{H}x_k \quad (4.9)$$

dengan z_k adalah vektor pengukuran dan H adalah sebuah matriks identitas.

4.2 Penambahan Faktor Stokastik

Model dinamik jumlah populasi mangsa-pemangsa pada persamaan (4.8) dan model pengukuran pada persamaan (4.9) masih berbentuk deterministik. Sehingga perlu ditambahkan suatu faktor stokastik yang berupa *noise*. Dengan demikian diperoleh model stokastik untuk model sistem dan model pengukuran sebagai berikut:

$$x_{k+1} = f(x_k, u_k) + w_k \quad (4.10)$$

$$z_k = \mathbf{H}x_k + v_k \quad (4.11)$$

Dengan $f(x_k, u_k)$ merupakan fungsi nonlinier yang didefinisikan pada persamaan (4.7) dan w_k merupakan *noise* sistem, sedangkan v_k merupakan *noise* pengukuran. Sebaran dari kedua *noise* ini diasumsikan berdistribusi normal dengan *mean* nol, dan varian dari masing-masing *noise* dinyatakan dengan Q dan R . Varian dari kedua *noise* diasumsikan konstan yaitu $Q = 10^{-3}$ dan $R = 10^{-3}$. Jika ditulis secara lengkap, maka persamaan (4.10) dan (4.11) mempunyai bentuk:

$$\begin{bmatrix} M_{k+1} \\ N_{k+1} \end{bmatrix} \approx \begin{bmatrix} M_k + \Delta t((aM_k - \alpha M_k N_k)) \\ N_k + \Delta t(-bN_k + \beta M_k N_k) \end{bmatrix} + \begin{bmatrix} w_k^x \\ w_k^y \end{bmatrix} \quad (4.12)$$

$$z_k = \mathbf{H} \begin{bmatrix} x_k \\ y_k \end{bmatrix} + \begin{bmatrix} v_k^x \\ v_k^y \end{bmatrix} \quad (4.13)$$

4.3 Implementasi Algoritma EKF

Sebelum mengimplementasikan algoritma EKF, terlebih dahulu perlu dihitung matriks Jacobi $A = [A_{i,j}]$ dari persamaan (4.12) sehingga matriks Jacobi A ditulis sebagai berikut

$$A = \begin{bmatrix} \frac{\partial M_{k+1}}{\partial M_k} & \frac{\partial M_{k+1}}{\partial N_k} \\ \frac{\partial N_{k+1}}{\partial M_k} & \frac{\partial N_{k+1}}{\partial N_k} \end{bmatrix} = \begin{bmatrix} 1 + \Delta t(a - \alpha N_k) & -\alpha M_k \Delta t \\ \beta N_k \Delta t & 1 + \Delta t(-b + \beta M_k) \end{bmatrix} \quad (4.14)$$

Sedangkan untuk matriks \mathbf{H} tidak dilinierkan dengan pencarian matriks Jacobi karena model pengukuran yang digunakan diasumsikan linier.

Langkah-langkah selanjutnya dalam metode EKF adalah sebagai berikut:

1. Linierisasi Model Sistem

Persamaan (4.12) dilinierkan menggunakan matriks Jacobi pada persamaan (4.14) sehingga menjadi:

$$\begin{bmatrix} M_{k+1} \\ N_{k+1} \end{bmatrix} = \begin{bmatrix} \frac{\partial M_{k+1}}{\partial M_k} & \frac{\partial M_{k+1}}{\partial N_k} \\ \frac{\partial N_{k+1}}{\partial M_k} & \frac{\partial N_{k+1}}{\partial N_k} \end{bmatrix} \begin{bmatrix} M_k \\ N_k \end{bmatrix} + \begin{bmatrix} w_k^M \\ w_k^N \end{bmatrix} \quad (4.15)$$

Secara umum, persamaan (4.15) yang telah linier dan diskrit dapat ditulis

$$x_{k+1} = \mathbf{A}x_k + w_k$$

2. Inisialisasi dari masing-masing variabel keadaan pada saat $k = 0$ untuk populasi Kutu Hijau $M = 216$, nilai awal dari populasi Kumbang Merah $N = 6$, konstanta $a = 0,40$; $b = 0,12$; $\alpha = 0,11$; dan $\beta = 0,0032$ (Angga, 2010)

3. Tahap prediksi

Pada tahap prediksi, matriks Jacobi yang digunakan yaitu persamaan (4.14). Nilai variabel yang diestimasi diperoleh dengan menggunakan persamaan

$$\hat{x}_{k+1}^- = \mathbf{A}\hat{x}_k + w_k$$

dan nilai kovariansi *error*-nya diperoleh dengan menggunakan persamaan

$$P_{k+1}^- = \mathbf{A}P_k\mathbf{A}^T + Q_k$$

4. Tahap koreksi

Pada tahap koreksi, dicari nilai *kalman gain* menggunakan persamaan berikut

$$K_k = P_k^- \mathbf{H}_k^T (\mathbf{H}_k P_k^- \mathbf{H}_k^T + R_k)^{-1}$$

nilai estimasi EKF merupakan nilai yang diperoleh dari

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - \mathbf{H}(\hat{x}_k^-, 0))$$

Sedangkan untuk kovarian *error*-nya diperoleh menggunakan persamaan berikut

$$P_k = [\mathbf{I} - K_k \mathbf{H}] P_k^-$$

5. Kembali ke langkah ke-3 untuk iterasi selanjutnya sampai waktu k yang diinginkan.

4.4 Implementasi Algoritma UKF

Model yang dipakai merupakan model dinamik populasi pemangsa dan mangsa (4.12) dan persamaan (4.13) sebagai model pengukuran. Selanjutnya didefinisikan sebagai sistem yang diperbesar (*augmented*) yang merupakan gabungan dari variabel keadaan, *noise* sistem dan *noise* pengukuran dalam bentuk matriks berikut.

$$x_k^{aug} = \begin{bmatrix} M_k \\ N_k \\ w_k \\ v_k \end{bmatrix}$$

Sedangkan kovarian dari sistem yang diperbesar, yaitu:

$$\mathbf{P}_k^{aug} = \begin{bmatrix} P_{MM} & 0 & 0 & 0 \\ 0 & P_{NN} & 0 & 0 \\ 0 & 0 & P_W & 0 \\ 0 & 0 & 0 & P_V \end{bmatrix}$$

Langkah-langkah selanjutnya dalam metode UKF adalah:

1. Inisialisasi dari masing-masing variabel dari sistem yang diperbesar pada saat $k = 0$

$$x_k^{aug} = \begin{bmatrix} M_k \\ N_k \\ W_k \\ v_k \end{bmatrix} = \begin{bmatrix} 216 \\ 6 \\ 0 \\ 0 \end{bmatrix}$$

dan kovarian yang diperbesar yaitu

$$\mathbf{P}_k^{aug} = \begin{bmatrix} P_{MM} & 0 & 0 & 0 \\ 0 & P_{NN} & 0 & 0 \\ 0 & 0 & P_W & 0 \\ 0 & 0 & 0 & P_V \end{bmatrix}$$

Dengan unscented transformation, akan dibentuk titik-titik sigma dari sistem yang telah diperbesar x_k^{aug} yang berdimensi-4 menggunakan persamaan (2.27) dan (2.28) sehingga diperoleh 9 titik sigma untuk $k = 1, 2, 3, \dots$

$$\begin{aligned} & \chi_{k+1}^{aug} \\ & = \begin{bmatrix} M_{k-1}^{aug} & M_{k-1}^{aug} + \left(\sqrt{(L + \lambda) P_{x_{k-1}}^{aug}} \right)_1 & M_{k-1}^{aug} + \left(\sqrt{(L + \lambda) P_{x_{k-1}}^{aug}} \right)_2 & M_{k-1}^{aug} + \left(\sqrt{(L + \lambda) P_{x_{k-1}}^{aug}} \right)_3 \\ M_{k-1}^{aug} & N_{k-1}^{aug} + \left(\sqrt{(L + \lambda) P_{x_{k-1}}^{aug}} \right)_1 & N_{k-1}^{aug} + \left(\sqrt{(L + \lambda) P_{x_{k-1}}^{aug}} \right)_2 & N_{k-1}^{aug} + \left(\sqrt{(L + \lambda) P_{x_{k-1}}^{aug}} \right)_3 \\ 0 & 0 & 0 & \left(\sqrt{(L + \lambda) P_w} \right)_3 \\ 0 & 0 & 0 & 0 \\ \\ M_{k-1}^{aug} + \left(\sqrt{(L + \lambda) P_{x_{k-1}}^{aug}} \right)_4 & M_{k-1}^{aug} - \left(\sqrt{(L + \lambda) P_{x_{k-1}}^{aug}} \right)_5 & M_{k-1}^{aug} - \left(\sqrt{(L + \lambda) P_{x_{k-1}}^{aug}} \right)_1 \\ N_{k-1}^{aug} + \left(\sqrt{(L + \lambda) P_{x_{k-1}}^{aug}} \right)_4 & N_{k-1}^{aug} - \left(\sqrt{(L + \lambda) P_{x_{k-1}}^{aug}} \right)_5 & N_{k-1}^{aug} - \left(\sqrt{(L + \lambda) P_{x_{k-1}}^{aug}} \right)_1 \\ 0 & 0 & 0 \\ \left(\sqrt{(L + \lambda) P_v} \right)_4 & 0 & 0 \end{bmatrix} \end{aligned}$$

$$\left[\begin{array}{ccc} M_{k-1}^{aug} - \left(\sqrt{(L + \lambda)P_{x_{k-1}}^{aug}} \right)_2 & M_{k-1}^{aug} - \left(\sqrt{(L + \lambda)P_{x_{k-1}}^{aug}} \right)_3 & M_{k-1}^{aug} - \left(\sqrt{(L + \lambda)P_{x_{k-1}}^{aug}} \right)_4 \\ N_{k-1}^{aug} - \left(\sqrt{(L + \lambda)P_{x_{k-1}}^{aug}} \right)_2 & N_{k-1}^{aug} - \left(\sqrt{(L + \lambda)P_{x_{k-1}}^{aug}} \right)_3 & N_{k-1}^{aug} - \left(\sqrt{(L + \lambda)P_{x_{k-1}}^{aug}} \right)_4 \\ 0 & \left(\sqrt{(L + \lambda)P_v} \right)_3 & 0 \\ 0 & 0 & \left(\sqrt{(L + \lambda)P_v} \right)_4 \end{array} \right]$$

2. Tahap prediksi

Pada tahap prediksi ini, titik-titik sigma yang diperoleh selanjutnya akan ditransformasikan melalui fungsi berikut

$$X_k^f = f(X_{k-1}^x, X_{k-1}^w)$$

Hasil transformasi dari titik-titik sigma tersebut akan digunakan untuk mencari mean dan kovarian pada tahap prediksi menurut persamaan berikut

$$\bar{x}_k^f = E[f(X_k)] = \sum_{i=0}^{2L} W_k^{(i)} X_k^{(i)f}$$

$$P_k^{x^f} = \sum_{i=0}^{2L} W_c^{(i)} \left[(X_k^{(i)f} - \bar{x}_k^f) (X_k^{(i)f} - \bar{x}_k^f)^T \right]$$

Titik-titik sigma diatas juga ditransformasikan melalui fungsi pengukuran dan dicari kovariannya

$$Z_k^i = \mathbf{H}(X_k^x, X_k^v)$$

$$z_k^f = \sum_{i=0}^{2L} W_i^{(m)} Z_k^i$$

3. Tahap koreksi

Pada tahap ini dicari kovariansi dari fungsi pengukuran:

$$P_k^{zz} = \sum_{i=0}^{2L} W_i^{(c)} [(Z_k^i - z_k^f)(Z_k^i - z_k^f)^T]$$

Sedangkan untuk menghitung kovarian silang dari sistem dan pengukuran:

$$P_k^{xz} = \sum_{i=0}^{2L} W_i^{(c)} [(X_k^i - \bar{x}_k^f)(Z_k^i - z_k^f)^T]$$

Langkah selanjutnya yaitu mencari *kalman gain*

$$K_k = P_k^{xz} (P_k^{xz})^{-1}$$

Kalman gain yang sudah diperoleh, akan digunakan menghitung nilai estimasi dan kovariansi analisis.

$$\hat{x}_k = \bar{x}_k^f + K_k (z_k - z_k^f)$$

$$P_k = P_k^{x^f} - K_k P_k^{zz} K_k^T$$

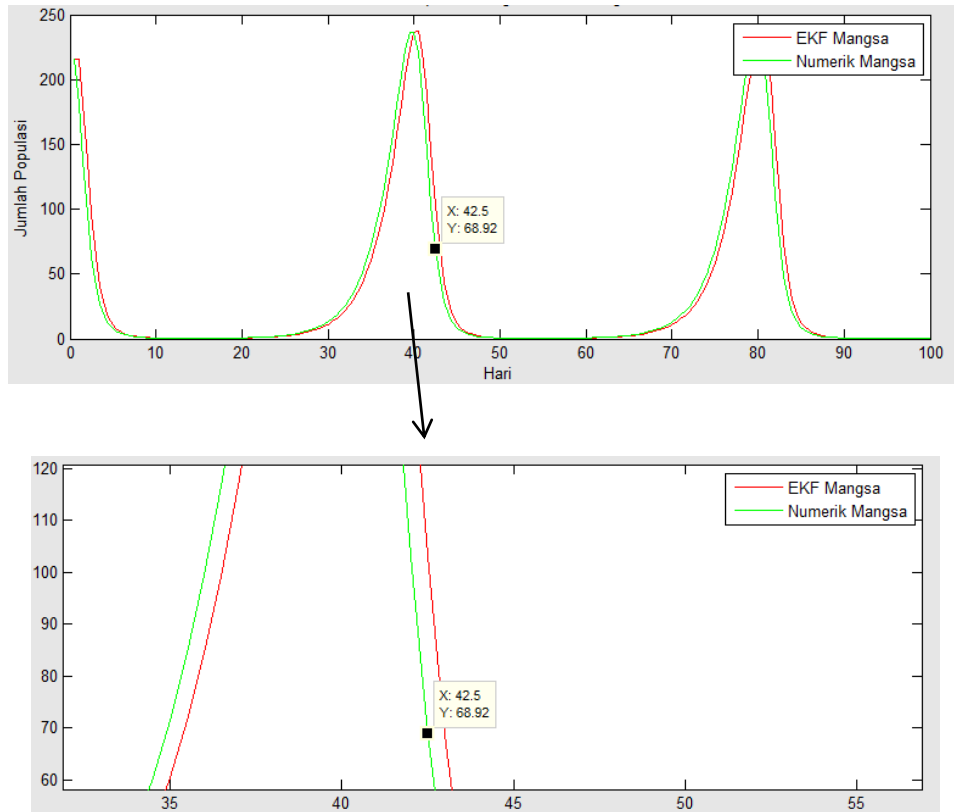
4. Kembali ke langkah 2 untuk iterasi selanjutnya sampai waktu k yang diinginkan.

4.5 Simulasi Program

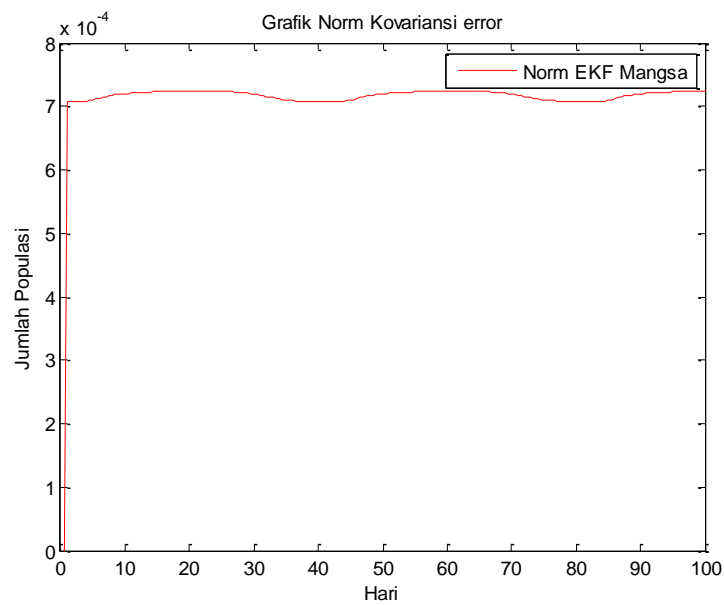
Simulasi dilakukan dengan menerapkan algoritma EKF dan UKF pada model *Predator-Prey* Lotka-Volterra. Sebelum melakukan simulasi, terlebih dahulu menentukan grid waktu (Δt). Dalam hal ini, grid waktu diambil $\Delta t = 0,5$. Sedangkan nilai awal yang akan digunakan adalah $M_0 = 216$, $N_0 = 6$, konstanta $a = 0,40$; $b = 0,12$; $\alpha = 0,11$; dan $\beta = 0,0032$. Hasil simulasi ini akan dievaluasi dengan cara membandingkan hasil estimasi *norm* kovariansi *error*. Simulasi dilakukan dengan menerapkan metode EKF terlebih dahulu dengan nilai awal sebesar $P_{M_0} = 0,006$ dan $P_{N_0} = 0,0025$ yang kemudian akan dibandingkan dengan hasil estimasi dari metode UKF.

4.5.1 Estimasi dengan metode EKF

Hasil estimasi populasi mangsa (kutu hijau) dan pemangsa (kumbang merah) dengan menggunakan metode EKF dapat dilihat pada gambar 4.1 sampai 4.5.

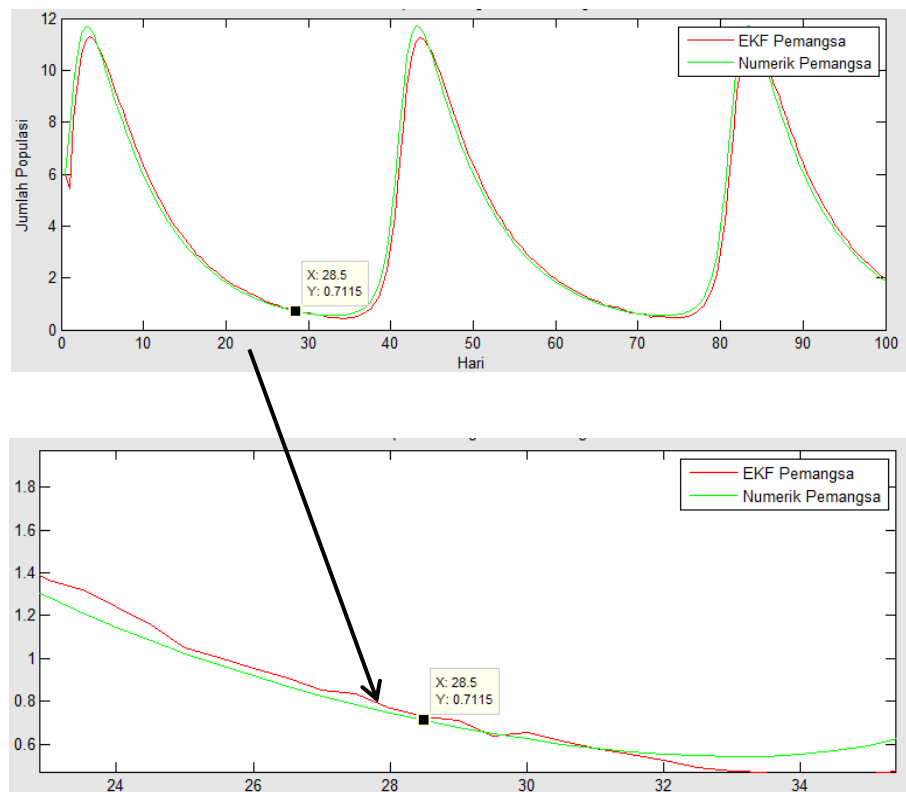


Gambar 4.1 Hasil Estimasi Mangsa (Kutu Hijau) pada Metode EKF

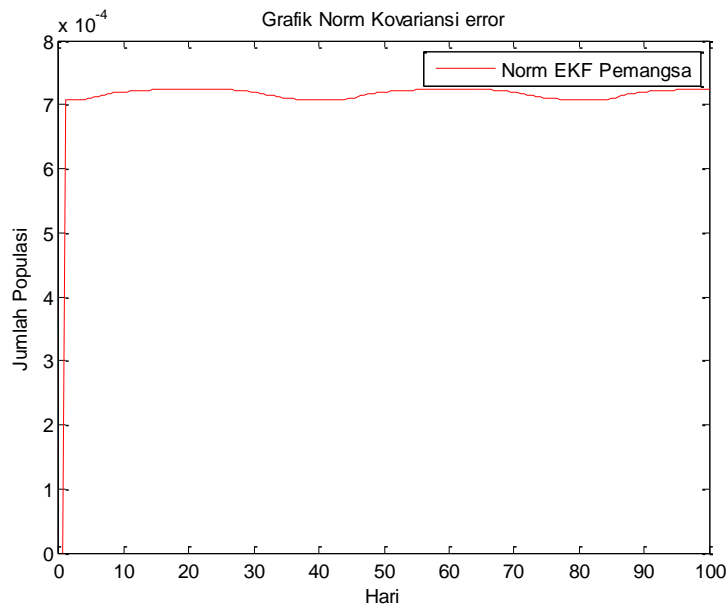


Gambar 4.2 Norm Kovariansi *Error* Mangsa (Kutu Hijau) pada Metode EKF

Pada gambar 4.1 menunjukkan bahwa ukuran populasi awal mangsa 216 ekor dan mengalami penurunan pada awal interval 10 hari. Setelah hari ke-20 mengalami kenaikan sampai hari ke-40. Setelah hari ke-40 kembali mengalami penurunan sampai hari ke-50. Populasi kembali mengalami kenaikan setelah hari ke-60 sampai hari ke-80. Setelah hari ke-80 populasi kembali menurun sampai pada hari ke-100. Dari gambar 4.1 dapat dilihat bahwa grafik mengalami perulangan setiap ± 40 hari, ini menunjukkan siklus hidup mangsa (kutu hijau). Jika dilihat dari gambar 4.1, grafik estimasi dengan grafik numerik hampir berhimpit. Hal ini menunjukkan bahwa estimasi cukup akurat yang ditandai dengan besarnya *norm kovariansi error* sebesar 0,00071374.

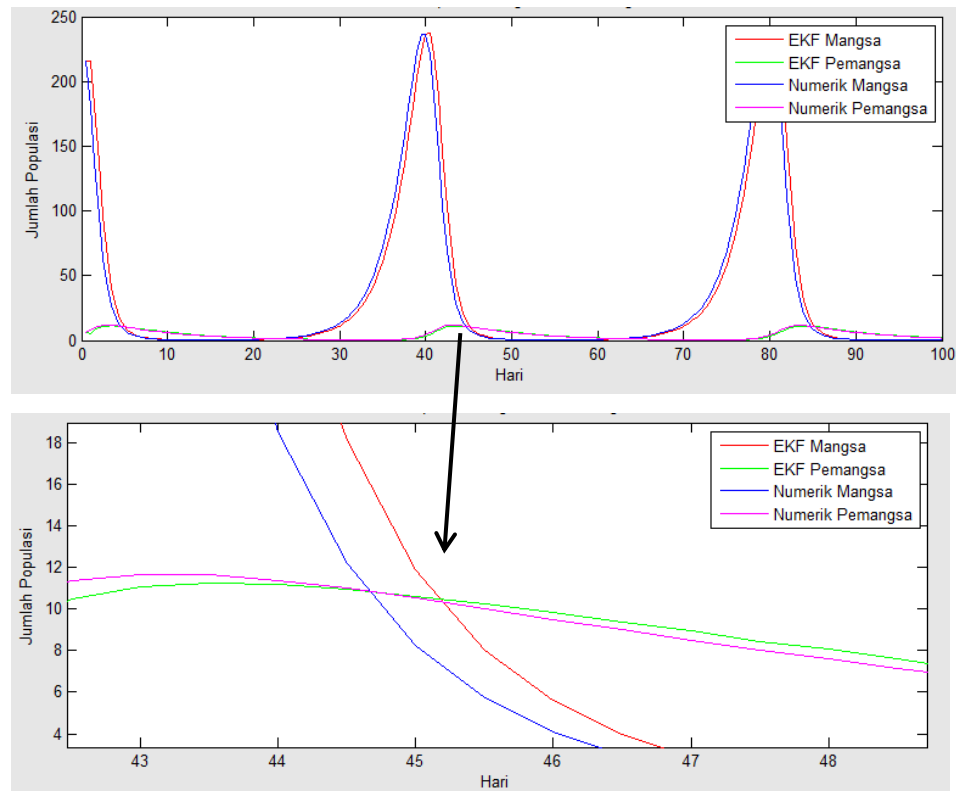


Gambar 4.3 Hasil Estimasi Populasi Pemangsa (Kumbang Merah) pada Metode EKF



Gambar 4.4 Norm Kovariansi *error* Pemangsa (Kumbang Merah) pada Metode EKF

Pada gambar 4.3 menunjukkan bahwa jumlah awal populasi pemangsa sebesar 6 ekor. Pada periode $t < 4$ hari, populasinya menunjukkan kenaikan secara signifikan karena ketersediaan sumber makanan yang cukup melimpah. Pada periode $4 < t < 34$ hari, populasi pemangsa mengalami penurunan. Periode $35 < t < 44$ hari populasi pemangsa kembali mengalami kenaikan dan pada periode $44 < t < 74$ hari kembali menurun. Pada periode selanjutnya yaitu $75 < t < 85$ hari populasi pemangsa kembali naik dan setelah periode $85 < t < 100$ hari populasi kembali menurun. Dari gambar 4.3 dapat dilihat bahwa grafik mengalami perulangan setiap ± 40 hari, ini menunjukkan siklus hidup pemangsa (kumbang merah). Jika dilihat dari grafik 4.3, grafik estimasi dengan grafik numerik hampir berhimpit. Hal ini menunjukkan bahwa estimasi ini cukup akurat yang ditandai dengan besarnya *norm kovariansi error* sebesar 0,00071374.

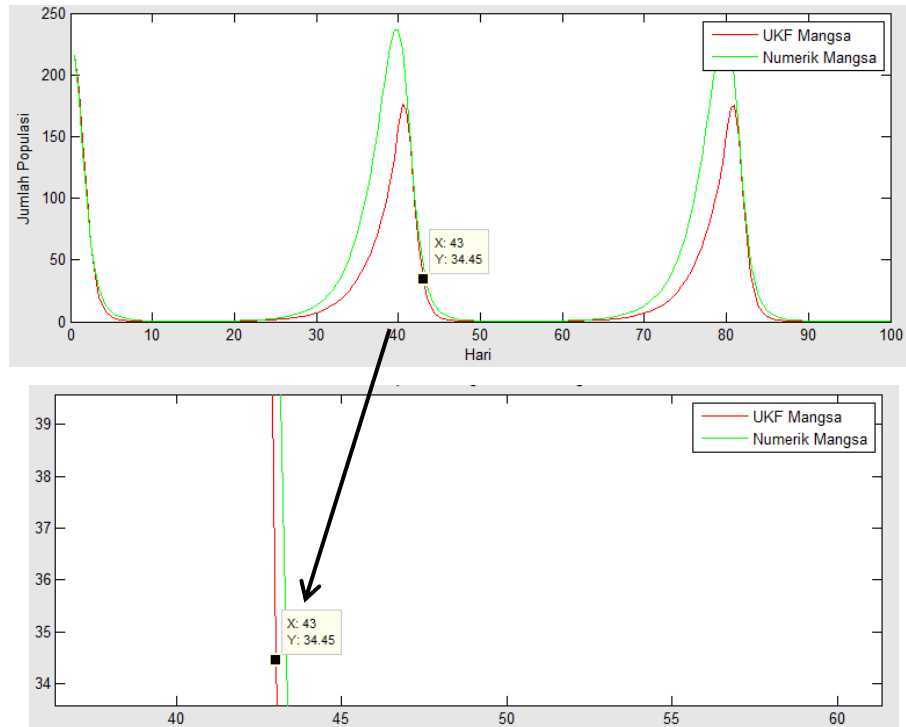


Gambar 4.5 Hasil Estimasi Populasi Mangsa (Kutu Hijau) dan Pemangsa (Kumbang Merah) pada Metode EKF

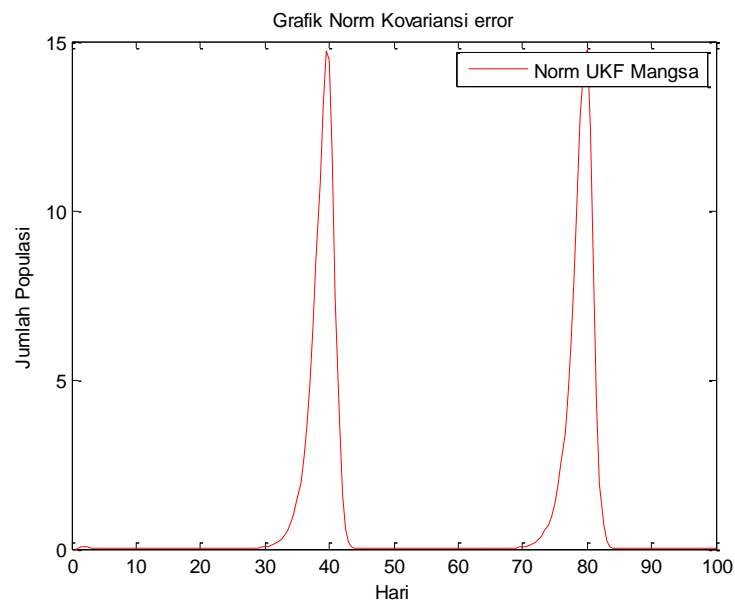
Pada gambar 4.5 menunjukkan populasi awal mangsa 216 ekor dan pemangsa 6 ekor. Pada awal interval populasi mangsa mengalami penurunan sedangkan populasi pemangsa mengalami kenaikan. Hal ini dikarenakan populasi pemangsa memakan populasi mangsa. Pada interval hari ke-25 sampai hari ke-40 populasi mangsa kembali meningkat seiring dengan menurunnya populasi pemangsa. Dari gambar 4.5 dapat dilihat bahwa grafik mengalami perulangan setiap ± 40 hari, ini menunjukkan siklus hidup mangsa (kutu hijau) dan pemangsa (kumbang merah).

4.5.2 Estimasi dengan Metode UKF

Hasil estimasi populasi mangsa dan pemangsa dengan menggunakan metode UKF dapat dilihat pada gambar 4.6 sampai 4.10.

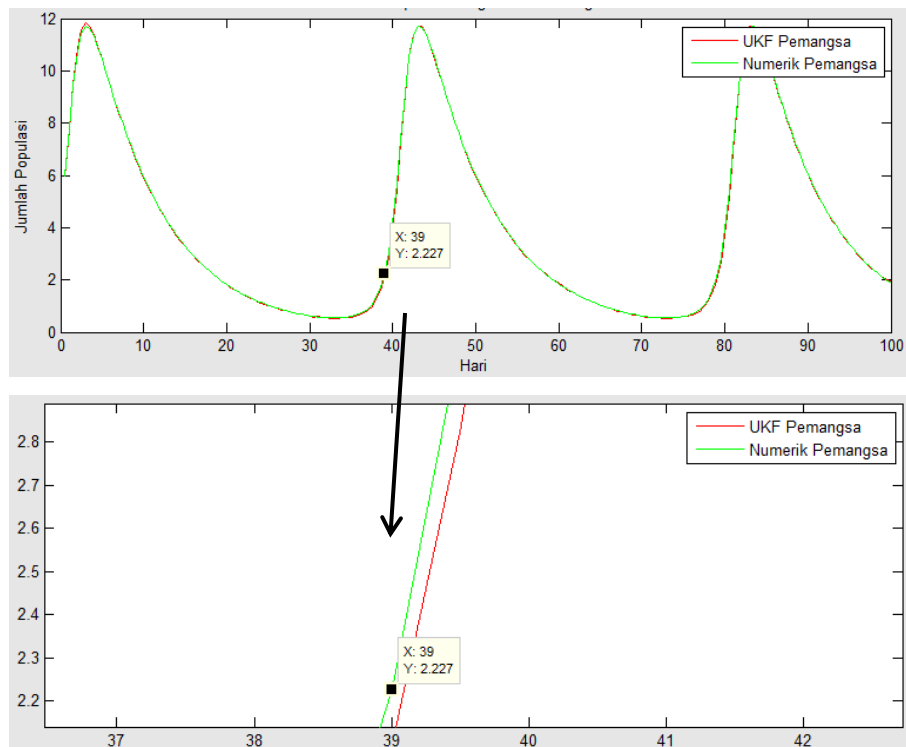


Gambar 4.6 Hasil Estimasi Populasi Mangsa (Kutu Hijau) pada Metode UKF

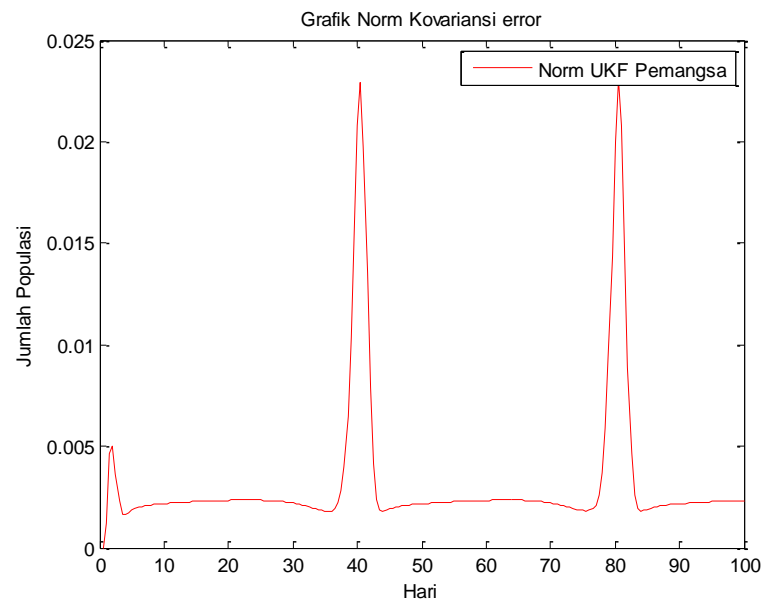


Gambar 4.7 Norm Kovariansi *Error* Mangsa (Kutu Hijau) Pada Metode UKF

Pada gambar 4.6 menunjukkan bahwa ukuran populasi awal mangsa 216 ekor dan mengalami penurunan pada awal interval 10 hari. Setelah hari ke-20 mengalami kenaikan sampai hari ke-40. Setelah hari ke-40 kembali mengalami penurunan sampai hari ke-50. Populasi kembali mengalami kenaikan setelah hari ke-60 sampai hari ke-80. Setelah hari ke-80 populasi kembali menurun sampai pada hari ke-100. Dari gambar 4.5 dapat dilihat bahwa grafik mengalami perulangan setiap ± 40 hari, ini menunjukkan siklus hidup mangsa (kutu hijau). Jika dilihat dari gambar 4.6, grafik estimasi dengan grafik numerik berhimpit hanya pada beberapa selang waktu saja. Hal ini menunjukkan bahwa estimasi kurang begitu akurat yang ditandai dengan besarnya *norm* kovariansi *error* sebesar 1,1238.

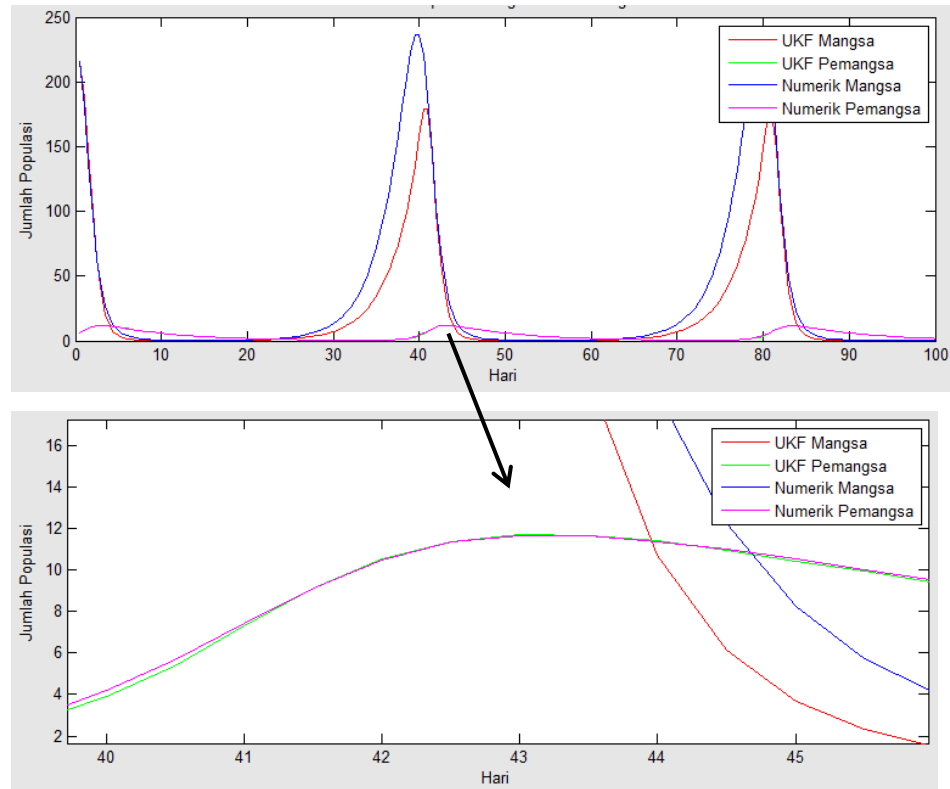


Gambar 4.8 Hasil Estimasi Populasi Pemangsa (Kumbang Merah) pada Metode UKF



Gambar 4.9 Norm Kovariansi *error* Pemangsa (Kumbang Merah) pada Metode UKF

Pada gambar 4.8 menunjukkan bahwa jumlah awal populasi pemangsa sebesar 6 ekor. Pada periode $t < 3$ hari, populasinya menunjukkan kenaikan secara signifikan karena ketersediaan sumber makanan yang cukup melimpah. Pada periode $2 < t < 32$ hari, populasi pemangsa mengalami penurunan. Periode $32 < t < 43$ hari populasi pemangsa kembali mengalami kenaikan dan pada periode $43 < t < 74$ hari kembali menurun. Pada periode selanjutnya yaitu $74 < t < 83$ hari populasi pemangsa kembali naik dan setelah periode $83 < t < 100$ hari populasi kembali menurun. Dari gambar 4.8 dapat dilihat bahwa grafik mengalami perulangan setiap ± 40 hari, ini menunjukkan siklus hidup pemangsa (kumbang merah). Jika dilihat dari grafik 4.8, grafik estimasi dengan grafik numerik sangat berhimpit. Hal ini menunjukkan bahwa estimasi ini cukup akurat yang ditandai dengan besarnya *norm kovariansi error* sebesar 0,0031978.

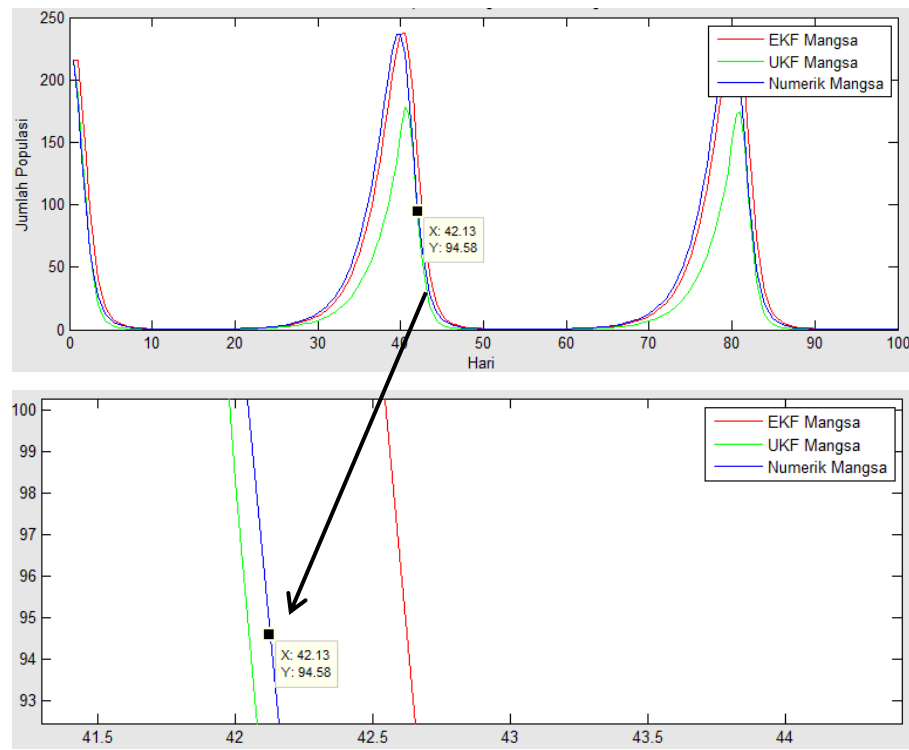


Gambar 4.10 Hasil Estimasi Populasi Mangsa (Kutu Hijau) dan Pemangsa (Kumbang Merah) pada Metode UKF

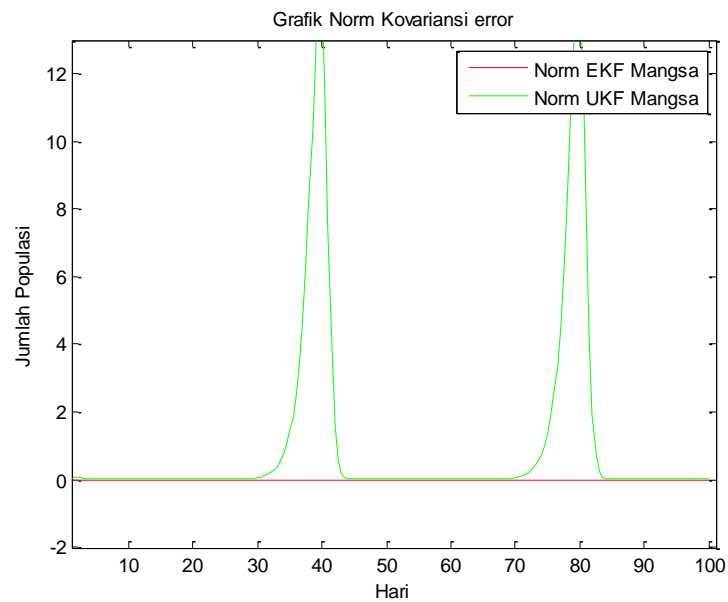
Pada gambar 4.10 menunjukkan populasi awal mangsa 216 ekor dan pemangsa 6 ekor. Pada awal interval populasi mangsa mengalami penurunan sedangkan populasi pemangsa mengalami kenaikan. Hal ini dikarenakan populasi pemangsa memakan populasi mangsa. Pada interval hari ke-25 sampai hari ke-40 populasi mangsa kembali meningkat seiring dengan menurunnya populasi pemangsa. Dari gambar 4.10 dapat dilihat bahwa grafik mengalami perulangan setiap ± 40 hari, ini menunjukkan siklus hidup mangsa (kutu hijau) dan pemangsa (kumbang merah).

4.6 Analisis Hasil Simulasi

Hasil estimasi populasi mangsa (kutu hijau) dan pemangsa (kumbang merah) dengan menggunakan metode EKF dan UKF dapat dilihat pada gambar 4.11 sampai 4.14.



Gambar 4.11 Hasil Estimasi Populasi Mangsa (Kutu Hijau) pada Metode EKF dan UKF

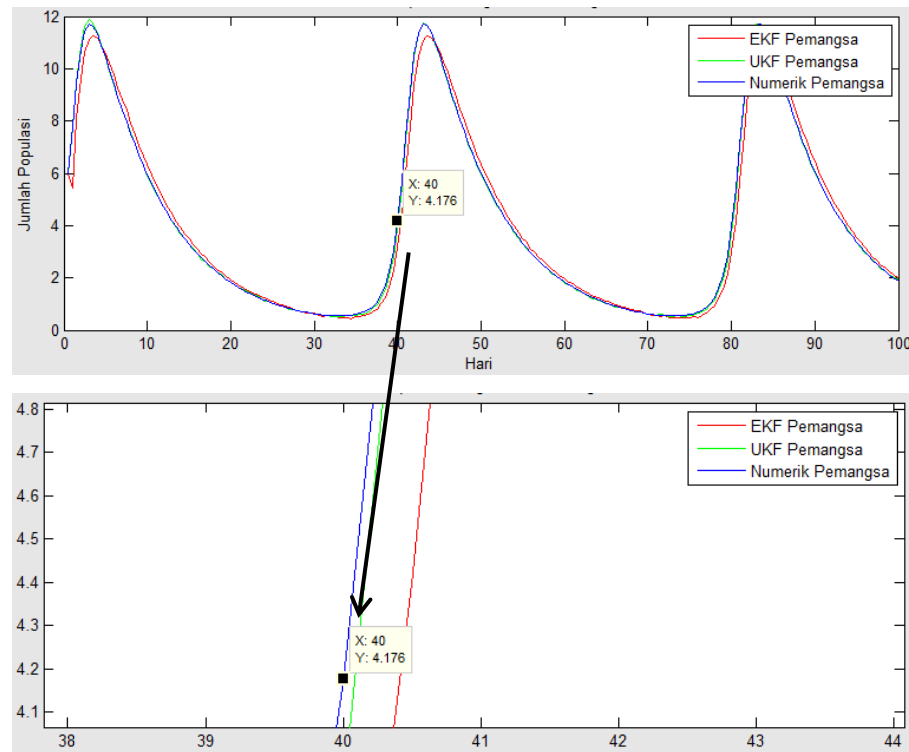


Gambar 4.12 Norm Kovariansi *error* Mangsa (Kutu Hijau) pada Metode EKF dan UKF

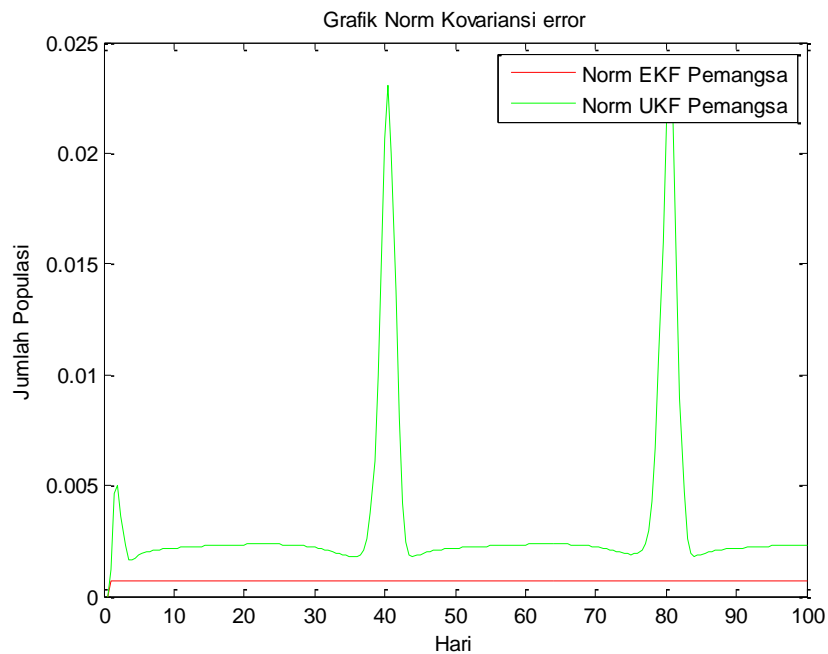
Dari hasil simulasi program dapat disimpulkan bahwa jika dilihat dari besarnya norm kovariansi *error* maka estimasi terbaik secara umum dari metode EKF dan metode UKF dalam mengestimasi populasi mangsa adalah metode EKF karena nilai *normkovariansi error* EKF lebih kecil dari UKF.

Tabel 4.1 Perbandingan Norm Kovariansi *error* Mangsa (Kutu Hijau)

Metode	Norm Kovariansi <i>error</i>
Metode EKF	0,00071374
Metode UKF	1,1238



Gambar 4.13 Hasil Estimasi Populasi Pemangsa (Kumbang Merah) pada Metode EKF dan Metode UKF



Gambar 4.14 Norm Kovariansi *error* Pemangsa (Kumbang Merah) pada Metode EKF dan Metode UKF

Dari hasil simulasi program dapat disimpulkan bahwa jika dilihat dari besarnya *norm kovariansi error* maka estimasi terbaik secara umum dari metode EKF dan metode UKF adalah metode EKF karena nilai *norm kovariansi error* EKF lebih kecil dari UKF.

Tabel 4.2 Perbandingan Norm Kovariansi *error* Pemangsa (Kumbang Merah)

Metode	Norm Kovariansi <i>error</i>
Metode EKF	0,00071374
Metode UKF	0,0031978

BAB 5. PENUTUP

Pada bab ini diperoleh kesimpulan dari hasil analisis dan pembahasan yang diperoleh dari penerapan metode *Extended Kalman Filter* dan metode *Unscented Kalman Filter* pada estimasi jumlah populasi mangsa (kutu hijau) dan pemangsa (kumbang merah) pada model *Predator-Prey* Lotka-Volterra, serta diberikan saran yang dapat dilakukan sebagai kelanjutan dari skripsi ini.

5.1 Kesimpulan

Berdasarkan hasil analisis dan pembahasan yang telah dilakukan diperoleh kesimpulan berikut.

- a. Pada model yang telah dikaji, estimasi populasi kutu hijau menggunakan EKF lebih baik dibandingkan estimasi UKF yang ditunjukkan dengan nilai kovariansi *error* EKF untuk variabel tersebut lebih kecil dari nilai kovariansi *error* UKF.
- b. Pada model yang telah dikaji, estimasi populasi kumbang merah menggunakan EKF lebih baik dibandingkan estimasi UKF yang ditunjukkan dengan nilai kovariansi *error* EKF untuk variabel tersebut lebih kecil dari nilai kovariansi *error* UKF.

5.2 Saran

Penelitian ini, permasalahan yang dibahas adalah estimasi populasi mangsa dan pemangsa pada model *Predator-Prey* Lotka-Volterra. Pada penelitian selanjutnya masih dimungkinkan untuk dilakukan penelitian dengan menggunakan persamaan *Predator-Prey* yang lain seperti *Predator-Prey* Leslie-Gower dengan pemanenan atau dengan menggunakan metode pengembangan dari Kalman Filter yang lain seperti Fuzzy Kalman Filter.

DAFTAR PUSTAKA

- Angga, T.F. 2010. *Penerapan Model Mangsa-Pemangsa Lotka-Volterra (Studi Kasus Perkebunan Kopi, Kakao (PTPN X), dan Kelapa Rakyat di Jember)*. Tidak Dipublikasikan. Skripsi. Jember : FMIPA Universitas Jember.
- Fitria, R. 2011. *Implementasi Ensemble Kalman Filter Pada Estimasi Kecepatan Kapal Selam*. [online] <http://digilib.its.ac.id/implementasi-ensemble-kalman-filter-enkf-pada-estimasi-kecepatan-kapal-selam-15084.html>
- Gumilar, A. 2011. *Estimasi Posisi Peluru Kendali Pada Lintasan Menggunakan Unscented Kalman Filter*. [online] <http://digilib.its.ac.id/estimasi-posisi-peluru-kendali-pada-lintasan-menggunakan-unscented-kalman-filter-15917.html>
- Hasbullah, H. 2011. *Algoritma Adaptive Covariance Rank unscented Kalman Filter Untuk Estimasi Keadaan Pada Persamaan Air Dangkal*. Tidak Diterbitkan. Tesis. Surabaya: Institut Teknologi Sepuluh Nopember.
- Huo, H. F, Wang, X & Carlos, C., 2011. Dynamics of a Stage Structured Leslie-Gower Predator-Prey Model. *Mathematical Problem of Engineering*, **2011**. Institute of Applied Mathematics Lanzou University of Technology.
- Ichwan, A. 2010. *Estimasi Posisi Kapal Selam Menggunakan Metode Extended Kalman Filter*. [on line]. <http://digilib.its.ac.id/estimasi-posisi--kapal-selam-menggunakan--metode-extended-kalman-filter-9296.html>
- Juwariya. 2013. Perbandingan Metode *Extended Kalman Filter* (EKF) dan *Unscented Kalman Filter* (UKF) dalam Estimasi Lintasan Misil. Tidak Dipublikasikan. Skripsi. Jember: FMIPA Universitas Jember.
- Kleinbauer, R. 2004. *Kalman Filtering Implementation with Matlab*. Helsinki: University Stuttgart.
- Masduqi, A., & Apriliani, E. 2008. Estimation of Surabaya River Water Quality Using Kalman Filter Algorithm. *The Journal for Technology and Science*, **19** (3) : 87-91.
- Miftahuddin. 2011. “*Estimasi Posisi Robot Mobil Menggunakan Unscented Kalman Filter*”. Tidak Diterbitkan. Skripsi. Surabaya: Institut Teknologi Sepuluh Nopember.

- Pradjaningsih, A. & Hidayat, R. 2011. Aplikasi Model Mangsa-Pemangsa Lotka-Volterra. *Jurnal Saintek*. Jember: Lembaga Penelitian Universitas Jember. **10**(1): 59-65.
- Prayudi, V. 2012. Penerapan *Extended Kalman Filter* Pada Model *Predator-Prey*. Tidak Dipublikasikan. Skripsi. Jember: FMIPA Universitas Jember.
- Purnomo, K. D. 2008. "Aplikasi Metode Ensemble Kalman Filter pada Model Populasi Plankton." Tidak Diterbitkan. Tesis. Surabaya: Program Pascasarjana Institut Teknologi Sepuluh Nopember.
- Tirta, I. M. 2009. *Analisis Regresi Dengan R (ANRER)*. Jember: UNEJ Press.
- Triadmodjo, Bambang. 2002. *Metode Numerik*. Yogyakarta: Beta Offset
- Wan, E. A. & Merwe, R. V. D. 2000. *The Unscented Kalman Filter for Nonlinear Estimation*. Oregon: Oregon Graduate Institute of Science & Technology.
- Welch, G. & Bishop, G. 2006. *An Introduction to The Kalman Filter*. Chapel Hill: University of North Carolina.
- Wicaksono, R. D. 2010. "Penerapan *Extended Kalman Filter* Untuk Mendeteksi Waktu Terjadinya Kerak Pada Alat Penukar Panas". Tidak Diterbitkan. Skripsi. Surabaya: Institut Teknologi Sepuluh Nopember.
- Yuanxin, Dewen, Meiping, & Xiaoping. 2005, June 8-10. *Unscented Kalman Filter for Additive Noise Case: Augmented vs. Non-augmented*. Portland, OR, USA: American Control Conference.

LAMPIRAN

A. Skrip Program Metode EKF

```
% clc; clear all; close all;
% Variabel jumlah populasi kutu hijau dinotasikan dengan
% M, sedangkan jumlah populasi kumbang merah dinotasikan
% dengan N
% INISIALISASI AWAL
%-----
% Penentuan grid variabel waktu (t)
% dt=0.1; % panjang grid dari t(dalam hari)
nt=steps;%250; % jumlah grid dari waktu
% nt=waktu;
% Matriks pengukuran H
nH=2; % jumlah baris taknol dari matriks H
H=eye(2,2);
% Variansi dari model sistem
% Q=0.001;
% Variansi dari model pengukuran
% R=0.001;
% Nilai awal kovariansi error
POM=P0;%0.006; % nilai kovariansi error awal
%untuk M
PON=P0;%0.0025; % nilai kovariansi error awal
%untuk N
P_awal=[POM 0;0 PON];
% Jumlah populasi kutu hijau dan populasi kumbang merah
%saat t=0(dalam ekor)
z=[];
M(1)=x0;
N(1)=y0;
% Perhitungan variabel keadaan real X_re dan data
%pengukuran z
%-----
M_re(1)=M(1); %M_re = variabel keadaan real M
N_re(1)=N(1); %N_re = variabel keadaan real N
for k=1:(nt-1)
w1=sqrt(Q)*randn(1,1);
w2=sqrt(Q)*randn(1,1);
M_re(k+1)=Reall(1,k);%M_re(k)+dt*(a*M_re(k)-alfa*M_re(k)*N_re(k))+w1;
N_re(k+1)=Reall(2,k);%N_re(k)+dt*(b*N_re(k)+beta*M_re(k)*N_re(k))+w2;
X_re([1:2],k+1)=[M_re(k+1);N_re(k+1)];
v=sqrt(R)*randn(nH,1);
z(:,k+1)=H*X_re(:,k+1)+v;
end
% Implementasi algoritma EKF dalam menghitung nilai
%estimasi variabel keadaan dan norm kovariansi error
%-----
% Inisialisasi X koreksi
M_pred(1)=M(1);
N_pred(1)=N(1);
M_si(1)=M_re(1);
N_si(1)=N_re(1);
for k=1:(nt-2)
```

```

M_si(k+1)=M_re(k+1);
N_si(k+1)=N_re(k+1);
end
M_kor(1)=M(1);
N_kor(1)=N(1);
for k=1:(nt-1)
% Matriks Jacobi A
A = [1+dt*(a-alfa*N_si(k)) -alfa*M_si(k)*dt ;
beta*N_si(k)*dt 1+dt*(-b+beta*M_si(k))];
% Tahap Prediksi (time update)
%-----
P_pred=A*P_awal*A'+Q;
M_pred(k+1)=M_si(k)+dt*(a*M_si(k)-alfa*M_si(k)*N_si(k));
N_pred(k+1)=N_si(k)+dt*(-b*N_si(k)+beta*M_si(k)*N_si(k));
X_pred([1:2],k+1)=[M_pred(k+1);N_pred(k+1)];
% Tahap Koreksi (measurement update)
%-----
K=P_pred*H'*inv(H*P_pred*H'+R);
P_kor=(eye(2,2)-K*H)*P_pred;
X_kor([1:2],k+1)=X_pred(:,k+1)+K*(z(:,k+1)-H*X_pred(:,k+1));
% Rekapitulasi Hasil
%-----
M_kor(k+1)=X_kor([1],k+1);
N_kor(k+1)=X_kor([2],k+1);
NormP_kor1(k+1)=norm(P_kor(1,:));
NormP_kor2(k+1)=norm(P_kor(2,:));
selisih([1:2],k)=X_kor([1:2],k)-X_re([1:2],k);
selisihM(k)=selisih([1],k);
selisihN(k)=selisih([2],k);
% pause(0.1);
end

```

B. Skrip Program Metode UKF

```

% clc; clear all; %close all
%inisialisasi nilai awal sesuai jurnal
M1=x0;%216;
N1=y0;%6;
% a=0.4; % konstanta pertumbuhan(mangsa)
% b=0.12; % konstanta penurunan (pemangsa)
% alfa=0.11; % konstanta interaksi
% beta=0.0032; % konstanta interaksi
% dt = 0.1;
clc;
% N=250; % tahun
% Q1=0.001;
% Q=[Q1];
R1=R;%0.001;
n=2;
x=[];
f1=@(x) [x(1)+dt*(a*x(1)-alfa*x(1)*x(2));
x(2)+dt*(-b*x(2)+beta*x(1)*x(2))];

% h=@(x) [];

```

```

h1=@(x) [x(2)];
s=[M1;N1];
x=s+Q*randn(2,1);
z=[];

P = eye(n)*P0;
xV = zeros(n,waktu/dt);           %estimate           % allocate memory
sV = zeros(n,waktu/dt);           %actual
zV = zeros(1,waktu/dt);
xV(:,1)=s;
sV(:,1)=s;

for k=2:waktu/dt
    s =Reall(:,k);% f1(s) + normrnd(0,sqrt(Q),2,1);           % update
process
    z = h1(s) + normrnd(0,sqrt(Q),1,1);                       % measurements
    sV(:,k)= s;                                               % save actual state
    zV(:,k)= z;                                               % save measurment
    [x,P] = ukf_exm(f1,x,P,h1,z,Q,R1);                       % ukf
    xV(:,k) = x;                                              % save estimate

    P1(1,k)=norm(P(1,:));
    P1(2,k)=norm(P(2,:));
end

```