

Optimasi Filter Kalman dengan Metode Steepest Descent dan Least Mean Square pada Rekonstruksi Citra Dinamis

Rully Soelaiman*, Irfan Subakti* dan Roy Nugroho*

* Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember (ITS), Surabaya, 60111, Indonesia.
Email: rully130270@gmail.com, yifana@gmail.com

ABSTRAK

Filter Kalman terbukti dapat melakukan proses rekonstruksi citra dinamis, dimana permasalahan rekonstruksi citra dinamis dimodelkan sebagai permasalahan estimasi dinamis dengan fungsi tujuan meminimalkan Mean Squared Error (MSE). Namun terdapat keterbatasan pada biaya komputasi filter Kalman yang mahal, sehingga dilakukan optimasi pada filter Kalman dengan menggunakan metode iteratif.

Pada paper ini, pemodelan sistem linier dari rekonstruksi citra dinamis diselesaikan dengan metode iteratif, yaitu metode Steepest Descent (SD) dan Least Mean Square (LMS). Kedua metode ini merupakan penurunan dari filter Kalman.

Uji coba dilakukan dengan menggunakan data sintesis, dengan membandingkan filter Kalman, metode langsung (Pseudo-RLS) dan metode iteratif (SD dan LMS). Hasil uji coba menunjukkan bahwa metode iteratif lebih singkat dalam biaya komputasinya, dengan kinerja yang hampir sama dengan filter Kalman dan metode Pseudo-RLS.

Kata Kunci: Filter Kalman, estimasi dinamis, rekonstruksi citra dinamis, sistem linier, metode Recursive Least Squares (RLS), metode iteratif, Steepest Descent, Least Mean Square.

1. PENDAHULUAN

Rekonstruksi citra dinamis merupakan permasalahan estimasi dinamis, yang dapat diselesaikan dengan algoritma estimasi dinamis, yaitu filter Kalman [1]. Keterbatasan pada filter Kalman adalah biaya komputasi yang mahal, sehingga diperlukan optimasi untuk mendapatkan biaya komputasi yang lebih efisien. Optimasi dilakukan dengan memodelkan permasalahan rekonstruksi citra dinamis menjadi

permasalahan sistem linier yang diselesaikan dengan metode langsung (Pseudo-RLS) dan metode iteratif yaitu metode Steepest Descent dan Least Mean Square.

Model observasi yang digunakan pada rekonstruksi citra dinamis dalam domain spasial dibagi menjadi dua [2]. Model observasi pertama menyatakan model degradasi, yang dinyatakan pada persamaan (1):

$$\underline{Y}(t) = DH(t)\underline{X}(t) + \underline{N}(t) \quad (1)$$

$\underline{Y}(t) \in \mathfrak{R}^{M^2}$ adalah vektor yang merepresentasikan rangkaian citra resolusi rendah. $\underline{X}(t) \in \mathfrak{R}^{L^2}$ adalah vektor yang merepresentasikan rangkaian citra resolusi tinggi, dimana $L > M$. t adalah urutan gambar atau jumlah citra. D adalah matriks desimasi (*downsample*), dengan ukuran $M^2 \times L^2$. $H(t)$ adalah matriks blur, dengan ukuran $L^2 \times L^2$, dan $\underline{N}(t)$ adalah vektor Gaussian *noise* dengan mean = 0 dan ukuran $M^2 \times 1$, yang memiliki matriks autokorelasi $W^{-1}(t) = E\{\underline{N}(t)\underline{N}^T(t)\}$, Matriks $W^{-1}(t)$ diasumsikan bernilai $\sigma^2 I$, dimana σ^2 adalah varians dari vektor noise $\underline{N}(t)$. Persamaan (1) kemudian disederhanakan menjadi (2), yang disebut juga dengan persamaan pengukuran.

$$\underline{Y}(t) = H_A(t)\underline{X}(t) + \underline{N}(t) \quad (2)$$

Model observasi kedua adalah untuk menyatakan keterkaitan antar citra resolusi tinggi, yang ditunjukkan pada persamaan sistem (3):

$$\underline{X}(t) = G(t)\underline{X}(t-1) + \underline{V}(t) \quad (3)$$

dimana $G(t)$ merepresentasikan matriks penghubung antara citra $\underline{X}(t)$ dan $\underline{X}(t-1)$, yang dinyatakan dengan matriks Identitas. Vektor $\underline{V}(t)$ yang memiliki matriks autokorelasi $Q^{-1}(t) = E\{\underline{V}(t)\underline{V}^T(t)\}$, diasumsikan bernilai nol sebagai akibat dari matriks $G(t)$ yang merupakan matriks identitas.

Pemodelan sistem linier dari rekonstruksi citra dinamis dilakukan dengan pendekatan fungsi tujuan untuk meminimalkan *error* dari *Weighted Least Squares* (WLS). Dengan mengasumsikan bahwa matriks autokorelasi noise $Q^{-1}(t) = 0$, sehingga persamaan sistem (3) menjadi $\underline{X}(t) = G(t)\underline{X}(t-1)$, atau dapat ditulis menjadi $\underline{X}(t-1) = F(t)\underline{X}(t)$, dimana $F(t)$ menunjukkan *backward motion*, atau gerak

kebalikan dari vektor citra $\underline{X}(t)$ ke $\underline{X}(t-1)$. Sehingga dengan menggabungkan persamaan pengukuran (2) dan persamaan sistem (3), menjadi persamaan (4):

$$\begin{aligned} \underline{Y}(t-k) &= H_A(t-k) \prod_{j=1}^k F(t-k+j) \underline{X}(t) + \underline{N}(t-k) \\ &= H_A(t-k) F(t,k) \underline{X}(t) + \underline{N}(t-k) \end{aligned} \quad (4)$$

Untuk mencari solusi optimal dari $\underline{X}(t)$ yaitu $\hat{\underline{X}}(t)$, maka *error* WLS dari persamaan (4) dinyatakan pada persamaan (5) berikut:

$$\varepsilon^2(t) = \sum_{k=0}^{\infty} \left\| \underline{Y}(t-k) - H_A(t-k) F(t,k) \hat{\underline{X}}(t) + \underline{N}(t-k) \right\|_{W(t,k)}^2 \quad (5)$$

Dengan menurunkan *error* WLS $\varepsilon^2(t)$ terhadap $\hat{\underline{X}}(t)$, kemudian di set sama dengan nol, maka didapatkan persamaan linier (6):

$$\hat{\underline{Z}}(t) = \hat{L}(t) \hat{\underline{X}}(t) \quad (6)$$

dengan $\hat{\underline{Z}}(t) = \lambda(t) F^T(t) \hat{\underline{Z}}(t-1) + H_A^T(t) W(t) \underline{Y}(t)$ dan $\hat{L}(t) = \lambda(t) F^T(t) \hat{L}(t) F(t) + H_A^T(t) W(t) H_A(t)$.

Persamaan sistem linier (6) akan diselesaikan dengan metode iteratif yang biaya komputasinya lebih efisien dari filter Kalman.

Pada paper ini, penjelasan mengenai metode *Steepest Descent* dijelaskan pada bagian 2, kemudian metode *Least Mean Square* pada bagian 3. Hasil uji coba ditampilkan pada bagian 4, kemudian penarikan kesimpulan pada bagian 6.

2. METODE STEEPEST DESCENT

Metode iteratif pertama yang digunakan dalam paper ini adalah metode *Steepest Descent*, yang didasarkan pada gradien. Bentuk umum dari algoritma *Steepest Descent* dalam [3] dinyatakan pada persamaan (7) berikut:

$$x(n+1) = x(n) + \alpha(n)p(n) \quad (7)$$

dengan tujuan untuk mendapatkan nilai $x(n+1)$ yang paling optimal dari $x(n)$ yang diupdate dengan nilai $\alpha(n)p(n)$. $\alpha(n)$ adalah nilai skalar yang disebut dengan *step size*. $p(n)$ adalah arah dari pergerakan untuk mengoptimalkan nilai $x(n+1)$. Nilai $p(n)$ diupdate terhadap indeks n , dimana pemilihan nilai $p(n)$ bergantung pada fungsi tujuannya. Bila digunakan untuk memaksimalkan suatu fungsi, maka dinamakan *Steepest Ascent*, dan bila digunakan untuk meminimalkan suatu fungsi, maka dinamakan *Steepest Descent*.

Dalam konteks rekonstruksi citra dinamis, persamaan sistem linier ditunjukkan pada persamaan (6) hasil dari

pendekatan metode Pseudo-RLS. Arah pergerakan untuk mengoptimalkan nilai $\hat{\underline{X}}(t)$ adalah vektor residu dari persamaan (8) yang merupakan hasil penurunan (gradien) dari fungsi tujuan meminimalkan kriteria *error Weighted Least Square*:

$$\hat{\underline{Z}}(t) - \hat{L}(t) \hat{\underline{X}}(t) \quad (8)$$

Penerapan dari metode *Steepest Descent* untuk rekonstruksi citra dinamis dinyatakan pada persamaan (9) dan (10):

$$\hat{\underline{X}}_0(t) = G(t) \hat{\underline{X}}_R(t-1) \quad (9)$$

$$\hat{\underline{X}}_k(t) = \hat{\underline{X}}_{k-1}(t) + \mu_k [\hat{\underline{Z}}(t) - \hat{L}(t) \hat{\underline{X}}_{k-1}(t)], 1 \leq k \leq R \quad (10)$$

dimana $\hat{\underline{X}}_0(t)$ adalah inisial awal untuk perhitungan solusi optimal $\hat{\underline{X}}_R(t)$ secara iteratif sebanyak R kali pada persamaan (9), waktu ke t . $\hat{\underline{X}}_0(t)$ juga sebagai nilai update dari waktu $t-1$. Sedangkan perhitungan iteratif pada persamaan (10), pencarian solusi optimal dari waktu ke t dilakukan dengan arah pergerakan vektor residu, step size μ_k , dan dengan indeks k dalam kondisi $1 \leq k \leq R$. Sehingga persamaan (9) dan (10) disebut dengan algoritma R-*Steepest Descent* (R-SD), dengan tahap inialisasi pada persamaan (11):

$$\begin{aligned} \hat{\underline{X}}_R(0) &= E\{\underline{X}(0)\} \\ \hat{L}(0) &= \hat{P}^{-1}(0) \\ &= [E\{\underline{X}(0)\underline{X}^T(0)\}]^{-1} \\ \hat{\underline{Z}}(0) &= \hat{L}(0) \hat{\underline{X}}_R(0) \end{aligned} \quad (11)$$

Normalisasi R-SD

Metode R-SD pada persamaan (11) terdapat *step size* μ_k , yang nilainya mempengaruhi konvergensi dari metode ini. Suatu pendekatan untuk pemilihan nilai μ_k adalah dengan menggunakan normalisasi metode R-SD, dimana tiap iterasinya didefinisikan pada persamaan (12):

$$\begin{aligned} \mu_k(t) &= \frac{\underline{E}_k^T(t) \underline{E}_k(t)}{\underline{E}_k^T(t) \hat{L}(t) \underline{E}_k(t)} \\ \underline{E}_k(t) &= \hat{\underline{Z}}(t) - \hat{L}(t) \hat{\underline{X}}_{k-1}(t) \\ \hat{\underline{X}}_k(t) &= \hat{\underline{X}}_{k-1}(t) + \mu_k(t) \underline{E}_k(t), 1 \leq k \leq R \end{aligned} \quad (12)$$

Sedangkan pemilihan forgetting factor $\lambda(t)$, dilakukan dengan *trial* dan *error*, dengan persyaratan $0 \ll \lambda(t) < 1$, yang akan ditunjukkan pada bagian 4 yaitu hasil uji coba.

Metode normalisasi R-SD ini lebih baik dibandingkan dengan metode R-SD, yang akan dibahas pada bagian 4, hasil uji coba. Pseudocode dari metode normalisasi R-SD ditunjukkan pada gambar 1. Vektor *initX* dan matriks *initP* merupakan inialisasi vektor dan matriks saat $t = 0$. Vektor

$initX$ berisi hasil interpolasi bilinear dari citra resolusi rendah $Y(t)$ pada $t = 0$, sedangkan matriks $initP$ adalah matriks kovarians dari vektor $initX$ [2].

3. METODE LEAST MEAN SQUARE

Metode iteratif kedua yang digunakan untuk melakukan rekonstruksi citra dinamis adalah metode *R - Least Mean Squares* (R-LMS). Metode R-LMS dalam konteks rekonstruksi citra Super-Resolusi, didapatkan dengan cara mengganti nilai $\hat{Z}(t)$ dan nilai $\hat{L}(t)$, yang dinyatakan pada persamaan (13) dan (14) seperti pada [1]:

$$\hat{Z}(t) = H_A^T(t)W(t)Y(t) \quad (13)$$

$$\hat{L}(t) = H_A^T(t)W(t)H_A(t) \quad (14)$$

Dengan mensubstitusikan nilai $\hat{Z}(t)$ pada persamaan (13) dan $\hat{L}(t)$ pada persamaan (14) ke dalam persamaan (10), didapatkan persamaan (15):

$$\hat{X}_k(t) = \hat{X}_{k-1}(t) + \mu_k H_A^T(t)W(t)[Y(t) - H_A(t)\hat{X}_{k-1}(t)] \quad (15)$$

dengan indeks k memenuhi $1 \leq k \leq R$, dan tahap inisialisasi dari metode R-LMS dinyatakan pada persamaan (16):

$$\hat{X}_R(0) = E\{X(0)\} \quad (16)$$

Persamaan (15) dan (16) disebut dengan metode R-LMS.

```

Normalisasi-RSD (initX, initP, F, Ha, Wa, Ya, lambda, R, G)
  T ← length[Y[1]]
  for t ← 1 to T
    do if t = 1
      then prevX ← initX
           prevL ← initP-1
           prevZ ← prevL x prevX
      else prevX ← Xfilt[1..T][t-1]
           prevZ ← Z[1..T][t-1]
           prevL ← hasilL
  Z[1..T][t] ← lambda[1][t] x FT x prevZ + HaT x W x Y[1..T][t]
  hasilL ← lambda[1][t] x FT x prevL x F + HaT x W x Ha
  XrNol = G x prevX
  for k ← 1 to R
    do if k = 1
      then prevXk ← XrNol
      else prevXk ← Xk[1..T][k-1]
      E[1..T][k] ← Z[1..T][t] - hasilL x prevXk
      myu ← ET[1..T][k] x E[1..T][k] / (ET[1..T][k] x hasilL x E[1..T][k])
      Xfilt[1..T][t] ← Xk[1..T][R]
    
```

Gambar 1. Pseudocode metode Normalisasi R-SD

```

Normalisasi-RLMS (initX, G, Ha, Wa, Ya, R)
1  T ← length[Y[1]]
2  for t ← 1 to T
3    do if t = 1
4      then prevX ← initX
5      else prevX ← Xfilt[1..T][t-1]
6      XrNol = G x prevX
7      for k ← 1 to R
8        do if k = 1
9          then prevXk ← XrNol
10         else prevXk ← Xk[1..T][k-1]
11         E[1..T][k] ← HaT x W x (Y[1..T][t] - HaT x prevXk)
12         myu ← ET[1..T][k] x E[1..T][k] / (ET[1..T][k] x HaT x W x Ha x E[1..T][k])
13         Xfilt[1..T][t] ← Xk[1..T][R]
    
```

Gambar 2. Pseudocode metode Normalisasi R-LMS







Normalisasi R-LMS

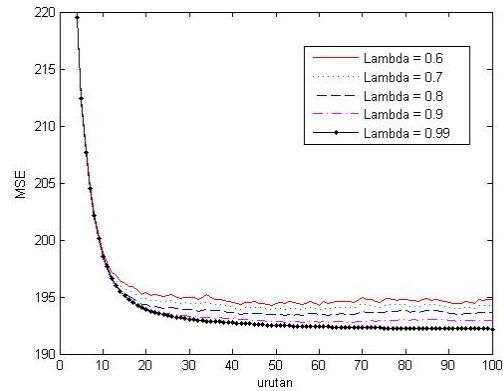
Seperti pada metode R-SD, metode R-LMS juga memiliki *step size* μ_k yang mempengaruhi nilai konvergensi, sehingga dalam pemilihan *step size* μ_k dilakukan dengan pendekatan normalisasi metode R-LMS, yang dinyatakan pada persamaan (17). Metode Normalisasi R-LMS lebih baik daripada metode R-LMS. Bila dibandingkan dengan metode normalisasi R-SD, metode normalisasi R-LMS lebih sederhana dalam persamaan matematikanya dan lebih singkat dalam waktu komputasinya, yang ditunjukkan pada hasil uji coba.

Kedua metode iteratif R-SD dan R-LMS juga dikatakan sebagai algoritma filter adaptif. Karakteristik dari algoritma filter adaptif memiliki 2 proses dasar yang meliputi proses filtering dan proses adaptif. Proses filtering menghasilkan sinyal output yaitu $\hat{x}_R(t)$ dan *error* estimasi yaitu *error Weighted Least Squares*. Sedangkan proses adaptif melakukan penyesuaian pada *step size* tiap waktu diskrit, yang ditunjukkan dengan perubahan pada nilai *step size* μ_k . Pada kedua metode iteratif ini, pemilihan nilai iterasi R ditunjukkan pada bagian hasil uji coba.

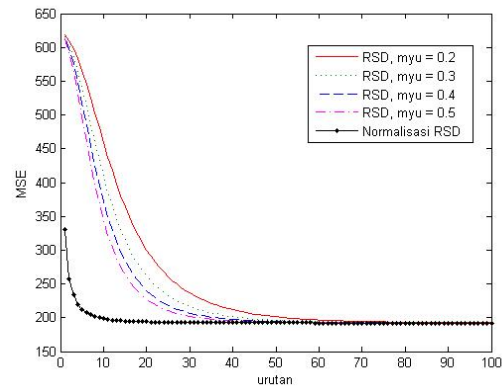
$$\begin{aligned} \hat{X}_0(t) &= G(t)\hat{X}_R(t-1) \\ \mu_k(t) &= \frac{E_k^T(t)E_k(t)}{E_k^T(t)H_A^T(t)W(t)H_A(t)E_k(t)} \\ E_k(t) &= H_A^T(t)W(t)[Y(t) - H_A(t)\hat{X}_{k-1}(t)] \\ \hat{X}_k(t) &= \hat{X}_{k-1}(t) + \mu_k(t)E_k(t), 1 \leq k \leq R \end{aligned} \tag{17}$$

Tabel 1. Data Citra

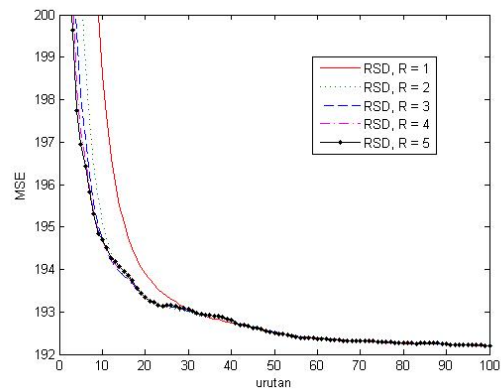
Nama	Citra	Nama	Citra
Lena		Boat	
Peppers		Baboon	
Bridge		Cameraman	



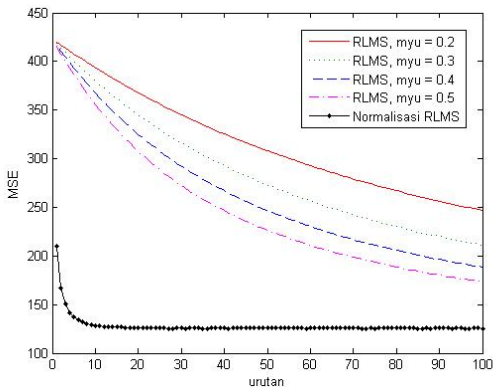
Gambar 3. Perbandingan R-SD dengan nilai $\lambda(t)$ berbeda



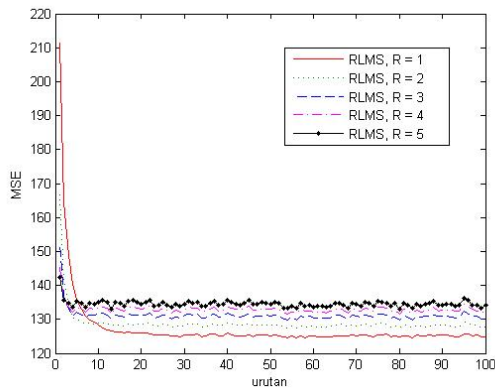
Gambar 4. Perbandingan R-SD dan Normalisasi R-SD



Gambar 5. Perbandingan R-SD dengan nilai R berbeda



Gambar 6. Perbandingan R-LMS dan Normalisasi R-LMS



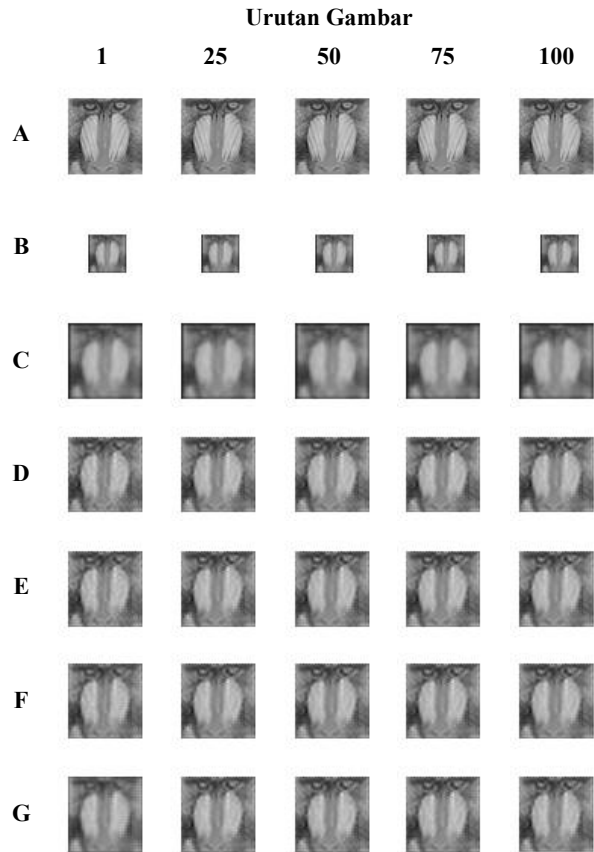
Gambar 7. Perbandingan R-LMS dengan nilai R Berbeda

4. HASIL UJI COBA

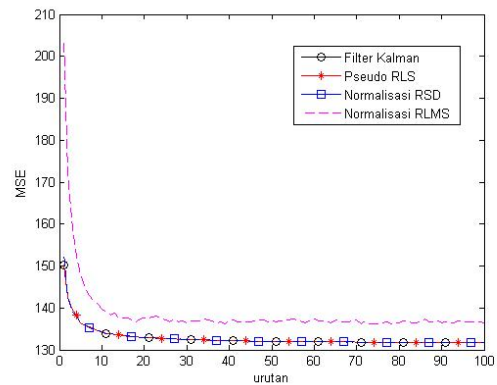
Uji coba dilakukan terhadap rangkaian citra resolusi rendah untuk direkonstruksi menjadi rangkaian citra resolusi tinggi. Data uji coba yang digunakan dalam format grayscale, dengan ukuran citra 50x50 piksel, dan jumlah citra sebanyak 100 citra. Penggunaan ukuran citra terkait dengan keterbatasan memori Tools Pembangun (Matlab 7.0). Data citra yang digunakan dinyatakan pada tabel 1.

Uji coba diterapkan pada masing-masing algoritma, dan kemudian dilakukan perbandingan pada aspek kualitas gambar, dan nilai parameter pengukuran MSE, dan PSNR dengan formulasi $10 \log_{10} \frac{255^2}{MSE}$, serta waktu komputasinya.

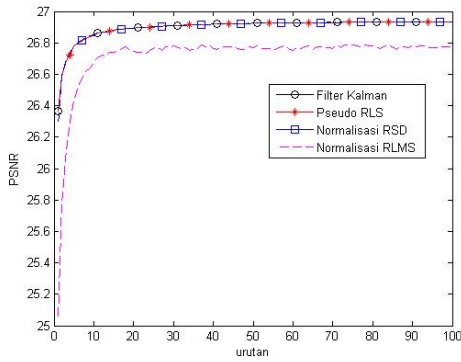
Pada paper ini hanya diberikan contoh untuk 2 citra gambar, yaitu rangkaian citra *Baboon* dan *Cameraman*.



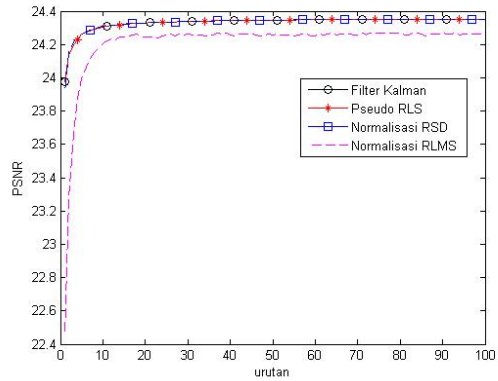
Gambar 8. Rangkaian Citra Baboon



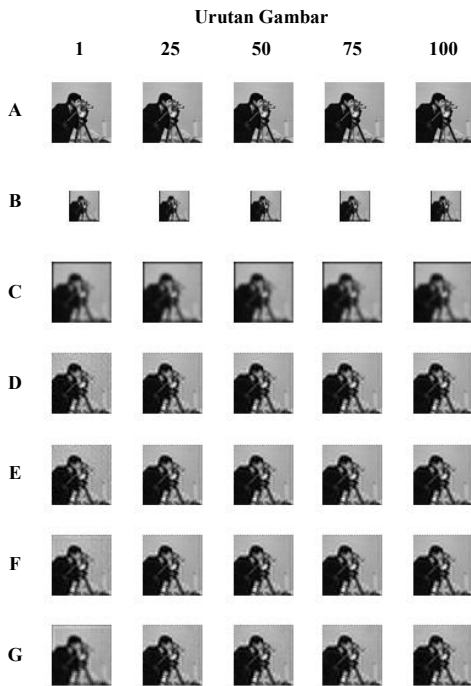
Gambar 9. MSE Hasil Rekonstruksi Rangkaian Citra Baboon



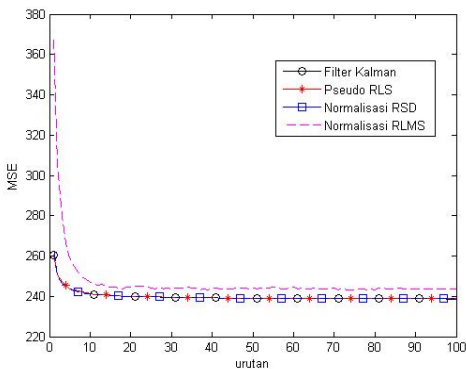
Gambar 10. PSNR Hasil Rekonstruksi Rangkaian Citra Baboon



Gambar 13. PSNR Rekonstruksi Rangkaian Citra Cameraman



Gambar 11. Rangkaian Citra Cameraman



Gambar 12. MSE Hasil Rekonstruksi Rangkaian Citra Cameraman

Tabel 2. Rata-rata PSNR

Algoritma	PSNR (dB)						Rata-rata
	Lena	Peppers	Bridge	Boat	Baboon	Cameraman	
Filter Kalman	27.289	24.34	23.913	24.669	26.9	24.332	25.241
Pseudo-RLS	27.288	24.339	23.913	24.669	26.9	24.332	25.24
R-SD Normal	27.288	24.338	23.912	24.669	26.899	24.331	25.24
R-LMS Normal	27.085	24.215	23.802	24.553	26.717	24.212	25.097

Tabel 3. Waktu Komputasi Algoritma

Algoritma	Waktu Komputasi (detik)						Rata-rata
	Lena	Peppers	Bridge	Boat	Baboon	Cameraman	
Filter Kalman	11233	8025	9108	8011	8032	12266	9445.8
Pseudo-RLS	8029	6093.5	6851	6046	6092	9150.7	7043.8
R-SD Normal	5349	4232.5	4654	4207	4221	6184.9	4808
R-LMS Normal	131.3	130.2	130.3	130.3	130.2	129.98	130.37

Rangkaian citra resolusi rendah yang direkonstruksi adalah rangkaian citra yang terdegradasi dengan blur *uniform* [3 x 3], dengan operator desimasi 2, dan *noise* Gaussian dengan *mean* = 0 dan *varians* = 5, yang dihasilkan secara sintesis. Hasil perbandingan algoritma dapat dilihat pada gambar 8 sampai 13. Gambar 8 dan 11 memiliki keterangan A: rangkaian citra resolusi tinggi yang asli, B: rangkaian citra resolusi rendah, C: rangkaian citra interpolasi *bilinear*, D: hasil rekonstruksi filter Kalman, E: hasil rekonstruksi Pseudo-RLS, F: hasil rekonstruksi normalisasi R-SD, G: hasil rekonstruksi normalisasi R-LMS.

Tabel 2 dan tabel 3 menunjukkan perbandingan hasil PSNR dan waktu komputasi tiap algoritma, yang menunjukkan bahwa filter Kalman adalah algoritma terbaik dalam

melakukan rekonstruksi citra Super-Resolusi, dilanjutkan dengan Pseudo-RLS, R-SD dan R-LMS. Sedangkan dalam waktu komputasinya filter Kalman merupakan algoritma paling lambat, dilanjutkan dengan Pseudo-RLS, R-SD dan R-LMS dengan waktu paling cepat.

Gambar 4 dan 6, menunjukkan bahwa metode normalisasi R-SD dan R-LMS lebih cepat mengalami konvergensi daripada tanpa normalisasi. Kemudian uji coba pemilihan nilai *forgetting factor* $\lambda(t)$ dan R pada metode R-SD ditunjukkan pada gambar 3 dan 5. Sedangkan gambar 7 adalah uji coba pemilihan nilai R pada algoritma R-LMS.

Hasil pada gambar 3 dan 5 menunjukkan bahwa semakin besar nilai $\lambda(t)$, yang mendekati nilai 1 dan semakin besar nilai R , semakin cepat algoritma R-SD mengalami konvergensi. Hasil pada gambar 7 menunjukkan bahwa semakin kecil nilai R , semakin cepat algoritma R-LMS mengalami konvergensi.

5. KESIMPULAN

Dari hasil uji coba yang telah dilakukan dapat diambil kesimpulan:

1. Algoritma estimasi dinamis dan algoritma penurunannya terbukti dapat melakukan rekonstruksi citra Super-Resolusi. Hal ini ditunjukkan pada hasil rekonstruksi citra tiap algoritma, dimana terdapat peningkatan pada kualitas resolusi rangkaian citra hasil rekonstruksi.
2. Algoritma Filter Kalman merupakan algoritma paling baik, dilanjutkan dengan algoritma Pseudo-RLS, algoritma R-SD dan akhirnya algoritma R-LMS yang

merupakan algoritma paling lemah dalam melakukan proses rekonstruksi citra Super-Resolusi.

3. Algoritma Filter Kalman memerlukan waktu komputasi paling lama, dilanjutkan dengan algoritma Pseudo-RLS, algoritma R-SD dan algoritma R-LMS yang memerlukan waktu komputasi paling pendek.
4. Dari point no 2 dan 3, dapat disimpulkan bahwa kinerja dari algoritma iteratif (R-SD dan R-LMS) mendekati kinerja algoritma estimasi dinamis (Filter Kalman), dengan waktu komputasi yang lebih singkat.
5. Normalisasi algoritma R-SD dan R-LMS lebih cepat mencapai konvergensi daripada algoritma R-SD dan R-LMS tanpa normalisasi.

DAFTAR PUSTAKA

- [1] R. Soelaiman, I. Subakti dan R. Nugroho, "Penerapan Filter Kalman dan Metode Pseudo-Recursive Least Squares pada Rekonstruksi Citra Dinamis," *Laporan Penelitian*, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia, 2006.
- [2] M. Elad and A. Feuer, "Super-Resolution Reconstruction of Image Sequences", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, September 1999.
- [3] T. K. Moon and W. C. Stirling, *Mathematical Methods and Algorithms for Signal Processing*, Prentice-Hall, Inc., 2000.
- [4] S. C. Park, M. K. Park and M. G. Kang, "Super-Resolution Image Reconstruction: A Technical Overview," *IEEE Signal Processing Magazine*, 2003.