

Prediksi Pemakaian Beban Listrik Menggunakan Metode *Least Mean Square* dan *Compressive Sensing*

Prediction Of Electrical Power Usage Using Least Mean Square and Compressive Sensing

Velia Khaerunnisa^{1*}, Koredianto Usman¹, Muhammad Ary Murti²

¹ Department of Telecommunication Engineering/ Faculty of Engineering Electro

² Department of Electrical Engineering/ Faculty of Engineering Electro

Telkom University, Jl. Telekomunikasi Jl. Terusan Buah Batu phone (022) 7565930

Veliakhrns@student.telkomuniversity.ac.id^{1*}, Korediantousman@telkomuniversity.ac.id¹,

Arymurti@telkomuniversity.ac.id²

Abstrak – Penelitian ini adalah tentang prediksi dari pemakaian beban listrik di suatu gedung dengan teknik prediksi *Least Mean Square (LMS)* dan *Compressive Sensing (CS)*. Penelitian ini penting karena pemakaian energi listrik khususnya pada ruangan dengan jumlah perangkat elektronik berdaya tinggi seperti komputer, AC, dispenser, dapat menyebabkan pemborosan energi serta keterbatasan sumber daya energi yang tidak terbarukan seperti batu bara, minyak bumi dan sebagainya. Oleh karena itu, kemampuan memprediksi pemakaian beban listrik penting artinya untuk melihat trend penggunaan energi ke depan antara lain sebagai langkah antisipasi khususnya pada pihak yang membangkitkan energi tersebut. Pemilihan teknik LMS didasarkan pada kesederhanaan teknik ini untuk prediksi, sedangkan pemilihan teknik CS adalah karena teknik ini belum banyak dipakai pada bidang prediksi, sehingga eksplorasi kapabilitas CS pada bidang prediksi dapat membantu pengayaan pengetahuan khususnya di bidang CS ini. Pada penelitian ini, prediksi dengan LMS dan CS dilakukan dengan tahapan akuisisi data, serta pengolahan data tersebut. Data yang diolah berasal dari pemakaian beban listrik aktual di Gedung P fakultas teknik elektro yang diambil dalam rentang 31 hari dalam periode 16 April 2020-16 Mei 2020 dengan besaran yang diamati adalah *Time Average Energy (TAE)* dengan akuisisi dilakukan pada setiap menit selama 24 jam. Prediksi menggunakan LMS dilakukan dengan beberapa parameter yaitu orde filter serta koefisien adaptasi. Pada CS, pengamatan dilakukan dengan perubahan parameter tingkat sparsitas (k). Kinerja sistem dinilai *root mean square error (RMSE)* dan waktu komputasi. Hasil simulasi dengan data yang diujikan diperoleh nilai RMSE untuk LMS adalah 5,1 dan waktu komputasi sebesar $1,05 \times 10^{-4}$ detik. Di sisi lain, CS nilai RMSE sebesar 3,51 serta waktu komputasi $2,22 \times 10^{-4}$ detik. Dengan demikian LMS memiliki kinerja yang lebih baik dalam bidang waktu komputasi namun CS memiliki kinerja yang lebih baik dalam bidang akurasi RMSE.

Kata Kunci: Energi listrik, Prediksi, LMS, CS.

Abstract – This research is about the prediction of the use of electrical loads in a building with the *Least Mean Square (LMS)* and *Compressive Sensing (CS)* prediction techniques. This research is important because the use of electrical energy, especially in rooms with several high-power electronic devices such as computers, air conditioners, water dispensers, can cause energy waste and limitation to non-renewable energy resources such as coal, petroleum, and so on. Therefore, the ability to predict the use of electricity loads is important to see future trends in energy usage as a precautionary step, especially for those who generate energy. LMS technique was chosen based on its simplicity for prediction, while the CS technique was chosen because it has not been widely used in the field of prediction, so that the exploration of CS capabilities in the field of prediction can help enrich knowledge, especially in this CS field. In this research, prediction using LMS and CS was carried out through stages of data acquisition and data processing. The data processed came from the actual use of electrical loads in the P Building of the Faculty of Electrical Engineering taken within a span of 31 days in the period 16 April 2020–16 May 2020 with the observed magnitude is *Time Average Energy (TAE)* with acquisitions carried out every minute for 24 hours. Prediction using LMS was done with several parameters, which were filter order as

well as the adaptation coefficient. For CS, the observation was done by changing the sparsity level parameter (k). System performance was assessed with Root Mean Square Error (RMSE) and computation time. The simulation results with the tested data show that the RMSE value for LMS is 5.1 and the computation time is $1,05 \times 10^{-4}$ seconds. On the other hand, the RMSE value for the CS is 3.51 and the computation time is $2,22 \times 10^{-4}$ seconds. Thus, LMS has better performance in the computation time, but CS has better performance in the RMSE accuracy.

Keywords: Electrical energy, Prediction, LMS, CS.

1. Pendahuluan

Salah satu energi yang memiliki banyak digunakan untuk kehidupan manusia adalah energi listrik. Kehidupan manusia banyak bergantung dengan listrik sebagai sumber energi seperti televisi, mesin cuci, dispenser, setrika, pendingin ruangan dan sebagainya. Tidak dapat menghitung secara pasti besarnya konsumsi listrik pada waktu tertentu [1]. Jumlah konsumsi beban listrik yang tidak pasti tanpa perkiraan terlebih dahulu dapat berpengaruh pada tagihan listrik kepada konsumen. Oleh karena itu, perlu sistem prediksi pemakaian beban listrik pada suatu gedung yang dapat membantu perencanaan yang efektif dan efisien.

Penelitian terhadap prediksi beban listrik telah banyak dilakukan. Saat ini, metode prediksi terus dikembangkan, diantaranya seperti Prediksi Beban Listrik Pada PT.PLN (PERSERO) Menggunakan Regresi Interval Dengan Neural Fuzzy [1], metode *Neural Fuzzy* menggunakan regresi interval menghasilkan dua nilai keluaran berupa hasil prediksi nilai interval yang dapat diperkirakan antara nilai maksimum sebagai batas atas dan nilai minimum sebagai batas bawah prediksi. Selain itu, telah diterapkan metode *Support Vector Machines* (SVM) menghasilkan prediksi beban listrik jangka pendek sehingga SVM hanya dilakukan sekali *training* untuk mendapatkan solusi optimal dapat berupa regresi ataupun klasifikasi untuk prediksi [2].

Sistem prediksi membutuhkan metode prediksi yang memiliki performa tinggi. Pada penelitian ini dilakukan perbandingan antara metode *Least Mean Square* (LMS) dan *Compressive Sensing* (CS) untuk memprediksi pemakaian beban listrik. LMS merupakan algoritma yang cepat dalam mengolah data dan struktur sederhana biasa digunakan dalam aplikasi pemrosesan sinyal adaptif [3]. Permasalahan yang dapat ditangani oleh sistem adaptif salah satunya yaitu memprediksi besaran sinyal di masa depan. LMS untuk prediksi karena algoritma dengan strukturnya *Finite Impulse Response* (FIR) tidak mengeksplorasi struktur data namun memperbarui bobot baru dari bobot awal untuk prediksi, sehingga LMS memiliki kinerja yang kurang baik.

CS merupakan teknik kompresi untuk merepresentasikan data yang jarang ke dalam data yang sangat kecil melalui proses kompresi serta pengembaliannya ke data semula dengan sempurna melalui proses yang disebut dengan rekonstruksi [4]. CS mengambil informasi terkait dengan struktur data sinyal yakni kejarangannya sehingga jika asumsi ini dipenuhi maka teknik rekonstruksi CS dapat memberikan kinerja baik. Dengan kelebihan ini, CS pada umumnya memiliki kelemahan yaitu waktu komputasi yang lama. Penggunaan CS telah dilakukan pada berbagai bidang, antara lain untuk kompresi citra microwave oleh Fatwasauri dan Rizkina [5]. Penggunaan metode CS untuk keperluan prediksi belum banyak dieksplorasi oleh para peneliti, oleh karena itu, pada penelitian ini kami menyelidiki kinerja CS untuk keperluan prediksi khususnya pada beban daya listrik dan membandingkannya dengan teknik prediksi LMS yang standar pada bidang pengolahan sinyal. Kinerja keduanya dibandingkan dari sisi akurasi dengan nilai RMSE dan waktu komputasi relatif pada komputer yang penulis gunakan.

2. Metode Penelitian

2.19. Prediksi Beban Listrik

Peramalan adalah memprediksi keadaan yang akan datang atas dasar pola-pola keadaan di masa yang lalu. Hal ini dapat dilakukan dengan menggunakan data di masa lalu sehingga memperkirakan keadaan yang terjadi di masa yang akan datang dengan menggunakan model matematis[6]. Terdapat 3 kelompok prediksi beban listrik[7]:

1. **Prediksi Beban Jangka Panjang**
Prediksi jangka panjang adalah memprediksi permintaan beban dengan jangka waktu lebih dari satu tahun.
2. **Prediksi Beban Jangka Menengah**
Prediksi jangka menengah adalah memprediksi permintaan beban dengan jangka waktu satu bulan sampai satu tahun.
3. **Prediksi Beban Jangka Pendek**
Prediksi beban jangka pendek adalah memprediksi permintaan beban dengan jangka waktu beberapa jam sampai satu minggu. Besarnya beban untuk setiap jam bahkan setiap hari ditentukan dengan memerhatikan pola-pola keadaan beban di waktu lalu dengan memerhatikan berbagai informasi yang dapat mempengaruhi besarnya pemakaian beban.

2.20. *Least Mean Square (LMS)*

Algoritma LMS adalah algoritma adaptif yang sederhana dan sering digunakan untuk prediksi *time-series* [8]. Sistem adaptif merupakan sistem yang dapat mengatasi gangguan yang bersifat tidak tetap atau berubah-ubah dan menyesuaikan diri terhadap perubahan tersebut. Pada algoritma LMS, vektor observasi \mathbf{b} diperbarui dari n setiap langkah waktu dalam (1) [9]:

$$b_{n+1} = b(n) + \mu e(n). x(n) \quad (1)$$

dengan $x(n)$ dan $b(n)$ merupakan *vector observasi* serta μ merupakan faktor konvergen parameter tingkat pembelajaran penentu kondisi kestabilan. Sinyal kesalahan dinyatakan seperti Persamaan (2),

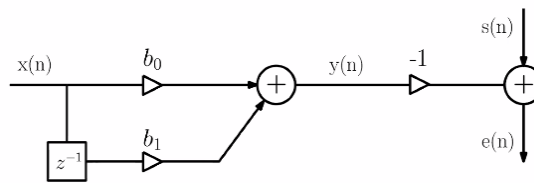
$$e(n) = s(n) - y(n) \quad (2)$$

dengan $s(n)$ merupakan sinyal yang diinginkan. Selama proses pemfilteran diperlukan $s(n)$ untuk mendapatkan nilai kesalahan $e(n)$. Nilai kesalahan didefinisikan sebagai selisih antara sinyal yang diinginkan $s(n)$ dengan keluaran dari filter $y(n)$. Pemilihan μ dipilih seperti Persamaan (3)

$$0 < \mu < 2/(PR_x[0]) \quad (3)$$

dengan P adalah urutan filter dan $R_x [0]$ adalah variansi dari sinyal pengamatan. Tingkat pembelajaran bergantung pada pemilihan μ . Jika μ terlalu kecil tingkat pembelajarannya akan lambat namun jika μ terlalu besar tingkat pembelajaran akan cepat tetapi kurang stabil. Oleh karena itu, pemilihan nilai μ harus tepat [10].

Filter LMS merupakan salah satu filter optimal. Disebut optimal karena filter LMS akan meminimalisasi rata-rata kesalahan kuadrat (*Mean Square Error*, MSE) antara sinyal keluaran filter LMS dengan sinyal target. Filter LMS bekerja mengupdate bobot dari sampel ke sampel. Merealisasikan filter LMS yaitu menggunakan realisasi FIR (*Finite Impulse Response*) karena filter FIR memiliki kelebihan yaitu kestabilan filter terjamin. Filter FIR bersifat lebih mudah direalisasikan karena secara prinsip filter FIR adalah kombinasi linier dari sinyal saat ini dengan sinyal sebelumnya. Prediksi yang dihasilkan dengan realisasi FIR adalah prediksi yang bersifat linier. Realisasi filter FIR dapat dilakukan menggunakan orde 1, orde 2, dan seterusnya bergantung kepada tingkat akurasi yang dikehendaki serta sifat sinyal yang ingin diprediksi atau difilter. Dengan alasan kestabilan maka pada penelitian ini akan digunakan realisasi LMS dengan FIR. Orde filter optimal untuk data beban listrik yang disimulasikan dipilih berdasarkan nilai RMSE yang paling kecil. Secara umum struktur filter LMS dengan realisasi FIR orde 1 dapat dilihat pada Gambar 1.



Gambar 41. Struktur LMS dengan realisasi FIR orde 1.

Pada struktur di Gambar 1, sinyal $s(n)$ merupakan sinyal referensi, sedangkan $x(n)$ adalah sinyal *input* filter. Sinyal $y(n)$ adalah sinyal keluaran filter yang diharapkan sama dengan atau mirip dengan sinyal referensi $s(n)$. Perbedaan antara sinyal $s(n)$ dengan sinyal $y(n)$ adalah sinyal *error* yang dinotasikan dengan $e(n)$. Nilai sinyal $e(n)$ akan diumpan balik ke filter LMS untuk memperbaiki bobot koefisien filter untuk membuat nilai *error* berikutnya lebih kecil. Proses untuk memperbaiki bobot diberikan pada Persamaan (1).

2.21. Compressive Sensing (CS)

CS merupakan metode baru untuk melakukan kompresi. Teknik kompresi dengan CS dilakukan dengan cara mengalikan vektor sinyal sepanjang N , dengan matriks A , dimensi $M \times N$ dengan $M < N$. Berdasarkan dua kondisi yang memungkinkan sinyal dapat dicuplikan dan direkonstruksi kembali sinyal aslinya seperti semula dengan sempurna menggunakan teknik CS [11]. Syarat pertama sinyal yang dicuplik bersifat *sparse* (jarang) dan syarat kedua dilakukan dengan suatu matrik untuk pencuplikan sinyal dengan memenuhi sifat *Restricted Isometric Property* (RIP). CS berupaya untuk prediksi atau memberikan perkiraan data dalam domain terkompresi.

2.3.1. Sinyal Jarang

Sinyal jarang merupakan asumsi utama dalam CS. Sinyal jarang didefinisikan sebagai sinyal sinyal diskrit yang memiliki banyak nol dan sedikit nilai tak nol[11]. Jumlah elemen tak nol pada sinyal jarang disebut dengan sparsitas yang dinotasikan dengan K . Nilai sparsitas minimum adalah sama dengan satu [12]. Konsep sparsitas dapat diperluas sebagai sinyal dengan nilai tak nol sedikit pada basis apapun, seperti basis seperti Fourier, DCT, Hadamard, dan sebagainya.

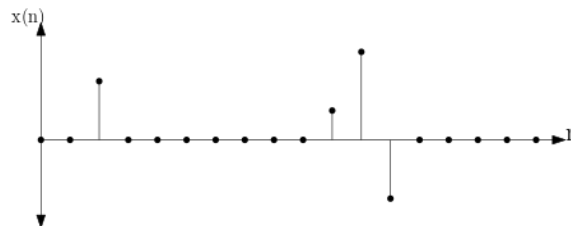
Jika suatu sinyal bersifat jarang pada suatu basis, maka pada umumnya CS memerlukan dua basis untuk proses pencuplikan yaitu basis transformasi *sparsity* Ψ untuk mencari komponen *sparse* dan basis pencuplikan Φ [13]. Misalkan suatu sinyal b merupakan sinyal waktu diskrit yang dinyatakan sebagai *vector* dengan ukuran $N \times 1$. Sinyal x adalah *sparse* pada basis *orthonormal* yang diperoleh dari sinyal b dengan basis transformasi Ψ . Dalam hal ini, hubungan antara x dan b dapat dinyatakan seperti Persamaan (4) [9].

$$b = \Psi x \tag{4}$$

Sinyal b disebut *sparse* dalam basis Ψ . Selanjutnya teknik CS dapat dilakukan dengan mengkompresi sinyal b dengan basis pencuplikan Φ seperti Persamaan (5)

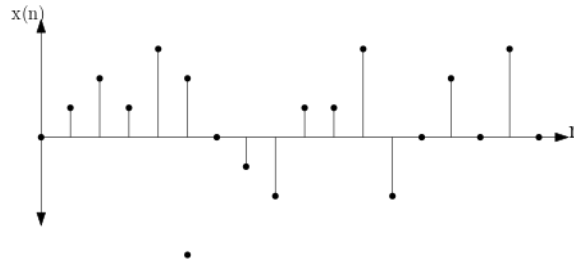
$$D_x = \Phi b = \Phi \Psi x = Ax \tag{5}$$

dengan $A = \Phi \Psi$ dalam matrik kompresi CS. Gambar 2 memperlihatkan contoh sinyal jarang dengan nilai sparsitas $k=4$.



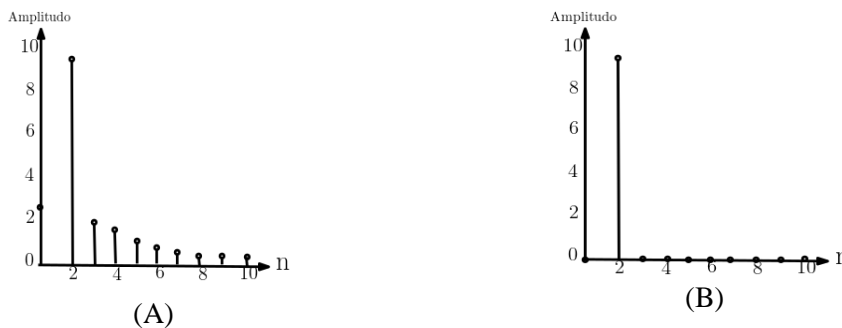
Gambar 42. Sinyal jarang.

Gambar 3 memperlihatkan sinyal tak jarang pada ranah waktu, karena nilai sinyal pada umumnya adalah tak-nol.



Gambar 43. Bukan sinyal jarang.

Sinyal tak jarang pada Gambar 3 dapat diubah ke sinyal jarang dengan mencari basis transformasi yang tepat. Jika kita menggunakan tranformasi Fourier misalnya, maka hasil transformasi sinyal pada Gambar 3 diberikan seperti hasil pada Gambar 4(A). Selanjutnya jika kita terapkan teknik *thresholding*, sebagai contoh, amplitudo sinyal yang kurang dari 2 dijadikan nol, maka hasil *thresholding* adalah sinyal jarang yang tampak pada Gambar 4(B).



Gambar 44. (A) Amplitudo hasil transformasi Fourier dari sinyal pada Gambar 3. (B) Sinyal jarang dengan menerapkan *thresholding* sebesar 2 pada sinyal (A).

2.3.2. Kompresi

Suatu sinyal sudah dibuktikan *sparse*, maka selanjutnya sinyal tersebut dapat dilakukan proses pecuplikan di bawah dari batas nilai Shannon-Nyquist. Proses tersebut secara matematis sinyal asli diproyeksikan ke dalam suatu matriks pengukuran Φ dengan ukuran $M \times N$ dengan $M < N$ [11]. Dengan demikian proses kompresi dinyatakan pada Persamaan (6)

$$y = \Phi b = A x \tag{6}$$

dengan y merupakan sinyal terkompresi dengan panjang M ($M \ll N$), x sinyal jarang atau vektor dengan panjang N dan $A = \Phi \Psi$ adalah matriks kompresi atau matriks pengukuran dengan dimensi $M \times N$.

2.3.3. Rekonstruksi

Rekonstruksi ketika ingin mendapatkan sinyal asli x dari y sinyal terkompresi dan A matriks kompresi. Rekonstruksi CS bukan permasalahan sederhana, karena mempunyai solusi yang banyak dan CS harus memilih dengan tepat salah satu solusi dari sekian banyak solusi tersebut. Algoritma rekontruksi harus membantu untuk memperbaiki dari banyak solusi agar mendapatkan solusi yang tepat.

Syarat kedua Teknik CS dikatakan berhasil adalah terkait pemilihan matriks penginderaan A . Syarat ini merupakan bahwa matriks penginderaan A harus memenuhi sifat *restricted isometric property* RIP [11]. Prinsip RIP menyatakan bahwa norma orde 2 sinyal semuladan hasil rekonstruksi memiliki panjang *euclidean* (norma l_2) tidak berubah [11]. Batasan untuk matriks A yang bersifat RIP dapat dinyatakan pada Persamaan (7)

$$(1 - \delta_x) \|s\|_2 \leq \|As\|_2 \leq (1 + \delta_x) \|s\|_2 \tag{7}$$

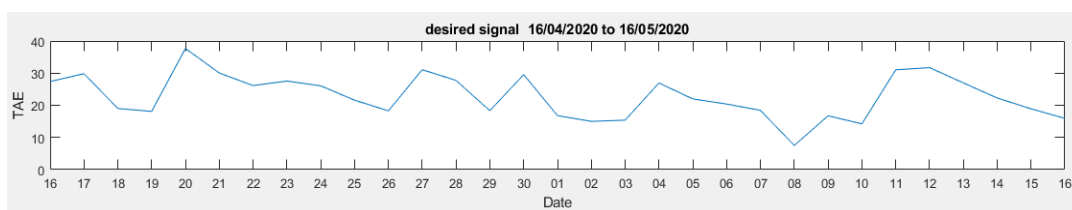
dengan parameter δ_x merupakan suatu bilangan positif kecil.

Terdapat banyak algoritma rekonstruksi CS dan pada penelitian ini digunakan algoritma rekonstruksi Orthogonal Matching Pursuit (*OMP*) oleh karena algoritma OMP memiliki komputasi yang cepat. OMP berfokus pada prinsip *orthogonalization*, untuk memastikan bahwa koefisien sebelumnya tidak dihitung untuk iterasi berikutnya. Rincian lengkap dari algoritma OMP dapat dilihat pada referensi [12]. Kelebihan dari algoritma OMP adalah cepat dan akurat. Selain itu, OMP dapat digunakan untuk sinyal riil maupun kompleks. Algoritma OMP membutuhkan *input* matriks penginderaan A yang berdimensi $M \times N$ serta sinyal kompresi y dengan panjang M untuk memperoleh sinyal rekonstruksi x vektor dengan panjang N .

3. Hasil dan Pembahasan

3.8. Karakteristik Perilaku Penggunaan Energi Listrik Harian Gedung P Fakultas Teknik Elektro di Telkom University

Data dari penelitian ini diperoleh dari pengukuran penggunaan beban pada gedung P di Fakultas Teknik Elektro Universitas Telkom. Pengambilan data dilakukan dalam periode 31 hari dengan interval pengambilan dilakukan setiap 1 menit. Untuk setiap hari, beban penggunaan listrik diakumulasikan untuk diperoleh beban pemakaian listrik sebesar satu hari. Data harian ini ditampilkan sebagaimana Gambar 5 Sumbu datar adalah hari sedangkan sumbu tegak adalah nilai *Time Average Energy* (TAE) dalam satuan kilowatt.



Gambar 45. Pola pemakaian beban energi dalam 31 hari.

Pada Gambar 5 dapat dilihat bahwa terdapat perbedaan pada pola beban pemakaian energi listrik harian yang sangat signifikan. Penyebabnya yaitu perbedaan konsumsi energi listrik setiap harinya. Pada saat hari libur akhir pekan dan hari libur nasional, konsumsi energi listrik tidak sebesar dengan konsumsi listrik pada hari kerja karena pada hari kerja terdapat aktifitas yang jauh lebih banyak menggunakan energi listrik untuk suatu gedung.

Salah satu parameter untuk menghitung akurasi hasil rekonstruksi adalah dengan menggunakan nilai *Root Mean Squared Error* (RMSE). RMSE dinyatakan seperti pada Persamaan (8)

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - s)^2} . \tag{8}$$

Pada persamaan tersebut s menyatakan sinyal asli, sedangkan y menyatakan sinyal rekonstruksi dan N menyatakan panjang sinyal. Nilai RMSE yang kecil menyatakan bahwa sinyal prediksi yang semakin mirip dengan sinyal asli, atau dengan kata lain kinerja algoritma berjalan baik. Di sisi lain, nilai RMSE yang tinggi menyatakan sinyal hasil prediksi yang berbeda jauh dengan sinyal asli, atau dengan kata lain, kinerja algoritma adalah buruk.

Parameter kinerja berikutnya adalah waktu komputasi. Waktu komputasi dirumuskan seperti Persamaan (9)

$$t_c = t_2 - t_1 \tag{9}$$

dengan t_c adalah waktu komputasi, t_2 adalah waktu akhir komputasi dan t_1 adalah waktu awal.

Waktu komputasi pada penelitian ini diukur secara relatif pada simulasi menggunakan spesifikasi komputer yang penulis gunakan. Ada pun prosesor yang digunakan adalah Intel(R) Core(TM) i5-7200U dengan kecepatan prosesor 2,5 GHz, serta RAM 8 GB dan sistem operasi Windows 10 Pro 64 bit dengan perangkat lunak pengolah adalah MATLAB.

3.9. Analisis LMS untuk Prediksi

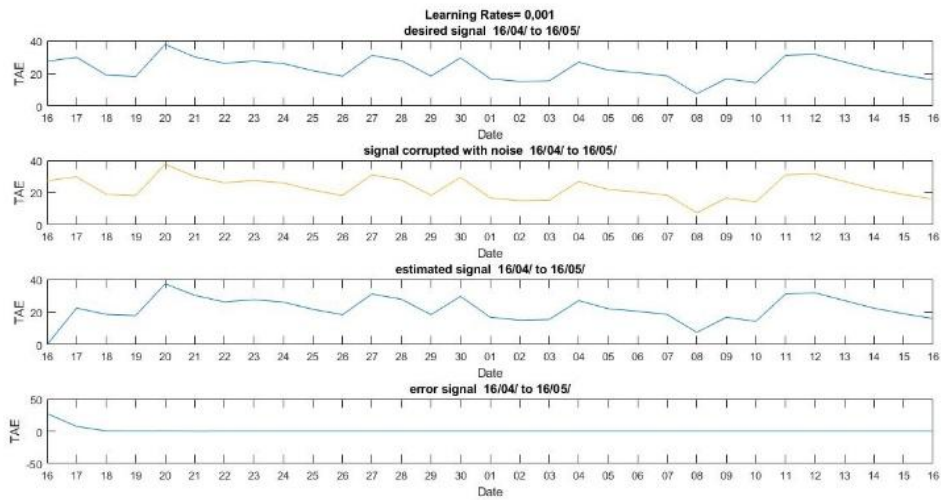
Pada simulasi ini, dilakukan pengujian dengan menggunakan data seperti pada Gambar 6. Untuk setiap simulasi, dilakukan perubahan parameter pada LMS yaitu nilai kecepatan pembelajaran atau *learning rate* (μ), serta orde filter. Nilai μ dipilih dari rentang 0,00001 sampai dengan 0,02. Orde filter dipilih dari 1 sampai 5. Hasil nilai RMSE untuk pengaturan ini adalah seperti pada Tabel 1.

Tabel 16. Uji RMSE untuk tingkat pembelajaran dan orde filter yang berbeda.

μ	Orde				
	1	2	3	4	5
0.00001	21.935	20.134	18.18.618	17.622	16.827
0.00005	16.558	13.274	11.617	10.926	10.633
0.0001	12.805	10.151	9.1855	9.0225	9.1823
0.0005	6.3602	6.8219	7.0348	7.0959	15.522
0.0006	5.9405	6.9097	7.3785	7.4043	63.78
0.0008	5.403	7.5229	12.086	12.376	9938.2
0.001	5.1011	8.8565	56.478	1576.2	4039456.23
0.002	5.7998	10947.94629	25789226.78	3.21×10^{12}	4.36×10^{16}

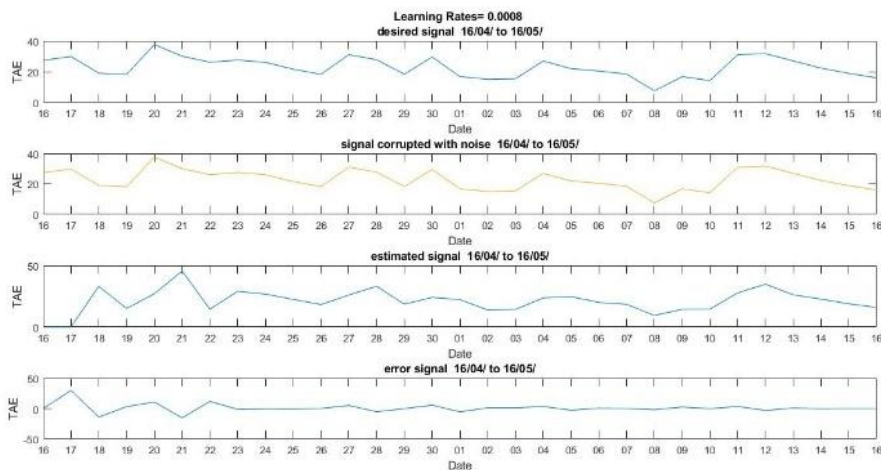
Terlihat pada Tabel 1. bahwa RMSE minimum dicapai dengan menggunakan nilai *learning rate* 0,001 dan orde 1 filter FIR dengan hasil RMSE 5,1.

Hasil tampilan simulasi untuk pengaturan learning rate 0,001 dan FIR orde 1 diperlihatkan pada Gambar 6, gambar pertama yang atas adalah sinyal asli atau sinyal referensi. Gambar kedua adalah sinyal yang telah terkena derau, dalam kasus transmisi. Dengan asumsi bahwa transmisi bersifat sempurna, maka nilai derau dianggap sebagai nol, sehingga sinyal terkena derau adalah sama dengan sinyal asli. Gambar ketiga menunjukkan hasil rekonstruksi LMS dengan perbedaan dengan sinyal asli berupa sinyal *error*, dan sinyal *error* ini ditampilkan pada gambar keempat atau gambar yang paling bawah. Terlihat pada sinyal *error*, bahwa algoritma dapat menyesuaikan dengan cepat menggunakan learning rate 0,001 tersebut, ditandai dengan adanya nilai *error* pada bagian awal sinyal sampai dengan data ke 3 yaitu tanggal 18 April 2020, sedangkan bagian tengah dan akhir, nilai *error* telah mendekati nol, dengan kata lain, LMS sudah dapat memprediksi sinyal dengan baik.



Gambar 46. Sinyal prediksi dan sinyal kesalahan pada orde 1.

Sebagai perbandingan, Gambar 7 memperlihatkan hasil simulasi untuk orde filter 2 dan nilai *learning rate* 0,0008. Pada kurva dari Gambar 7, terlihat bahwa filter orde 2 memerlukan waktu belajar yang lebih panjang dibandingkan dengan orde 1. Hal ini terlihat dari nilai *error* yang masih tinggi untuk nilai data sampai dengan data ke 7 yaitu tanggal 22 April 2020 sedangkan orde 1 telah mencapai nilai *error* yang rendah untuk data ke 3 yaitu 18 April 2020. Hasil simulasi ini menunjukkan bahwa sinyal beban listrik yang digunakan bersifat dinamis dengan filter LMS orde 1 yang cocok untuk memprediksinya.



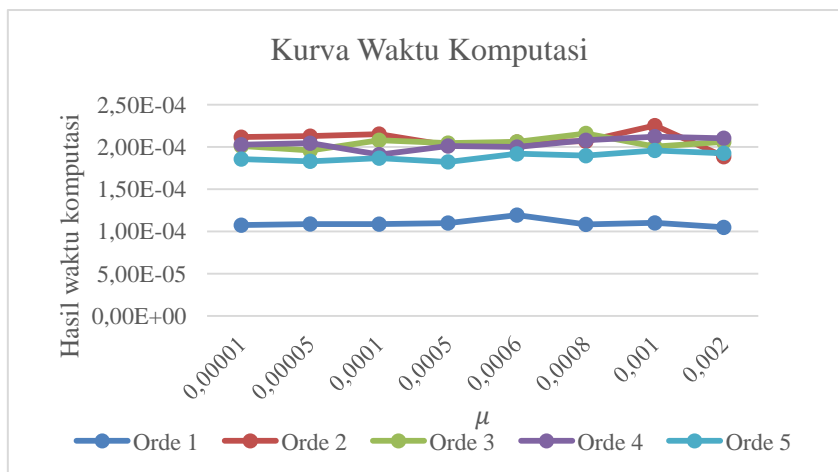
Gambar 47. Sinyal prediksi dan sinyal kesalahan pada orde 2.

Terkait dengan waktu komputasi, pengukuran dilakukan untuk setiap parameter *learning rate* dan orde filter yang sama seperti simulasi sebelumnya. Hasil simulasi pengukuran waktu komputasi adalah seperti yang terlihat pada Tabel 2.

Tabel 17. Waktu komputasi untuk tingkat pembelajaran dan orde filter yang berbeda (dalam detik).

μ	orde				
	1	2	3	4	5
0.00001	$1,07 \times 10^{-4}$	$2,11 \times 10^{-4}$	$2,01 \times 10^{-4}$	$2,03 \times 10^{-4}$	$1,86 \times 10^{-4}$
0.00005	$1,09 \times 10^{-4}$	$2,13 \times 10^{-4}$	$1,95 \times 10^{-4}$	$2,04 \times 10^{-4}$	$1,83 \times 10^{-4}$
0.0001	$1,09 \times 10^{-4}$	$2,15 \times 10^{-4}$	$2,08 \times 10^{-4}$	$1,91 \times 10^{-4}$	$1,87 \times 10^{-4}$
0.0005	$1,10 \times 10^{-4}$	$2,01 \times 10^{-4}$	$2,04 \times 10^{-4}$	$2,01 \times 10^{-4}$	$1,82 \times 10^{-4}$
0.0006	$1,19 \times 10^{-4}$	$2,06 \times 10^{-4}$	$2,06 \times 10^{-4}$	2×10^{-4}	$1,92 \times 10^{-4}$
0.0008	$1,08 \times 10^{-4}$	$2,07 \times 10^{-4}$	$2,16 \times 10^{-4}$	$2,08 \times 10^{-4}$	$1,89 \times 10^{-4}$
0.001	$1,1 \times 10^{-4}$	$2,25 \times 10^{-4}$	2×10^{-4}	$2,12 \times 10^{-4}$	$1,96 \times 10^{-4}$
0.002	$1,05 \times 10^{-4}$	$1,88 \times 10^{-4}$	$2,06 \times 10^{-4}$	$2,1 \times 10^{-4}$	$1,92 \times 10^{-4}$

Hasil kurva dari waktu komputasi sebagaimana yang diberikan dari Tabel 2 ditampilkan dalam bentuk kurva seperti pada Gambar 8.



Gambar 48. Waktu komputasi untuk tingkat pembelajaran dan orde filter yang berbeda.

Terlihat pada Gambar 8, bahwa waktu komputasi minimum dicapai dengan menggunakan nilai kecepatan pembelajaran 0,002 dan orde 1 filter FIR dengan hasil waktu komputasi $1,05 \times 10^{-4}$ detik.

3.10. Analisis CS untuk Prediksi

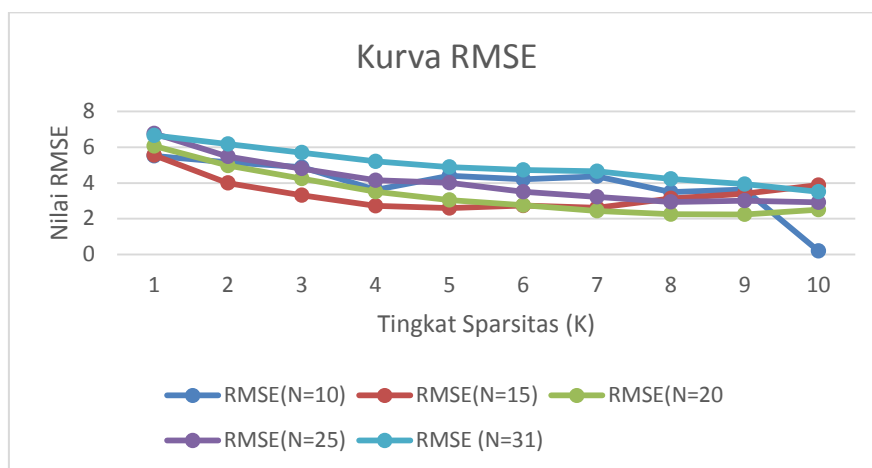
Pada simulasi ini, dilakukan pengujian dengan menggunakan data seperti pada Gambar 6. Untuk setiap simulasi, dilakukan perubahan parameter pada LMS yaitu nilai tingkat sparsitas (K), serta panjang data yang berbeda. Nilai K dipilih dari rentang 1 sampai dengan 10. Panjang data dipilih 10, 15, 20, 25 dan 31. Hasil nilai RMSE untuk pengaturan ini adalah seperti pada Tabel 3.

Tabel 18. Uji RMSE untuk tingkat sparsitas dan panjang sinyal yang berbeda.

Tingkat Sparsitas (K)	Panjang Data (N)				
	10	15	20	25	31
1	5.5144	5.5574	6.0636	6.7665	6.6557
2	5.1555	4	4.9686	5.4821	6.1779
3	4.8788	3.3139	4.2439	4.8061	5.6997
4	3.6198	2.7185	3.5105	4.1472	5.2041
5	4.4071	2.5931	3.0461	4.0109	4.8948
6	4.2087	2.734	2.7526	3.5038	4.7279

7	4.3751	2.6103	2.4424	3.2217	4.6521
8	3.4893	3.1376	2.2507	2.9401	4.2262
9	3.6429	3.3995	2.2446	3.0107	3.934
10	0.1982	3.8772	2.5031	2.9143	3.5101

Hasil kurva dari nilai RMSE sebagaimana yang diberikan dari Tabel 3 ditampilkan dalam bentuk kurva seperti pada Gambar 9.

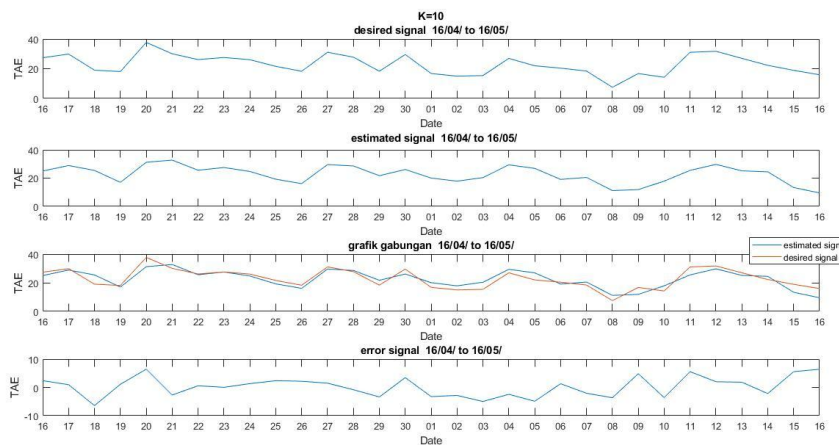


Gambar 49. Nilai RMSE untuk tingkat sparsitas dan panjang sinyal yang berbeda.

Terlihat pada gambar 9 bahwa RMSE minimum dicapai pada algoritma CS dengan menggunakan nilai tingkat sparsitas 10 dengan panjang data 10 dengan nilai RMSE 0,01982.

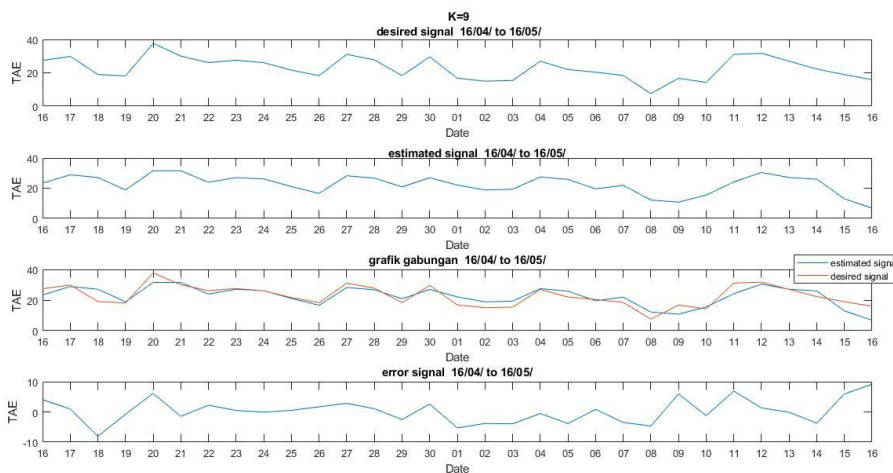
Jika melihat dari Tabel 3 dan Gambar 9 jika dilihat dari panjang data, ketika panjang data 10 maka RMSE minimum dicapai dengan tingkat sparsitas 10 dengan nilai RMSE 0,01982. Ketika panjang data 15 maka RMSE minimum dicapai dengan tingkat sparsitas 5 dengan nilai RMSE 2,59. Ketika panjang data 20 maka RMSE minimum dicapai dengan tingkat sparsitas 9 dengan nilai RMSE 2,24. Ketika panjang data 25 maka RMSE minimum dicapai dengan tingkat sparsitas 10 dengan nilai RMSE 2,91 dan ketika panjang data 31 maka RMSE minimum dicapai dengan tingkat sparsitas 10 dengan nilai RMSE 3.51.

Hasil tampilan simulasi untuk pengaturan tingkat sparsitas 10 dengan panjang data 31 diperlihatkan pada Gambar 10, gambar pertama yang atas adalah sinyal asli atau sinyal referensi. Gambar kedua menunjukkan sinyal hasil prediksi CS. Gambar ketiga menunjukkan gabungan dari sinyal asli dan sinyal hasil prediksi dengan perbedaan dengan sinyal asli berupa sinyal *error*, dan sinyal *error* ini ditampilkan pada gambar keempat atau gambar yang paling bawah. Terlihat pada sinyal *error*, bahwa algoritma dapat menyesuaikan dengan cepat menggunakan tingkat sparsitas 10 tersebut, ditandai dengan adanya nilai *error* pada bagian awal sinyal sampai dengan data terakhir, nilai *error* tidak terlalu tinggi atau mendekati nol dan pada gambar ketiga grafik gabungan terlihat sinyal prediksi mirip dengan sinyal data aktual. dengan kata lain, CS sudah dapat memprediksi sinyal dengan baik.



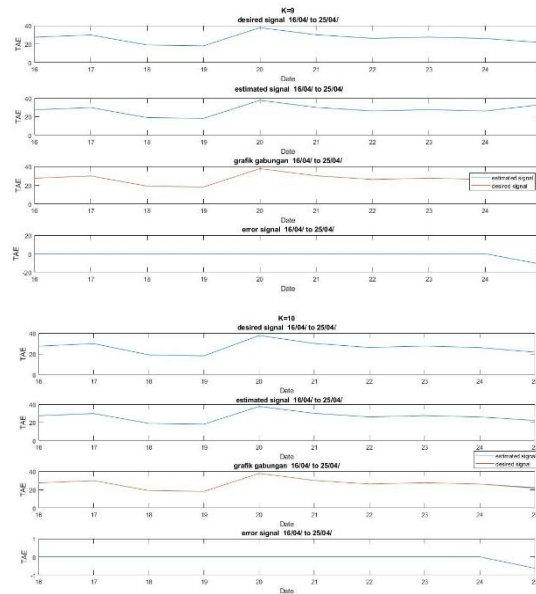
Gambar 50. Sinyal prediksi dan sinyal kesalahan dengan panjang data 31 saat sparsitas 10.

Sebagai perbandingan, Gambar 11 memperlihatkan hasil simulasi untuk panjang sinyal 31 dan tingkat sparsitas 9. Pada kurva dari Gambar 11, terlihat bahwa tingkat sparsitas memerlukan waktu belajar yang lebih panjang dibandingkan dengan menggunakan tingkat sparsitas 10. Hal ini terlihat dari nilai *error* yang masih tinggi dibandingkan dengan nilai *error* saat tingkat sparsitas 10 pada Gambar 10. Hasil simulasi ini menunjukkan bahwa sinyal beban listrik yang digunakan bersifat dinamis dengan CS dengan sparsitas 10 yang cocok untuk memprediksi pada panjang data 31.



Gambar 51. Sinyal prediksi dan sinyal kesalahan dengan panjang data 31 saat sparsitas 9.

Pada Gambar 10 dan Gambar 11 data dengan sparsitas dapat merepresentasikan sinyal data prediksi dengan sinyal data aktual. Disetiap grafik, sinyal error selama iterasi ditampilkan untuk tingkat sparsitas (K) yang berbeda. Seperti yang diperkirakan, algoritma akan lebih cepat konvergen atau akan lebih cepat menghasilkan sinyal data prediksi lebih mirip dengan sinyal data aktual saat tingkat sparsitas ditingkatkan. Terlihat bahwa kecepatan konvergensi lebih tinggi ketika data yang lebih panjang. Namun jika nilai K ditingkatkan mendekati atau sama dengan jumlah data maka tidak berpengaruh lagi terhadap perubahan sinyal prediksi seperti pada Gambar 12.



Gambar 52. Sinyal prediksi dan sinyal kesalahan dengan panjang data 10 saat sparsitas 9 dan 10

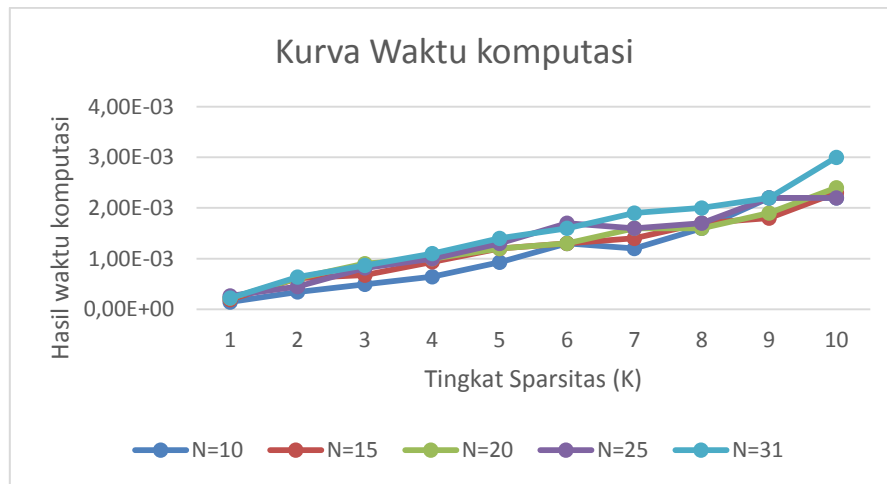
Untuk melihat kecepatan konvergensi algoritma CS, nilai data pemakaian beban listrik aktual dan nilai prediksinya untuk panjang data dan tingkat sparsitas yang berbeda Gambar 10 dan Gambar 11 seperti yang ditunjukkan pada grafik nilai prediksi beban listrik mendekati lebih cepat dengan nilai data aktual saat tingkat sparsitas ditingkatkan.

Terkait dengan waktu komputasi, pengukuran dilakukan untuk setiap parameter tingkat sparsitas dan panjang data yang sama seperti simulasi sebelumnya. Hasil simulasi pengukuran waktu komputasi adalah seperti yang terlihat pada Tabel 4.

Tabel 19. Uji waktu komputasi untuk tingkat sparsitas dan panjang data yang berbeda.

Tingkat Sparsitas (K)	Panjang Data (N)				
	10	15	20	25	31
1	1.42E-04	1.90E-04	2.22E-04	2.64E-04	2.22E-04
2	3.43E-04	6.10E-04	6.14E-04	4.48E-04	6.36E-04
3	4.93E-04	6.73E-04	9.01E-04	8.16E-04	8.66E-04
4	6.41E-04	9.37E-04	1.00E-03	1.00E-03	1.10E-03
5	9.28E-04	0.0012	0.0012	0.0013	0.0014
6	0.0013	0.0013	0.0013	0.0017	0.0016
7	0.0012	0.0014	0.0016	0.0016	0.0019
8	0.0016	0.0017	0.0016	0.0017	0.002
9	0.0022	0.0018	0.0019	0.0022	0.0022
10	0.0022	0.0023	0.0024	0.0022	0.003

Hasil kurva dari waktu komputasi sebagaimana yang diberikan dari Tabel 4 ditampilkan dalam bentuk kurva seperti pada Gambar 13.



Gambar 53. Waktu komputasi untuk tingkat sparsitas dan panjang data yang berbeda.

Terlihat pada Gambar 13 bahwa waktu komputasi minimum dicapai pada algoritma CS dengan menggunakan nilai tingkat sparsitas 1 dengan panjang data 10 dengan hasil waktu komputasi $1,42 \times 10^{-4}$.

Pada Tabel 4 dan Gambar 13 jika dilihat dari panjang data, ketika panjang data 10 maka waktu komputasi minimum dicapai dengan tingkat sparsitas 1 dengan hasil waktu komputasi $1,42 \times 10^{-4}$. Ketika panjang data 15 maka waktu komputasi minimum dicapai dengan tingkat sparsitas 1 dengan hasil waktu komputasi $1,90 \times 10^{-4}$. Ketika panjang data 20 maka waktu komputasi minimum dicapai dengan tingkat sparsitas 1 dengan hasil waktu komputasi $2,22 \times 10^{-4}$. Ketika panjang data 25 maka waktu komputasi minimum dicapai dengan tingkat sparsitas 1 dengan hasil waktu komputasi $2,64 \times 10^{-4}$ dan panjang data 31 maka waktu komputasi minimum dicapai dengan tingkat sparsitas 1 dengan hasil waktu komputasi $2,22 \times 10^{-4}$.

4. Kesimpulan

Penelitian ini mengajukan metode untuk memprediksi pemakaian beban listrik dengan membandingkan performa metode *Least Mean Square* (LMS) dan *Compressive Sensing* (CS) berdasarkan tingkat akurasi RMSE dan waktu komputasi algoritma yang digunakan. Pada algoritma CS dan LMS memprediksi data pemakaian beban listrik dari tanggal 16 April 2020 sampai 16 Mei 2020.

Hasil analisis menggunakan algoritma CS didapatkan hasil waktu komputasi minimum $2,22 \times 10^{-4}$ dengan tingkat sparsitas 1. Jika dilihat dari akurasi RMSE minimum dicapai dengan tingkat sparsitas 10 dengan nilai RMSE 3,5. Ketika dianalisis menggunakan algoritma LMS didapatkan hasil waktu komputasi minimum $1,05 \times 10^{-4}$ dengan nilai kecepatan pembelajaran 0,002 dan orde 1 filter FIR. Jika dilihat dari akurasi RMSE minimum dicapai dengan menggunakan nilai kecepatan pembelajaran 0,001 dan orde 1 filter FIR dengan hasil RMSE 5,1011.

Pada analisis penelitian ini, jika mengutamakan performa waktu komputasi maka algoritma LMS lebih baik untuk sistem prediksi pemakaian beban listrik menggunakan kecepatan pembelajaran 0,002 dengan orde 1 hasil waktu komputasi $1,05 \times 10^{-4}$. Namun, jika mengutamakan performa akurasi RMSE, algoritma CS lebih baik untuk sistem prediksi pemakaian beban listrik menggunakan sparsitas 10 menghasilkan nilai RMSE 3,51.

Referensi

- [1] P. Ayuningtyas, D. Triyanto, and T. Rismawan, "Prediksi Beban Listrik Pada PT.PLN (PERSERO) Menggunakan Regresi Interval Dengan Neural Fuzzy," *J. Coding, Sist. Komput. UNTAN*, vol. 04, no. 1, pp. 1–10, 2016.
- [2] V. Apriana and R. I. Handayani, "Analisis Algoritma Prediksi Untuk Menghasilkan Prediksi Beban Listrik Jangka Pendek," *J. Ilmu Pengetah. dan Teknol. Komput.*, vol. 3, no. 1, pp. 73–78, 2017.
- [3] B. Jalal, S. Member, X. Yang, S. Member, Q. Liu, and S. Member, "Fast and Robust Variable Step Size LMS Algorithm for Adaptive Beamforming," vol. 0, no. 1, pp. 1–4, 2020, doi: 10.1109/LAWP.2020.2995244.
- [4] L. Huang, Q. Lin, X. Zhang, and P. Shuai, "Compressed Sensing," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 53, no. 1, pp. 137–147, 2017, doi: 10.1109/TAES.2017.2649698.
- [5] I. Fatwasauri and M. Rizkinia, "Compressive sensing image reconstruction with total variation and $l_{2,1}$ norm for microwave imaging," *2nd IEEE Int. Conf. Innov. Res. Dev. ICIRD 2019*, no. Mi, pp. 2–5, 2019, doi: 10.1109/ICIRD47319.2019.9074702.
- [6] "Peramalan beban listrik jangka menengah pada sistem kelistrikan kota samarinda," no. 09, pp. 113–121, 2015.
- [7] R. Dwisatya and M. R. Kirom, "Prediksi Beban Listrik Jangka Pendek Menggunakan Algoritma Feed Forward Back Propagation dengan Mempertimbangkan Variasi Tipe Hari," *Electrans*, vol. 14, no. 1, pp. 34–40, 2016.
- [8] T. Shu, J. Chen, V. K. Bhargava, and C. W. De Silva, "An Energy-Efficient Dual Prediction Scheme Using LMS Filter and LSTM in Wireless Sensor Networks for Environment Monitoring," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6736–6747, 2019, doi: 10.1109/IIOT.2019.2911295.
- [9] A. Ukte and A. Kizilkaya, "Comparing the performances of least mean squares based multirate adaptive filters," in *2016 26th International Conference Radioelektronika, RADIOELEKTRONIKA 2016*, 2016, no. 6, pp. 325–329, doi: 10.1109/RADIOELEK.2016.7477383.
- [10] H. G. Alfarizi, F. T. Elektro, U. Telkom, and A. Filter, "Penghapusan Derau Pada Sinyal Wicara Menggunakan Sistem Least Mean Square Noise Cancellation of Speech Signal Using Dual Microphone System With Discrete Cosine Transform Least Mean Square," vol. 5, no. 2, pp. 2161–2168, 2018.
- [11] K. Usman, I. Irawati, A. B. Suksmono, and H. Gunawan, "Teknik Penginderaan Kompresif: Prinsip dan Aplikasinya," vol. 2, no. 9, 2016.
- [12] I. Orović, V. Papić, C. Ioana, X. Li, and S. Stanković, "Compressive Sensing in Signal Processing: Algorithms and Transform Domain Formulations," *Math. Probl. Eng.*, p. 16, 2016, doi: 10.1155/2016/7616393.
- [13] N. Hafizah, "Perancangan dan Implementasi Compressive Sensing Untuk Sistem Audio Watermarking dengan Metode Kombinasi Discrete Cosine Transform dan Discrete Wavelet Transform," in *Ekp*, 2017, vol. 13, no. 3, pp. 1576–1580.