

Sun Position Forecasting Menggunakan Metode RNN – LSTM Sebagai Referensi Pengendalian Daya Solar Cell

Eros Fikri Syahram¹, M.Machmud Effendy², Novendra Setyawan³
Jurusan Teknik Elektro, Fakultas Teknik, Universitas Muhammadiyah Malang
e-mail: erosfikri8@gmail.com

Abstract - Solar energy is used as electrical energy for daily life by using solar cell. There are several models for maximizing the absorption of solar cell power by using reflectors and solar tracking. Reflector focuses sunlight on the solar cell to get maximum light. Solar tracking real-time tracking the sun by moving two motors and sensors to read the direction of the sun's angle. Both have the disadvantage of tracking the sun continuously. So it is necessary to optimize the system to determine the tilt angle of the actuator and be able to streamline the intensity of sunlight that can be absorbed by solar cells with the identification system that can be done in realtime or directly to minimize the occurrence of power wastage. Prediction is one of the most important elements for decision-makers of the problems that occur above. In this paper, the method used in predicting the sun's angle is by using the Recurrent Neural Network (RNN) method which has a Long-Short Term Memory (LSTM) structure in the RNN system. In this research, the data used to predict the sun angle from the sunscal.org site in the area of the University of Muhammadiyah Malang campus 3 with a period of one year. Testing with the RNN-LSTM structure is done with two different prediction models, namely weekly, monthly, and annual data for daily data results, and hourly data for daily data for one week. The test results on weekly data detect Root Mean Square Error of 0.12%, the monthly data is 0.1% and the annual data is 0.24%. The monthly data model has the fewest error values, so predictive data has a high degree of accuracy.

Keywords — Sun Panel , RNN (Recurrent Neural Network), LSTM (Long Short Term Memory), Prediction, Time series

Abstrak— Energi matahari yang besar dimanfaatkan sebagai energi listrik untuk kehidupan sehari-hari yaitu dengan menggunakan solar cell. Metode dalam memaksimalkan penyerapan daya solar cell ada beberapa model yaitu dengan menggunakan reflektor dan solar tracking. Reflektor memusatkan cahaya matahari kepada solar cell agar mendapatkan cahaya yang maksimal. Solar tracking melakukan penjejakan matahari secara realtime dengan menggerakkan dua buah motor dan sensor untuk membaca arah sudut matahari. Keduanya mempunyai kekurangan yaitu melacak matahari terus menerus. Sehingga diperlukan pengoptimalan sistem untuk menentukan sudut kemiringan pada aktuator dan mampu mengefisienkan intensitas cahaya matahari yang dapat diserap oleh sel surya dengan sistem identifikasinya dapat dilakukan secara realtime atau langsung dengan tujuan meminimalisir terjadinya pemborosan daya. Prediksi adalah salah satu unsur yang paling penting untuk pengambil keputusan dari masalah yang terjadi diatas. Dalam penulisan ini metode yang digunakan

dalam memprediksi sudut matahari yaitu dengan menggunakan metode Recurrent Neural Network (RNN) yang didalam sistem RNN terdapat struktur Long – Short Term Memory (LSTM). Dalam penelitian ini data yang digunakan untuk memprediksi sudut matahari dari situs sunscal.org di wilayah Universitas Muhammadiyah Malang kampus 3 dengan periode waktu satu tahun. Pengujian dengan struktur RNN – LSTM dilakukan dengan dua model prediksi yang berbeda, yaitu data mingguan, bulanan dan tahunan untuk hasil data harian, dan data per-jam untuk data harian selama satu minggu. Hasil pengujian pada data mingguan mendeteksi Root Mean Square Error sebesar 0.12 %, pada data bulanan bernilai 0.1% dan pada data tahunan bernilai 0.24 %. Model data bulanan memiliki nilai error yang paling sedikit, sehingga data prediksi memiliki tingkat akurasi yang tinggi.

Kata Kunci— Panel Surya, RNN (Recurrent Neural Network), LSTM (Long Short Term Memory), Prediksi, Time Series

I. PENDAHULUAN

Indonesia juga negara yang dilewati garis khatulistiwa menjadikan pencahayaan cahaya matahari didapatkan sangat baik, sehingga dapat dimanfaatkan untuk keperluan pembangkitan energi listrik dari cahaya matahari menggunakan sel surya [1]. Energi ini dapat dimanfaatkan untuk mengatasi permasalahan yang terjadi di masa yang akan datang. Untuk mengoptimalkan energi listrik yang dihasilkan sel surya diperlukan sebuah sistem prediksi arah sudut radiasi matahari.

Salah satu cara memaksimalkan energi keluaran dari sel surya dengan menggunakan reflektor (cermin pemantul sinar matahari). Masalah yang terjadi saat ini adalah cahaya yang diserap oleh sel surya hanya mengarah kesatu arah (statis). Sehingga dibutuhkan sistem yang dapat mengarahkan sudut elevasi yang tepat pada aktuator pada reflektor, dan mampu mengoptimalkan penyerapan cahaya matahari yang diserap oleh sel surya dengan sistem pendekatannya dapat dilakukan secara langsung (*realtime*) [2], [3].

Solar tracking melakukan penjejakan matahari secara *realtime* dengan menggerakkan dua buah motor dan sensor untuk membaca arah sudut matahari [4]. *Solar tracking* mempunyai kekurangan yaitu daya yang dihasilkan oleh solar cell digunakan juga untuk menggerakkan kedua buah motor dan

sensor yang terus menerus membaca pergerakan arah sudut matahari.

Perancangan sistem ini bertujuan untuk mendapatkan hasil prediksi posisi matahari yang dimana data tersebut menjadi acuan untuk refrensi sudut optimal sel surya. Dikarenakan matahari berpindah dari timur ke barat, sel surya tidak bisa menerima cahaya matahari secara maksimal karena kebanyakan dari sel surya modelnya permanen dengan sudut elevasi yang statis[4], [5], [6].

Dalam menjawab permasalahan tersebut, di temukan solusi untuk melacak posisi matahari secara *realtime*, Pada penelitian [7] merancang prototipe *solar tracking system*. Prototipe ini bekerja dengan menggerakkan motor pada *solar tracking system*, hal ini menyebabkan pemborosan energi listrik. Sehingga dibutuhkan sistem yang dapat mengurangi pemborosan, dengan cara memprediksi sudut matahari yang bertujuan agar motor tidak bergerak secara terus - menerus. Metode yang bisa digunakan dalam memproses data *sequential* adalah *Recurrent Neural Network* (RNN), pada penelitian [8] RNN digunakan untuk memprediksi energi listrik dalam waktu tertentu (hari, mingguan, dan bulanan). Penelitian ini menghasilkan *error* sebesar 0.25 % pada mode data bulanan. Hal ini menunjukkan bahwa RNN sangat cocok untuk menghasilkan data prediksi yang akurat dalam jangka panjang.

II. METODE PENELITIAN

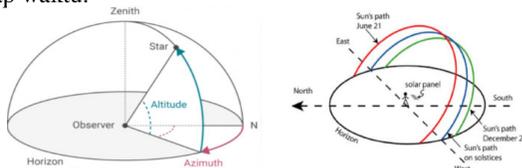
A. Energi Matahari

Energi panas yang berupa cahaya dengan intensitas radiasi yang cukup besar banyak memberikan banyak kebermanfaatannya. Dengan tingginya intensitas tersebut dapat memberikan energi terbarukan yang bebas dari limbah dan polusi. Oleh sebab itu energi dari panas matahari diteliti dalam pengoptimalan penyerapan sebagai sumber energi terbarukan dimasa yang akan datang.

B. Geometri Matahari

Bumi selalu berputar mengelilingi matahari dan memiliki rotasi harian pada poros kutubnya sendiri, penting bagi sistem panel surya untuk menentukan secara matematis atau secara optikal menentukan lokasi / posisi matahari [9].

Selain itu titik awal terbitnya matahari setiap bergeser dari selatan ke utara **Gambar 1** dibawah ini menunjukkan posisi *Azimuth & Altitude / Altitude* dan posisi terbit yang bergeser setiap waktu.



Gambar 1. (a) *Azimuth & Altitude* Matahari (b) Perubahan Sudut Matahari Terbit

C. Preprocessing

Tahap preprocessing yang dilakukan adalah proses normalisasi data. Normalisasi data diperlukan karena nilai data mentah memiliki rentang nilai yang terlalu besar. Sehingga ketika diproses, hasilnya akan tidak bagus atau bahkan tidak nilai menjadi *null*. Berikut teknik normalisasi yang umum digunakan untuk *scaling data*:

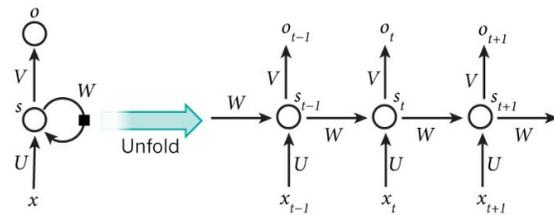
a. Min - Max Scaling

Min - Max Scaling bekerja dengan menyesuaikan data dalam rentang (range) tertentu (range nilai minimum hingga nilai minimum), dengan rentang yang biasa digunakan adalah 0 hingga 1. Berikut ini adalah uraian matematisnya[10]:

$$X_{sc} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

D. RNN (Recurrent Neural Network)

RNN adalah jenis arsitektur jaringan saraf tiruan yang pemrosesannya dipanggil berulang untuk memroses masukan yang biasanya adalah data *sequential*. Dalam tiap pemrosesan, output yang dihasilkan bukan hanya merupakan fungsi dari sample itu saja, tetapi juga berdasarkan *state* internal yang merupakan hasil pemrosesan pada sampel - sampel sebelumnya (atau setelahnya, pada *bidirectional RNN*)[12].



Gambar 2. Proses RNN saat menghitung proses didepanya

Pada gambar diatas merupakan diagram yang menunjukkan RNN pada posisi yang tidak dibuka ke jaringan penuh. Dengan membuka gulungan (*unrolling*) RNN maka sama seperti kita menuliskan seluruh jaringan dengan urutan (*sequence*) secara lengkap.

Keterangan :

1. x_t adalah input pada setiap langkah (*time step*).
2. s_t adalah *hidden* pada setiap *time step* t.

Hidden state bisa disebut sebagai “*memory*” pada sebuah jaringan yang berfungsi menyimpan hasil kalkulasi dan rekaman yang telah dilakukan. s_t dihitung sesuai dengan *hidden state* sebelumnya dan pada input yang sedang di proses. Berikut rumus fungsi s_t :

$$s_t = f(Ux_t + WS_{t-1}) \quad (2)$$

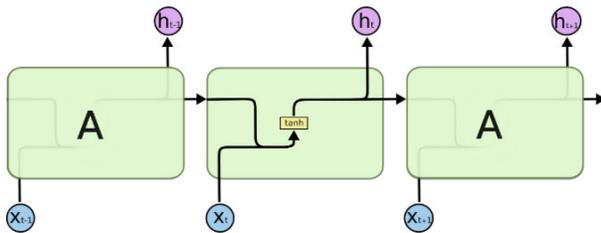
Fungsi f biasanya adalah non-linearitas seperti tanh atau ReLU. Kemudian s_{t-1} digunakan untuk menghitung *hidden*

state yang pertama, biasanya pada inialisasi selalu diawali dari 0.

3. O_t adalah output untuk setiap step t . Berikut rumus O_t :

$$O_t = \text{softmax}(Vs_t) \quad (3)$$

RNN memiliki struktur yang sangat sederhana, dimana didalamnya hanya terdapat lapisan tanh tunggal seperti pada gambar di bawah ini :

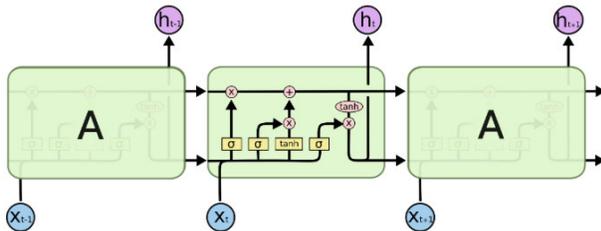


Gambar 3. RNN sederhana

RNN sederhana akan mengalami masalah *vanishing gradient* seiring bertambahnya panjang data berurut yang akan ditraining. *Vanishing gradient* adalah situasi dimana nilai gradient yang digunakan untuk mengupdate bobot pada neuron bernilai 0 atau mendekati 0.

E. LSTM (Long Short Term Memory)

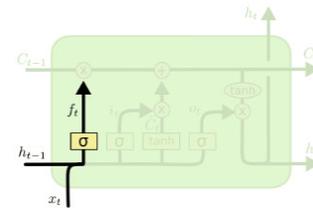
LSTM (Long Short Term Memory) memiliki isi cell yang lebih kompleks didalamnya. Berbeda dengan RNN yang hanya memiliki satu struktur (tanh), LSTM memiliki empat struktur yang akan memproses data *time series* (berurut).



Gambar 4. Struktur di dalam LSTM

Pada Gambar 4 dapat dilihat jika LSTM memiliki dua keluaran (*output*), yang mana satu keluaran akan digunakan sebagai inputan *cell* selanjutnya dan keluaran satunya lagi adalah *cell state*. Di dalam setiap cell terdapat empat struktur yang memiliki fungsi masing – masing untuk memproses data. Berikut penjelasan masing – masing struktur:

1) Alur Informasi Forget Gate

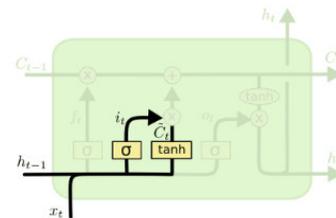


Gambar 5. Alur informasi forget gate

Langkah pertama LSTM adalah dengan memutuskan informasi apa yang akan disimpan atau dibuang. Keputusan ini dibuat oleh lapisan sigmoid yang disebut “*forget gate*”. Keputusan terlihat dari $h_t - 1$, x_t dan angka *output* yang dihasilkan dari formula (3). Jika angka yang dihasilkan mendekati 0 maka data akan dibuang, tetapi jika angka yang dihasilkan mendekati 1 maka data akan disimpan. Berikut rumus *forget gate* :

$$f_t = \sigma(W_f \cdot [h_t - 1, x_t] + b_f) \quad (4)$$

2) Alur Informasi Input Gate



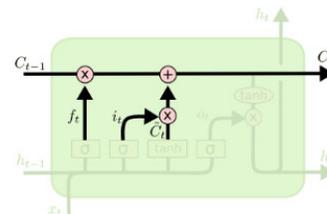
Gambar 6. Alur Informasi Input Gate

Selanjutnya adalah *input gate* (lapisan neuron dengan fungsi aktivasi sigmoid dan mempunyai fungsi aktivasi tanh). Selanjutnya, layer tanh menghasilkan nilai vector kandidat baru yaitu C_t , yang akan di gunakan di state. Gate ini menentukan bagian mana yang akan di *update*. Berikut adalah rumus untuk bagian *input gate* :

$$i_t = \sigma(W_i \cdot [h_t - 1, x_t] + b_i) \quad (5)$$

$$C_t = \tanh(W_C \cdot [h_t - 1, x_t] + b_C) \quad (6)$$

3) Update Cell

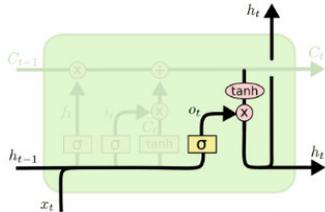


Gambar 7. Update cell

Selanjutnya yaitu memperbarui status sel yang lama ($C_t - 1$) ke dalam status yang baru (C_t). Dengan cara mengkalikan kondisi lama dengan f_t , melupakan hal-hal yang sudah diputuskan untuk dilupakan sebelumnya. Kemudian kita menambahkan $i_t * C_t$ (kandidat baru). Berikut rumus untuk *update* sel :

$$C_t = f_t * C_{t-1} + i_t * C_t \quad (7)$$

4) Alur informasi output gate



Gambar 8. Alur informasi yang melewati output gate

Pada *output gate*, pertama yang dijalankan adalah layer sigmoid yang mana menentukan bagian sel apa yang akan dijadikan *output*. Selanjutnya *cell state* diproses pada tanh (mengubah nilai menjadi antara 1 dan -1) dan dikalikan dengan *output* dari layer sigmoid. Berikut rumus alur informasi *output gate* :

$$o_t = \sigma(W_o \cdot [h_t - 1, x_t] + b_o) \quad (8)$$

$$h_t = o_t * \tanh(C_t) \quad (9)$$

Kriteria MSE (Mean Square Error) digunakan untuk memeriksa kondisi saat berhenti;

$$MSE = 0.5 * \{f(tk1 - tk1)^2 + \dots + (tkm - ykm)^2\} \quad (10)$$

Prosedur pelatihan dilakukan secara berulang sampai mencapai fungsi kinerja (performance) yang minimum atau fungsi kerja < target goal, dimana target goal 0,00001.

F. Penerapan Algoritma RNN-LSTM pada sistem

Tahap pemrosesan data training yang digunakan untuk mendapatkan hasil prediksi sudut matahari sebagai berikut :

```

    • Mengambil dan menormalisasi data
    Start;
    Mengambil data set dengan pandas;
    Memilih kolom data yang digunakan dengan iloc;
    Memilih kolom waktu dan daya sebagai data training;
    Normalisasi data dengan MinMaxScaler pada pers (1);
    Transform data training;
    End;
    • loopback
    Innisialisasi nilai x data training
    Innisialisasi nilai Y data training
    FOR setiap kolom data (loopback, total data)
    X_train Append data [loopback: all data]
    Y_train Append data [loopback: all data]
    Transform data ke array
    
```

```

    END FOR
    • building RNN
    X_train=real_data_usage.iloc[semua baris: daya dan waktu]
    Input: X_train
    Output: predicted_electrical_usage [hourly]
    Initialization : i = 1; epoch=100;loop=0, units = 1
    while loop > epoch do
        while i > 4 do
            for units to 50 step 1 do
                i = i + 1
                LSTM input (pers 2 )
                Input gate (pers 5 dan 6)
                Forget gate (pers 4)
                Update cell (pers 7)
                Output gate (pers 8 dan 9)
                Layer Dense (units = 1)
                Dense layer
                hitung mean_squared_error (pers 10)
            End for
        End while
    End while
    
```

Gambar 9. Pseudocode Sistem RNN – LSTM

III. HASIL DAN PEMBAHASAN

A. Pengujian RNN – LSTM

Melalui pengujian sistem ini akan diketahui sejauh mana akurasi dan hasil yang bisa dilakukan oleh sistem. Sistem yang digunakan yaitu sistem dengan algoritma RNN –LSTM. Hal ini dilakukan agar dapat diketahui perbandingan hasil antara situs *suncalc.org* dan algoritma RNN – LSTM, sehingga dapat diputuskan sistem dapat diimplementasikan sebagai referensi untuk penggerakan aktuator reflektor/solar tracker. Pengujian dilakukan dengan model RNN – LSTM yang sama pada tiga model data yang berbeda, yaitu data mingguan, bulanan dan tahunan untuk hasil data harian, dan data per-jam untuk data harian selama satu minggu.

B. Hasil Model Mingguan

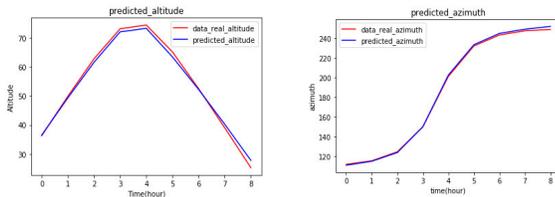
Data *train* yang digunakan adalah data yang telah terkumpul selama 7 hari yang terdiri dari tiga bagian yaitu sudut *Azimuth* pada saat terjadi matahari terbit (*rise*), matahari tenggelam (*sunset*) dan sudut titik puncak (*Culmination*) pada 1 - 7 Januari 2020, serta data test yang digunakan adalah *azimuth – rise – set* dan (*Culmination*) pada 8 Januari 2020. Pada data mingguan akan dilakukan pengujian antara data yang diambil dalam bentuk harian dan per jam (08.00 – 16.00). Hasil dari proses *training* model RNN-LSTM nilai *Azimuth – rise* adalah sebagai berikut :

```

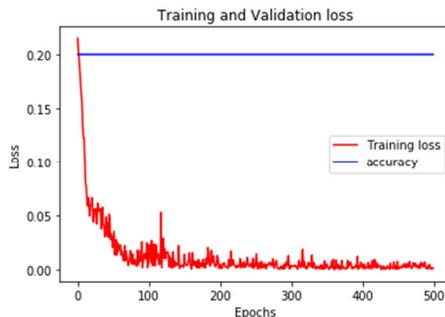
accuracy: 0.2000
Epoch 497/500
5/5 [=====] - 0s 3ms/step - loss:
6.5849e-04 - accuracy: 0.2000
Epoch 498/500
5/5 [=====] - 0s 3ms/step - loss:
7.8842e-04 - accuracy: 0.2000
Epoch 499/500
5/5 [=====] - 0s 3ms/step - loss: 0.0018 -
accuracy: 0.2000
Epoch 500/500
5/5 [=====] - 0s 3ms/step - loss: 0.0012 -
accuracy: 0.2000
    
```

Gambar 10. Nilai *error* pengujian data mingguan nilai *Azimuth – rise* model RNN-LSTM

Dari model prediksi RNN-LSTM yang diterapkan maka dideteksi *mean square error* sebesar 0.0012 dan akurasi yang stabil selama proses training dilakukan yang ditunjukkan pada Gambar 10 pada nilai nilai *Azimuth – rise*. Sedangkan pada nilai *Culmination* juga didapatkan nilai yang *error* yang kecil yaitu sebesar 0.0010 dan juga akurasi yang stabil.



Gambar 11. Grafik prediksi *altitude* dan *azimuth* dengan *output* perjam



Gambar 12. Grafik *loss* dan akurasi model RNN-LSTM model data mingguan nilai nilai *Azimuth – rise*

Error tersebut menunjukkan bahwa proses training berjalan dengan baik karena memiliki nilai *error* yang sangat kecil dan nilai akurasi yang stabil. Idealnya, *error* harus dibawah 1%. Dengan *error* sebanyak 0.12 %, dihasilkan data prediksi seperti pada Tabel 1. Berikut hasil prediksi dengan data test real nilai *azimuth – rise* dan *Culmination* satu hari pada 8 Januari 2020 menggunakan model RNN-LSTM yang dapat dilihat pada Tabel 1 dan Tabel 2, sedangkan data real dan prediksi nilai *Culmination* adalah data dalam bentuk per jam (08.00 – 16.00) dalam seminggu dapat dilihat pada Tabel 3.

Tabel 1. Data real dan data prediksi *azimut - rise – sunsite* dari data mingguan dengan algoritma RNN – LSTM

	DateTime	Azimuth° real	Azimuth° prediksi
rise	08/01/20205:18:18	112.85	113.268
sunset	08/01/20217:00:00	247.36	246.772

Tabel 2. Data real dan data prediksi *culmination* dari data mingguan dengan algoritma RNN – LSTM

DateTime	Culmination° real	Culmination° prediksi
08/01/20205:18:18	75.61	75.0522

Tabel 3. Data real dan data prediksi *Altitude* dari data mingguan dalam bentuk per jam (08.00 – 16.00)

Datetime	Altitude° real	Altitude° prediksi	Azimuth° real	Azimuth° prediksi
08/01/208:00:00	36.18	36.8421	111.94	110.95
08/01/209:00:00	49.82	49.8262	115.47	115.027
08/01/210:00:00	62.79	62.5402	124.63	123.841
08/01/211:00:00	73.27	72.8515	149.98	150.123
08/01/212:00:00	74.66	73.4227	201.23	202.694
08/01/213:00:00	65.36	64.5052	232.24	233.49
08/01/214:00:00	52.68	52.1803	243.37	244.978
08/01/215:00:00	39.12	39.8421	247.66	249.269
08/01/216:00:00	25.3	24.8262	249.04	252.12

C. Hasil Model Bulanan

Data *train* yang digunakan adalah data yang telah terkumpul selama 1 bulan yang terdiri dari tiga bagian yaitu sudut *Azimuth* pada saat terjadi matahari terbit (*rise*), matahari tenggelam (*sunset*) dan sudut titik puncak (*Culmination*) pada 1 - 31 Januari 2020, serta data test yang digunakan adalah *azimuth – rise – set* dan sudut titik puncak (*Culmination*) pada 2 Februari 2020. Pada data bulanan akan di lakukan pengujian antara data yang diambil dalam bentuk harian dan per jam (08.00 – 16.00). Hasil dari proses *training* model RNN-LSTM nilai *Azimuth – rise* adalah sebagai berikut :

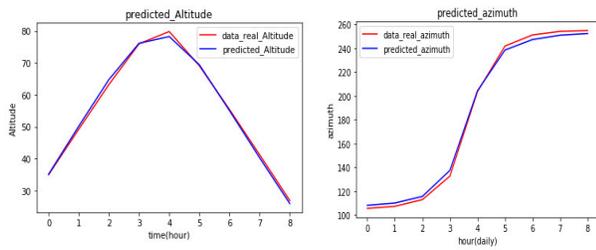
```

- accuracy: 0.0417
Epoch 497/500
24/24 [=====] - 0s 4ms/step - loss: 0.0019
- accuracy: 0.0417
Epoch 498/500
24/24 [=====] - 0s 4ms/step - loss: 0.0014
- accuracy: 0.0417
Epoch 499/500
24/24 [=====] - 0s 4ms/step - loss: 0.0022
- accuracy: 0.0417
Epoch 500/500
24/24 [=====] - 0s 4ms/step - loss: 0.0010
- accuracy: 0.0417
    
```

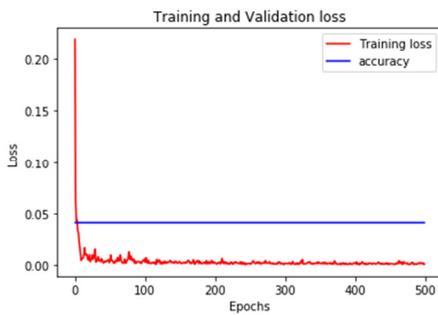
Gambar 13. Nilai *error* pengujian data bulanan nilai *Azimuth – rise* model RNN-LSTM

Dari model prediksi RNN-LSTM yang diterapkan maka dideteksi *mean square error* sebesar 0.0010 dan akurasi yang stabil selama proses training dilakukan yang ditunjukkan pada Gambar 13 pada nilai nilai *Azimuth – rise*. Sedangkan pada

nilai Culmination juga didapatkan nilai yang *error* yang kecil yaitu sebesar 0.0021 dan juga akurasi yang stabil.



Gambar 14. Grafik prediksi *altitude* dan *azimuth* dengan output perjam



Gambar 15. Grafik loss dan akurasi model RNN-LSTM model data bulanan nilai nilai *Azimuth – rise*

Error tersebut menunjukkan bahwa proses training berjalan dengan baik karena memiliki nilai *error* yang lebih kecil dari model data bulanan dan nilai akurasi yang stabil. Idealnya, *error* harus dibawah 1 %. Dengan *error* sebanyak 0.10 % pada nilai *azimuth-rise –sunset* data bulanan, dihasilkan data prediksi seperti pada Tabel 4. Berikut hasil prediksi dengan data real nilai *azimuth – rise- sunset* dan *Culmination* satu hari pada 1 Februari 2020 menggunakan model RNN-LSTM yang dapat dilihat pada Tabel 4 dan Tabel 5, sedangkan data *real* dan prediksi nilai *Culmination* adalah data dalam bentuk per jam (08.00 – 16.00) dalam bulanan dapat dilihat pada Tabel 6.

Tabel 4. Data real dan data prediksi *azimuth – rise- sunset* dari data bulanan dengan algoritma RNN – LSTM

	DateTime	Azimuth ^o real	Azimuth ^o prediksi
rise	01/02/2020	107.64	107.745
sunset	01/02/2020	251.9	246.772

Tabel 5. Data real dan data prediksi *culmination* dari data bulanan dengan algoritma RNN – LSTM

Tanggal	Rise	Date Time	Culminati on ^o real	Culminati on ^o prediksi
01/02/2020	11:43:03	01/02/2020 11:43:03	80.67	80.3442

Tabel 6. Data *real* dan data prediksi *Altitude^o* dan *Azimuth^o* dalam bentuk per jam (08.00 – 16.00) dari data bulanan dengan algoritma RNN – LSTM

Datetime	Altitude ^o real	Altitude ^o prediksi	Azimuth ^o real	Azimuth ^o prediksi
01/02/208:00:00	34.99	35.1496	105.61	107.128
01/02/209:00:00	49.33	50.1864	107.34	115.027
01/02/210:00:00	63.22	64.7698	112.93	117.508
01/02/211:00:00	75.96	76.1282	132.71	140.066
01/02/212:00:00	79.8	78.2235	203.46	205.506
01/02/213:00:00	69.07	69.3939	241.72	238.921
01/02/214:00:00	55.39	55.0328	251.05	247.21
01/02/215:00:00	41.21	40.2681	254.02	250.134
01/02/216:00:00	26.91	25.956	254.68	250.776

D. Hasil Model Tahunan

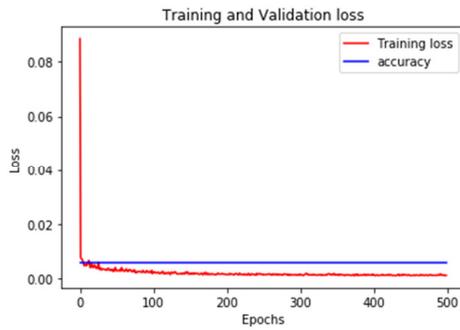
Data yang digunakan adalah data *azimuth – rise- sunset* dan nilai titik puncak (*Culmination*) yang telah terkumpul selama 365 hari pada Januari 2020 sampai 31 Januari 2021. Hasil dari proses training model RNN-LSTM nilai *Azimuth – rise* adalah sebagai berikut.

```

9.3264e-04 - accuracy: 0.0056
Epoch 498/500
359/359 [=====] - 1s 2ms/step - loss:
0.0011 - accuracy: 0.0056
Epoch 499/500
359/359 [=====] - 1s 2ms/step - loss:
0.0011 - accuracy: 0.0056
Epoch 500/500
359/359 [=====] - 1s 2ms/step - loss:
0.0010 - accuracy: 0.0056
    
```

Gambar 16. Nilai *error* pengujian data bulanan nilai *Azimuth – rise* model RNN-LSTM

Dari model prediksi RNN-LSTM yang diterapkan dengan menggunakan *time-step* sebanyak (t-7), dan data ditraining sebanyak satu tahun penuh, serta model prediksi yang sama dideteksi *mean square error* sebesar 0.0054 dan akurasi yang stabil selama proses training dilakukan yang ditunjukkan pada Gambar 16 pada nilai nilai *Azimuth – rise - sunsite*. Sedangkan pada nilai *Culmination* juga didapatkan nilai yang *error* yang kecil yaitu sebesar 0.0084 dan juga akurasi yang stabil.



Gambar 17. Grafik loss dan akurasi model RNN-LSTM model data tahunan nilai nilai Azimuth – rise

Error tersebut menunjukkan bahwa proses training berjalan dengan baik dan nilai akurasi yang stabil. Idealnya, error harus dibawah 1 %. Dengan error sebanyak 0.54 % pada nilai azimuth-rise-sunset data tahunan, dihasilkan data prediksi seperti pada Tabel 7. Berikut hasil prediksi dengan data real nilai azimuth– rise –sunset dan Culmination satu hari pada 1 Januari 2021 menggunakan model RNN-LSTM yang dapat dilihat pada Tabel 7 dan Tabel 8.

Tabel 7. Data real dan data prediksi azimuth - rise dari data bulanan dengan algoritma RNN – LSTM

	DateTime	Azimuth° real	Azimuth° prediksi
rise	01/01/2021	113.4	112.78
sunset	01/01/2021	246.69	245.89

Tabel 8. Data real dan data prediksi culmination dari data bulanan dengan algoritma RNN – LSTM

DateTime	Culmination° real	Culmination° prediksi
01/01/202111:33:08	74.94	74.31

IV. KESIMPULAN

Pembangkit listrik tenaga surya selama ini hanya berfokus pada pencarian sudut matahari yang diperoleh sel surya pada kondisi stagnan tergantung letak matahari yang dinamis dan kemiringan aktuator, tanpa mempertimbangkan rugi daya yang digunakan untuk menggerakkan reflektor itu sendiri.

Oleh karenanya dibutuhkan optimasi sistem yang ditujukan untuk memastikan sudut kemiringan yang sesuai pada aktuator, dan meminimalisir terjadinya pemborosan daya dengan memaksimalkan intensitas cahaya matahari yang diperoleh sel surya menggunakan sistem identifikasinya yang dilakukan secara langsung. Pada penelitian ini difokuskan pada model prediksi yang tepat untuk mendapatkan prediksi prediksi posisi matahari yang dimana data tersebut menjadi

acuan untuk referensi sudut optimal sel surya. Pada pendekatan ini, RNN – LSTM dibangun empat layer LSTM dimana masing – masing hidden layer terdapat 50 memory units. Pada setiap layer LSTM terdapat layer dropout yang disetting sebanyak 20% untuk mencegah terjadinya overfitting. Jadi pada sistem ini terdapat 10 units memory yang diabaikan, dan satu unit output yang sepenuhnya dengan fungsi aktivasi linear. Optimizer yang digunakan adalah adam optimizer dengan konfigurasi default dan perhitungan error menggunakan mean square error.

Pengujian dilakukan pada tiga model data yaitu data mingguan, data bulanan dan data tahunan. Pada data mingguan dilakukan dua percobaan pada data yang diambil harian dan per jam (08.00 – 16.00). Data mingguan menggunakan data sebanyak 7 hari, didapatkan hasil error sebesar 0.12 %. Error ini menunjukkan bahwa hasil prediksi sudah sesuai dengan data real. Pada data bulanan menggunakan data sebanyak 31 data didapatkan hasil error sebesar 0.10 % dan pada data tahunan (365 data) didapatkan error 0.10 % Dapat dilihat dari perbandingan error, dengan menggunakan model yang sama dan di uji dengan tiga model data yang berbeda mendapatkan hasil yang berbeda pula. Tetapi semua error berada pada presentase dibawah satu persen atau sudah sesuai dengan model sistem / prediksi yang diharapkan (pada situs suncalc.org).

Dapat dikemukakan keterkaitan secara teoritis dari hasil penelitian tersebut yaitu pemilihan metode dan model algoritma yang tepat sangat berpengaruh terhadap keakuratan hasil prediksi. Model algoritma yang dibangun menghasilkan error yang sangat kecil yaitu 0.10 %. Hal ini menunjukkan jika model berjalan sesuai dengan harapan. Dengan demikian model ini dapat digunakan untuk pengendalian pemborosan daya pada reflektor panel surya.

V. DAFTAR PUSTAKA

- [1] Nursalam, 2016, metode penelitian. (2013). SOLAR TRACKING DUAL – AXIS BERBASIS ARDUINO UNO DENGAN MENGGUNAKAN LENS FRESNEL GUNA MENINGKATKAN EFISIENSI PENGFOKUSAN CAHAYA MATAHARI. Journal of Chemical Information and Modeling, 53(9), 1689–1699. <https://doi.org/10.1017/CBO9781107415324.004>
- [2] S. S. Yatmani, “Sistem kendali Solar Tracker Untuk Meningkatkan efisiensi Daya,” J. Tek. Mesin ITI, vol. 4, no. 1, p. 1, 2020, doi: 10.31543/jtm.v4i1.354..
- [3] M. Nur Qomaruddin and M. Khairi, “Real Time Clock Sebagai Tracking Sinar Matahari Pada Solar Cell Berbasis Mikrokontroler Untuk Lampu Taman (Real Watch Tracking As A Sun Ray On Microcontroller Based Solar Cells For Park Lights),” JEEE-U (Journal

-
- Electr. Electron. Eng., vol. 3, no. 2, p. 305, 2019, doi: 10.21070/jeee-u.v3i2.2547.
- [4] Saputra, Rahmad Aryangga (2018) ANALISIS POSISI MATAHARI SEBAGAI REFERENSI PERGERAKAN REFLEKTOR PANEL SURYA DALAM UPAYA OPTIMASI DAYA KELUARAN. Undergraduate (S1) thesis, University of Muhammadiyah Malang.
- [5] Shalih, Y., & Keluaran, D. (n.d.). Pengaruh Arah Posisi Pemasangan Panel Surya Terhadap Putput Daya Keluaran.
- [6] Chong, K.-K., & Wong, C.-W. (2010). General Formula for On-Axis Sun-Tracking System. *Solar Collectors and Panels, Theory and Applications*, May 2014. <https://doi.org/10.5772/10341>
- [7] Fauziah, Lailatul (2018) ACCURATE POSITIONING CONTROL OF SOLAR PANEL SYSTEM FOR AZIMUTH AND ELEVATION TRACKING USING SMC BASED SUN POSITION IMAGE. Undergraduate (S1) thesis, University of Muhammadiyah Malang.
- [8] Khasanah, Hidayatul (2020) DESIGN SMART MONITORING ENERGI LISTRIK BERBASIS PREDICTIVE ANALYTICS MENGGUNAKAN ALOGARITMA RECURRENT NEURAL NETWORK (RNN). Undergraduate (S1) thesis, University of Muhammadiyah Malang.
- [9] N. Soedjarwanto, "Sistem Pelacak Otomatis Energi Surya Berbasis Mikrokontroler ATMega8535," *J. ELTEK Polinema*, pp. 11–20, 2015.
- [10] Hartati, Sri dan Ahmad Ashril Rizal (2017, 1 April) Prediksi Kunjungan Wisatawan dengan Recurrent Neural Network Extended Kalman Filter. *Jurnal Ilmiah ILMU KOMPUTER Universitas Udayana Vol. X*, 1, 7-18
- [11] K. Ivanedra and M. Mustikasari, "Implementasi Metode Recurrent Neural Network Pada Text Summarization Dengan Teknik Abstraktif the Implementation of Text Summarization With Abstractive Techniques Using Recurrent Neural Network Method," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 6, no. 4, pp. 377–382, 2019, doi: 10.25126/jtiik.201961067.