

Otomatisasi Pengoperasian Alat Elektronik Berdasarkan Hasil Prediksi Algoritma *Long Short Term Memory*

Afriansyah, Ade Irawan

Ilmu Komputer, Universitas Pertamina Jalan Teuku Nyak Arief, Simprug, Kebayoran Lama, Jakarta 12220, Indonesia

ARTICLE INFORMATION

Received: July 21st, 2020
 Revised: September 8th, 2020
 Available online: September 30th, 2020

KEYWORDS

power consumption, electricity usage, time series data, LSTM, internet of things

CORRESPONDENCE

Phone: +62 821 2552 7597
 E-mail: afriunderscore@gmail.com

A B S T R A C T

Excessive use of household electricity is one of the causes of the largest amount of national electricity consumption coming from households. One way to reduce the amount of household electricity consumption is to automate the operation of electronic devices. This research proposes utilizing Long Short Term Memory (LSTM) algorithm to predict the habit of operating an electronic device. The prediction is then applied to automate the operation of that by exploiting the time series data from the usage. A series of experiments are conducted to capture the data of operating a manual lamp. Then, an LSTM model is built by training the data. The experiment results show the prediction accuracy of 99,28% and Root Mean Square Error of 0.091. Furthermore, the LSTM model is used to automatically operate a lamp in a month. The electricity cost from the automation is 36,38% lower than the manual.

PENDAHULUAN

Masyarakat Indonesia termasuk konsumen yang sangat boros dalam penggunaan energi listrik. Berdasarkan data penjualan energi listrik tahun 2018 dari PLN, konsumsi listrik nasional terbesar berasal dari rumah tangga yaitu 97.832,28 GWh (41,70%) [2]. Banyak penelitian telah berlangsung selama lebih dari satu dekade hingga saat ini untuk meningkatkan efisiensi daya pada tingkat konsumen. Salah satunya adalah pengembangan sistem rumah pintar (*smart home*).

Smart home adalah istilah yang umum digunakan untuk mendefinisikan tempat tinggal yang mengintegrasikan teknologi dan layanan melalui jaringan rumah seperti mengendalikan alat elektronik dari jauh untuk meningkatkan efisiensi daya listrik. Sistem *smart home* terbentuk dari beberapa perangkat sehari-hari, seperti telepon cerdas, peralatan elektronik, sensor, dan aktuator yang terhubung dengan internet, yang umum disebut *Internet of Things* (IoT). Teknologi IoT banyak digunakan untuk membuat konsep baru serta pengembangan sistem *smart home* untuk memberikan kecerdasan, kemudahan dan peningkatan taraf hidup [1]. Salah satu keunggulan dari pengembangan sistem ini adalah adanya kemampuan untuk memantau dan mengendalikan penggunaan listrik. Informasi konsumsi daya listrik pada peralatan elektronik ini sangat penting karena banyak pengguna

yang ingin mengetahui gambaran yang jelas tentang konsumsi daya aktual secara *real time* [3].

Peneliti di [3] mengembangkan sebuah sistem *smart home* yang dapat menunjukkan informasi konsumsi daya listrik pada segala jenis peralatan elektronik dan melihat status *on/off* serta dapat dikendalikan melalui telepon cerdas atau web. Pada penelitian lainnya di [4], suatu sistem *smart home* dikembangkan dengan kemampuan untuk memantau serta mengendalikan peralatan elektronik melalui suatu aplikasi berbasis Android. Namun, sistem tersebut masih memerlukan bantuan manusia untuk memantau pemakaian peralatan elektronik. Ditambah lagi, waktu pengoperasian alat elektronik tidak selalu sama setiap harinya. Oleh karena itu, sistem ini memerlukan kemampuan untuk pengoperasian alat elektronik berdasarkan kebiasaan yang dilakukan oleh manusia serta hasil penginderaan lingkungannya.

Penelitian ini mengusulkan pemanfaatan salah satu algoritma *machine learning*, yaitu *Long Short Term Memory* (LSTM) untuk otomatisasi pengoperasian peralatan elektronik pada sistem *smart home* yang dikembangkan di Universitas Pertamina, bernama NeuronThings. LSTM memprediksi pola pemakaian listrik dengan memanfaatkan informasi data deret waktu penggunaan alat elektronik yang diperoleh dari NeuronThings. Hasil prediksi LSTM digunakan untuk mengotomatiskan hidup dan matinya alat elektronik yang terbatas pada pemakaian lampu pada waktu

tertentu sehingga dapat membantu konsumen menghemat penggunaan listrik.

Data Deret Waktu

Data deret waktu $x_t, t = \{0, 1, 2, \dots\}$ merupakan suatu rangkaian data yang diukur pada waktu t yang berurutan. Data deret waktu yang berisi satu variabel disebut *univariate* dan yang lebih dari satu variabel disebut *multivariate*, keduanya dapat berupa kontinu atau diskrit [5]. Pengukuran data deret waktu dilakukan secara berurutan pada suatu kejadian. Data deret waktu kontinu diukur pada setiap langkah waktu yang terus bertambah, sedangkan data deret waktu diskrit diukur berdasarkan waktu yang sama pada suatu interval, misalnya pada selang waktu yang telah ditentukan seperti harian, bulanan, dan tahunan. Variabel yang diamati dalam data deret waktu diskrit dapat diukur berdasarkan variabel kontinu, yaitu dengan mengubah rangkaian waktu kontinu dengan menggabungkan data selama selang waktu tertentu. Data yang digunakan untuk memprediksi pengoperasian lampu otomatis pada penelitian ini merupakan data deret waktu *multivariate* kontinu. Sedangkan data konsumsi daya lampu per bulan merupakan data *univariate* diskrit.

LSTM

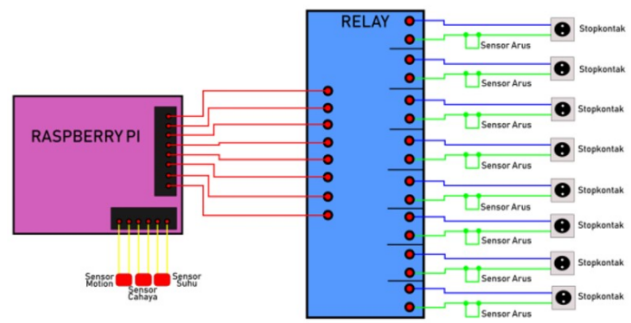
LSTM merupakan turunan dari *Recurrent Neural Network* (RNN) untuk mengatasi ketidakmampuan RNN untuk mengolah informasi *sequential* dalam jangka panjang terutama pengolahan data berupa deret waktu [6]. LSTM memiliki struktur kompleks yang disebut unit LSTM dalam *hidden layer* yang berisi tiga gerbang (*gate*) yaitu *input gate*, *forget gate* dan *output gate* agar ia mampu selektif dalam menambah atau menyimpan informasi ke *cell state* yang berfungsi untuk menyimpan sebuah nilai atau keadaan.

NeuronThings

NeuronThings adalah sistem rumah pintar berbasis IoT yang dikembangkan di Universitas Pertamina, yang memungkinkan pengguna untuk mengoperasikan peralatan elektronik menggunakan aplikasi telepon cerdas untuk mengontrol dan memantau peralatan elektronik di rumah. Sistem ini memanfaatkan Raspberry Pi yang berfungsi sebagai komputer mini yang memiliki sistem operasi khusus untuk menjalankan berbagai jenis perangkat lunak sehingga dapat mengubah peralatan elektronik menjadi berbasis IoT dengan menyambungkannya ke sistem.

NeuronThings memiliki beberapa sensor, diantaranya sensor arus, suhu, kelembaban, gerakan, dan intensitas cahaya. Hasil penginderaan sensor-sensor tersebut disimpan dalam bentuk data deret waktu. Semua sensor tersebut terpasang pada perangkat NT-Device. Desain dari perangkat ini dapat dilihat pada Gambar 1. Spesifikasi dari NT-Device dapat dilihat pada Tabel 1.

NT-Device dapat diletakkan di setiap ruangan yang berbeda untuk dapat mengontrol peralatan elektronik di rumah. Pengguna dapat melihat total penggunaan dan perkiraan biaya listrik dari aplikasi di ponsel cerdas, berdasarkan data yang didapat dari perangkat tersebut. Penggunaan listrik dapat dilihat dengan memilih NT-Device yang terdapat di suatu ruangan melalui aplikasi. Pengguna kemudian dapat mengontrol peralatan elektro-



Gambar 1. Desain Perangkat NT-Device

Tabel 1. Spesifikasi Perangkat NeuronThings

Nama Perangkat	Konsumsi Daya (watt)
Raspberry Pi 4 Model B 2GB	3.5
Sensor Arus ACS712 30A	< 0.1
Motion Sensor Infrared PIR HC-SR505	< 0.1
Sensor Cahaya BH1750 GY-302	< 0.1
DHT22 AM2302	< 0.1
Modul Relay 4 Channel 5V	< 0.1
Arduino UNO R3 Atmega328	< 0.1
Estimasi total	3.6

nik, seperti menghidupkan dan mematikan peralatan dari jarak jauh dengan aplikasi tersebut. Pengguna juga dapat mengatur jangka waktu penggunaan perangkat melalui aplikasi yang sama.

METODE PENELITIAN

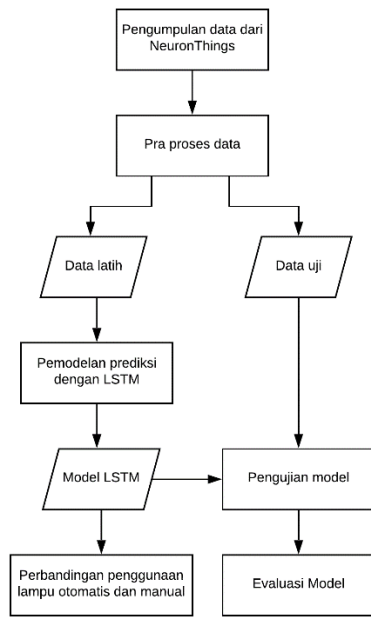
Fokus penggunaan NeuronThings pada penelitian ini adalah pada aspek pemanfaatan algoritma LSTM. Suatu NT-Device menerima nilai hasil tangkapan dari sensor-sensor yang terpasang, lalu mengirim data tersebut ke basis data *cloud* yang diperbarui setiap 60 detik sekali. Data tersebut kemudian diproses dengan algoritma LSTM untuk memprediksi waktu hidup atau matinya alat elektronik yang terhubung dengan NT-Device tersebut. Hasil prediksi kemudian dikirim ke pengguna sehingga dapat dilihat pada aplikasi di ponsel cerdasnya. Hasil prediksi tersebut juga menjadi perintah *on/off* ke *relay* pada NT-Device.

Metodologi Penelitian

Penelitian ini dilakukan dengan melakukan eksperimen pembuatan model LSTM berdasarkan data deret waktu yang diperoleh dari perilaku penggunaan lampu yang terhubung dengan NT-Device. Data perilaku penggunaan tersebut digunakan untuk membentuk model LSTM yang hasil prediksinya digunakan untuk pengoperasian lampu secara otomatis. Pengoperasian lampu secara otomatis akan dibandingkan dengan pengoperasian lampu manual pada tahap akhir dari penelitian ini. Tahapan yang dilakukan pada penelitian ini dapat dilihat pada Gambar 2.

Pengumpulan Data dari NeuronThings

Penelitian ini menggunakan data deret waktu dari perilaku penggunaan lampu yang terhubung dengan NT-Device. Untuk mendapatkan data tersebut, pengumpulan data dilakukan selama satu bulan dengan cara menggunakan lampu yang dikontrol secara manual melalui aplikasi, yang kemudian disimpan ke basis data *cloud*.



Gambar 2. Tahapan Penelitian.

Lampu ini digunakan di ruang kerja dengan pengoperasian yang memperhatikan efisiensi dan penghematan listrik. Data deret waktu selama satu bulan, yang terdiri dari berbagai data sensor dan keadaan harian serta sudah mencakup perilaku penggunaan pada hari kerja dan akhir pekan, direkam dengan satuan menit. Data tersebut digunakan sebagai data latih untuk model LSTM yang diusulkan.

Pra Proses Data

Data deret waktu yang telah terkumpul selama satu bulan tersebut kemudian diproses dan disesuaikan terlebih dahulu untuk dapat diterima model LSTM. 13 atribut dari data perilaku penggunaan lampu diperlukan di *backend* sistem NeuronThings untuk mendukung alur informasi dari *backend* ke *frontend* atau sebaliknya, yaitu *status*, *temperature*, *humidity*, *light_intensity*, *is_motion*, *days*, *weather*, *time_minutes*, *id*, *device_id*, *switch_id*, *created_at*, dan *updated_at*. Dari 13 atribut tersebut, hanya 8 atribut pertama yang digunakan sebagai informasi yang dipelajari saat proses pelatihan LSTM. Sedangkan 5 atribut lainnya: *id*, *device_id*, *switch_id*, *created_at*, dan *updated_at* digunakan untuk keperluan di aplikasi dan tidak membantu dalam algoritma LSTM.

Pra pemrosesan data dilakukan dalam beberapa tahap, yaitu seleksi atribut, persiapan data, dan pembagian data. Status menandakan keadaan lampu saat hidup atau mati yang dijadikan sebagai label prediksi atau *output*.

1. Seleksi atribut.

Pemilihan atribut yang dibutuhkan dari seluruh data perilaku penggunaan lampu dilakukan untuk membuat model prediksi. Data mentah pada Tabel 2 di-load kemudian diubah ke format *dataframe*. Informasi atribut *created_at* digunakan sebagai indeks yang kemudian disimpan dengan format *dataframe* untuk dapat digunakan pada pembuatan model LSTM.

2. Persiapan data.

Persiapan data untuk model LSTM melibatkan normalisasi

Tabel 2. Atribut perilaku penggunaan lampu

Atribut	Keterangan
id	Menandakan indeks
device_id	Menandakan alat elektronik yang digunakan
switch_id	Menandakan <i>switch/relay</i> yang digunakan
humidity	Menandakan kelembaban
temperature	Menandakan temperatur
is_motion	Menandakan gerakan
light_intensity	Menandakan intensitas cahaya
status	Menandakan status perangkat <i>on/off</i>
days	Menandakan hari (senin-jumat dalam format 1-7)
weather	Menandakan keadaan cuaca
time_minutes	Menandakan waktu dalam satuan menit
created_at	Menandakan kapan dibuat
updated_at	Menandakan kapan diperbarui

data dan *framing dataset*. Seluruh data dinormalisasi untuk mempercepat proses komputasi dan menghindari nilai yang memiliki rentang nilai besar lebih banyak daripada nilai yang memiliki rentang nilai kecil. Data perilaku penggunaan lampu dinormalisasi Min Max sehingga seluruh atribut memiliki nilai dengan interval 0-1 [7]. Data yang ternormalisasi, $\tilde{x}_{r,t}$ dengan atribut r pada waktu t , didefinisikan sebagai:

$$\tilde{x}_{r,t} = \frac{x_{r,t} - \min(X_r)}{\max(X_r) - \min(X_r)}, \quad (1)$$

dengan $x_{r,t}$ adalah nilai yang dinormalisasi, X_r adalah himpunan seluruh nilai atribut r dari data pelatihan, $\min(X_r)$ adalah fungsi yang menghasilkan nilai minimum dari X_r , dan $\max(X_r)$ fungsi yang menghasilkan nilai maksimum dari X_r . Atribut-atribut yang bisa dinormalisasi adalah *humidity*, *light_intensity*, dan *time_minutes* karena memiliki nilai yang jauh lebih besar dibandingkan atribut lainnya. Maka, seluruh variabel input dan *output* dinormalisasi untuk menghindari rentang nilai tersebut.

Framing dataset dilakukan untuk membuat data menjadi *supervised learning problem* yang disebut *supervisedFrame*. Data tersebut diubah menjadi sampel input dan *output* untuk proses pelatihan model, sehingga suatu algoritma dapat belajar bagaimana memprediksi pola *output* dari pola input [8]. *supervisedFrame* digunakan pada proses pelatihan untuk memprediksi status pada waktu saat ini (t) dengan mengingat data status dan atribut yang digunakan pada langkah waktu sebelumnya ($t - 1$) dalam satuan menit.

3. Pembagian data.

supervisedFrame dibagi menjadi data latih dan data uji. Data pada 3 minggu pertama digunakan sebagai data latih dan data pada 1 minggu terakhir digunakan sebagai data uji. Data latih dan data uji masing-masing memiliki data deret waktu sepanjang T_{latih} dan T_{uji} . Data latih digunakan untuk pelatihan model LSTM dan data uji digunakan untuk mengetahui seberapa baik model LSTM dapat memprediksi berdasarkan hasil pelatihan. Model LSTM yang telah dibuat kemudian digunakan untuk memprediksi dengan cara *generate* status untuk mengotomatiskan penggunaan lampu pada bulan berikutnya.

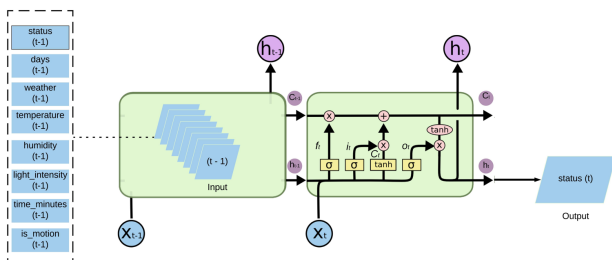
Prediksi dengan LSTM

Penelitian ini menggunakan *framework* Keras dengan *backend* Tensorflow untuk membuat model LSTM dan melakukan prediksi. Parameter input tensor 3D yang diperlukan adalah berupa sampel, langkah waktu, dan fitur. Variabel input diubah ke format 3D menjadi [23.961, 1, 8] serta data uji menjadi [6.102, 1, 8]. Dengan demikian, terdapat 8 variabel input yang digunakan untuk memprediksi sehingga diperoleh *output* status pada langkah waktu t . Desain LSTM pada penelitian ini dapat dilihat pada Tabel 3. Pada penelitian ini, arsitektur dari LSTM dapat dilihat pada Gambar 3. Tahapan yang dilakukan pada proses pemodelan prediksi dengan LSTM yaitu:

1. Pendefinisian parameter-parameter dari model LSTM dengan 100 neuron pada *hidden layer* LSTM dan 1 *output layer*. supervisedFrame diubah ke format 3D dengan input satu langkah waktu per menit sebelumnya menggunakan variabel input dan variabel *output* berupa status pada menit saat ini.
2. Model di-*compile* menggunakan Adam *optimizer* dan *loss function* MSE
3. Model di-*fit* dengan 2000 *epochs* dan ukuran *batch* 70.

Tabel 3. Desain LSTM

Jenis	Perincian
Arsitektur	1 LSTM layer 1 output layer
Fungsi aktivasi	sigmoid tanh
Optimizer	Adam
Loss function	MSE



Gambar 3. Arsitektur LSTM untuk pemodelan prediksi penggunaan lampu secara otomatis.

Luaran dari proses pelatihan ini adalah berupa model LSTM yang digunakan untuk memprediksi pengoperasian alat elektronik secara otomatis. Model disimpan dalam format .h5py untuk digunakan pada perangkat NeuronThings.

Pengujian dan Evaluasi Model

Setelah model di-*fit*, prediksi kemudian dilakukan menggunakan data uji untuk menghasilkan luaran yaitu status pada langkah waktu saat t . Nilai aktual dalam skala aslinya dibandingkan dengan nilai hasil prediksi sehingga diperoleh skor kesalahan untuk model. *Root Mean Square Error* (RMSE) digunakan sebagai metrik penilaian untuk mengukur kinerja model. RMSE adalah akar dari rata-rata perbedaan nilai kuadrat antara data aktual dan hasil prediksi atau dapat diekspresikan secara matematis sebagai berikut:

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (x_t - h_t)^2}{n}} \quad (2)$$

dengan x_t adalah nilai aktual saat t , h_t adalah nilai prediksi saat t , dan n adalah jumlah data yang diprediksi.

Perbandingan Penggunaan Lampu Otomatis dan Manual

Penelitian ini menggunakan dua buah lampu yang terhubung dengan sistem NeuronThings, yaitu: lampu A yang dioperasikan secara otomatis, dan lampu B yang dioperasikan manual. Model prediksi LSTM diimplementasikan pada NT-Device untuk pengoperasian lampu A. Mekanisme dari implementasi model yaitu:

- a. Model di-*load* pada perangkat NeuronThings untuk pengoperasian lampu A secara otomatis berdasarkan data harian.
- b. Data perilaku penggunaan lampu A selama satu hari dikumpulkan pada perangkat NeuronThings sebagai input model LSTM untuk memprediksi status pada hari berikutnya.
- c. Hasil prediksi dikirim ke basis data *cloud* per menit sebanyak 1440 kali dengan data satu hari dalam satuan menit yaitu 1440 menit sebagai perintah untuk pengoperasian lampu A.
- d. Proses pada poin b dan c dilakukan berulang hingga satu bulan.

Penggunaan lampu pengoperasian otomatis tersebut kemudian dibandingkan dengan penggunaan lampu pengoperasian manual. Biaya listrik penggunaan kedua lampu dihitung berdasarkan persamaan (3) di Lampiran.

HASIL DAN PEMBAHASAN

Pra Proses Data

Data perilaku penggunaan lampu diambil dari basis data *cloud* sistem NeuronThings yang terkumpul selama satu bulan. Data dalam format .json tersebut diubah ke format .csv untuk dapat dilakukan pra proses data. Saat melakukan percobaan, data yang diperoleh adalah sebanyak 23.503 dari yang semestinya 40.320. Hal ini dikarenakan adanya permasalahan pada koneksi internet yang tidak stabil sehingga data tidak terekam seluruhnya. Kekurangan data tersebut diisi berdasarkan pertimbangan keadaan lampu hidup saat sebelum dan sesudah hilangnya informasi dengan perbedaan nilai yang tidak signifikan antar keduanya sehingga diperoleh sebanyak 30.064. Misalkan data yang hilang berada pada interval waktu $t = j$ hingga $t = k$, maka pengisian data yang hilang ke dataset dilakukan dengan menyalin seluruh atribut x_{j-1} atau x_{k+1} dengan toleransi perbedaan nilai yang menunjukkan tidak signifikan berdasarkan atribut pada Tabel 4.

Tabel 4. Toleransi perbedaan nilai

Atribut x	Toleransi perbedaan
temperature	<2
humidity	<25
light_intensity	<20

```

1  Prosedur pengisian data yang hilang
2  Input: deret Xa, daftar toleransi: Tol(Xt, Xs)
3  t = 1
4  while t <= Tpelatihan:
5    j <- X_(time_minutes,t)
6    k <- X_(time_minutes,t+1)
7    if (k - j) > 1:
8      if Xt in Tol(Xt, Xs): // mengecek nilai dari seluruh atribut daftar toleransi
9        A <- Xt or Xs
10       B <- A * (k-j)
11       t+ <- k
12       insert(A, t, t+1, B)
13     else:
14       A <- find(Xt, j) // mencari nilai yang serupa berdasarkan
15       B <- A * (k-j) // seluruh atribut daftar toleransi
16       t+ <- j
17       insert(A, t, t+1, B)
18     t <- t+1
19   else:
20     t <- t+1
21
22

```

Gambar 4. Algoritma Pengisian Data yang Hilang

Strategi pengisian untuk data yang kosong dijelaskan pada algoritma yang terdapat pada Gambar 4. Data yang hilang dinyatakan dengan tidak adanya data di antara dua data yang berurutan tetapi memiliki selisih menitnya lebih dari 1 seperti yang diterangkan oleh variabel j dan k pada baris 5 hingga 7. Prosedur *Tol* pada baris ke-8 menerangkan bahwa data sebelum hilangnya informasi atau X saat t dan data setelah hilangnya informasi atau X saat s diperiksa selisih nilai keduanya berdasarkan pada Tabel 4, sehingga diperoleh nilai *true* yang menunjukkan nilai tersebut ditoleransi atau *false* untuk sebaliknya. Prosedur *insert* pada baris ke-12 dan ke-18 menerangkan bahwa seluruh atribut data X saat t akan disisipkan sebagai data pada menit yang hilang di antara menit saat j dan menit saat k , dengan toleransi perbedaan nilai atribut antara keduanya berdasarkan Tabel 4. Fungsi *find* pada baris ke-14 menerangkan bahwa data atribut X saat t berdasarkan Tabel 4 dengan nilai yang serupa dengan atribut X saat j diambil untuk disisipkan sebagai data pada menit yang hilang di antara menit saat j dan menit saat k .

Data yang diamati dalam penelitian ini, dari sisi perilaku penggunaan lampu, adalah berupa data sensor dan keadaan harian. Maka atribut yang digunakan yaitu status, *temperature*, *humidity*, *light_intensity*, *is_motion* yang diperoleh dari sensor dan *days*, *weather*, *time_minutes* yang diperoleh dari keadaan harian. Data tersebut digunakan sebagai informasi yang dipelajari saat proses pelatihan LSTM dan status *on/off* lampu yang menjadi target prediksi dapat dipengaruhi oleh perilaku penggunaan seperti kondisi di siang hari yang cerah mendapat pencahayaan dari luar, menit yang menunjukkan waktu, keadaan ruangan seperti temperatur dan kelembapan cenderung mengikuti keadaan cuaca, serta terdeteksinya gerakan menandakan adanya seseorang sehingga berdasarkan perilaku tersebut saat itu lampu sedang dimatikan.

Hasil dari pra proses data setelah dinormalisasi dan di-*frame* ke bentuk supervisedFrame dapat dilihat pada Tabel 5 dan 6. Terdapat 30.064 data supervisedFrame yang dibagi menjadi data latih dan data uji. Rasio yang digunakan untuk pelatihan dan pengujian adalah 80:20. Dengan demikian, diperoleh 23.961 data latih dan 6.102 data uji. Kemudian pemisahan variabel input dan *output* dilakukan pada masing-masing data latih dan data uji. Variabel input terdiri dari data dengan lag ke-1 ($t - 1$) dan variabel output yaitu status (t).

Tabel 5. Sampel data supervisedFrame

Status ($t - 1$)	Days ($t - 1$)	Weather ($t - 1$)	Temperature ($t - 1$)	Humidity ($t - 1$)
1	0.16667	1	0.634812	0.808097

0	0.16667	1	0.679181	0.740081
0	0.16667	1	0.679181	0.74251
1	0.16667	1	0.680034	0.753138
0	0.16667	1	0.685079	0.737652

Tabel 6. Sampel data supervisedFrame (lanjutan)

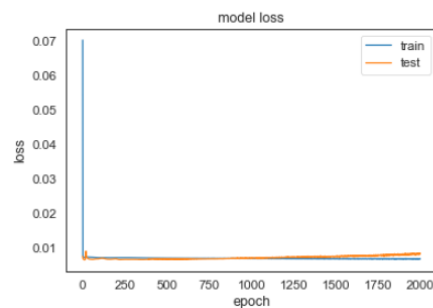
light_intensity ($t - 1$)	time_minutes ($t - 1$)	is_motion ($t - 1$)	Status (t)
0.109005	0.392634	0	0
0.151659	0.397498	0	0
0.151659	0.397498	0	1
0.122927	0.398193	1	0
0.151766	0.398193	1	0

Prediksi dengan LSTM

Hasil Pelatihan

Model LSTM yang dihasilkan dari proses pelatihan ini disimpan dengan format *file* .h5py bernama 'LSTM_final_model.h5py' untuk digunakan pada NT-Device. Hasil pelatihan dengan *lag* ke-1 menghasilkan akurasi sebesar 99,28%. Model tersebut sudah mampu mengingat informasi yang diberikan berdasarkan data dengan x_{t-1} . Grafik model loss selama proses pelatihan dengan 2.000 epochs dapat dilihat pada Gambar 5.

Epoch 2000/2000
- 1s - loss: 0.0069 - acc: 0.9928 - val_loss: 0.0083 - val_acc: 0.9934



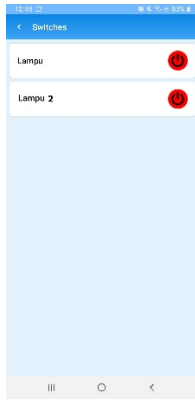
Gambar 5. Model loss

Pengujian dan Evaluasi Model

Pengujian model LSTM dilakukan dengan memprediksi status pada langkah waktu t menggunakan data uji. Karena adanya proses normalisasi data di awal, nilai hasil prediksi kemudian dikembalikan ke nilai aslinya. Setelah proses pengujian, model dievaluasi menggunakan RMSE. Nilai RMSE yang dihasilkan dari hasil prediksi dan aktual adalah 0,091. Nilai ini cenderung kecil karena perbedaan nilai hasil prediksi dan aktual juga memiliki rentang nilai kecil. Nilai tersebut menunjukkan dekatnya nilai hasil prediksi dan aktual. Berdasarkan Gambar 5, model LSTM sudah baik dalam memprediksi status menggunakan input pada $t - 1$ dengan akurasi sebesar 99,28% dan RMSE sebesar 0,091, sehingga dapat digunakan untuk NT-Device.

Perbandingan Penggunaan Lampu Otomatis dan Manual

Terdapat dua lampu yang dihubungkan pada sistem Neuronthings, yaitu lampu A yang dioperasikan secara otomatis dari hasil prediksi model LSTM dan lampu B yang dioperasikan secara manual melalui fitur *Device* pada aplikasi NeuronThings. Tangkapan layar fitur *Device* dapat dilihat pada Gambar 6.



Gambar 6. Tangkapan layar fitur *Device*.

Kedua lampu tersebut memiliki spesifikasi yang sama yang dapat dilihat pada Tabel 7.

Tabel 7. Spesifikasi lampu

Spesifikasi lampu	
Input voltage	AC 150-240/50-60Hz
Luminous efficacy	100 Lm/W
Total Power	3W
CRI	>80
Working life	25000 hrs

Lampu diletakan di bagian ruang kerja sebagai penerangan untuk kegiatan rutin sehari-hari selama satu bulan. Adapun kegiatan yang dilakukan yaitu beraktivitas menggunakan komputer desktop. Lampu dihidupkan hanya pada saat dibutuhkan penerangan saat melakukan kegiatan karena ruang tersebut mendapat pencahayaan dari luar khususnya di siang hari. Kondisi lampu dihidupkan sehari-hari berada pada waktu fajar dan sore hari menjelang matahari terbenam hingga malam sebelum tidur. Tempat lampu diletakan dapat dilihat pada Gambar 7.



Gambar 7. Penempatan lampu.

Pengimplementasian Model

File 'LSTM_final_model.h5py' di-load pada perangkat NeuronThings untuk memprediksi status berdasarkan data perilaku penggunaan lampu A pada kegiatan rutin sehari-hari. Data deret waktu per menit direkam pada perangkat NeuronThings pada hari pertama sebanyak 1440 pada perangkat untuk digunakan sebagai input ke dalam model untuk memprediksi status *on/off* lampu A pada hari berikutnya, kemudian siklus tersebut berlangsung selama satu bulan. Hasil prediksi digunakan sebagai perintah ke sistem NeuronThings

untuk otomatisasi pengoperasian lampu A. Nilai hasil prediksi bertipe *float* dibulatkan sehingga diperoleh nilai *boolean* yang menandakan *on/off*.

Hasil prediksi LSTM dapat mengotomatiskan pemakaian lampu A dengan sangat baik. Dari pengamatan selama satu bulan lampu A dapat beroperasi tanpa diperintahkan oleh pengguna melalui aplikasi NeuronThings berdasarkan data perilaku penggunaan lampu sehari-hari. Lampu B tidak luput dari kelalaian dalam pemakaian, seperti lupa dimatikan saat tidak digunakan, sedangkan lampu A dapat mati secara otomatis jika tidak digunakan.

Perhitungan Biaya Listrik

Biaya listrik dihitung secara manual dari data status lampu A dan lampu B selama satu bulan. Kedua data tersebut memiliki nilai *boolean* 0 atau 1. Maka masing-masing data tersebut dijumlahkan sehingga diperoleh 10.329 menit masa operasi lampu A, dan 16.235 menit lampu B. Kemudian, hasil penjumlahan tersebut dibagi 60 untuk mendapatkan data status dalam satuan jam. Golongan tarif listrik yang digunakan berada pada 1.300 VA hingga 6.600 VA sehingga besarnya biaya yang dibayarkan yaitu Rp 1467,28 per kWh. Lampu yang digunakan membutuhkan daya sebesar 3 Watt untuk beroperasi. Biaya listrik dihitung menggunakan persamaan (3), sehingga diperoleh biaya penggunaan lampu A sebesar Rp 757,766 dan lampu B sebesar Rp 1.191,048.

Hasil perhitungan biaya listrik menunjukkan penggunaan lampu A 36,38% lebih rendah dibandingkan lampu B. Model LSTM yang diusulkan tidak hanya mempelajari kebiasaan pengguna, tetapi juga mempelajari data lingkungan sekitar, seperti intensitas cahaya dan keberadaan manusia melalui sensor gerakan (*motion*). Sehingga, untuk beberapa kasus, lampu dapat segera mati ketika tidak dibutuhkan, atau ketika terlewat untuk mematikan lampu pada waktunya. Hal ini menjadi salah satu faktor penyebab lebih rendahnya penggunaan listrik oleh lampu A.

KESIMPULAN

Penelitian ini telah menghasilkan model LSTM untuk memprediksi waktu hidup atau matinya suatu lampu. Dengan memanfaatkan data deret waktu, model LSTM tersebut mampu memprediksi pola pemakaian lampu dengan akurasi 99,28% dan RMSE sebesar 0,091. Hasil prediksi model tersebut juga dimanfaatkan untuk otomatisasi pengoperasian lampu sehari-hari.

Hasil eksperimen menunjukkan bahwa biaya listrik pemakaian lampu otomatis 36,38% lebih rendah daripada lampu yang dioperasikan manual.

DAFTAR PUSTAKA

- [1] S. Kumar, "UBIQUITOUS SMART HOME SYSTEM USING ANDROID APPLICATION", International Journal of Computer Networks & Communications, vol. 6, no. 1, p. 33, 2014.

- [2] PLN, Statistik PLN 2018. Jakarta: Sekretariat Perusahaan PT PLN, 2019.
- [3] H. Lin, "Implementing Smart Homes with Open Source Solutions", *International Journal of Smart Home*, vol. 7, no. 4, 2013.
- [4] M. Abd El-Latif Mowad, A. Fathy and A. Hafez, "Smart Home Automated Control System Using Android Application and Microcontroller", *International Journal of Scientific & Engineering Research*, pp. 935-939, 2014.
- [5] R. Adhikari and R. Agrawal, "An Introductory Study on Time Series Modeling and Forecasting", arXiv.org, 2013. Tersedia: <http://arxiv.org/abs/1302.6613>. [Diakses: 15 Juli 2020].
- [6] S. Hochreiter and J. Schmidhuber, "LONG SHORT-TERM MEMORY", *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997.
- [7] Y. Jain and S. Bhandare, "Min Max Normalization Based Data Perturbation Method for Privacy Protection", 2011, vol. 2, pp. 45-49, 2011.
- [8] J. Brownlee, *Deep Learning for Time Series Forecasting*. 2020. Tersedia: <https://machinelearningmastery.com/deep-learning-for-time-series-forecasting/>. [Diakses: 02 Juni 2020].
- [9] A. Akbar, "Biar Gak Kebobolan, Gini Nih Langkah Komplit Menghitung Tarif Listrik di Rumah", *Lifepal Media*, 2020. Tersedia: <https://lifepal.co.id/media/begini-lho-cara-menghitung-tarif-listrik-benar/>. [Diakses: 03- Jun- 2020].
- [10] A. Pribadi, "Tarif Listrik Triwulan I 2020 Tetap", *ESDM*, 2020. Tersedia: <https://www.esdm.go.id/id/media-center/arsip-berita/tarif-listrik-triwulan-i-2020-tetap>. [Diakses: 05 Juni 2020].

LAMPIRAN

Perhitungan Biaya Listrik

Biaya listrik dari penggunaan peralatan elektronik diperlukan agar pengguna dapat mengontrol serta mencegah biaya listrik berlebihan. Perhitungan dapat dilakukan dengan dua cara yaitu secara manual atau melalui sistem.

Perhitungan Secara Manual

Biaya listrik yang dikeluarkan untuk penggunaan alat elektronik di rumah dapat dihitung dengan informasi berupa golongan tarif listrik dan rincian alat elektronik [9].

Golongan tarif listrik dapat dilihat pada *bargainser* di rumah yang menunjukkan besaran daya listrik yang digunakan dalam satuan Volt Ampere (VA). Adapun penyesuaian tarif tenaga listrik Rp/kWh (*Tariff Adjustment*) yang ditetapkan pemerintah Indonesia saat tulisan ini dibuat adalah sebesar Rp 1,352 untuk golongan 900 VA, dan Rp 1,467 untuk selain itu. [10]. Spesifikasi peralatan elektronik yang perlu diketahui adalah banyaknya alat

yang digunakan, besar konsumsi daya yang dibutuhkan dalam satuan Watt, dan lamanya waktu beroperasi. Perhitungan biaya listrik dapat dihitung dengan mengalikan banyaknya alat elektronik, lamanya waktu alat elektronik beroperasi dalam satuan jam, besar konsumsi daya alat elektronik, dan tarif per kWh kemudian dibagi 1000 agar satuannya sama dengan kWh. Rumus dari biaya listrik tersebut, b , adalah sebagai berikut.

$$b = \frac{n t P C}{1000}, \quad (3)$$

Dengan n adalah banyaknya alat elektronik, P adalah besarnya daya listrik, dan C adalah besarnya tarif per kWh.

Perhitungan Melalui Sistem

Proses perhitungan biaya listrik dilakukan di server dengan data arus alat elektronik yang diperoleh dari sensor arus, kemudian dikalikan dengan tarif per kWh. Hasil dari perhitungan tersebut adalah konsumsi daya serta biaya yang harus dibayar.