

Analisis Prediksi pada variabel Temperatur dan Kelembapan di sensor IoT menggunakan metode LSTM

Muhammad Idham Habibie
University of Indonesia
IEEE
Depok, Indonesia
muhammad.idham.habibie@gmail.com

Abstrak— Smart House, salah satu terminologi konsep di era Industri 4.0 yang sedang marak saat ini untuk membangun sebuah automasi di perumahan. Optimisasi yang bisa dilakukan adalah optimisasi parameter Temperatur dan Kelembapan, dimana aplikasinya adalah mengoptimisasi *Air Conditioner* (AC) di sebuah ruangan agar bisa digunakan seefektif mungkin. Kedua variabel ini diteliti selama 2 minggu yang di ambil selama 1 menit. Total dataset yang didapatkan adalah 250.000 baris dengan 2 variabel. Metode untuk melakukan *forecasting* ini adalah metode kuantitatif, menggunakan LSTM. Evaluasi yang digunakan adalah RMS, Loss, MAE, dan R2 Square dengan hasil 0.032, 0.001, 0.007, dan 0.68.

Kata kunci— *Smart House, LSTM, RMS, MAE, Forecast*

I. PENDAHULUAN

Salah satu implementasi aplikasi dalam konsep Internet of Things (IoT) adalah *Smart House*. Dengan berbagai akses Internet yang mudah di jangkau seperti *Internet Protocol* (IP) base, *Smart House* menjadi lebih mudah di implementasikan. Konsep rumah pintar ini menjadikan sebuah rumah memiliki kecerdasan [1] untuk bisa mengatur aktivitas beberapa perangkat di rumah seperti *Air Conditioner*, Televisi, Lampu, Keran Air, dan beberapa perangkat lainnya. Konsep *Autonomous* ini dapat mengoptimisasi energi yang dihasilkan dari rumah tersebut, sebagai contoh adalah penggunaan AC yang cukup berlebihan di dalam rumah. Gambar 1 menggambarkan konsep implementasi *Smart House* menggunakan automasi sistem *Cloud*.



Gambar 1. Konsep sederhana sebuah implementasi *Cloud Base*

Pada dasarnya, konsep *Smart House* sudah di gagas sejak tahun 1975, ketika protokol X10 sudah diperkenalkan ke dalam market. Protokol X10 masih menggunakan *wire* untuk signalling [2], dimana *wire signalling* konsepnya hampir sama dengan sistem telepon. Namun, protokol X10 marketnya berubah dan berevolusi sehingga X10 ini kehilangan market di berbagai pasaran. Di tahun yang sama, konsep *Smart House* masih hanya sebatas komunikasi *controller* antara *User* dan *Client* tanpa monitoring aktivitas seluruh tipe sensor. Namun, dengan teknologi IP, semua

akses ini bisa di jangkau dengan sebuah sistem *intelligent* yang di implementasikan di *Cloud*.

Karena di era Industri 4.0 sudah memperkenalkan berbagai Wi-Fi akses yang sudah terjangkau dimanapun, protokol X10 pun berevolusi menjadi Internet di perumahan. Oleh karenanya, Internet pun menjadi salah satu akses yang sudah bisa digunakan untuk *Smart House*. Namun, Wi-Fi sendiri masih menggunakan konsumsi *power* yang relatif besar dikarenakan karena isu konsumsi *power*. Oleh karenanya, protokol IoT-pun sudah terbagi lagi menjadi 3 tipe, diantaranya adalah NB-IoT, LoRA, maupun SigFox. Ketiga protokol ini pada prinsipnya adalah sama, mereka ingin memberikan komunikasi pada perangkat dengan konsumsi *power* yang kecil seperti sensor agar bisa terhubung ke Internet.

II. TINJAUAN PUSTAKA

Beberapa protokol yang digunakan untuk mengimplementasi IoT adalah NB-IoT, LoRA, dan SigFox. Ketiga protokol ini saat ini sudah di implementasikan ke dalam berbagai aplikasi IoT.

NB-IoT sendiri adalah *licensed spectrum* yang memanfaatkan teknologi dari LTE, dengan menggunakan modulasi OFDM. Namun, NB-IoT bisa melakukan transfer data dalam jumlah ukuran paket yang kecil yang memiliki modulasi jauh lebih cepat. Hanya saja, NB-IoT tidak memiliki protokol yang sama dengan IP lainnya, NB-IoT lebih fokus kepada perangkat dengan konsumsi *power* yang rendah [3].

Berbeda dengan protokol LoRA, protokol LoRA merupakan *unlicensed spectrum* yang memanfaatkan teknologi BPSK / QPSK yang sedikit berbeda dengan NB-IoT. Namun, konsepnya hampir sama dengan NB-IoT, LoRA cocok digunakan untuk perangkat konsumsi *power* yang relatif rendah [3]. LoRA membutuhkan tambahan perangkat gateway untuk bisa di koneksikan ke Internet. Dengan demikian, secara total kebutuhan perangkat, solusi LoRA membutuhkan lebih banyak perangkat dibandingkan dengan NB-IoT. Selain itu, LoRA juga memiliki *drawbacks* di antaranya adalah *Data Rate* yang kecil dan *Latency* yang jauh lebih besar.

Protokol lainnya adalah SigFox, yang merupakan inisiasi dari protokol IoT untuk menggunakan konsumsi *power* yang relatif rendah. Di bandingkan dengan LoRA dan NB-IoT, SigFox tergolong perangkat termurah sekitar <5\$.

Selain membahas berbagai protokol yang ada untuk membantu koneksi IoT, di penelitian ini memfokuskan bagaimana sebuah sistem *Cloud* bisa bekerja untuk

melakukan automasi sebuah sistem *Smart House*. Automasi yang dilakukan dalam penelitian ini adalah melakukan sistem *forecasting* untuk bisa memprediksi *time series* ke beberapa variabel yang dihitung sensor. Beberapa tipe untuk melakukan deteksi Forecasting adalah sebagai berikut :

- Metode Kualitatif
- Metode Kuantitatif
- Metode Kausal

A. Metode Kualitatif

Metode kualitatif menggunakan data dari berbagai pendapat analisis dan cenderung deskriptif. Salah satu metode kualitatif yang biasa di gunakan adalah Delphi Method, yang memanfaatkan *questionnaire* ke beberapa panelis *expertise* di bidang tersebut [4].

B. Metode Kuantitatif

Metode ini sangat bergantung pada data-data sebelumnya untuk bisa mempengaruhi *forecasting* data tersebut. Salah satu metode yang biasa digunakan adalah *Automatic Regression Moving Average* (ARIMA) sebagai metode *forecasting*. Implementasi yang bisa dilakukan menggunakan metode ARIMA ini adalah prediksi Saham. Penelitian ini akan menggunakan Metode Kuantitatif untuk melakukan deteksi dataset ini [5].

Di dalam penelitian ini, metode kuantitatif yang digunakan adalah *Long Short Term Memory* (LSTM) sebagai salah satu metode *deep learning Recurrent Neural Network* (RNN) yang memanfaatkan koneksi *feedback*. Berbeda dengan metode *Neural Network* lainnya, RNN selalu memanfaatkan data input dikombinasikan dengan *internal state (memory)* untuk menghasilkan sebuah output secara bersamaan. Oleh karenanya, RNN sangat baik untuk meningkatkan akurasi *forecasting* tersebut [6].

RNN ini akan memanfaatkan *Gradient Descent* untuk mengoptimisasi nilai akurasi tersebut. *Gradient Descent* sendiri adalah algoritma untuk mengoptimisasi dan *update* parameter model yang telah di definisikan yang bertujuan untuk meminimisasi *error rate* tersebut. Di dalam RNN, hal ini di tuangkan ke dalam kalkulasi *epoch*.

C. Metode Kausal

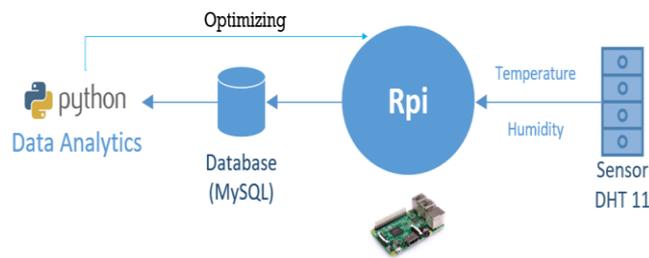
Metode ini adalah metode yang digunakan untuk menghubungkan keterkaitan antara variabel dengan variabel lainnya [5], dengan kata lain, metode kausal ini menggunakan sebuah model *equations* untuk bisa menghubungkan antara satu data dengan data yang lain. Metode yang biasa digunakan adalah Metode Ekonometri untuk menghubungkan satu data dengan data lainnya.

III. METODE PENELITIAN

Peneliti akan memanfaatkan *Raspberry Pi* (mini-PC) sebagai salah satu instrument penelitian untuk melakukan running program selama 2 minggu yang akan di ukur kalkulasinya setiap 1 menit. Adapun program yang digunakan adalah python untuk mengukur temperatur dan kelembapan dari sebuah ruangan. Satuan unit temperatur tersebut adalah derajat *celcius* dan kelembapannya adalah *Relative Humidity* (RH). Total dataset yang didapatkan adalah 250.000 baris dengan 2 *observations*.

Python ini akan mendeteksi jumlah temperatur dan kelembapan setiap 1 menit-nya, dan akan di simpan di

database MySQL. Database MySQL ini akan di analisis menggunakan Python yang terpisah. Lalu, program ini akan melakukan analisa data menggunakan salah satu metode LSTM. Gambar 2 adalah *Flow Diagram* yang digunakan dalam penelitian ini.



Gambar 2. Flow Diagram pengambilan dataset kelembapan dan temperatur

A. Tipe Sensor

Tipe sensor yang digunakan adalah DHT 11, yang bisa menghitung variabel *temperatur* dan kelembapan dalam waktu yang sama. Spesifikasi untuk menghitung temperatur dan kelembapan adalah sebagai berikut [7]:

Pengukuran Kelembapan Udara

- Resolusi pengukuran: 16 bit
- *Repeatability*: $\pm 1\%$ RH
- Akurasi Pengukuran : $25^{\circ}\text{C} \pm 5\%$ RH
- *Interchangeability* : fully interchangeable

Pengukuran Temperatur

- Resolusi pengukuran : 16 bit
- *Repeatability*: $\pm 0.2^{\circ}\text{C}$
- *Range* : $25^{\circ}\text{C} \pm 0.2^{\circ}\text{C}$

B. Parameter LSTM

Metode *Deep Learning* yang digunakan dalam penelitian ini adalah LSTM. Beberapa parameter LSTM yang di definisikan dalam penelitian ini adalah:

- Separate Training and Test Set with a Ratio 0.2 (Training Set 80% dan Test Set 20%). Testing Ratio 20% akan digunakan dalam hal ini.
- LSTM menggunakan beberapa variasi 1, 3, 5, 10, 20, 50, 100 *epochs*
- Batch Size: 50
- Total number of Hidden Layers : 3 layers
- Activation Function: *Relu*

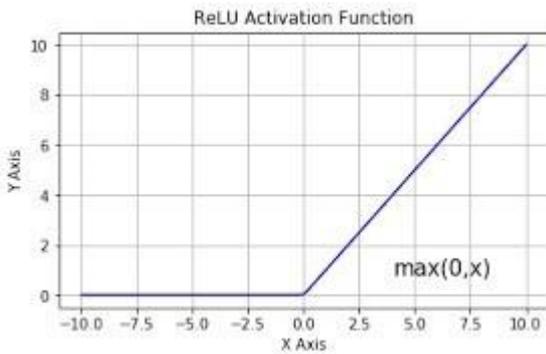
Parameter pertama, bertujuan untuk membagi Training dan Test Set ke dalam 2 database berbeda. Tujuannya adalah sebelum melakukan *forecasting*, untuk mengimplementasi *deep learning*, peneliti dapat melakukan testing dengan menggunakan training dan test set ini, agar bisa mengetahui apakah Dataset ini cenderung *Underfitting* atau *Overfitting*.

Parameter ke-2, terkait dengan variasi *epochs*, *epoch* sendiri adalah parameter untuk menghitung *Forward Pass* dan *Backward Pass* secara bersamaan yang bertujuan untuk meminimisasi *error rate* sebuah fungsi. Jika *Epoch* = 1, artinya *Forward Pass* dan *Backward Pass* dihitung 1 kali. Penelitian ini akan mencoba menggunakan berbagai variasi dari *Epoch* tersebut.

Parameter ke-3, yaitu *Batch Size*, jumlah sampel dalam sebuah training dataset. Jika *Batch Size*-nya semakin kecil, memori yang dibutuhkan dalam komputer akan cenderung lebih tinggi. Di paper ini akan menggunakan 50 training sampel di dalam fungsi ini.

Parameter ke-4, yaitu Hidden Layers, adalah salah satu basis layer di dalam sebuah *neural network*. Pada dasarnya, jumlah *hidden layer* akan menentukan separasi antara satu variabel dengan variabel lainnya. Karena variabel yang digunakan hanya temperatur dan kelembapan, separasi yang digunakan tidak terlalu signifikan antara satu sama lain. Oleh karenanya, jumlah *Hidden Layers* yang digunakan adalah 3 buah[8].

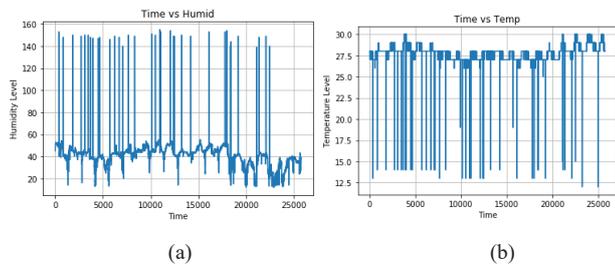
Terakhir, parameter ke-5, yaitu *Activation Function*, adalah sebuah pengukuran performansi di dalam *machine learning* untuk bisa membagi antara *error rate* dengan *real number* menggunakan sebuah nilai kuantifikasi numerik. Dalam hal ini, peneliti menggunakan nilai *Rectified Linear Unit* (Relu) [9]. Gambar 3 mengilustrasikan fungsi Relu dalam *activation function*.



Gambar 3. Relu Activation Function [9]

IV. EXPLAROTARY DATA ANALYSIS (EDA)

Dataset yang di ambil dari sensor DHT11 yang terkoneksi ke *Raspberry Pi* adalah Temperatur dan Kelembapan. Kedua variabel ini diambil setiap menitnya lalu di analisis ke dalam Python. Di dalam Python ini akan muncul dua variabel perbandingan antara waktu vs temperatur dan waktu vs kelembapan, sebagaimana di ilustrasikan dalam gambar 4 (a) dan (b)



Gambar 4. Exploratory Data Analysis (EDA) dalam variabel temperatur dan kelembapan

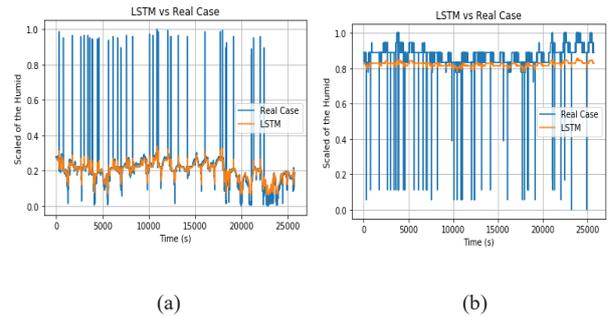
Level kelembapan di setiap waktunya memiliki rata-rata level kelembapan sekitar 43.57 RH, dan temperatur adalah 27.51 derajat celsius. Namun, dari grafik (a) (b)

Gambar 4 ini, ada beberapa anomali yang terjadi seperti level kelembapan yang mencapai peak rata-rata 140 RH, dan temperatur-nya mencapai nilai minimum yaitu 13 derajat celsius.

V. RESULTS AND EVALUATION

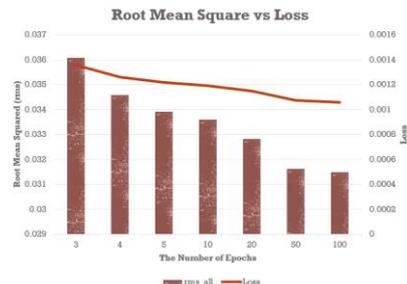
Setelah melakukan dan investigasi EDA, peneliti mencoba mengimplementasikan metode LSTM di dua variabel yang berbeda, diantaranya adalah Kelembapan dan Temperatur. Warna Jingga pada Gambar 5 menunjukkan

hasil kalkulasi LSTM dibanding *Real Case*-nya (warna biru). Hasil yang ditunjukkan oleh warna jingga ini tidak terlalu berubah di bandingkan dengan nilai dataset awalnya.



Gambar 5. (a) Implementing LSTM vs Real Case in Temperatur (b). LSTM vs Real Case in Humidity

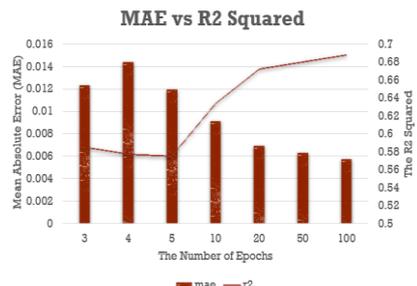
Salah satu cara untuk mengimplementasikan tingkat akurasi hasil testing adalah melakukan evaluasi terhadap metode tersebut. Evaluasi ini akan menggunakan *Root Mean Squared (RMS)*, *Mean Absolute Error (MAE)*, beserta *R2 Squared*. Loss dari sebuah sistem akumulasi LSTM juga di pertimbangkan dalam perhitungan ini. Gambar 6 menunjukkan hasil evaluasi perbandingan antara LSTM dengan *Real Number* yang di ukur dari sebuah parameter RMS dan *Losses*, sebagaimana telah di jelaskan di Gambar 5.



Gambar 6. Metode Evaluasi (RMS vs Loss) di perhitungan LSTM

Pada Gambar 6 tersebut, jumlah *epoch* yang semakin besar akan memiliki hasil akurasi yang lebih baik. Hal ini dapat dilihat dimana nilai *rms* semakin kecil ketika jumlah *epoch* diperbesar. Nilai *epoch* ini berkorelasi dengan nilai *loss*, yang akurasi-nya lebih baik jika *epoch* semakin besar. Namun, penambahan nilai *epoch* di dataset ini tidak signifikan peningkatan akurasi-nya (hanya 0.003).

Di pengukuran parameter yang lain, Gambar 7 menunjukkan sebuah hasil evaluasi perbandingan antara LSTM dengan *Real Number*, yang di ukur oleh parameter MAE dan *R2 Squared*.



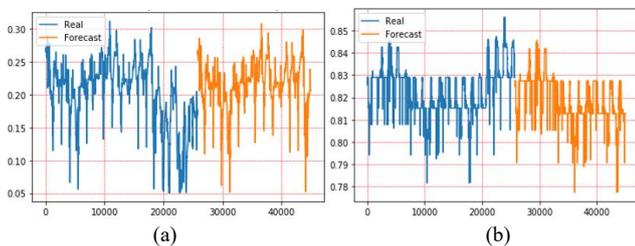
Gambar 7. Metode Evaluasi (MAE vs R2 Squared) di perhitungan LSTM

Gambar 7 menunjukkan hasil yang relatif sama dengan Gambar 6, dimana nilai *epoch* yang semakin besar akan meningkatkan nilai akurasi yang signifikan. Namun, dengan kenaikan nilai *epoch* ini, nilai peningkatannya juga tidak begitu signifikan dalam dataset ini. Kenaikan MAE dari jumlah *epoch* 3 ke jumlah *epoch* 100 adalah 0.006.

Dengan variabel nilai *epoch* yang tidak begitu signifikan untuk mengurangi nilai RMS, MAE, dan R2 Squared, oleh karenanya, peneliti menggunakan *epoch* 20 untuk bisa di kalkulasikan dalam di estimasi *time series*-nya. Hasil evaluasi yang dihasilkan adalah sebagai berikut:

$$\begin{aligned} \text{RMS} &= 0.032 \\ \text{Loss} &= 0.001 \\ \text{MAE} &= 0.007 \\ \text{R2 Square} &= 0.68 \end{aligned}$$

Dengan demikian, Gambar 8 mendeskripsikan sebuah simulasi hasil forecasting untuk Kelembapan dan Temperatur.



Gambar 8. (a) Hasil Forecasting Time vs Kelembapan (b) Hasil Forecasting Time vs Temperatur

VI. PENUTUP

Di dalam penulisan makalah ini, penulis telah mengimplementasikan sebuah metode *LSTM* untuk *Forecasting* di dalam data variabel Temperatur dan Kelembapan ke dalam Raspberry Pi untuk IoT sistem. Dalam hasil penelitiannya, metode *LSTM* memiliki tingkat akurasi yang sangat baik untuk bisa menghasilkan output *forecast* dengan minimalisasi *error*. Selain itu, nilai *epoch* yang besar tidak mengubah nilai RMS untuk menurunkan error rate tersebut. Oleh karenanya, peneliti tidak menggunakan *epoch* yang besar dikarenakan adanya penggunaan memori yang berlebihan.

Hasil output metode untuk *LSTM* adalah rata-rata RMS = 0.032, Loss = 0.001, MAE = 0.007, dan R2 Square = 0.68. Kategori output ini tergolong relatif baik.

UCAPAN TERIMA KASIH

Peneliti mengucapkan terima kasih banyak kepada Dr. Ajjib Setyo Arifin dari Universitas Indonesia untuk masukan terkait dengan *Forecasting* ini.

DAFTAR PUSTAKA

- [1] V. Ricquebourg, D. Menga, D. Durand, B. Marhic, L. Delauche, and C. Logé, "The smart home concept: Our immediate future," *2006 1st IEEE Int. Conf. E-Learning Ind. Electron. ICELIE*, pp. 23–28, 2006.
- [2] TechTarget, "Smart Home / Building." [Online]. Available: <https://internetofthingsagenda.techtarget.com/definition/smart-home-or-building>.
- [3] B. Ray, "NB-IoT vs. LoRa vs. Sigfox," *Link Labs*, 2018. [Online]. Available: <https://www.link-labs.com/blog/nb-iot-vs-lora-vs-sigfox>.
- [4] M. I. Habibie and A. Arifin, "The prediction of mobile data traffic based on the ARIMA model and a disruptive formula in Industry 4.0," *Teknomika*, 2019.
- [5] J. C. Chambers, S. K. Mullick, and D. D. Smith, "How to choose the right forecasting technique," *Harv. Bus. Rev.*, vol. 49, no. 4, pp. 45–70, 1971.
- [6] M. Venkatachalam, "Recurrent Neural Networks - Towards Data Science," *March 1*, 2019. [Online]. Available: <https://towardsdatascience.com/recurrent-neural-networks-d4642c9bc7ce>. [Accessed: 23-Feb-2020].
- [7] Ajie, "Mengukur Suhu dan Kelembapan dengan menggunakan Arduino," 2016. [Online]. Available: <http://saptaji.com/2016/08/10/mengukur-suhu-dan-kelembapan-udara-dengan-sensor-dht11-dan-arduino/>.
- [8] A. Gad, "How Many Hidden Layers/Neurons to Use in Artificial Neural Networks?," *Towards Data Science*, 2018. [Online]. Available: <https://towardsdatascience.com/beginners-ask-how-many-hidden-layers-neurons-to-use-in-artificial-neural-networks-51466afa0d3e>. [Accessed: 23-Feb-2020].
- [9] P. Gadosey, "A beginner's guide to NumPy with Sigmoid, ReLU and Softmax activation functions," *Medium*, 2019. [Online]. Available: <https://medium.com/ai3-theory-practice-business/a-beginners-guide-to-numpy-with-sigmoid-relu-and-softmax-activation-functions-25b840a9a272>. [Accessed: 23-Feb-2020].