

## Analisa dan Implementasi Algoritma *Decision Tree* pada *Smart Home Water Heater*

Bayu Kusuma<sup>1</sup>, Dodi W.Sudiharto, S.T.,M.Kom<sup>2</sup>, Aji Gautama Putra, S.T.,M.T<sup>3</sup>

<sup>1,2,3</sup>Fakultas Informatika, Universitas Telkom, Bandung

<sup>1</sup>bayuksma96@gmail.com, <sup>2</sup>dodiwicaksono@telkomuniversity.ac.id, <sup>3</sup>ajigautama@telkomuniversity.ac.id,

---

### Abstrak

Ketika seseorang hendak mandi dan memerlukan air panas yang dihasilkan dari pemanas listrik, terdapat kebiasaan berupa memanaskan air tersebut sebelum orang tersebut mandi. Untuk seseorang yang memiliki kesibukan padat, tentu hal ini dapat membuang waktunya hanya untuk menunggu air menjadi panas baru kemudian ia mandi. Sedangkan bila pemanas air dinyalakan sepanjang malam, hal ini akan memboroskan listrik. Pada studi ini hendak dibuat pemanas air listrik pintar dengan mengadopsi metode *Decision Tree*. Dengan metode ini diharapkan dapat diketahui kapan pemanas listrik menyala otomatis pada waktu tertentu berdasarkan kebiasaan mandi penggunanya. Diharapkan otomatisasi ini juga akan mengurangi waktu tunggu pengguna untuk menunggu kesiapan air panas dari pemanas air elektrik. Selain itu, juga menghemat pemakaian listrik karena setelah digunakan, pemanas air listrik tersebut akan mati secara otomatis pada waktu tertentu. Pemanas listrik pintar ini dikembangkan menggunakan perangkat IoT (*Internet of Things*) yang terintegrasi dengan *ThingSpeak*. Metode *Decision Tree* yang digunakan, dikembangkan dengan menggunakan *RapidMiner* dan *MATLAB*.

**Kata kunci :** *Decision Tree, ThingSpeak, Pemanas Air, MATLAB.*

---

### Abstract

When someone wants to take a bath and needs hot water produced from an electric heater, there is a habit of heating the water before the person is bathing. For someone who has a busy life, of course this can waste time just to wait for the water to heat up and then take a shower. Whereas if the water heater is turned on all night, this will waste electricity. In this study, a smart electric water heater will be made by adopting the *Decision Tree* method. With this method it is expected to know when the electric heater turns on automatically at a certain time based on the user's shower habits. It is expected that this automation will also reduce the waiting time for users to wait for the readiness of hot water from electric water heaters. In addition, it also saves electricity usage because after use, the electric water heater will automatically turn off at a certain time. This smart electric heater was developed using IoT (*Internet of Things*) devices that are integrated with *ThingSpeak*. The *Decision Tree* method used is developed using *RapidMiner* and *MATLAB*.

**Keywords:** *Decision Tree, ThingSpeak, Water Heater, MATLAB.*

---

## 1. Pendahuluan

### Latar Belakang

Teknologi berbasis IoT saat ini sudah sangat berkembang, di mana segala aktifitas dapat diatur dari jarak jauh, seperti pada penerapan *smart home*. Dengan berkembangnya IoT ini memungkinkan untuk menghemat penggunaan sumber daya listrik di rumah, sebagai contoh pada penggunaan *electric water heater* [1].

Saat ini sudah banyak rumah yang menggunakan pemanas air listrik (*electric water heater*) untuk kebutuhan mandi sehari-hari. Namun, operasional *electric water heater* saat ini masih terbilang manual, seperti pengguna masih harus menunggu air menjadi panas, atau menyalakan semalaman dalam posisi *standby* yang mengakibatkan banyak menghabiskan daya listrik. Oleh sebab itu, perlu adanya suatu sistem yang dipasang pada *electric water heater* agar dapat mengefisienkan penggunaan listrik dan juga membuat nyaman penggunaannya dalam arti tidak terlalu lama menunggu air menjadi panas.

Pada studi ini akan dibangun sistem yang dapat menghemat penggunaan listrik pada *electric water heater*, dengan cara memasang *microcontroller* pada *relay power heater* dan merekam pola penggunaan *electric water heater* dengan menggunakan sensor PIR (*Passive Infra-Red*), DHT11, dan *Magnetic Door* yang terhubung dengan *microcontroller* NodeMCU [2]. Pola penggunaan *electric water heater* tersebut dikirim melalui internet untuk disimpan menggunakan *ThingSpeak* [3]. Data pola penggunaan *electric water heater* tersebut kemudian dianalisis menggunakan *Decision Tree* yaitu C.45. Dengan data yang sudah diolah, dapat diketahui pola kebiasaan pengguna seperti kapan penghuni rumah akan menggunakan *electric water heater*.

### Topik dan Batasannya

Tugas akhir ini membahas cara implementasi *Decision Tree* pada *smart water heater* dalam menentukan waktu terbaik kapan penghuni rumah menggunakan *water heater*. Dalam proses ini sensor PIR (*Passive Infra-Red*), DHT11, dan *Magnetic Door* akan merekam setiap pola penggunaan kamar mandi dan mengirimkan ke *ThingSpeak* setiap lima menit. Berdasarkan data yang diperoleh akan dibangun model *Tree* menggunakan metode C4.5.

Batasan masalah yang terdapat dalam tugas akhir ini adalah sensor PIR (*Passive Infra-Red*), DHT11, dan *Magnetic Door* yang berfungsi untuk mendeteksi dan menangkap setiap pola penggunaan kamar mandi selama tujuh hari. *Water Heater* akan menyala berdasarkan hasil yang didapat oleh *Machine Learning* menggunakan algoritma C4.5. *microcontroller* yang digunakan adalah NodeMCU.

### Tujuan

Tujuan yang hendak dicapai dalam tugas akhir ini adalah mengimplementasikan *Decision Tree* pada *smart water heater* dan menguji performansi algoritma C4.5 dalam memprediksi penggunaan *water heater* berdasarkan data yang diperoleh sensor PIR, DHT11 dan *Magnetic Door*.

### Organisasi Tulisan

Penelitian ini terbagi menjadi beberapa bagian. pada bagian 1 diatas sudah dipaparkan pendahuluan. selanjutnya pada bagian 2 akan dijelaskan studi terkait berdasarkan penelitian-penelitian sebelumnya. sistem yang dibangun pada penelitian ini dijelaskan pada bagian 3. pada bagian 4 akan dijelaskan hasil pengujian dan analisis dari sistem yang dibangun. terakhir penelitian ini disimpulkan pada bagian 5 beserta saran untuk penelitian selanjutnya.

## 2. Studi Terkait

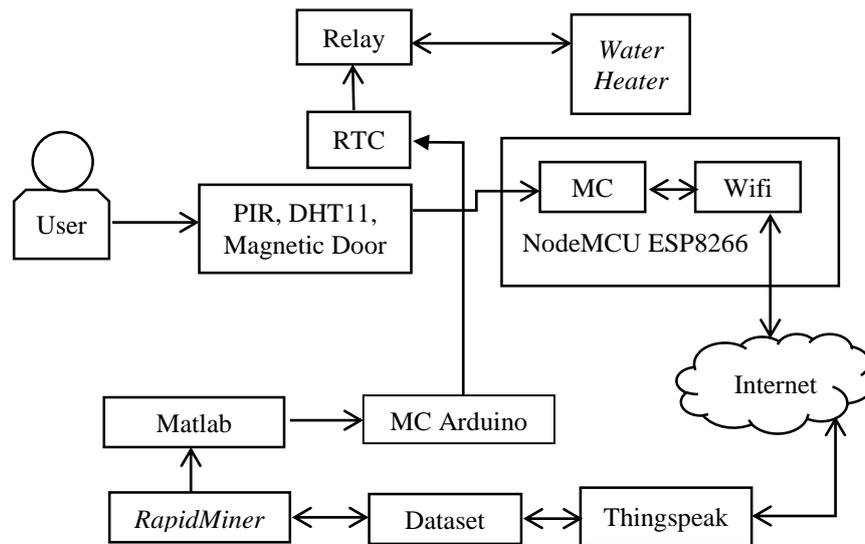
Penggunaan *Microcontroller* terhadap dunia IoT saat ini sangat berpengaruh, sudah banyak penelitian yang melibatkan *Microcontroller* seperti pada penelitian [4] yang melibatkan *Microcontroller* dan *machine learning* . dan pengimplementasian NodeMCU ESP8266 pada *Smart Home* [5], Pada jurnal [6] dilakukan penelitian untuk memprediksi perilaku *Smart Home Lighting* dengan menggunakan *Machine Learning*, penelitian tersebut menggunakan metode *Decision Tree*. Penerapan algoritma C4.5 dalam memprediksi sudah banyak digunakan, algoritma C4.5 adalah salah satu metode untuk membuat *decision Tree* berdasarkan data train dan data test. Algoritma C4.5 merupakan perkembangan dari algoritma ID3 [7]. Algoritma C4.5 mampu mengatasi *missing value*, *continue* data dan *Pruning*. Secara umum algoritma C4.5 memiliki beberapa tahapan untuk membangun *decision tree* seperti pilih atribut sebagai akar, buat cabang untuk tiap-tiap nilai, bagi kasus dalam cabang, lalu ulangi proses untuk setiap cabang sampai semua kasus pada cabang memiliki kelas yang sama [8].

Maka dari itu pada tugas akhir ini akan menggunakan NodeMCU dan sensor PIR, DHT11, dan *Magnetic Door* sebagai sensor yang akan merekam setiap pola penggunaan kamar mandi. Penggunaan NodeMCU untuk *smart home* sendiri sudah banyak, contohnya pada [5] . Yang kemudian akan di olah menggunakan MATLAB.

## 3. Sistem yang Dibangun

### Deskripsi Sistem

Penelitian tugas akhir ini akan membuat *Smart Water Heater* yang mampu menyala secara otomatis berdasarkan pola yang didapat dan telah diolah menggunakan MATLAB. Alat yang digunakan adalah sensor PIR, DHT11, *Magnetic door*, Relay, NodeMCU ESP8266, *ThingSpeak*, dan *software RapidMiner*. Berikut adalah gambaran sistem *Smart Home Water Heater* pada gambar 3.1.



Gambar 3.1 Gambaran Umum Sistem

Dijelaskan terdapat sensor PIR, DHT11, *Magnetic Door* yang terhubung dengan NodeMCU, digunakan untuk membaca pola ketika pengguna masuk kamar mandi yang kemudian data dikirimkan melalui *ThingSpeak* menggunakan internet, pola penggunaan kamar mandi akan direkam atau disimpan selama tujuh hari pada *ThingSpeak* kemudian data di *export* ke Excel yang kemudian akan dilakukan pelabelan untuk dijadikan *dataset*, setelah *dataset* terbentuk *dataset* akan diproses menggunakan *software RapidMiner* untuk proses pembangunan *Tree* [7], dan mencari akurasi dengan metode C4.5, hasil dari *software RapidMiner* yaitu terbentuknya *Decision Tree* dan Akurasi dari hasil *split* data *Train* dan data *Test*. Setelah terbentuknya *Tree*, proses selanjutnya yaitu menggunakan MATLAB yang menghasilkan waktu *Water Heater* untuk menyala. *Water Heater* akan menyala berdasarkan waktu prediksi yang didapat oleh MATLAB dengan cara menggunakan Arduino dan modul RTC (*Real Time Clock*) dan terhubung dengan *relay* agar pemanas dapat menyala atau mati secara otomatis berdasarkan waktu prediksi.

### Record Data Sensor ke ThingSpeak

Sensor PIR, DHT11, dan *Magnetic Door* yang berfungsi untuk merekam setiap pola penggunaan kamar mandi selama tujuh hari akan disimpan pada *ThingSpeak*, *ThingSpeak* adalah antarmuka *open source Internet of Things* (IoT) aplikasi dan *Application Programming Interface* (API) untuk menyimpan dan mengambil data dari hal menggunakan protokol HTTP melalui Internet atau melalui *Local Area Network* [10]. Data yang direkam akan membentuk grafik seperti pada gambar 3.2.



Gambar 3.2 Hasil Record sensor pada ThingSpeak

Pada Gambar 3.2 dijelaskan hasil yang didapat oleh sensor pada ThingSpeak yaitu berupa grafik yang dihasilkan oleh sensor setiap lima menit, berisi waktu dan kondisi sensor. Pola ini dapat digunakan untuk melihat waktu penggunaan kamar mandi.

**Proses Pelabelan Pada Dataset**

Data yang telah disimpan pada ThingSpeak dapat di export menjadi berkas excel. Kemudian data akan dilakukan pelabelan, berikut tampilan data pada excel saat proses pelabelan pada Tabel 1.

Tabel 1. Dataset yang didapat dari ThingSpeak.

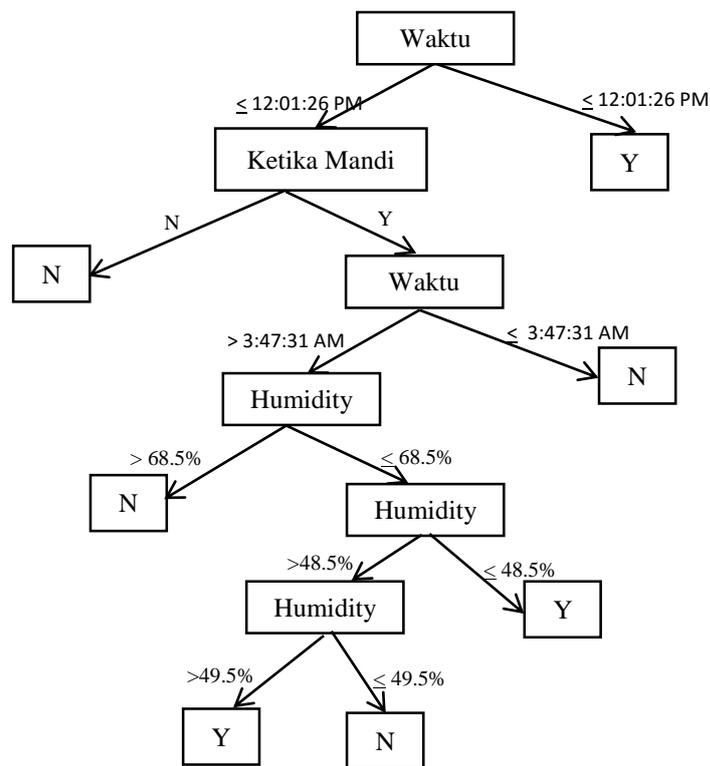
waktu	Temperature	Humidity	Pir	Door	Ketika mandi	pemanas nyala
28/07/18 11:16:23	24 °C	48%	0	0	N	Y
28/07/18 11:46:28	23 °C	48%	0	1	Y	Y
28/07/18 12:16:30	23 °C	49%	0	1	Y	N
28/07/18 12:46:32	23 °C	60%	0	0	N	N
28/07/18 13:16:35	23 °C	63%	0	0	N	N
28/07/18 13:46:40	23 °C	65%	0	0	N	N
28/07/18 14:16:43	23 °C	61%	0	0	N	N
28/07/18 14:46:46	23 °C	58%	0	0	N	Y
28/07/18 15:16:50	23 °C	49%	0	1	Y	N
28/07/18 15:46:58	23 °C	49%	0	0	N	Y
28/07/18 16:16:59	23 °C	62%	0	1	Y	N
28/07/18 16:47:03	23 °C	61%	0	0	N	N
28/07/18 17:47:12	23 °C	68%	0	0	N	N
28/07/18 18:17:15	23 °C	69%	0	0	N	N
28/07/18 18:47:19	23 °C	62%	0	0	N	N
28/07/18 19:17:21	23 °C	59%	0	0	N	N
28/07/18 20:17:32	23 °C	57%	0	0	N	N
28/07/18 20:26:32	24 °C	60%	1	0	N	N
28/07/18 20:35:56	23 °C	57%	0	0	N	Y
28/07/18 20:41:00	23 °C	58%	1	1	Y	N
28/07/18 20:46:03	24 °C	70%	1	0	N	Y
28/07/18 20:51:07	24 °C	58%	1	1	Y	N

Pada Tabel 1. adalah sebuah proses saat pelabelan dataset, hasil yang telah didapat oleh sensor selama tujuh hari akan ditampilkan di excel, data yang didapat sebanyak 1860 data. Data yang diperoleh adalah hasil dari rekaman selama tujuh hari pada rumah saya, yang berisikan 5 orang dewasa dan 2 anak-anak.

Sebenarnya pada atribut waktu tidak bisa diolah secara langsung oleh *RapidMiner*, karena memiliki format berupa jam, menit, dan detik. Sehingga data ini perlu diubah dulu dalam format *boolean*. Sebagai contoh untuk format 02:30:20 maka perlu diubah menjadi bernilai 2,50 di mana untuk format detik diabaikan. Data tersebut kemudian diolah oleh *RapidMiner* untuk mendapatkan *Tree*. Setelah itu, nilai tersebut dikembalikan kembali untuk memperoleh nilai waktu dengan format jam:menit di mana untuk format detik langsung di-set dengan angka 00. Nilai format waktu itu dikembalikan ke aslinya supaya dapat dilakukan otomatis kapan perangkat menyala atau mati.

### Proses Pembangunan *Tree* Menggunakan Algoritma C4.5

Proses pembangunan *Tree* dilakukan menggunakan *RapidMiner*. Pada *RapidMiner* di-set untuk membangun *Tree* berdasarkan nilai *Gain Ratio*. Metode yang dipilih pada *RapidMiner* adalah C4.5. Hasil eksekusi *RapidMiner* untuk membangun *Tree* sesuai *dataset* (Tabel 1) dapat dilihat seperti pada gambar 3.3.



Gambar 3.3 Hasil *Tree* menggunakan *RapidMiner*

Untuk penjelasan sesuai *Tree* di gambar 3.3 adalah sbb:

- Jika waktu  $< 12:01:26$  PM pemanas akan menyala, dan jika waktu  $> 12:01:26$  PM akan dilihat dari nilai “ketika mandi”. Jika nilai “ketika mandi” adalah N yaitu pada  $< 3:47:31$  AM, maka pemanas akan mati. Tetapi bila nilai “ketika mandi” adalah selain N, maka pemanas akan atribut *Humidity*.
- Berdasarkan atribut *Humidity* (baris ke-4), jika nilai *Humidity*  $> 68.5\%$  maka pemanas akan mati karena atribut *Humidity* bernilai N, dan jika nilai *Humidity*  $< 68.5\%$ , maka akan dilihat berdasarkan nilai *Humidity* pada baris berikutnya.
- Jika nilai atribut *Humidity* baris berikutnya (baris ke-5) bernilai  $< 48.5\%$ , maka pemanas akan menyala, karena nilai atributnya Y. Tetapi jika nilai *Humidity*  $> 48.5\%$ , maka akan dilihat berdasarkan nilai atribut pada baris terakhir, yaitu jika nilai *Humidity* bernilai  $< 49.5\%$ , maka pemanas akan mati, dan jika nilainya  $> 49.5\%$  pemanas akan menyala.

Untuk penjelasan singkatnya adalah lebih dari jam 12:01:26 PM sampai jam 3:47:31 AM pemanas mati, selain waktu itu pemanas menyala. Untuk *Humidity* bernilai lebih dari  $> 49.5\%$  sampai  $< 68.5\%$  pemanas akan menyala, selain dari nilai tersebut untuk atribut *Humidity* pemanas akan mati.

Pada Gambar 3.3 dijelaskan hasil *Tree* yang dibangun menggunakan *RapidMiner*, hasil tersebut didapat dengan algoritma C4.5 dengan max depth *Tree* sebanyak 20. Leaf paling luar merupakan label ketika pemanas nyala (Y), dan ketika pemanas mati (N). Dengan mengetahui bentuk *Tree* memudahkan proses MATLAB dalam mengeluarkan prediksi. Hasil pembangunan *Tree* ini berdasarkan algoritma C4.5 untuk membangun pohon keputusan yaitu :

- Pilih atribut sebagai akar
- Buat cabang untuk masing-masing nilai
- Bagi kasus dalam cabang
- Ulangi proses untuk masing-masing cabang sampai semua kasus pada cabang memiliki kelas yang sama.

Untuk memilih atribut sebagai akar didasarkan pada nilai gain tertinggi dari atribut yang ada. Untuk menghitung gain digunakan rumus [9] (3.1) :

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} * Entropy(S_i) \quad (3.1)$$

Keterangan :

$S$ : Himpunan Kasus,  $A$ : Atribut,  $N$ : Jumlah Partisi atribut  $A$ ,  $|S_i|$ : Jumlah kasus pada partisi ke- $i$ ,  $|S|$ : Jumlah kasus dalam  $S$

Sedangkan perhitungan nilai *Entropy* dapat dilihat pada rumus [9] (3.2):

$$Entropy(A) = \sum_{i=1}^n - p_i * \log_2 p_i \quad (3.2)$$

$S$ : Himpunan Kasus,  $A$ : Fitur,  $N$ : Jumlah partisi  $S$ ,  $P_i$ : Proporsi dari  $S_i$  terhadap  $S$

## 4. Evaluasi

### 4.1 Hasil Pengujian

Berikut hasil pengujian yang dilakukan dalam tugas akhir ini :

#### 4.1.1 Performansi *Decision Tree*

Pengujian ini dilakukan untuk melihat akurasi dari *Tree* yang telah dibangun, berdasarkan hasil *split atribut data* dari *dataset* kemudian diolah menggunakan algoritma C4.5 pada *RapidMiner* untuk proses pembangunan *Tree* dengan berbagai kemungkinan yang mungkin muncul, di mana untuk pemilihan *split atribut* menggunakan metode *gain ratio*. Hasil dari *split data* akan dihitung menggunakan metode perhitungan akurasi *Confusion Matrix*. *Split data* menggunakan *data training* sebanyak 0,7715053763440 dari keseluruhan data (data untuk 5 hari), dan untuk *data testing* sebanyak 0,228494623655914 dari keseluruhan data (data untuk 2 hari) dari data yang didapat oleh sensor sebanyak 1.860 data. Berikut adalah tabel *Confusion Matrix* (yang dihasilkan *RapidMiner*) seperti pada gambar 4.1.

accuracy: 93.65%			
	true Y	true N	class precision
pred. Y	16	18	47.06%
pred. N	9	382	97.70%
class recall	64.00%	95.50%	

Gambar 4.1 Hasil Perhitungan Metode *Confusion Matrix*

Pada gambar 4.1 dijelaskan bahwa hasil dari metode perhitungan *Confusion Matrix*, sistem yang dibangun memiliki akurasi sebesar 93,65%. Metode ini melakukan perhitungan dengan output seperti :

- Recall*, adalah tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi.
- Precision*, adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem.
- Accuracy*, didefinisikan sebagai tingkat kedekatan antara nilai prediksi dengan nilai actual.

#### 4.1.2 Akurasi Alat Dalam Memprediksi Hari ke-Enam dan ke-Tujuh

Pada Pengujian ini dataset yang telah didapat selama tujuh hari akan dipecah menjadi dua yaitu, data *Train* sebanyak lima hari dan data *test* sebanyak dua hari. *Dataset* dibuat pada tanggal 28 Juli 2018 hingga 4 Agustus 2018. Pengujian ini menggunakan MATLAB berdasarkan *Tree* yang didapat pada *Software RapidMiner*. Pengujian ini bertujuan untuk melihat akurasi yang didapat oleh algoritma yang telah dibangun berdasarkan *Tree*. Berikut hasil prediksi yang didapat pada Tabel 2.

Tabel 2. Hasil Waktu Prediksi Menggunakan MATLAB

Waktu	Prediksi pemanas nyala
03/08/2018 12:04:06 AM	N
03/08/2018 12:09:08 AM	N
03/08/2018 12:14:10 AM	N
03/08/2018 12:19:12 AM	N
03/08/2018 12:24:14 AM	N
03/08/2018 12:29:16 AM	N
03/08/2018 12:34:17 AM	N
03/08/2018 12:39:20 AM	N
03/08/2018 12:44:21 AM	N
03/08/2018 12:49:23 AM	N
03/08/2018 12:54:25 AM	N
03/08/2018 12:59:27 AM	N
03/08/2018 01:04:30 AM	N
03/08/2018 01:09:33 AM	N
03/08/2018 01:14:36 AM	N
03/08/2018 01:19:39 AM	N
03/08/2018 01:24:42 AM	N
03/08/2018 01:29:45 AM	N
03/08/2018 01:34:48 AM	N

Pada Tabel 2. dijelaskan terkait hasil prediksi *Water Heater* menyala pada ke enam dan tujuh. Terdapat *output* tanggal pada hari ke enam dan ketujuh, dan waktu kondisi *Water Heater*.

## 4.2 Analisis Hasil Pengujian

Berikut Hasil Analisis pada tugas akhir ini:

### 4.2.1 Analisis Performansi *Decision Tree*

Analisis dari hasil pengujian performansi *Decision Tree* bahwa *Tree* yang dibangun oleh *software RapidMiner* Memiliki akurasi yang baik yaitu sebesar 93,65%. Akurasi ini didapat berdasarkan *split data*. Hasil yang didapat dengan metode perhitungan *Confusion Matrix* yaitu :

- Berdasarkan *Recall*, ada 64% dalam memprediksi kebenaran data Y tetapi ada 36% yang seharusnya diprediksi Y tetapi tidak terprediksi, dan N sebesar 95,50% dalam memprediksi kebenaran data N tetapi ada 4,5% data yang seharusnya diprediksi N tetapi tidak terprediksi.
- Berdasarkan *precision*, Banyaknya nilai N dianggap sebagai Y sebesar 47,06%, dan banyaknya nilai N di prediksi sebagai nilai N sebesar 97,70%.

### 4.2.2 Analisis Akurasi Alat Dalam Memprediksi Hari Ke-Enam dan Ke-Tujuh

Analisis hasil prediksi waktu kapan *Water Heater* seharusnya menyala sudah baik seperti pada gambar 4.2.

Name ▲	Value
datatest	425x7 cell
hasil	425x8 cell
hasil_data	425x8 cell
num	1860x4 double
num1	425x5 double
persentase	94.5882
raw	1861x8 cell

Gambar 4.2 Tingkat Akurasi Menggungakan MATLAB

Pada gambar 4.2 dijelaskan bahwa tingkat akurasi yang didapat setelah menguji prediksi untuk hari ke-6 dan ke-7, berdasarkan penggunaan *Tree*, memiliki tingkat akurasi sebesar 94,58% (dalam memprediksi waktu penggunaan *water heater* pada hari ke enam dan tujuh). Hasil tersebut didapat dengan cara menyesuaikan data hari ke enam dan tujuh yang didapat secara *real* (dari *ThingSpeak*) dengan hasil prediksi berdasarkan *Tree*.

## 5. Kesimpulan

Berdasarkan hasil analisis pengujian *Smart Home Water Heater* dengan menggunakan *software RapidMiner* dalam membangun *Decision Tree* berdasarkan Algoritma C4.5 dapat ditarik kesimpulan :

1. Sensor yang digunakan seperti PIR, DHT11, dan *Magnetic Door* berjalan dengan semestinya yaitu mampu mendeteksi setiap pola penggunaan kamar mandi, dapat dilihat dari grafik yang dihasilkan pada *ThingSpeak*.
2. Kinerja Algoritma C4.5 dalam membangun sebuah *Decision Tree* memiliki hasil yang akurat sebesar 93,65% dalam memberikan prediksi.

Saran dari tugas akhir ini yaitu :

1. Lebih banyak parameter yang digunakan akan semakin baik *Decision Tree* dalam memberikan prediksi.
2. Penambahan waktu monitoring untuk meningkatkan performansi *Decision Tree* dalam memprediksi.

## Daftar Pustaka

- [1] D. Burmester, R. Rayudu, and W. K. G. Seah, "Instantaneous control of a DC water heater for a PV system," in *Power Syst. Technol. (POWERCON)*, 2016, pp. 1-6.
- [2] Z. Z. Z. Zhang, X. G. X. Gao, J. Biswas, and J. K. W. J. K. Wu, "Moving targets detection and localization in passive infrared sensor networks," in *Int. Conf. Inf. Fusion*, 2007, pp. 1-6.
- [3] A. I. Abdul-Rahman and C. A. Graves, "Internet of Things application using tethered MSP430 to ThingSpeak cloud," in *Service-Oriented System Engineering (SOSE)*, 2016, pp. 352-357.
- [4] S. A. Zulkifli, M. N. Hussin, and A. S. Saad, "MATLAB-Arduino as a low cost microcontroller for 3 phase inverter," in *Research and Development (SCORED)*, 2014, pp. 1-5.
- [5] M. F. Wicaksono, "Implementasi modul wifi NodeMCU Esp8266 untuk smart home," *Jurnal Sistem Komputer*, vol. 6, no. 1, pp. 9-14, 2017.
- [6] Dinata, I. B. P. P., & Hardian, B. "Predicting smart home lighting behavior from sensors and user input using very fast decision tree with Kernel Density Estimation and improved Laplace correction," in *Advanced Computer Science and Information Systems (ICACSIS)*, 2014, (pp. 171-175).
- [7] P. Chandrasekar, K. Qian, H. Shahriar, and P. Bhattacharya, "Improving the prediction accuracy of decision tree mining with data preprocessing," *Computer Software and Applications Conference (COMPSAC)*, 2017, pp. 481-484.
- [8] S. Informasi, U. P. Batam, K. Kunci, K. Kerja, and D. Mining, "Analisa dan penerapan Algoritma C4.5 dalam data mining untuk mengidentifikasi faktor-faktor penyebab kecelakaan kerja kontruksi PT. Arupadhatu Adisesanti," *Jurnal Online Informatika*, vol. 2, no. 1, pp. 36-41, 2017.
- [9] S. S. Yuni, H. Dedy, W. Agus Perdana, "Penerapan algoritma C4.5 untuk klasifikasi predikat keberhasilan mahasiswa di AMIK Tunas Bangsa," *Jurnal Riset Sistem Informasi & Teknik Informatika*, vol. 1, no. 1, 2016.
- [10] M. A. G. Maureira and L. Teernstra, "ThingSpeak – an API and Web Service for the Internet of Things," *Leiden Institute of Advanced Computer Science*, 2011.