

RANCANG BANGUN ALAT DENGAN FITUR SISTEM POLA PENJADWALAN LAMPU OTOMATIS MENGGUNAKAN JARINGAN SARAF TIRUAN PADA RUMAH PINTAR SUSI CANTIX (*SMARTHOME UPGRADE SYSTEM CONTROL AND AUTOMATIC XTRA*)

Tegar Amri ^{*a}, , Arief Syaichu ^b, , Ali Muhtar ^c

^a Program Studi Teknik Elektro, Jurusan Teknik Elektro, Informatika dan Sistem Fisis,
Insititut Teknologi Sumatera

^b Program Studi Teknik Elektro, Jurusan Teknik Elektro, Informatika dan Sistem Fisis,
Insititut Teknologi Sumatera

^c Program Studi Teknik Elektro, Jurusan Teknik Elektro, Informatika dan Sistem Fisis,
Insititut Teknologi Sumatera

* Corresponding E-mail: tegaramri8@gmail.com

Abstract: *In the current era, the rapid advancement of technology is felt in all aspects of life, as now advances in technology have also begun to penetrate into the property sector, one of which is the house. There are already many kinds of smart home appliances on the market, but this can actually be a problem when we are traveling and the situation of the house we leave behind is very clear that there are no inhabitants. because smart home equipment has a price that is quite expensive it will be the target of theft when the house is empty. Therefore, to minimize the things that can be detrimental to it has been made a tool with an automatic scheduling pattern system with the operation of lights in accordance with the use of residents of the house. The device is successfully operated in accordance with the habits of the residents of the house, and can make criminals stop their intention to steal when the condition of the house left as if there are residents who are operating the lights at certain hours.*

Keywords: *Smart House, Departure, Theft, Scheduling Pattern, Lights.*

Abstrak: Di era saat ini pesatnya kemajuan teknologi terasa diseluruh aspek kehidupan, seperti saat ini kemajuan teknologi juga sudah mulai merambah ke bidang properti yang salah satu-nya adalah rumah. Sudah banyak berbagai macam peralatan rumah pintar yang ada dipasaran, tetapi hal ini malah dapat menjadi suatu permasalahan ketika kita sedang berpergian dan keadaan rumah yang kita tinggalkan sangat jelas sedang tidak ada penghuninya. karena peralatan rumah pintar memiliki harga yang cukup mahal maka akan menjadi sasaran pencurian ketika rumah tersebut kosong. Oleh karena itu, untuk meminimalisir hal-hal yang dapat merugikan tersebut telah dibuat sebuah alat dengan sistem pola penjadwalan otomatis dengan pengoperasian lampu sesuai dengan pemakaian penghuni rumah. Alat tersebut berhasil dioperasikan sesuai dengan kebiasaan penghuni rumah, serta dapat membuat pelaku kriminalitas mengurungkan niat untuk mencuri ketika keadaan rumah yang ditinggalkan seakan-akan terdapat penghuni yang sedang mengoperasikan lampu pada jam-jam tertentu

Kata Kunci : *Rumah Pintar, Berpergian, Pencurian, Pola Penjadwalan, Lampu*

I. PENDAHULUAN

Kemajuan teknologi merupakan sesuatu yang tidak bisa dihindari dalam kehidupan ini, karena kemajuan teknologi akan berjalan sesuai dengan kemajuan ilmu pengetahuan. Setiap inovasi diciptakan untuk memberikan manfaat positif bagi kehidupan manusia [1]. Di era saat ini pesatnya kemajuan teknologi terasa diseluruh aspek kehidupan, seperti pada dunia industri yang kebanyakan sudah menggantikan pekerjaannya dengan robot-robot agar pekerjaan lebih cepat dan efisien. Saat ini kemajuan teknologi juga sudah mulai merambah ke bidang properti yang salah satu-nya adalah rumah. Dimulai dari kebiasaan penghuni rumah yang terkadang lupa untuk mematikan lampu saat berpergian dan juga lupa menghidupkan nya. Sehingga sering terjadi pembengkakan tagihan listrik yang seharusnya dapat dihindari jika saat berpergian kita dapat mengeceknya pada suatu sistem aplikasi rumahan pada smartphone yang sering kita gunakan. Data total kebutuhan listrik di Indonesia selama kurun waktu 17 tahun (2003 s.d. 2020) diperkirakan tumbuh sebesar 6,5% per tahun dari 91,72 TWh pada tahun 2002 menjadi 272,34 TWh pada tahun 2020. Secara nasional, kebutuhan listrik terbesar adalah sektor industri, disusul sektor rumah tangga, usaha dan umum [2].

Selain permasalahan kenaikan pemakaian kebutuhan listrik. Hal lain yang sering terjadi saat ini yaitu maraknya pencurian yang sering terjadi di lingkungan rumah ketika kita sedang berpergian dalam waktu yang cukup lama. Dari data laporan pencurian yang dihimpun oleh Polda Metro Jaya, terdapat total 3.138 laporan masuk dengan kasus pencurian. Dari laporan tersebut mengerucut menjadi 3 jenis pencurian yang terjadi di lingkungan rumah tangga yaitu, pencurian kendaraan bermotor, pencurian dengan pemberatan dan pencurian dengan kekerasan. Dari ketiga pencurian tersebut pencurian dengan pemberatan menjadi angka statistik tertinggi dengan laporan kasus 2043, pencurian kendaraan bermotor 1637 dan pencurian dengan kekerasan 470 [3].

Oleh karena itu, untuk meminimalisir hal-hal yang dapat merugikan tersebut diperlukan sebuah alat dengan sistem rumah pintar yang dapat dikontrol dari kejauhan dengan menggunakan *smartphone* serta dapat membuat pelaku kriminalitas mengurungkan niat untuk mencuri ketika keadaan rumah yang ditinggalkan seakan-akan terdapat penghuni yang sedang mengoperasikan lampu pada jam-jam tertentu.

II. LANDASAN TEORI

2.1 Jaringan Syaraf Tiruan

Jaringan syaraf tiruan (JST) dikembangkan berdasarkan proses pembelajaran otak manusia, disebut tiruan karena jaringan syaraf ini diimplementasikan dengan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama pembelajaran. Ada beberapa teknik pembelajaran JST, salah satu yang paling sering digunakan adalah *backpropagation* (BP). Inti dari BP adalah melakukan perhitungan maju untuk mengetahui output dan kinerja jaringan, selanjutnya dilakukan perhitungan mundur untuk mengetahui *error* jaringan yang kemudian digunakan sebagai perubahan bobot. Seiring dengan kebutuhan mendapatkan proses pembelajaran yang lebih cepat, maka dikembangkan beberapa algoritma pembelajaran baru dengan prinsip BP, diantaranya: *gradient descent*, *resilient backpropagation*, *quasi newton*, dan *levenberg marquardt* [4].

2.1.1 Proses Normalisasi

Proses normalisasi merupakan proses yang dilakukan untuk merubah data *input* dan *target* yang akan digunakan dalam proses pelatihan dan pengujian JST berada pada suatu *range* tertentu. Terdapat banyak beberapa cara metode dalam melakukan normalisasi seperti, *min-max*, *decimal scaling*, *sigmoid*, dan lainnya. Metode normalisasi yang sering digunakan adalah normalisasi *min-max* dimana normalisasi ini akan menskalakan suatu nilai yang diberikan kedalam nilai baru antara 0 sampai dengan 1 berdasarkan nilai maksimum dan minimum *dataset* tersebut. Berikut *formula* untuk mendapatkan nilai baru dengan menggunakan normalisasi *min-max* [4]. Diberikan nilai yang bersesuaian ($\hat{S}$), dimana $k = 0, 1, \dots, n$, maka nilai normalisasinya adalah.

$$\hat{S} = \frac{S - \min(S_k)}{\max(S_k) - \min(S_k)}$$

keterangan:

$\hat{S}$ = nilai *input* yang telah dinormalisasi
 S = nilai *input* yang belum dinormalisasi
 $\min(S_k)$ = nilai *input* terkecil
 $\max(S_k)$ = nilai *input* terbesar

2.1.2 Fungsi Aktivasi Sigmoid

Fungsi Sigmoid *biner* memiliki nilai pada *range* 0 sampai 1. Oleh karena itu fungsi ini sering digunakan untuk JST yang membutuhkan nilai *output* pada interval 0 sampai 1. Fungsi sigmoid *biner* dirumuskan sebagai berikut [4]:

$$f(x) = \frac{1}{1 + e^x}$$

dengan,

$$f'(x) = x * (1 - x)$$

keterangan:

$f(x)$ = nilai aktivasi sebelumnya

$f'(x)$ = nilai aktivasi yang telah ter-update

x = nilai *input* yang telah dinormalisasi

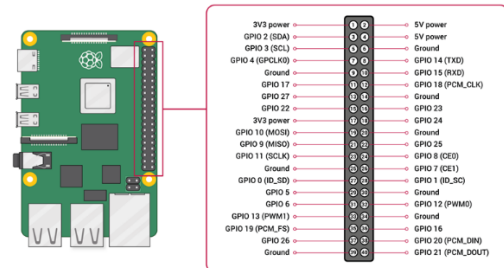
2. 1. 3 Pelatihan Data

Pelatihan data merupakan *validasi* model jaringan. Pelatihan data dilakukan beberapa kali *trial* dan *error* untuk mendapatkan jaringan terbaik dengan menentukan jumlah neuron [4]. Pelatihan data pada alat sistem rumah pintar dilakukan dengan memberikan *learning rate* 15000, hal ini dilakukan agar hasil *output* yang didapat *error* nya tidak terlalu jauh dari data input yang diolah. Dalam pelatihan data harus mempertimbangkan laju dari iterasi pelatihan agar proses nya tidak terlalu lambat. Bobot hasil pelatihan terbaik rancangan penjadwalan JST disimpan untuk proses pengujian data agar didapatkan hasil uji yang baik juga.

2.2 Raspberry Pi

Raspberry Pi, sering juga disingkat dengan nama *Raspi*, adalah komputer papan tunggal (*Single board Circuit/SBC*) yang memiliki ukuran sebesar kartu kredit. *Raspberry Pi* bisa digunakan untuk berbagai keperluan, seperti *spreadsheet*, *game*, bahkan bisa digunakan sebagai *media player* karena kemampuannya dalam memutar video *high definition*. *Raspberry Pi* dikembangkan oleh yayasan nirlaba, *Raspberry Pi Foundation* yang digawangi sejumlah *developer* dan ahli komputer dari Universitas Cambridge, Inggris.

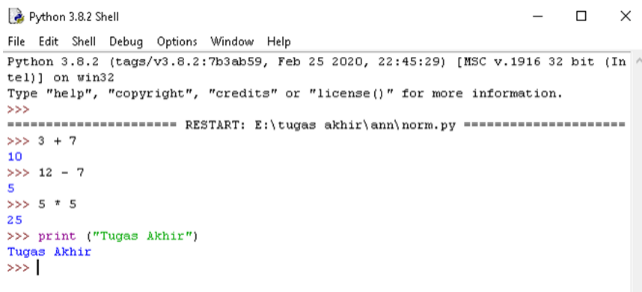
Raspberry Pi memiliki dua model yaitu model A dan model B. Secara umum *Raspberry Pi* Model B, 512MB RAM. Perbedaan model A dan B terletak pada memori yang digunakan, Model A menggunakan memori 256 MB dan model B 512 MB. Selain itu model B juga sudah dilengkapi dengan *ethernet port* (kartu jaringan) yang tidak terdapat di model A. Desain *Raspberry Pi* didasarkan seputar SoC (*System-on-a-chip*) *Broadcom BCM2835*, yang telah menanamkan prosesor ARM1176JZF-S dengan 700 MHz, *VideoCore IV GPU*, dan 256 *Megabyte RAM* [5]. Penyimpanan data didesain tidak untuk menggunakan hard disk atau *solid state drive*, melainkan mengandalkan kartu SD (*SD memory card*) untuk *booting* dan penyimpanan jangka panjang.



Gambar 2. 2. 1 *Raspberry Pi* Model B beserta *Pin Out*

2.3 Bahasa Pemrograman Python

Bahasa pemrograman *Python* ini pertama kali dibuat oleh Guido van Rossum pada awal tahun 1990 di negeri Belanda sebagai pengganti bahasa pemrograman yang disebut ABC. Walaupun Guido adalah orang yang pertama kali menciptakan bahasa pemrograman ini, tetapi bahasa pemrograman *Python* yang digunakan sekarang merupakan kontribusi dari berbagai sumber [6]. Bahasa pemrograman *Python* merupakan bahasa pemrograman yang dapat dikembangkan oleh siapa saja karena bersifat *Open Source* atau dengan kata lain bahasa pemrograman ini gratis, dapat digunakan tanpa *lisensi*, dan dapat dikembangkan semampu yang dapat dilakukan. Sebenarnya bahasa pemrograman *Python* ini mudah dipelajari karena penulisan *sintaks* yang lebih *fleksibel*. Selain itu, bahasa pemrograman *Python* ini memiliki *efisiensi* tinggi untuk struktur data level tinggi, pemrograman berorientasi objek lebih sederhana tetapi efektif, dapat bekerja pada *multi platform*, dan dapat digabungkan dengan bahasa pemrograman lain untuk menghasilkan aplikasi yang diinginkan. *Python* dikenal sebagai bahasa pemrograman interpreter, karena *Python* dieksekusi dengan sebuah *interpreter*. Terdapat dua cara untuk menggunakan *interpreter*, yaitu dengan mode baris perintah dan *modus script*. Pada mode baris, perintah diketikkan pada *shell* atau *command line* dan *Python* langsung menampilkan hasilnya. Bila menggunakan *shell*, semua definisi yang telah dibuat baik fungsi atau variabel akan dihapus. Cara lain adalah dengan menyimpan perintah – perintah *python* dalam satu *file*, yang disebut selanjutnya sebagai *script*. Kita dapat mengetikkan perintah-perintah *Python* dengan menggunakan *text editor* seperti *Notepad*. Lalu menyimpannya dengan akhiran ".py". kemudian menjalankannya dengan *Python* [7]. Pada gambar 2.3.1 ditunjukkan tampilan *Integrated DeveLopment Enviroment* (IDLE). *Prompt >>>* menyatakan *interpreter Python* siap menerima perintah dari pemakai.



Gambar 2. 3. 1 Tampilan IDLE Python pada Windows XP

2.4 Database Server

Database server adalah program komputer yang menyediakan layanan data lainnya ke komputer atau program komputer, seperti yang ditetapkan oleh model *client-server*. Istilah ini juga merujuk kepada sebuah komputer yang didedikasikan untuk menjalankan program server database. Database sistem manajemen database yang sering menyediakan fungsi server, dan beberapa DBMSs (misalnya, MySQL) secara *eksklusif* bergantung pada model *client-server* untuk akses data.

Client-server model dapat diartikan sebagai model dari suatu sistem yang membagi proses sistem antara server yang mengolah database dan *client* yang menjalankan *aplikasi*. Database server mengurangi beban akses data oleh *client* pada server. Database dapat diakses oleh beberapa *client* secara bersamaan dimana data yang diakses hanya atau diubah berasal dari satu sumber yaitu database pada server. Server tersebut diakses baik melalui suatu *front end* yang berjalan di komputer pengguna yang menampilkan data yang diminta atau *back end* yang berjalan pada server dan menangani tugas-tugas seperti analisis data dan penyimpanan [10].

Dalam model *master-slave*, database server *master* adalah lokasi pusat dan utama data sementara database server pembantu disinkronisasi *backup* dari master bertindak sebagai *proxy*. Beberapa contoh dari server basis data *Oracle*, *DB2*, *Informix*, *Ingres*, *SQL Server*. Setiap server menggunakan *query* sendiri *logika* dan struktur. Bahasa *query* SQL kurang lebih sama di semua server database.

2.5 Crontab

Cron adalah *daemon* penjadwalan yang menjalankan tugas dengan interval yang ditentukan. Tugas-tugas ini disebut *cron jobs* dan sebagian besar digunakan untuk mengotomatisasi pemeliharaan atau administrasi sistem. Kegunaan utama dari cron yaitu menjadwalkan *cron jobs*

untuk dijalankan berdasarkan menit, jam, hari dalam bulan, tahun, hari dalam seminggu atau kombinasi dari semuanya.

Crontab (***cron table***) adalah file teks yang menentukan jadwal *cron jobs*. Ada dua jenis file crontab. File crontab seluruh sistem dan file crontab pengguna individu. File crontab pengguna disimpan dengan nama pengguna dan lokasinya bervariasi menurut sistem operasi. Dalam sistem berbasis *Red Hat* seperti *CentOS*, file crontab disimpan di direktori */var/spool/cron* sementara pada file *Debian* dan *Ubuntu* disimpan di direktori *var/spool/cron/crontabs*. Mengedit crontab dengan menggunakan perintah crontab melalui baris perintah. */etc/crontab* dan file di dalam direktori */etc/cron.d* adalah file crontab di seluruh sistem yang hanya dapat diedit oleh *administrator* sistem. Didalam *linux* peletakan *script* di dalam direktori */etc/cron{hourly,daily,weekly,monthly}* dan *script* akan dieksekusi setiap jam/harian/mingguan/bulanan. [11]

Lima bidang pertama dapat berisi satu atau lebih nilai, dipisahkan oleh koma atau rentang nilai yang dipisahkan oleh tanda hubung.

- * – Operator tanda bintang berarti nilai apa pun atau selalu. Jika memiliki simbol tanda bintang di bidang Jam, itu berarti tugas dilakukan setiap jam.
- , – Operator koma memungkinkan menentukan daftar nilai untuk pengulangan. Misalnya, jika memiliki 1,3,5 di bidang Jam, tugas akan berjalan pada jam 1 pagi, 3 pagi dan 5 pagi (format jam 24 jam).
- - Operator tanda hubung memungkinkan untuk menentukan rentang nilai. Jika memiliki 1-5 di bidang *Day of week*, tugas akan berjalan setiap hari kerja (Dari Senin hingga Jumat).
- / - Operasi garis miring memungkinkan Anda untuk menentukan nilai yang akan diulang selama *interval* tertentu di antara mereka. Misalnya, jika memiliki * / 4 di bidang Jam, maka tindakan akan dilakukan setiap empat jam. Ini sama dengan menentukan 0,4,8,12,16,20. Alih-alih tanda bintang sebelum operator garis miring, dapat menggunakan rentang nilai, 1-30/10 artinya sama dengan 1,11,21.

III. METODE PENELITIAN

Dalam merancang sistem alat akan dilakukan dengan beberapa alur tahapan-tahapan prosesnya. Pada gambar

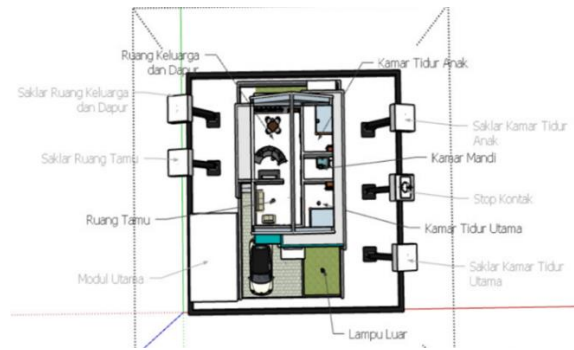
3.1.1 berikut merupakan proses tahapan rancangan yang dibuat.



Gambar 3. 1. 1 Alur Tahapan Perancangan Penelitian

3.2 Desain Perancangan Prototipe Rumah Susi Cantix

Susi Cantix merupakan singkatan dari Smarthome Upgrade System Control and Automatic Xtra. Dalam Susi Cantix terdapat sebuah sistem otomatis dalam pengoperasian lampu dengan cara membentuk sebuah pola pemakaian lampu dari kebiasaan penghuni rumah. Sehingga dalam hal ini penulis merancang sebuah desain miniatur rumah, dapat dilihat pada Gambar 3.2.1. Miniatur rumah yang dibuat dengan skala 1 : 15 dan purwarupa rumah seperti tipe 65. Didalam miniatur rumah terdapat lampu yang nantinya sebagai acuan apakah lampu yang dikontrol menggunakan sistem pola penjadwalan ANN memberikan aksi yang sesuai dengan input yang telah diolah.

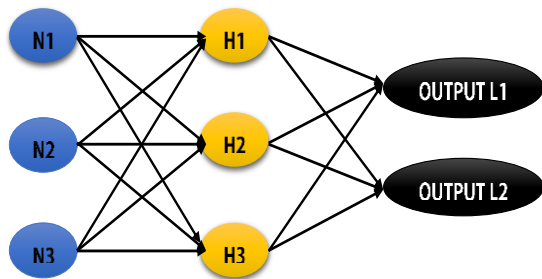


Gambar 3. 2. 1 Tampilan Tampak Atas Ruangan Miniatur Rumah Susi Cantix

3.3 Perancangan Fitur Pola Penjadwalan

Sistem pola penjadwalan merupakan salah satu fitur cerdas dari susi cantix, dimana ketika diaktifkan nya fitur ini keadaan lampu rumah yang dikontrol dengan sistem pola penjadwalan akan beroperasi sesuai pemakaian lampu pengguna sehari-hari. Dalam pembentukan sebuah pola dibutuhkan beberapa bagian *source code* dengan fungsi yang berbeda-beda. Bagian-bagian *source code* tersebut dibuat dengan menggunakan bahasa pemrograman *python* dan bekerja dalam satu perintah *running script*, dapat dilihat *flowchart* sistem kerja alat pada gambar 3.4.2 dibawah.

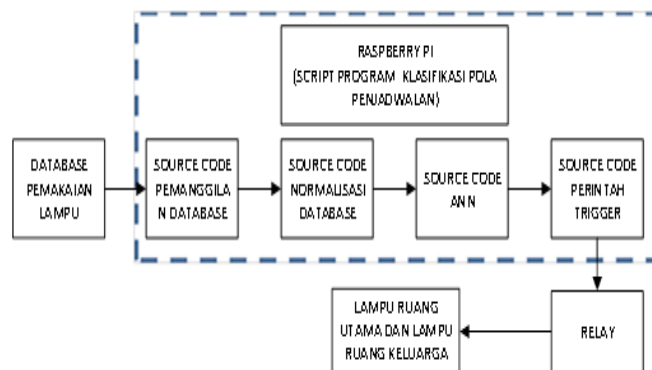
Diawali dengan bagian *source code* untuk mengambil data *input* dari server database. Data yang diambil sebagai *input* yaitu, tanggal, jam dan id lokasi lampu lalu selanjutnya data *input* tersebut dinormalisasi dengan *source code* normalisasi yang telah dibuat. Setelah data input dinormalisasi kumpulan data *input* tersebut akan diolah dengan *source code neural network*, dimana bagian ini merupakan inti dalam pembentukan pola penjadwalan. Dalam *source code neural network* arsitektur nya terdiri dari 3 lapisan, dapat dilihat pada gambar 3.3.1. Lapisan pertama adalah *input*, lapisan kedua adalah *hidden layer* dan lapisan ketiga adalah *output*. Setelah selesai melakukan *iterasi* didapatkan dua hasil *output* yang berbeda karena terdapat dua lokasi pengontrolan, yaitu untuk mengirim *trigger* ke relay lampu ruang tamu dan relay lampu ruang keluarga . Dapat dilihat alur bagian masing-masing *source code* yang dibuat pada gambar 3.3.2.



Gambar 3. 3. 1 Arsitektur Jaringan Saraf Tiruan

Keterangan:

- N1 = Input 1/ data input hari
 N2 = Input 2/ data input jam
 N3 = Input3/ data input id_lampu
 H1,H2,H3 = Proses pengolahan data input
 Output1 = Relay pada lampu ruang tamu
 Output2 = Relay pada lampu ruang keluarga

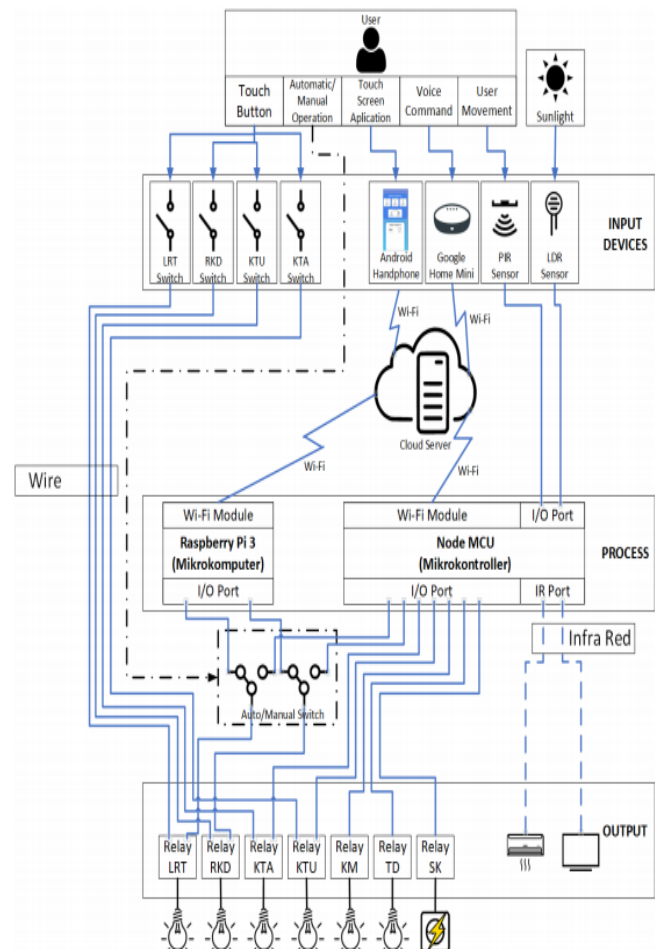


Gambar 3. 3. 2 Alur Script Masing-Masing Program

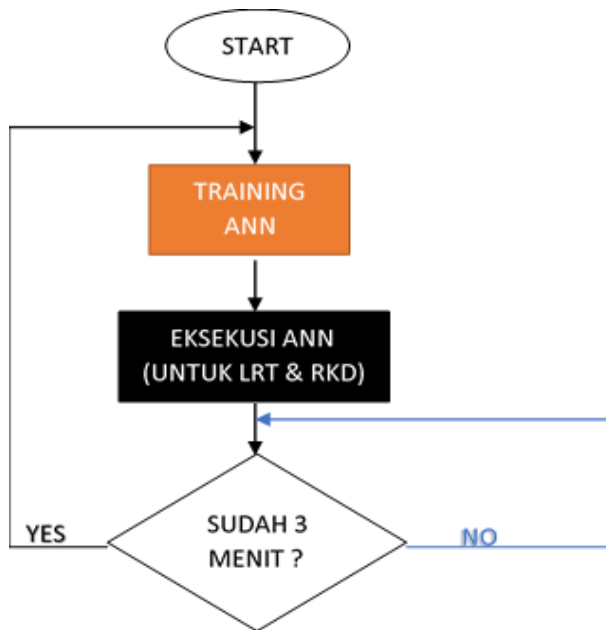
3.4 Arsitektur Sistem SUSICANTIK dan Flowchart Sistem Program Utama Raspberry Pi

Sistem pola penjadwalan merupakan salah satu bagian fitur cerdas dari susi cantix. Dapat dilihat pada gambar 3.4.1 arsitektur keseluruhan sistem susi cantix, pada flowchart tersebut terdapat banyak pilihan pengontrolan baik *online*, *offline* dan otomatis. Ketika fitur pola penjadwalan diaktifkan maka fitur *online* lainnya akan tidak dapat digunakan atau dioperasikan, hanya fitur *offline* saja yang dapat beroperasi ketika fitur otomatisasi pola penjadwalan aktif. Pada gambar 3.4.2 merupakan *flowchart* Program utama *Raspberry Pi* dengan sistem pola penjadwalan dan melakukan training data setiap 3 menit sekali. Terdapat juga *flowchart subprogram training* ANN dan *flowchart subprogram* eksekusi relay, dimana kedua *flowchart* ini saling

berhubungan dan bekerja secara berurutan sesuai dengan *flowchart* program utama *Raspberry Pi*. Untuk mengaktifkan atau mematikan pengoperasian lampu dengan sistem pola penjadwalan ANN dapat menggunakan saklar, sehingga ketika pengguna telah mematikan fitur sistem ini maka lampu akan kembali normal dan dapat dioperasikan kembali dengan fitur *online*..



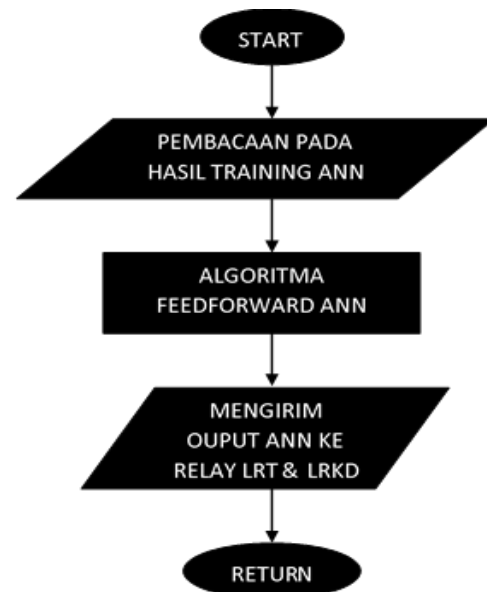
Gambar 3. 4. 1 Arsitektur Keseluruhan Sistem SUSICANTIX



Gambar 3. 4. 2 Flowchart Program Utama Raspberry Pi



Gambar 3. 4. 3 Flowchart Subprogram Training ANN



Gambar 3. 4. 4 Flowchart Subprogram Eksekusi ANN

3.5 Desain Alat

Alat yang dibuat akan diimplementasikan dalam sebuah miniatur rumah dengan skala 1:15, didalam miniatur rumah tersebut terdapat satu ruang tamu, satu ruang keluarga yang tergabung dengan dapur dan dua buah kamar yaitu kamar tidur utama dan kamar tidur anak. Dalam pengimplementasian alat dengan fitur sistem pola penjadwalan tidak semua lampu akan dikontrol dengan fitur ini melainkan hanya dua lokasi lampu utama saja yang akan dikontrol yaitu ruang tamu dan ruang keluarga. Komponen yang digunakan dalam proses implementasi merupakan komponen dengan skala 1:1 atau real sehingga apabila pengguna ingin memindahkan perangkatnya kedalam aplikasi nyata sangat mungkin dilakukan.

Untuk pengimplementasian alat dengan fitur pola penjadwalan ini dapat diaktifkan pengguna dengan menggunakan saklar yang ada pada mesin utama, seperti pada gambar 3.5.1. Sehingga ketika fitur otomatisasi pola penjadwalan telah aktif fitur *online* yang lainnya tidak dapat digunakan selain fitur *offline* saja yang dapat dioperasikan. Untuk hasil pengimplementasian alat tersebut dapat dilihat pada bab berikutnya.



Gambar 3. 5. 1 Saklar Fitur Otomatisasi ANN

IV. HASIL IMPLEMENTASI DAN PENGUJIAN

4.1 Hasil Implementasi

Berikut ini adalah hasil implementasi dari miniatur rumah SUSI CANTIX dan *smartbox* dengan bahan *akrilik* yang sudah dijelaskan pada bagian perancangan.



Gambar 4. 1 Tampak Atas Smartbox dan Miniatur Rumah Susi Cantix



Gambar 4. 2 Tampak Samping Smartbox dan Miniatur Rumah Susi Cantix

4.2 Pengujian Normalisasi Data Input Dari Database

Pengujian normalisasi data *input* dari database dilakukan untuk mengetahui *galat* dari ketiga *input* dengan membandingkan masing-masing hasilnya. Ketiga *input* tersebut yaitu, hari, jam dan id_lampu, dimana pada *input* hari digunakan deklarasi kamus *khusus*, seperti (Minggu= 0, Senin= 1, Selasa= 2, Rabu = 3, Kamis = 4, dst). Dalam pengujian normalisasi metode yang digunakan yaitu normalisasi *min-max*. Cara yang dilakukan untuk pengujian adalah dengan menghitung menggunakan penurunan rumus di Mc. Excel dan *running script* program lalu melihat hasilnya dan mencatat pada tabel. Namun sebelum melakukan pengujian dengan *running script*, syarat pengujian yang

harus terpenuhi adalah tersambung dengan database dan juga tersambung jaringan *wi-fi* Dapat dilihat hasil normalisasi pada tabel 4.2.1 berikut.

Tabel 4. 2. 1 Hasil Pengujian Normalisasi Data Input Dari Database

DATA SEBELUM DINORMALISASI			DATA SETELAH DINORMALISASI MENGGUNAKAN RUNNING SCRIPT PADA PYTHON		
HARI	JAM	ID	HARI	JAM	ID
3	715	1	0.3333333333333333	0.30279898218829515	0
3	745	2	0.3333333333333333	0.3155216284987277	1
3	1700	1	0.3333333333333333	0.7205258693808312	0
3	1745	2	0.3333333333333333	0.73960983884648	1
4	630	1	0.5	0.26675148430873624	0
4	710	2	0.5	0.3006785411365564	1
4	1750	2	0.5	0.7417302798982188	1
4	1805	1	0.5	0.7650551314673452	0
5	700	1	0.6666666666666666	0.2964376590330789	0
5	700	2	0.6666666666666666	0.2964376590330789	1
5	1700	1	0.6666666666666666	0.7205258693808312	0
5	1700	2	0.6666666666666666	0.7205258693808312	1
6	725	1	0.8333333333333334	0.3070398642917727	0
6	725	2	0.8333333333333334	0.3070398642917727	1
6	1725	1	0.8333333333333334	0.731128074639525	0
6	1725	2	0.8333333333333334	0.731128074639525	1
0	721	1	1	0.3053435114503817	0
0	724	2	1	0.30661577608142493	1
0	1734	1	1	0.7349448685326548	0
0	1805	2	1	0.7650551314673452	1
1	801	1	0	0.33927056827820185	0
1	824	2	0	0.34902459711620015	1
1	1754	1	0	0.7434266327396098	0
1	1805	2	0	0.7650551314673452	1
2	624	2	0.16666666666666666	0.2642069550466497	1
2	640	1	0.16666666666666666	0.27099236641221375	0

0.8333333	0.731128	0	7.5×10^{-7}	0.000075
0.8333333	0.731128	1	7.5×10^{-7}	0.000075
1	0.305344	0	4.8×10^{-7}	0.000048
1	0.306616	1	4.4×10^{-7}	0.000044
1	0.734945	0	1.4×10^{-7}	0.000014
1	0.765055	1	1.3×10^{-7}	0.000013
0	0.339271	0	4.3×10^{-7}	0.000043
0	0.349025	1	4×10^{-7}	0.00004
0	0.743427	0	3.6×10^{-7}	0.000036
0	0.765055	1	1.3×10^{-7}	0.000013
0.1666667	0.264207	1	4.4×10^{-7}	0.000044
0.1666667	0.270992	0	3.6×10^{-7}	0.000036
0.1666667	0.722222	0	2.2×10^{-7}	0.000022
0.1666667	0.722646	1	3.1×10^{-7}	0.000031
0.3333333	0.304919	1	3.4×10^{-7}	0.000034
0.3333333	0.30916	0	3×10^{-7}	0.00003
0.3333333	0.763783	0	1.3×10^{-7}	0.000013
0.3333333	0.765055	1	1.3×10^{-7}	0.000013
0.5	0.311281	0	2.5×10^{-7}	0.000025
0.5	0.315522	1	3.7×10^{-7}	0.000037
0.5	0.701442	0	1×10^{-7}	0.00001
0.5	0.726887	1	1.9×10^{-7}	0.000019
0.6666667	0.340967	0	7.8×10^{-7}	0.000078
0.6666667	0.341815	1	9.7×10^{-7}	0.000097
0.6666667	0.74173	1	2.7×10^{-7}	0.000027
0.6666667	0.745547	0	0.7×10^{-7}	0.000007
0.8333333	0.347328	0	2.4×10^{-7}	0.000024
0.8333333	0.353689	1	5.6×10^{-7}	0.000056
0.8333333	0.769296	0	1.3×10^{-7}	0.000013
0.8333333	0.770568	1	2.7×10^{-7}	0.000027
1	0.264631	0	0.4×10^{-7}	0.000004
1	0.273113	1	1.9×10^{-7}	0.000019
1	0.771416	0	4.5×10^{-7}	0.000045
1	0.782019	1	3.4×10^{-7}	0.000034

[1.0, 0.3333333333333333, 0.30279898210829515, 0.0], [0.3333333333333333, 0.7205258690809312, 0.0], [0.5, 0.2667514940873624, 0.0], [0.5, 0.76503551314673452, 0.0], [0.6666666666666666, 0.2964765930370789, 0.0], [0.6666666666666666, 0.7205258690809312, 0.0], [0.8333333333333334, 0.307039846219727, 0.0], [0.8333333333333334, 0.71128074395825, 0.0], [1.0, 0.30534514503817, 0.0], [1.0, 0.7349440685326549, 0.0], [0.0, 0.33927056827820185, 0.0], [0.0, 0.7434266327936098, 0.0], [0.16666666666666666, 0.2709923614221375, 0.0], [0.16666666666666666, 0.7222222222222222, 0.0], [0.3333333333333333, 0.3016030505451147, 0.0], [0.3333333333333333, 0.76782686683019, 0.0], [0.5, 0.3112807463952502, 0.0], [0.5, 0.7014418999151824, 0.0], [0.6666666666666666, 0.34096692111959287, 0.0], [0.6666666666666666, 0.745547073791485, 0.0], [0.8333333333333334, 0.347238242748092, 0.0], [0.8333333333333334, 0.7692960135708228, 0.0], [1.0, 0.26463104325697943, 0.0], [1.0, 0.771416453226515, 0.0]]

[0.333333333333333, 0.315521628497277, 1.0], [0.333333333333333, 0.73960983884648, 1.0], [0.5, 0.3008785413165564, 1.0], [0.5, 0.741730279898218, 1.0], [0.666666666666666, 0.2964376590330789, 1.0], [0.666666666666666, 0.720523869380832, 1.0], [0.833333333333334, 0.307039864291727, 1.0], [0.833333333333334, 0.731128074639525, 1.0], [1.0, 0.3061657768014249, 1.0], [1.0, 0.765055131467452, 1.0], [0.0, 0.34902459711620015, 1.0], [0.0, 0.765055131467452, 2.0], [0.166666666666666, 0.2642069550466497, 1.0], [0.166666666666666, 0.72264631043257, 1.0], [0.333333333333333, 0.304919423400339, 1.0], [0.333333333333333, 0.765055131467452, 1.0], [0.5, 0.315521628497277, 1.0], [0.5, 0.7268871923560475, 1.0], [0.666666666666666, 0.3418150975402884, 1.0], [0.666666666666666, 0.741730279898218, 1.0], [0.833333333333333, 0.345369565743002544, 1.0], [0.833333333333334, 0.770568278201866, 1.0], [1.0, 0.2731128074639525, 1.0], [1.0, 0.7820186598812553, 1.0]]

J. Sci. Appl. Tech. vol. 1 | 9

4.3 Pengujian Training Data Pola Penjadwalan

Setelah mendapatkan nilai input yang telah dinormalisasi lalu dapat dilakukan pengujian *training* data pola penjadwalan dengan jaringan saraf tiruan. Pada pengujian ini yang menjadi acuan adalah saat kondisi lampu sedang menyala atau tidak, dengan menggunakan waktu real dan pengujian dilakukan dalam beberapa kali. Setelah didapatkan hasil dari tiap pengujian selanjutnya akan dibandingkan dengan data pemakaian lampu yang digunakan sebagai input. Hal ini dilakukan untuk mengetahui seberapa besar perbedaan *galat* antara kondisi pemakaian pengguna dengan kondisi ketika sistem pola penjadwalan di aktifkan. Dapat dilihat hasil kondisi lampu sistem pola penjadwalan ketika aktif pada gambar 4.3.3 dan gambar 4.4.4.

Tabel 4. 3. 1 Kondisi Lampu Ketika Data Input Sama dari Pemakaian Pengguna

ID_LAMPU	HARI	JAM	KONDISI LAMPU
1	3 (RABU)	715	0
2	3 (RABU)	745	0
1	3 (RABU)	1700	1
2	3 (RABU)	1745	1
2	4 (KAMIS)	630	0
1	4 (KAMIS)	710	0
2	4 (KAMIS)	1750	1
1	4 (KAMIS)	1805	1
1	5 (JUM'AT)	700	0
2	5 (JUM'AT)	700	0
1	5 (JUM'AT)	1700	1
2	5 (JUM'AT)	1700	1
1	6 (SABTU)	725	0
2	6 (SABTU)	725	0
1	6 (SABTU)	1725	1
2	6 (SABTU)	1725	1
1	0 (MINGGU)	721	0
2	0 (MINGGU)	724	0
1	0 (MINGGU)	1734	1
2	0 (MINGGU)	1805	1
1	1 (SENIN)	801	0
2	1 (SENIN)	824	0
1	1 (SENIN)	1754	1
2	1 (SENIN)	1805	1
2	2 (SELASA)	624	0
1	2 (SELASA)	640	0
1	2 (SELASA)	1704	1
2	2 (SELASA)	1705	1
2	3 (RABU)	720	0
1	3 (RABU)	730	0
1	3 (RABU)	1802	1
2	3 (RABU)	1805	1

Tabel 4. 3. 2 Kondisi Lampu Ketika Data Input Berbeda dari Pemakaian Pengguna

ID_LAMPU	HARI	JAM	KONDISI LAMPU
1	3 (RABU)	715	0

2	3 (RABU)	745	1
1	3 (RABU)	1700	1
2	3 (RABU)	1745	0
2	4 (KAMIS)	630	0
1	4 (KAMIS)	710	1
1	4 (KAMIS)	1750	1
2	4 (KAMIS)	1805	0
1	5 (JUM'AT)	700	0
2	5 (JUM'AT)	700	1
1	5 (JUM'AT)	1700	1
2	5 (JUM'AT)	1700	0
1	6 (SABTU)	725	0
2	6 (SABTU)	725	1
1	6 (SABTU)	1725	1
2	6 (SABTU)	1725	0
1	0 (MINGGU)	721	0
2	0 (MINGGU)	724	1
1	0 (MINGGU)	1734	1
2	0 (MINGGU)	1805	0
1	1 (SENIN)	801	0
2	1 (SENIN)	824	1
1	1 (SENIN)	1754	1
2	1 (SENIN)	1805	0
1	2 (SELASA)	624	0
2	2 (SELASA)	640	1
1	2 (SELASA)	1704	1
2	2 (SELASA)	1705	0
1	3 (RABU)	720	0
2	3 (RABU)	730	1
1	3 (RABU)	1802	1
2	3 (RABU)	1805	0

Tabel 4. 3. 3 Hasil Uji Kondisi Lampu Ketika Sama dari Sistem Pola Penjadwalan aktif

ID_LAMPU	HARI	JAM	KONDISI LAMPU
1 (LRT)	4 (kamis)	1835	0.96223860968 (ON)
2 (LRKD)	4 (kamis)	1835	0.99742400633 (ON)
1 (LRT)	4 (kamis)	625	0.33432974203 (OFF)
2 (LRKD)	4 (kamis)	625	0.00163066812 (OFF)

Tabel 4. 3. 4 Hasil Uji Kondisi Lampu Ketika Berbeda dari Sistem Pola Penjadwalan aktif

ID_LAMPU	HARI	JAM	KONDISI LAMPU
1 (LRT)	4 (kamis)	632	0.33940416770 (OFF)
2 (LRKD)	4 (kamis)	632	0.99824257471 (ON)
1 (LRT)	4 (kamis)	1827	0.96116293495 (ON)
2 (LRKD)	4 (kamis)	1827	0.00280339996 (OFF)

```

Data Output Baru:
hari:
4
jam:
1835
kondisi lampu ruang tamu saat ini:
0.9622386096865043
bobot pada Training akhir:
[[-5.42729137]
 [ 7.65208412]
 [-0.99977125]]
Ln: 64 Col: 130
Ln: 130 Col: 1
IND 18.36 04/06/2020

Data Output Baru:
hari:
4
jam:
1835
kondisi lampu ruang keluarga saat ini:
0.9974240063348365
bobot pada Training akhir:
[[-0.2553594]
 [ 24.11800219]
 [-12.67182584]]
Ln: 64 Col: 130
Ln: 130 Col: 1
IND 18.35 04/06/2020

```

Gambar 4. 3. 1 Hasil Sistem Pola Penjadwalan Dengan Kondisi Input Sama Jam 18.35

```

Data Output Baru:
hari:
4
jam:
625
kondisi lampu ruang tamu saat ini:
0.3343297420340642
bobot pada Training akhir:
[[-5.42729137]
 [ 7.65208412]
 [-0.99977125]]
Ln: 44 Col: 4
Ln: 44 Col: 4
IND 06.27 04/06/2020

Data Output Baru:
hari:
4
jam:
625
kondisi lampu ruang keluarga saat ini:
0.0016306681245113475
bobot pada Training akhir:
[[-0.2553594]
 [ 24.11800219]
 [-12.67182584]]
Ln: 44 Col: 4
Ln: 44 Col: 4
IND 06.25 04/06/2020

```

Gambar 4. 3. 2 Hasil Sistem Pola Penjadwalan Dengan Kondisi Input Sama pada Jam 06.25

```

Data Output Baru:
hari:
4
jam:
632
kondisi lampu ruang tamu saat ini:
0.3394041677084566
bobot pada Training akhir:
[[-5.42729137]
 [ 7.65208412]
 [-0.99977125]]
Ln: 48 Col: 4
Ln: 48 Col: 4
IND 06.34 04/06/2020

Data Output Baru:
hari:
4
jam:
632
kondisi lampu ruang keluarga saat ini:
0.9982425747197468
bobot pada Training akhir:
[[ 0.255391]
 [-24.10539292]
 [ 12.66504595]]
Ln: 48 Col: 4
Ln: 48 Col: 4
IND 06.33 04/06/2020

```

Gambar 4. 3. 3 Hasil Sistem Pola Penjadwalan Dengan Kondisi Input Berbeda pada Jam 06.32

```

Data Output Baru:
hari:
4
jam:
1826
kondisi lampu ruang tamu saat ini:
0.9611629349572364
Ln: 576 Col: 1
Ln: 576 Col: 1
IND 18.27 04/06/2020

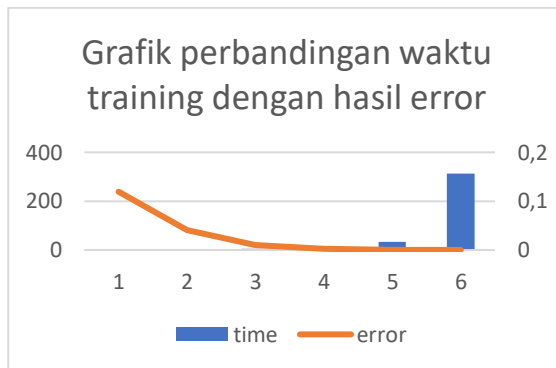
Data Output Baru:
hari:
4
jam:
1827
kondisi lampu ruang keluarga saat ini:
0.002803399964486953
Ln: 610 Col: 1
Ln: 610 Col: 1
IND 18.28 04/06/2020

```

Gambar 4. 3. 4 Hasil Sistem Pola Penjadwalan Dengan Kondisi Input Berbeda pada Jam 18.27

Tabel 4. 3. 5 Hasil Perbandingan Data Training

training	Time (s)	error
15	2,93	0,11936
150	3,24	0,0411
1500	4,63	0,010492
15000	7,15	0,002576
150000	33,25	0,000626
1500000	313,17	0,000151

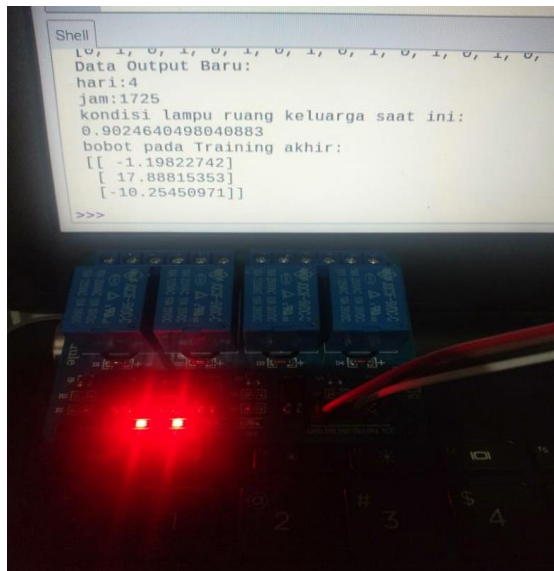


Gambar 4. 3. 5 Grafik Perbandingan waktu training dengan hasil error

Pada tabel 4.3.5 dan pada gambar 4.3.5 didapatkan hasil perbandingan setiap training dengan jumlah iterasi yang berbeda. Ketika jumlah iterasi yang digunakan hanya 15 maka hasil error yang didapat cukup besar, jika menggunakan jumlah iterasi sebanyak 1500000 maka hasil error yang didapat sangat kecil selisih nya akan tetapi waktu yang diperlukan ketika training data akan sangat lama sekitar 5-6 menit. Sehingga setelah didapatkan hasil data perbandingan training tersebut jumlah iterasi yang digunakan adalah 15000 dengan selisih error yang didapat ideal dan waktu training tidak terlalu lama.

4.4 Pengujian Trigger Relay

Demi menunjang keberhasilan sistem, sistem pola penjadwalan harus memberikan sebuah *trigger* ke relay yang terdapat di *smartbox* susi cantix. Pengujian ini dilakukan untuk melihat aksi relay ketika mendapat *trigger* dari hasil *training* data. Untuk melakukan pengujian terlebih dahulu menghubungkan *port* dari *pin gpio raspberry* ke relay dan terhubung dengan koneksi *wi-fi*. Berikut hasil pengujian *trigger* relay dapat dilihat pada gambar 4.4.1 dan 4.4.2.



Gambar 4. 4.1 Hasil Pengujian Ketika *Output* Kondisi Lampu LRT dan LRK Lebih dari 0,5



Gambar 4. 4. 2 Hasil Pengujian Ketika *Output* Kondisi Lampu LRT dan LRK Kurang dari 0,5

Terlihat pada gambar 4.3.1 diatas bahwa ketika *output* kondisi lampu lebih dari 0,5 maka akan memberikan *trigger* untuk menyalakan lampu dan dapat dilihat lampu indikasi relay menyala. Sedangkan pada gambar 4.3.2, ketika *output* kondisi lampu kurang dari 0,5 maka memberikan *trigger* aksi untuk mematikan lampu dan dapat dilihat juga lampu indikasi relay mati

4.5 Pengujian Fitur Otomatisasi Pola Penjadwalan

Fitur otomatisasi pola penjadwalan akan melakukan *running script* program secara otomatis setiap 4 jam. Fitur otomatisasi pola penjadwalan juga akan memberikan target waktu sesuai dengan waktu real. Fitur ini akan memperlihatkan kondisi lampu rumah akan hidup dan mati sesuai dengan kebiasaan pengguna. Berikut adalah hasil kondisi lampu ketika fitur otomatisasi pola penjadwalan diaktifkan.



Gambar 4. 5. 1 Hasil Otomatisasi Pola Penjadwalan Diaktifkan (Jam 16.38 LRT dan LRK Hidup Bersamaan)



Gambar 4. 5. 2 Hasil Otomatisasi Pola Penjadwalan Diaktifkan (Jam 16.38 LRT dan LRK Mati Bersamaan)



Gambar 4. 5. 3 Hasil Otomatisasi Pola Penjadwalan Diaktifkan (Jam 16.23 LRT Hidup dan LRT Mati)



Gambar 4. 5. 4 Hasil Otomatisasi Pola Penjadwalan Diaktifkan (Jam 6.44 LRT Mati dan LRT Hidup)

Dapat dilihat dari gambar 4.5.1 sampai 4.5.4 bahwa kondisi lampu pada kedua ruangan yang dikontrol sudah dapat berjalan dengan baik dan sudah sesuai dengan waktu aslinya. Pada gambar 4.5.1 dan 4.5.2 merupakan kondisi lampu ketika input database pemakaian lampu dioperasikan bersamaan, lalu pada gambar 4.5.3 dan 4.5.4 merupakan kondisi lampu ketika input pada database pengoperasian lampunya berbeda. Sehingga ketika pengguna Susi Cantix mengaktifkan fitur pola penjadwalan maka keadaan kondisi lampu rumah sudah dapat otomatis berjalan sesuai dengan kebiasaan pemakaian lampu pengguna sendiri. Dapat disimpulkan fitur pola penjadwalan pada Susi Cantix berhasil diterapkan.

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah menyelesaikan proses perancangan, implementasi dan pengujian yang dilakukan pada alat dengan fitur sistem pola penjadwalan dengan menggunakan jaringan saraf tiruan pada rumah pintar Susi Cantix. Hasil-hasil yang dapat disimpulkan sebagai berikut.

1. Telah dapat dirancang dan dibangun sebuah alat dengan Fitur Sistem Otomatis Pola Penjadwalan yang dapat mengoperasikan lampu sesuai dengan kebiasaan pengguna.
2. Fitur otomatis pola penjadwalan ini akan bekerja secara *realtime* ketika diaktifkan. Fitur otomatis ini juga dapat meminimalisir kejahatan ketika berpergian yang cukup lama dan meminimalisir pembengkakan tagihan listrik. Selain itu, kedua kondisi lampu yang dikontrol sudah dapat berjalan dengan baik sesuai dengan kebiasaan pengguna dalam mengoperasikan lampu.

5.2 Saran

Berdasarkan pengimplementasian dan pengujian alat dengan fitur sistem otomatis pola penjadwalan, dapat diberikan beberapa saran sebagai berikut.

1. Alat ini masih kesulitan ketika mengolah data input yang tidak beraturan seperti perbedaan pengoperasian lampu hanya berbeda 1 jam, sehingga dibutuhkan data input dengan range minimal 6 jam.
2. Alat ini juga terdapat kekurangan pada monitoring, sehingga jika dikembangkan lebih lanjut dapat menambahkan layar *monitor mini* agar pengguna dapat melihat hasil dari pembentukan pola yang baru.

Sehingga dari beberapa saran di atas, diharapkan akan terwujud suatu alat dengan sistem yang *optimal* dan lebih sempurna.

DAFTAR PUSTAKA

- [1] M. Ngafifi, "Kemajuan Teknologi dan Pola Hidup Manusia dalam Perspektif Sosial Budaya," *Jurnal Pembangunan Pendidikan*, 2014.
- [2] Dinas Perencanaan Sistem PT PLN (Persero), "Proyeksi Kebutuhan Listrik per Sektor per Provinsi PLN dari Tahun 2003 s.d. Tahun 2013," *tirto.id*, Jakarta, 2004.
- [3] Polda Metro Jaya, "Jakarta, Kota dengan 9 Kasus Pencurian Setiap Hari," *Tirto.id*, 28 Maret 2019. [Online]. Available: <https://tirto.id/jakarta-kota-dengan-9-kasus-pencurian-setiap-hari-dko5>. [Diakses 01 Juni 2020].
- [4] S. Kusumadewi, "Membangun Jaringan Syaraf Tiruan," Yogyakarta, Graha Ilmu, 2004.
- [5] GitHub, "Raspberry Pi, Jenis dan Spesifikasinya," 13 Oktober 2019. [Online]. Available: <https://github.com/yogidm/Raspberry-Pi-Indonesia/blob/master/Raspberry%20Pi.md#jenis-dan-spesifikasinya>. [Diakses 01 Juni 2020].
- [6] Pythonindo, "Sejarah Python," 2019. [Online]. Available: [https://www.pythonindo.com/sejarah-python/#:~:text=Sejarah%20Python,terinspirasi%20dari%20bahasa%20pemrograman%20ABC.&text=Pada%20tahun%202001%2C%20dibentuklah%20Organisasi,Python%20Software%20Foundation%20\(PSF\)..](https://www.pythonindo.com/sejarah-python/#:~:text=Sejarah%20Python,terinspirasi%20dari%20bahasa%20pemrograman%20ABC.&text=Pada%20tahun%202001%2C%20dibentuklah%20Organisasi,Python%20Software%20Foundation%20(PSF)..) [Diakses 2020 Juni 02].
- [7] Python.org, "Referensi Bahasa Pemrograman Python," 25 November 2019. [Online]. Available: <https://docs.python.org/id/3.8/reference/index.html>. [Diakses 02 Juni 2020].
- [8] H. Khuluq, "Implementasi Voip (Voice Over Internet Protocol) Server Berbasis Raspberry Pi Sebagai Media Komunikasi," *Jurnal Pendidikan dan Informatika*, vol. 3, 2016.
- [9] LinuxID, "Penggunaan Cron job dan Crontab di Terminal," LinuxID, 2018. [Online]. Available: <https://www.linuxid.net/25323/tutorial-cron-job-dan-crontab-untuk-otomatisasi-tindakan/>. [Diakses 21 06 2020].