

RANCANG BANGUN SISTEM MONITORING KONSUMSI LISTRIK BERBASIS WEB

SKRIPSI

LABORATORIUM SISTEM KOMPUTER DAN ROBOTIKA

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer



Disusun Oleh:

NICKY PRABOWO

NIM. 105060801111069

**KEMENTERIAN RISET TEKNOLOGI DAN PENDIDIKAN TINGGI
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
PROGRAM STUDI INFORMATIKA / ILMU KOMPUTER**

UNIVERSITAS BRAWIJAYA

MALANG

2015

LEMBAR PERSETUJUAN

RANCANG BANGUN SISTEM MONITORING KONSUMSI LISTRIK BERBASIS WEB

SKRIPSI

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer



Disusun Oleh :

NICKY PRABOWO

NIM. 105060801111069

Telah diperiksa dan disetujui oleh :

Pembimbing I

Pembimbing II

Barlian Henryranu Prasetio, S.T., M.T.

NIK. 82102406110254

Issa Arwani, S.Kom.,M.Sc.

NIK. 198309222012121003

LEMBAR PENGESAHAN

RANCANG BANGUN SISTEM MONITORING KONSUMSI LISTRIK BERBASIS WEB

SKRIPSI

LABORATORIUM SISTEM KOMPUTER DAN ROBOTIKA

Untuk memenuhi sebagian persyaratan untuk mencapai gelar Sarjana Komputer

Disusun oleh:

NICKY PRABOWO
105060801111069

Skrripsi ini telah dinyatakan lulus pada tanggal 23 Juni 2015

Penguji I,

Penguji II,

Fajar Pradana, S.ST., M.Eng.
NIK. 87112116110371

Himawat Aryadita, ST., M.Sc.
NIP. 19801018 200801 1 003

Penguji III,

Gembong Edhi Setyawan, S.T., M.T.
NIK. 871121 16110371

Mengetahui,
Ketua Program Studi Informatika/Ilmu Komputer

Drs. Mardji, MT
NIP. 19670801 199203 1 001

PERNYATAAN ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah SKRIPSI ini tidak terdapat karya ilmiah yang pernah diajukan orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata di dalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 1 Juli 2015

Mahasiswa,

Nicky Prabowo

105060801111069

KATA PENGANTAR

Puji syukur Penulis panjatkan kehadiran Allah SWT karena hanya dengan rahmat dan karunia-Nya, penulis dapat menyelesaikan skripsi dengan judul “Rancang Bangun Sistem Monitoring Konsumsi Listrik berbasis Web”.

Terima kasih Penulis sampaikan kepada pihak-pihak yang telah membantu dalam penyelesaian skripsi ini. Pihak-pihak tersebut antara lain :

1. Orang tua Penulis, Antijo Arifin dan Lilik Sunarsih yang telah memberikan motivasi, kasih sayang serta dukungan moril dan materil kepada penulis. Adik Gebylia Kusumadewi yang telah memberikan semangat dari awal sampai akhir pengerjaan skripsi ini.
2. Bapak Ir.Sutrisno, MT., Drs. Mardji, M.T. dan Issa Arwani, S.Kom., M.Sc.selaku Ketua Program Teknologi Informasi dan Ilmu Komputer, Kepala Program Studi Ilmu Komputer/Informatika dan Sekretaris Program Studi Ilmu Komputer/Informatika serta segenap Bapak/Ibu Dosen, Staff Administrasi dan Perpustakaan Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
3. Bapak Barlian Henryranu Prasetio, S.T., M.T. dan Issa Arwani, S.Kom., M.Sc. selaku dosen pembimbing I dan dosen pembimbing II yang telah banyak memberikan pengarahan, ilmu serta saran dalam penyusunan skripsi ini.
4. Kawan-kawan di Villa Bukit Tidar E2-125 Mas Tono, Mas Reza, Mas Gaplek, dan Choirul Huda.
5. Seluruh rekan-rekan seperjuangan Affan, Dedi, Dewey, Rana, Satria, Fahmi, Aan, Idris, Tara, Novia, Arik, Anggi, Shindu dan seluruh sahabat-sahabat 2010 atas segala doa, bantuan, serta dukungan semangatnya kepada penulis selama menempuh studi di Ilmu Komputer/Informatika Universitas brawijaya.
6. Reka-rekan di tim futsal POWER Dina, Elliya, Yasmin, Choy, Ayu, Ana, Nyau, Icha, Esti, Rania, Suryadi, Opan, Halimah, Fita, dan rekan-rekan lainnya yang tidak mungkin disebutkan satu per satu.

7. Semua pihak yang tidak dapat penulis sebutkan satu per satu membantu baik secara langsung maupun tidak langsung demi terselesainya skripsi ini.

Pada penulisan skripsi ini masih banyak kekurangan. Oleh karena itu, penulis mengharap kritik dan saran yang bersifat membangun guna menyempurnakan skripsi ini. Semoga skripsi ini memberikan manfaat bagi penyusun maupun pihak lainnya.

Malang, 30 Juni 2015

Penulis



ABSTRAK

Nicky Prabowo. 2015 : Rancang Bangun Sistem Pemantau Konsumsi Listrik berbasis Web. Skripsi Program Studi Informatika/Ilmu Komputer, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya. Dosen Pembimbing : Barlian Henryranu Prasetio, S.T., M.T. dan Issa Arwani, S.Kom., M.Sc.

Untuk memantau konsumsi listrik dari berbagai peralatan elektronik yang ada di rumah dibutuhkan pemasangan meteran untuk setiap perangkat elektronik yang ada. Pendekatan ini kurang efisien secara materi karena dibutuhkan banyak meteran dan juga biaya yang mahal. Oleh karena itu muncul konsep *Non-Intrusive Load Monitoring* (NILM) yang menawarkan solusi yang lebih murah dan efisien secara materi dengan menambahkan penggunaan metode pengenalan pola. Sistem pemantau konsumsi listrik berbasis *web* ini merupakan langkah awal untuk mewujudkan konsep *Smart Home*. Kita dapat memantau aktifitas perangkat elektronik di suatu bangunan dari mana saja hanya dengan 1 sensor arus dan tegangan yang terhubung dengan jaringan internet.

Dari penelitian ini dilakukan pengujian keseluruhan sistem untuk mengetahui performa sistem dalam mendeteksi aktifitas nyala dan mati dari 4 perangkat elektronik yang meliputi lampu, laptop, monitor, dan setrika. Skenario 1 menghasilkan nilai *recall* dan *precision* sebesar yaitu 50%, skenario 2 menghasilkan nilai *recall* dan *precision* sebesar yaitu 50%, dan skenario 3 menghasilkan nilai *recall* 21.42% dan *precision* 30%. Berdasarkan pengujian sistem, didapatkan kesimpulan bahwa sistem masih membutuhkan penelitian lebih lanjut untuk meningkatkan performa sistem dalam mendeteksi aktifitas nyala dan mati dari perangkat elektronik.

Kata Kunci: Konsumsi Listrik, *Smart Home*, NILM

ABSTRACT

Nicky Prabowo. 2015 : The Blueprint of Web based Electricity Monitoring System. A Thesis of Computer/Informatic Study Program, Information Technology and Computer Program of Brawijaya University. Councillng Lecturer: Mr. Barlian Henryranu Prasetio, S.T., M.T. and Mr. Issa Arwani, S.Kom.,M.Sc.

Monitoring the electricity consumption of various electronic equipment at home required the installation of the meter to any existing electronic devices. This approach is less efficient and expensive. A web based system to monitor electricity consumption is the first step to realize the concept of Smart Home. Then Non-Intrusive Load Monitoring is introduced to solve this problem. This approach offer a more efficient and cheaper solution on electricity monitoring system with the addition of pattern recognition approach. We can monitor activity of any electrical devices at home via single current and voltage sensor which connected to the internet.

Then we perform a test to measure system performance to detect on and off activities of electrical appliances such as laptop, monitor, lamp, and iron. The value of recall and precision from the 1st scenario is 50%, the 2nd scenario also resulting 50% of recall and precision, and the value of recall and precision of the 3rd scenario is 21.42% and 30%. Based on the test result, the system still need a further development and research to increase the system performance in detecting on and off activities of electrical appliances.

Keywords: *Electricity Consumption, Smart Home, NILM*

DAFTAR ISI

| | |
|---|-----------|
| LEMBAR PERSETUJUAN | ii |
| LEMBAR PENGESAHAN..... | iii |
| PERNYATAAN ORISINALITAS SKRIPSI..... | iv |
| KATA PENGANTAR..... | v |
| ABSTRAK | vii |
| ABSTRACT | viii |
| DAFTAR ISI..... | ix |
| DAFTAR TABEL..... | xiii |
| DAFTAR GAMBAR..... | xv |
| DAFTAR KODE..... | xvii |
| 1 BAB I PENDAHULUAN..... | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Rumusan Masalah | 2 |
| 1.3 Batasan Masalah..... | 2 |
| 1.4 Tujuan..... | 3 |
| 1.5 Manfaat..... | 3 |
| 1.6 Sistematika penulisan | 3 |
| 2 BAB II KAJIAN PUSTAKA DAN DASAR TEORI..... | 5 |
| 2.1 Kajian Pustaka | 5 |
| 2.2 Dasar Teori | 6 |
| 2.2.1 <i>Smart Home</i> | 6 |
| 2.2.2 Mikrokontroler | 6 |
| 2.2.3 Arduino Uno R3..... | 7 |
| 2.2.4 Arduino Ethernet Shield R3..... | 8 |
| 2.2.5 Arduino IDE..... | 8 |
| 2.2.6 OpenEnergyMonitor | 9 |
| 2.2.7 MySQL..... | 9 |
| 2.2.8 <i>Current Transformer Sensor (CT Sensor)</i> | 10 |
| 2.2.9 <i>Modified K-Nearest Neighbor</i> | 10 |
| 3 BAB III METODOLOGI PENELITIAN | 12 |

| | | |
|---------|---|----|
| 3.1 | Analisis Permasalahan..... | 13 |
| 3.2 | Studi Pustaka | 13 |
| 3.3 | Identifikasi Kebutuhan Perangkat Keras (Hardware)..... | 13 |
| 3.4 | Identifikasi kebutuhan Perangkat Lunak (Software)..... | 13 |
| 3.5 | Identifikasi Kebutuhan Fungsional | 14 |
| 3.6 | Perancangan..... | 14 |
| 3.6.1 | Pengambilan Data | 14 |
| 3.6.2 | Pengolahan Data..... | 14 |
| 3.5 | Implementasi | 18 |
| 3.6 | Pengujian | 18 |
| 3.7 | Kesimpulan dan Saran..... | 18 |
| 4 | BAB IV PERANCANGAN DAN IMPLEMENTASI | 20 |
| 4.1 | Perancangan..... | 20 |
| 4.1.1 | Analisa Kebutuhan | 20 |
| 4.1.1.1 | Analisa Kebutuhan Perangkat Keras | 20 |
| 4.1.1.2 | Analisa Kebutuhan Perangkat Lunak | 20 |
| 4.1.1.3 | Analisis Kebutuhan Fungsional | 21 |
| 4.1.2 | Diagram Blok..... | 22 |
| 4.1.2.1 | Sensor Arus dan Tegangan | 22 |
| 4.1.2.2 | Arduino | 23 |
| 4.1.2.3 | Server | 23 |
| 4.1.2.4 | Database | 23 |
| 4.1.2.5 | Aplikasi Web | 23 |
| 4.1.3 | Diagram Alir Sistem | 23 |
| 4.1.4 | Diagram Konteks | 25 |
| 4.1.5 | Diagram Aliran Data | 25 |
| 4.1.6 | Perancangan Rangkaian Elektronik | 26 |
| 4.1.7 | Pemrograman Mikrokontroler..... | 27 |
| 4.1.8 | Perancangan Diagram <i>Use Case</i> | 29 |
| 4.1.8.1 | Skenario <i>Use Case</i> Melihat Total Konsumsi Listrik | 30 |
| 4.1.8.2 | Skenario <i>Use Case</i> Melihat Aktifitas Perangkat Elektronik.... | 31 |
| 4.1.8.3 | Skenario <i>Use Case</i> Mengubah Tarif Dasar Listrik dan Beban Daya Maksimum | 31 |

| | | |
|----------|---|----|
| 4.1.9 | Perancangan <i>Database</i> | 32 |
| 4.1.10 | Perancangan Proses <i>Preprocessing</i> | 36 |
| 4.1.10.1 | Proses <i>Data Smoothing</i> | 37 |
| 4.1.10.2 | Proses Deteksi Kenaikan dan Penurunan Daya | 38 |
| 4.1.10.3 | Proses Deteksi <i>Steady State</i> | 39 |
| 4.1.11 | Perancangan Proses Validasi <i>Data Training</i> | 42 |
| 4.1.11.1 | Proses Hitung <i>Euclidian Distance</i> | 43 |
| 4.1.11.2 | Proses Perankingan Hasil <i>Euclidian Distance</i> | 44 |
| 4.1.11.3 | Proses Perhitungan Nilai Validasi <i>Data Training</i> | 45 |
| 4.1.12 | Perancangan Klasifikasi MKNN..... | 46 |
| 4.1.12.1 | Proses Hitung <i>Euclidian Distance</i> | 48 |
| 4.1.12.2 | Proses Hitung Bobot Klasifikasi..... | 48 |
| 4.1.12.3 | Proses Mengurutkan Bobot Klasifikasi dan Pengambilan Kesimpulan | 49 |
| 4.2 | Implementasi | 50 |
| 4.2.1 | Implementasi Perangkat Keras..... | 50 |
| 4.2.2 | Implementasi Perangkat Lunak..... | 51 |
| 4.2.2.1 | Implementasi <i>Database</i> | 51 |
| 4.2.2.2 | Implementasi Pemrograman Mikrokontroler..... | 51 |
| 4.2.2.3 | Implementasi Program <i>Preprocessing</i> Pendeteksian Perangkat 54 | |
| 4.2.2.4 | Implementasi Program Validasi <i>Data Training</i> | 57 |
| 4.2.2.5 | Implementasi Program Klasifikasi MKNN | 60 |
| 5 | BAB V PENGUJIAN | 64 |
| 5.1 | Pengujian Komunikasi Data..... | 64 |
| 5.2 | Pengujian Sensor | 66 |
| 5.2.4 | Analisa Pengujian Sensor..... | 67 |
| 5.3 | Pengujian Algoritma <i>Modified K-Nearest Neighbor</i> | 67 |
| 5.3.4 | Analisa Pengujian Algoritma <i>Modified K-Nearest Neighbor</i> | 70 |
| 5.4 | Pengujian Fungsional | 70 |
| 5.4.1 | Kasus Uji Memantau Total Konsumsi listrik..... | 70 |
| 5.4.2 | Kasus Uji Memantau Aktivitas Perangkat Elektronik | 71 |

| | | |
|-------|--|----|
| 5.4.3 | Kasus Uji Mengubah Tarif Dasar Listrik dan Beban Daya Maksimal 72 | |
| 5.4.4 | Hasil Pengujian Fungsional..... | 73 |
| 5.5 | Pengujian Keseluruhan Sistem..... | 74 |
| 5.5.1 | Skenario Pengujian 1..... | 74 |
| 5.5.2 | Analisa Skenario Pengujian 1 | 75 |
| 5.5.3 | Skenario Pengujian 2..... | 75 |
| 5.5.4 | Analisa Skenario Pengujian 2 | 76 |
| 5.5.5 | Skenario Pengujian 3..... | 76 |
| 5.5.6 | Analisa Skenario Pengujian 3 | 77 |
| 6 | BAB VI PENUTUP..... | 78 |
| 6.1 | Kesimpulan..... | 78 |
| 6.2 | Saran..... | 79 |
| | DAFTAR PUSTAKA..... | 80 |



DAFTAR TABEL

| | |
|--|----|
| Tabel 2.1 Spesifikasi Arduino Uno R3 | 8 |
| Tabel 3.1 Data Training | 15 |
| Tabel 3.2 Perhitungan Euclidian distance | 15 |
| Tabel 3.3 Perankingan Euclidian distance | 16 |
| Tabel 3.4 Mengecek Kesesuaian Data | 16 |
| Tabel 3.5 Menghitung Nilai Validasi | 17 |
| Tabel 3.6 Data Uji | 17 |
| Tabel 3.7 Perhitungan Euclidian distance | 17 |
| Tabel 3.8 Perhitungan Bobot Klasifikasi | 18 |
| Tabel 3.9 Hasil Klasifikasi | 18 |
| Tabel 4.1 Kebutuhan Perangkat Keras | 20 |
| Tabel 4.2 Analisa Kebutuhan Perangkat Lunak | 21 |
| Tabel 4.3 Fitur Aplikasi Bidgely | 21 |
| Tabel 4.4 Analisis Kebutuhan Fungsional | 22 |
| Tabel 4.5 Spesifikasi Server | 23 |
| Tabel 4.6 Skenario Use Case Melihat Total Konsumsi Listrik | 30 |
| Tabel 4.7 Use Case Melihat Aktifitas Perangkat Elektronik | 31 |
| Tabel 4.8 Skenario Use Case Mengubah Tarif Dasar Listrik dan Beban Daya Maksimum | 32 |
| Tabel 4.9 Database User | 33 |
| Tabel 4.10 Database Konsumsi Listrik | 34 |
| Tabel 4.11 Database Validasi Data Training | 34 |
| Tabel 4.12 Database Laporan Konsumsi Listrik | 35 |
| Tabel 4.13 Database Data Training | 35 |
| Tabel 4.14 Database Perangkat Elektronik | 36 |
| Tabel 5.1 Daftar Alat untuk Diuji | 66 |
| Tabel 5.2 Pengujian Sensor Arus | 66 |
| Tabel 5.3 Pengujian Sensor Tegangan | 67 |
| Tabel 5.4 Urutan Aktivitas | 67 |

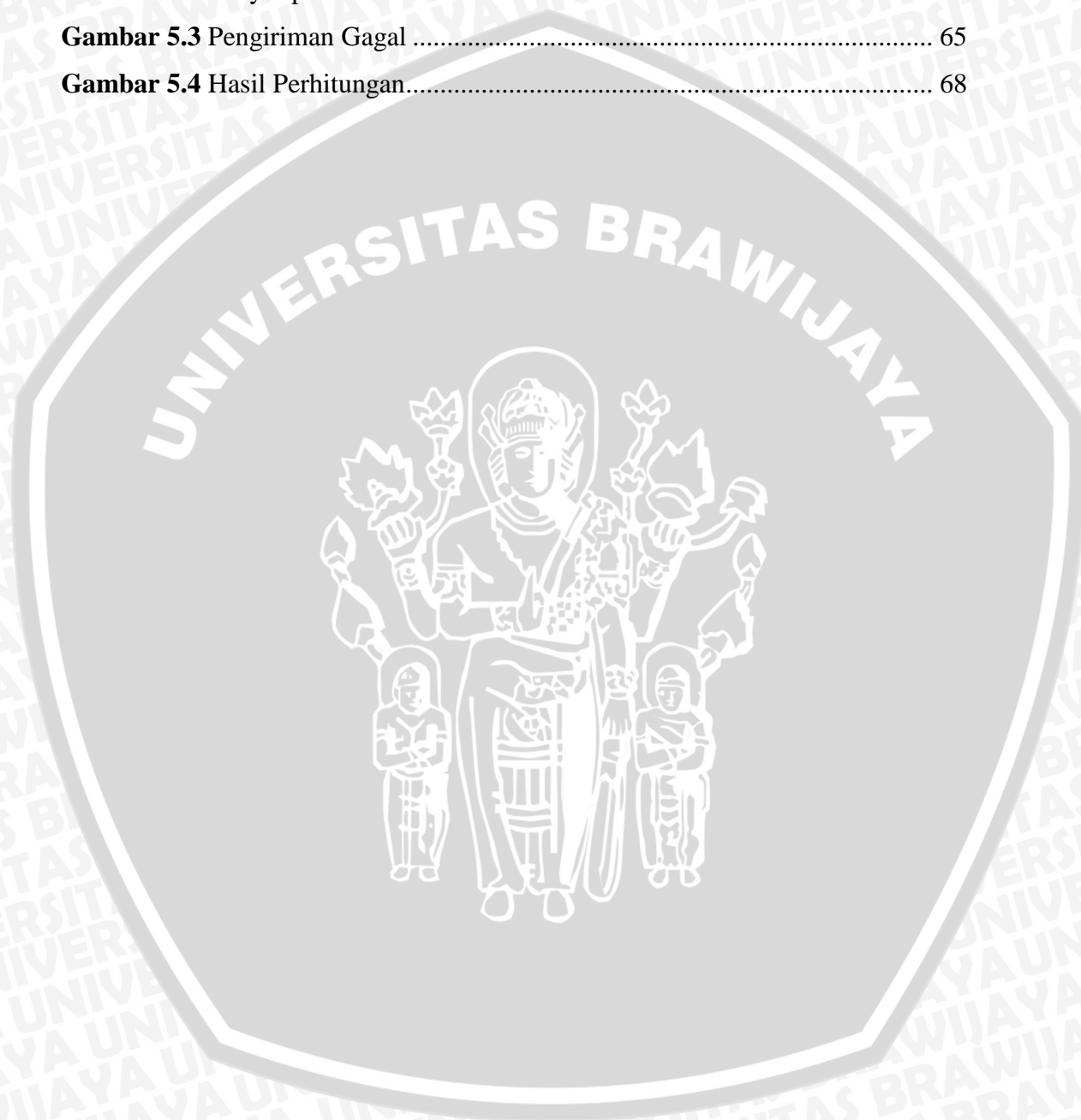
| | |
|---|----|
| Tabel 5.5 Perhitungan Euclidian Distance | 68 |
| Tabel 5.6 Perhitungan Bobot Klasifikasi | 69 |
| Tabel 5.7 Hasil Klasifikasi | 69 |
| Tabel 5.8 Kasus Uji Memantau Total Konsumsi listrik | 70 |
| Tabel 5.9 Kasus Uji Memantau Aktivitas Perangkat Elektronik | 71 |
| Tabel 5.10 Kasus Uji untuk mengubah tarif dasar listrik dan beban daya maksimal | 72 |
| Tabel 5.11 Hasil Pengujian Kebutuhan Fungsional | 73 |
| Tabel 5.12 Daftar Peralatan Elektronik sebagai Bahan Uji | 74 |
| Tabel 5.13 Pengujian Sistem Skenario 1 | 74 |
| Tabel 5.14 Pengujian Sistem Skenario 2 | 75 |
| Tabel 5.15 Pengujian Sistem Skenario 3 | 76 |



DAFTAR GAMBAR

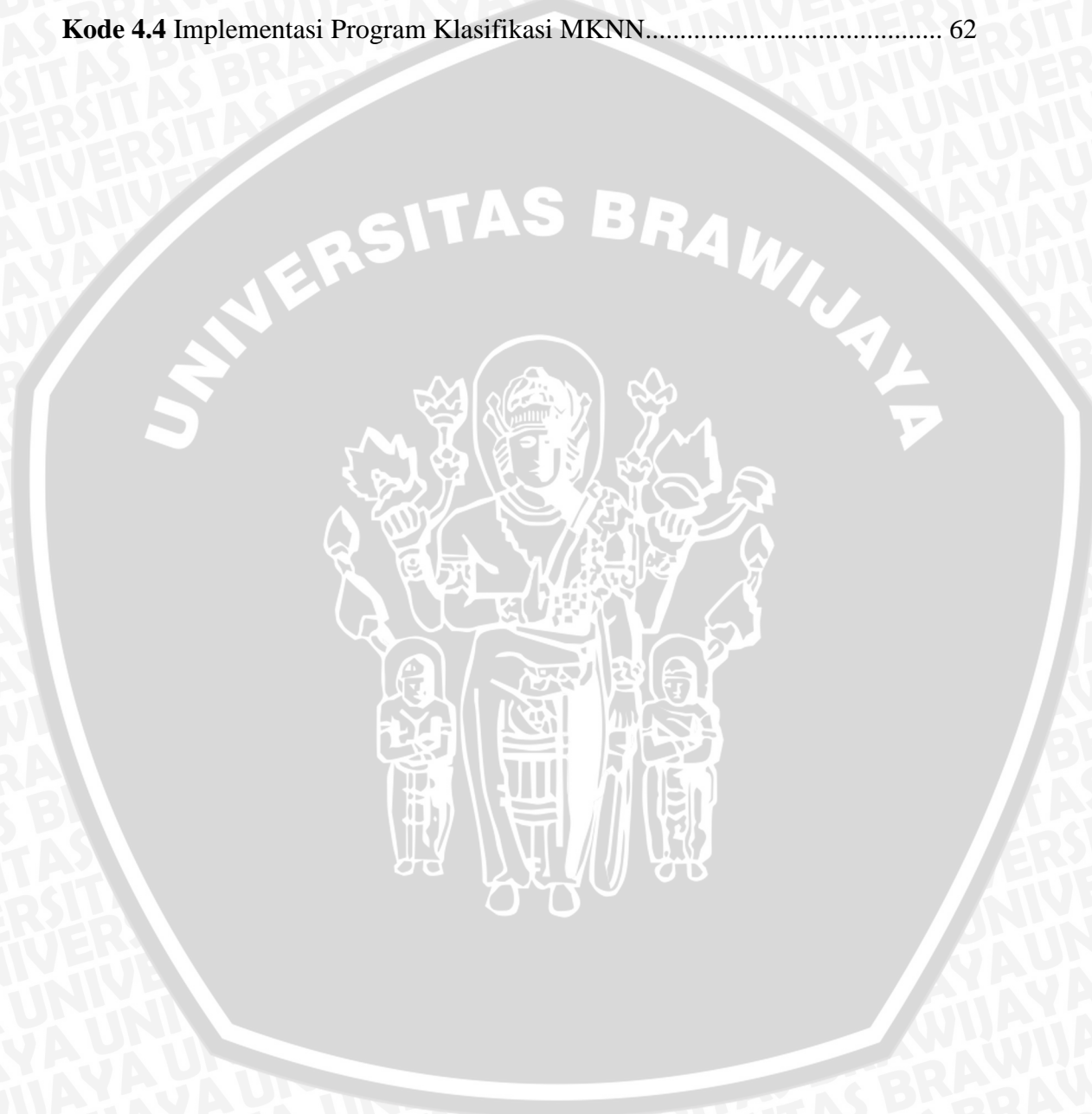
| | |
|--|----|
| Gambar 2.1 Tampak Depan Arduino Uno R3 | 7 |
| Gambar 2.2 Arduino IDE | 9 |
| Gambar 2.3 Current Transformer Sensor | 10 |
| Gambar 3.1 Diagram Alir Metode Penelitian..... | 12 |
| Gambar 4.1 Diagram Blok Sistem..... | 22 |
| Gambar 4.2 Diagram Alir Perancangan Sistem..... | 24 |
| Gambar 4.3 Diagram Konteks | 25 |
| Gambar 4.4 Diagram Aliran Data..... | 25 |
| Gambar 4.5 Rangkaian Elektronik | 26 |
| Gambar 4.6 Pemrograman Mikrokontroler | 27 |
| Gambar 4.7 Pemrograman Mikrokontroler | 28 |
| Gambar 4.8 Diagram Use Case | 30 |
| Gambar 4.9 Entity Relational Database Sistem Monitoring Konsumsi Listrik berbasis Web | 33 |
| Gambar 4.10 Diagram Alir Preprocessing | 36 |
| Gambar 4.11 Proses Data Smoothing..... | 37 |
| Gambar 4.12 Proses Data Smoothing..... | 38 |
| Gambar 4.13 Proses Deteksi Kenaikan dan Penurunan Daya..... | 39 |
| Gambar 4.14 Proses Deteksi Steady State..... | 40 |
| Gambar 4.15 Proses Deteksi Steady State..... | 41 |
| Gambar 4.16 Diagram Alir Proses Validasi Data Training..... | 42 |
| Gambar 4.17 Proses Hitung Euclidian Distance | 43 |
| Gambar 4.18 Proses Perankingan Hasil Euclidian Distance | 44 |
| Gambar 4.19 Proses Perhitungan Nilai Validasi Data Training..... | 45 |
| Gambar 4.20 Proses Perhitungan Nilai Validasi Data Training..... | 46 |
| Gambar 4.21 Diagram alir proses klasifikasi MKNN | 47 |
| Gambar 4.22 Proses Hitung Euclidian Distance | 48 |
| Gambar 4.23 Proses Hitung Bobot Klasifikasi..... | 49 |
| Gambar 4.24 Proses Pengurutan dan Pengambilan Kesimpulan..... | 49 |
| Gambar 4.25 Proses Pengurutan dan Pengambilan Kesimpulan..... | 50 |

| | |
|--|----|
| Gambar 4.26 Implementasi Perangkat Keras | 51 |
| Gambar 4.27 Implementasi Database | 51 |
| Gambar 5.1 Arduino Berhasil Mengirim Data ke Server | 64 |
| Gambar 5.2 Penyimpanan di Database Server | 65 |
| Gambar 5.3 Pengiriman Gagal | 65 |
| Gambar 5.4 Hasil Perhitungan..... | 68 |



DAFTAR KODE

| | |
|--|----|
| Kode 4.1 Implementasi Pemrograman Mikrokontroler..... | 53 |
| Kode 4.2 Implementasi Program Preprocessing Pendeteksian Perangkat..... | 56 |
| Kode 4.3 Implementasi Program Validasi Data Training | 59 |
| Kode 4.4 Implementasi Program Klasifikasi MKNN..... | 62 |



BAB I PENDAHULUAN

1.1 Latar Belakang

Secara tradisional, konsumsi listrik hanya dapat dipantau melalui kwh meter dan itupun hanya terbatas pada total konsumsi listrik. Sementara itu untuk memantau konsumsi listrik dari berbagai peralatan elektronik yang ada di rumah dibutuhkan pemasangan meteran untuk setiap perangkat elektronik yang ada. Pendekatan ini dirasa kurang efisien secara materi karena dibutuhkan banyak meteran dan juga biaya yang mahal. Berdasarkan masalah tersebut, NILM menjadi pilihan yang tepat untuk sistem pemantau konsumsi listrik yang lebih murah. *Non-Intrusive Load Monitoring* (NILM) merupakan suatu metode untuk memantau konsumsi listrik yang lebih murah dan efisien melalui pemanfaatan *smart meter* dan metode pengenalan pola

Sebelumnya Ruzzelli telah membuat sistem pemantau konsumsi listrik yang diberi nama RECAP. Sistem ini menggunakan data dari *smart meter* yang kemudian diproses oleh jaringan syaraf tiruan untuk mengenali perangkat elektronik yang nyala atau mati [RZL-10]. Manaswi Saha juga membuat sistem yang sama dengan nama EnergyLens yang memanfaatkan data dari *smart meter* dan data audio dari *smart phone* yang kemudian diolah dengan metode *K-Nearest Neighbor* dan *Support Vector Machine* untuk mendeteksi perangkat elektronik yang aktif [SHA-14]. Sistem ini menggunakan sensor arus yang dipasang pada kabel positif yang menuju ke rumah untuk memantau konsumsi listrik dan menggunakan algoritma tertentu untuk mengidentifikasi peralatan elektronik yang sedang aktif. Sistem tersebut sering disebut juga dengan NILM (*Non-intrusive Load Monitoring*). Bahkan Bidgely [BLY-11] telah memanfaatkan teknologi ini secara komersil. Namun di Indonesia masih sedikit yang mengembangkan sistem yang menggunakan sensor arus dan metode pengenalan pola untuk memantau konsumsi listrik dari perangkat elektronik di rumah.

Sistem pemantau konsumsi listrik berbasis *web* ini merupakan langkah awal untuk mewujudkan konsep *Smart Home*. *Smart Home* didefinisikan sebagai teknologi dan layanan yang terintegrasi melalui jaringan yang ada di rumah untuk

meningkatkan kualitas hidup manusia. Secara sederhana, kita dapat memantau dan mengendalikan perangkat yang ada di rumah melalui sensor dan aktuator yang terhubung dengan jaringan internet ataupun kabel dari mana saja [ASD-10]. Berdasarkan masalah yang dipaparkan, penulis mengusulkan judul “**Sistem Monitoring Konsumsi Listrik berbasis Web**”. Berbeda dengan struk tagihan bulanan yang hanya menyediakan informasi total pemakaian listrik dan total pembayaran, sistem ini akan menyajikan informasi penggunaan listrik suatu bangunan beserta konsumsi listrik dari masing-masing perangkat elektronik yang ada di dalamnya. Sistem ini akan menggunakan mikrokontroler Arduino Uno yang dikombinasikan dengan sensor tunggal yaitu CT sensor untuk mendapatkan data pemakaian listrik. Lalu data akan dikirim pada sebuah *web server* yang akan menjadi pusat pemrosesan data dari sistem ini untuk kemudian dapat dengan mudah diakses melalui *web browser*. Data akan diproses menggunakan metode *Modified K-Nearest Neighbor* (MKNN) untuk mendeteksi aktifitas nyala dan mati dari perangkat elektronik.

Sistem pemantau konsumsi listrik berbasis *web* ini diharapkan bisa mengubah cara pandang masyarakat mengenai kebiasaan konsumsi listrik mereka dan menjadi dasar untuk mengembangkan sistem kendali perangkat pada *Smart Home*.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan diatas, dapat dibuat rumusan masalah sebagai berikut :

1. Bagaimana membuat sistem monitoring konsumsi listrik berbasis web?
2. Bagaimana metode *Modified K-Nearest Neighbor* dapat digunakan untuk mendeteksi aktifitas perangkat elektronik?
3. Bagaimana performa sistem dalam mendeteksi aktifitas perangkat elektronik?

1.3 Batasan Masalah

Peneliti membatasi pembahasan dalam penelitian skripsi agar penelitian tidak diperluas pada permasalahan lainnya :

1. Perangkat mikrokontroler yang digunakan ialah Arduino Uno R3.

2. Perangkat mikrokontroler akan menggunakan Arduino Ethernet Shield untuk transfer data ke internet.
3. Sensor arus YHDC SCT-013 (100A max) digunakan untuk mengukur konsumsi listrik dari kabel listrik utama pada bangunan.
4. *Database Management System* (DBMS) yang digunakan adalah MySQL.
5. Fokus penelitian adalah tersampainya data ke basis data *server*, tidak membahas keamanan data.
6. Perangkat elektronik yang digunakan sebagai objek penelitian adalah monitor, laptop, lampu, dan setrika

1.4 Tujuan

Adapun tujuan penelitian Sistem Monitoring Konsumsi Listrik berbasis Web ini antara lain :

1. Membuat sistem berbasis web yang mampu memantau konsumsi listrik.
2. Mengetahui cara kerja metode *Modified K-Nearest Neighbor* dalam mendeteksi aktifitas perangkat elektronik

1.5 Manfaat

Manfaat yang diperoleh dari penerapan Sistem Monitoring Konsumsi Listrik berbasis Web antara lain :

1. Memudahkan pengguna memantau besar konsumsi listrik dari waktu ke waktu secara *online*.
2. Membantu pengguna memahami bagaimana mereka menggunakan listrik untuk kemudian berusaha menghemat pemakaian listrik.
3. Mengetahui implementasi algoritma pengenalan pola dalam sistem pemantau konsumsi listrik.

1.6 Sistematika penulisan

Penelitian ini diuraikan dengan sistematika penulisan sebagai berikut :

BAB I Pendahuluan

Pendahuluan menjelaskan latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penulisan pada penelitian skripsi.

BAB II Kajian Pustaka dan Dasar Teori

Memaparkan teori dasar dan teori pendukung yang berhubungan dengan “Rancang Bangun Sistem Monitoring Konsumsi Listrik berbasis Web”.

BAB III Metodologi Penelitian

Membahas metode yang digunakan dalam penelitian untuk merancang serta membangun system pemantau dengan tahapan kegiatan yaitu studi pustaka, identifikasi kebutuhan perangkat, analisa data, perancangan aplikasi, implementasi, pengujian, dokumentasi, serta pengambilan kesimpulan dan saran.

BAB IV Perancangan dan Implementasi

Membahas tentang perancangan sistem yang berisi analisis dan kebutuhan. Pada bab ini juga berisi implementasi perancangan aplikasi pada perangkat sistem pemantau dengan metode ilmiah pada studi literatur yang telah dilakukan.

BAB V Pengujian

Memuat proses uji serta analisis pada sistem pemantau yang telah diimplementasikan guna memperoleh hasil yang diinginkan serta maksimal.

BAB VI Penutup

Memuat kesimpulan yang diperoleh dari perancangan sistem pemantau serta saran yang diharapkan untuk membangun dan menyempurnakan penelitian ini.

BAB II KAJIAN PUSTAKA DAN DASAR TEORI

Bab ini membahas kajian pustaka yang digunakan untuk menunjang penulisan skripsi “Rancang Bangun Sistem Monitoring Konsumsi Listrik berbasis Web”. Dasar teori yang dibutuhkan untuk penyusunan skripsi ini meliputi perangkat elektronik yang digunakan yaitu Arduino Uno R3, Arduino Ethernet Shield, dan Current Transformer Sensor.

2.1 Kajian Pustaka

Saat ini NILM (*Non-Intrusive Load Monitoring*) atau sistem pemantau konsumsi listrik merupakan topik yang hangat dibicarakan utamanya sistem yang dilengkapi dengan fitur yang mampu memantau aktifitas peralatan elektronik di rumah karena kaitannya dengan konsep *Smart Home*. Sebelumnya, untuk memantau konsumsi listrik dari peralatan elektronik di rumah dibutuhkan pemasangan sensor pada setiap perangkat elektronik. Namun sistem ini menyediakan alternatif yang lebih murah dan efisien dalam memantau konsumsi listrik rumah tangga yaitu hanya dengan memasang satu sensor yang dipasang pada kabel listrik utama [HRT-85].

Manaswi Saha telah mengembangkan sistem pemantau konsumsi listrik EnergyLens yang memanfaatkan data dari *smart meter* dan data audio dari *smart phone* untuk mendeteksi perangkat elektronik yang aktif. EnergyLens juga membutuhkan data tambahan berupa data pemetaan lokasi dari perangkat elektronik dan juga informasi daya dari setiap perangkat elektronik sebagai *data training*. EnergyLens menggunakan metode *Support Vector Machine* untuk mendeteksi perangkat yang sedang aktif. Sebagai keluarannya, EnergyLens mampu mengenali perangkat yang aktif di ruangan tertentu dan juga lamanya perangkat menyala [SHA-14].

Sementara itu Bidgely telah memanfaatkan teknologi NILM secara komersil. Bidgely juga menggunakan data dari *smart meter* yang kemudian akan diproses dengan metode pengenalan pola untuk menganalisa perangkat elektronik yang sedang menyala [BLY-11].

Modified K-Nearest Neighbor merupakan pengembangan dari metode *K-Nearest Neighbor*. Metode ini menambahkan proses validasi dataset yang akan meningkatkan akurasi klasifikasi daripada KNN pada umumnya [HAP-08]. Berdasarkan beberapa penelitian sebelumnya, penulis memperoleh ide untuk membuat sistem pemantau listrik yang memanfaatkan data dari *smart meter* dan juga metode klasifikasi MKNN (*Modified K-Nearest Neighbour*) untuk mengenali perangkat elektronik yang menyala.

2.2 Dasar Teori

Bagian ini membahas mengenai informasi umum dari perangkat, aplikasi, dan metode perhitungan yang akan digunakan pada penelitian ini.

2.2.1 *Smart Home*

Smart Home (Rumah Pintar) merupakan teknologi yang mengimplementasikan sebuah sistem canggih disebuah rumah atau bangunan yang memungkinkan penggunanya untuk memantau dan mengendalikan perangkat yang ada di rumah secara otomatis maupun jarak jauh. Teknologi *Smart Home* diharapkan mampu terintegrasi dengan smart home lain serta menciptakan kemudahan, kenyamanan, keamanan dan efisiensi bagi penggunanya. Teknologi *Smart Home* merupakan gabungan teknologi jaringan, sensor, aktuator, serta kecerdasan buatan. Sehingga teknologi smart home ini mampu dapat membantu penggunanya memonitor, mengontrol, memprediksi, mengantisipasi serta membuat keputusan.

2.2.2 Mikrokontroler

Mikrokontroler adalah sebuah sistem komputer fungsional dalam sebuah chip. Di dalamnya terkandung sebuah inti prosesor, memori (sejumlah kecil RAM, memori program, atau keduanya), dan perlengkapan input output. Dengan kata lain, mikrokontroler adalah suatu alat elektronika digital yang mempunyai masukan dan keluaran serta kendali dengan program yang bisa ditulis dan dihapus dengan cara khusus, cara kerja mikrokontroler sebenarnya membaca dan menulis data. Mikrokontroler yang digunakan pada penelitian ini ialah Arduino yang merupakan platform yang terdiri dari *hardware* dan *software*. *Hardware*nya

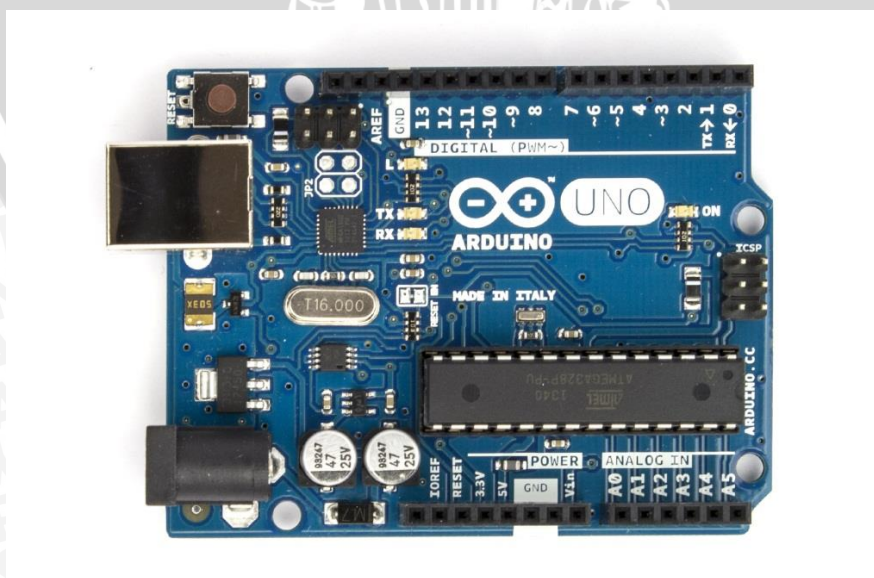
berupa mikrokontroler yang ditambahkan penamaan pin agar mudah diingat. *Software* Arduino merupakan *software open source* sehingga dapat diunduh secara gratis. *Software* ini digunakan untuk membuat dan memasukkan program ke dalam Arduino.

Software Arduino terdiri dari 3 bagian, yaitu :

1. *Editor*, untuk menulis dan memanipulasi program (*source code*). Kode program pada Arduino disebut dengan *sketch*.
2. *Compiler*, modul yang berfungsi mengubah *source code* menjadi bentuk biner agar bisa dieksekusi oleh mikrokontroler.
3. *Uploader*, modul yang berfungsi memasukkan kode biner ke dalam memori mikrokontroler.

2.2.3 Arduino Uno R3

Arduino Uno R3 adalah mikrokontroler berbasis ATmega328. Arduino sendiri merupakan perangkat elektronik berbasis *open source*. Arduino bekerja dengan menerima input dari sensor dan bertindak dengan lampu, motor, dan aktuator lainnya. Arduino juga dilengkapi dengan *Arduino programming language* yang memungkinkan kita untuk menuliskan perintah apa saja dalam bahasa C/C++. Gambar 2.1 berikut ini merupakan gambar tampak depan dari Arduino Uno R3.



Gambar 2.1 Tampak Depan Arduino Uno R3

Spesifikasi dari Arduino Uno R3 dapat dilihat pada Tabel 2.1 berikut ini.

Tabel 2.1 Spesifikasi Arduino Uno R3

| | |
|-----------------------------------|---|
| Mikrokontroler | ATmega328 |
| Tegangan Kerja | 5V |
| Input Tegangan (direkomendasikan) | 7V – 12V |
| Input Tegangan (batas) | 6V – 20V |
| Digital I/O Pin | 14 (6 PWM Output) |
| Analog Input Pin | 6 |
| Arus DC per pin I/O | 40 mA |
| Arus DC pin 3.3V | 50 mA |
| Flash Memory | 32 KB (ATmega328) yang mana 0.5 KB digunakan untuk bootloader |
| SRAM | 2 Kb (ATmega328) |
| EEPROM | 1 Kb (ATmega328) |
| Clock Speed | 16 MHz |

Arduino dapat ditenagai oleh arus listrik dari koneksi kabel USB maupun sumber listrik lainnya seperti baterai dan adaptor AC/DC. Arduino akan secara otomatis mengenali sumber listrik yang digunakan. Direkomendasikan untuk digunakan pada tegangan 7V – 12V untuk menjaga kestabilan ArduinoUno.

2.2.4 Arduino Ethernet Shield R3

Arduino Ethernet Shield adalah perangkat yang menghubungkan Arduino Uno dengan internet. Kita hanya perlu menghubungkan Arduino Ethernet Shield ke papan Arduino Uno, kemudian gunakan kabel RJ45 untuk terhubung ke jaringan internet. Perangkat ini dibangun dengan basis Wiznet W5100 Ethernet Chip yang menunjang protokol TCP dan UDP.

2.2.5 Arduino IDE

Arduino IDE adalah sebuah editor yang digunakan untuk menulis, meng*compile*, dan mengunggah program ke papan Arduino. Perangkat lunak yang ditulis menggunakan Arduino IDE disebut *sketches* yang disimpan pada file berekstensi .ino. Gambar 2.2 merupakan contoh Arduino IDE.



```
Arduino - 0011 Alpha
File Edit Sketch Tools Help
Blink
/*
 * Blink
 *
 * The basic Arduino example. Turns on an LED on for one second,
 * then off for one second, and so on... We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */
int ledPin = 13;          // LED connected to digital pin 13
void setup()             // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}
void loop()              // run over and over again
{
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(1000);              // waits for a second
  digitalWrite(ledPin, LOW);  // sets the LED off
  delay(1000);              // waits for a second
}
Done compiling.
Binary sketch size: 1098 bytes (of a 14336 byte maximum)
22
```

Gambar 2.2 Arduino IDE

2.2.6 OpenEnergyMonitor

OpenEnergyMonitor merupakan sistem pemantau konsumsi energi *open source*. Proyek ini menyediakan informasi untuk membuat alat pemantau konsumsi listrik yang berbasis mikrokontroler mulai dari rancangan elektronika dan juga *library* pemrograman untuk mengembangkan alat pemantau konsumsi listrik kita sendiri.

2.2.7 MySQL

MySQL merupakan basis data relasional *open source* yang paling banyak digunakan di dunia. MySQL sering sekali digunakan dalam aplikasi berbasis web. Dikembangkan dan didistribusikan oleh Oracle, MySQL dapat dijalankan pada sistem operasi Windows, Linux, Solaris, OS X, dan FreeBSD. Beberapa aplikasi terkenal yang menggunakan MySQL adalah Joomla, Wordpress, dan Drupal. Selain itu MySQL juga digunakan oleh banyak *website* terkenal seperti Google, Facebook, Twitter, dan YouTube.

2.2.8 *Current Transformer Sensor (CT Sensor)*

Current Transformer adalah sensor yang digunakan untuk mengukur arus listrik. Sensor ini sangat berguna untuk mengukur total konsumsi listrik pada suatu bangunan. Gambar 2.3 menunjukkan penampakan *current transformer* sensor.



Gambar 2.3 Current Transformer Sensor

Trafo seperti gambar 2.3 di atas memiliki kepala yang bisa dibuka, sehingga sangat cocok dan mudah dipasangkan pada kabel yang bermuatan listrik atau kabel *ground* yang menuju ke bangunan tanpa harus melakukan aktifitas listrik yang melibatkan arus tinggi.

Seperti trafo pada umumnya, trafo ini terdiri dari 3 komponen yaitu lilitan utama, inti magnet, dan lilitan sekunder. Untuk mengukur konsumsi listrik suatu bangunan maka salah satu dari kabel yang bermuatan atau kabel *ground* dimasukkan pada lubang di kepala trafo. Arus listrik pada lilitan primer akan menciptakan medan magnet pada inti trafo yang kemudian menginduksi listrik ke lilitan sekunder.

2.2.9 *Modified K-Nearest Neighbor*

K-Nearest Neighbor merupakan metode pengenalan pola yang tergolong *supervised learning* dimana objek yang ada pada dataset telah memiliki label/kelas. KNN merupakan metode pengelompokan yang paling dasar dan mudah. Pada perkembangannya, muncul varian baru dari KNN yang disebut *Modified K-Nearest Neighbor* (MKNN). Pada dasarnya proses algoritma KNN dan MKNN

serupa, mulai dari penentuan nilai K dan penghitungan jarak antar data. Yang membedakan adalah ditambahkannya proses pembobotan terhadap dataset untuk menguji validitas dataset. Teknik ini terbukti bisa menghasilkan akurasi pengelompokan data yang lebih baik daripada KNN [HAP-08]. Berikut ini adalah langkah-langkah dalam proses MKNN.

1. Untuk menggunakan MKNN, hal pertama yang perlu dilakukan adalah menghitung nilai validasi dari setiap data pada *data training*. Untuk menghitung nilai validasi diawali dengan menghitung *Euclidian distance* antar data dengan rumus

$$d(p, q) = \sqrt{(q_i - p_i)^2 + (q_i - p_i)^2}$$

$d(p, q)$ = jarak antara data q dan p

p_i = *data training* ke- i

q_i = data uji ke- i

2. Setelah didapat nilai *Euclidian distance*, langkah selanjutnya ialah meranking nilai *Euclidian distance* kemudian memilih k nilai teratas.
3. Lalu dicari kesesuaian label dari setiap data pada *data training* dengan k nilai teratas hasil perankingan. Jika nilai sama maka diberi nilai 1, jika tidak sama diberi nilai 0.
4. Selanjutnya ialah menghitung nilai validasi berdasarkan tabel 2.2 dengan rumus

$$\text{nilai validasi} = \frac{\text{jumlah nilai 1}}{k} * \text{sum}$$

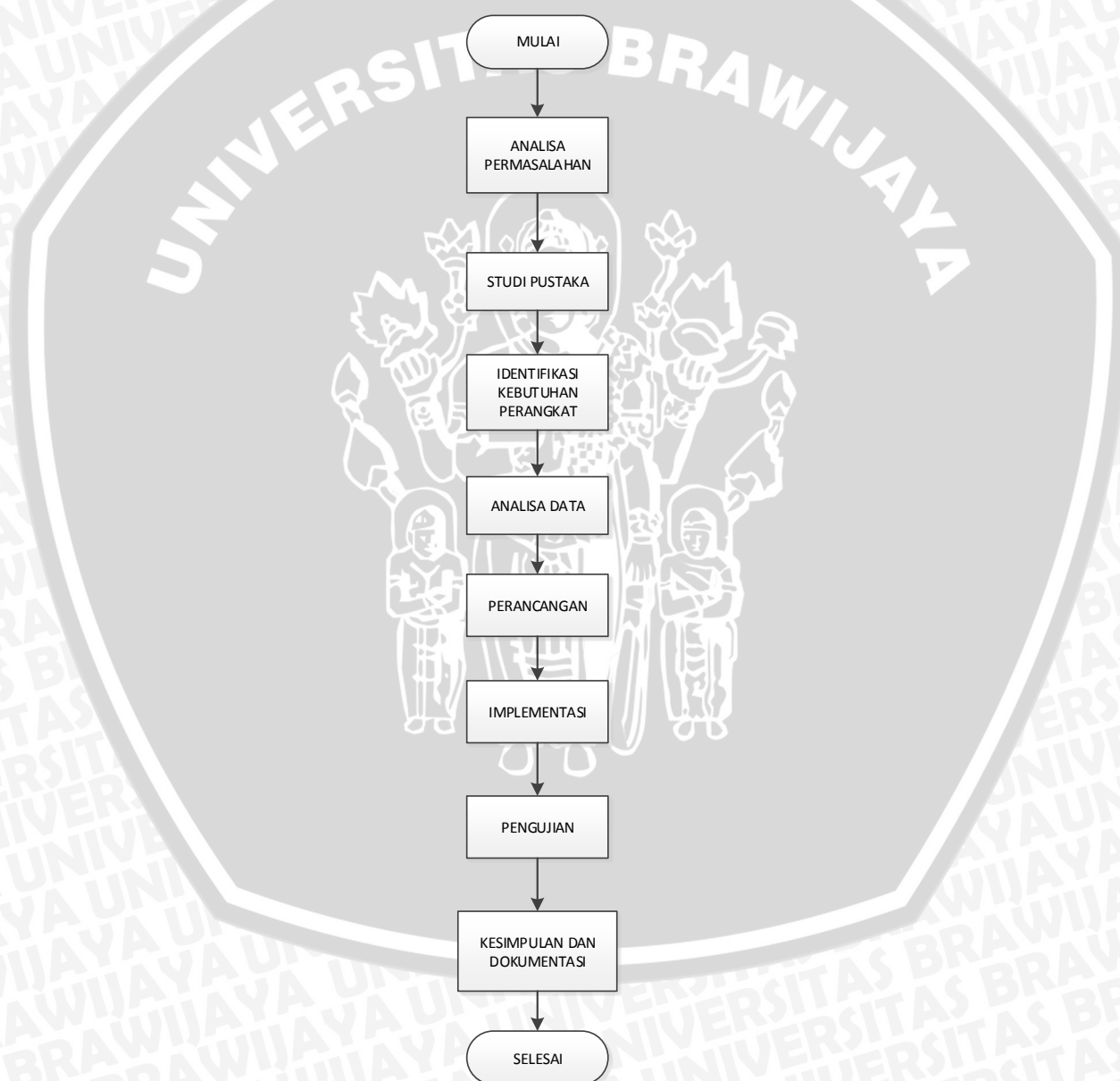
5. Setelah mendapatkan nilai validasi dari *data training*, maka kita dapat menggunakan nilai validasi ini untuk menghitung bobot klasifikasi. Pertama-tama hitung *Euclidian distance* data x dengan *data training*, kemudian hitung bobot klasifikasi dengan rumus berikut

$$\text{bobot} = \text{nilai validasi} * \left(\frac{1}{\text{euclidian distance}} + 0.5 \right)$$

6. Kemudian dipilih sebanyak k data dengan bobot yang paling besar sebagai hasil dari klasifikasi.

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan metode dan langkah yang akan di tempuh dalam pengerjaan skripsi ini yaitu Analisis Permasalahan, Studi Pustaka, Identifikasi Kebutuhan Perangkat, Analisis Data, Perancangan Aplikasi, Implementasi, Pengujian serta Kesimpulan dan Dokumentasi. Gambar 3.1 menunjukkan diagram alir proses pengerjaan skripsi.



Gambar 3.1 Diagram Alir Metode Penelitian

3.1 Analisis Permasalahan

Analisis permasalahan ini digunakan untuk mencari solusi dari permasalahan yang dihadapi. Permasalahan tersebut yaitu melakukan monitoring konsumsi listrik dari peralatan elektronik di rumah dengan biaya yang murah..

3.2 Studi Pustaka

Studi pustaka digunakan untuk mencari literatur dari sumber yang terpercaya guna mendapatkan metode-metode ilmiah guna memecahkan masalah yang dihadapi. Literatur yang digunakan dapat berupa buku, jurnal, laporan penelitian maupun artikel. Referensi utama yang diperlukan untuk menunjang penulisan skripsi ini berhubungan dengan kajian pustaka, Arduino Uno R3, *Current Transformer* (CT) sensor, dan metode MKNN. Dengan melakukan studi pustaka diharapkan mampu memilih perangkat dan metode klasifikasi yang tepat untuk menyelesaikan masalah yang dihadapi.

3.3 Identifikasi Kebutuhan Perangkat Keras (Hardware)

Identifikasi kebutuhan perangkat atau *device requirements* berisi spesifikasi perangkat yang diperlukan guna menunjang penulisan skripsi dan aplikasi yang akan dibuat. Kebutuhan perangkat keras pada penelitian ini didapatkan dari dokumentasi OpenEnergyMonitor.org. Perangkat Keras (*Hardware*) berupa peralatan elektronika utama seperti Arduino Uno R3, Arduino Ethernet Shield, dan *Current Transformer* sensor untuk membuat alat pengambilan data. Peneliti juga memerlukan komponen elektronika yang berupa resistor, kapasitor dan *Printed Circuit Board* (PCB). Penunjang lainnya yang tidak kalah penting yaitu kabel UTP RJ45 atau lebih dikenal kabel LAN agar Arduino bisa terkoneksi ke internet.

3.4 Identifikasi kebutuhan Perangkat Lunak (Software)

Pada bagian ini diidentifikasi perangkat lunak yang dibutuhkan oleh pengguna. Perangkat Lunak (*Software*) mendasar yang dibutuhkan berupa Sistem Operasi Windows yang akan digunakan untuk pemasangan *Database Management System* (DBMS) atau Sistem Manajemen *Database*, MySQL digunakan untuk pengolahan serta penyimpanan database, Arduino IDE untuk pemrograman Arduino.

3.5 Identifikasi Kebutuhan Fungsional

Identifikasi kebutuhan fungsional sistem berisi tentang fitur-fitur pengguna yang akan disediakan oleh sistem. Kebutuhan fungsional didapatkan dari observasi terhadap sistem yang telah ada sebelumnya. Observasi ini dilakukan untuk mengetahui fitur dasar sebuah sistem monitoring energi listrik dan juga mengetahui kemungkinan penambahan atau pengurangan fitur untuk sistem baru yang sedang dikembangkan oleh penulis.

3.6 Perancangan

Perancangan dilakukan setelah semua data dan spesifikasi kebutuhan diketahui. Perancangan ini dibagi menjadi 2 yaitu perancangan perangkat keras dan perancangan perangkat lunak.

Perancangan perangkat keras akan dititikberatkan dalam pembuatan alat untuk pengambilan data dan juga pengiriman data ke *web server*. Sedangkan perancangan perangkat lunak akan mengarah pada pengolahan data untuk mendeteksi peralatan elektronik yang sedang menyala atau mati.

3.6.1 Pengambilan Data

Pada penelitian ini data diambil dengan cara observasi. Observasi dilakukan dengan menggunakan sensor untuk mendapatkan data arus dan tegangan listrik. Lalu data tersebut dikirim ke *server* dengan interval 1 detik untuk kemudian diolah oleh program pengolahan data.

3.6.2 Pengolahan Data

Pada sistem ini, data hasil observasi akan diolah dengan metode klasifikasi *Modified K-Nearest Neighbor*. Proses klasifikasi ini berguna untuk mendeteksi aktifitas nyala dan mati dari perangkat elektronik. Berikut adalah langkah-langkah klasifikasi MKNN dilengkapi dengan contoh menggunakan data sederhana.

1. Pertama-tama kita harus menyiapkan *data training* seperti yang ditunjukkan oleh Tabel 3.1

Tabel 3.1 Data Training

| NO | KELAS | X | Y |
|----|-------|-------------|------------|
| 1 | A | 51.59072727 | 0.41132468 |
| 2 | B | 320.5715259 | 1.49877037 |
| 3 | C | 51.118016 | 0.314248 |
| 4 | A | 40.50401942 | 0.28884466 |

2. Kemudian dilanjutkan dengan menghitung nilai validasi dari *data training*. Untuk menghitung nilai validasi, maka pertama-tama hitung nilai *Euclidian distance* dari setiap data pada *data training*. *Euclidian distance* dihitung dengan rumus

$$d(p,q) = \sqrt{(q_i - p_i)^2 + (q_i - p_i)^2}$$

$d(p,q)$ = jarak antara data q dan p

p_i = data training ke- i

q_i = data uji ke- i

Perhitungan *Euclidian distance* ditunjukkan oleh Tabel 3.2.

Tabel 3.2 Perhitungan *Euclidian distance*

| KELAS | DATA | 1 | 2 | 3 | 4 |
|-------|------|-------------|----------|----------|----------|
| A | 1 | 0 | 268.983 | 0.482576 | 11.08738 |
| B | 2 | 268.9829968 | 0 | 269.4561 | 280.0701 |
| C | 3 | 0.482576241 | 269.4561 | 0 | 10.61403 |
| A | 4 | 11.08738438 | 280.0701 | 10.61403 | 0 |

3. Kemudian dilanjutkan dengan meranking hasil perhitungan *Euclidian distance* mulai dari nilai yang terkecil. Perankingan *Euclidian distance* ditunjukkan oleh Tabel 3.3.

Tabel 3.3 Perankingan *Euclidian distance*

| KELAS | DATA | 1 | 2 | 3 | 4 |
|-------|------|---|---|---|---|
| A | 1 | 1 | 2 | 2 | 3 |
| B | 2 | 4 | 1 | 4 | 4 |
| C | 3 | 2 | 3 | 1 | 2 |
| A | 4 | 3 | 4 | 3 | 1 |

4. Langkah selanjutnya yaitu mengecek kesesuaian kelas. Data yang memiliki nama/kelas yang sesuai diberi nilai 1 sedangkan lainnya 0. Mula-mula tentukan k data dari hasil perankingan yang akan diambil sebagai pembanding. Urutan ranking ke-1 tidak diikutsertakan pada tahap ini karena data dengan urutan ke-1 adalah data itu sendiri. Pada contoh di tabel 3.3 nilai k ditentukan sebesar 2, sehingga data yang diambil ialah data dengan nomor ranking ke-2 dan ke-3. Misalnya untuk data nomor 1 termasuk dalam kelas A, kemudian diambil ranking ke-2 ialah data nomor 3 dengan kelas C maka nilai k_1 adalah 0. Lalu diambil ranking ke-3 yaitu data nomor 4 yang termasuk ke dalam kelas A, maka nilai k_2 adalah 1. Tabel 3.4 menunjukkan proses pengecekan kesesuaian data.

Tabel 3.4 Mengecek Kesesuaian Data

| KELAS | DATA | K 1 | K 2 | SUM |
|-------|------|-----|-----|-----|
| A | 1 | 0 | 1 | 1 |
| B | 2 | 0 | 0 | 0 |
| C | 3 | 0 | 0 | 0 |
| A | 4 | 0 | 1 | 1 |

5. Langkah selanjutnya adalah menghitung nilai validasi. Nilai validasi didapat dengan rumus

$$\text{nilai validasi} = \frac{\text{jumlah nilai 1}}{k} * \text{sum}$$

Tabel 3.5 menunjukkan perhitungan nilai validasi.

Tabel 3.5 Menghitung Nilai Validasi

| KELAS | DATA | K 1 | K 2 | SUM | Validasi |
|-------|------|-----|-----|-----|----------|
| A | 1 | 0 | 1 | 1 | 0.3333 |
| B | 2 | 0 | 0 | 0 | 0 |
| C | 3 | 0 | 0 | 0 | 0 |
| A | 4 | 0 | 1 | 1 | 0.3333 |

6. Setelah mendapatkan nilai validasi *data training*, maka kita dapat menggunakan *data training* untuk proses klasifikasi. Misalkan kita memiliki data uji seperti pada Tabel 3.6.

Tabel 3.6 Data Uji

| NO | X | Y |
|----|--------|-------|
| 1 | 51.353 | 0.313 |

7. Selanjutnya ialah menghitung *Euclidian distance* antara data uji dengan *data raining*. Perhitungan *Euclidian distance* ditunjukkan oleh Tabel 3.7.

Tabel 3.7 Perhitungan *Euclidian distance*

| Data | Kelas | <i>Euclidian distance</i> |
|------|-------|---------------------------|
| 1 | A | 0.257258621 |
| 2 | B | 269.2211373 |
| 3 | C | 0.234987314 |
| 4 | A | 10.84900747 |

8. Kemudian dilanjutkan dengan menghitung bobot klasifikasi dengan rumus

$$\text{bobot} = \text{nilai validasi} * \left(\frac{1}{\text{euclidian distance}} + 0.5 \right)$$

Perhitungan bobot klasifikasi ditunjukkan oleh Tabel 3.8

Tabel 3.8 Perhitungan Bobot Klasifikasi

| Data | Kelas | Euclidian | Bobot |
|------|-------|-------------|-------------|
| 1 | A | 0.257258621 | 1.462379643 |
| 2 | B | 269.2211373 | 0.167904806 |
| 3 | C | 0.234987314 | 1.585182958 |
| 4 | A | 10.84900747 | 0.197391443 |

9. Setelah itu pilih sebanyak k data yang memiliki bobot klasifikasi terbesar. Pada contoh nilai k ditentukan sebesar 2. 2 nilai yang dipilih akan menjadi hasil dari klasifikasi. Proses pengambilan hasil klasifikasi ditunjukkan oleh Tabel 3.9.

Tabel 3.9 Hasil Klasifikasi

| Data | Kelas | Euclidian | Bobot |
|------|-------|-------------|-------------|
| 1 | B | 269.2211373 | 0.167904806 |
| 2 | A | 10.84900747 | 0.197391443 |

3.5 Implementasi

Tahap implementasi menjelaskan penerapan desain dalam sistem mulai dari pembuatan perangkat keras, pengiriman data ke *web server*, dan pembuatan perangkat lunak untuk mendeteksi aktifitas perangkat elektronik.

3.6 Pengujian

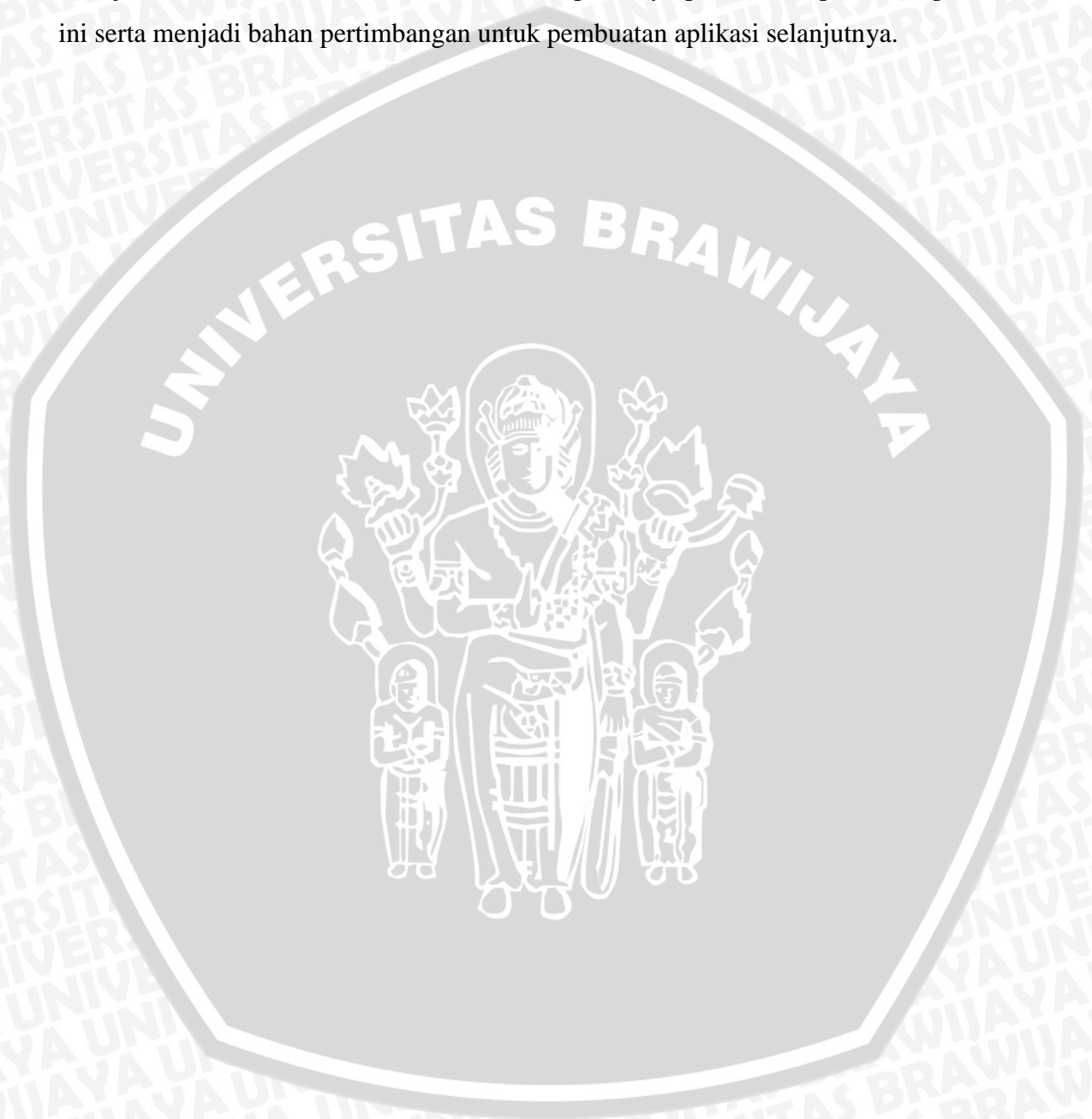
Pengujian pada sistem dilakukan dengan mengacu pada perancangan sistem, yang meliputi:

1. Pengiriman data dari Arduino ke *database*
2. Pengujian sensor
3. Pengujian fungsional sistem
4. Pengujian keseluruhan sistem

3.7 Kesimpulan dan Saran

Penarikan kesimpulan dilakukan setelah semua tahap perancangan, implementasi, pengujian dan analisis sistem selesai dilakukan. Kesimpulan

didapatkan dari hasil pengujian dan analisis sistem yang dibuat. Kesimpulan diambil bertujuan untuk menjawab permasalahan yang ada didalam rumusan masalah. Tahap akhir dari penulisan laporan skripsi ini adalah saran yang bertujuan untuk memberi masukan demi sempurnanya penulisan laporan skripsi ini serta menjadi bahan pertimbangan untuk pembuatan aplikasi selanjutnya.



BAB IV PERANCANGAN DAN IMPLEMENTASI

4.1 Perancangan

Pada subbab ini membahas perancangan sistem pemantau konsumsi listrik berbasis web.

4.1.1 Analisa Kebutuhan

Bagian ini akan menjelaskan kebutuhan perangkat keras, perangkat lunak, dan juga kebutuhan fungsional untuk membangun sistem monitoring konsumsi listrik.

4.1.1.1 Analisa Kebutuhan Perangkat Keras

Kebutuhan perangkat keras ini berdasarkan panduan dari proyek *open source* OpenEnergyMonitor. Proyek *open source* inilah yang menjadi acuan untuk membuat alat pengambil data konsumsi listrik. Perangkat keras yang dibutuhkan untuk pengembangan sistem adalah sebagai berikut. Tabel 4.1 menunjukkan analisa kebutuhan perangkat keras pada penelitian ini.

Tabel 4.1 Kebutuhan Perangkat Keras

| No | Perangkat | Fungsi |
|----|---|---|
| 1 | Arduino Uno R3 | Mikrokontroler untuk mengirim data ke <i>server</i> |
| 2 | Arduino Ethernet Shield | Menghubungkan Arduino ke internet |
| 3 | Sensor arus (YHDC-SCT-0013-00 max 100A) | Mengambil data arus listrik |
| 4 | Sensor tegangan (AC/AC 9V) | Mengambil data tegangan listrik |
| 5 | Rangkaian elektronika/converter | Untuk mengubah sinyal arus dari sensor ke tegangan yang sesuai dengan Arduino |

4.1.1.2 Analisa Kebutuhan Perangkat Lunak

Pada bagian ini dijelaskan perangkat lunak yang digunakan untuk mengolah data dan membuat *source code* untuk menjalankan aplikasi berbasis

web dan juga mengoperasikan mikrokontroler. Perangkat lunak yang dibutuhkan untuk pengembangan sistem ditunjukkan oleh Tabel 4.2.

Tabel 4.2 Analisa Kebutuhan Perangkat Lunak

| No | Aplikasi | Fungsi |
|----|---------------------|---|
| 1 | Arduino IDE | <i>Editor</i> untuk pemrograman Arduino |
| 2 | Microsoft Excel | Aplikasi untuk mengolah data perhitungan MKNN |
| 3 | DBMS MySQL | Basis data |
| 4 | Sublime Text Editor | <i>Editor</i> untuk pemrograman aplikasi |

4.1.1.3 Analisis Kebutuhan Fungsional

Pada bagian ini dijabarkan kebutuhan fungsional yang ada pada sistem. Kebutuhan fungsional ini menjelaskan fitur-fitur yang disajikan kepada pengguna oleh aplikasi berbasis web. Penulis melakukan observasi terhadap sistem monitoring konsumsi listrik yang telah ada sebelumnya yaitu Bidgely. Fitur-fitur yang ada pada aplikasi Bidgely dapat dilihat pada Tabel 4.3.

Tabel 4.3 Fitur Aplikasi Bidgely

| No | Fitur |
|----|---|
| 1 | Memantau total konsumsi listrik selama sehari |
| 2 | Memantau konsumsi listrik perangkat elektronik di rumah |
| 3 | Perbandingan konsumsi listrik dengan pengguna lainnya |
| 4 | Rekomendasi tindakan penghematan listrik |

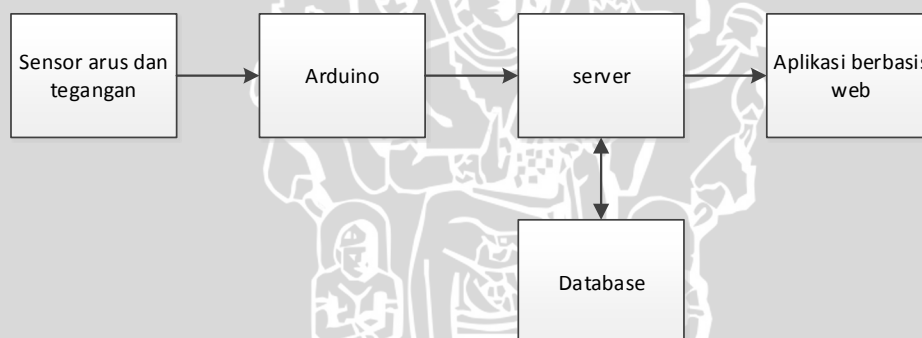
Tabel 4.3 menunjukkan fitur-fitur yang dimiliki oleh Bidgely. Berdasarkan data pada Tabel 4.3, penulis ingin mengadaptasi fitur untuk memantau konsumsi listrik seluruh bangunan dan juga perangkat elektronik yang ada di rumah. Namun penulis ingin menambahkan fitur untuk memantau status nyala atau mati dari perangkat elektronik dan juga fitur untuk mengubah tarif dasar listrik. Tabel 4.4 menunjukkan kebutuhan fungsional dari sistem monitoring konsumsi listrik berbasis web dengan kode penomoran FRXX.

Tabel 4.4 Analisis Kebutuhan Fungsional

| Nomor | Deskripsi | Use Case |
|-------|--|--|
| FR01 | Pengguna dapat memantau total konsumsi listrik rumah | Memantau total konsumsi listrik rumah |
| FR02 | Pengguna dapat memantau status nyala dan mati dari perangkat elektronik di rumah | Memantau aktifitas perangkat elektronik |
| FR03 | Pengguna dapat mengganti nilai tarif dasar listrik dan beban daya maksimal | Mengubah tarif dasar listrik dan beban daya maksimal |

4.1.2 Diagram Blok

Dari analisis kebutuhan yang ada dapat kita buat diagram blok dari sistem tersebut yang terdiri dari bagian-bagiannya. Diagram blok ditunjukkan oleh Gambar 4.1.



Gambar 4.1 Diagram Blok Sistem

Diagram blok pada gambar 4.1 memperlihatkan beberapa entitas/blok yang saling terhubung untuk membentuk sistem monitoring konsumsi listrik berbasis web. Penjelasan lebih lanjut tentang masing-masing entitas akan dibahas pada subbab 4.1.2.1 sampai dengan 4.1.2.4.

4.1.2.1 Sensor Arus dan Tegangan

Sensor arus yang digunakan adalah YHDC-SCT-0013-00 max 100A dan sensor tegangan berupa trafo AC/AC 9 V. Untuk memperoleh data arus listrik, sensor arus dipasang pada kabel positif atau *ground* dan sensor tegangan dipasangkan pada sumber listrik.

4.1.2.2 Arduino

Pada sistem ini, Arduino Uno R3 digunakan sebagai alat untuk menerima data dari sensor dan mengolah data arus dan tegangan menjadi daya listrik, daya total, dan faktor daya. Arduino juga terhubung dengan Arduino Ethernet Shield untuk dapat mengirimkan data ke *server*.

4.1.2.3 Server

Pada sistem ini, *server* berfungsi sebagai penerima data dari sensor dan juga sebagai penyedia data untuk diproses oleh aplikasi berbasis web. Spesifikasi *server* yang digunakan pada sistem ini ditunjukkan oleh Tabel 4.5.

Tabel 4.5 Spesifikasi *Server*

| | |
|-------------------|----------------------------------|
| Produk | Acer Veriton |
| <i>Processor</i> | Intel® Core i3-2120 CPU 3.30 GHz |
| <i>RAM</i> | 8.00 GB |
| Sistem Operasi | Debian 7 64-bit |
| <i>Web Server</i> | Apache 2.4.10 Debian |

4.1.2.4 Database

Database digunakan untuk menyimpan data yang dikirim oleh mikrokontroler. DBMS yang digunakan ialah MySQL.

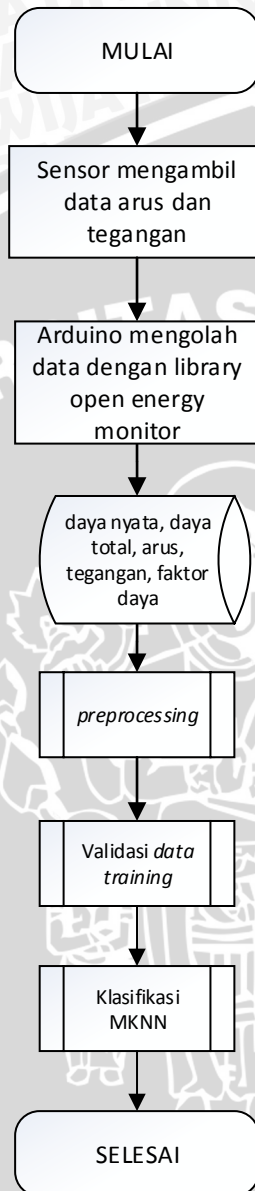
4.1.2.5 Aplikasi Web

Aplikasi web digunakan untuk menampilkan data aktifitas perangkat elektronik dan juga total konsumsi listrik penggunaannya.

4.1.3 Diagram Alir Sistem

Pada penelitian ini, alur kerja sistem dimulai dari pengambilan data arus dan tegangan dari sensor yang kemudian diteruskan ke mikrokontroler Arduino. Selanjutnya Arduino memproses data arus dan tegangan menjadi daya nyata, daya total, dan faktor daya. Pemrosesan data di Arduino dibantu dengan *library* OpenEnergyMonitor. Kemudian data akan disimpan ke *database* setiap detik. Saat pengguna mengakses sistem maka aktifitas pendeteksian perangkat akan dilakukan setiap 5 menit. Sistem akan otomatis mengambil data dari *database* dan

melakukan proses pendeteksian status perangkat elektronik. Gambar 4.2 merupakan diagram alir sistem.

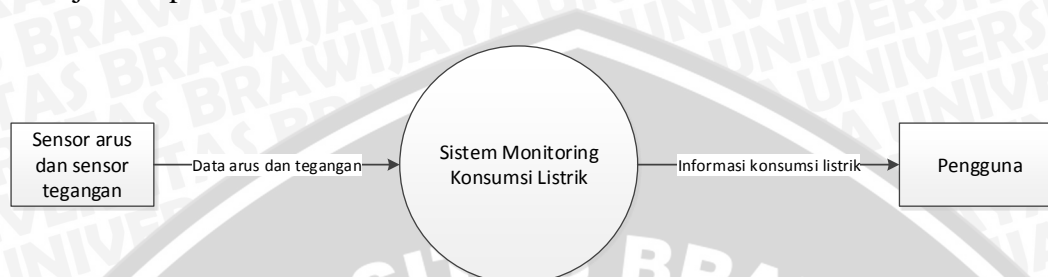


Gambar 4.2 Diagram Alir Perancangan Sistem

Diagram alir perancangan sistem pada gambar 4.2 menunjukkan adanya 3 subproses yang meliputi *preprocessing*, *validasi data training*, dan klasifikasi MKNN. 3 subproses ini akan dijelaskan lebih detil pada subbab 4.1.10 sampai dengan 4.1.12.

4.1.4 Diagram Konteks

Diagram konteks digunakan untuk menggambarkan interaksi aktor dengan sistem secara garis besar. Diagram konteks dari sistem pemantau konsumsi listrik ditunjukkan pada Gambar 4.3.

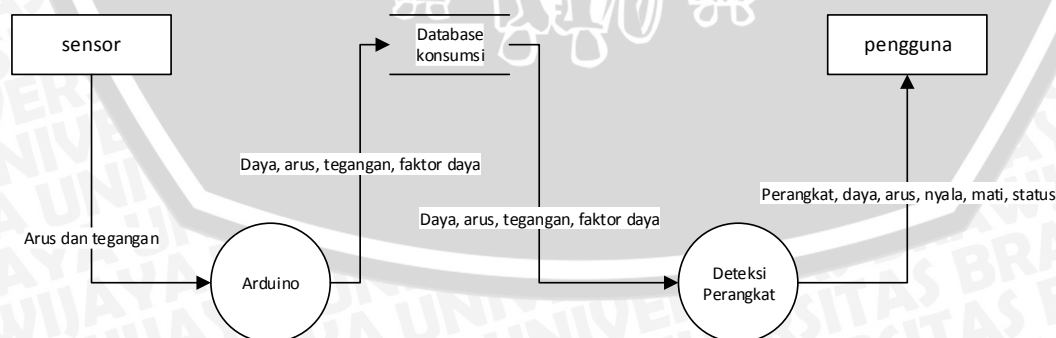


Gambar 4.3 Diagram Konteks

Aktor yang terlibat pada sistem ini adalah sensor arus dan juga pengguna aplikasi. Sensor arus bertindak sebagai perangkat pengambilan data yang mengirimkan data arus listrik ke *server* sistem monitoring konsumsi listrik. Data yang diterima dari sensor akan diproses oleh sistem untuk kemudian disajikan ke pengguna dalam bentuk informasi konsumsi listrik kapanpun pengguna mengakses *website*.

4.1.5 Diagram Aliran Data

Pada bagian ini dijelaskan bagaimana data mengalir dan berubah bentuk mulai dari sensor sampai akhirnya dapat dibaca oleh pengguna melalui aplikasi web. Gambar 4.4 menggambarkan aliran dan perubahan data mulai dari sensor sampai ke hadapan pengguna sistem.



Gambar 4.4 Diagram Aliran Data

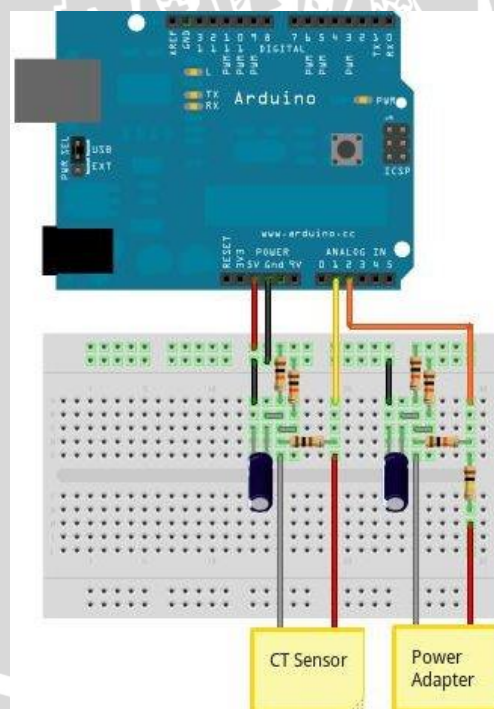
Keterangan dari Gambar 4.4 adalah sebagai berikut:

1. Data analog hasil pembacaan sensor dikirim ke Arduino

2. *Analog to Digital Converter* pada Arduino mengubah data analog menjadi data digital
3. Kemudian Arduino mengirim data digital dalam bentuk *http request*. Data arus, daya, dan tegangan dikirim ke *database* dengan tipe float
4. Data dari *database* dengan tipe float diambil untuk melakukan pendeteksian perangkat
5. Hasil deteksi perangkat ditampilkan ke pengguna berupa nama perangkat, daya, arus, waktu nyala, waktu mati, dan status

4.1.6 Perancangan Rangkaian Elektronik

Berikut adalah skema elektronik dari alat pemantau konsumsi listrik. Rangkaian elektronika ini bertindak sebagai konverter untuk mengubah sinyal arus listrik menjadi sinyal tegangan. Gambar 4.5 merupakan rancangan rangkaian elektronik yang digunakan pada penelitian ini.



Gambar 4.5 Rangkaian Elektronik

Alat-alat yang digunakan untuk menyusun alat pemantau adalah sebagai berikut :

1. Arduino, mikrokontroler sebagai unit pemroses data analog ke digital dan juga mengirim data ke *server*

2. Arduino Ethernet Shield, alat untuk menghubungkan Arduino dengan internet

Rangkaian elektronika / konverter untuk pengukuran arus :

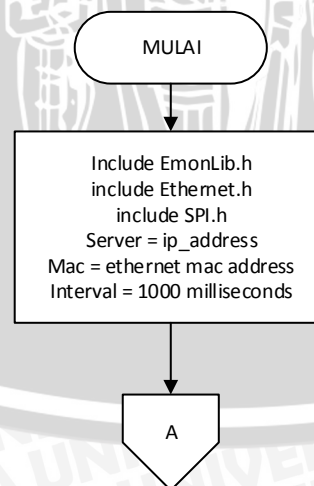
1. *CT sensor*, sensor untuk memantau arus listrik
2. 1 buah resistor 33 Ohm untuk suplai tegangan 5V
3. 2 buah resistor 10 kOhm sebagai pemisah tegangan
4. 1 buah kapasitor 10 μ F

Rangkaian elektronika / konverter pengukur tegangan :

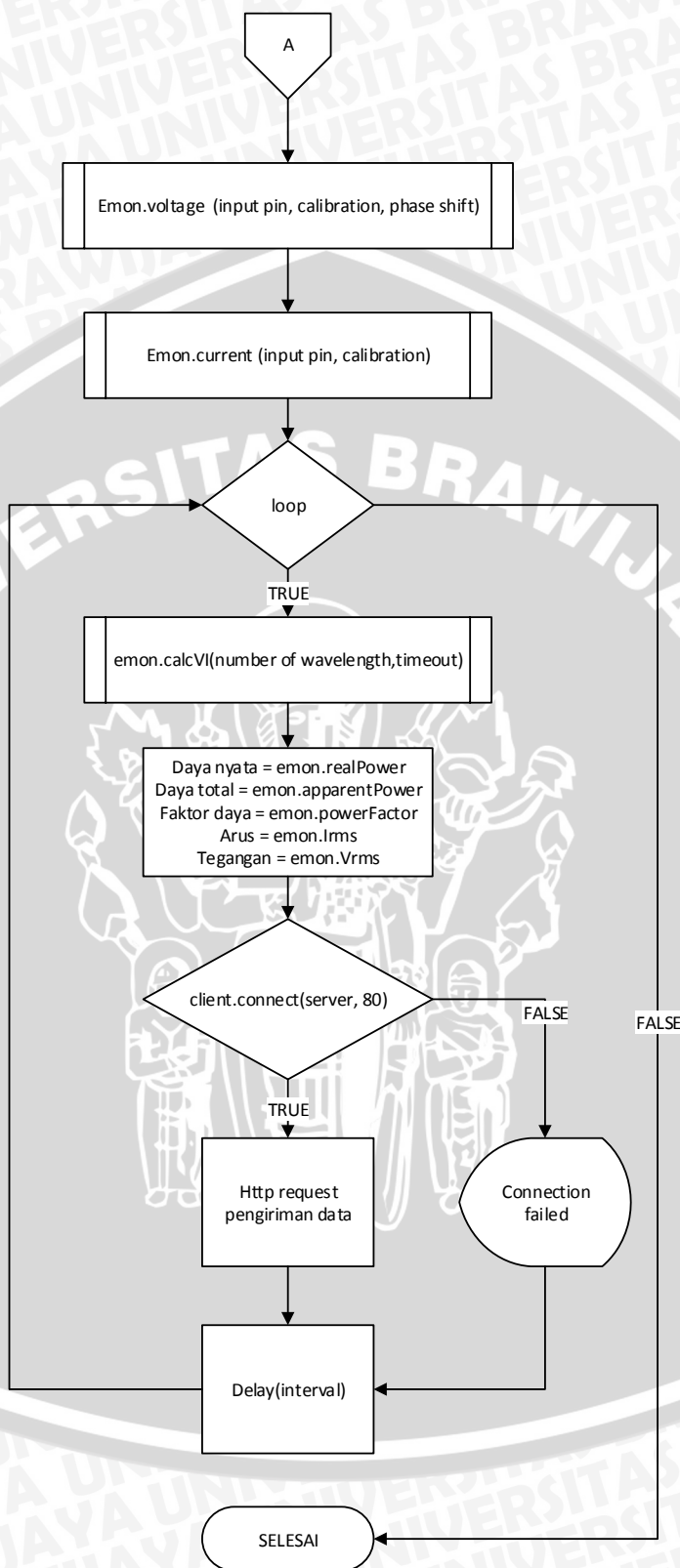
1. *Voltage sensor*, transformator yang bertindak sebagai sensor tegangan
2. 1 buah resistor 100 kOhm
3. 3 buah resistor 10 kOhm
4. 1 buah kapasitor 10 μ F

4.1.7 Pemrograman Mikrokontroler

Penelitian ini menggunakan *library* dari OpenEnergyMonitor yang dilengkapi fungsi untuk menghitung daya, arus, tegangan, dan faktor daya. Diagram pemrograman mikrokontroler dapat dilihat pada Gambar 4.6 dan Gambar 4.7.



Gambar 4.6 Pemrograman Mikrokontroler



Gambar 4.7 Pemrograman Mikrokontroler

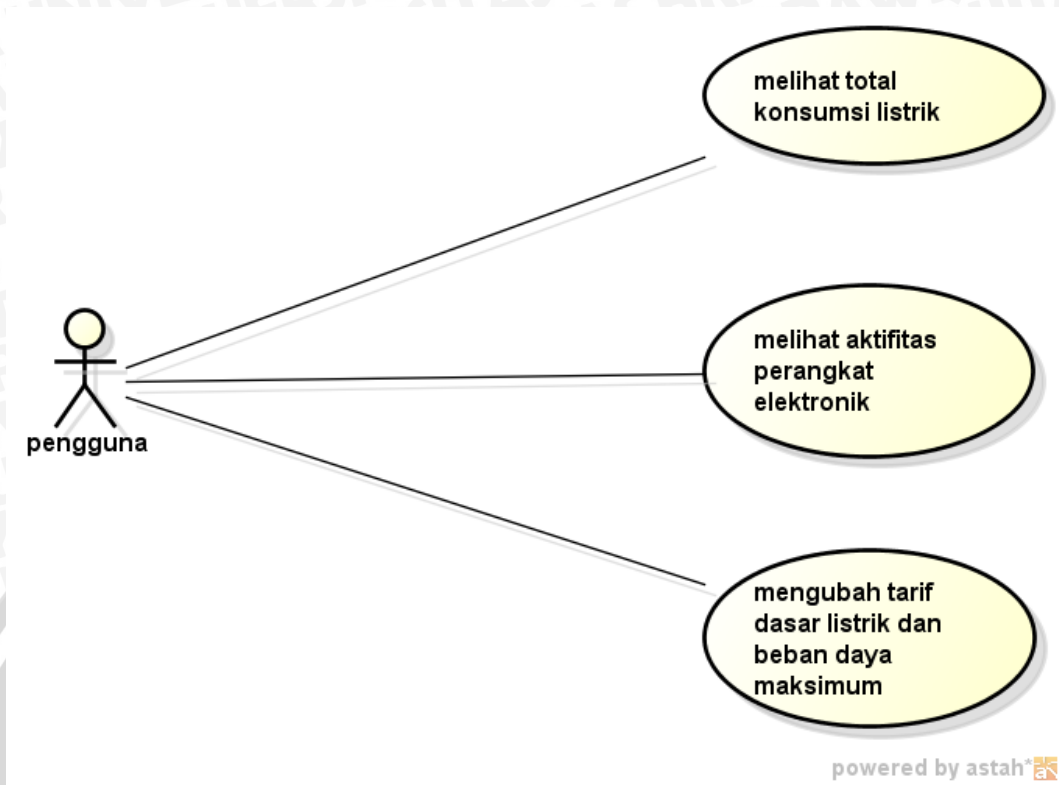
Adapun proses yang dijalankan dalam pemrograman mikrokontroler meliputi:

1. Pengaturan mikrokontroler dalam proses ini meliputi penggunaan *library* Open Energy Monitor untuk memproses data dari sensor, penentuan *ip address server*, penentuan *mac address Ethernet*, penentuan *input pin* untuk arus, penentuan *input pin* tegangan.
2. Perhitungan daya nyata, daya total, dan juga faktor daya yang dilakukan dengan menggunakan fasilitas *library* Open Energy Monitor.
3. Pengiriman data ke *database* dilakukan dengan mengirimkan *http request* ke *server*. Berikut merupakan pesan yang dikirimkan ke *server*.

```
POST/energon/index.php/home/push_data/daya_nyata/daya_t
otal/arus/tegangan/factor_daya
HTTP/1.1
Host:
ip_address_server
Connection: Close
```

4.1.8 Perancangan Diagram *Use Case*

Diagram *use case* menjelaskan interaksi antara aktor dan sistem pemantau konsumsi listrik. Aktor pada sistem ini adalah pengguna/pemilik rumah. Interaksi aktor dan sistem meliputi kegiatan memantau total konsumsi listrik, aktifitas perangkat elektronik, dan juga mengubah nilai tarif dasar listrik serta beban daya maksimal. Diagram *use case* sistem monitoring konsumsi listrik berbasis web ditunjukkan oleh Gambar 4.8.



powered by astah[®]

Gambar 4.8 Diagram *Use Case*

4.1.8.1 Skenario *Use Case* Melihat Total Konsumsi Listrik

Skenario *use case* ini akan menjelaskan secara detail proses melihat total konsumsi listrik. Skenario *use case* untuk melihat total konsumsi listrik dijabarkan pada Tabel 4.6.

Tabel 4.6 Skenario *Use Case* Melihat Total Konsumsi Listrik

| Skenario Kasus pada Sistem | |
|---------------------------------------|--|
| Nomor <i>Use Case</i> | FR01 |
| Nama | Melihat total konsumsi listrik |
| Tujuan | Untuk melihat total konsumsi listrik |
| Deskripsi | <i>Use Case</i> ini menjelaskan bagaimana pengguna dapat memantau total pemakaian listriknya |
| Skenario Utama | |
| Kondisi Awal | Pengguna telah terdaftar dalam sistem |
| Aksi Aktor | Reaksi Sistem |
| Pengguna <i>login</i> ke dalam system | Sistem mengecek <i>database</i> data diri pengguna |



| | |
|---------------|---|
| Kondisi Akhir | Sistem menampilkan konsumsi listrik seluruh rumah dalam satuan kwh dan Rupiah |
|---------------|---|

4.1.8.2 Skenario *Use Case* Melihat Aktifitas Perangkat Elektronik

Skenario *use case* ini akan menjelaskan secara detil proses melihat aktifitas perangkat elektronik. Skenario *use case* untuk melihat aktifitas perangkat elektronik dijabarkan pada Tabel 4.7.

Tabel 4.7 Use Case Melihat Aktifitas Perangkat Elektronik

| Skenario Kasus pada Sistem | |
|---------------------------------------|---|
| Nomor <i>Use Case</i> | FR02 |
| Nama | Melihat aktifitas perangkat elektronik |
| Tujuan | Untuk melihat aktifitas perangkat elektronik |
| Deskripsi | <i>Use Case</i> ini menjelaskan bahwa pengguna dapat melihat aktifitas perangkat elektronik yang sedang menyala atau mati |
| Skenario Utama | |
| Kondisi Awal | Pengguna telah terdaftar dalam sistem |
| Aksi Aktor | Reaksi Sistem |
| Pengguna <i>login</i> ke dalam system | Mengecek <i>database</i> data diri pengguna |
| Kondisi Akhir | Pengguna perangkat elektronik yang sedang menyala atau mati beserta biaya yang dikeluarkan dalam satu hari |

4.1.8.3 Skenario *Use Case* Mengubah Tarif Dasar Listrik dan Beban Daya Maksimum

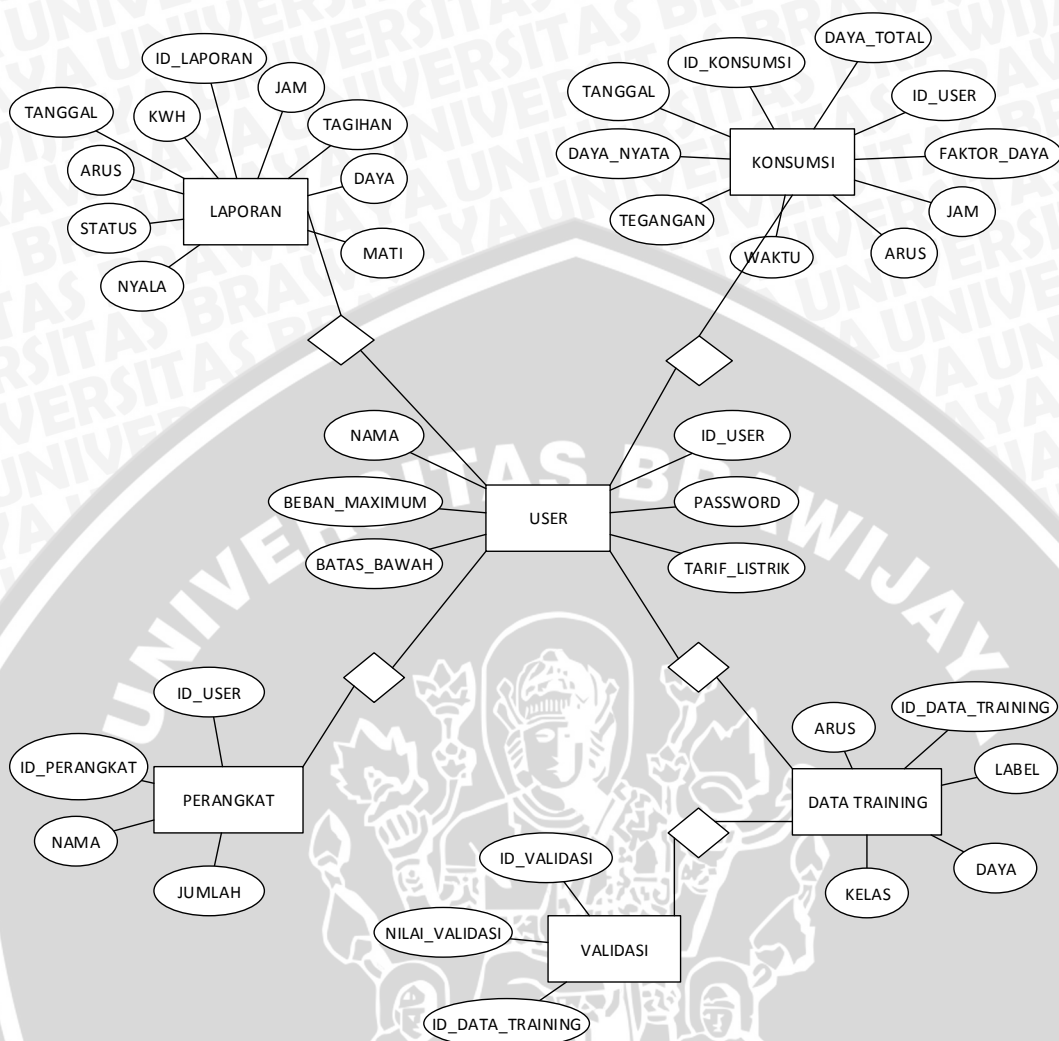
Skenario *use case* ini akan menjelaskan secara detil proses mengubah tarif dasar listrik dan beban daya maksimum. Skenario *use case* untuk melihat proses mengubah tarif dasar listrik dan beban daya maksimum dijabarkan pada Tabel 4.8.

Tabel 4.8 Skenario *Use Case* Mengubah Tarif Dasar Listrik dan Beban Daya Maksimum

| Skenario Kasus pada Sistem | |
|------------------------------------|---|
| Nomor <i>Use Case</i> | FR03 |
| Nama | Proses mengubah tarif dasar listrik dan beban daya maksimum |
| Tujuan | Untuk mengubah tarif dasar listrik dan beban daya maksimum |
| Deskripsi | <i>Use Case</i> ini menjelaskan bahwa pengguna dapat mengubah tarif dasar listrik dan beban daya maksimum |
| Skenario Utama | |
| Kondisi Awal | Admin berhasil login, data diri pengunjung sudah disimpan pada database |
| Aksi Aktor | Reaksi Sistem |
| Pengguna menekan tombol pengaturan | Menyajikan halaman untuk mengubah tarif dasar listrik dan beban daya maksimum |
| Kondisi Akhir | Pengguna memasukkan data baru |

4.1.9 Perancangan *Database*

Perancangan basis data bertujuan untuk menjaga data berelasi antara satu dengan lainnya. Relasi antar elemen sangatlah penting guna menjaga keterkaitan data sehingga memudahkan peneliti untuk mengelompokkan dan menemukan data yang diperlukan. Pada skripsi ini perancangan basis data direpresentasikan dalam bentuk *Entity Relationship Diagram* (ERD). ERD menunjukkan hubungan yang terjadi diantara objek (entitas) yang terlibat dalam suatu *database*. ERD berisi komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan beberapa atribut yang mempresentasikan seluruh fakta yang ditinjau dari keadaan yang nyata. Atribut dapat membedakan tabel satu dengan tabel lainnya pada perancangan *database*. Rancangan ERD dapat dilihat pada Gambar 4.9.



Gambar 4.9 Entity Relational Database Sistem Monitoring Konsumsi Listrik berbasis Web

1. Tabel *User*

Tabel ini digunakan untuk menyimpan data pengguna sistem pemantau konsumsi listrik. Rancangan tabel *user* ditunjukkan oleh Tabel 4.9.

Tabel 4.9 Database *User*

| No | Atribut | Tipe Data | Keterangan |
|----|---------------------|-------------|---------------|
| 1 | <i>id_user</i> | Int(11) | Primary Key |
| 2 | Nama | Varchar(20) | Nama pengguna |
| 3 | <i>Password</i> | Varchar(20) | Kata sandi |
| 4 | Tarif dasar listrik | Float | Tarif listrik |

2. Tabel Konsumsi listrik

Tabel ini menyimpan data hasil pengamatan yang dilakukan oleh sensor arus. Data konsumsi listrik akan dimasukkan ke tabel ini setiap detik.

Rancangan tabel konsumsi listrik ditunjukkan oleh Tabel 4.10.

Tabel 4.10 Database Konsumsi Listrik

| No | Atribut | Tipe Data | Keterangan |
|----|---------------------|--------------|-------------|
| 1 | Id_konsumsi_listrik | Int(11) | Primary Key |
| 2 | Id_user | Varchar(200) | Foreign key |
| 3 | Timestamp | timestamp | Waktu input |
| 4 | Daya Nyata | Float | |
| 5 | Daya Total | Float | |
| 6 | Arus | Float | |
| 7 | Tegangan | Float | |
| 8 | Faktor Daya | Float | |
| 9 | Read | Enum(0,1) | Status data |

3. Tabel Validasi

Tabel ini menyimpan hasil perhitungan validasi data training. Rancangan tabel validasi *data training* ditunjukkan oleh Tabel 4.11.

Tabel 4.11 Database Validasi Data Training

| No | Atribut | Tipe Data | Keterangan |
|----|------------------|-------------|----------------|
| 1 | Id_validasi | Int(11) | Primary Key |
| 2 | Id_data_training | Varchar(20) | Foreign Key |
| 3 | Validasi | Float | Nilai validasi |

4. Tabel Laporan Konsumsi Listrik

Tabel ini menyimpan laporan konsumsi listrik tiap jam dari setiap perangkat elektronik. Rancangan tabel laporan konsumsi listrik ditunjukkan oleh Tabel 4.12.

Tabel 4.12 Database Laporan Konsumsi Listrik

| No | Atribut | Tipe Data | Keterangan |
|----|------------|-------------|-----------------------|
| 1 | Id_laporan | Int(11) | Primary Key |
| 2 | Id_user | Varchar(20) | Nama pengguna |
| 3 | Perangkat | Varchar(20) | Nama perangkat |
| 4 | Tanggal | Date | Waktu Pencatatan |
| 5 | kWh | Float | Konsumsi Daya per Jam |
| 6 | Jam | Int | Jam pencatatan |
| 7 | Tagihan | Float | Jumlah tagihan |
| 8 | Daya | Float | Daya listrik |
| 9 | Arus | Float | Arus listrik |
| 10 | Nyala | Time | Waktu nyala |
| 11 | Mati | Time | Waktu mati |
| 12 | Status | Enum | Status alat |

5. Tabel Data Training

Tabel ini menyimpan data latihan yang akan digunakan untuk mendeteksi perangkat elektronik yang sedang aktif. Rancangan tabel *data training* ditunjukkan oleh Tabel 4.13.

Tabel 4.13 Database Data Training

| No | Atribut | Tipe Data | Keterangan |
|----|------------------|-------------|--------------------|
| 1 | Id_data_training | Int(11) | Primary Key |
| 2 | Id_perangkat | Varchar(20) | Id Perangkat |
| 3 | Daya | Varchar(20) | Konsumsi Daya |
| 4 | Kelas | Varchar(20) | Kategori Perangkat |

6. Tabel Perangkat

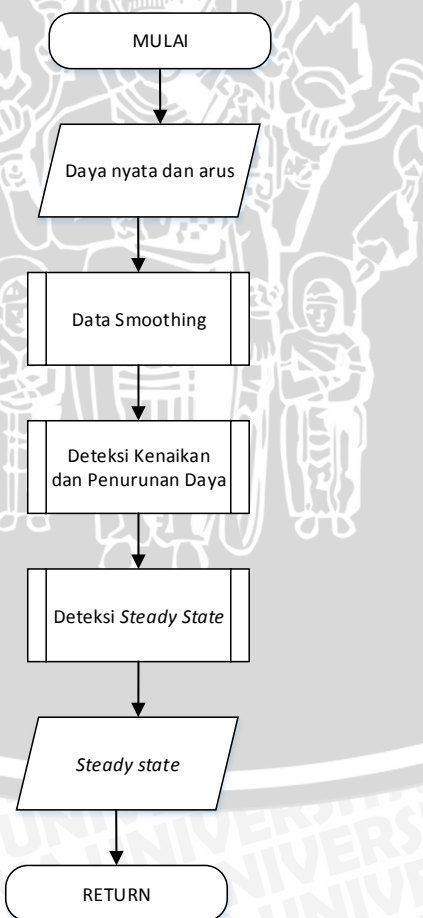
Tabel ini menyimpan data perangkat elektronik yang ada di rumah. Rancangan tabel perangkat ditunjukkan oleh Tabel 4.14.

Tabel 4.14 Database Perangkat Elektronik

| No | Atribut | Tipe Data | Keterangan |
|----|--------------|-------------|------------------|
| 1 | Id_perangkat | Int(11) | Primary Key |
| 2 | User id | Varchar(20) | Id pengguna |
| 3 | Nama | Varchar(20) | Nama perangkat |
| 4 | Jumlah | Varchar(20) | Jumlah perangkat |

4.1.10 Perancangan Proses *Preprocessing*

Pada tahap ini dilakukan pengolahan data daya dan arus listrik dengan menggunakan metode klasifikasi MKNN untuk mendeteksi aktifitas perangkat elektronik. Perancangan algoritma pendeteksian aktifitas perangkat elektronik dapat dilihat pada Gambar 4.10.

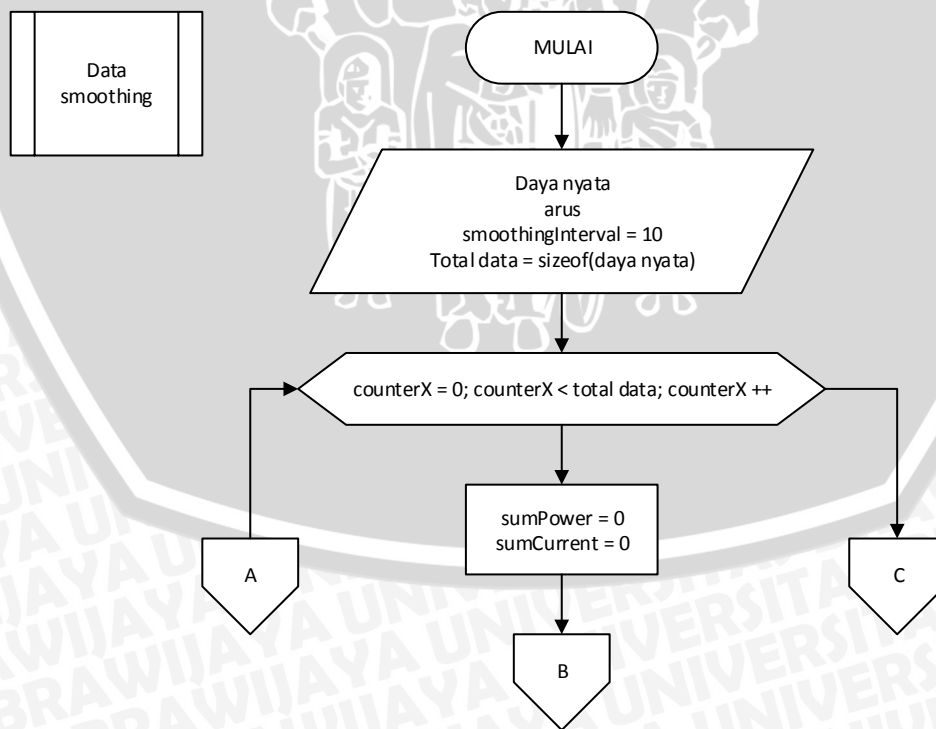


Gambar 4.10 Diagram Alir *Preprocessing*

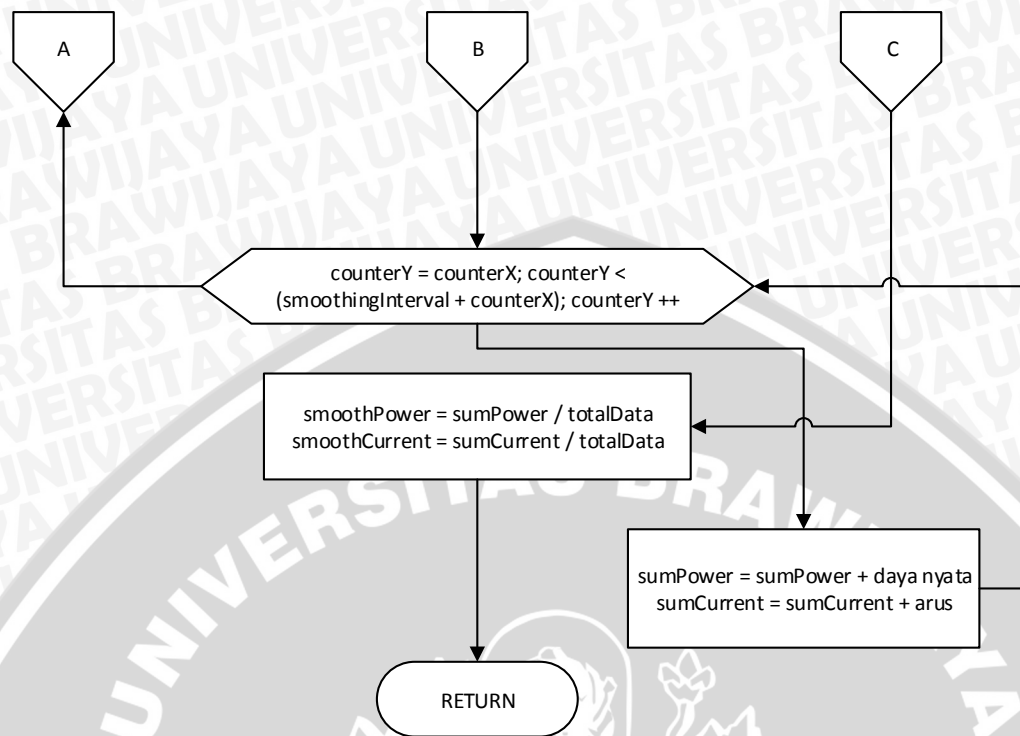
Proses pendeteksian perangkat dimulai dengan proses *smoothing* data arus dan daya. *Smoothing* diperlukan untuk mengurangi *noise*. Hasil *smoothing* digunakan untuk mendeteksi kenaikan dan penurunan daya yang diduga sebagai aktifitas menyalakan atau mematikan perangkat elektronik. Setelah aktifitas terdeteksi, dilanjutkan dengan pendeteksian *steady state*. Nilai daya dan arus pada *steady state* akan digunakan sebagai inputan bagi MKNN untuk menentukan perangkat apa yang sedang menyala atau mati. Subproses *data smoothing*, deteksi kenaikan dan penurunan daya, deteksi *steady state* akan dibahas lebih mendalam pada subbab 4.1.10.1 sampai dengan 4.1.10.3.

4.1.10.1 Proses Data Smoothing

Proses *data smoothing* dilakukan untuk mengurangi *noise* pada data. Berkurangnya *noise* akan memudahkan dalam proses pencarian *steady state*. Adapun metode yang digunakan adalah *moving average*. Pada proses ini dilakukan pengambilan nilai rata-rata daya dan arus dengan interval 10 data. Untuk lebih jelas proses *data smoothing* bisa dilihat pada gambar. Perancangan algoritma *data smoothing* dapat dilihat pada Gambar 4.11 dan Gambar 4.12.



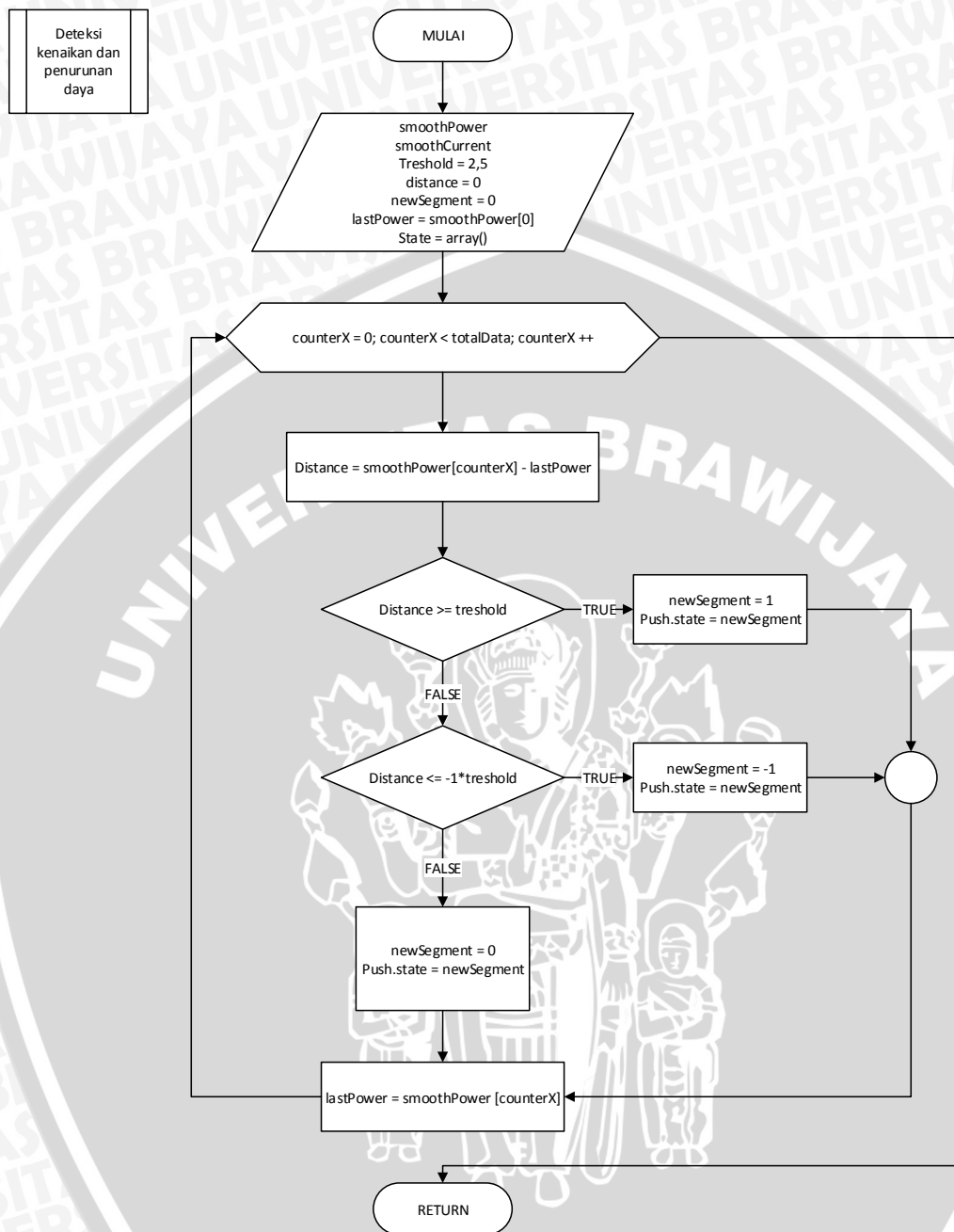
Gambar 4.11 Proses Data Smoothing



Gambar 4.12 Proses *Data Smoothing*

4.1.10.2 Proses Deteksi Kenaikan dan Penurunan Daya

Pada tahap ini data yang telah melalui proses *data smoothing* akan dianalisa untuk mencari kenaikan daya, penurunan daya, dan *steady state*. Untuk itu penulis menentukan selisih daya dengan nilai 2,5 watt sebagai ambang batas yang menentukan terjadinya perubahan daya. Hal ini diperlukan untuk meminimalisir kesalahan deteksi karena data yang terlalu fluktuatif. Keluaran dari proses ini berupa pelabelan terhadap data konsumsi listrik, 1 menandakan kenaikan daya, 0 menandakan *steady state*, dan -1 menandakan penurunan daya. informasi *segmentdata* yang tergolong ke dalam *steady state*. Diagram alir proses untuk mendeteksi kenaikan dan penurunan daya dapat dilihat pada Gambar 4.13.

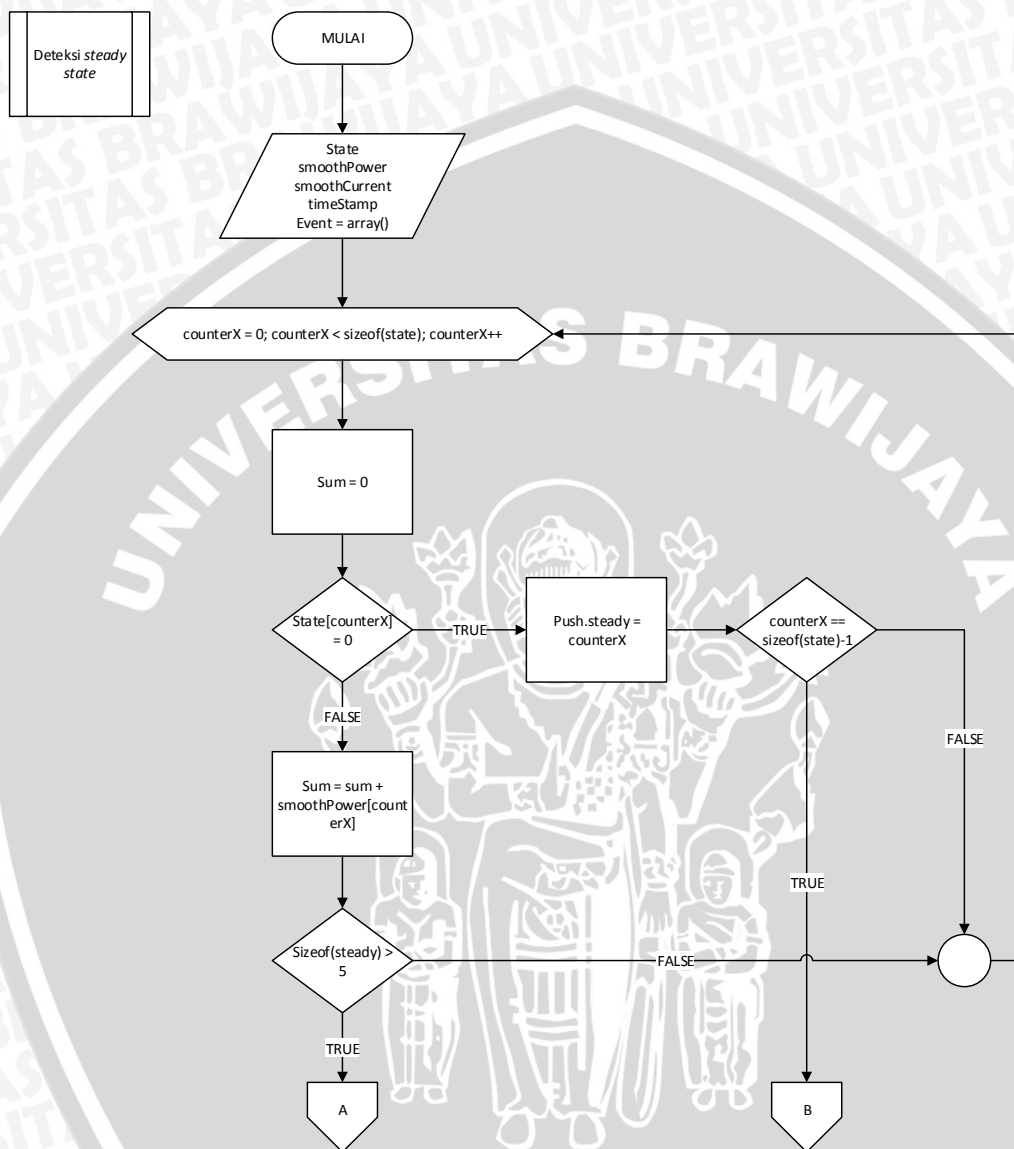


Gambar 4.13 Proses Deteksi Kenaikan dan Penurunan Daya

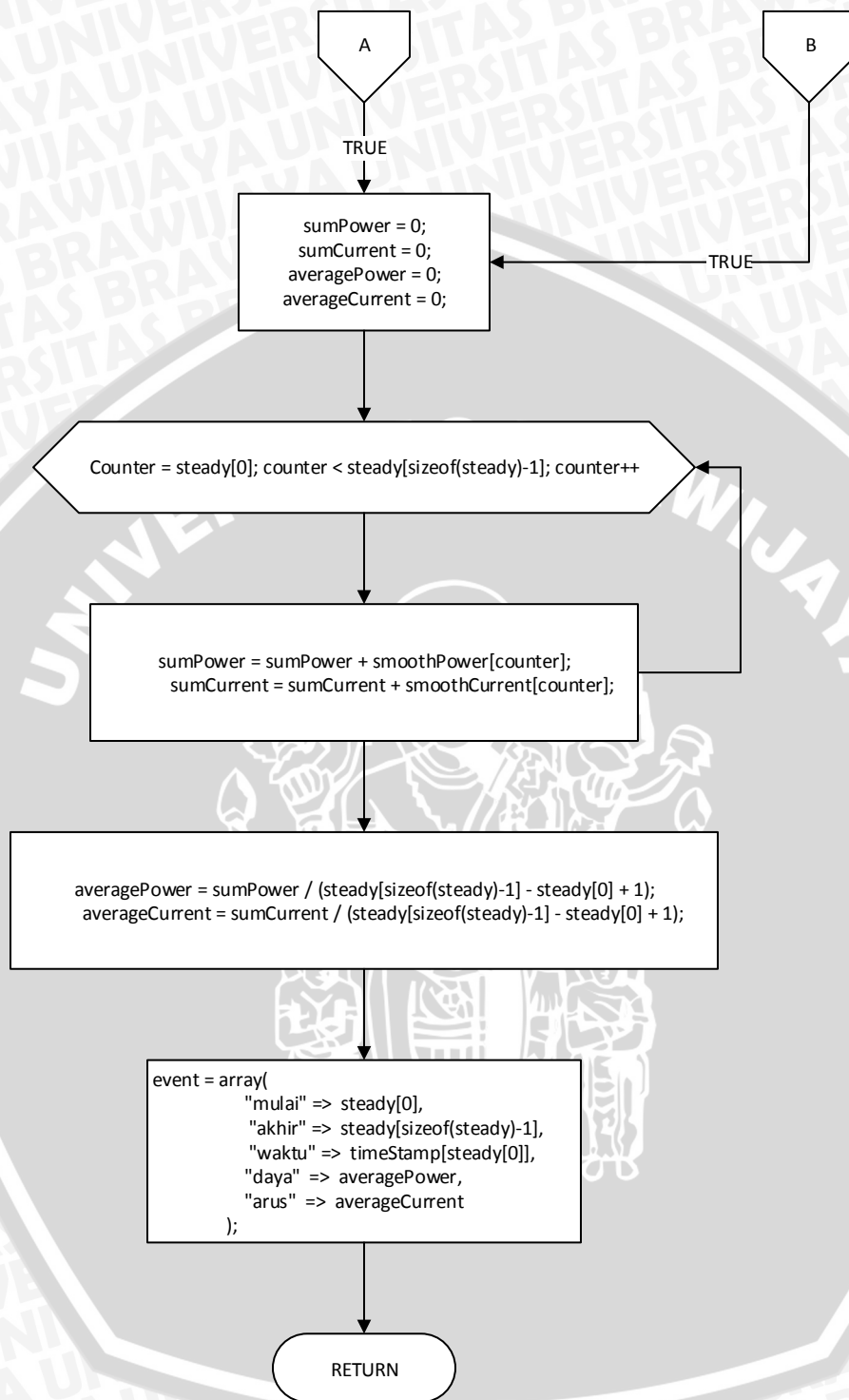
4.1.10.3 Proses Deteksi Steady State

Proses ini bertujuan untuk mencari segmen data yang termasuk dalam golongan *steady state*. *Steady state* inilah yang akan diklasifikasikan untuk mengetahui alat apa yang sedang menyala atau mati. Keluarannya berupa informasi titik awal terjadinya *steady state*, titik akhir terjadinya *steady state*, rata-

rata daya, rata-rata arus, dan waktu terjadinya *steady state*. Diagram alir proses deteksi *steady state* dapat dilihat pada Gambar 4.14 dan Gambar 4.15.



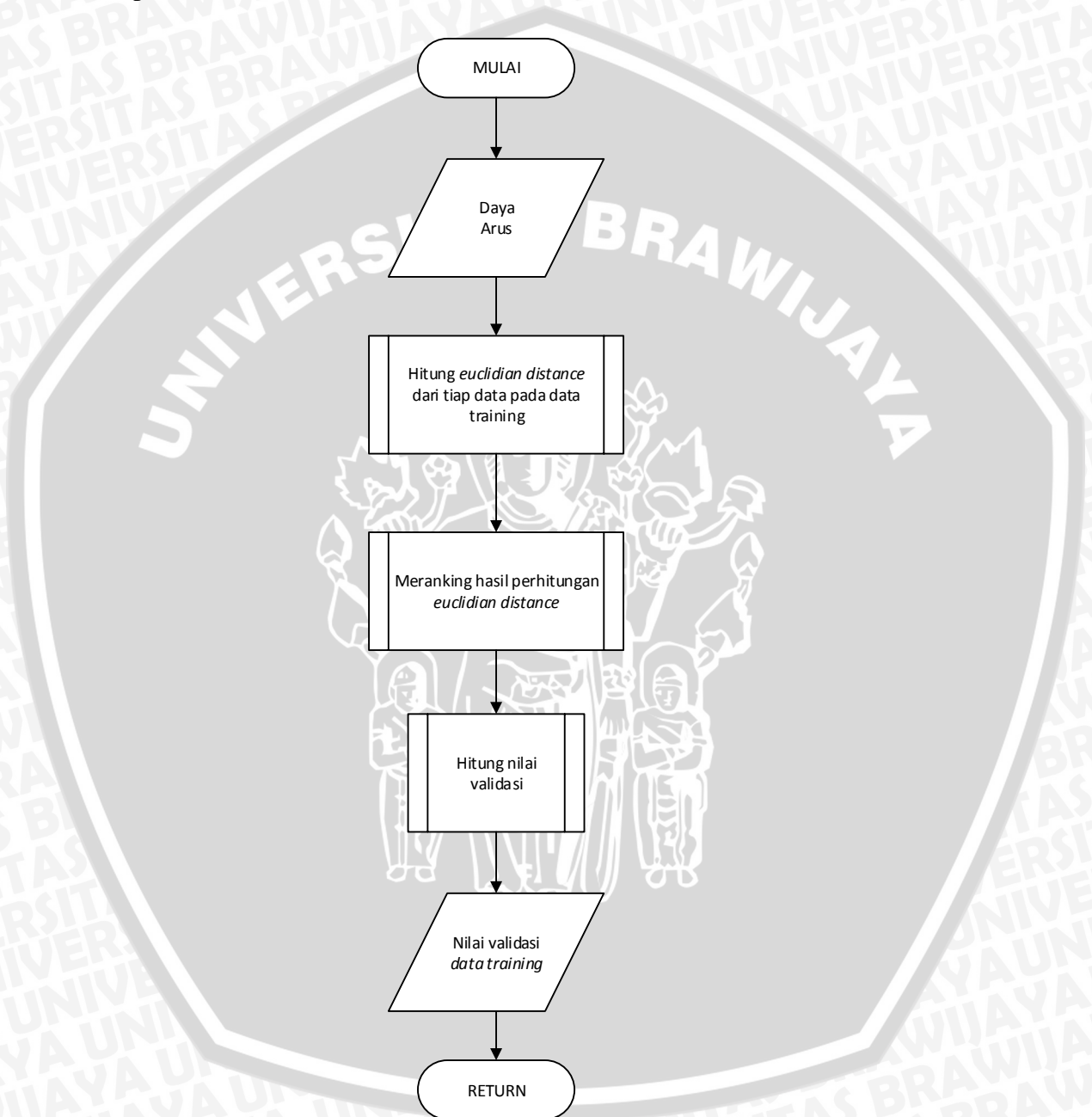
Gambar 4.14 Proses Deteksi *Steady State*



Gambar 4.15 Proses Deteksi *Steady State*

4.1.11 Perancangan Proses Validasi *Data Training*

Proses ini bertujuan untuk mencari nilai validasi dari dataset yang akan dijadikan sebagai *data training*. Diagram alir proses validasi *data training* dapat dilihat pada Gambar 4.16.

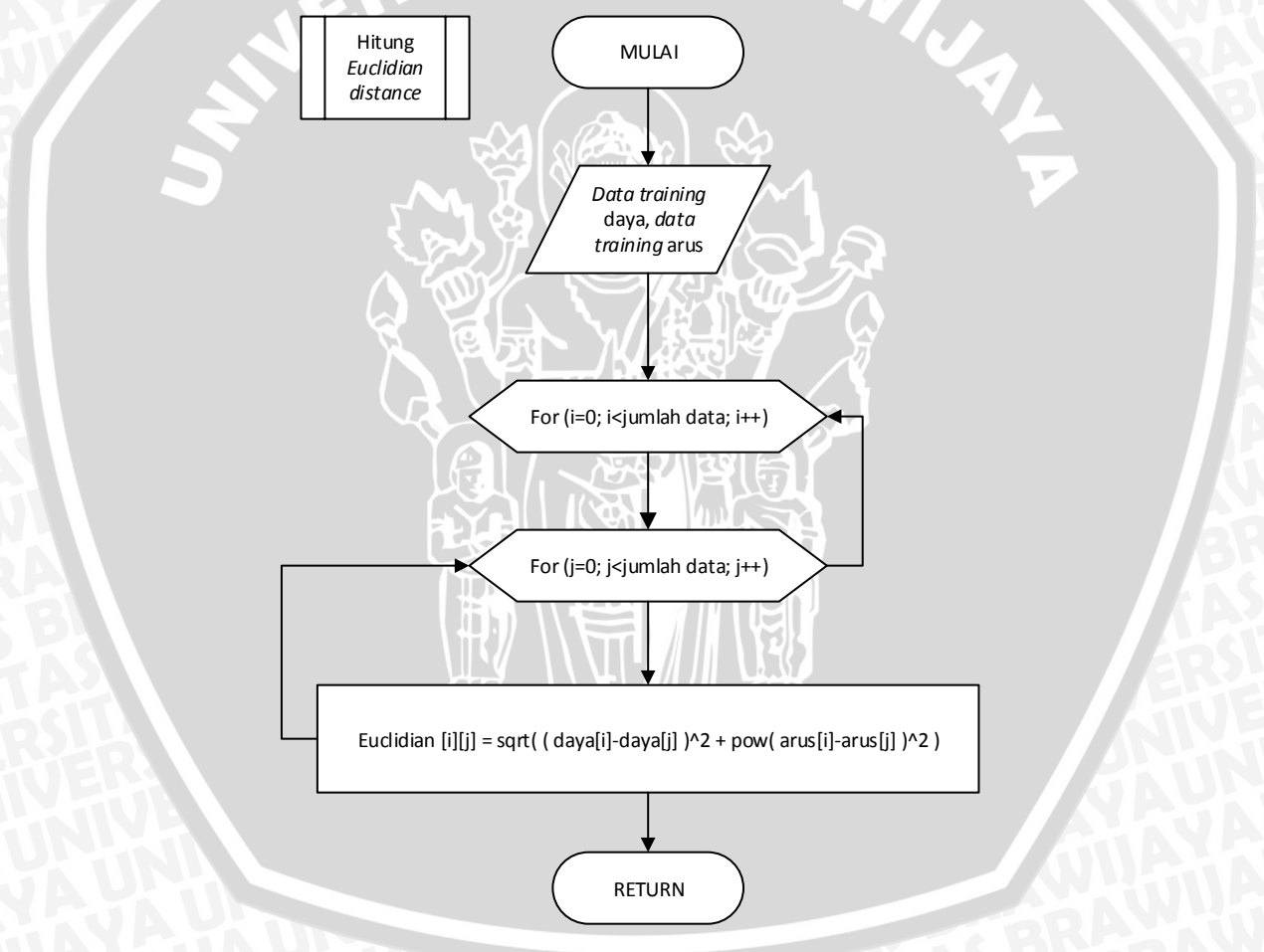


Gambar 4.16 Diagram Alir Proses Validasi Data Training

Tahap inilah yang membedakan proses KNN biasa dengan MKNN. Terdapat 3 subproses dalam proses validasi *data training* yang meliputi proses hitung *Euclidian distance* dari tiap data pada *data training*, meranking hasil perhitungan *Euclidian distance*, dan hitung nilai validasi. 3 subproses tersebut akan dijelaskan lebih detil pada subbab 4.1.11.1 sampai dengan 4.1.11.3.

4.1.11.1 Proses Hitung *Euclidian Distance*

Pada tahapan ini dilakukan penghitungan *euclidian distance* antar data pada *data training*. Inputan yang digunakan adalah daya listrik dan arus listrik. Diagram alir perhitungan *euclidian distance* dapat dilihat pada Gambar 4.17.

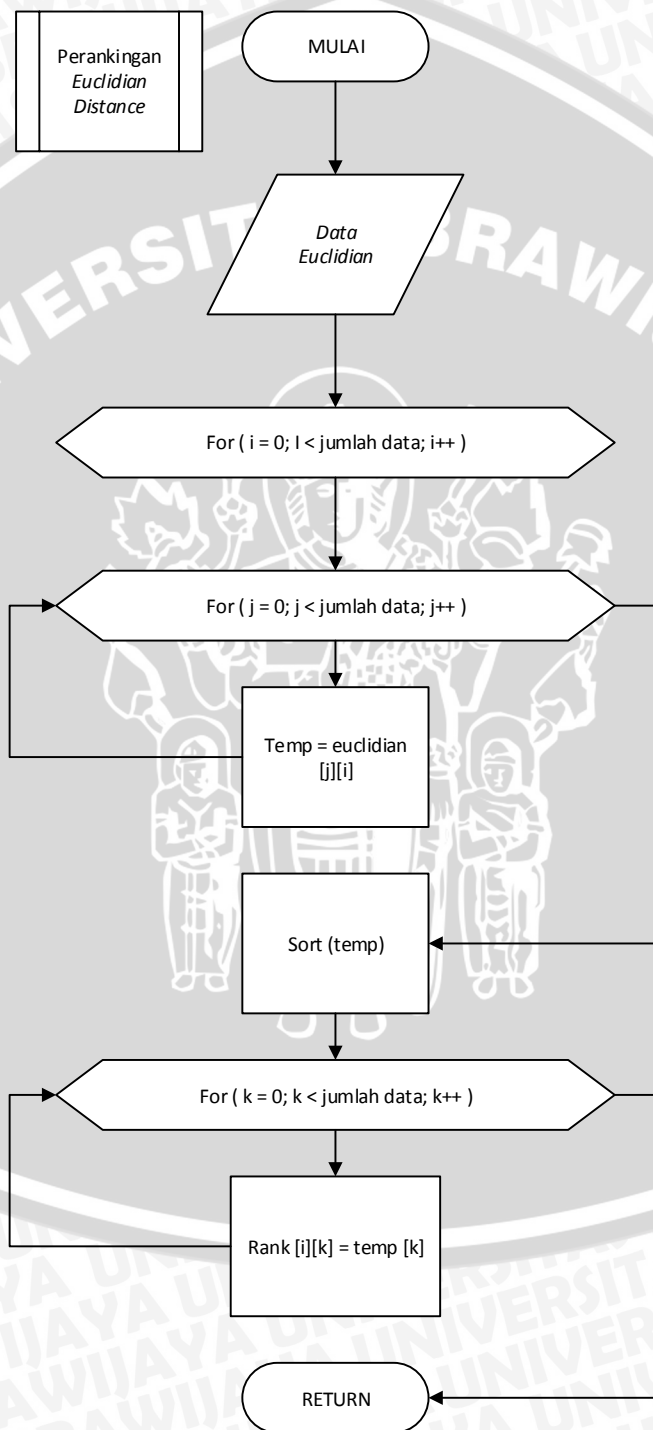


Gambar 4.17 Proses Hitung *Euclidian Distance*

4.1.11.2 Proses Perankingan Hasil *Euclidian Distance*

Pada tahapan ini dilakukan perankingan *euclidian distance*. Data *euclidian distance* akan diurutkan dari nilai yang tertinggi sampai yang terendah.

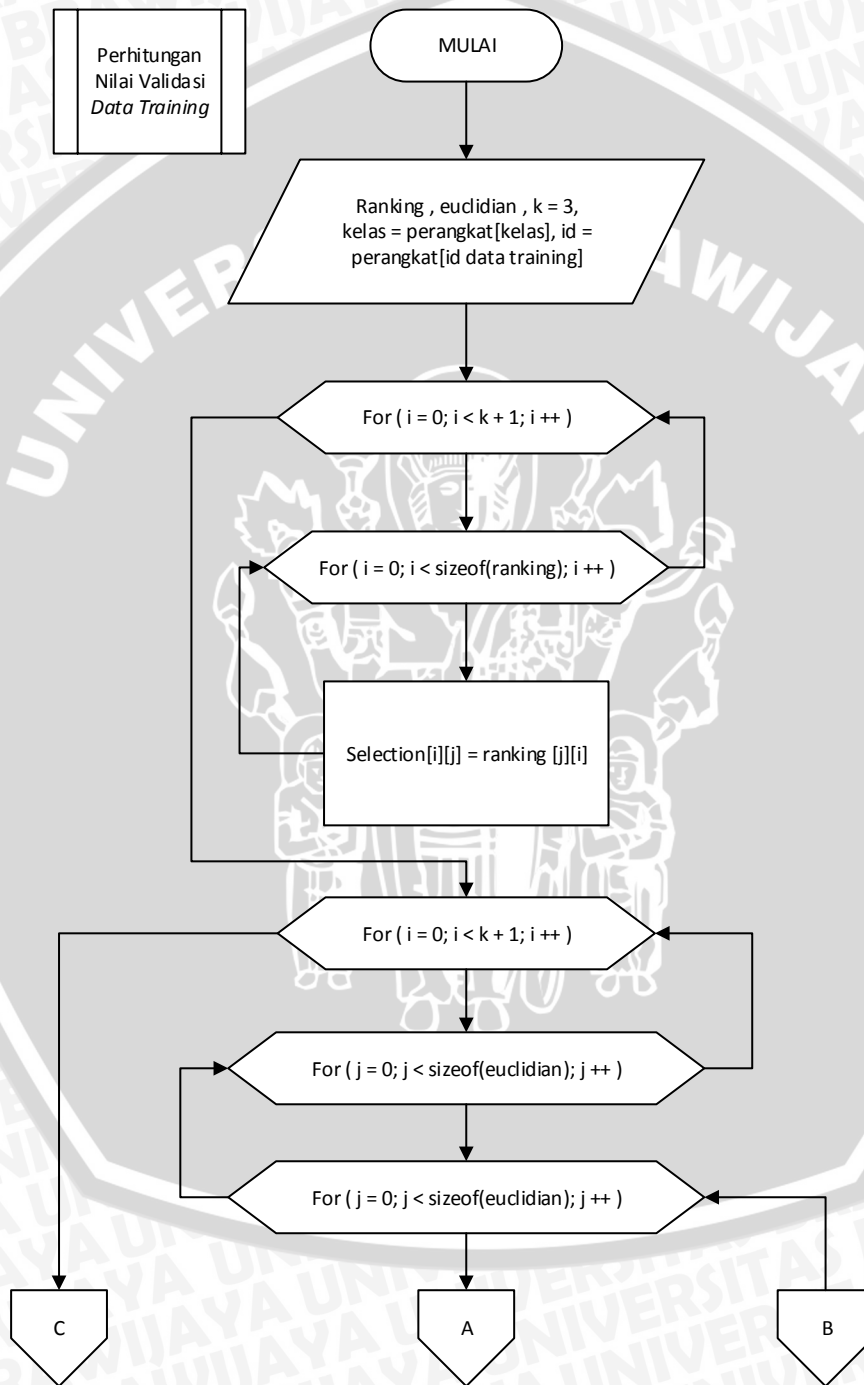
Diagram alir perhitungan *euclidian distance* dapat dilihat pada Gambar 4.18.



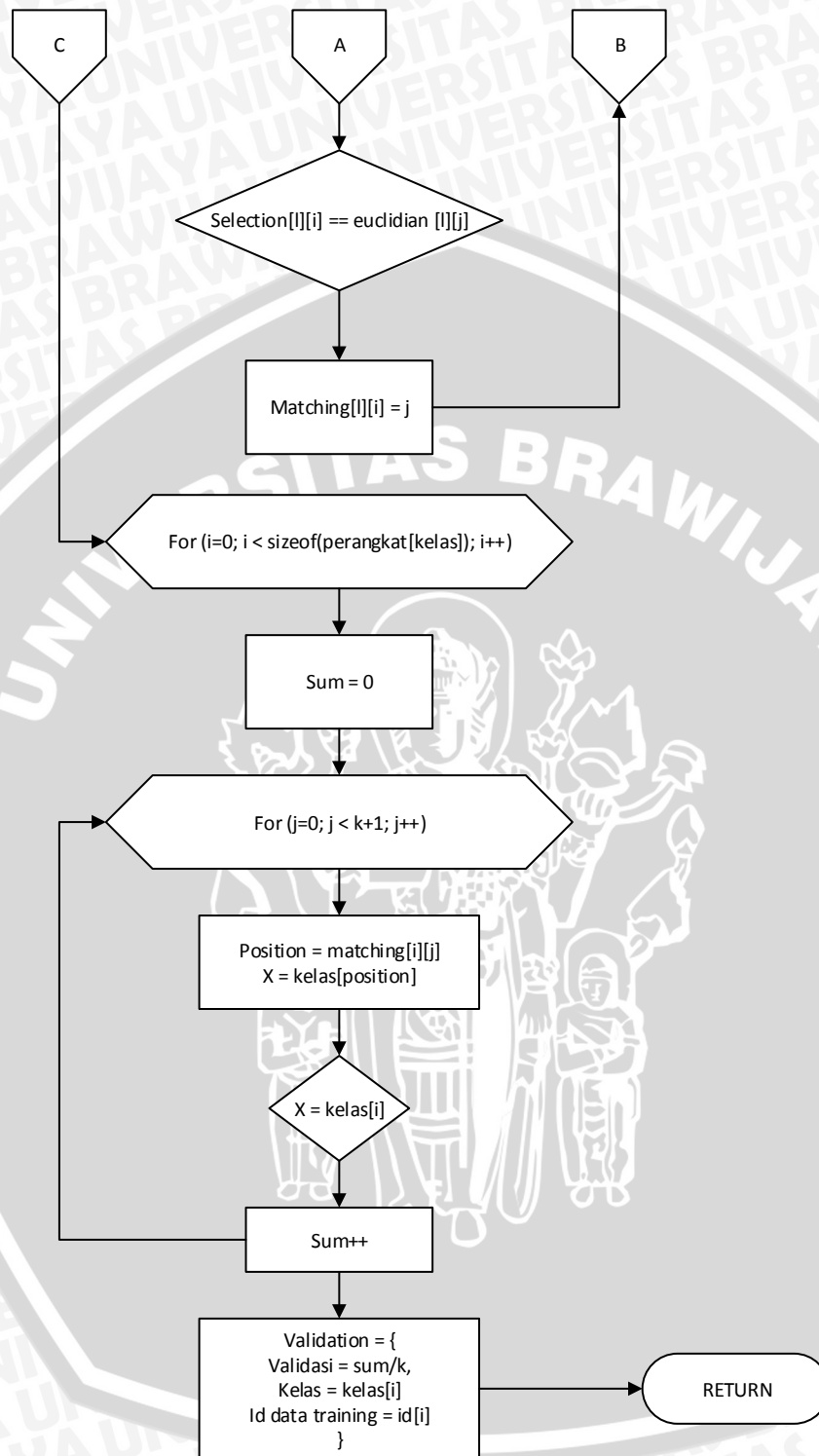
Gambar 4.18 Proses Perankingan Hasil *Euclidian Distance*

4.1.11.3 Proses Perhitungan Nilai Validasi *Data Training*

Pada proses ini dilakukan perhitungan nilai validasi. Proses perhitungan nilai validasi *data training* ditunjukkan oleh Gambar 4.19 dan Gambar 4.20 berikut ini.



Gambar 4.19 Proses Perhitungan Nilai Validasi *Data Training*

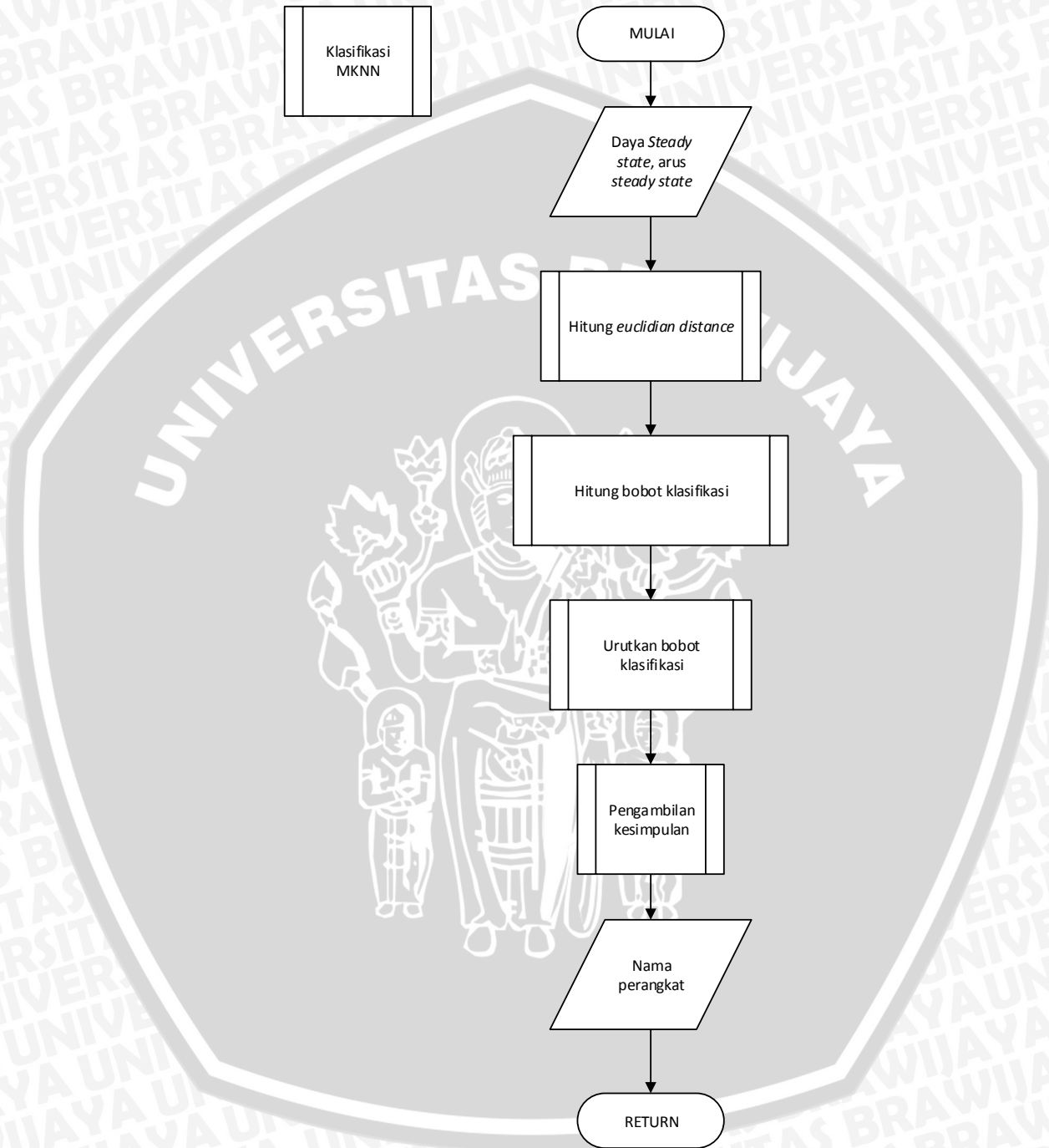


Gambar 4.20 Proses Perhitungan Nilai Validasi *Data Training*

4.1.12 Perancangan Klasifikasi MKNN

Pada tahap ini dilakukan proses klasifikasi data *steady state* yang menghasilkan keluaran berupa nama perangkat, daya, arus, waktu nyala, waktu

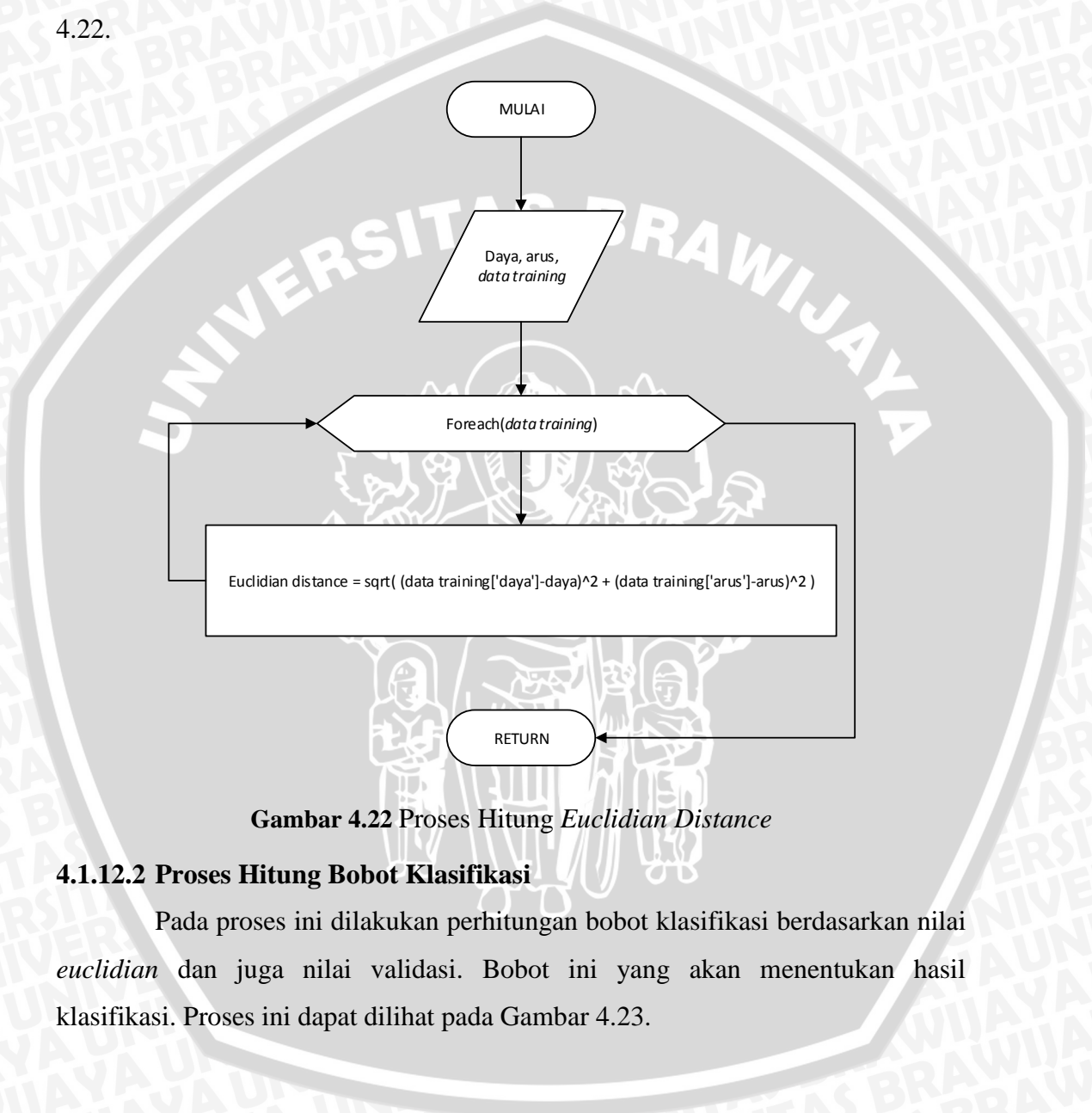
mati, dan status perangkat. Diagram alir proses klasifikasi MKNN dapat dilihat pada Gambar 4.21.



Gambar 4.21 Diagram alir proses klasifikasi MKNN

4.1.12.1 Proses Hitung *Euclidian Distance*

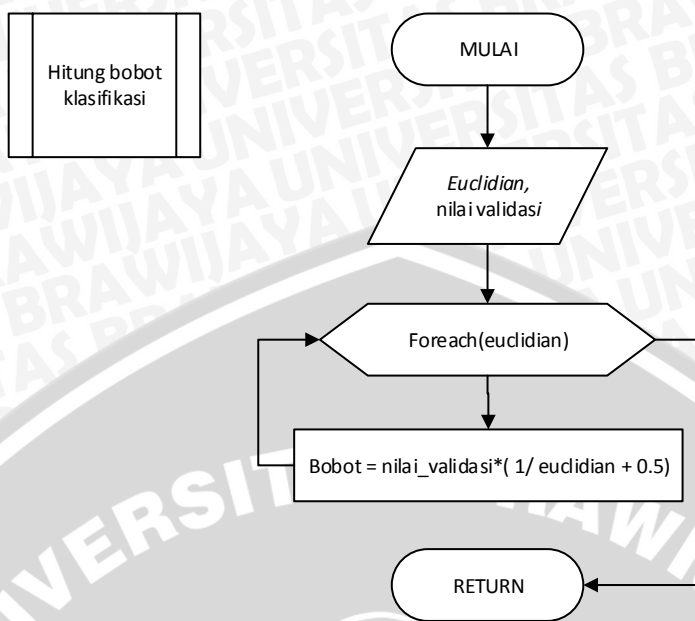
Pada tahapan ini dilakukan perhitungan *euclidian distance* antara data input dan *data training*. Perhitungan ini berguna untuk mengetahui kedekatan antar data. Diagram alir perhitungan *euclidian distance* dapat dilihat pada Gambar 4.22.



Gambar 4.22 Proses Hitung *Euclidian Distance*

4.1.12.2 Proses Hitung Bobot Klasifikasi

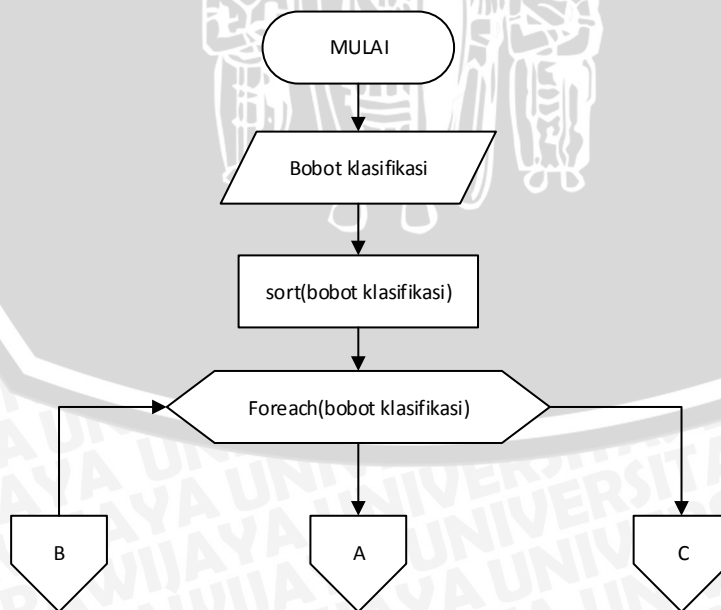
Pada proses ini dilakukan perhitungan bobot klasifikasi berdasarkan nilai *euclidian* dan juga nilai validasi. Bobot ini yang akan menentukan hasil klasifikasi. Proses ini dapat dilihat pada Gambar 4.23.



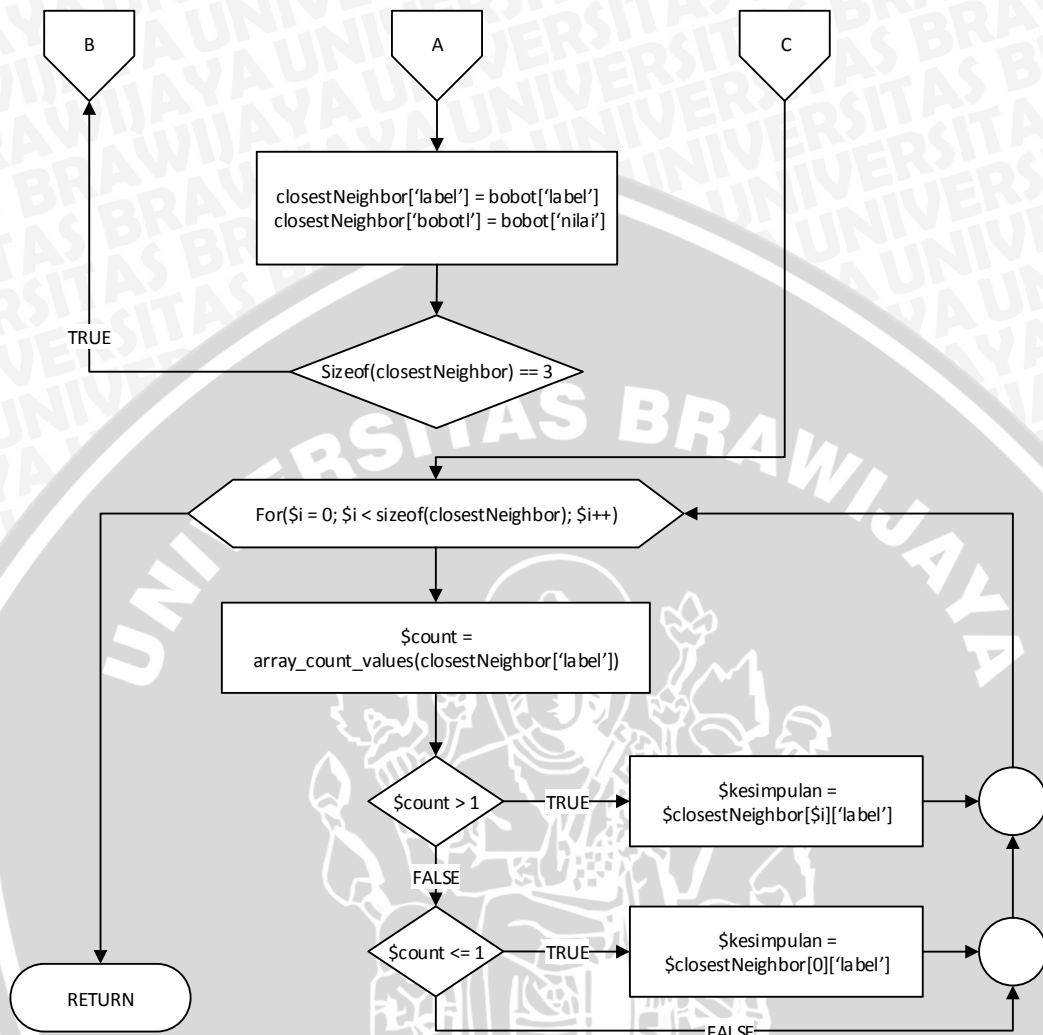
Gambar 4.23 Proses Hitung Bobot Klasifikasi

4.1.12.3 Proses Mengurutkan Bobot Klasifikasi dan Pengambilan Kesimpulan

Nilai bobot klasifikasi akan diurutkan untuk mendapatkan 3 bobot yang tertinggi. Dari 3 bobot tertinggi itu akan diambil kelas yang paling dominan untuk kemudian dijadikan sebagai hasil klasifikasi. Proses pengurutan dan penarikan kesimpulan dapat dilihat pada Gambar 4.24 dan Gambar 4.25.



Gambar 4.24 Proses Pengurutan dan Pengambilan Kesimpulan



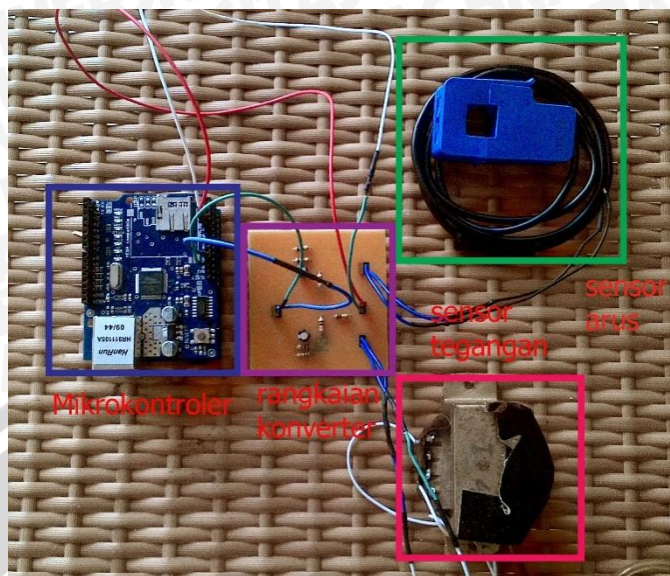
Gambar 4.25 Proses Pengurutan dan Pengambilan Kesimpulan

4.2 Implementasi

Pada subbab ini akan membahas tahapan implementasi berdasarkan perancangan yang telah dijelaskan pada subbab sebelumnya.

4.2.1 Implementasi Perangkat Keras

Implementasi perangkat keras merupakan implementasi rangkaian elektronika untuk pengambilan data arus dan tegangan untuk menghasilkan daya nyata, daya total, dan juga faktor daya untuk kemudian dikirimkan ke *server* setiap detik. Implementasi perangkat keras ditunjukkan pada Gambar 4.26.

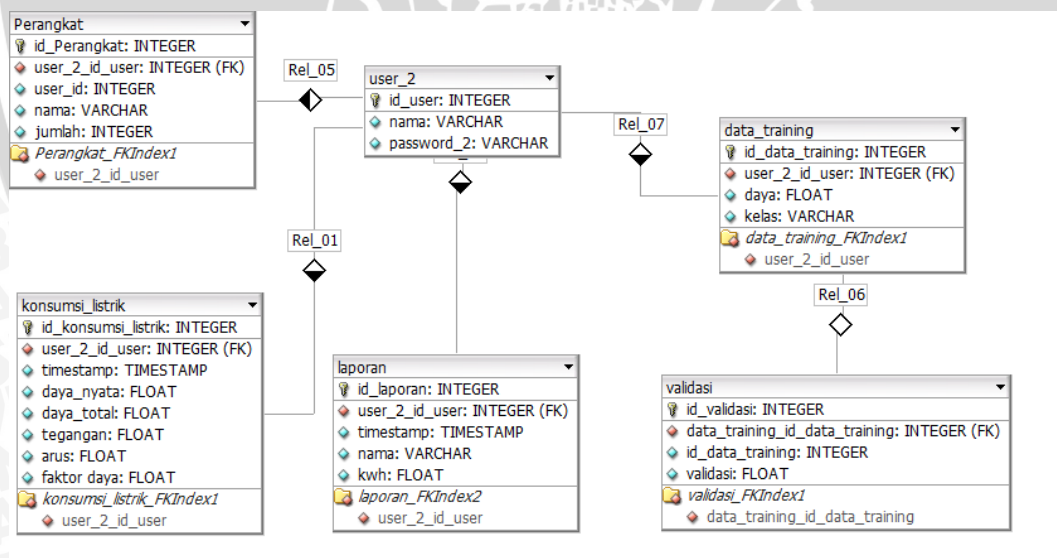


Gambar 4.26 Implementasi Perangkat Keras

4.2.2 Implementasi Perangkat Lunak

4.2.2.1 Implementasi Database

Untuk implementasi *database* pada sistem pemantau konsumsi listrik berbasis web ditunjukkan oleh Gambar 4.27.



Gambar 4.27 Implementasi Database

4.2.2.2 Implementasi Pemrograman Mikrokontroler

Program ini digunakan untuk mengolah data dari sensor untuk menghasilkan daya nyata, daya total, dan faktor daya. Kemudian program akan

mengirimkan data tersebut ke *server* dengan interval 1 detik. Potongan kode program mikrokontroler ditunjukkan oleh Kode 4.1.

```
1  #include "EmonLib.h"
2  #include <Ethernet.h>
3  #include <SPI.h>
4  EnergyMonitor emon1;
5
6  // **** ETHERNET SETTING ****
7  byte mac[] = { 0x54, 0x34, 0x41, 0x30, 0x30, 0x31 };
8
9  EthernetClient client;
10 char server[] = "172.21.3.7";
11 int interval = 1000;
12
13 void setup() {
14     Serial.begin(9600);
15     Ethernet.begin(mac);
16     emon1.voltage(2, 273.01, 1.7);
17     emon1.current(1, 60.6);
18     Serial.println("Kirim Data ke Server");
19     Serial.println("-----");
20     Serial.println("");
21     Serial.print("IP Address      : ");
22     Serial.println(Ethernet.localIP());
23     Serial.print("Subnet Mask      : ");
24     Serial.println(Ethernet.subnetMask());
25     Serial.print("Default Gateway IP: ");
26     Serial.println(Ethernet.gatewayIP());
27     Serial.print("DNS Server IP    : ");
28     Serial.println(Ethernet.dnsServerIP());
29 }
30
31 void loop() {
32     emon1.calcVI(20,2000);
33     emon1.serialprint();
34
35     float realPower = emon1.realPower;
```

```
36 float apparentPower = emon1.apparentPower;
37 float powerFactor   = emon1.powerFactor;
38 float supplyVoltage = emon1.Vrms;
39 float Irms          = emon1.Irms;
40 // if you get a connection, report back via serial:
41 if (client.connect(server, 80)) {
42     Serial.println("-> Connected");
43     // Make a HTTP request:
44     client.print("POST
45 /energon/index.php/home/push_data/");
46     client.print( realPower );
47     client.print("/");
48     client.print( apparentPower );
49     client.print("/");
50     client.print( Irms );
51     client.print("/");
52     client.print( supplyVoltage );
53     client.print("/");
54     client.print( powerFactor );
55     client.println( " HTTP/1.1");
56     client.print( "Host: " );
57     client.println(server);
58     client.println( "Connection: close" );
59     client.println();
60     client.println();
61     client.stop();
62 }
63 else {
64     // you didn't get a connection to the server:
65     Serial.println("--> connection failed/n");
66 }
67 delay(interval);
68 }
```

Kode 4.1 Implementasi Pemrograman Mikrokontroler

Berikut penjelasan Gambar 4.27 di atas:

- 1) Baris 1-3 deklarasi *library*.

- 2) Baris 10 pengaturan alamat ip *server*.
- 3) Baris 16-17 proses pemanggilan fungsi untuk menghitung arus dan tegangan.
- 4) Baris 41-65 proses pengiriman data ke *server*.
- 5) Baris 67 pengaturan *delay* untuk mengirim data setiap interval waktu tertentu.

4.2.2.3 Implementasi Program *Preprocessing* Pendeteksian Perangkat

Program ini digunakan sebagai proses awal untuk mengurangi *noise* pada data dan juga memberikan tanda terhadap kenaikan daya, penurunan daya, dan daya yang tetap (*steady state*). Kode program proses *preprocessing* ditunjukkan pada Kode 4.2.

```

1 function smoothing($data)
2     {
3         foreach($data->result() as $row)
4         {
5             $konsumsi[] = array(
6                 'id' => $row->id_konsumsi,
7                 'daya' => $row->daya_nyata,
8                 'arus' => $row->arus,
9                 'waktu' => $row->waktu
10            );
11        }
12
13        $interval = 7;
14        $count = $data->num_rows();
15        $totalData = $count-$interval;
16        $dataStream = array();
17
18        for($i=0; $i < $totalData; $i++)
19        {
20            $sumPower = 0;
21            $sumCurrent = 0;
22            $j = 0;
23            for($j = $i; $j < $interval+$i; $j++)
24            {
25                $sumPower = $sumPower +
26                $konsumsi[$j]['daya'];
27                $sumCurrent = $sumCurrent +
28                $konsumsi[$j]['arus'];
29            }
30
31            $dataStream[] = array(
32                'daya' => $sumPower/$interval,
33                'arus' =>
34                $sumCurrent/$interval,
35                'waktu' =>
36                $konsumsi[$interval+$i]['waktu']
37            );
38        }
39
40        for ($i=0; $i < sizeof($konsumsi); $i++)

```

```
41     {
42         $data = array(
43             'baca' => '1'
44         );
45         $where = array(
46             'id_konsumsi' => $konsumsi[$i]['id']
47         );
48         $this->save($data,$where);
49     }
50
51     return $dataStream;
52 }
53
54 function event_detection($smoothing)
55 {
56     $treshold = 2.5 ;
57     $distance = 0;
58     $newSegment = 0;
59     $y = $smoothing[0]['daya'];
60     for($i = 0; $i < sizeof($smoothing); $i++)
61     {
62         $distance = $smoothing[$i]['daya'] - $y;
63         if( $distance >= $treshold ) // jika selisih daya lebih
64         besar sama dengan treshold
65         {
66             $newSegment = 1;
67             $smoothing[$i]['state'] = $newSegment;
68         }
69         else if( $distance <= (-1*$treshold) )
70         {
71             $newSegment = -1;
72             $smoothing[$i]['state'] = $newSegment;
73         }
74         else
75         {
76             $newSegment = 0;
77             $smoothing[$i]['state'] = $newSegment;
78         }
79         $y = $smoothing[$i]['daya'];
80     }
81     return $smoothing;
82 }
83
84 function steady_state_check($eventDetection)
85 {
86     $event = array();
87     $waktu = array();
88     $sumDaya = 0;
89     $sumArus = 0;
90     $count = 0;
91
92     for($i=0; $i<sizeof($eventDetection); $i++)
93     {
94         if( $eventDetection[$i]['state']==0 )
95         {
96             $sumDaya = $sumDaya +
97             $eventDetection[$i]['daya'];
```



```

98         $sumArus      =      $sumArus      +
99     $eventDetection[$i]['arus'];
100        $waktu[] = $eventDetection[$i]['waktu'];
101        $count++;
102
103        if( $i==sizeof($eventDetection)-1 )
104        {
105            $event[] = array(
106                'daya' => $sumDaya/$count,
107                'arus' => $sumArus/$count,
108                'mulai' => $waktu[0],
109                'akhir' =>
110                $waktu[sizeof($waktu)-1]
111            );
112            $waktu = array();// clear array
113            $count = 0;//reset count
114            $sumDaya = 0;
115            $sumArus = 0;
116        }
117    }
118    else
119    {
120        if( $count > 5 )
121        {
122            $event[] = array(
123                'daya' => $sumDaya/$count,
124                'arus' => $sumArus/$count,
125                'mulai' => $waktu[0],
126                'akhir' =>
127                $waktu[sizeof($waktu)-1]
128            );
129        }
130        $waktu = array();// clear array
131        $count = 0;// reset count
132        $sumDaya = 0;
133        $sumArus = 0;
134    }
135 }
136 return $event;
}

```

Kode 4.2 Implementasi Program *Preprocessing* Pendeteksian Perangkat

Berikut merupakan penjelasan kode program *preprocessing* pendeteksian perangkat:

- 1) Baris 1-52 merupakan proses *smoothing* menggunakan metode *moving average*.
- 2) Baris 54-82 merupakan proses untuk mendeteksi perubahan daya yang mengindikasikan kegiatan menyalakan atau mematikan perangkat elektronik.
- 3) Baris 84-136 merupakan proses untuk mendeteksi *steady state*.

4.2.2.4 Implementasi Program Validasi *Data Training*

Program ini berguna untuk menghitung nilai validasi dari setiap *data training*. Kode program proses validasi *data training* ditunjukkan oleh Kode 4.3.

```
1 function euclidian()
2 {
3     //ambil data training dari database
4     $dataTraining = $this->get(null, null, 0);
5     foreach ($dataTraining->result() as $row)
6     {
7         $label[] = $row->label;
8         $daya[] = $row->daya;
9         $arus[] = $row->arus;
10    }
11
12    //menghitung euclidian distance
13    $euclidian = array();
14    for ($i=0; $i < sizeof($label); $i++)
15    {
16        for ($j=0; $j < sizeof($label); $j++) {
17            $euclidian[$i][$j] = sqrt(pow($daya[$i]-
18 $daya[$j], 2) + pow($arus[$i]-$arus[$j], 2));
19        }
20    }
21    return $euclidian;
22 }
23
24 function ranking($euclidian)
25 {
26     $rank = array();
27     for ($i=0; $i < sizeof($euclidian); $i++)
28     {
29         $temp = array();
30         for ($j=0; $j < sizeof($euclidian); $j++)
31         {
32             $temp[] = $euclidian[$j][$i];
33         }
34         sort($temp); //urutkan dari jarak yang paling
```

```
35  dekat
36
37      for ($k=0; $k < sizeof($euclidian); $k++)
38      {
39          $rank[$i][$k] = $temp[$k]; //simpan dalam
40  matriks ranking
41      }
42  }
43  return $rank;
44  }
45
46  function validate_training($ranking, $euclidian)
47  {
48      //ambil 4 tetangga terdekat
49      $k = 3;
50      $selection = array();
51      for ($i=0; $i < $k+1; $i++)
52      {
53          for ($j=0; $j < sizeof($ranking); $j++)
54          {
55              $selection[$j][$i] = $ranking[$j][$i];
56          }
57      }
58
59      //ambil data posisi sebelumnya dari 4 tetangga
60  terdekat
61      $matching = array();
62      for ($i=0; $i < $k+1; $i++) {
63          for ($j=0; $j < sizeof($euclidian); $j++) {
64              for ($l=0; $l < sizeof($euclidian); $l++) {
65                  if ($selection[$l][$i] ==
66  $euclidian[$l][$j])
67                  {
68                      $matching[$l][$i] = $j;
69                  }
70              }
71          }
72      }
```

```
73
74     $label = $this->get(null, null, 0);
75     foreach ($label->result() as $row)
76     {
77         $kelas[] = $row->kelas;
78         $id[] = $row->id_data_training;
79     }
80
81     $validation = array();
82     for ($i=0; $i < sizeof($kelas); $i++)
83     {
84         $sum = 0;
85         for ($j=0; $j < $k+1; $j++) {
86             $position = $matching[$i][$j];
87             $x = $kelas[$position];
88             if ($x == $kelas[$i])
89             {
90                 $sum++;
91             }
92         }
93         $validation[] = array(
94             'validasi' => ($sum-1)/$k,
95             'kelas' => $kelas[$i],
96             'id_data_training' => $id[$i]
97         );
98     }
99     return $validation;
100 }
```

Kode 4.3 Implementasi Program Validasi *Data Training*

Berikut penjelasan dari kode program proses validasi *data training*.

- 1) Baris 1-22 merupakan proses perhitungan *Euclidian distance* antar data pada *data training*.
- 2) Baris 24-44 merupakan proses untuk mengurutkan/meranking hasil *Euclidian distance*.
- 3) Baris 46-100 merupakan proses untuk menghitung nilai validasi.

4.2.2.5 Implementasi Program Klasifikasi MKNN

Program ini digunakan untuk mendeteksi aktifitas perangkat elektronik di dalam rumah melalui proses klasifikasi daya dan arus. Kode program proses klasifikasi MKNN ditunjukkan pada Kode 4.4.

```
1 function euclidian($event, $data_training)
2     {
3         $size = $data_training->num_rows();
4         $euclidian = array();
5
6         foreach ($data_training->result() as $database)
7         {
8             $euclidian[] = array(
9                 'distance' =>
10 sqrt(pow($event['daya']-$database->daya,2) +
11 pow($event['arus']-$database->arus,2)),
12                 'label' => $database->
13 >kelas,
14                 'id' => $database->
15 >id_data_training
16             );
17         }
18         return $euclidian;
19     }
20
21     function weight_voting($euclidian)
22     {
23         $weight = array();
24         foreach ($euclidian as $row)
25         {
26             $where = array('id_data_training' =>
27 $row['id']);
28             $validasi = $this->get($where, null, 0);
29             $result = $validasi->row();
30             $weight[] = array(
31                 'bobot' => $result->
32 >validasi*( 1/ $row['distance'] + 0.5),
33                 'label' => $row['label']
```

```
34         );
35     }
36     return $weight;
37 }
38
39 function rank($weight, $k)
40 {
41     uasort($weight, array($this, 'weight_compare'));
42     $closestNeighbour = array();
43
44     foreach ($weight as $value)
45     {
46         //ambil 3 tetangga terdekat
47         $closestNeighbour[] = array(
48             'label' =>
49 $value['label'],
50             'bobot' =>
51 $value['bobot']
52         );
53         if( sizeof( $closestNeighbour)==$k )break;
54     }
55     return $closestNeighbour;
56 }
57
58 function weight_compare($a, $b)
59 {
60     if ($a['bobot'] == $b['bobot'])
61     {
62         return 0;
63     }
64     if ($a['bobot'] > $b['bobot'])
65     {
66         return -1;
67     }
68     return 1;
69 }
70
71 function classify($top)
```

```
72     {
73         $label = array();
74         foreach ($stop as $row)
75         {
76             $label[] = $row['label'];
77         }
78         $count = array_count_values($label);
79         return $count;
80     }
81
82     function class_to_array($class)
83     {
84         foreach ($class as $key => $value)
85         {
86             $x[] = array(
87                 'nama' => $key,
88                 'jumlah' => $value
89             );
89         }
90         return $x;
91     }
92     function get_name($x)
93     {
94         for ($i=0; $i < sizeof($x); $i++)
95         {
96             if( $x[$i]['jumlah']>1 )
97             {
98                 $name = $x[$i]['nama'];break;
99             }
100            else if( $x[$i]['jumlah']<=1 )
101            {
102                $name = 'ragu';
103            }
104        }
105        return $name;
106    }
```

Kode 4.4 Implementasi Program Klasifikasi MKNN

Berikut penjelasan kode program klasifikasi MKNN:

- 1) Baris 1-17 merupakan proses perhitungan *Euclidian distance*.
- 2) Baris 20-36 merupakan proses perhitungan bobot klasifikasi.
- 3) Baris 38-55 merupakan proses pengambilan *k*-tetangga terdekat.
- 4) Baris 57-68 merupakan proses meranking *multidimensional associative array*.
- 5) Baris 70-80 merupakan proses untuk menghitung jumlah kelas/alat hasil perhitungan MKNN.
- 6) Baris 82-91 merupakan proses penyimpanan nama alat dan juga jumlah hasil perhitungan MKNN.
- 7) Baris 93-107 merupakan proses penarikan kesimpulan.

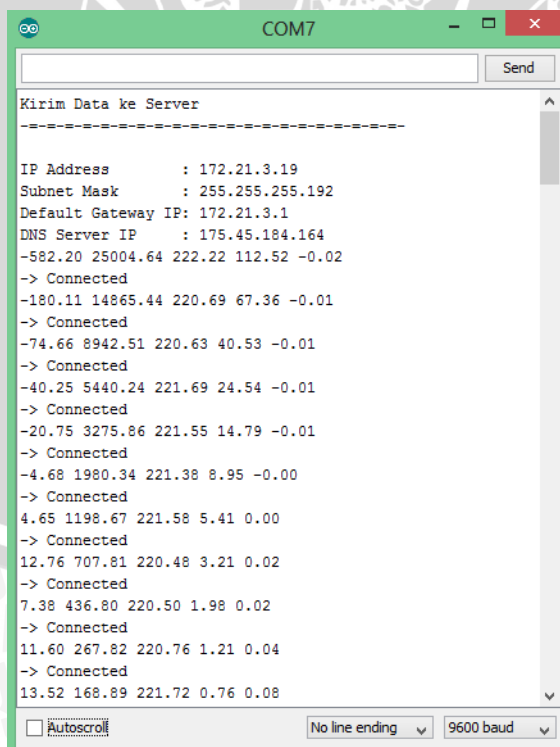


BAB V PENGUJIAN

Bab ini membahas pengujian pada Rancang Bangun Sistem Pemantau Konsumsi Listrik berbasis Web. Pengujian meliputi pengujian komunikasi data, pengujian sensor, pengujian algoritma MKNN, validasi kebutuhan fungsional, dan juga pengujian keseluruhan sistem.

5.1 Pengujian Komunikasi Data

Pengujian komunikasi data ini dilakukan dengan menghubungkan Arduino dengan internet untuk kemudian melakukan pengiriman data ke *server*. Pengujian ini dilakukan untuk mengetahui apakah Arduino dapat terhubung ke *server* dan mengirimkan data serta mengetahui kesesuaian data yang dikirim dan disimpan dalam *database*. Sesuai dengan rancangan pemrograman mikrokontroler, maka Arduino akan mengirimkan data yang berupa daya nyata, daya total, arus, tegangan dan faktor daya dengan tipe data float ke *server* dengan selang waktu 1 detik. Berikut merupakan gambar jika Arduino berhasil mengirim data ke *server*. Gambar 5.1 memperlihatkan Arduino berhasil mengirim data ke *server*.



```
COM7
Send
Kirim Data ke Server
-----
IP Address       : 172.21.3.19
Subnet Mask      : 255.255.255.192
Default Gateway IP: 172.21.3.1
DNS Server IP    : 175.45.184.164
-582.20 25004.64 222.22 112.52 -0.02
-> Connected
-180.11 14865.44 220.69 67.36 -0.01
-> Connected
-74.66 8942.51 220.63 40.53 -0.01
-> Connected
-40.25 5440.24 221.69 24.54 -0.01
-> Connected
-20.75 3275.86 221.55 14.79 -0.01
-> Connected
-4.68 1980.34 221.38 8.95 -0.00
-> Connected
4.65 1198.67 221.58 5.41 0.00
-> Connected
12.76 707.81 220.48 3.21 0.02
-> Connected
7.38 436.80 220.50 1.98 0.02
-> Connected
11.60 267.82 220.76 1.21 0.04
-> Connected
13.52 168.89 221.72 0.76 0.08
Autoscroll No line ending 9600 baud
```

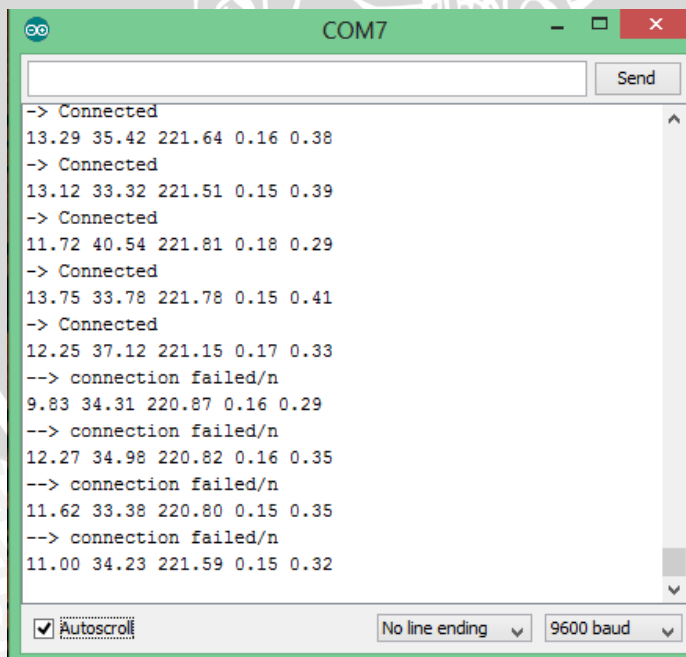
Gambar 5.1 Arduino Berhasil Mengirim Data ke *Server*

Hasil penyimpanan di *database server* dapat dilihat pada Gambar 5.2.

| id_konsumsi | user_2_id_user | daya_nyata | daya_total | tegangan | faktor_daya | arus | tanggal | waktu | jam | baca |
|-------------|----------------|------------|------------|----------|-------------|--------|------------|----------|-----|------|
| 5266 | 1 | -4802.6 | 25441 | 226.09 | -0.19 | 112.53 | 2015-05-19 | 06:56:46 | 6 | 0 |
| 5267 | 1 | -693.84 | 15025.2 | 222.22 | -0.05 | 67.61 | 2015-05-19 | 06:56:47 | 6 | 0 |
| 5268 | 1 | 148.55 | 8984.45 | 221.56 | 0.02 | 40.55 | 2015-05-19 | 06:56:49 | 6 | 0 |
| 5269 | 1 | -582.2 | 25004.6 | 222.22 | -0.02 | 112.52 | 2015-05-19 | 06:56:55 | 6 | 0 |
| 5270 | 1 | -180.11 | 14865.4 | 220.69 | -0.01 | 67.36 | 2015-05-19 | 06:56:57 | 6 | 0 |
| 5271 | 1 | -74.66 | 8942.51 | 220.63 | -0.01 | 40.53 | 2015-05-19 | 06:56:58 | 6 | 0 |
| 5272 | 1 | -40.25 | 5440.24 | 221.69 | -0.01 | 24.54 | 2015-05-19 | 06:57:00 | 6 | 0 |
| 5273 | 1 | -20.75 | 3275.86 | 221.55 | -0.01 | 14.79 | 2015-05-19 | 06:57:01 | 6 | 0 |
| 5274 | 1 | -4.68 | 1980.34 | 221.38 | -0 | 8.95 | 2015-05-19 | 06:57:03 | 6 | 0 |
| 5275 | 1 | 4.65 | 1198.67 | 221.58 | 0 | 5.41 | 2015-05-19 | 06:57:04 | 6 | 0 |
| 5276 | 1 | 12.76 | 707.81 | 220.48 | 0.02 | 3.21 | 2015-05-19 | 06:57:05 | 6 | 0 |
| 5277 | 1 | 7.38 | 436.8 | 220.5 | 0.02 | 1.98 | 2015-05-19 | 06:57:07 | 6 | 0 |
| 5278 | 1 | 11.6 | 267.82 | 220.76 | 0.04 | 1.21 | 2015-05-19 | 06:57:08 | 6 | 0 |
| 5279 | 1 | 13.52 | 168.89 | 221.72 | 0.08 | 0.76 | 2015-05-19 | 06:57:10 | 6 | 0 |
| 5280 | 1 | 11.52 | 106.14 | 221.27 | 0.11 | 0.48 | 2015-05-19 | 06:57:11 | 6 | 0 |
| 5281 | 1 | 11.53 | 67.11 | 221.42 | 0.17 | 0.3 | 2015-05-19 | 06:57:12 | 6 | 0 |
| 5282 | 1 | 15.85 | 65.31 | 220.77 | 0.24 | 0.3 | 2015-05-19 | 06:57:14 | 6 | 0 |
| 5283 | 1 | 12.2 | 48.24 | 221.4 | 0.25 | 0.22 | 2015-05-19 | 06:57:15 | 6 | 0 |
| 5284 | 1 | 11.33 | 39.19 | 221.42 | 0.29 | 0.18 | 2015-05-19 | 06:57:17 | 6 | 0 |
| 5285 | 1 | 13.07 | 35.62 | 221.72 | 0.37 | 0.16 | 2015-05-19 | 06:57:18 | 6 | 0 |

Gambar 5.2 Penyimpanan di *Database Server*

Sedangkan saat pengiriman data gagal maka tampilan dapat dilihat pada Gambar 5.3.



Gambar 5.3 Pengiriman Gagal

Berdasarkan pengujian ini didapatkan hasil bahwa setiap data yang dikirimkan ke *server* tidak secara konstan disimpan dalam interval 1 detik. Interval paling lama dalam penyimpanan data ke *database* adalah 2 detik. Selain itu nilai yang disimpan dalam *database* sama dengan nilai yang dikirim oleh mikrokontroler.

5.2 Pengujian Sensor

Pengujian sensor ini bertujuan untuk menguji keakuratan sensor arus dan juga tegangan yang digunakan pada sistem pemantau konsumsi listrik. Pengujian ini dilakukan dengan membandingkan nilai ukur dari sensor arus dan tegangan dengan multimeter. Adapun multimeter yang digunakan adalah SANWA Digital Clamp Meter 60L dan SANWA Digital Multimeter DC-800. Adapun alat-alat elektronik yang digunakan dalam pengujian ini ditunjukkan oleh Tabel 5.1

Tabel 5.1 Daftar Alat untuk Diuji

| No | Nama | Spesifikasi |
|----|---------|-------------------------------------|
| 1 | Monitor | Acer |
| 2 | Laptop | Asus K45DR |
| 3 | Setrika | Phillips HI 114 |
| 4 | Lampu | Panasonic Light Capsule Eco 8 Watts |

Hasil pengujian sensor arus dapat dilihat pada Tabel 5.2.

Tabel 5.2 Pengujian Sensor Arus

| No | Perangkat | Sensor Arus (Ampere) | Multimeter (Ampere) | Clamp meter (Ampere) |
|----|-----------|-------------------------|------------------------|-------------------------|
| 1 | Lampu | 0.035674244 | 0.03 | 0 |
| 2 | Monitor | 0.242263648 | 0.03 | 0 |
| 3 | Laptop | 0.06 | 0.04 | 0 |
| 4 | Setrika | 2.05 | 1.7 | 1.8 |

Berikut adalah hasil pengujian untuk sensor tegangan. Data pengukuran dapat dilihat pada Tabel 5.3.

Tabel 5.3 Pengujian Sensor Tegangan

| No | Tempat | Sensor Tegangan (Volt) | Multimeter (Volt) | Clamp Meter (Volt) |
|----|--------|---------------------------|----------------------|-----------------------|
| 1 | Rumah | 220 | 220 | 224 |

5.2.4 Analisa Pengujian Sensor

1. Dari pengujian ini juga didapatkan hasil bahwa tanpa adanya alat elektronik yang terpasang, sensor tetap memberikan keluaran daya sebesar 0-30 Watt. Pembacaan seperti ini bisa terjadi karena adanya digital *noise* pada Arduino itu sendiri atau *noise* dari PCB atau sistem pengkabelan.
2. Perhitungan arus pada penelitian ini dilakukan dengan menghitung selisih antara rata-rata nilai arus sebelum ada alat elektronik yang terpasang dengan nilai arus setelah alat elektronik terpasang. Kemudian didapatkan hasil bahwa perbedaan nilai arus terbesar antara sensor dengan 2 alat ukur lainnya sebesar 0.35 Ampere sedangkan untuk sensor tegangan perbedaan nilai terbesar ialah 4 Volt.

5.3 Pengujian Algoritma *Modified K-Nearest Neighbor*

Pengujian algoritma M-KNN bertujuan untuk mengetahui hasil dari perhitungan sistem sama atau tidak dengan perhitungan manual. Keluaran dari proses ini akan menunjukkan aktivitas nyala atau mati dari perangkat-perangkat elektronik di dalam suatu bangunan. Pengujian ini dilakukan tanpa menggunakan *database server* melainkan *database* lokal. Adapun urutan aktivitas perangkat pada percobaan ini dapat dilihat pada Tabel 5.4.

Tabel 5.4 Urutan Aktivitas

| No | Perangkat | Status |
|----|-----------|--------|
| 1 | Setrika | Nyala |
| 2 | Setrika | Mati |
| 3 | Monitor | Nyala |
| 4 | Monitor | Mati |
| 5 | Laptop | Nyala |

| | | |
|---|--------|-------|
| 6 | Laptop | Mati |
| 7 | Lampu | Nyala |
| 8 | Lampu | Mati |

Berdasarkan perhitungan otomatis yang dilakukan oleh sistem didapatkan hasil yang dapat dilihat pada Gambar 5.4.

| id_laporan | tanggal | jam | nyala | mati | daya | arus | status | perangkat | daya_terakhir | arus_terakhir |
|------------|------------|-----|----------|----------|---------|----------|--------|-----------|---------------|---------------|
| 93 | 2015-04-08 | 4 | 04:57:39 | 04:58:08 | 320.115 | 1.49709 | nyala | setrika | 320.115 | 1.49709 |
| 94 | 2015-04-08 | 4 | 04:58:16 | 04:59:01 | 320.901 | 1.49904 | nyala | setrika | 320.901 | 1.49904 |
| 95 | 2015-04-08 | 4 | 04:59:10 | 04:59:55 | 320.356 | 1.49714 | nyala | setrika | 320.901 | 1.49904 |
| 96 | 2015-04-08 | 5 | 05:00:14 | 05:00:34 | 312.622 | 1.30861 | mati | setrika | 8.27893 | 0.190429 |
| 97 | 2015-04-08 | 5 | 05:00:47 | 05:00:48 | 46.6582 | 0.208143 | nyala | monitor | 54.9371 | 0.398571 |
| 98 | 2015-04-08 | 5 | 05:13:29 | 05:14:13 | 49.1739 | 0.382193 | nyala | monitor | 54.9371 | 0.398571 |
| 99 | 2015-04-08 | 5 | 05:14:35 | 05:15:05 | 45.9338 | 0.21261 | mati | monitor | 9.0033 | 0.185961 |
| 100 | 2015-04-08 | 5 | 05:15:15 | 05:16:00 | 40.8484 | 0.119022 | nyala | monitor | 49.8517 | 0.304983 |
| 101 | 2015-04-08 | 5 | 05:16:09 | 05:16:54 | 52.0944 | 0.322458 | nyala | monitor | 52.0944 | 0.322458 |
| 102 | 2015-04-08 | 5 | 05:17:02 | 05:17:26 | 48.654 | 0.299441 | nyala | monitor | 52.0944 | 0.322458 |
| 103 | 2015-04-08 | 5 | 05:17:34 | 05:17:47 | 40.9996 | 0.118452 | mati | monitor | 7.6544 | 0.180989 |
| 104 | 2015-04-08 | 5 | 05:17:56 | 05:21:21 | 8.2404 | 0.196312 | nyala | lampu | 8.2404 | 0.196312 |
| 105 | 2015-04-08 | 5 | 05:21:29 | 05:22:14 | 7.43528 | 0.189435 | nyala | lampu | 8.2404 | 0.196312 |

Gambar 5.4 Hasil Perhitungan

Dalam melakukan pendeteksian aktifitas perangkat dengan data yang statis digunakan 551 data. Pendeteksian perangkat dilakukan dengan cara mengambil 50 data dari *database* setiap 1 menit. Berdasarkan data diatas dilakukan perhitungan manual untuk mengecek kecocokan antara perhitungan M-KNN yang dilakukan oleh sistem dengan perhitungan manual. Hasil perhitungan manual dapat dilihat pada Tabel 5.5

Tabel 5.5 Perhitungan *Euclidian Distance*

| DAYA | ARUS | euclidian distance | | | | | | | | | |
|---------|----------|--------------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 320.115 | 1.49709 | 277.9991 | 277.5273 | 297.2662 | 9.616394 | 314.4442 | 314.7528 | 283.7418 | 292.1132 | 44.47798 | 297.855 |
| 320.901 | 1.49904 | 278.7851 | 278.3133 | 298.0522 | 8.83042 | 315.2302 | 315.5388 | 284.5278 | 292.8992 | 43.69198 | 298.641 |
| 320.356 | 1.49714 | 278.2401 | 277.7683 | 297.5071 | 9.375407 | 314.6852 | 314.9938 | 283.9828 | 292.3542 | 44.23698 | 298.096 |
| 312.622 | 1.30861 | 270.5055 | 270.0335 | 289.7725 | 17.11132 | 306.9506 | 307.2591 | 276.2481 | 284.6196 | 51.97218 | 290.3614 |
| 46.6582 | 0.208143 | 4.544728 | 4.067473 | 23.80679 | 283.0761 | 40.98475 | 41.29341 | 10.28278 | 18.6552 | 317.9378 | 24.39593 |
| 49.1739 | 0.382193 | 7.055944 | 6.585522 | 26.32277 | 280.5596 | 43.50085 | 43.80941 | 12.79844 | 21.17002 | 315.4213 | 26.91166 |
| 45.9338 | 0.21261 | 42.07877 | 42.50977 | 32.33128 | 434.8828 | 40.62901 | 40.89486 | 37.40495 | 33.0732 | 484.0616 | 32.35009 |
| 40.8484 | 0.119022 | 1.30283 | 1.744512 | 17.99745 | 288.8862 | 35.17502 | 35.48375 | 4.476024 | 12.84735 | 323.748 | 18.58677 |
| 52.0944 | 0.322458 | 9.97678 | 9.504383 | 29.24303 | 277.6394 | 46.42112 | 46.72973 | 15.71866 | 24.09061 | 312.5011 | 29.83201 |
| 48.654 | 0.299441 | 6.536942 | 6.063988 | 25.80259 | 281.0799 | 42.98067 | 43.28929 | 12.27823 | 20.65031 | 315.9416 | 26.3916 |
| 40.9996 | 0.118452 | 1.156127 | 1.593551 | 18.14865 | 288.7351 | 35.32622 | 35.63495 | 4.627135 | 12.99853 | 323.5968 | 18.73797 |
| 8.2404 | 0.196312 | 33.8783 | 34.35033 | 14.61116 | 321.4935 | 2.566945 | 2.8757 | 28.13554 | 19.76442 | 356.3552 | 14.02231 |
| 7.43528 | 0.189435 | 34.68345 | 35.15545 | 15.4163 | 322.2987 | 1.761828 | 2.070702 | 28.94068 | 20.56957 | 357.1604 | 14.82746 |

Data pada tabel merupakan perhitungan *Euclidian distance* antara data uji dan data latih. Pada tahap selanjutnya, hasil perhitungan *Euclidian distance* diatas digunakan untuk menghitung bobot klasifikasi. Perhitungan bobot klasifikasi ditunjukkan oleh Tabel 5.6.

Tabel 5.6 Perhitungan Bobot Klasifikasi

| DAYA | ARUS | Bobot | | | | | | | | | |
|---------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| | | laptop | monitor | laptop | setrika | lampu | lampu | monitor | laptop | setrika | monitor |
| 320.115 | 1.49709 | 0.167866 | 0.167868 | 0.167788 | 0.201329 | 0.167727 | 0.167726 | 0.167841 | 0.167808 | 0.174161 | 0.167786 |
| 320.901 | 1.49904 | 0.167862 | 0.167864 | 0.167785 | 0.204415 | 0.167724 | 0.167723 | 0.167838 | 0.167805 | 0.174296 | 0.167783 |
| 320.356 | 1.49714 | 0.167865 | 0.167867 | 0.167787 | 0.20222 | 0.167726 | 0.167725 | 0.16784 | 0.167807 | 0.174202 | 0.167785 |
| 312.622 | 1.30861 | 0.167899 | 0.167901 | 0.167817 | 0.186147 | 0.167752 | 0.167751 | 0.167873 | 0.167838 | 0.17308 | 0.167814 |
| 46.6582 | 0.208143 | 0.240011 | 0.248617 | 0.180668 | 0.167844 | 0.1748 | 0.174739 | 0.199083 | 0.184535 | 0.167715 | 0.18033 |
| 49.1739 | 0.382193 | 0.213908 | 0.217283 | 0.17933 | 0.167855 | 0.174329 | 0.174275 | 0.192711 | 0.182412 | 0.167723 | 0.179053 |
| 45.9338 | 0.21261 | 0.174588 | 0.174508 | 0.176976 | 0.167433 | 0.174871 | 0.174817 | 0.175578 | 0.176745 | 0.167355 | 0.17697 |
| 40.8484 | 0.119022 | 0.42252 | 0.357742 | 0.185188 | 0.16782 | 0.176143 | 0.17606 | 0.241137 | 0.192612 | 0.167696 | 0.1846 |
| 52.0944 | 0.322458 | 0.200077 | 0.201738 | 0.178065 | 0.167867 | 0.173847 | 0.1738 | 0.187873 | 0.180503 | 0.167733 | 0.17784 |
| 48.654 | 0.299441 | 0.217659 | 0.221636 | 0.179585 | 0.167852 | 0.174422 | 0.174367 | 0.193815 | 0.182808 | 0.167722 | 0.179297 |
| 40.9996 | 0.118452 | 0.454985 | 0.375843 | 0.185033 | 0.167821 | 0.176102 | 0.176021 | 0.238705 | 0.19231 | 0.167697 | 0.184456 |
| 8.2404 | 0.196312 | 0.176506 | 0.17637 | 0.18948 | 0.167703 | 0.296522 | 0.28258 | 0.178514 | 0.183532 | 0.167602 | 0.190438 |
| 7.43528 | 0.189435 | 0.176277 | 0.176148 | 0.188289 | 0.167701 | 0.355864 | 0.327642 | 0.178184 | 0.182872 | 0.1676 | 0.189147 |

Dari perhitungan bobot klasifikasi, diambil 3 tetangga terdekat yaitu yang memiliki nilai bobot yang paling besar. Dari 3 nilai tersebut kemudian diambil kesimpulan dengan mengambil nilai yang paling sering muncul. Hasil klasifikasi ditunjukkan oleh Tabel 5.7.

Tabel 5.7 Hasil Klasifikasi

| FITUR | | HASIL |
|---------|----------|---------|
| DAYA | ARUS | |
| 320.115 | 1.49709 | setrika |
| 320.901 | 1.49904 | setrika |
| 320.356 | 1.49714 | setrika |
| 312.622 | 1.30861 | setrika |
| 46.6582 | 0.208143 | monitor |
| 49.1739 | 0.382193 | monitor |
| 45.9338 | 0.21261 | monitor |
| 40.8484 | 0.119022 | monitor |
| 52.0944 | 0.322458 | monitor |
| 48.654 | 0.299441 | monitor |
| 40.9996 | 0.118452 | monitor |
| 8.2404 | 0.196312 | lampu |
| 7.43528 | 0.189435 | lampu |

5.3.4 Analisa Pengujian Algoritma *Modified K-Nearest Neighbor*

Dari percobaan ini didapatkan hasil klasifikasi yang sama dari perhitungan sistem maupun perhitungan manual. Namun, baik sistem maupun perhitungan manual tidak mampu mendeteksi aktifitas laptop seperti dalam kondisi nyata karena laptop dan monitor memiliki daya yang hampir serupa.

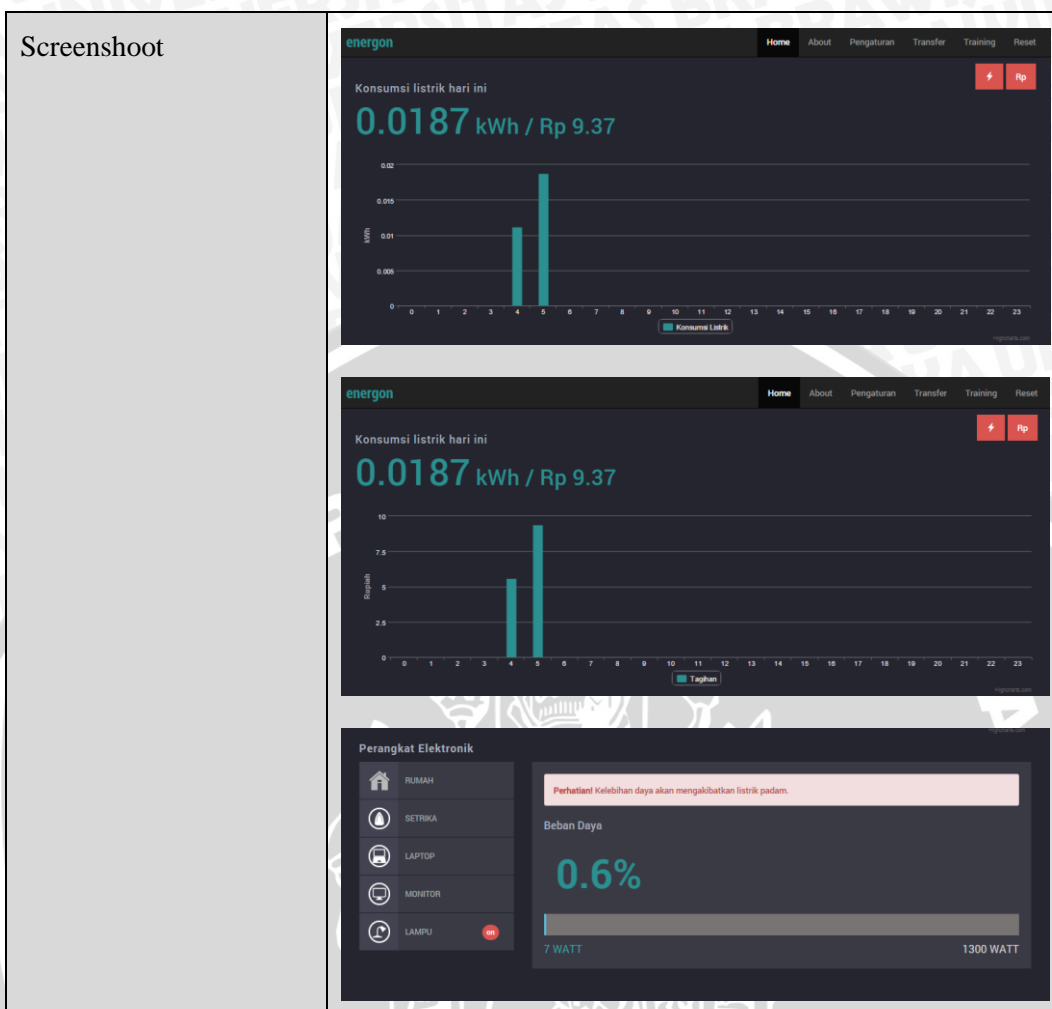
5.4 Pengujian Fungsional

Pengujian fungsional digunakan untuk mengetahui apakah sistem sesuai dengan perancangan pada kebutuhan fungsional. Pengujian validasi menggunakan metode pengujian *Black-Box*. Penggunaan pengujian ini bertujuan untuk menemukan kesesuaian antara kinerja sistem dengan daftar kebutuhan yang telah dijabarkan.

5.4.1 Kasus Uji Memantau Total Konsumsi listrik

Tabel 5.8 Kasus Uji Memantau Total Konsumsi listrik

| | |
|-----------------------|--|
| Kode Uji | T01 |
| Nama Kasus Uji | Kasus Uji Memantau Total Konsumsi listrik |
| Objek Uji | Kebutuhan Fungsional (FR01) |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan fungsional pengguna ingin memantau total konsumsi listrik |
| Prosedur Pengujian | <ol style="list-style-type: none"> 1. Pengguna melakukan login pada website. 2. Pengguna membuka halaman utama |
| Hasil yang Diharapkan | Aplikasi dapat menampilkan total konsumsi listrik per jam dalam satuan kwh dan juga biaya listrik yang harus dikeluarkan dalam satuan Rupiah serta beban daya secara keseluruhan pada saat itu |



5.4.2 Kasus Uji Memantau Aktivitas Perangkat Elektronik

Tabel 5.9 Kasus Uji Memantau Aktivitas Perangkat Elektronik

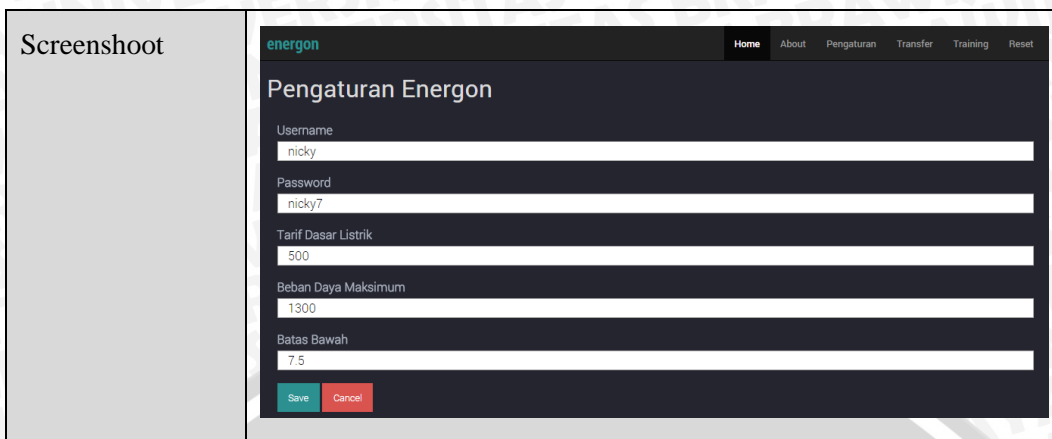
| | |
|--------------------|---|
| Kode Uji | T02 |
| Nama Kasus Uji | Kasus Uji Memantau Aktifitas Perangkat Elektronik |
| Objek Uji | Kebutuhan Fungsional (FR02) |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan fungsional saat pengguna memantau aktifitas perangkat elektronik |
| Prosedur Pengujian | <ol style="list-style-type: none"> 1. Pengguna melakukan login pada website. 2. Pengguna membuka halaman utama |

| | |
|-----------------------|--|
| Hasil yang Diharapkan | Aplikasi dapat menampilkan status nyala dan mati dari perangkat elektronik |
| Screenshoot |  |

5.4.3 Kasus Uji Mengubah Tarif Dasar Listrik dan Beban Daya Maksimal

Tabel 5.10 Kasus Uji untuk mengubah tarif dasar listrik dan beban daya maksimal

| | |
|-----------------------|---|
| Kode Uji | T06 |
| Nama Kasus Uji | Kasus uji mengubah tarif dasar listrik dan beban daya maksimal |
| Objek Uji | Kebutuhan Fungsional (FR06) |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan fungsional dalam mengubah tarif dasar listrik dan beban daya maksimal |
| Prosedur Pengujian | <ol style="list-style-type: none"> 1. Pengguna melakukan login pada website. 2. Pengguna membuka halaman utama |
| Hasil yang Diharapkan | Sistem dapat menampilkan formulir untuk mengganti nilai tarif dasar listrik dan beban daya maksimum. Kemudian dapat melakukan penyimpanan data ke <i>database</i> |



5.4.4 Hasil Pengujian Fungsional

Tabel 5.11 Hasil Pengujian Kebutuhan Fungsional

| No. | Kode Uji | Nama Kasus Uji | Hasil yang Diharapkan | Hasil yang Didapatkan | Status Validasi |
|-----|----------|---|--|--|-----------------|
| 1 | T01 | Kasus Uji Memantau Total Konsumsi Listrik | Aplikasi dapat menampilkan total konsumsi listrik per jam dalam satuan kwh dan juga biaya listrik yang harus dikeluarkan dalam satuan Rupiah serta beban daya secara keseluruhan pada saat itu | Aplikasi dapat menampilkan total konsumsi listrik per jam dalam satuan kwh dan juga biaya listrik yang harus dikeluarkan dalam satuan Rupiah serta beban daya secara keseluruhan pada saat itu | Valid |
| 2 | T02 | Kasus Uji Memantau Aktifitas Perangkat Elektronik | Aplikasi dapat menampilkan status nyala dan mati dari perangkat elektronik | Aplikasi dapat menampilkan status nyala dan mati dari perangkat elektronik | Valid |
| 3 | T03 | Kasus uji | Sistem dapat | Sistem dapat | Valid |



| | | | |
|--|--|--|--|
| | mengubah tarif dasar listrik dan beban daya maksimal | menampilkan formulir untuk mengganti nilai tarif dasar listrik dan beban daya maksimum. Kemudian dapat melakukan penyimpanan data ke <i>database</i> | menampilkan formulir untuk mengganti nilai tarif dasar listrik dan beban daya maksimum. Kemudian dapat melakukan penyimpanan data ke <i>database</i> |
|--|--|--|--|

5.5 Pengujian Keseluruhan Sistem

Pada pengujian ini sistem akan diuji untuk mengenali aktifitas perangkat elektronik dengan mengakses *database server* yang diperbarui secara langsung dan berkala oleh sensor. Pengujian ini bertujuan untuk mengetahui kinerja sistem dalam mendeteksi aktivitas perangkat elektronik di keadaan nyata. Daftar perangkat elektronik yang menjadi bahan uji ditunjukkan oleh tabel 5.12

Tabel 5.12 Daftar Peralatan Elektronik sebagai Bahan Uji

| No | Kelas | Merk dan Spesifikasi |
|----|---------|-------------------------------------|
| 1 | Lampu | Panasonic Light Capsule Eco 8 Watts |
| 2 | Laptop | HP Pavilion 431 |
| 3 | Monitor | Acer V193HQV |
| 4 | Setrika | Phillips HI 114 |

5.5.1 Skenario Pengujian 1

Pengujian untuk skenario pertama ditunjukkan oleh tabel 5.13.

Tabel 5.13 Pengujian Sistem Skenario 1

| No | Keadaan Nyata | Sistem | Nilai |
|----|---------------|-------------|-------|
| 1 | Monitor on | Monitor on | 1 |
| 2 | Monitor off | Monitor off | 1 |
| 3 | Lampu on | Lampu on | 1 |

| | | | |
|------------------|-------------|-------------|------|
| 4 | Lampu off | Lampu off | 1 |
| 5 | Monitor on | Laptop on | 0 |
| 6 | Monitor off | Monitor on | 0 |
| 7 | Lampu on | Monitor off | 0 |
| 8 | Lampu off | Lampu on | 0 |
| <i>Precision</i> | | | 50 % |
| <i>Recall</i> | | | 50 % |

5.5.2 Analisa Skenario Pengujian 1

Pada skenario 1 terdapat 8 aktifitas perangkat yang harus dideteksi oleh sistem. Dari pengujian didapatkan hasil bahwa sistem mendeteksi 8 aktifitas perangkat elektronik dengan 4 hasil deteksi yang sesuai dengan skenario 1. Jika kita lihat lebih dalam lagi maka sebenarnya sistem mampu mendeteksi semua aktifitas perangkat elektronik namun kesalahan terjadi dengan adanya pendeteksian laptop yang menyala dan kegagalan dalam mendeteksi lampu yang mati. Sistem mendeteksi laptop yang menyala karena daya listrik dan arus dari laptop dan monitor hampir sama. Kemudian, sistem tidak mampu mendeteksi aktifitas lampu yang mati dikarenakan panjang data belum cukup untuk dideteksi sebagai aktifitas lampu yang mati.

5.5.3 Skenario Pengujian 2

Pengujian skenario kedua ditunjukkan oleh tabel 5.14.

Tabel 5.14 Pengujian Sistem Skenario 2

| No | Keadaan Nyata | Sistem | Nilai |
|----|---------------|-------------|-------|
| 1 | Lampu on | Lampu on | 1 |
| 2 | Lampu off | Lampu off | 1 |
| 3 | Setrika on | Setrika on | 1 |
| 4 | Setrika off | Setrika off | 1 |
| 5 | Monitor on | Lampu on | 0 |
| 6 | Monitor off | Lampu off | 0 |
| 7 | Laptop on | Monitor on | 0 |
| 8 | Laptop off | Monitor off | 0 |

| | |
|------------------|------|
| <i>Precision</i> | 50 % |
| <i>Recall</i> | 50 % |

5.5.4 Analisa Skenario Pengujian 2

Pada skenario 2 terdapat 8 aktifitas perangkat elektronik yang harus dideteksi oleh sistem. Dari pengujian didapatkan hasil bahwa sistem mendeteksi 8 aktifitas perangkat elektronik dengan 4 hasil deteksi sesuai dengan skenario 2. Kesalahan terjadi saat sistem mendeteksi lampu yang menyala dan lampu mati. Hal ini terjadi karena daya yang dideteksi sebagai lampu masih dalam proses untuk mencapai daya maksimum. Sementara itu sistem mendeteksi aktifitas monitor karena daya monitor dan laptop hampir mirip.

5.5.5 Skenario Pengujian 3

Pengujian untuk skenario ketiga ditunjukkan oleh tabel 5.15

Tabel 5.15 Pengujian Sistem Skenario 3

| No | Keadaan Nyata | Sistem | Akurasi |
|------------------|---------------|-------------|---------|
| 1 | Monitor on | Monitor on | 1 |
| 2 | Laptop on | Lampu on | 0 |
| 3 | Lampu on | Lampu on | 1 |
| 4 | Setrika on | Lampu on | 0 |
| 5 | Setrika off | Lampu off | 0 |
| 6 | Setrika on | Setrika on | 1 |
| 7 | Setrika off | Lampu on | 0 |
| 8 | Lampu off | Lampu off | 0 |
| 9 | Laptop off | Setrika off | 0 |
| 10 | Monitor off | Setrika on | 0 |
| 11 | - | Setrika off | 0 |
| 12 | - | Lampu off | 0 |
| 13 | - | Lampu off | 0 |
| 14 | - | Monitor off | 0 |
| <i>Precision</i> | | | 21.42 % |
| <i>Recall</i> | | | 30 % |

5.5.6 Analisa Skenario Pengujian 3

Pada skenario ketiga terdapat 10 aktifitas perangkat elektronik yang harus dideteksi oleh sistem. Dari pengujian didapatkan hasil bahwa sistem mendeteksi 14 aktifitas perangkat dan 3 diantaranya sesuai dengan skenario 3. Pada uji coba ini terjadi banyak kesalahan deteksi aktifitas nyala dan mati lampu dikarenakan adanya fluktuasi nilai yang melebihi ambang batas. Yang dimaksud ialah ambang batas yang menandakan adanya aktifitas nyala/mati dari suatu perangkat elektronik. Pada sistem ini ambang batas ditentukan dengan nilai 2.5 Watt. Sehingga selisih daya yang sama dengan atau lebih dari 2.5 Watt dideteksi sebagai aktifitas menyalakan alat elektronik. Jika dilihat lebih seksama sebenarnya sistem mampu mendeteksi semua aktifitas perangkat elektronik yang ada pada skenario 3.



BAB VI PENUTUP

6.1 Kesimpulan

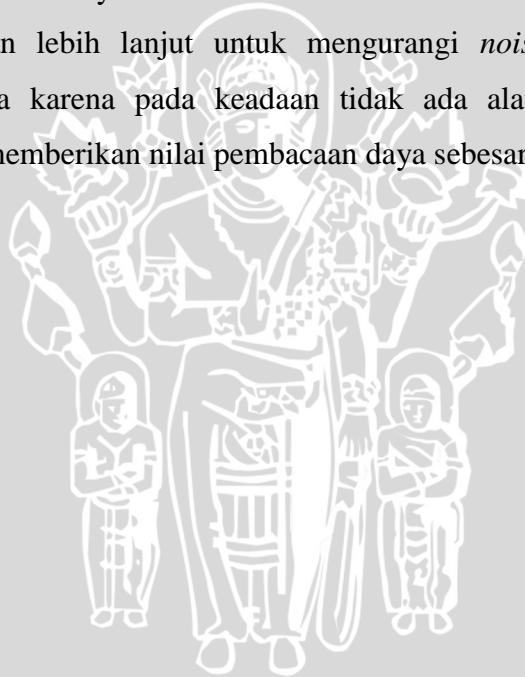
Dari tahapan-tahapan penulisan yang telah dilakukan, dapat diambil kesimpulan sebagai berikut:

1. Untuk membuat sistem monitoring konsumsi listrik diperlukan alat yang mampu memantau data kelistrikan berupa daya, arus, faktor daya, dan tegangan listrik dari suatu bangunan. Alat ini dibuat dengan mikrokontroler Arduino Uno, sensor arus, sensor tegangan, dan juga *library* pemrograman Open Energy Monitor. Data tersebut nantinya disimpan dalam *database server* dan ditampilkan oleh sebuah aplikasi berbasis web untuk disajikan kepada pengguna.
2. Untuk menggunakan metode *Modified K-Nearest Neighbor* pada sistem pemantau konsumsi listrik berbasis web yang menerapkan konsep *Non-Intrusive Load Monitoring*, hal pertama yang perlu dilakukan adalah mendapatkan data daya listrik, arus listrik, dan tegangan listrik dari suatu bangunan. Kemudian dilakukan pemilihan fitur untuk diolah menggunakan metode klasifikasi *Modified K-Nearest Neighbor*. Setelah itu, data disimpan dalam *database* untuk kemudian dilakukan proses *smoothing* terlebih dahulu dengan metode *moving average*. Kemudian data tersebut akan diproses dengan metode klasifikasi *Modified K-Nearest Neighbor*. Setelah proses klasifikasi selesai, maka hasilnya akan ditampilkan pada halaman web sebagai aktifitas nyala/mati dari perangkat elektronik pada suatu bangunan.
3. Sistem belum menunjukkan performa yang bagus dalam implementasi di keadaan nyata dibuktikan dengan pengujian yang melibatkan 3 skenario. Skenario 1 menghasilkan nilai *recall* dan *precision* sebesar 50%, skenario 2 menghasilkan nilai *recall* dan *precision* sebesar 50%, dan skenario 3 menghasilkan nilai *recall* 21.42% dan *precision* 30%.

6.2 Saran

Saran yang dapat diberikan untuk pengembangan sistem monitoring konsumsi listrik berbasis web adalah :

1. Perlu dilakukan penelitian lebih lanjut untuk meningkatkan akurasi klasifikasi dengan mengembangkan metode MKNN misalnya dengan menambahkan konsep *fuzzy* agar dapat mengenali peralatan elektronik yang memiliki nilai daya dan arus yang hampir serupa.
2. Perlu dilakukan penelitian lebih mendalam mengenai karakteristik konsumsi listrik dari berbagai perangkat elektronik untuk menentukan fitur klasifikasi yang mampu membedakan aktifitas perangkat elektronik yang satu dengan lainnya.
3. Perlu penelitian lebih lanjut untuk mengurangi *noise* pada peralatan pengambil data karena pada keadaan tidak ada alat elektronik yang menyala, alat memberikan nilai pembacaan daya sebesar 0-30 Watt.



DAFTAR PUSTAKA

- [RZL-10] Ruzzeli, Antonio G. (2010). *Real-time recognition and profiling of appliances through a single electricity sensor*. Dublin.
- [SHA-14] Saha, Manaswi (2014). *EnergyLens : Combining Smartphones with Electricity Meter for Accurate Activity Detection and User Annotation*. Delhi.
- [HRT-85] Hart, George W. (1985). *PROGRESS REPORT #2 : Prototype Nonintrusive Load Monitor*. Massachusetts, United States.
- [BLY-11] <http://www.bidgely.com>
- [HAP-08] Parvin, Hamid (2008). *MKNN : Modified K-Nearest Neighbor*. San Fransisco.
- [ASD-10] DiCarlo, Antonio Scotto and Cove, Glen. (2010). *Smart Homes (Home Automation)*. New York.

