DISERTASI

Penempatan Optimum Perangkat Ukur pada Jaringan Distribusi Menggunakan Metode *Integer Linear K-Means Clustering*

Riny Sulistyowati
NRP 07111560010004

Pembimbing:
Prof. Ir. Mochamad Ashari, M.Eng., Ph.D
Dedet Candra Riawan, ST, M. Eng. ,Ph.D
Dr. Rony Seto Wibowo, ST., MT

PROGRAM DOKTOR
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2021

LEMBAR PENGESAHAN DISERTASI

Disertasi disusun untuk memenuhi salah satu syarat memperoleh gelar
Doktor (Dr.)
di
Institut Teknologi Sepuluh Nopember
Oleh:

RINY SULISTYOWATI
NRP: 07111560010004

Tanggal Ujian: 25 Januari 2021
Periode Wisuda: Maret 2021

Disetujui oleh:
Pembimbing:

1. Prof. Ir. Mochamad Ashari, M.Eng., Ph.D. .....
   NIP: 196510121990031003

2. Dedet Candra Riawan, ST., M.Eng. , Ph.D .....
   NIP: 197311192000031001

3. Dr. Rony Seto Wibowo, ST., MT. .....
   NIP: 197411292000121001

Penguji:

1. Djuwari, ST., Ph.D. .....
   NIP: 196404051990021001

2. Heri Suryoatmojo, ST., MT., Ph.D .....
   NIP: 198006032006041003

3. Dr.Eng. Ardyono Priyadi, ST., M.Eng .....
   NIP: 197309271998031004

Kepala Departemen Teknik Elektro
Fakultas Teknologi Elektro dan Informatika Cerdas

Dedet Candra Riawan, S.T, M.Eng, Ph.D
NIP: 197311192000031001

<Halaman ini sengaja dikosongkan>

PERNYATAAN KEASLIAN DISERTASI

Dengan ini Saya menyatakan bahwa isi sebagian maupun keseluruhan disertasi saya dengan judul :

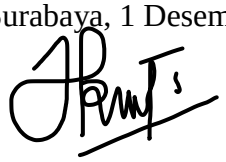
Penempatan Optimum Perangkat Ukur pada Jaringan Distribusi Menggunakan Metode *Integer Linear K-Means Clustering*

adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 1 Desember 2020



Riny Sulistyowati
NRP. 07111560010004

<Halaman ini sengaja dikosongkan>

PENEMPATAN OPTIMUM PERANGKAT UKUR PADA JARINGAN DISTRIBUSI MENGGUNAKAN METODE *INTEGER LINEAR K-MEANS CLUSTERING*

Nama	: Riny Sulistyowati
NRP	: 07111560010004
Pembimbing I	: Prof. Ir. Mochamad Ashari, M.Eng., Ph.D
Pembimbing II	: Dedet Candra Riawan, S.T, M.Eng, Ph.D
Pembimbing III	: Dr. Rony Seto Wibowo, S.T., M.T

ABSTRAK

Jaringan listrik cerdas atau dikenal dengan s*mart-grid* merupakan salah satu bentuk transformasi dan reformasi teknologi di industri ketenagalistrikan. Untuk mendapatkan perfomansi *smart-grid* yang diharapkan, maka dibutuhkan hasil analisis yang mampu mengimbangi kondisi dinamis sistem. Data tersebut di implementasikan dalam bentuk instalasi perangkat pengukuran atau di sebut pula *Phasor Measurement Unit (PMU).* Permasalahan menjadi kompleks karena di butuhkan banyak perangkat *PMU* yang harus dipasang di jaringan listrik. Sehingga dibutuhkan optimasi untuk mereduksi jumlah dan penempatan *PMU* yang mampu mengontrol semua jaringan listrik.

Penelitian ini membahas tentang metode optimasi untuk mereduksi jumlah dan penempatan *PMU* pada jaringan distribusi menggunakan metode *Integer Linear K-Means.* Metode ini merupakan penggabungan antara metode *Integer Linear* dan metode *K-Means*. Keunggulan dari metode tersebut mampu menganalisa data non linear dengan proses iterasi yang lebih cepat dan sangat akurat. Metode *Integer Linear* digunakan untuk membangkitkan data pada bus yang terpasang *PMU* dan *K-Means* untuk proses *Cluster*.

Tahap pertama membuat model simulasi menggunakan banyak *PMU* yang di optimalkan pada jaringan distribusi Bendul Merisi dengan metode *Integer Linear K-Means Clustering* dengan tujuan memperoleh solusi optimum penempatan dan jumlah minimal *PMU* yang di pasang. Tahap kedua adalah estimasi daya dan tegangan pada bus yang tidak terpasang *PMU* menggunakan metode modifikasi *ANFIS Hybrid PSO_GA.* Tahap ketiga pengujian simulasi *hardware* untuk rancangan pada penelitian ini untuk data *offline* dan *online*. Dalam penelitian juga dilakukan pengujian terhadap jaringan transmisi Jawa-Bali 500 kV.

Hasil penelitian dengan penerapan metode *Integer Linear K-Means Clustering* pada jaringan distribusi Bendul Merisi, maka dari 11 PMU yang seharusnya, dapat cukup diwakili 3 *PMU* saja, sehingga terjadi pengurangan sebesar 73%. Untuk estimasi tegangan dan daya pada saluran yang tidak terpasang *PMU* dengan metode modifikasi *ANFIS Hybrid PSO_GA* memiliki tingkat akurasi 99%. Hasil pengujian simulasi hardware menunjukkan kinerja yang baik. Adapun hasil pengujian pada jaringan transmisi Jawa-Bali 500 kV terjadi pengurangan jumlah PMU sebesar 84 %.

Kata kunci : *smart grid*, optimasi, *Interger linear K-Mean Clusterings,* simulasi *hardware offline* dan *online, ANFIS Hybrid PSO_GA.*

<Halaman ini sengaja dikosongkan>

OPTIMUM PLACEMENT OF MEASURING DEVICES ON A DISTRIBUTION NETWORK USING THE *INTEGER LINEAR K-MEANS CLUSTERING* METHOD

Name	: Riny Sulistyowati
NRP	: 07111560010004
Promotor	: Prof. Ir. Mochamad Ashari, M.Eng., Ph.D
Co promotor I	: Dedet Candra Riawan, S.T, M.Eng, Ph.D
Co promotor II	: Dr. Rony Seto Wibowo, S.T., M.T

ABSTRACT

The smart electricity network, also known as the smart grid, is a form of technological transformation and reform in the electricity industry. In order to obtain the expected smart-grid performance, analysis results are needed that are able to balance the dynamic conditions of the system. The data is implemented in the form of the installation of measurement devices or also known as the Phasor Measurement Unit (PMU). The problem becomes complex because many PMU devices are required to be installed in the power grid. So that optimization is needed to reduce the number and placement of PMUs that can control all electricity networks.

This study discusses the optimization method to reduce the number and placement of PMUs in distribution networks using the Integer Linear K-Means method. This method is a combination of the Integer Linear method and K-Means method. The advantage of this method is that it is able to analyze non-linear data with a faster and more accurate iteration process. The Linear Integer Method is used to generate data on buses installed by PMU and K-Means for the clustering process.

The first stage is to create a simulation model using many optimized PMUs on the Bendul Merisi distribution network with the Integer Linear K-Means Clustering method with the aim of obtaining the optimum solution for placement and the minimum number of PMUs installed. The second stage is the power and voltage estimation on the bus that is not installed by the PMU using the ANFIS PSO_GA hybrid modification method. The third stage is testing the hardware simulation for the design in this study for offline and online data. The research also conducted testing of the Java-Bali 500 kV transmission network.

The results of the study by applying the Integer Linear K-Means Clustering method to the Bendul Merisi distribution network, from the 11 PMUs that should have been represented, only 3 PMUs could be represented, resulting in a reduction of 73%. For the estimation of voltage and power on lines that are not installed PMU with the modified ANFIS PSO_GA method has an accuracy rate of 99%. Hardware simulation test results show good performance. The test results on the Java-Bali 500 kV transmission network showed a reduction in the number of PMUs by 84%.

Keywords: smart grid, optimization, linear interger K-Mean Clusterings, offline and online hardware simulation, ANFIS PSO_GA hybrid.

<Halaman ini sengaja dikosongkan>

KATA PENGANTAR

Puji syukur ke hadirat Allah SWT, karena penulis dapat menyelesaikan penelitian dan penulisan buku Disertasi ini. Pada kesempatan ini, penulis menyampaikan rasa terimakasih kepada :

1. Prof. Ir. Mochamad Ashari, M.Eng., Ph.D; selaku promotor yang telah memberikan bimbingan, motivasi untuk penelitian hingga publikasi serta penyelesaian studi.
2. Dedet Candra Riawan, S.T, M.Eng, Ph.D dan Dr. Rony Seto Wibowo, S.T., M.T selaku co promotor yang telah memberikan bimbinGAn dengan petuah-petuah bijak.
3. Dr. Djuwari, ST S.T, Heri Suryotmojo,S.T., M.T., Ph.D dan Dr, M.Eng. Ardyono Priyadi, S.T., Eng sebagai penguji yang telah memberikan masukan, bimbingan dan saran.
4. Prof. DR. Ir. Yoyon Kusnendar Suprapto M.S. dan Prof. Ir. Ontoseno Penangsang, M.Sc., Ph.D selaku reviewer buku disertasi.
5. Kemenristek Dikti, untuk beasiswa BPPDN yang diberikan untuk penyelesaian studi S3.
6. PT. Persero Perusahaan Listrik Negara Cabang Surabaya Selatan untuk mendukung penelitian ini.
7. Pembina, ketua, pengurus Yayasan Pendidikan Teknik Surabaya (YPTS) dan sivitas akademika Institut Teknologi Adhi Tama Surabaya; khususnya rekan-rekan dosen Elektro ITATS yang telah memberikan dukungan dalam melewati semua proses selama studi.
8. Manajemen Program Pasca Sarjana DTE, FTE – ITS beserta seluruh staf administrasi atas dukungannya terhadap kelancaran urusan pemberkasan dan administrasi.
9. Orang tua, suami dan anak-anak tercinta, serta keluarga yang senantiasa memberikan semangat dan doa untuk kelancaran penyelesaian studi.
10. Teman-teman yang masih aktif dan telah menyelesaikan studi S3 dari Laboratorium KE Teknik Elektro ITS; khususnya DR. Antonius, DR. DwiAna dan DR. Asepta untuk diskusi dan dukungannya.

11. Semua pihak yang belum disebutkan dan memberikan kontribusi dalam penyelesaian studi penulis.

Akhir kata, penulis berharap adanya pendapat, koreksi, kritik dan saran untuk penyempurnaan disertasi ini. Akhir kata, semoga buku disertasi ini bermanfaat bagi yang memerlukan.

Surabaya, Januari 2021

Penulis

PUBLIKASI

A. Seminar Internasional

1. Riny Sulistyowati, Mochamad Ashari, Dedet Candra Riawan "*PV Farm Placement And Sizing Using* GA *for Area Development Plan Of Distribution Network*" ,International Seminar On Intelegent Technology And ITS Application , Fakultas Teknik Institut Sepuluh Nopember (ITS) Surabaya, ISBN : 978-1-5090-3871-8, pp. 513-518, 2016.
2. Riny Sulistyowati, Mochamad Ashari, Dedet Candra Riawan "*Pengelompkanng Based Optimal Sizing And Placement Of PV-DG Using Neural Network*", *2015 International Conference on Energy, Enviroment and Information System (ICENIS)*, IEEE Conference Proceeding, DOI: , 2016.

B. Jurnal International

1. Riny Sulistyowati, Mochamad Ashari, Dedet Candra Riawan, Rony Seto Wibowo, "*Optimum Placement of Measurement Devices on Distribution Networks using Integer Linear K-Means Clustering Method*," Vol. 96 No. 10 DOI:10.15199/48.2020.10.23, *Przglad Elektrotechnizy* terindeks Scopus Quartile 3.
2. Riny Sulistyowati, Mochamad Ashari, Dedet Candra Riawan, Rony Seto Wibowo, "*Power and Voltage Estimation Of Unobserved Bus On Distribution Network Using ANFIS Algorithm With The Modified PSO*-GA *Hybrid*" *International Review of Electrical Engineering (I.R.E.E.);* Terindeks Scopus Quartile 2.

C. HAKI dan hak Cipta

1. Riny Sulistyowati, Mochamad Ashari, Dedet Candra Riawan, Rony Seto Wibowo, "*HARDWARE* MONITORING INTERNET OF THINGS" PATEN: No.P00202000082, 6 Januari 2020.

2. Riny Sulistyowati, Mochamad Ashari, Dedet Candra Riawan, Rony Seto Wibowo, “Aplikasi Kontrol Menggunakan Ardouno Untuk *Internet of Things (IoT)*, Hak cipta No.EC00202000468, 6 Januari 2020.

DAFTAR ISI

LEMBAR PENGESAHAN ..... **iii**
PERNYATAAN KEASLIAN DISERTASI ..... **v**
ABSTRAK ..... **vii**
ABSTRACT ..... **ix**
KATA PENGANTAR ..... **xi**
PUBLIKASI ..... **xiii**
DAFTAR GAMBAR ..... **xviii**
DAFTAR TABEL ..... **xxii**
DAFTAR SINGKATAN DAN NOTASI ..... **xi**
BAB 1 PENDAHULUAN **1**
1.1 Latar Belakang 1
1.2 Perumusan Masalah 8
1.3 Tujuan Penelitian 8
1.4 Kontribusi dan Posisi Penelitian 8
BAB 2 OPTIMASI JUMLAH DAN PENEMPATAN *PMU* **11**
2.1 Metode *Network Topology* 15
2.2. Teknik *Cluster* pada Jaringan Distribusi Bendul Merisi 17
2.2.1 *Cluster* Hirarki 17
2.2.2 *Cluster* Partisional 18
2.3 *Neural Network Cluster K – Means* 19
2.4 Teori Estimasi untuk Jaringan Distribusi Bendul Merisi 22
2.4.1 Teori *Particle Swarm Optimization (PSO)* 22
2.4.2 *Genetic Algorithm* 24
2. 5. *Internet Of Things (IOT)* 26
BAB 3 OPTIMASI PENEMPATAN *PMU* PADA JARINGAN DISTRIBUSI 20 KV SURABAYA SELATAN DAN TRANSMISI JAWA- BALI 500KV **29**
3.1 Metode *Cluster PMU* Menggunakan Algoritma *Integer Linear K-Means* 34
3.2 Analisa dengan Metode *Integer Linear K-Means Clustering* Pada Jaringan Distribusi di Bendul Merisi 37

3.3 Analisa Jaringan dengan Metode *Integer Linear K-Mean* Pada Jaringan Transmisi 53

3.4 Perbandingan *Cluster Integer Linear K-Mean* dengan Metode *Artificial Intelegent* yang lain 61

BAB 4 ESTIMASI DAYA DAN TEGANGAN UNTUK BUS YANG TIDAK TERPASANG *PMU* PADA SISTEM JARINGAN 65

4.1 Algoritma Estimasi pada sistem Jaringan Distribusi 65

4.2 Perencanaan Estimasi Daya dan Tegangan menggunakan Modifikasi *ANFIS Hybrid PSO*-GA 65

4.3 Algoritma *ANFIS* dengan modifikasi *hybrid PSO-GA* 69

4.4 Algoritma Implementasi modifikasi *ANFIS* dengan *Hybrid PSO-GA* pada jaringan Distribusi di Bendul-Merisi 74

BAB 5 VERIFIKASI HASIL ESTIMASI PADA JARINGAN DISTRIBUSI SURABAYA SELATAN DI BENDUL MERISI 81

5.1 Analisa Aliran Daya menggunakan Simulasi 81

5.2 Perancangan Sistem *Hardware* dan *Software)* 83

5.2.1 Blok Diagram *Hybrid* Sistem *Online dan Offline* 83

5.2.2 Diagram Alir Pengujian dan Perancangan sistem 85

5.3 Perancangan *Software Simulink* 87

5.3.1 Data Pengujian Simulasi pada Sistem 88

5.3.2 Pengujian Simulasi pada Sistem 89

5.3.3 Analisa Akurasi Simulasi pada Sistem 95

5.4 Perancangan *Hardware* dan Monitoring *Internet Of Thing (IOT*) 98

5.5 Pengujian *Hardware* 99

5.6 Pengujian *Internet of Things (IOT)* 102

BAB 6 PENUTUP 112

6.1 Kesimpulan 113

6.2 Saran 114

DAFTAR PUSTAKA

LAMPIRAN I 1

LAMPIRAN II 2
LAMPIRAN III 3
LAMPIRAN IV 6
LAMPIRAN V 13
LAMPIRAN VI 21

<Halaman ini sengaja dikosongkan>

DAFTAR GAMBAR

Gambar 1.1 *Road Map* Penelitian 10
Gambar 2.1 *Single Line diagram* contoh analisis menggunakan metode *Network Topology* 15
Gambar 2.2 *Hierarchical Cluster* 18
Gambar 2.3 Proses *cluster* obyek menggunakan metode *K-Means* 18
Gambar 2.4 *K-Means* dua *cluster* 21
Gambar 2.5 *K-Means* tiga *cluster* 22
Gambar 2.6 Alur kerja Algoritma *PSO* 24
Gambar 2.7 Alur kerja *Genetic Algoritm* 25
Gambar 2.8 Semua aktivitas manusia terhubung Internet 27
Gambar 2.9 Blok diagram sistem 28
Gambar 3.1 *Single Line* Jaringan Distribusi di Bendul Merisi Surabaya-Indonesia 30
Gambar 3.2 Algoritma *Integer Linear K-Means* 36
Gambar 3.3 Matrik koneksi bus biner Bendul Merisi..... 37
Gambar 3.4 Hasil visualisasi dari Tabel 3.9 41
Gambar 3.5 Hasil visualisasi dari dengan perubahan anggota *cluster* 43
Gambar 3.6 Proses akhir dari *cluster* yang mencapai konvergen 46
Gambar 3.7 Visualisasi hasil penentuan *cluster* yang menggunakan parameter tegangan, arus dan impedansi 51
Gambar 3.8 Grafik Jumlah *cluster* optimal 53
Gambar 3.9 *Single line* diagram sistem Bus Jawa-Bali 500 kV 53
Gambar 3.10 Hasil *cluster* 19 bus Jawa-Bali 500 kV Grafik Jumlah *cluster* optimal 60
Gambar 3.11 Grafik jumlah *cluster* optimal 61
Gambar 4.1 Rangkaian pengganti antar bus 66
Gambar 4.2 Daya aktif selama 24 jam dalam 7 hari 68
Gambar 4.3 Daya reaktif selama 24 jam dalam 7 hari 68
Gambar 4.4 Beban (*Load*) selama 24 jam di 7 hari 68
Gambar 4.5 Arsitektur *ANFIS* 69

Gambar 4.6 Modifikasi algoritma *ANFIS* dengan *hybrid PSO*-GA 71
Gambar 4.7 Transformasi fitur dari parameter individual ke premis 71
Gambar 4.8 Diagram Alur Penelitian 74
Gambar 4.9.a Perbandingan hasil estimasi dan pengukuran daya di*cluster* 1..... 76
Gambar 4.9.b Perbandingan estimasi dan pengukuran tegangan di*cluster* 1 76
Gambar 4.10.a Perbandingan hasil estimasi dan pengukuran daya di*cluster* 2... 77
Gambar 4.10.b Perbandingan hasil estimasi dan pengukuran tegangan di*cluster* 2 78
Gambar 4.11. a Perbandingan antara hasil estimasi dan pengukuran daya di*cluster* 3 79
Gambar 4.11. b Perbandingan hasil estimasi dan pengukuran tegangan di*cluster 3* 79
Gambar 5.1 *Single line* diagram di Bendul Merisi dengan Simulasi..... 82
Gambar 5.2 Proses Tahap Awal Pengujian *Hardware*..... 84
Gambar 5.3 Tahap Dua Pengujian *Hardware* 84
Gambar 5.4 Monitoring *Simulink On Line* dan *Off line* 85
Gambar 5.5 Diagram alir perancangan *Hardware* dan pengujian sistem..... 86
Gambar 5.6 Tampilan proses simulasi *Simulink* 87
Gambar 5.7 Hasil simulasi untuk *cluster* I pada bus 2, 3 dan 4 90
Gambar 5.8 Hasil simulasi untuk *cluster* II pada bus 5, 6, 7 dan 8 91
Gambar 5.9 Hasil simulasi untuk *cluster* III pada bus 9, 10 dan 11..... 92
Gambar 5.10 Grafik Perbandingan daya aktif dan daya reaktif 94
Gambar 5.11 a Grafik Perbandingan *error* daya aktif..... 95
Gambar 5.11 b Grafik Perbandingan *error* daya reaktif 95
Gambar 5.12 a. Grafik hasil *error* perbandingan daya aktif antara perhitungan dengan *running* Simulasi dengan *Simulink.* 96
Gambar 5.12 b. Grafik hasil *error* perbandingan daya reaktif antara perhitungan dengan Simulasi dengan *Simulink*..... 96
Gambar 5.13 Perancangan *Hardware Internet Of Things (IOT)*..... 97
Gambar 5.14 Perangkat *Hardware* 99
Gambar 5.15 *Output* dari sensor tegangan dan arus 99
Gambar 5.16 Program tampilan di server komputer ……..100

Gambar 5.17 Proses penerima data melalui di *smartphone* ……..100
Gambar 5.18 Tampilan di *smartphone*……101
Gambar 5.19 Tampilan data di *smartphone*…….101
Gambar 5.20 Diagram Alur Sistem Monitoring Smartphon……102
Gambar 5.21 Grafik hasil monitoring menggunakan *IOT* 103
Gambar 5.22 Grafik hasil monitoring menggunakan *IOT* 104
Gambar 5.23 Grafik hasil monitoring skala laboratorium selama 7 hari.....104
Gambar 5.24 Peta jarak lokasi *transmitte*r terhadap *receiver* *smartphone*.....105
Gambar 5.25 Grafik hasil monitoring……..106
Gambar 5.26 Grafik hasil monitoring uji lapangan selama 24 jam 107
Gambar 5.27 Grafik hasil monitoring uji lapangan selama 7 hari 108
Gambar 5.28 Grafik hasil monitoring uji lapangan jarak terhadap data diterima *smartphone* selama 7 hari 109

<Halaman ini sengaja dikosongkan>

DAFTAR TABEL

Tabel 2.1 Hasil Resume *Literatur Review* 14
Tabel 3.1 Data Saluran distribusi Bendul Merisi 29
Tabel 3.2 Tabel data drop tegangan 34
Tabel 3.3 Data Bendul Merisi dengan parameter tegangan dan arus 38
Tabel 3.4 Penentuan pusat *cluster* awal yang secara acak 39
Tabel 3.5 Jarak bus n ke pusat *cluster* 39
Tabel 3.6 Gabungan jarak bus ke pusat masing-masing *cluster* 40
Tabel 3.7 Penentuan anggota *cluster* 40
Tabel 3.8 Hasil penentuan anggota *cluster* pada iterasi pertama 41
Tabel 3.9 Hasil perhitungan menentukan pusat baru kelompok *cluster* 41
Tabel 3.10 Penentuan anggota *cluster* pada iterasi kedua 42
Tabel 3.11 Hasil penentuan anggota *cluster* pada iterasi kedua 43
Tabel 3.12 Pusat *cluster* baru hasil iterasi ke 2 43
Tabel 3.13 Menghitung jarak ke pusat dan anggota *cluster* iterasi 3 44
Tabel 3.14 Hasil penentuan anggota *cluster* iterasi 3 44
Tabel 3.15 Pusat *cluster* baru hasil iterasi ke 3 44
Tabel 3.16 Menghitung jarak ke pusat dan anggota *cluster* iterasi 4 45
Tabel 3.17 Hasil penentuan akhir anggota *cluster* 45
Tabel 3.18 Penentuan puasat *cluster* secara acak 46
Tabel 3.19 Jarak bus n ke pusat *cluster* pada jarak *Euclidean* pada iterasi 1 47
Tabel 3.20 Penentuan anggota *cluster* dari jarak terpendek iterasi 1 47
Tabel 3.21 Hasil penentuan anggota *cluster* iterasi 1 48
Tabel 3.22 Tegangan, arus, dan impedansi dalam *cluster*1 48
Tabel 3.23 Pusat baru dari hasil iterasi ke 1 48
Tabel 3.24 Menghitung jarak ke pusat dan anggota *cluster* iterasi 2 48
Tabel 3.25 Hasil penentuan anggota *cluster* iterasi 2 49
Tabel 3.26 Pusat baru pada iterasi 2 49
Tabel 3.27 Menghitung jarak ke pusat dan anggota *cluster* iterasi 3 50
Tabel 3.28 Pusat baru pada iterasi 3 50
Tabel 3.29 Menghitung jarak ke pusat dan anggota *cluster* pada iterasi 4 50
Tabel 3.30 Hasil penentuan akhir anggota *cluster* 51

Tabel 3.31 Optimasi *PMU* 11 Bus metode *Elbow* 52
Tabel 3.32 Data Saluran Jaringan Transmisi Jawa-Bali 54
Tabel 3.33 Perhitungan hasil aliran daya sistem *Newhon Rhapson* 55
Tabel 3.34 Data tegangan dan sudut jaringan Transmisi Jawa-Bali..... 55
Tabel 3.35 Penentuan Pusat *cluster* awal secara acak..... 56
Tabel 3.36 Perhitungan *Distance* terhadap pusat *cluster* pada iterasi 1 57
Tabel 3.37 Hasil anggota dari masing- masing *cluster* pada iterasi 1 57
Tabel 3.38 Hasil penentuan anggota *cluster*..... 58
Tabel 3.39 Hasil pusat *cluster* baru 58
Tabel 3.40 Optimasi *PMU* 19 Bus Jawa-Bali 500 KV metode *Elbow* 59
Tabel 3.41 Optimasi PMU 19 Bus Jawa-Bali 500 KV metode Elbow 60
Tabel 3.42 Perbandingan Optimasi penempatan jumlah *PMU* untuk
11 Bus dan 19 bus 62
Tabel 4.1 Hasil penempatan *PMU* pada jaringan distribusi Bendul Merisi 66
Tabel 4.2 Perbandingan pengukuran serta estimasi daya dan tegangan
Untuk *cluster* 1 75
Tabel 4.3 Perbandingan pengukuran serta estimasi daya dan tegangan
Untuk *cluster* 2 77
Tabel 4.4 Perbandingan pengukuran serta estimasi daya dan tegangan
Untuk *cluster* 3 78
Tabel 4.5 Akurasi hasil simulasi dengan *ANFIS Hybrid PSO-GA* 80
Tabel 5.1 Tabel nilai beban puncak berdasarkan pengukuran di Bendul
Merisi 82
Tabel 5.2 Tabel data tegangan dan arus jam 8:00 – 10:00 88
Tabel 5.3 Tabel data tegangan dan arus jam 11:00 – 13:00 88
Tabel 5.4 Tabel hasil simulasi pada *Simulink Matlab* 93
Tabel 5.5 Tabel hasil *error* perbandingan antara perhitungan dengan segi
daya dengan simulasi *simulink* 94
Tabel 5.6 Hasil *error* perbandingan dengan hasil simulasi *Simulasi*
dengan *simulik* 95
Tabel 5.7 Hasil pengujian laboratorum pada hari Senin 102
Tabel 5.8 Hasil pengujian laboratorum pada hari Minggu.....103

Tabel 5.9 Hasil pengujian lapangan pada hari Senin 106
Tabel 5.10 Hasil pengujian laboratorum pada hari Minggu 107
Tabel 5.11 Hasil pengujian lapangan selama 7 hari.....108

<Halaman ini sengaja dikosongkan>

DAFTAR SINGKATAN DAN NOTASI

A1, A2	*Premise parameters in ANFIS architecture*
a, b, c	*Adjustable parameters for activation function*
B1, B2	*Premise parameters in ANFIS architecture*
Bx	*Branch current in network (x=1,2, …. 10)*
C1, C2	*Crossover variables (new individual)*
Cr	*Crossover ratio*
p, q, r	*Neuron parameters*
I	*Current*
ind_GA	*Individual GA*
ind_PSO	*Individual PSO*
k	*Constanta.*
n_ind	*Individuals*
V	*Voltage*
Vbus	*Bus voltage*
W	*Weight*
W1, W2	*Weights at layer 2 and 3*
X	*Voltage parameter in ANFIS*
Y	*Current parameter in ANFIS*
Z	*Impedance*

Greek Symbol

λ1	*Random value with range between 0 and 1,*

Abbreviations

ADS	*Active Distribution System.*
ANFIS	*Adaptive Neuro Fuzzy Inference System*
ACPSO	*Adaptive Cat Particle Swarm Optimization*
ESS	*Estimation System Storage*
GA	*Genetic Algorithm*
GD	*Gradient Descent*

HPSOGA	*Hybrid Particle Swarm Optimization and Genetic Algorithm*
L-PSO	*Learning Particle Swarm Optimization*
LSE	*Least Square Estimator*
Maxiter	*Maximum Iteration*
MAPE	*Mean Absolute Perscentage Error*
MSE	*Mean Square Error*
MR	*Mixture Ratio*
Mr	*Mutation ratio .*
NN	*Neural Network*
PSO	*Particle Swarm Optimization*
PSO-GA-KF	*Particle Swarm Optimization - Genetic Algorithm-Kalman Filter*
PMU	*Phasor Measurement Unit*
PS	*Power System*
PSSE	*Power System State Estimation*
SE	*State Estimation*
SSE	*Sum Square Error*

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Smart grid adalah jaringan listrik pintar yang mampu mengintegrasikan aksi-aksi atau kegiatan dari semua pengguna, mulai dari pembangkit sampai ke konsumen dengan tujuan agar efisien, berkelanjutan, ekonomis dan aman [1]. *Smart grid* membutuhkan penggunaan komunikasi dua arah yang tangguh, sensor dalam jumlah besar dan teknologi komputasi. *Smart grid* mampu melaksanakan komunikasi secara terpadu untuk area yang luas meliputi, respon permintaan daya, otomasi distribusi, pengendalian pengawasan dan akuisisi data (SCADA) [2]. Komunikasi sistem yang terpadu mampu mengontrol sistem jaringan listrik secara *real-time,* pertukaran informasi dan data sehingga dapat mengoptimalkan keandalan sistem, pemanfaatan aset, dan sistem keamanan [3]. Teknologi *smart grid* membutuhkan data untuk mengontrol pendistribusian arus dan tegangan serta memantau kondisi peralatan dan strategi pengendalian sistem. Pada analisa Aliran daya terdapat dua metode yaitu metode matematik dan metode heuristik. Sedangkan metode aliran heuristik menggunakan sistem cerdas. Metode matematik untuk *Power Flow* yang banyak di gunakan saat ini adalah metode berbasis antara lain : *Newton Raphson,* Gauss – *Seidel, Decoupled* dan lain - lain. Metode analisis ini yang di gunakan pada penelitian ini adalah metode heuristik. Simulasi menggunakan perangkat lunak tersebut memerlukan parameter-parameter data yang didapat dari hasil pengukuran manual di lapangan dan analisisnya dilakukan secara *off-line* sehingga hasil analisis hanya memberikan informasi statis berdasarkan saat pengukuran dilakukan. Untuk smart grid diperlukan suatu hasil analisis yang akurat dan bersifat kontinyu yang memberikan informasi secara kontinyu terhadap perubahan yang terjadi pada sistem secara dinamis.

Phasor Measurement Unit (PMU) merupakan perangkat kontrol yang dapat digunakan untuk melakukan pengukuran secara kontinyu dan memberikan data real time. *PMU* ini dipasang pada setiap bus di sistem jaringan. Pemasangan *PMU* di setiap bus pada sistem jaringan di butuhkan untuk mengukur dan mengontrol pada sistem jaringan supaya semua nilai dan kondisinya bias di

monitor setiap saat. Namun di butuhkan harga yang mahal untuk sebuah *PMU*. Untuk itu, di butuhkan reduksi *PMU*, tidak semua bus dipasang *PMU*, hanya bus-bus tertentu yang dipasang *PMU*. Bus-bus yang terpasang *PMU* (*observed bus*) dianggap dapat mewakili pengukuran di bus-bus yang tidak terpasang *PMU* (*Unobserved bus*).

Penelitian yang telah dilakukan bertujuan untuk mereduksi jumlah *PMU* yang harus dipasang dan menetukan secara optimal lokasi pemasangan *PMU*. Beberapa peneliti telah melakukan dengan metode metode matematis, dan metode heurisitik. Penelitian matematis antara lain: *Integer Linear Programming (ILP)* untuk mengurangi jumlah *PMU*. Penelitian ini menerapkan konsep *zero injection* dan hasilnya beberapa meter pengukuran pada bus yang dapat diwakili oleh bus lain [4]. Penempatan optimal *PMU* dengan *Binary Integer Programming (BIP)* pada sistem dengan konsep *incomplete observability* yang memberikan variabel dengan pilihan nilai 0 dan 1. Untuk nilai 1 menunjukkan tempat yang harus dipasang *PMU*. Simulasi dengan konsep BIP *incomplete observability* memberikan hasil jumlah *PMU* yang lebih sedikit dibandingkan dengan sistem tanpa konsep *incomplete observability* [5]. Penelitian menggunakan metode *Weight Least Square (WLS*) berdasarkan parameter tegangan dan arus untuk penempatan *PMU* pada sistem distribusi untuk state estimasi di area yang luas dan menunjukkan hasil yang baik [6-8]. Berikutnya adalah teknik menggunakan daya untuk penempatan dan pemantauan *PMU* dan hasilnya dapat mengurangi jumlah penggunaan *PMU* [9]. Algoritma *Binary Imperialistic Competition (BICA)* dengan menggunakan konsep *zero injection* bus, konsep ini menggunakan aturan konvensional dan aturan baru untuk mengurangi jumlah *PMU* dan hasilnya memiliki lebih banyak kovarians [10]. Studi lain yang menggunakan metode *Approximately Optimal Position PMU (AOPP)* digunakan untuk penempatan *PMU* dengan mengidentifikasi ruang pencarian menggunakan konsep deterministik, dan hasilnya ditemukan di ruang tertentu [11]. Merancang algoritma *Weighted Least Square* untuk Optimal Penempatan *PMU* untuk masalah minimalisasi kuadratik dengan faktor keputusan kontinu diformulasikan sesuai dengan batasan observasi nonlinear. Solusi optimal diperoleh dengan tidak membatasi metode *weighted least squar*e nonlinear. Jumlah pengurangan *PMU*

dengan bus injeksi nol sulit dilakukan menggunakan *Weighted Least Square* [12]. Metode *Biner Imperialistik Competation Algoritm (BICA*) untuk OPP di kembangkan pada penelitian ini dengan menggunakan konsep bus injeksi nol untuk semua pengamatan. Menggunakan aturan konvensional dan aturan baru untuk mengurangi jumlah *PMU* yang akan ditempatkan. State estimasi dan *error* kovarian lebih banyak [13]. Konsep *Random Component Outages (RCO)* OPP untuk estimasi pada sistem tenaga. Metode untuk pemilihan lokasi optimal menggunakan state estimasi dan *error* kovarian [14]. Metode ini untuk pengamatan estimasi daya berdasarkan topologis dan perhitungan numerik. Tahap 1 mengidentifikasi jumlah minimum *PMU* untuk membuat sistem dapat diamati secara topologis. Tahap 2 diusulkan untuk mengidentifikasi apakah penempatan *PMU* yang didapat menunjukkan hasil pengukuran Jacobian. Hasilnya mampu menjamin optimal global. Namun tidak sesuai untuk di terapkan pada sistem skala besar dengan ruang pencarian yang besar [15]. Sebuah *OPP* dengan formulasi linear untuk pencarian yang komplek. State estimasi diimplementasikan berdasarkan penempatan linear lengkap. Hasilnya meski perkiraan akurat, tetapi identifikasi ruang pencarian itu sulit [16]. OPP untuk penempatan *PMU* dengan metode *Approximately Optimal Placement PMU (AOPP)* juga di kembangkan untuk mengidentifikasi ruang pencarian. *AOPP* menggunakan konsep deterministik dengan secara rinci. Bus tetangGA digunakan untuk mendapatkan ruang pencarian yang terbatas [17]. Kelemahannya dari penelitian dengan analisa matematis yaitu variabel yang terlalu banyak menbutuhkan waktu yang lama dan perhitungan yang rumit maka algoritma heuristik yang diterapkan untuk penempatan *PMU* yang optimal.

Penelitian menggunakan metode heuristik, untuk penempatan *PMU* optimal untuk menjaga keutuhan jaringan dengan menggunakan algoritma *Binary Particle Swarm Optimization (BPSO)* yang dimodifikasi. Analisis observasi dilakukan berdasarkan aturan observasi topologis. Aturan baru ditambahkan untuk meminimalkan jumlah *PMU* agar dapat diamati secara lengkap. Hasilnya jumlah *PMU* minimal dan lokasi yang sesuai [18]. Penelitian *OPP* berdasarkan *Binary Particle Swarm Optimisation (BPSO)* dengan pemrograman kuadratik. Observasi pseudo diperkenalkan untuk kedalaman tingkat satu dan kedalaman tingkat dua

dengan dan tanpa pengukuran injeksi nol. Ini diuji pada sistem bus *IEEE-7, IEEE-14, IEEE-30 dan IEEE-57* menggunakan teknik *BPSO* . Hasilnya perbandingan antara injeksi nol dan tanpa injeksi nol harus ditingkatkan [19]. Metode *Improved Tabu Search (ITS)* untuk observasi yang lengkap dan prediksi estimasi *PMU* . Sistem ini dengan analisa numerik dan kecerdasan buatan. Algoritma prediksi *Predictive Out of Step (OOS)* diusulkan berdasarkan pengamatan perbedaan fasa tegangan di antara GArdu induk [20]. Optimasi berbasis biogeografi. Dua tujuan yang saling bertentanGAn minimisasi jumlah *PMU* dan maksimalisasi redundansi pengukuran dilakukan. Solusi optimal Pareto (grafik batang yang menunjukkan masalah berdasarkan urutan banyaknya jumlah kejadian) diperoleh berdasarkan jarak sortir dan jarak yang tidak didominasi. *Fuzzy* di gunakan untuk solusi optimal Pareto. Hasilnya keandalan sistem dengan analisis *fuzzy* kurang baik [21]. Optimal penempatan *PMU* dengan berdasarkan topologi menggunakan *Geografis Information System (GIS)*. Hasilnya melalui berbagai konfigurasi konektivitas *PMU* maka sistem jaringan handal [22]. Metode *Genetik Algoritm (GA)* digunakan untuk mengoptimalkan pada jaringan ditribusi agar jumlah *PMU*nya minimum secara berurutan tujuannya untuk menjamin pengamatan agar estimasi dalam keadaan linear [23]. Penelitian metode Algoritma Genetika digunakan juga menempatkan *PMU* dan monitoring SCADA untuk masalah state estimasi, seperti pengamatan jaringan untuk pengidentifikasian data dan akurasi, hasilnya estimasi kurang baik [24].Penelitian dengan Algoritma Genetika untuk topologi baru masalah penempatan optimal *PMU* bersama dengan unit pengukuran konvensional. State estimasi dapat dicapai dengan meningkatkan akurasi dan kekebalan terhadap kontinjensi jaringan listrik [25].

Dengan metode heuristik dapat juga di selesaikan dengan pendekatan menggunakan teknik *cluster* untuk masalah penempatan *PMU*. Penelitian memaparkan teknik *cluster* bus baru menggunakan jaring saraf tiruan Kohonen untuk membentuk kelompok bus *cluster* untuk kestabilan tegangan listrik [26]. Metode *cluster* Kohonen pada jaringan untuk masalah karakteristik dinamis listrik pada beban, yang hasilnya terbukti memuaskan dengan menggunakan karakteristik sintesis beban [27]. Metode pendekatan untuk menentukan tingkat kerugian energi dengan menggunakan *cluster K – Means* dan hasilnya mampu

mengurangi *losess* 57% [28]. Metode *cluster K – Means* juga di gunakan untuk mengembangkan pendekatan *Voltage Controlled Amplifier (VCA)* untuk manajemen daya reaktif dan kontrol tegangan dan hasilnya sangat baik [29]. Penelitian berbasis Algoritma Genetika untuk masalah penempatan *PMU* dan hasilnya mampu untuk mengoptimasi secara optimal [30]. Penelitian menggunakan *Particle Swarm Optimization (PSO)* untuk aplikasi dengan monitoring menggunakan *Wireless Sensor Networks (WSN)* [31].

Penelitian yang dilaksanakan pertama menyajikan konsep baru pada menggabungkan metode *Integer Linear* dan metode *K-Mean* untuk mengelompokkan dan mengoptimalkan penempatan *PMU* di jaringan distribusi di Bendul Merisi, Surabaya-Indonesia. Kombinasi ini memiliki keuntungan karena dapat menghitung data acak non-integer dengan hasil yang lebih akurat. Sehingga mendapatkan hasil jumlah optimalisasi jumlah *PMU* dan lokasi penempatan *PMU* [32].

Setelah penempatan *PMU* yang optimal di tentukan, maka bus yang tidak terpasang *PMU* dilakukan estimasi daya dan tegangan. Ada dua kelompok algoritma estimasi, yaitu algoritma estimasi klasik dan estimasi metaheuristik. Algoritma estimasi klasik biasanya digunakan dalam kasus-kasus di mana nilai target diatur (diawasi) sedangkan estimasi metaheuristik umumnya digunakan dalam kasus-kasus dengan nilai target yang tidak diawasi. Pendekatan yang paling umum digunakan adalah nilai berbasis populasi dan acak. Hasil yang diperoleh dipengaruhi oleh hasil terbaik individu dan nilai-nilai individu lainnya. Estimasi tersebut antara lain, *Particle swarm optimization (PSO)* adalah algoritma optimisasi metaheuristik. Dalam beberapa penelitian disebutkan bahwa algoritma *PSO* dengan cepat menyatu dalam ruang solusi lokal yang optimal atau yang biasa disebut *premature convergence*. Ketika algoritma *PSO* mendekati solusi akhir, algoritma tidak akan dapat menemukan ruang solusi baru. Ini membuat *PSO* terjebak dalam menemukan ruang solusi lokal yang optimal. Untuk menghindari optimal lokal, dalam penelitian menggunakan algoritma *Adaptive Mutation Particle Swarm Optimization*, dilakukan konfigurasi ulang algoritma. Fitur yang menonjol dari algoritma ini adalah kemampuan algoritma *Particle Swarm Optimization (PSO)* untuk mencapai dari optimal lokal sangat ditingkatkan oleh

pindah silang. Hasilnya mampu mencapai global optimal [33-36]. Metodologi penelitian rekonfigurasi yang didasarkan pada algoritma *PSO* diuji dalam dua model jaringan distribusi radial *IEEE*. Hasil penelitian menunjukkan bahwa konfigurasi jaringan distribusi yang optimal dapat mengurangi kehilangan daya dan meningkatkan tegangan profil jaringan secara signifikan [37]. Untuk menemukan ruang solusi baru, *Genetic Algortm (GA)* menggunakan operasi pindah silang untuk mendapatkan individu baru. Dalam penelitian ini, Algoritma Genetika digunakan untuk memperkirakan penempatan PV dalam distribusi pembangkit listrik. Hasilnya menunjukkan tingkat akurasi yang baik dibandingkan dengan perkiraan klasik [38-39]. Selain itu, penelitian yang dikembngkan saat ini dengan menggunakan algoritma *hybrid*, salah satunya adalah modifikasi Algoritma Genetika dengan metode Lagrange-Multiplier memberikan solusi yang efisien untuk merumuskan masalah dalam pengujian sistem tenaga[40]. Algoritma hybrid lain, Optimisasi *Bi-level Fuzzy Multi Objektif* digunakan untuk memodelkan perencanaan *Energy Storage System (ESS*) dalam *Active Distribution System (ADS*). Hasil penelitian menunjukkan bahwa model ini memiliki keunggulan obyektif dan lebih baik [41-42]. Demikian pula, kombinasi dari sistem pendekatan *Adaptation Neoro Fuzzy Inference System (ANFIS)* dan *PSO* digunakan untuk memperkirakan permintaan listrik. *PSO* digunakan untuk mengoptimalkan fungsi keanggotaan *ANFIS* dengan mengubah parameter premis dan parameter konsekuensi *ANFIS*. Hasil penelitian menunjukkan bahwa akurasi metode yang diusulkan meningkat secara signifikan dibandingkan dengan sistem *ANFIS* standar [43-45]. *PSO* dan *GA* adalah algoritma *hybrid* menggunakan paradigma *Learning PSO (L-PSO).* Hasilnya mampu mencapai optimal global dan efisiensi serta menunjukkan tidak hanya diversifikasi tetapi juga berkualitas tinggi [46]. Demikian juga, metode *Adaptive Cat Particle Swarm Optimization (ACPSO)* menggunakan metode adaptif dengan memilih mode pencarian berdasarkan pada *Mixture Ratio (MR).* Hasil penelitian menunjukkan bahwa *ACPSO* memiliki kinerja yang baik [47]. Penelitian *Binary Genetic Swarm Optimization* adalah kombinasi *GA* dan *PSO* yang digunakan untuk pemilihan fitur. Tujuannya adalah untuk menggabungkan kapasitas eksploitasi *GA* dengan kapasitas eksplorasi *PSO*. Secara keseluruhan metode dengan algoritma *hybrid* menunjukkan superior

dibandingkan tanpa *hybrid* [48-49]. Metode hybrid yang lebih kompleks termasuk *PSO* dan algoritma genetika dibantu oleh filter Kalman yaitu *PSO-GA*-KF untuk memperkirakan keadaan harmonik dinamis dalam sistem tenaga. Hasil simulasi menunjukkan bahwa algoritma *hybrid* yang diusulkan memiliki estimasi yang lebih akurat, konvergensi yang lebih cepat, dan ketahanan yang lebih baik terhadap noise dibandingkan dengan *Kalman Filter(KF)* konvensional[50].

Selanjutnya Optimalisasi hybrid *ANFIS PSO* dan algoritma genetik *PSO* diusulkan untuk meminimalkan model fungsi yang disederhanakan . Algoritma ini disebut *Hybrid Particle Swarm Optimization dan Genetic Algorithm (HPSO*GA*). HPSO*GA didasarkan pada tiga mekanisme. Mekanisme pertama adalah menerapkan optimasi kawanan partikel untuk menyeimbangkan antara proses eksplorasi dan eksploitasi dalam algoritma yang diusulkan. Mekanisme kedua adalah proses pengurangan dimensi dan proses pembagian populasi dengan membagi populasi menjadi sub-populasi dan menerapkan operator *crossover* aritmatika di setiap sub populasi untuk meningkatkan keragaman pencarian dalam algoritma. Mekanisme yang terakhir diterapkan untuk menghindari konvergensi prematur dan menghindari terjebak pada minimum lokal dengan menggunakan operator pindah silang genetik di seluruh populasi. Hasil percobaan numerik menunjukkan bahwa algoritma yang diusulkan bekerja dengan baik [51 -53]. Dari berbagai metode yang digunakan untuk *Power System State Estimation (PSEE)* adalah menentukan estimasi terbaik dari variabel keadaan yang diwakili oleh tegangan dan sudut tegangan semua bus menggunakan proyeksi statistik. Karena kerumitan dan non-linearitas sistem tenaga, perlu untuk menggunakan metode yang lebih maju untuk analisis dan kontrol dalam realtime. Ketika memecahkan masalah state estimasi, metode yang diusulkan harus mempertimbangkan set pengukuran yang tersedia di pusat kontrol dan memprediksi pengukuran yang tidak dipasang di *PMU* [54-55].

Selanjutnya, kontribusi yang diusulkan dari penelitian ini adalah modifikasi kombinasi *ANFIS* hybrid *PSO-GA*. Penelitian tahap kedua dilakukan estimasi untuk bus yang tidak terpasang *PMU* dengan menggunakan metode estimasi. modifikasi *ANFIS Hybrid PSO-GA* berdasarkan hasil referensi penelitian menunjukkan hasil terbaik untuk estimasi [56].

Penelitian tahap ketiga adalah validasi dengan perancangan *Hardware* yang di gunakan untuk monitoring nilai daya dan tegangan menggunakan sistem *hibrid off line* dan *on line* atau disebut *Internet of Things (IoT).*

1.2 Perumusan Masalah

1. Bagaimana cara mengembangkan metode untuk pengurangan jumlah *PMU* dan menentukan secara optimal penempatan *PMU*.
2. Bagaimana cara mengembangkan metode untuk mengestimasi daya dan tegangan pada bus yang tidak terpasang *PMU* (*Unobserved* bus) pada sistem jaringan.
3. Bagaimana cara memvalidasi perangkat keras simulasi yang telah dibuat yang diterapkan pada sistem jaringan Distribusi Surabaya Selatan.

1.3 Tujuan Penelitian

Berkaitan dengan permasalahan tersebut, tujuan yang ingin dicapai pada penelitian ini adalah :

1. Menggunakan metode *Integer Liner K- means* untuk mengurangi jumlah *PMU* dan menentukan secara optimal penempatan *PMU* di lokasi yang sesuai.
2. Mengembangkan metode modifikasi *ANFIS hybrid PSO–GA* pada sistem jaringan untuk mengetahui estimasi daya dan tegangan pada bus yang tidak terpasang *PMU*.
3. Melakukan validasi menggunakan sistem model simulasi dan *prototype hybrid off line* dan *online* menggunakan data sistem jaringan Distribusi Surabaya Selatan di Bendul Merisi.

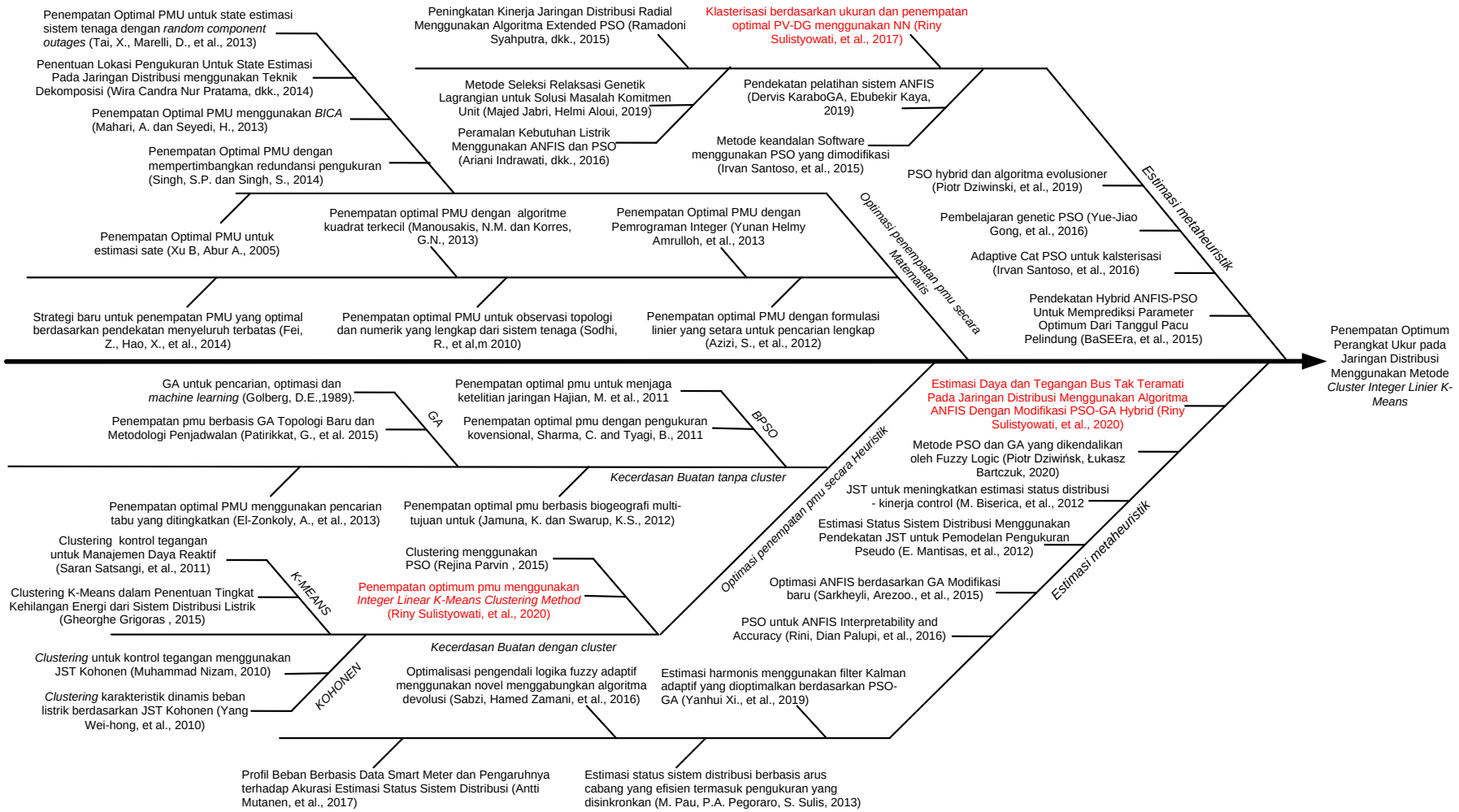
1.4 Kontribusi dan Posisi Penelitian

Penelitian ini difokuskan untuk menghasilkan 2 kontribusi sebagai luaran penelitian, yaitu :

1. Penggunakan metode optimasi penempatan dan menentukan jumlah PMU dengan *Integer Linear K-Means Clustering* yang telah di validasi menggunakan lokasi pada sistem Jaringan Surabaya Selatan di Bendul Merisi dan Transmisi Jawa-Bali 500 kV.

2. Pengembangan sistem pemodelan estimasi dengan metode modifikasi *ANFIS Hybrid PSO-GA.*

Posisi penelitian ini ditunjukkan pada *road-map* penelitian secara lengkap pada Gambar 1.1



Gambar 1.1. *Road Map* Penelitian

BAB 2 OPTIMASI JUMLAH DAN PENEMPATAN *PMU*

Pengoperasian sistem jaringan listrik memerlukan perencanaan, pengoperasian, dan pemantauan data yang akurat. Pemantauan dan pengukuran dilakukan oleh Gardu Induk secara terus menerus. Pengukuran dilakukan dengan menggunakan peralatan yang disebut dengan *Phasor Measurement Unit* (*PMU*). *PMU* ini sangat mahal harganya. Diperlukan inovasi baru untuk mengurangi jumlah *PMU* yang terpasang dengan optimasi jumlah dan penempatan *PMU* tetapi tetap dapat mengamati semua bus.

Penelitian tentang mereduksi jumlah *PMU* dan optimasi penempatan *PMU* serta mengestimasi daya dan tegangan yang tidak terpasang *PMU* telah dilakukan oleh beberapa peneliti. Berikut dijabarkan penelitian terdahulu terkait dengan topik disertasi ini. Beberapa peneliti telah melakukan dengan beberapa metode diantaranya metode *Integer Linear Programming (ILP)* untuk mereduksi jumlah *PMU*. Penelitian [4] ini menerapkan konsep *zero injection* dan hasilnya beberapa meter pengukuran pada bus yang dapat direpresentasikan oleh bus lain.

Peneliti [5-6] telah meneliti dengan menggunakan metode *Weight Least Square (WLS)* berdasarkan parameter tegangan dan arus untuk menempatkan *PMU* dengan Sistem Distribusi Pengukuran di area yang luas dan menunjukkan hasil yang baik.

Selanjutnya adalah penelitian estimasi daya untuk penempatan dan pemantauan *PMU* dan hasilnya dapat mengurangi jumlah penggunaan *PMU* [8]. Algoritma *Binary Imperialistic Competition (BICA)* dari penelitian yang sama menggunakan konsep bus injeksi nol, konsep ini menggunakan aturan konvensional dan aturan baru untuk mengurangi jumlah *PMU* dan hasilnya memiliki kovarian yang lebih banyak [9].

Penelitian lain yang menggunakan *metode Approximately Optimal Position PMU (AOPP)* digunakan untuk penempatan *PMU* dengan mengidentifikasi ruang pencarian menggunakan konsep deterministik, dan hasilnya ditemukan pada ruang tertentu [10].

Peneliti [11] menggunakan algoritma *Binary Particle Swarm Optimization (BPSO)* dengan penempatan data integer dan *PMU* pada lokasi yang sesuai.

Penelitian [12] menggunakan konsep *Binary Integer Programming (BIP)* dengan menggunakan vektor biner 1 dan 0. Algoritma ini akan memperbarui batas minimum dan maksimum, dan hasilnya kurang optimal.

Integer Linear dengan konsep *One Depth Unobservability* diperiksa dengan [34] dan hasilnya hanya satu bus yang tidak diawasi dan hasilnya kurang optimal. Penempatan *PMU* dengan *Integer Linear* dengan berdasarkan topologi jaringan juga diperiksa [35] dan hasilnya dapat ditentukan dengan memper-timbangkan bus yang terhubung, hasilnya dibatasi pada nilai integer 0 dan 1.

Penelitian selanjutnya menggunakan algoritma dengan penggabungan yang disebut proses *cluster*. *Cluster* adalah proses data menjadi beberapa kelompok memiliki sifat serupa. Ini banyak digunakan di banyak bidang, termasuk data mining, statistik, biologi, dan pembelajaran mesin. Sebuah *cluster* memiliki objek dengan tinggi kesamaan, dan tidak mirip maka lain kelompok . Penelitian memaparkan teknik *cluster* bus baru menggunakan jaring saraf tiruan Kohonen untuk membentuk kelompok bus *cluster* untuk kestabilan tegangan listrik [26]. Dengan menggunakan metode *cluster* Kohonen untuk jaringan listrik yang memiliki karakteristik sintesis pada beban, dari hasil penelitian ini menunjukkan hasil yang signifikan [27]. Metode pendekatan untuk menentukan tingkat kerugian energi dengan menggunakan *cluster* K – Means dan hasilnya mampu mengurangi *losess* 57% [28]. Metode *cluster K-Means* juga di gunakan untuk mengembangkan pendekatan *Voltage Controlled Amplifier (VCA)* untuk manajemen daya reaktif dan kontrol tegangan dan hasilnya *K-Means* algoritma paling cocok untuk optimalisasi penempatan *PMU* karenanya efisiensi dalam memproses kumpulan data yang besar. *K-Means* telah diimplementasikan pada dataset yang sama dan hasilnya sangat bagus untuk data yang besar [29]. Penelitian berbasis Algoritma Genetika untuk masalah penempatan *PMU* dan hasilnya mampu untuk mengoptimasi secara optimal [30]. Penelitian menggunakan *Particle Swarm Optimization (PSO)* untuk aplikasi dengan monitoring menggunakan *Wireless Sensor Networks (WSN)* [31].

K-Means algoritma paling cocok untuk optimalisasi penempatan *PMU* sangat efisiensi dalam memproses kumpulan data yang besar. *K-Means* telah

diimplementasikan pada dataset yang sama dan hasilnya sangat kompleks untuk jarak data yang besar

Didasarkan dari hasil jumlah dan penentuan lokasi *PMU* yang *Integer Linear* dengan konsep *One Depth Unobservability* [34] dan hasilnya hanya satu bus yang tidak diawasi dan hasilnya kurang optimal. Selain itu juga di dasarkan pada penempatan *PMU* dengan *Integer Linear* dengan berdasarkan topologi jaringan radial dan hasilnya dapat ditentukan dengan mempertimbangkan bus yang terhubung, hasilnya dibatasi pada nilai integer 0 dan 1[35].

Berdasarkan pada referensi penelitian tersebut maka penelitian yang di kembangkan konsep baru yang menggabungkan metode Integer Linear dan metode *K-Means* dengan teknik *cluster*, digunakan untuk optimalisasi penempatan *PMU* di jaringan distribusi Bendul-Merisi Surabaya Indonesia. Kombinasi ini memiliki keunggulan yaitu dapat menghitung data acak non-integer dengan hasil yang lebih akurat.

Hasil *resume* dari pembahasan penelitian tentang penempatan *PMU* telah dilakukan oleh beberapa peneliti dengan berbagai metode optimasi dengan pertimbangan berbagai aspek pendekatan deterministik dan stokastik dengan metode matematik, heuristik dan *cluster*.

Literatur Review dapat di lihat pada Tabel 2.1.

Penempatan *PMU* adalah proses pencarian hasil terbaik di ruang solusi yang harus memenuhi semua kreteria. Selain itu dibutuhkan parameter sebagai tujuan optimasi menggunakan model deterministik, yaitu model matematika dapat di ukur dengan derajat kepastian yang tinggi dan tidak mengandung komponen yang sifatnya random (*probabilistik*) dan *output* dapat ditentukan bila sejumlah input dan hubungan tertentu dimasukkan. Sedangkan model stokastik menghasilkan *output* random, karenanya diuji hanya berupa estimasi (perkiraan) kebenaran karakteristiknya pada model.

Tabel 2.1 Hasil *Resume Literatur Revie*w

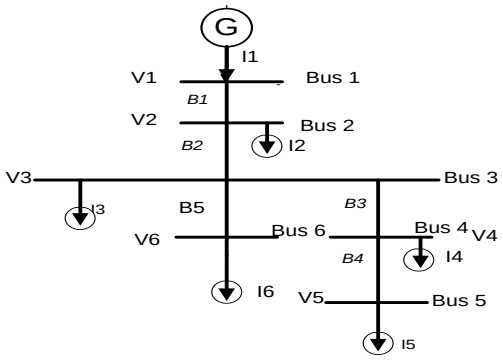
No. Ref.	Aspek teknikal		Metode Analisa			Metode optimasi
	Determi-nistik	Stokastik	Matematis	Heuristik	*Cluster*	
[4]	✓		✓			*Integer Linear Programming (ILP)*
[5]	✓		✓			*Binary Integer Programming (BIP)*
[6]		✓	✓			*weighted least square (WLS)*
[7] [8] [9]		✓	✓			*WLS Distribution System State Estimator Based on Voltage or Branch-currents*
[11]		✓	✓			*An optimal PMU placement technique for power system observability*
[12]		✓				*weighted least square*
[13]		✓				*Biner Imperialistik Competation Algoritm (BICA)*
[14]		✓	✓			*Random Component Outages (RCO)*
[15]	✓		✓			*Optimal PMU placement method for complete topological and numerical observability of power system*
[16]		✓	✓			*Optimal PMU placement by an equivalent linear formulation for exhaustive search*
[17]	✓		✓			Approximately Optimal Placement *PMU*(AOPP)
[18] [19]	✓			✓		*Binary Particle Swarm Optimization (BPSO)*
[20]	✓			✓		*Improved Tabu Search (ITS)*
[21]	✓			✓		*Multi-objective biogeography based optimization for optimal PMU placement*
[23] [24] [25]		✓		✓		Algorima Genetika
[26]		✓		✓		*Kohonen Neural Network Clustering For Voltage Control In Power Systems*
[27]	✓			✓	✓	*Dynamic Characteristics Clustering of Electric Loads Based onKohonen*
[28]	✓			✓	✓	*Using K-Means Clustering Method in Determination of the Energy Loses Levels fromElectricDistribution Systems*

Tabel 2.1. (Lanjutan)

No. Ref.	Aspek teknikal		Metode Analisa			Metode optimasi
	Determi-nistik	Stokastik	Matematis	Heuristik	*Cluster*	
[29]	✓			✓	✓	*Voltage Control Areas for Reactive Power Management using Clustering Approach in Deregulated Power System*
[30]	✓			✓	✓	*Genetic algorithms for optimal placement of phasor measurement units in electrical*

2.1 Metode *Network Topology*

Metode *Network topology* adalah suatu metode pembentukan matrik impedansi dengan alur saluran distribusi. Jalur saluran distribusi di representasikan dalam bentuk matrik yang menggambarkan injeksi dari satu bus ke bus lain. Sebagai contoh single line untuk analisis aliran daya berdasarkan *network topology* di tunjukkan pada Gambar 2.1



Gambar 2.1 *Single Line diagram contoh analisis menggunakan metode Network Topology*

Dari Gambar 2.1 maka didapatkan persamaan untuk mencari arus pada cabang tunjukkan persamaan 2.1 sebagai berikut:

$$B_5 = I_6$$
$$B_4 = I_5$$
$$B_3 = I_4 + I_5$$
$$B_2 = I_3 + I_4 + I_5 + I_6$$
$$B_1 = I_2 + I_3 + I_4 + I_5 + I_6 \quad (2.1)$$

Dimana : $B_1,\ B_2, B_3,\ B_4,\ B_5$ adalah dari arus cabang yang mengalir pada bus 1 sampai bus 6.

Selanjutnya untuk mencari arus yang mengalir melalui cabang di bus 1 sampai bus 6 di bentuk matrik yaitu di sebut *Bus Injection Branch Current (BIBC).* Matrik tersebut dapat di lihat pada persamaan 2.2.

$$\begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} I_2 \\ I_3 \\ I_4 \\ I_5 \\ I_6 \end{bmatrix} \quad (2.2)$$

Persamaan umum arus pada bus dapat disederhanakan dalam persamaan 2.3.

$$[B] = [BIBC][\mathrm{I}] \quad (2.3)$$

Dimana B adalah bus cabang $B_1,\ B_2, B_3,\ B_4,\ B_5$, sedangkan *BIBC* berupa matrik konektivitas antar bus dan arus cabang, dan $I_2,\ I_3, I_4,\ I_5,\ I_6$ adalah arus yang mengalir pada bus 1 sampai bus 6, sedangkan I adalah arus yang mengalir pada cabang 1 sampai 6.

Persamaan untuk mencari nilai drop tegangan berdasarkan V_2 jalur injeksi arus dirumuskan pada persamaan 2.4

$$\begin{aligned} V_2 &= V_1 - B_1.Z_{12} \\ V_3 &= V_1 - B_1.Z_{12} - B_2.Z_{23} \\ V_4 &= V_1 - B_1.Z_{12} - B_2.Z_{23} - B_3.Z_{34} \\ V_5 &= V_1 - B_1.Z_{12} - B_2.Z_{23} - B_3.Z_{34} - B_4.Z_{45} \\ V_6 &= V_1 - B_1.Z_{12} - B_2.Z_{23} - B_3.Z_{34} - B_4.Z_{45} - B_5.Z_{26} \end{aligned} \quad (2.4)$$

Persamaan 2.4 dapat di selesaikan menggunakan bentuk matrik dengan komponen matrik *BCBV (Branch Current To Branch Voltage).*

Hubungan tegangan bus dengan arus cabang dalam bentuk matrik dapat di tuliskan pada persamaan 2.5

$$\begin{bmatrix} V_1 - V_2 \\ V_1 - V_3 \\ V_1 - V_4 \\ V_1 - V_5 \\ V_1 - V_6 \end{bmatrix} = \begin{bmatrix} Z_{12} & 0 & 0 & 0 & 0 \\ Z_{12} & Z_{23} & 0 & 0 & 0 \\ Z_{12} & Z_{23} & Z_{34} & 0 & 0 \\ Z_{12} & Z_{23} & Z_{34} & Z_{45} & 0 \\ Z_{12} & Z_{23} & 0 & 0 & Z_{36} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \end{bmatrix} \quad (2.5)$$

Dimana : Z adalah impedansi

Z_{12} adalah impedansi pada arus cabang ke 1 dan tegangan cabang ke 2.

Z_{23} adalah impedasi pada arus cabang ke 2 dan tegangan cabang ke 3.

Z_{34} adalah impedansi pada arus cabang ke 3 dan tegangan cabang ke 4.

Z_{45} adalah impedansi pada arus cabang ke 4 dan tegangan cabang ke 5.

Z_{56} adalah impedansi pada arus cabang ke 5 dan tegangan cabang ke 6.

Persamaan umum tegangan bus dapat disederhanakan dalam persamaan 2.6.

$$[\Delta V] = [BCBV][B] \quad (2.6)$$

$[\Delta V]$ adalah perubahan pada tegangan.

2.2. Teknik *Cluster* pada Jaringan Distribusi Bendul Merisi

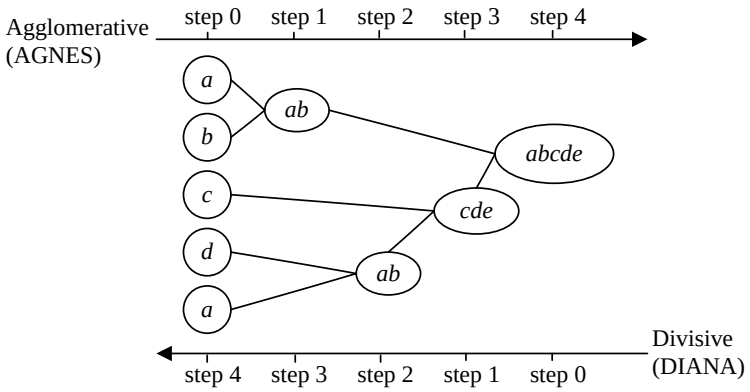
Metode *cluster K-Means* algoritma paling cocok untuk optimalisasi penempatan *PMU* karenanya efisiensi dalam memproses kumpulan data yang besar dan sangat bagus [29]. Metode *cluster* secara umum dapat dibagi menjadi dua yaitu, *cluster* Hirarki (*Hierarchical Cluster*) dan partisional (*P**artitional Cluster***) [57]**.**

2.2.1 *Cluster* Hirarki

Pada *cluster* Hirarki data dikelompokkan melalui suatu bagan yang berupa hirarki, dimana terdapat penggabungan dua grup yang terdekat disetiap iterasinya ataupun pembagian dari seluruh set data kedalam kelompok. Proses *cluster* hirarki ditunjukkan pada Gambar 2.2.

Langkah melakukan *cluster* Hirarki

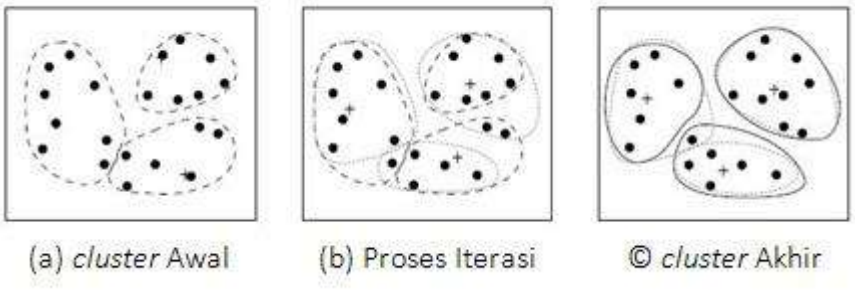
1. Identifikasi data dengan jarak terdekat.
2. Gabungkan data kedalam satu kelompok.
3. Hitung jarak antar kelompok.
4. Ulangi dari awal sampai semua terhubung.



Gambar 2.2 *Cluster* Hirarki [57]

2.2.2 *Cluster* **Partisional**

Cluster partisional yaitu data dikelompokkan ke dalam sejumlah kelompok tanpa adanya struktur hirarki antara satu dengan yang lainnya. Pada metode *cluster* Partisional setiap kelompok memiliki titik pusat *(centroid)* dan secara umum metode ini memiliki fungsi tujuan yaitu meminimumkan jarak ***(dissimilarity*)** dari seluruh data ke pusat masing-masing. Proses *cluster* ditunjukkan pada Gambar 2.3. Contoh metode pegelompokan *Partisional* antara lain **;** *K-Means, Fuzzy K-Means* dan *Mixture Modelling.*



Gambar 2.3 Proses *Cluster* Obyek Menggunakan metode *K-Means* [57]

Bagian dari metode yang di pergunakan pada penelitian ini adalah *K-Means*. *K-Means* merupakan metode pengelompokan yang paling sederhana dan umum. Hal ini dikarenakan *K-Means* mempunyai kemampuan mengelompokkan data dalam jumlah yang cukup besar dengan waktu komputasi yang cepat dan efisien. *K-Means*. merupakan salah satu algoritma *pengelompkan* dengan metode partisi (*partitioning method*) yang berbasis titik pusat (*centroid)* selain algoritma *K-Means*. yang berbasis obyek. Algoritma ini pertama kali diusulkan

oleh Mac Queen (1967) dan dikembangkan oleh Hartigan dan Wong tahun 1975 dengan tujuan untuk dapat membagi M data point dalam N dimensi kedalam sejumlah K dimana proses *pengelompkan* dilakukan dengan meminimalkan jarak *SEE* antara data dengan masing masing pusat *(centroid-based).* Algoritma *K-Means* dalam penerapannya memerlukan tiga parameter yang seluruhnya ditentukan pengguna yaitu jumlah inisialisasi *cluster*, dan jarak sistem, Biasanya, *K-Means* dijalankan secara independen dengan inisialisasi yang berbeda menghasilkan *cluster* akhir yang berbeda karena algoritma ini secara prinsip hanya mengelompokan data menuju lokal minimal. Salah satu cara untuk mengatasi lokal minimal adalah dengan mengimplementasikan algoritma *K-Means* untuk K yang diberikan, dengan beberapa nilai initial partisi yang berbeda dan selanjutnya dipilih partisi dengan kesalahan kuadrat terkecil *centroid* [57].

Algoritma *K-Means* mendefinisikan centroid atau pusat rata - rata dari *cluster*. Dalam penerapan algoritma *K-Means*, jika diberikan sekumpulan data X = $\{x_1, x_2, \ldots, x_n\}$ dimana xi = $(x_{i1}, x_{i2}, \ldots, x_{in})$ adalah sistem dalam ruang real RN, maka algoritma *K-Means* akan menyusun partisi X dalam sejumlah K. Setiap *cluster* memiliki titik tengah (*centroid*) yang merupakan nilai rata- rata dari data tersebut. Tahapan awal, algoritma *K-Means* adalah memilih secara acak K buah obyek sebagai *centroid* dalam data.

Kemudian, jarak antara obyek dihitung menggunakan jarak *Euclidian distance*. Algoritma *K-Means* secara *iterative* meningkatkan variasi nilai dalam dalam tiap- tiap kelompok. Obyek selanjutnya ditempatkan dalam kelompok yang terdekat, dihitung dari titik tengah *cluster*. Titik tengah baru ditentukan bila semua data telah ditempatkan dalam *cluster*. Proses penentuan titik tengah dan penempatan data diiulangi sampai nilai titik tengah dari semua kelompok yang terbentuk tidak berubah lagi [58-62].

2.3 *Neural Network Cluster K – Means*

K-Means merupakan algoritma untuk mengklasifikasikan atau kelompok objek berdasarkan atribut atau fitur ke sejumlah kelompok K, K merupakan bilangan bulat positif. *Cluster* dilakukan dengan meminimalkan jumlah kuadrat dari jarak antara data dan pusat *cluster* yang sesuai.

Langkah-langkah dari algoritma adalah sebagai berikut :

Langkah 1.Mentukan inisial K sebagai pusat $z_1{}^{(0)}, z_2{}^{(0)}, \ldots . z_k{}^{(0)}$

Langkah 2. Setiap iterasi distribusi input x mempunyai relasi terhadap K

$$x \in C_i{}^{(k)} \text{ if } d\left(x, Z_i{}^{(k)}\right) < d\left(x, Z_j{}^{(k)}\right) \quad (2.7)$$

$$i = 1,2, \ldots, K; \;\; i \neq j$$

Langkah 3. Menghitung*Cluster*baru untuk menentukan titik pusat

$$\min_{(E)} = \sum_{i=1}^{K} \sum_{x \in C_i} d \quad (x, Z_i) \quad (2.8)$$

Dimana zi adalah pusat *Cluster* Ci, d (x, zi) adalah jarak *Euclidean* antara titik x dan zi.

Dengan demikian, fungsi kriteria E untuk meminimalkan jarak dari setiap titik dari pusat *cluster* yang dimiliki titik. Lebih khusus, algoritma dimulai dengan menginisialisasi satu set pusat *cluster* K. Kemudian, akan mensimulasikan setiap objek dari dataset untuk *cluster* yang pusatnya adalah terdekat, dan menghitung ulang pusat. Proses berlanjut sampai pusat dari *cluster* berhenti berubah.

Langkah 3. Mengulangi langkah 2 dan 3 sampai konvergensi dicapai. Dalam algoritma ini bahwa *cluster* akhir akan tergantung pada saat memilih pusat *cluster* awal dan nilai-nilai dari K. Sebagai contoh, pada Gambar 2.7 dan 2.8 menggunakan dua set data untuk dua *cluster* dan tiga set data pada tiga *cluster*.

Untuk mendefinisikan jumlah *cluster* yang optimal dapat digunakan algoritma sebagai berikut: Penentuan *cluster* maksimum (K max). K max harus disimulasikan untuk memenuhi kondisi $2 \leq K\text{max} \leq n$, di mana n adalah benda berkerumun dari basis data. Untuk set objek dari basis data, K-berarti metode *cluster*dengan diberikan K ($2 \leq K \leq K$ max). Menurut struktur *cluster* yang diperoleh, kualitas partisi di evaluasi *determinate* nya.

Hasil penelitian ini dicapai melalui koefisien global yang siluet. Meningkatkan jumlah *cluster* ke K max untuk melihat apakah K berarti metode *cluster* menemukan *cluster* yang lebih baik dari data. (Untuk mengulang langkah 2 - 3). Tampilan jumlah *cluster* K optimal (K opt) yang telah memperoleh nilai optimal dari koefisien metode *Silhouette* global atau menggunakan metode *Elbow*.

Teknik validasi *silhouette* menghitung lebar siluet untuk setiap sampel, siluet lebar rata-rata untuk setiap *cluster* untuk total set data. Dan diterapkan untuk evaluasi validitas *cluster* digunakan untuk menentukan penentuan jumlah *cluster* yang optimal yang di tunjukkan pada persamaan 2.9 sampai dengan 2.10 .

$$SC = \frac{1}{K}\sum_{j=1}^{K} S_j \qquad \text{di mana ;} \qquad S_j = \frac{1}{n_j}\sum_{i=1}^{n_j} S_i \tag{2.9}$$

Sj = Koefisien lokal *Silhouette*

$$S_i = \frac{b_1 - a_i}{\max\{b_i . a_i\}} \tag{2.10}$$

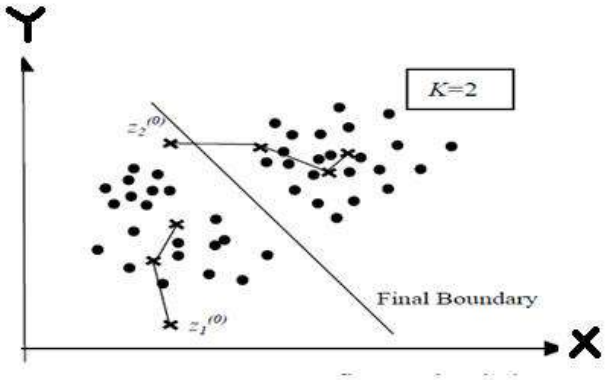
Si = indek lebar *Silhouette* untuk objek

Sedangkan metode *Elbow* merupakan suatu metode yang digunakan untuk menghasilkan informasi dalam menentukan jumlah *cluster* terbaik dengan cara melihat persentase hasil perbandingan antara jumlah *cluster* yang akan membentuk siku pada suatu titik. Untuk mendapatkan perbandingannya adalah dengan menghitung *SSE (Sum of Square Error)* dari masing-masing nilai *cluster*. Karena semakin besar jumlah kelopok K, maka nilai *SSE* akan semakin kecil. Rumus *SSE* pada *K-Means* di tunjukkan pada persamaan 2.11.

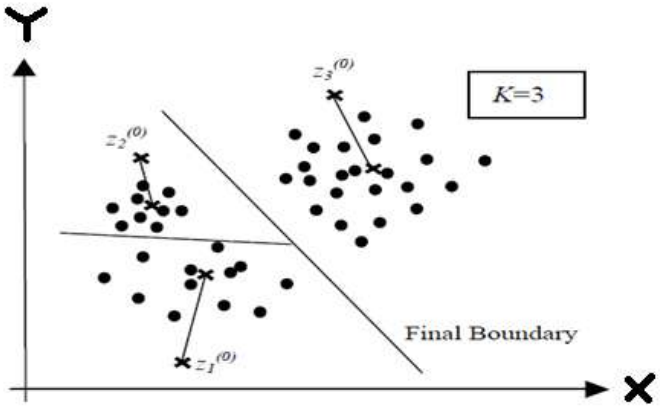
$$SSE = \sum_{k=1}^{k} \sum_{x_i < s_k} \left\| x_j - c_k \right\|_2^2 \tag{2.11}$$

SSE= Sum of Square Error) dari masing-masing nilai *cluster*

Sedangkan a adalah jarak antara objek i dan objek dari kelas yang sama j; bi adalah rata –rata jarak minimum antara objek i dan benda-benda di kelas paling dekat dengan kelas j. *Contoh K- Means* 2 kelompok dan tiga *cluster* ditunjukkan pada Gambar 2.4 dan 2.5.



Gambar 2.4 *K-Means* Dua *Cluster*



Gambar 2.5 *K–Means* Tiga *Cluster*

2.4 Teori Estimasi untuk Jaringan Distribusi Bendul Merisi

Metode estimasi untuk daya dan tegangan menggunakan hibrida antara *PSO* dan Genetika. Secara detail antara *PSO* dan *GA* akan di bahas pada sub bab 2.4.1 dan sub bab 2.4.2.

2.4.1 Teori *Particle Swarm Optimization (PSO)*

Particle swarm optimization merupakan salah satu cabang dari algoritma evolusi. *PSO* didasarkan pada perilaku sebuah kawanan burung atau ikan. Dimana sebuah kawanan tidak mempunyai pemimpin untuk menemukan makanan jadi mereka akan menyebar secara acak untuk menemukan letak makanan. Algoritma ini didasarkan dari perilaku sosial organisme ini. Perilaku sosial terdiri dari tindakan individu dan pengaruh dari individu-individu lainya.

Pada algoritma *PSO* ini, pencarian solusi dilakukan oleh suatu populasi yang terdiri dari beberapa partikel. Populasi dibangkitkan secara random dengan

batasan nilai terkecil dan nilai terbesar. Setiap partikel mempresentasikan posisi dan lokasi dari permasalahan yang dihadapi. Setiap partikel melakukan pencarian solusi yang optimal dengan kecerdasan individu itu dengan melintasi dimensi ruang pencarian search space D. Hal ini dilakukan dengan cara setiap partikel melakukan penyesuaian terhadap posisi terbaik dari pertikel tersebut (*local best*) dan penyesuaian posisi partikel terbaik dari nilai terbaik dari seluruh kawanan (*global best*) selama melintasi ruang pencarian. Pada setiap iterasi, setiap solusi yang direpresentasikan oleh posisi partikel, dievalusi kinerjanya dengan cara memasukkan solusi tersebut kedalam *fitness* function. Setiap partikel diperlalkukan seperti sebuah titik pada suatu dimensi ruang tertentu. Kemudian terdapat dua factor yang memberikan karakter terhadap status pertikel pada ruang pencarian yaitu posisi X dan kecepatan Y partikel disampaikan oleh Kennedy dan Eberhart (1995).

Berikut ini adalah persamaan yang menggambarkan posisi dan kecepatan pada persamaan 2.12 dan 2.13.

$$\mathrm{X_i(t) = x_{i1}(t), x_{i2}(t), x_{i3}(t), \dots, x_{iN}(t)} \quad (2.12)$$

$$\mathrm{V_i(t) = v_{i1}(t), v_{i2}(t), v_{i3}(t), \dots, v_{iN}(t)} \quad (2.13)$$

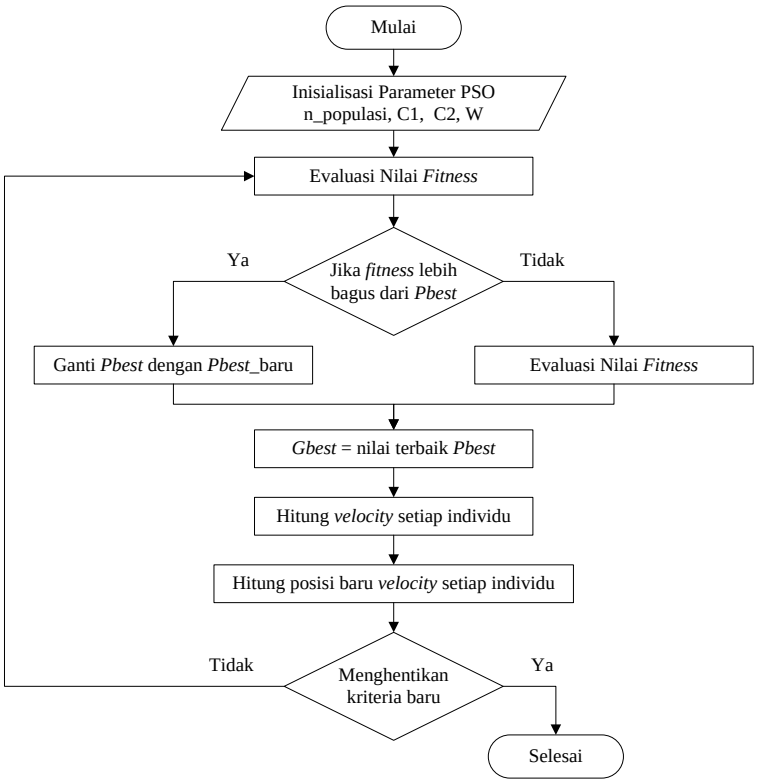
Dimana X adalah posisi partikel. V adalah kecepatan partikel. i dan t adalah index partikel dan iterasi ke-t, pada dimensi ruang N. Berikut ini model matematika yang menggambarkan mekanisme perbaikan status partikel dapat di lihat pada persamaan 2.14 dan 2.15.

$$V_i(t) = wV_i(t-1) + c_1 r_1 \left(X_i^L X_i(t-1)\right) + c_2 r_2 (X^G - X_i(t-1)) \quad (2.14)$$

$$X_i(t) = V_i(t) + X_i(t-1) \quad (2.15)$$

$X_i^L = X_{i1}^L, X_{i2}^L, \dots, X_{iN}^L$ mempresentasikan *local best* dari partikel ke-i. Sedangkan $X^G = X_1^G, X_2^G, \dots, X_N^G$ mempresentasikan *global best* dari seluruh kawanan. c_1 dan c_2 adalah suatu konstanta yang bernilai positif yang biasa disebut dengan *learning factor*. r_1 dan r_2 adalah suatu bilangan random positif bernilai antara 0 dan 1. *w* adalah parameter inersia Persamaan (2.14) digunakan untuk mendapatkan kecepatan partikel baru berdasarkan kecepatan sebelumnya, jarak antara posisi

saat ini dengan posisi terbaik partikel (*local best*), dan jarak saat ini dengan posisi terbaik kawanan (*global best*). Kemudian partikel terbang menuju posisi yang bari berdasarkan Persamaan 2.15. Alur kerja dari *PSO* dapat dilihat pada Gambar 2.6.



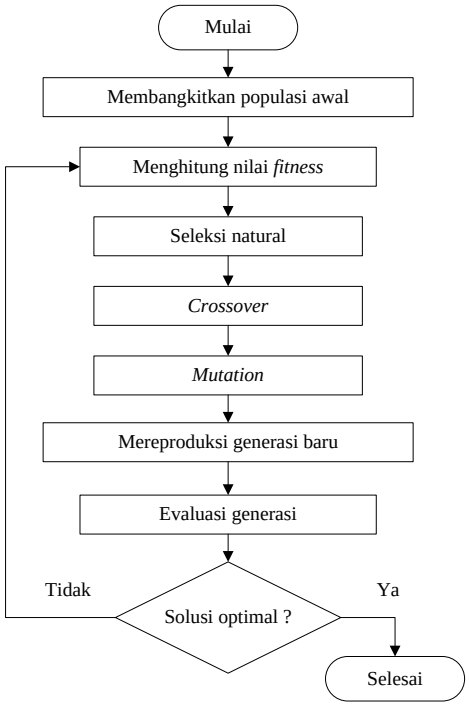
Gambar 2.6 Alur kerja Algoritma *PSO*

2.4.2 *Genetic Algorithm*

Genetic Algorithm (GA) adalah algoritma genetika masuk dalam kelompok algoritma evolusi. Algoritma ini diperkenalkan pertama kali oleh Holland pada tahun 1975 dan merupakan metode yang umum digunakan untuk metode pencarian dan terinspirasi oleh populasi genetika dalam mencari solusi suatu masalah. Algoritma ini juga mengikuti konsep *Carles Darwin* dengan teori evolusinya dimana individu yang kuat akan bertahan dari populasinya. Elemen-elemen dasar dari genetika alam adalah : seleksi alam *(natural selection)*, kawin silang *(crossover)*, dan pindah silang *(mutation*).

Seleksi alam adalah usaha untuk mempertahankan individu terbaik dengan cara mengadakan individu terbaik. Sehingga individu terbaik tidak hilang pada iterasi-iterasi selanjutnya. Operator kawin silang digunakan untuk membuat

individu baru. Untuk membuat individu baru dibutuhkan dua buah induk. Teknik pemilihan induk yang paling sering digunakan adalah *roulet wheelt*. Operator pindah silang digunakan untuk mengganti individu terjelek dengan individu baru. Banyak individu yang diganti bergantung pada parameter rasio pindah silang. Algoritma Genetika di tunjukkan pada Gambar 2.7.



Gambar 2.7 Alur kerja *Genetic Algoritm*

Konsep dasar Algoritma Genetika sebagai berikut **:**

Langkah pertama dilakukan menggunakan skema Pengkodean kromosom. Pada penelitian ini terdiri dua variabel. Variabel -1 merupakan bit 1 string kromosom. Apabila bit bernilai 1 maka dianggap adanya suatu keadaan bus adanya konektifitas jaringan , jika 0 menunjukkan tidak adanya konektifitas di dalamnya. Skema pengkodean biner dengan persamaan 2.16.

$$\text{Xi} = Rb + \frac{Ra-Rb}{2^{L}-1} \tag{2.16}$$

Selanjutnya di tentukan fungsi *fitness* adalah menentukan nilai *fitness* tertinggi dengan meminimalisasi nilai daya aktif (P) pada persamaan 2.17.

Fungsi objektif rugi daya aktif

$$\text{f}= aN_{pmu} - bN_h - cN_{pmu} \times N_h \tag{2.17}$$

P diatas di hitung terhadap analisa aliran daya dan Total P total = P daya aktif + daya Reaktif (Q).

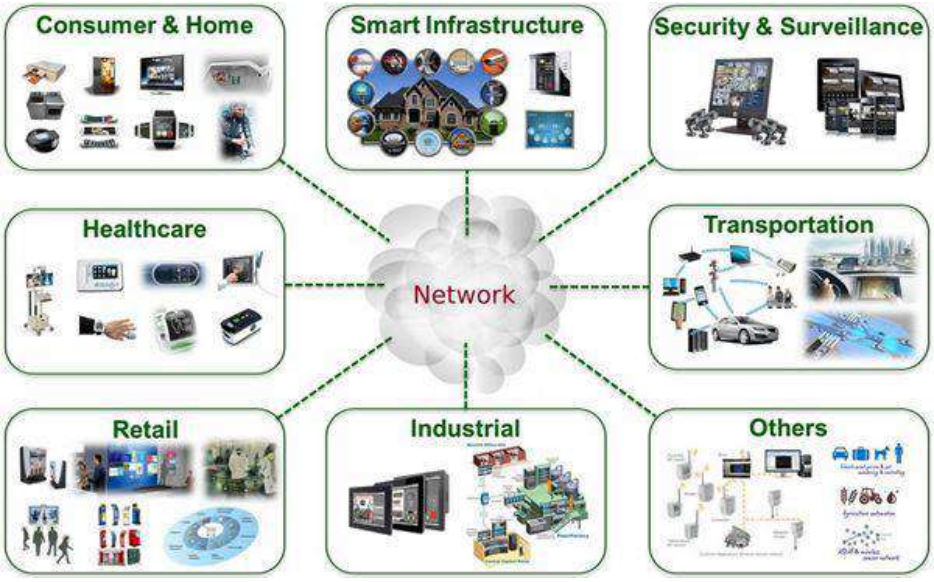
Kemudian dilakukan angking *fitness* yaitu dengan penskalaan sehingga nilai – nilai fitnesnya. Untuk itu dhitung nilai *roullette* yaitu nilai komulatif di dapat dari nilai *fitness* masing –masing kromosom dibagi total nilai *fitness* semua kromosom di tunjukkan 2.18.

$$P_{index} = \sum_{i=1}^{N} \frac{f(i)}{f\ populasi} \quad (2.18)$$

Proses berikutnya yaitu pindah silang (*crossover*) yaitu pindah silang satu titik potong diilih secara random range 0 -10. Pindah silang suatu operasi pindah silang dengan merubah gen yang bersesuaian dengan kebalikannya. Misalnya 1 menjadi 0 atau sebaliknya.

2.5. *Internet of Things (IoT)*

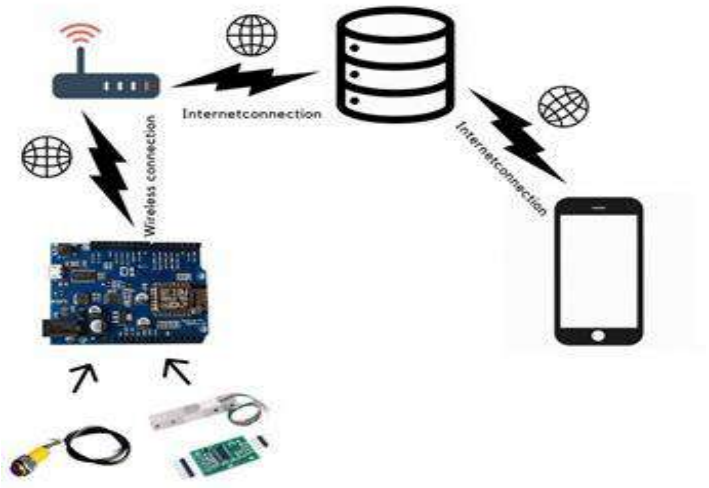
IoT telah berevolusi dalam cara luar biasa selama dekade terakhir dan masih tren yang muncul bagi para peneliti di kedua akademisi dan industri . Jaringan menjadi luas untuk dibuat komunikasi antara mesin dan mesin. Internet adalah jaringan yang menyediakan interaksi yang efisien di antara perangkat yang terletak di terminal jarak jauh. Dalam hal ini terlibat antara pertukaran informasi yang berguna di seluruh dunia dengan bantuan intervensi manusia. Hampir semua, setiap daerah, setiap perangkat, setiap sensor, setiap perangkat lunak terhubung satu sama lain. Ide *IoT* adalah dikembangkan secara paralel dengan *Wereless Sensor Networks (WSNs). WSN* menghasilkan peningkatan minat dari industri dan penelitian perspektif. Ini karena ketersediaan yang murah, komponen miniatur bertenaga rendah seperti prosesor, radio dan sensor yang sering diintegrasikan pada satu chip [63]. *WSN* memiliki banyak aplikasi dalam pemantauan cuaca, manajemen bencana, pelacakan inventaris, ruang pintar, pemantauan habitat, pelacakan target, pengawasan, dan banyak lagi [64]. Suatu peralatan yang mempunyai kemampuan untuk mengakses perangkat melalui *smartphone* atau melalui komputer disebut *IoT*. Perangkat ini diakses dari jarak jauh oleh manusia. Seperti yang di tunjukkan pada Gambar 2.8.



Gambar 2.8 Semua Aktivitas Manusia terhubung Internet [64]

IoT pada dasarnya adalah *platform* di mana kita menghubungkan hal-hal sehari-hari yang tertanam dengan elektronik, perangkat lunak, dan sensor ke internet memungkinkan mereka untuk mengumpulkan dan bertukar data. Teknologi ini memungkinkan mesin untuk melakukan tindakan seperti manusia Perangkat benda ini memiliki kemampuan untuk mengumpulkan dan mentransfer data melalui jaringan atau internet tanpa intervensi manual. *IoT* secara teknis membuat semuanya menjadi pintar, karena meningkatkan berbagai aspek kehidupan melalui penggunaan data, jaringan, dan algoritme itu. *IoT* membuat konsep komputasi di mana-mana dengan objek kehidupan kita sehari-hari seperti mobil, jalan, alat pacu jantung, kamera berbentuk di trek pencernaan kita, dilengkapi dengan sensor untuk berkomunikasi dengan manusia dan membantu mereka dalam setiap langkah [65]. Teknologi nirkabel seperti *Wi-Fi, Bluetooth, ZigBee, RFID, 6LoWPAN,* memungkinkan perangkat yang akan terhubung ke Internet dan masing-masing lain. Layanan cloud mengumpulkan, menyimpan, dan menganalisis data dikumpulkan oleh sensor dan memungkinkan orang untuk mengambil keputusan sesuai [66-69]. Sensor memainkan peran utama dalam mendefinisikan batas-batas *IoT* dengan mengonversinya dari pasif ke jaringan yang aktif. Data yang dikumpulkan bias dianalisis, divisualisasikan, dapat menghitung data baru, atau berinteraksi dengan situs web dan media sosial dengan bantuan perangkat lunak. Perangkat lunak ini membantu menghitung data baru,

dan memvisualisasikan data dalam bentuk plot, grafik, dan pengukur menggunakan alat analisis online. Informasi yang diproses adalah pada dasarnya disimpan dan dilihat oleh pengguna. Pengguna mengambil tindakan yang diperlukan dengan berkomunikasi dengan perangkat. Perangkat internet yang digunakan untuk komunikasi lebih banyak kelebihan kuat, lebih murah dan lebih kecil dari waktu ke waktu, skalabilitas, fleksibilitas, dan akurasi di tunjukkan 2.9.



Gambar 2.9 Blok diagram sistem [67]

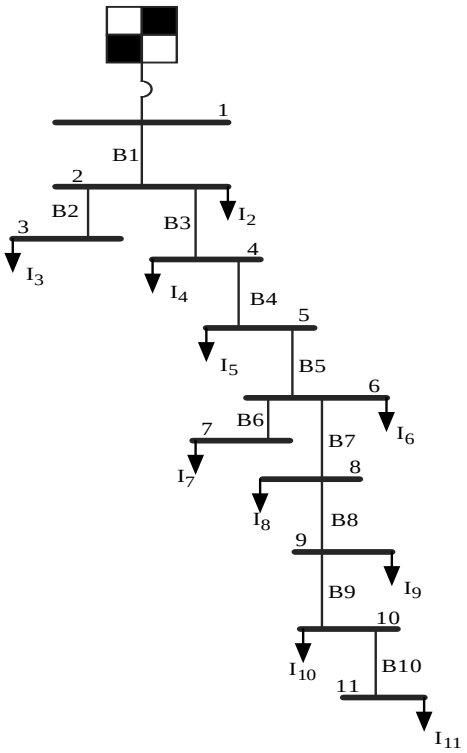
BAB 3 OPTIMASI PENEMPATAN *PMU* PADA JARINGAN DISTRIBUSI 20 kV SURABAYA SELATAN DAN TRANSMISI JAWA-BALI 500 kV

Jaringan distribusi 20 kV Surabaya Selatan di Bendul Merisi dan transmisi Jawa-Bali 500 kV yang di gunakan untuk pengujian data. Parameter input menggunakan matrik berdasarkan tegangan bus dapat menggunakan metode *Integer Linear* pada referensi no 9. Dalam metode ini, data yang digunakan hanya nilai 0 dan 1 untuk mengoptimalkan penempatan *PMU* di jaringan. Data yang di gunakan adalah data tegangan, arus, dan impedansi yang nilainya tidak bilangan bulat yang di tunjukkan Tabel 3.1.

Tabel 3.1 Data Saluran Penyulang Bendul Merisi

No. Bus	**Impedansi**	**Panjang Kabel (m)**	**Massa Jenis Kabel (gr/cm$^{3)}$**	**Jenis Kabel**	**Luas Penampang (mm^2)**	**Arus (A)**
Bus 2	4843+153i	310	2.7	A3C	3 x 150	21,4
Bus 3	208+0.04i	219	2.7	A3C	3 x 150	6
Bus 4	3821+122i	766	2.7	A3C	3 x 150	14,2
Bus 5	3214+73i	386	2.7	A3C	3 x 150	13,8
Bus 6	2689+55i	940	2.7	A3C	3 x 150	9,1
Bus 7	697+4i	1557	2.7	A3C	3 x 150	19,7
Bus 8	1599+18i	1095	2.7	A3C	3 x 150	8,3
Bus 9	1278+4i	192	2.7	A3C	3 x 150	13
Bus 10	819+3i	546	2.7	A3C	3 x 150	10
Bus 11	365+0.76i	1065	2.7	A3C	3 x 150	10,4

Dari data pada Tabel 3.1 ini, dengan metode *K-Means* digunakan untuk mengelompokkan dan mengoptimalkan penempatan *PMU*. Sedangkan *single line* pada jaringan distribusi Bendul Merisi ditunjukkan pada Gambar 3.1.



Gambar 3.1 *Single-Line* Jaringan Distribusi di Bendul Merisi Surabaya

Menggabungkan metode *Integer Liner* dengan *metode K-Means* untuk mengelompokkan dan mengoptimalkan penempatan *PMU* adalah inti dari penelitian ini. Metode *Integer Liner* digunakan untuk mendapatkan nilai drop tegangan pada setiap bus. Nilai dari drop tegangan yang diperoleh, data dari pengukuran arus dan impedansi digunakan untuk proses *pengelompkan* dan mengoptimalkan penempatan *PMU* pada jaringan distribusi Bendul Merisi.

Dalam single line distribusi di Bendul Merisi terdiri dari 11 bus dan dipasang 11 *PMU*. Selanjutnya, di buat matrik. Setiap set data dibangun oleh dimensi biner, yang menunjukkan ada *PMU* pada bus dengan simbol 1, jika tidak 0. Untuk mengimputkan data berdasarkan batasan metode Integer Linear ditunjukkan dalam persamaan 3.1.

$$A_{kxm} = \begin{cases} 1 \text{ if } k = m \\ 1 \text{ if } k \text{ and } m \text{ connected} \\ 0 \text{ if not connected} \end{cases}$$

$$A_{kxm} = \sum_{j=1}^{m} X_j \geq M : \text{dimana } m \in \{0,1\}, j \in \{0,1\} \quad (3.1)$$

Dari Gambar 3.1 dengan menerapkan Hukum *Kirchoff,* arus injeksi (I) dapat dirumuskan terhadap saluran (B) maka diperoleh persamaan berikut 3.2.

$$
\begin{aligned}
B_1 &= I_2 + I_3 + I_4 + I_5 + I_6 + I_7 + I_8 + I_9 + I_{10} + I_{11} \\
B_2 &= I_3 \\
B_3 &= I_4 + I_5 + I_6 + I_7 + I_8 + I_9 + I_{10} + I_{11} \\
B_4 &= I_4 + I_5 + I_6 + I_7 + I_8 + I_9 + I_{10} + I_{11} \\
B_5 &= I_5 + I_6 + I_7 + I_8 + I_9 + I_{10} + I_{11} \\
B_6 &= I_7 \\
B_7 &= I_8 + I_9 + I_{10} + I_{11} \\
B_8 &= I_9 + I_{10} + I_{11} \\
B_9 &= I_{10} + I_{11} \\
B_{10} &= I_{11}
\end{aligned}
\tag{3.2}
$$

Selanjutnya, dari persamaan 3.2 dirumuskan dalam bentuk matriks *Branch Injection Branch Current (BIBC)* dalam bentuk matriks Bendul-Merisi yang mengandung 11 *PMU* koneksi antar-bus (11x11) seperti pada persamaan 3.3.

$$
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\tag{3.3}
$$

Angka '1' dalam matriks BIBC menyatakan hubungan antara arus dan saluran dalam sistem distribusi antar bus, jika tidak, angka 0 menunjukkan tidak ada hubungan antara arus dan saluran. Selanjutnya diselesaikan dengan persamaan 3.4 dan 3.5.

$$[B] = [BIBC]x[I] \tag{3.4}$$

Berdasarkan 3.2 maka di masukkan nilai 1 untuk bus yang terpasang *PMU* dan 0 bus yang tidak terpasang *PMU* ke dalam matrik tanpa **bus 1(bus utama)** sehingga matrik seperti pada persamaan 3.5.

$$\begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \\ B_6 \\ B_7 \\ B_8 \\ B_9 \\ B_{10} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} x \begin{bmatrix} I_2 \\ I_3 \\ I_4 \\ I_5 \\ I_6 \\ I_7 \\ I_8 \\ I_9 \\ I_{10} \\ I_{11} \end{bmatrix} \quad (3.5)$$

Sedangkan persamaan 3.6 dengan memasukkan nilai arus ke dalam matrik

$$\begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \\ B_6 \\ B_7 \\ B_8 \\ B_9 \\ B_{10} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 21.4 \\ 6 \\ 14.2 \\ 13.8 \\ 9.1 \\ 19.7 \\ 8.3 \\ 13 \\ 10 \\ 10.4 \end{bmatrix} \quad (3.6)$$

Uraian dari persamaan 3.6 dapat di tunjukkan pada persamaan 3.7

$$\begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \\ B_6 \\ B_7 \\ B_8 \\ B_9 \\ B_{10} \end{bmatrix} = \begin{bmatrix} (1*21.4)+(1*6)+(1*14.2)+(1*13.8)+(1*9.1)+(1*19.7)+(1*8.3)+(1*13)+(1*10)+(1*10.4) \\ (0*21.4)+(1*6)+(0*14.2)+(0*13.8)+(0*9.1)+(0*19.7)+(0*8.3)+(0*13)+(0*10)+(0*10.4) \\ (0*21.4)+(0*6)+(1*14.2)+(1*13.8)+(1*9.1)+(1*19.7)+(1*8.3)+(1*13)+(1*10)+(1*10.4) \\ (0*21.4)+(0*6)+(0*14.2)+(1*13.8)+(1*9.1)+(1*19.7)+(1*8.3)+(1*13)+(1*10)+(1*10.4) \\ (0*21.4)+(0*6)+(0*14.2)+(0*13.8)+(1*9.1)+(1*19.7)+(1*8.3)+(1*13)+(1*10)+(1*10.4) \\ (0*21.4)+(0*6)+(0*14.2)+(0*13.8)+(0*9.1)+(1*19.7)+(0*8.3)+(0*13)+(0*10)+(1*10.4) \\ (0*21.4)+(0*6)+(0*14.2)+(0*13.8)+(0*9.1)+(0*19.7)+(1*8.3)+(1*13)+(1*10)+(1*10.4) \\ (0*21.4)+(0*6)+(0*14.2)+(0*13.8)+(0*9.1)+(0*19.7)+(0*8.3)+(1*13)+(1*10)+(1*10.4) \\ (0*21.4)+(0*6)+(0*14.2)+(0*13.8)+(0*9.1)+(0*19.7)+(0*8.3)+(0*13)+(1*10)+(1*10.4) \\ (0*21.4)+(0*6)+(0*14.2)+(0*13.8)+(0*9.1)+(0*19.7)+(0*8.3)+(0*13)+(0*10)+(1*10.4) \end{bmatrix} \quad (3.7)$$

Besarnya hasil bus cabang atau saluran bus [B] di tunjukkan persamaan 3.8.

$$\begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \\ B_6 \\ B_7 \\ B_8 \\ B_9 \\ B_{10} \end{bmatrix} = \begin{bmatrix} 125.9 \\ 6 \\ 98.5 \\ 84.3 \\ 70.5 \\ 19.7 \\ 41.7 \\ 33.4 \\ 20.4 \\ 10.4 \end{bmatrix} \quad (3.8)$$

Jaringan topologi yang radial di jaringan distribusi di Bendul Merisi, digunakan data arus seperti pada persamaan 3.6 sampai dengan 3.8 , sebagai data penelitian untuk menentukan saluran (B). Pada jaringan tersebut sebelum di lakukan

pengelompkan maka dihitung penurunan (*drop*) tegangan pada masing - masing bus. Perhitungan besarnya drop tegangan di tentukan oleh matriks dalam persamaan 3.9.

$$\begin{bmatrix} V_1 - V_2 \\ V_1 - V_3 \\ V_1 - V_4 \\ V_1 - V_5 \\ V_1 - V_6 \\ V_1 - V_7 \\ V_1 - V_8 \\ V_1 - V_9 \\ V_1 - V_{10} \\ V_1 - V_{11} \end{bmatrix} = \begin{bmatrix} Z_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ Z_{12} & Z_{23} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ Z_{12} & 0 & Z_{24} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ Z_{12} & 0 & Z_{24} & Z_{45} & 0 & 0 & 0 & 0 & 0 & 0 \\ Z_{12} & 0 & Z_{24} & Z_{45} & Z_{56} & 0 & 0 & 0 & 0 & 0 \\ Z_{12} & 0 & Z_{24} & Z_{45} & Z_{56} & Z_{67} & 0 & 0 & 0 & 0 \\ Z_{12} & 0 & Z_{24} & Z_{45} & Z_{56} & 0 & Z_{68} & 0 & 0 & 0 \\ Z_{12} & 0 & Z_{24} & Z_{45} & Z_{56} & 0 & Z_{68} & Z_{89} & 0 & 0 \\ Z_{12} & 0 & Z_{24} & Z_{45} & Z_{56} & 0 & Z_{68} & Z_{89} & Z_{910} & 0 \\ Z_{12} & 0 & Z_{24} & Z_{45} & Z_{56} & 0 & Z_{68} & Z_{89} & Z_{910} & Z_{1011} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \\ B_6 \\ B_7 \\ B_8 \\ B_9 \\ B_{10} \end{bmatrix} \quad (3.9)$$

Dengan memberikan nilai impedansi saluran yang sesuai dengan persamaan 3.9 dan juga memasukkan nilai dari matrik saluran [B], hasilnya adalah pada persamaan 3.8 sampai dengan 3.9. Selanjutnya perubahan tegangan yang dapat ditemukan menggunakan rumus *Branch Current Branch Voltage (BCBV)* seperti persamaan 3.10.

$$[\Delta V] = [BCBV] x [B] \quad (3.10)$$

Semua perubahan tegangan set data ini dengan di ketahui nilai saluran [B] arus dapat di lihat pada persamaan 3.11.

$$\begin{bmatrix} V_1 - V_2 \\ V_1 - V_3 \\ V_1 - V_4 \\ V_1 - V_5 \\ V_1 - V_6 \\ V_1 - V_7 \\ V_1 - V_8 \\ V_1 - V_9 \\ V_1 - V_{10} \\ V_1 - V_{11} \end{bmatrix} = \begin{bmatrix} Z_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ Z_{12} & Z_{23} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ Z_{12} & 0 & Z_{24} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ Z_{12} & 0 & Z_{24} & Z_{45} & 0 & 0 & 0 & 0 & 0 & 0 \\ Z_{12} & 0 & Z_{24} & Z_{45} & Z_{56} & 0 & 0 & 0 & 0 & 0 \\ Z_{12} & 0 & Z_{24} & Z_{45} & Z_{56} & Z_{67} & 0 & 0 & 0 & 0 \\ Z_{12} & 0 & Z_{24} & Z_{45} & Z_{56} & 0 & Z_{68} & 0 & 0 & 0 \\ Z_{12} & 0 & Z_{24} & Z_{45} & Z_{56} & 0 & Z_{68} & Z_{89} & 0 & 0 \\ Z_{12} & 0 & Z_{24} & Z_{45} & Z_{56} & 0 & Z_{68} & Z_{89} & Z_{910} & 0 \\ Z_{12} & 0 & Z_{24} & Z_{45} & Z_{56} & 0 & Z_{68} & Z_{89} & Z_{910} & Z_{1011} \end{bmatrix} \begin{bmatrix} 125.9 \\ 6 \\ 98.5 \\ 84.3 \\ 70.5 \\ 19.7 \\ 41.7 \\ 33.4 \\ 20.4 \\ 10.4 \end{bmatrix} \quad (3.11)$$

Sedangkan persamaam 3.11 adalah hasil dari perhitungan dari persamaan 3.12.

$$\begin{bmatrix} V_1 - V_2 \\ V_1 - V_3 \\ V_1 - V_4 \\ V_1 - V_5 \\ V_1 - V_6 \\ V_1 - V_7 \\ V_1 - V_8 \\ V_1 - V_9 \\ V_1 - V_{10} \\ V_1 - V_{11} \end{bmatrix} = \begin{bmatrix} 9.897 + 3.6901\text{i} \\ 10.23 + 3.8144\text{i} \\ 29.03 + 10.825\text{i} \\ 37.281 + 13.902\text{i} \\ 54.085 + 20.169\text{i} \\ 61.902 + 23.07\text{i} \\ 65.661 + 24.487\text{i} \\ 67.287 + 25.094\text{i} \\ 70.112 + 26.147\text{i} \\ 72.92 + 27.194\text{i} \end{bmatrix} \quad (3.12)$$

Dengan memasukkan nilai tegangan sumber maka di dapatkan hasil seperti pada persamaan 3.13.

$$\begin{bmatrix} V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \\ V_7 \\ V_8 \\ V_9 \\ V_{10} \\ V_{11} \end{bmatrix} = \begin{bmatrix} 19958 - 3.6901\text{i} \\ 19958 - 3.8144\text{i} \\ 19939 - 10.825\text{i} \\ 19931 - 13.902\text{i} \\ 19914 - 20.169\text{i} \\ 19906 - 23.07\text{i} \\ 19902 - 24.487\text{i} \\ 19901 - 25.094\text{i} \\ 19898 - 26.147\text{i} \\ 19895 - 27.194\text{i} \end{bmatrix} \quad (3.13)$$

Hasil dari persamaan 3.13 di peroleh drop tegangan di tunjukkan pada Tabel 3.2.

Tabel 3.2 Tabel data drop tegangan

Bus	Beban (kVA)	Tegangan (Volt)	Drop tegangan (Volt)
2	745	19913	55
3	208	19910	3
4	496	19869	44
5	483	19832	37
6	319	19801	31
7	689	19793	8
8	290	19783	18
9	455	19768	15
10	350	19760	8
11	363	19755	5

3.1 Metode *Cluster PMU* Menggunakan Algoritma *Integer Linear K-Means*

Pengelompokan (*cluster)* adalah metode yang menggunakan beberapa data yang memiliki karakteristik yang sama. Menurut penelitian referensi14 sampai dengan 24, adalah proses menggabungkan data ke dalam beberapa kelompok sehingga, data dalam satu *cluster* memiliki tingkat kemiripan maksimum. Ada berbagai metode untuk mengukur nilai kesamaan antara objek. Teknik *cluster* yang digunakan adalah metode *Integer Linear K-Means*. Parameter yang diujikan berupa data tegangan, arus, dan impedansi. Untuk metode ini dengan menghitung jarak, yang disebut dengan

jarak *Euclidean*. Perhitungan dengan tiga (3) set data: tegangan (Vx), arus (Ix), impedansi (Zx) ditunjukkan dalam persamaan (3.14).

$$V_x \in \{V_1, V_2, ..., V_n\} \; ; I_x \in \{I_1, I_2, ..., I_n\} \; ; Z_x \in \{Z_1, Z_2, ..., Z_n\} \tag{3.14}$$

Data dalam persamaan 3.14 untuk dikelompokkan menjadi beberapa *cluster*, maka anggota dari setiap *cluster* ditentukan secara acak. Setiap *cluster* akan memiliki data yang di rata-rata, dihasilkan dari anggota. Rata-rata ini disebut titik pusat (*centroid)*. Untuk titik pusat tegangan (V_{cx}), arus (I_{cx}) dan impedansi (Z_{cx}). Pengukuran jarak dengan metode *Euclidean* dapat ditulis untuk setiap *cluster* dengan persamaan 3.15.

$$D_x = \sqrt{(V_x - V_{cx})^2 + (I_x - I_{cx})^2 + (Z_x - Z_{cx})^2} \tag{3.15}$$

Dari persamaan 3.15, setiap data tunggal menghasilkan jarak tunggal (D_x). Langkah selanjutnya adalah mengevaluasi D_x minimum yang menunjukkan anggota dari *cluster* yang sesuai. Ketika D_x minimum ditambahkan bersama-sama, itu menghasilkan *Sum Square Error (SSE)*. Rumusnya ditunjukkan dalam persamaan 3.16.

$$SSE = \sum_{i=1}^{k} \sum_{p \in C_i} D(p, m_i) \tag{3.16}$$

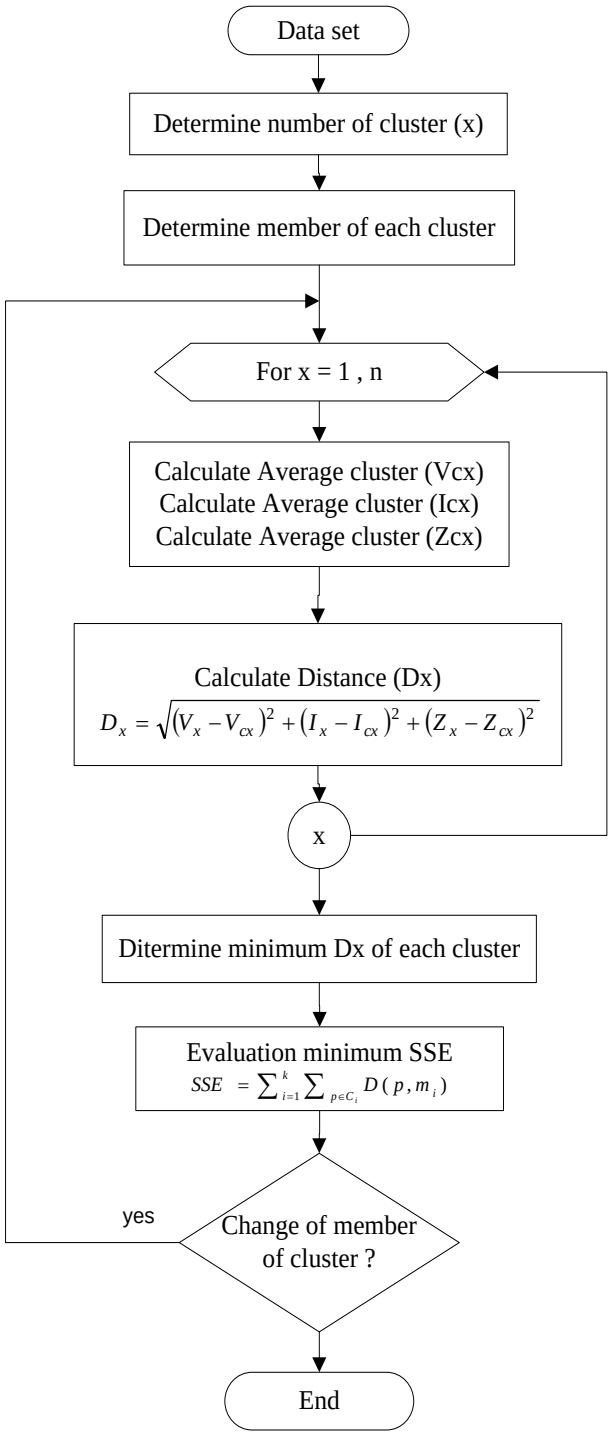
dimana: $p \in C_i$ = setiap titik data dalam *cluster i* ,

m_i = centroid dari *cluster i* ,

D = jarak varians terdekat di setiap *cluster i*.

Nilai *SSE* tergantung pada jumlah *cluster*. Semakin kecil nilai *SSE*, semakin baik hasil *cluster* dihasilkan.

Langkah-langkah keseluruhan dari algoritma ini diilustrasikan dalam Gambar 3.2.



Gambar 3.2 Alur kerja Algoritma *Integer Linear K-Means* [32]

3.2 Analisa dengan Metode *Integer Linear K-Means Clustering* Pada Jaringan Distribusi 20 kV di Bendul Merisi

Jaringan distribusi di Bendul Merisi digunakan sebagai obyek studi kasus menggunakan metode *Integer Linear K-Means Clustering* . Dalam penelitian ini analisis dilakukan dalam dua tahap. Tahap pertama dilakukan pengujian jaringan dengan menggunakan parameter tegangan dan arus. Tahap kedua dilakukan hal yang sama dengan menambahkan parameter impedansi untuk menentukan efek impedansi dalam penempatan *PMU*. Pada Tabel 3.1 hasil nilai drop tegangan dari masing - masing bus.Sedangkan Tabel 3.2 menunjukkan tegangan dan arus yang digunakan dalam pengujian tahap pertama. Data ini diambil dari jaringan distribusi di Bendul Merisi Surabaya. Langkah – langkah proses pengelompokan *Integer Linear K-Means* di tunjukkan pada Gambar 3.2. Proses di awali dengan menentukan batasan seperti pada matrik koneksi bus biner bendul Merisi ditunjukkan pada Gambar 3.3.

$$Bendul\ Merisi(\text{kxm}) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Gambar 3.3 Matrik koneksi bus biner Bendul Merisi

f_2 adalah bus ke 1 $= X_1 + X_2 + X_3 + X_4 + X_5 + X_6 + X_7 + X_8 + X_9 + X_{10} + X_{11} \geq 1$

f_3 adalah bus ke 2 $= X_3 \geq 1$

$f_4$adalah bus ke 3 $= X_4 + X_5 + X_6 + X_7 + X_8 + X_9 + X_{10} + X_{11} \geq 1$

f_5 adalah bus ke 4 $= X_5 + X_6 + X_7 + X_8 + X_9 + X_{10} + X_{11} \geq 1$

f_6 adalah bus ke 5 $= X_5 + X_6 + X_7 + X_8 + X_9 + X_{10} + X_{11} \geq 1$

f_7 adalah bus ke 6 $= X_7 \geq 1$

$f_8$adalah bus ke 7 $= X_8 + X_9 + X_{10} + X_{11} \geq 1$

f_9 adalah bus ke 8 $= X_9 + X_{10} + X_{11} \geq 1$

f_{10} adalah bus ke 9 $= X_{10} + X_{11} \geq 1$

f_{11} adalah bus ke 10 $= X_{11} \geq 1$

Selanjutnya di tentukan persamaan matematik untuk *Integer Linear* ditunjukkan pada persamaan 3.17.

$$Y = ax + bx$$
$$Y = (a_i + b_i)x_i \quad (3.17)$$
$$a = V \times I \times \cos\theta$$
$$b = V \times I \times \sin\theta$$

dimana;

a = daya aktif

b = daya reaktif

θ = faktor daya

Y = total daya

X_i : Matrik hubungan antar bus 1 sampai bus 11 dan cabang 1 sampai dengan 11.

Pengujian 2 parameter: Tegangan dan Arus

Tabel 3.3 menunjukkan data tegangan dan arus pada jaringan distribusi Bendul Merisi.

Tabel 3.3 Data tegangan dan arus beban Bendul Merisi

No. Bus	Tegangan (Volt)	Arus *load* (Ampere)
2	19913	22,2
3	19910	6,2
4	19862	15,6
5	19818	15,6
6	19781	10,9
7	19771	24,0
8	19768	10,3
9	19739	16,2
10	19727	16,2
11	19721	13,2

Pengujian diawali dengan mengacak data pada Tabel 3.3 sehingga didapat susunan data pada Tabel 3.4. Pada Tabel 3.4, ditetapkan pusat *cluster* 1 adalah pada bus 6, sedangkan pusat *cluster* 2 pada bus 7 dan pusat *cluster* 3 ditentukan pada bus 3.

Tabel 3.4 Penetapan pusat *Cluster* awal secara acak

No. Bus	Tegangan (Volt)	Arus (Ampere)	Pusat *Cluster*
6	19781	22,2	1 (Pusat *Cluster 1*)
9	19739	6,2	
2	19913	15,6	
5	19818	15,6	
7	19771	10,9	2 (Pusat *Cluster*2)
11	19721	24,0	
4	19862	10,3	
3	19910	16,2	3 (Pusat*Cluster*3)
8	19768	16,2	
10	19727	13,2	

Dari Tabel 3.4 didefinisikan bahwa pusat *cluster*1 adalah V_{c1}=19781 dan I_{c1}=22,2; pusat *cluster* 2 adalah V_{c2}=19771 dan I_{c2}=10,9; dan pusat *cluster* 3 adalah V_{c3}=19910 dan I_{c3}=16,2. Kemudian dilakukan **iterasi pertama** untuk proses *Integer Linear K-Means Clustering* dengan melakukan perhitungan jarak dengan menggunakan metode *Euclidean* yang ditunjukkan pada persamaan 3.15. Tabel 3.5 menunjukkan hasil perhitungan jarak bus n ke pusat masing-masing *cluster* dan Tabel 3.6 menunjukkan gabungan hasil perhitungan jarak dari bus n ke masing-masing pusat *cluster*.

Tabel 3.5 Jarak bus n ke pusat *Cluster*

Pusat *Cluster* 1: bus 6		Pusat *Cluster* 2:bus 7		Pusat *Cluster* 3: bus 3	
Bus n ke pusat	Jarak (D_x)	Bus n ke pusat	Jarak (D_x)	Bus n ke pusat	Jarak (D_x)
6 ke 6	0	6 ke 7	16,48	6 ke 3	129,09
9 ke 6	42,33	9 ke 7	32,94	9 ke 3	171,29
2 ke 6	132,49	2 ke 7	142,01	2 ke 3	16,28
5 ke 6	37,30	5 ke 7	47,75	5 ke 3	92,48
7 ke 6	16,48	7 ke 7	0	7 ke 3	140,14
11 ke 6	60,04	11 ke 7	51,15	11 ke 3	189,13
4 ke 6	81,14	4 ke 7	91,39	4 ke 3	48,91
3 ke 6	129,09	3 ke 7	140,14	3 ke 3	0
8 ke 6	13,01	8 ke 7	14,03	8 ke 3	142,06
10 ke 6	54,26	10 ke 7	44,69	10 ke 3	183,27

Tabel 3.6 Gabungan jarak bus ke pusat masing-masing *Cluster*

Bus n ke pusat	Jarak (D_x)		
	Cluster 1 Pusat: bus 6	*Cluster* 2 Pusat: bus 7	*Cluster* 3 Pusat: bus 3
6 ke 6	0	16,48	129,09
9 ke 6	42,33	32,94	171,29
2 ke 6	132,49	142,01	16,28
5 ke 6	37,30	47,75	92,48
7 ke 6	16,48	0	140,14
11 ke 6	60,04	51,15	189,13
4 ke 6	81,14	91,39	48,91
3 ke 6	129,09	140,14	0
8 ke 6	13,01	14,03	142,06
10 ke 6	54,26	44,69	183,27

Langkah berikutnya menentukan anggota masing-masing *cluster* yaitu dengan cara memilih jarak terpendek antara bus n ke masing-masing pusat *cluster* dari 3 *cluster*, yang ditunjukkan pada Tabel 3.7. Dari iterasi yang pertama maka anggota masing-masing *cluster* telah ditentukan, ditunjukkan pada Tabel 3.8.

Tabel 3.7 Penentuan anggota *Cluster*

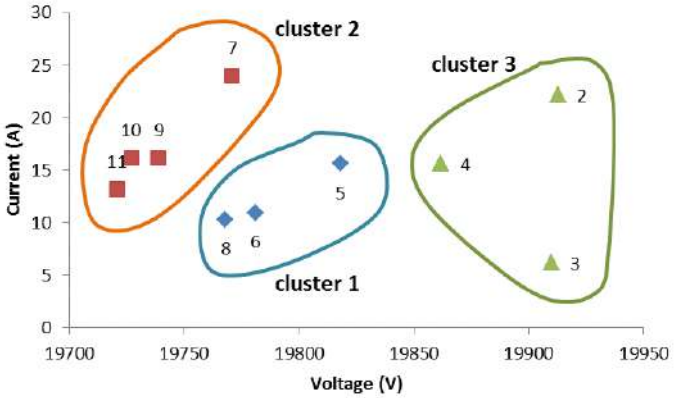
No. Bus	Jarak (Dx) *Cluster*1	Jarak (Dx) *Cluster*2	Jarak (Dx) *Cluster*3	Jarak (Dx) Minimum	Anggota *Cluster*
6	0	16,48	129,09	0	1
9	42,33	32,94	171,29	32,94	2
2	132,49	142,01	16,28	16,28	3
5	37,30	47,75	92,48	37,30	1
7	16,48	0	140,14	0	2
11	60,04	51,15	189,13	51,15	2
4	81,14	91,39	48,91	48,91	3
3	129,09	140,14	0	0	3
8	13,01	14,03	142,06	13,01	1
10	54,26	44,69	183,27	44,69	2
			SSE	244,28	

Pada Tabel 3.8 menunjukkan anggota untuk *cluster* 1 adalah bus 6, 5 dan 8; anggota untuk cluster 2 adalah bus 9, 7, 11, dan 10. Sedangkan anggota untuk *cluster* 3 adalah bus 2, 4 dan 3. Kemudian data tegangn dan arus pada Tabel 3.8 divisualisasikan menggunakan grafik dua dimensi dari X dan Y. Sumbu

horizontal X mewakili nilai tegangan dan sumbu vertikal Y mewakili arus. Hasil visualisasi dari Tabel 3.8 ditunjukkan pada Gambar 3.4.

Tabel 3.8 Hasil penentuan anggota *Cluster* (iterasi pertama)

No. Bus	Tegangan (Volt)	Arus (Ampere)	Anggota *Cluster*
6	19781	10,9	1
5	19818	15,6	1
8	19768	10,3	1
9	19739	16,2	2
7	19771	24	2
11	19721	13,2	2
10	19727	16,2	2
2	19913	22,2	3
4	19862	15,6	3
3	19910	6,2	3



Gambar 3.4 Hasil visualisasi dari Tabel 3.8

Tabel 3.9. Hasil perhitungan menentukan pusat baru kelompok *(Cluster*)

Cluster	Bus	Tegangan (Volt)	Rata-rata Tegangan (Volt)	Arus (Ampere)	Rata-rata Arus (Ampere)
1	6	19781	V_{c1} = 19789,0	10,9	I_{c1}=12,27
	5	19818		15,6	
	8	19768		10,3	
2	9	19739	V_{c2}=19739,5	16,2	I_{c2}=17,40
	7	19771		24	
	11	19721		13,2	
	10	19727		16,2	
3	2	19913	V_{c3}=19895,0	22,2	I_{c3}=14,67
	4	19862		15,6	
	3	19910		6,2	

Sebelum melangkah ke iterasi kedua maka dilakukan penentuan pusat baru untuk masing-masing *cluster* berdasarkan data tegangan dan arus pada Tabel 3.8. Pusat baru ditentukan dengan menghitung nilai rata-rata tegangan (Vcx) dan arus (Icx) dari semua bus anggota di masing-masing *cluster*, dengan x = nomor *cluster* dan c = pusat (*center*). Tabel 3.9 menunjukkan hasil perhitungan pusat baru di masing-masing *cluster*.

Pada **iterasi kedua**, dilakukan proses menghitung jarak bus n ke pusat baru untuk masing-masing cluster. Jarak setiap bus ke pusat (D_x) dapat dihitung menggunakan rumus dalam persamaan 3.15 dengan menggunakan dua parameter yaitu tegangan dan arus, berikut contoh untuk mendapatkan nilai D_x:

$$D_x = \sqrt{(V_x - V_{cx})^2 + (I_x - I_{cx})^2}$$

$$D_6 = \sqrt{(19781 - 19789)^2 + (10.9 - 12.667)^2} = 8.11590$$

$$D_5 = \sqrt{(19818 - 19789)^2 + (15.6 - 12.667)^2} = 29.19094$$

$$D_8 = \sqrt{(19768 - 19789)^2 + (10.3 - 12.667)^2} = 21.09189$$

Hasil selengkapnya nilai jarak (Dx) dari bus ke masing-masing pusat baru ditunjukkan pada Tabel 3.10. Setelah dicari nilai minimum Dx (jarak terpendek) maka dapat ditentukan anggota masing-masing *cluster* pada iterasi kedua, ditunjukkan pada Tabel 3.11.

Tabel 3.10 Penentuan anggota *Cluster* iterasi kedua

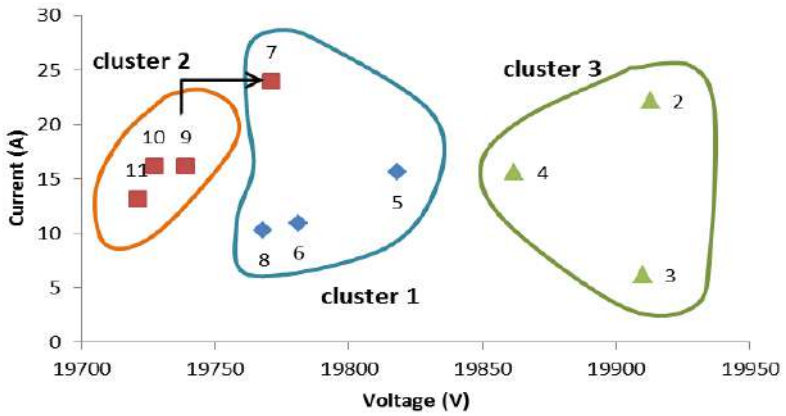
No. Bus	Jarak (D_x)			Jarak terpendek	Anggota *Cluster*
	Cluster 1	*Cluster* 2	*Cluster* 3		
6	8,12	42,01	114,06	8,12	1
5	29,19	78,52	77,01	29,19	1
8	21,09	29,37	127,08	21,09	1
9	50,16	1,30	156,01	1,30	2
7	21,49	32,18	124,35	21,49	1
11	68,01	18,97	174,01	18,97	2
10	62,13	12,56	168,01	12,56	2
2	124,40	173,57	19,51	19,51	3
4	73,08	122,51	33,01	33,01	3
3	121,15	170,87	17,23	17,23	3
			SSE	182,47	

Nilai *SSE* pada iterasi 2 dihitung, dan didapat bahwa nilai *SSE* pada iterasi 2 masih menunjukkan perubahan dibanding *SSE* pada iterasi 1, sehingga proses *cluster* dilanjutkan.

Pada Tabel 3.11 menunjukkan adanya perpindahan anggota *cluster* dari penentuan pada iterasi 1 yaitu bus 7 yang semula di *cluster* 2 berpindah ke *cluster* 1. Hasil visualisasi ditunjukkan pada Gambar 3.5.

Tabel 3.11 Hasil penentuan anggotap iterasi kedua

No. Bus	Tegangan (Volt)	Arus (mA)	Anggota *Cluster*
6	19781	10,9	1
5	19818	15,6	1
8	19768	10,3	1
7	19771	24	1
9	19739	16,2	2
11	19721	13,2	2
10	19727	16,2	2
2	19913	22,2	3
4	19862	15,6	3
3	19910	6,2	3



Gambar 3.5 Hasil visualisasi dari dengan perubahan anggota *Cluster*

Proses *cluster* di lanjutkan dengan menentukan pusat baru dari data pada Tabel 3.11. Hasil penentuan pusat baru ditunjukkan pada Tabel 3.12.

Tabel 3.12 Pusat *Cluster* baru hasil iterasi ke 2

Pusat baru	Tegangan	Arus
Cluster 1	19784,5	15,2
Cluster 2	19729	15,2
Cluster 3	19895	14,67

Pada **iterasi ketiga,** proses *Integer Linear K-Means Clustering* dilakukan dengan cara yang sama seperti pada iterasi sebelumnya dengan menggunakan pusat cluster baru pada Tabel 3.12. Hasilnya ditunjukkan pada Tabel 3.13 dan 3.14. Dari Tabel 3.13 dihitung nilai *SSE* untuk iterasi 3. Hasil perhitungan nilai *SSE* masih berubah dibanding dengan nilai *SSE* pada iterasi 2, sehinggu proses *cluster* dilanjutkan.

Tabel 3.13 Menghitung jarak ke pusat dan anggota *Cluster* iterasi 3

No. Bus	Jarak (D_x)			Jarak terpendek	Anggota *Cluster*
	Cluster 1	*Cluster* 2	*Cluster* 3		
6	5,54	52,18	114,06	5,54	1
5	33,50	89,00	77,01	33,50	1
8	17,21	39,31	127,08	17,21	1
7	16,12	42,91	124,35	16,12	1
9	45,51	10,05	156,01	10,05	2
11	63,53	8,25	174,01	8,25	2
10	57,51	2,24	168,01	2,24	2
2	128,69	184,13	19,51	19,51	3
4	77,50	133,00	33,01	33,01	3
3	125,82	181,22	17,23	17,23	3
			SSE	162,66	

Tabel 3.14 Hasil penentuan anggota *Cluster* iterasi 3

No. Bus	Tegangan (Volt)	Arus (A)	*Cluster*
6	19781	10,9	1
5	19818	15,6	1
8	19768	10,3	1
7	19771	24	1
9	19739	16,2	2
11	19721	13,2	2
10	19727	16,2	2
2	19913	22,2	3
4	19862	15,6	3
3	19910	6,2	3

Tabel 3.15 Pusat *Cluster* baru hasil iterasi ke 3

Pusat baru	Tegangan (Volt)	Arus (A)
Cluster 1	19784,5	15,2
Cluster 2	19729	15,2
Cluster 3	19895	14,67

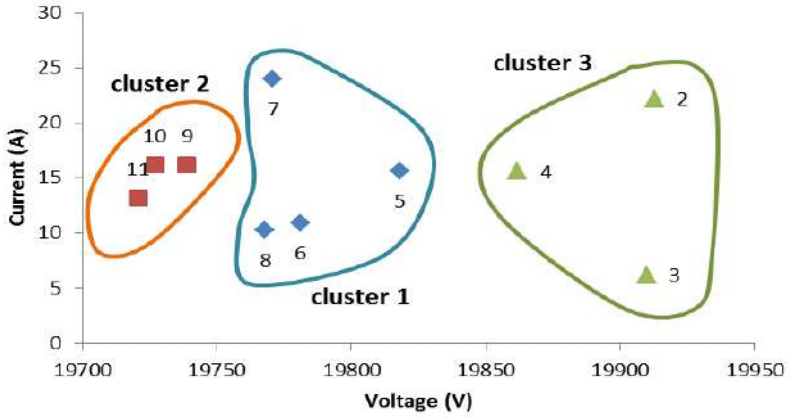
Pada **iterasi keempat,** proses *Integer Linear K-Means Clustering* dilakukan dengan cara yang sama seperti pada iterasi sebelumnya dengan menggunakan pusat *cluster* baru pada Tabel 3.15. Hasilnya ditunjukkan pada Tabel 3.16 dan 3.17. Dari Tabel 3.16 dihitung nilai *SSE* untuk iterasi 4. Hasil perhitungan nilai *SSE* ternyata **sama** dengan nilai *SSE* pada iterasi 3 sehingga proses *cluster* dihentikan. Visualisasi dari proses *cluster* yang mencapai *convergent* ditunjukkan pada Gambar 3.6.

Tabel 3.16 Menghitung jarak ke pusat dan anggota *cluster* iterasi 4

No. Bus	Jarak (D_x)			Jarak terpendek	Anggota *Cluster*
	Cluster 1	*Cluster* 2	*Cluster* 3		
6	5,54	52,18	114,06	5,54	1
5	33,50	89,00	77,01	33,50	1
8	17,21	39,31	127,08	17,21	1
7	16,12	42,91	124,35	16,12	1
9	45,51	10,05	156,01	10,05	2
11	63,53	8,25	174,01	8,25	2
10	57,51	2,24	168,01	2,24	2
2	128,69	184,13	19,51	19,51	3
4	77,50	133,00	33,01	33,01	3
3	125,82	181,22	17,23	17,23	3
			SSE	162,66	

Tabel 3.17 Hasil penentuan akhir anggota *Cluster*

No. Bus	Tegangan (Volt)	Arus (A)	*Cluster*
6	19781	10,9	1
5	19818	15,6	1
8	19768	10,3	1
7	19771	24	1
9	19739	16,2	2
11	19721	13,2	2
10	19727	16,2	2
2	19913	22,2	3
4	19862	15,6	3
3	19910	6,2	3



Gambar 3.6 Proses akhir dari *cluster* yang mencapai *convergent*

Untuk penempatan *PMU* yang optimal, ditentukan dari nilai D_x terkecil dari bus dalam *cluster* yang sama. Sehingga diperoleh pada *cluster* 1 dengan bus 5, 6, 7, dan 8 pada nilai D_x terkecil di bus 6. Sedangkan di *cluster* 2 dengan bus 9, 10, dan 11, nilai D_x terkecil ada di bus 9. Demikian pula untuk *cluster* 3 dengan bus 2, 3, dan 4, nilai D_x terkecil ada di bus 3. Sehingga penempatan *PMU* optimal pada bus 3, 6, dan 9.

Pengujian 3 parameter: Tegangan, Arus dan Impedansi

Pengujian kedua digunakan tiga parameter, yaitu tegangan, arus, dan impedansi untuk menentukan penempatan optimal *PMU* dalam jaringan distribusi Bendul Merisi secara acak ditunjukkan pada Tabel 3.18.

Tabel 3.18 Penentuan pusat *cluster* secara acak

No. Bus	Tegangan (Volt)	Arus (Ampere)	Impedansi	Anggota *Cluster*
6	19781	10,9	0,23836	1 (pusat *cluster*1)
9	19739	16,2	0,04869	
2	19913	22,2	0,07861	
5	19818	15,6	0,09788	
7	19771	24	0,39681	2 (pusat *cluster* 2)
11	19721	13,2	0,27005	
4	19862	15,6	0,19424	
3	19910	6,2	0,05553	3 (pusat *cluster* 3)
8	19768	10,3	0,2776	
10	19727	16,2	0,13845	

Jarak dari masing-masing bus ke pusat (D_x) dapat dihitung dengan menggunakan persamaan 3.15, untuk tiga parameter tegangan, arus, dan impedansi. Pada Tabel 3.18 juga telah ditentukan pusat *cluster* awal untuk 3 *cluster*. Sedangkan Tabel 3.19 menunjukkan hasil hitungan jarak *Euclidean* dari semua bus ke masing-masing pusat *cluster*. Tabel 3.20 menunjukkan anggota *cluster* untuk semua bus setelah didapatkan jarak terpendek dan menghitung nilai *SSE* pada itersi 1. Tabel 3.21 menunjukkan hasil penentuan anggota *cluster* untuk masing-masing bus pada iterasi 1.

Tabel 3.19. Jarak bus n ke pusat *Cluster* (jarak *Euclidean*) pada iterasi 1

Pusat bus 6		Pusat bus 7		Pusat bus 3	
Bus n ke pusat	Jarak (D_x)	Bus n ke pusat	Jarak (D_x)	Bus n ke pusat	Jarak (D_x)
6 ke 6	0	6 ke 7	16,48	6 ke 3	129,09
9 ke 6	42,33	9 ke 7	32,94	9 ke 3	171,29
2 ke 6	132,48	2 ke 7	142,01	2 ke 3	16,28
5 ke 6	37,30	5 ke 7	47,75	5 ke 3	92,48
7 ke 6	16,48	7 ke 7	0	7 ke 3	140,14
11 ke 6	60,04	11 ke 7	51,15	11 ke 3	189,13
4 ke 6	81,14	4 ke 7	91,39	4 ke 3	48,91
3 ke 6	129,09	3 ke 7	140,14	3 ke 3	0
8 ke 6	13,01	8 ke 7	14,03	8 ke 3	142,06
10 ke 6	54,26	10 ke 7	44,69	10 ke 3	183,27

Tabel 3.20 Penentuan anggota *Cluster* dari jarak terpendek iterasi 1

No. Bus	Jarak (D_x)			Jarak terpendek	Anggota *Cluster*
	Cluster 1	*Cluster* 2	*Cluster* 3		
6	0	16,48	129,09	0	1
9	42,33	32,94	171,29	32,94	2
2	132,48	142,01	16,28	16,28	3
5	37,30	47,75	92,48	37,30	1
7	16,48	0	140,14	0	2
11	60,04	51,15	189,13	51,15	2
4	81,14	91,39	48,91	48,91	3
3	129,09	140,14	0	0	3
8	13,01	14,03	142,06	13,01	1
10	54,26	44,69	183,27	44,69	2
			SSE	244,28	

Tabel 3.21 Hasil penentuan anggota *Cluster* iterasi 1

No. Bus	Tegangan (Volt)	Arus (Ampere)	Impedansi(Ohm)	*Cluster*
6	19781	10,9	0,24	1
5	19818	15,6	0,10	1
8	19768	10,3	0,28	1
9	19739	16,2	0,05	2
7	19771	24	0,40	2
11	19721	13,2	0,27	2
10	19727	16,2	0,14	2
2	19913	22,2	0,08	3
4	19862	15,6	0,19	3
3	19910	6,2	0,06	3

Tabel 3.22 Tegangan, Arus, dan Impedansi dalam *Cluster* 1

No. Bus	Tegangan (Volt)	Arus (Ampere)	Impedansi
6	$V_6 = 19781$	$I_6 = 10.9$	$Z_6 = 0,24$
5	$V_5 = 19818$	$I_5 = 15.6$	$Z_5 = 0,10$
8	$V_8 = 19768$	$I_8 = 10.3$	$Z_8 = 0,28$
Rata-rata	$V_{cx} = 19789$	$I_{cx} = 12.67$	$Z_{cx} = 0,21$

Tabel 3.23 Pusat baru dari hasil iterasi ke 1

	Tegangan (Volt)	Arus (Ampere)	Impedansi
Cluster 1	19789,0	12,27	0,21
Cluster 2	19739,5	17,40	0,21
Cluster 3	19895,0	14,67	0,11

Tabel 3.24 Menghitung jarak ke pusat dan anggota *Cluster* iterasi 2

No. Bus	Jarak (Dx)			Jarak terpendek	Anggota *Cluster*
	Cluster 1	*Cluster* 2	*Cluster* 3		
6	8,12	42,01	114,06	8,12	1
5	29,19	78,52	77,01	29,19	1
8	21,09	29,37	127,08	21,09	1
9	50,16	1,31	156,01	1,31	2
7	21,49	32,19	124,35	21,49	1
11	68,01	18,97	174,01	18,97	2
10	62,13	12,56	168,01	12,56	2
2	124,40	173,57	19,51	19,51	3
4	73,08	122,51	33,01	33,01	3
3	121,15	170,87	17,23	17,23	3
			SSE	182,48	

Tabel 3.22 menunjukkan perhitungan pusat baru *cluster* 1 dengan menghitung rata-rata dari tegangan (V_{cx}), arus (I_{cx}) dan impedansi (Z_{cx}) dari semua bus anggota *cluster* 1. Tabel 3.23 menunjukkan pusat baru dari semua *cluster.*

Setelah ditentukan pusat baru semua *cluster* pada iterasi 1, dilanjutkan proses mencari pusat baru pada iterasi 2. Tabel 3.24 menentukan jarak semua bus dengan pusat *cluster* baru dan Tabel 3.25 menunjukkan anggota cluster pada iterasi ke 2. Karena nilai *SSE* masih berubah dbanding nilai *SSE* pada iterasi ke 1, maka proses *cluster* dilanjutkan. Tabel 3.26 menunjukkan pusat baru pada iterasi 2.

Tabel 3.25 Hasil penentuan anggota *Cluster* iterasi 2

No. Bus	Tegangan (Volt)	Arus (Ampere)	Impedansi	*Cluster*
6	19781	10,9	0,24	1
5	19818	15,6	0,10	1
8	19768	10,3	0,28	1
7	19771	24,0	0,40	1
9	19739	16,2	0,05	2
11	19721	13,2	0,27	2
10	19727	16,2	0,14	2
2	19913	22,2	0,08	3
4	19862	15,6	0,19	3
3	19910	6,2	0,06	3

Tabel 3.26 Pusat baru pada iterasi 2

	Tegangan (Volt)	Arus (Ampere)	Impedansi
Cluster 1	19784,5	15,2	0,25
Cluster 2	19729,0	15,2	0,15
Cluster 3	19895,0	14,67	0,11

Data iterasi terakhir *SSE* adalah 162,6593 menunjukkan *convergent* di tunjukkan pada Tabel 3.27 dan hasil yang dapat dikatakan sama dibandingkan dengan pengujian pertama menggunakan dua parameter dengan selisih nilai 0,0001. Ini karena nilai impedansi yang sangat kecil dibandingkan dengan nilai tegangan dan arus. Hasil akhir penempatan *PMU* yang optimal, juga diperoleh di mana

penempatan pusat *PMU* pada bus 6 untuk *cluster*1, bus 9 untuk *cluster* 2 dan bus 3 untuk *cluster* 3.

Tabel 3.27 Menghitung jarak ke pusat dan anggota *Cluster* iterasi 3

No. Bus	Jarak (D_x)			Jarak terpendek	Anggota *Cluster*
	Cluster 1	*Cluster* 2	*Cluster* 3		
6	5,54	52,18	114,06	5,54	1
5	33,50	89,00	77,01	33,50	1
8	17,21	39,31	127,08	17,21	1
7	16,12	42,91	124,35	16,12	1
9	45,51	10,05	156,01	10,05	2
11	63,53	8,25	174,01	8,25	2
10	57,51	2,24	168,01	2,24	2
2	128,69	184,13	19,51	19,51	3
4	77,50	133,00	33,01	33,01	3
3	125,82	181,22	17,23	17,23	3
			SSE	162,66	

Tabel 3.28 Pusat baru pada iterasi 3

	Tegangan (Volt)	Arus (Ampere)	Impedansi
Cluster 1	19784,5	15,20	0,25
Cluster 2	19729,0	15,20	0,15
Cluster 3	19895,0	14,67	0,11

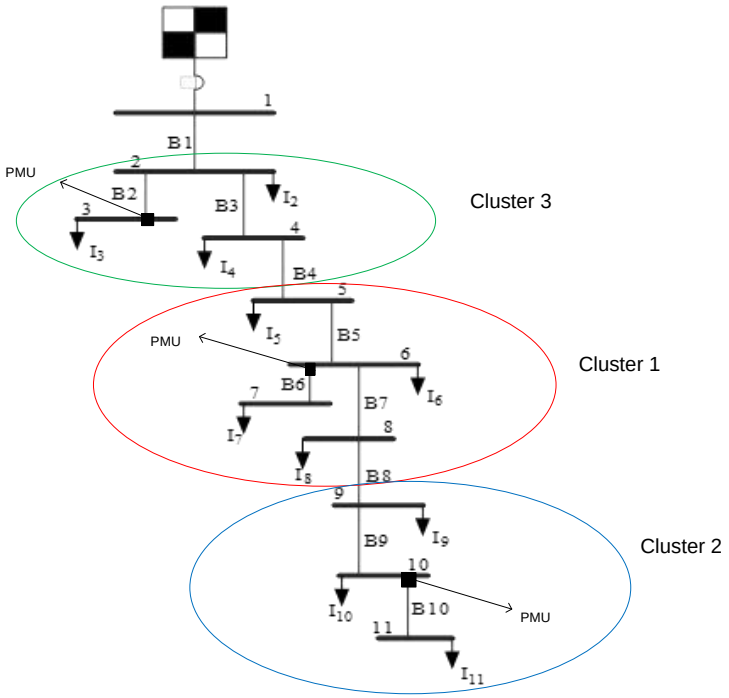
Tabel 3.29. Menghitung jarak ke pusat dan anggota *Cluster* pada iterasi 4

No. Bus	Jarak (D_x)			Jarak terpendek	Anggota *Cluster*
	Cluster 1	*Cluster* 2	*Cluster* 3		
6	5,54	52,18	114,06	5,54	1
5	33,50	89,00	77,01	33,50	1
8	17,21	39,31	127,08	17,21	1
7	16,12	42,91	124,35	16,12	1
9	45,51	10,05	156,01	10,05	2
11	63,53	8,25	174,01	8,25	2
10	57,51	2,24	168,01	2,24	2
2	128,69	184,13	19,51	19,51	3
4	77,50	133,00	33,01	33,01	3
3	125,82	181,22	17,23	17,23	3
			SSE	162,66	

Tabel 3.30. Hasil penentuan akhir anggota *Cluster*

No. Bus	Tegangan (Volt)	Arus (Ampere)	Impedansi	*Cluster*
6	19781	10,9	0,24	1
5	19818	15,6	0,10	1
8	19768	10,3	0,28	1
7	19771	24	0,40	1
9	19739	16,2	0,05	2
11	19721	13,2	0,27	2
10	19727	16,2	0,14	2
2	19913	22,2	0,08	3
4	19862	15,6	0,19	3
3	19910	6,2	0,06	3

Untuk memvisualisasikan hasil penentuan *cluster* yang dipilih untuk semua bus menggunakan *Integer Linear K-Means* hasilnya ditunjukkan pada Gambar 3.7. Gambar 3.7 menunjukkan bahwa bus 5, 6, dan 8 *cluster* 1. Sedangkan *cluster* 2 terdiri bus 9,10, dan 11 , dan bus 2, 3, dan 4 masuk *cluster* 3. Hasil ini sama dengan hasil pengujian pertama menggunakan tiga parameter tegangan , arus dan impedansi.



Gambar 3.7 Hasil akhir *Cluster* dari *single line* Bendul Merisi dengan parameter tegangan, arus dan impedansi

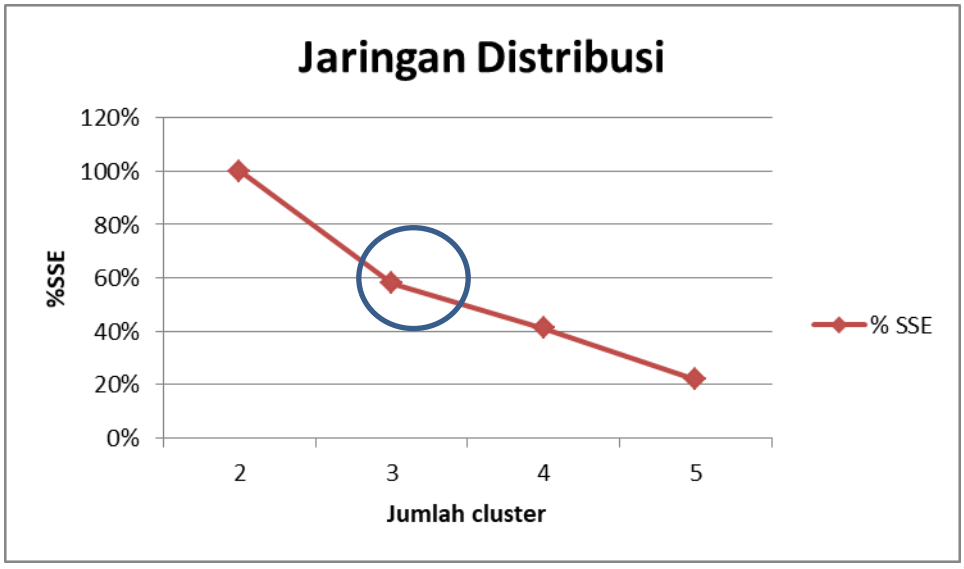
Optimasi penempatan *PMU* menggunakan persamaan metode *Elbow* yang di tunjukkan pada persamaan 2.11. Metode *Elbow* menentukan jumlah *cluster* terbaik dengan cara melihat persentase hasil perbandingan antara jumlah *cluster* yang akan membentuk siku pada suatu titik. Untuk mendapatkan perbandingannya adalah dengan menghitung *SSE (Sum of Square Error)* dari masing-masing nilai *cluster*. Karena semakin besar jumlah *cluster*K maka nilai *SSE* akan semakin kecil.

Hasil optimasi peletakan *PMU* dengan metode *Elbow* dari persamaan 2.11 dapat di simpulkan bahwa dengan 3 *cluster* merupakan jumlah *PMU* yang optimal dengan dapatkan perubahan penurunan nilai yang ekstrim. Hal ini di tunjukkan di lihat pada Tabel 3.31. Jumlah optimum penempatan *PMU* di tunjukkan pada Gambar 3.8.

Tabel 3.31 Optimasi *PMU* 11 Bus metode *Elbow*

	Jumlah *Cluster*			
	2	3	4	5
SSE	280,82 (100%)	162,66 (58%)	115,40 (41%)	61,92 (22%)
Elbow	-	118,17	47,26	53,48

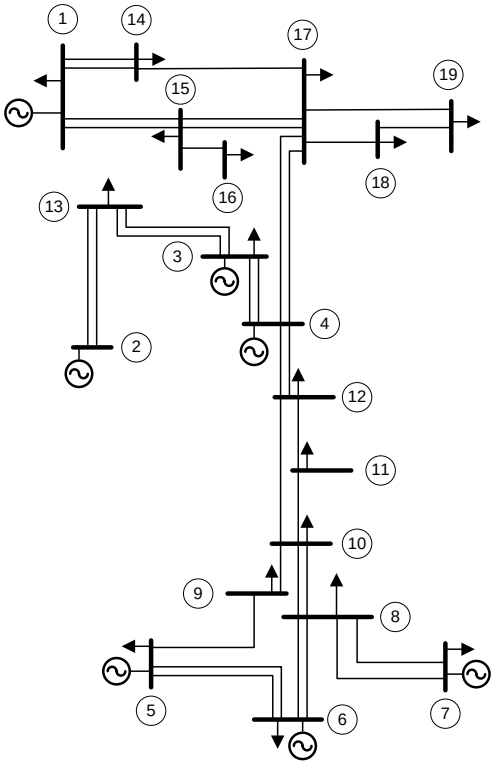
Dengan metode *Elbow* atau siku , perubahan *SSE* dari berbagai macam jumlah *cluster* yg paling ekstrim (membentuk siku) itu optimal. *SSE* kecil itu menunjukkan jumlah *cluster*nya banyak. Dalam hal ini menunjukkan *PMU* banyak. Sedangkan yang dicari jumlah *PMU* yang sedikit namun mampu mengoptimalkan semua bus dalam jaringan distribusi. untuk itu metode *Elbow* utk menentukan yg optimal. *Elbow* menyebutkan banyaknya perubahan penurunan *SSE* yang paling ekstrim yang membentuk siku dari berbagai hitungan dari berbagai macam jumlah *cluster*. Sedangkan dari pengujian ini menunjukkan jumlah *cluster* sesuai dengan jumlah *PMU*.



Gambar 3.8 Grafik Jumlah *Cluster* optimal

3.3 Analisa Jaringan dengan Metode *Integer Linear K-Means Clustering* Pada Jaringan Transmisi Jawa- Bali 500 kV

Sistem yang diuji adalah sistem Jawa-Bali 500 kV dengan 19 bus [39]. Gambar *single line* 19 bus di tunjukkan pada Gambar 3.9.



Gambar 3.9 *Single line* diagram sistem 19 Bus Jawa-Bali 500 kV [39]

Dari gambar *single line* diagram sistem 19 bus Jawa-Bali 500 kV tersebut dapat ditentukan koneksi antar bus dalam matrik.

f_1 adalah bus ke 1 $= X_1 + X_{14} + X_{15} \geq 1$

f_2 adalah bus ke 2 $= X_2 + X_{13} \geq 1$

f_3 adalah bus ke 3 $= X_3 + X_4 + X_{13} \geq 1$

f_4 adalah bus ke 4 $= X_4 + X_{12} + X_{17} \geq 1$

f_5 adalah bus ke 5 $= X_5 + X_6 + X_9 \geq 1$

f_6 adalah bus ke 6 $= X_6 + X_8 \geq 1$

f_7 adalah bus ke 7 $= X_7 + X_8 \geq 1$

f_8 adalah bus ke 8 $= X_9 + X_{10} \geq 1$

f_9 adalah bus ke 9 $= X_9 + X_{10} \geq 1$

f_{10} adalah bus ke 10 $= X_{10}+X_{11} + X_{12} \geq 1$

f_{11} adalah bus ke 11 $= X_{11} + X_{12} \geq 1$

f_{14} adalah bus ke 14 $= X_{17} \geq 1$

f_{15} adalah bus ke 15 $= X_{15} + X_{16} + X_{17} \geq 1$

f_{17} adalah bus ke 17 $= X_{17} + X_{18} + X_{19} \geq 1$

f_{18} adalah bus ke 18 $= X_{18} + X_{19} \geq 1$

Optimasi penempatan *PMU* di tentukan dengan daya beban yang datanya di tunjukkan pada Tabel 3.32 dan Tabel 3.33.

Tabel 3.32 Data Saluran Jaringan Transmisi Jawa-Bali

No	Bus Ke Bus	Panjang (km)	R (ohm)	X (ohm)
1	1-14	12,48	0,0251	0,2808
2	1-15	111	0,0293	0,2815
3	2-13	55	0,0293	0,2788
4	3-13	44,56	0,0293	0,2815
5	3-4	25,1	0,0293	0,2815
6	4-12	37,43	0,0251	0,2808
7	4-17	80,3	0,0293	0,2815
8	5-9	410	0,0293	0,2788
9	5-6	74	0,0251	0,2808
10	6-8	74	0,0251	0,2808
11	7-8	22,2	0,0293	0,2788
12	8-10	251	0,0293	0,2815
13	9-10	75	0,0293	0,2788
14	10-11	228,7	0,0293	0,2815
15	10-12	342,8	0,0293	0,2815

Tabel 3.32 (Lanjutan …)

No	Bus Ke Bus	Panjang (km)	R (ohm)	X (ohm)
16	11-12	130	0,0293	0,2815
17	14-17	116	0,0293	0,2815
18	15-16	31,9	0,0251	0,2808
19	15-17	21,3	0,0293	0,2815
20	17-18	37,92	0,0293	0,2815
21	17-19	57	0,0293	0,2815
22	18-19	18	0,0293	0,2815

Tabel 3.33 Perhitungan Hasil Aliran Daya Sistem *Newhon Rhapson*

No. Bus	Tegangan (pu)	Sudut (°)	Pembangkit		Konsumsi	
			P(MW)	Q(MVar)	P(MW)	Q(MVar)
1	1,00	0,00	2515	1437	132	44
2	0,98	-3,06	300	457	0	0
3	0,96	-3,16	600	686	527	195
4	0,95	-2,97	432	800	0	0
5	1,00	39,74	2800	533	609	235
6	1,00	36,12	300	1043	104	15
7	0,97	32,51	800	678	187	27
8	0,96	32,10	0	0	787	581
9	0,84	19,42	0	0	424	219
10	0,85	17,82	0	0	213	284
11	0,86	1,82	0	0	406	188
12	0,92	-2,40	0	0	718	496
13	0,97	-3,52	0	0	513	243
14	0,99	-0,43	0	0	551	214
15	0,95	-4,50	0	0	497	137
16	0,94	-5,14	0	0	569	194
17	0,94	-4,62	0	0	419	654
18	0,93	-6,02	0	0	427	289
19	0,93	-6,16	0	0	477	82

Tabel 3.34 Data Tegangan dan Sudut Jaringan Transmisi Jawa-Bali

No. Bus	Tegangan (pu)	Sudut (°)
1	1,00	0,00
2	0,98	-3,06
3	0,96	-3,16
4	0,95	-2,97
5	1,00	39,74
6	1,00	36,12
7	0,97	32,51
8	0,96	32,10

Tabel 3.34 (Lanjutan …)

No. Bus	Tegangan (pu)	Sudut (°)
9	0,84	19,42
10	0,85	17,82
11	0,86	1,82
12	0,92	-2,40
13	0,97	-3,52
14	0,99	-0,43
15	0,95	-4,50
16	0,94	-5,14
17	0,94	-4,62
18	0,93	-6,02
19	0,93	-6,16

Proses *cluster* untuk 19 bus jaringan transmisi Jawa-Bali 500 kV diawali dengan menentukan 3 pusat *cluster* pada data Tegangan dan Sudut yang ditunjukkan pada Tabel 3.35. Pusat *cluster* 1 ditetapkan pada Bus 5, pusat *cluster* 2 pada bus 1 dan pusat *cluster* 3 pada bus 2.

Tabel 3.35. Penentuan Pusat *Cluster* Awal secara acak

No. Bus	Tegangan (pu)	Sudut (°)	Pusat *Cluster*
9	0,84	19,42	
11	0,86	1,82	
18	0,93	-6,02	
5	1,00	39,74	Pusat *Cluster* 1
6	1,00	36,12	
8	0,96	32,10	
13	0,97	-3,52	
1	1,00	0	Pusat *Cluster* 2
10	0,85	17,82	
16	0,94	-5,14	
14	0,99	-0,43	
19	0,93	-6,16	
12	0,92	-2,40	
7	0,97	32,51	
3	0,96	-3,16	
2	0,98	-3,06	Pusat *Cluster* 3
4	0,95	-2,97	
15	0,95	-4,50	
17	0,94	-4,62	

Kemudian dilanjutkan dengan menentukan jarak *Euclidean* untuk semua bus terhadap masing-masing pusat *cluster*. Hasilnya ditunjukkan pada Tabel 3.36.

Tabel 3.36. Jarak bus n ke pusat *Cluster* (jarak *Euclidean*) iterasi 1

Pusat bus 5		Pusat bus 1		Pusat bus 2	
Bus n ke pusat	Jarak (D_x)	Bus n ke pusat	Jarak (D_x)	Bus n ke pusat	Jarak (D_x)
bus 9 ke 5	20,32	bus 9 ke 1	19,42	bus 9 ke 2	22,47
bus 11 ke 5	37,92	bus 11 ke 1	1,82	bus 11 ke 2	4,88
bus 18 ke 5	45,76	bus 18 ke 1	6,02	bus 18 ke 2	2,97
bus 5 ke 5	0	bus 5 ke 1	39,74	bus 5 ke 2	42,79
bus 6 ke 5	3,62	bus 6 ke 1	36,12	bus 6 ke 2	39,18
bus 8 ke 5	7,64	bus 8 ke 1	32,10	bus 8 ke 2	35,16
bus 13 ke 5	43,26	bus 13 ke 1	3,52	bus 23 ke 2	0,47
bus 1 ke 5	39,74	bus 1 ke 1	0	bus 1 ke 2	3,06
bus 10 ke 5	21,92	bus 10 ke 1	17,82	bus 10 ke 2	20,88
bus 16 ke 5	44,87	bus 16 ke 1	5,14	bus 16 ke 2	2,081
bus 14 ke 5	40,17	bus 14 ke 1	0,43	bus 14 ke 2	2,63
bus 19 ke 5	45,90	bus 11 ke 1	6,16	bus 11 ke 2	3,10
bus 12 ke 5	42,14	bus 12 ke 1	2,40	bus 12 ke 2	0,66
bus 7 ke 5	7,22	bus 7 ke 1	32,51	bus 7 ke 2	35,57
bus 3 ke 5	42,90	bus 3 ke 1	3,16	bus 3 ke 2	0,11
bus 2 ke 5	42,80	bus 2 ke 1	3,06	bus 2 ke 2	0
bus 4 ke 5	42,71	bus 4 ke 1	2,97	bus 4 ke 2	0,09
bus 15 ke 5	44,24	bus 15 ke 1	4,50	bus 15 ke 2	1,44
bus 17 ke 5	44,36	bus 17 ke 1	4,62	bus 17 ke 2	1,56

Anggota setiap *cluster* di tentukan dengan mencari jarak minimum dari D_x pada 3 *cluster*. Hasilnya menunjukkan anggota dari masing-masing *cluster*, ditunjukkan pada Tabel 3.37. Tabel 3.38 menunjukkan hasil akhir iterasi 1. Dari Tabel 3.39 dihitung pusat baru untuk masing-masing *cluster*, ditunjukkan pada Tabel 3.40.

Tabel 3.37 Hasil anggota dari masing-masing *Cluster* pada iterasi 1

No. Bus	Jarak (D_x)			Jarak terpendek	Anggota *Cluster*
	Cluster 1	*Cluster* 2	*Cluster* 3		
9	20,32	19,42	16,36	16,36	3
11	37,92	1,82	1,24	1,24	3
18	45,76	6,02	2,97	2,97	3
5	0,00	39,74	36,68	0,00	1
6	3,62	36,12	33,07	3,62	1

Tabel 3.37 (Lanjutan …)

No. Bus	Jarak (D_x)			Jarak terpendek	Anggota *Cluster*
	Cluster 1	*Cluster* 2	*Cluster* 3		
8	7,64	32,10	29,04	7,64	1
13	43,26	3,52	0,47	0,47	3
1	39,74	0,00	3,06	0,00	2
10	21,92	17,82	14,77	14,77	3
16	44,87	5,14	2,08	2,08	3
14	40,17	0,43	2,63	0,43	2
19	45,90	6,16	3,10	3,10	3
12	42,14	2,40	0,66	0,66	3
7	7,22	32,51	29,46	7,22	1
3	42,90	3,16	0,10	0,10	3
2	42,79	3,06	0,00	0,00	3
4	42,71	2,97	0,09	0,09	3
15	44,24	4,50	1,44	1,44	3
17	44,36	4,62	1,56	1,56	3
			SSE	63,76	

Tabel 3.38 Hasil penentuan anggota *Cluster* iterasi 1

No. Bus	Tegangan (pu)	Sudut (°)	*Cluster*
5	1	39,74	1
6	1	36,12	1
8	0,96	32,10	1
7	0,97	32,51	1
1	1	0	2
14	0,99	0,43	2
9	0,84	19,42	3
11	0,86	1,82	3
18	0,93	6,02	3
13	0,97	3,52	3
10	0,85	17,82	3
16	0,94	5,14	3
19	0,93	6,16	3
12	0,92	2,40	3
3	0,96	3,16	3
2	0,98	3,06	3
4	0,95	2,97	3
15	0,95	4,50	3
17	0,94	4,62	3

Tabel 3.39 Hasil pusat *Cluster* baru

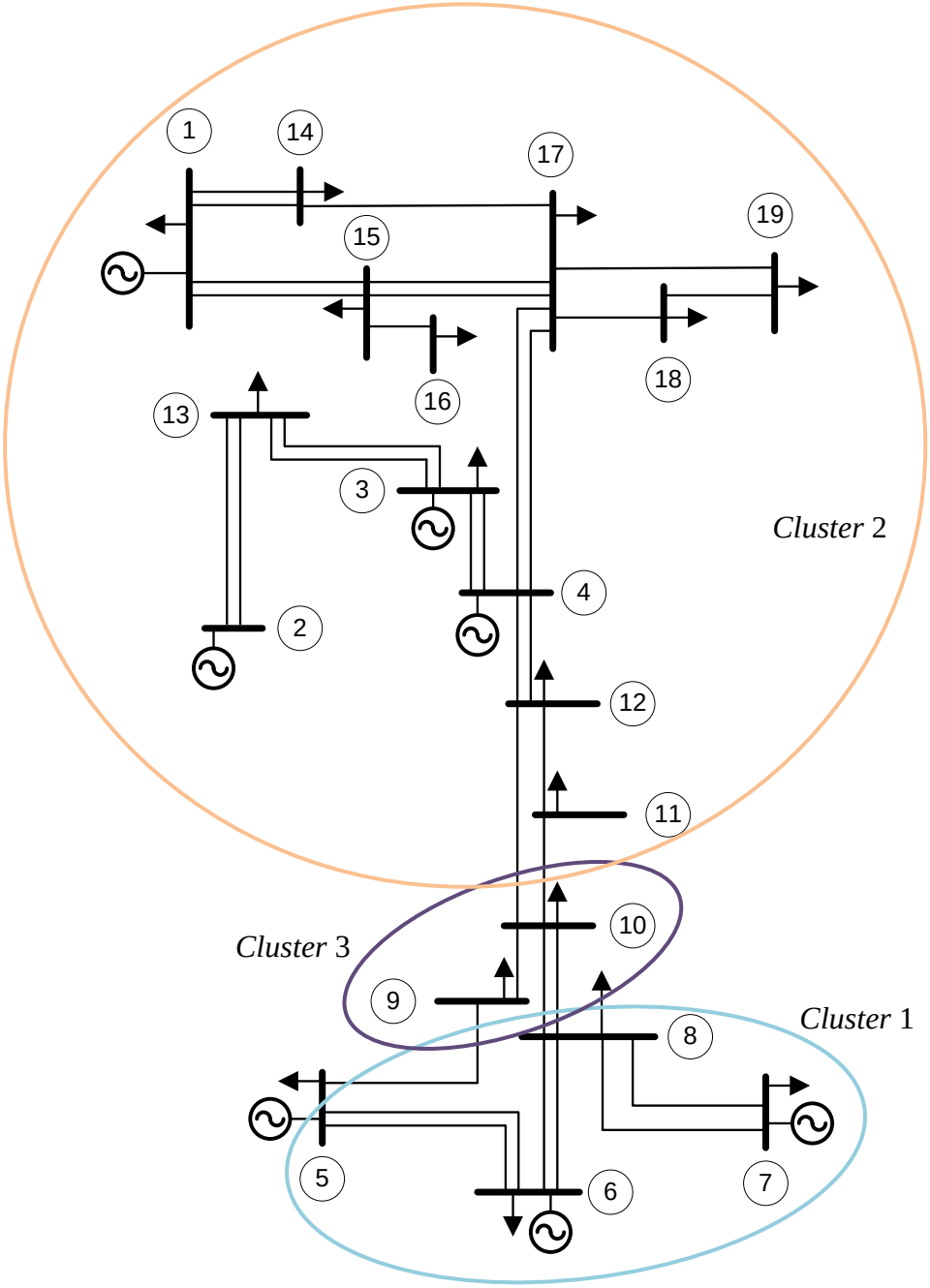
Pusat baru	Tegangan (pu)	Sudut (°)
Cluster 1	0,98	35,12
Cluster 2	1,00	0,21
Cluster 3	0,92	6,20

Dari Tabel 3.37 dihitung pula nilai *SSE* dengan cara menjumlahkan nilai jarak *Euclidean* terpendek dari ke 3 *cluster*. Nilai *SSE* pada iterasi ke 1 adalah 63,76.

Selanjutnya dilakukan proses iterasi berikutnya dengan menghitung jarak semua bus terhadap pusat baru masing-masing *cluster* dan menentukan perubahan anggota pada masing-masing *cluster*, begitu pula pada setiap iterasi dihitung nilai *SSE* nya. Pada iterasi kedua *SSE* = 56,09, pada iterasi ke tiga *SSE* = 53,58, pada iterasi keempat *SSE* = 43,92, pada iterasi kelima *SSE* = 32,35. Sedangkan pada iterasi keenam nilai *SSE* = 32,35, karena nilainya tidak berubah maka proses *cluster* dihentikan. Data selengkapnya dapat dilihat di Lampiran III. Adapun anggota masing-masing pada iterasi ke 5 atau 6 ditunjukkan pada Tabel 3.40 dan divisualisasikan pada Gambar 3.9.

Tabel 3.40 Hasil penentuan anggota *Cluster* iterasi 5

No. Bus	Tegangan (pu)	Sudut (^{o})	*Cluster*
5	1	39,737	1
6	1	36,121	1
8	0,961	32,099	1
7	0,97	32,513	1
1	1	0	2
14	0,995	0,428	2
11	0,858	1,817	2
12	0,921	2,399	2
3	0,96	3,159	2
2	0,98	3,056	2
4	0,95	2,97	2
13	0,965	3,524	2
16	0,94	5,137	2
15	0,945	4,499	2
17	0,941	4,619	2
18	0,927	6,021	2
19	0,928	6,159	2
9	0,844	19,418	3
10	0,853	17,821	3

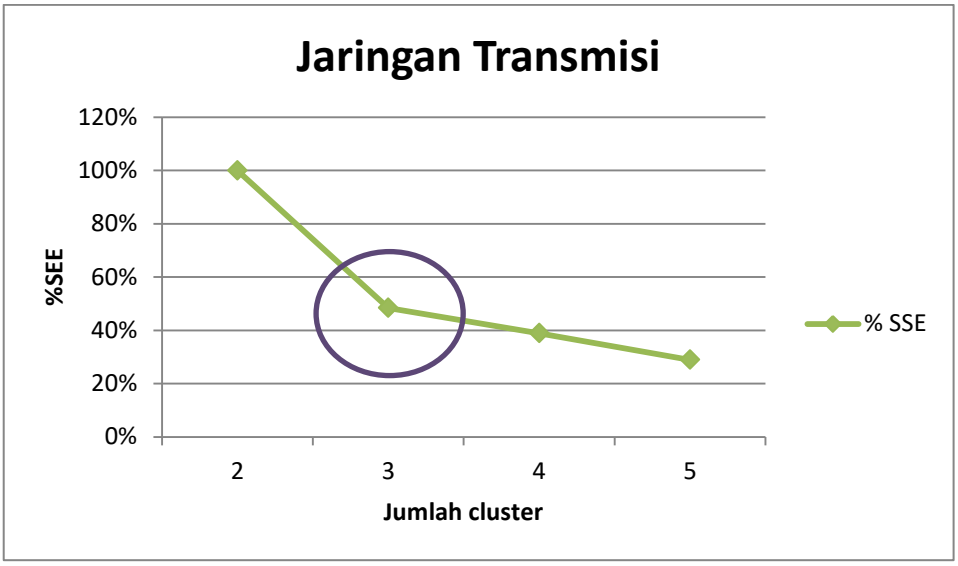


Gambar 3.10 Hasil *Clustering* 19 bus Jawa-Bali 500 kV

Hasil optimasi peletakan *PMU* dengan metode *Elbow* dari persamaan 2.11 dapat di lihat pada Tabel 3.41

Tabel 3.41 Optimasi *PMU* 19 Bus Jawa-Bali 500 kV metode *Elbow*

	Jumlah *Cluster*			
	2	3	4	5
SSE	66,87	32,35 (48%)	26,02 (39%)	19,34 (29%)
Elbow		34,53	6,33	6,67



Gambar 3.11 Grafik Jumlah *Cluster* optimal

3.4 Perbandingan *Integer Linear K-Means Clustering* dengan Metode *Artificial Intelegent* yang lain

Dalam penelitian optimasi penempatan *PMU* menggunakan metode Linear *Integer Programming* pada referensi 4 dan 5, *Depth of one unobservability* menjelaskan bahwa tidak boleh terdapat dua bus yang terhubung sedangkan kedua bus tersebut sama-sama tidak terobservasi. Pada Rumusan ini perlu dikembangkan untuk permasalahan seperti di atas karena perlu mempertimbangkan kondisi untuk dua bus yang saling terhubung. Matrik T_{PMU}, yang merupakan matrik hubungan antar bus antar bus, perlu didefinisikan lagi. yang di tunjukkan pada persamaan 3.18 dan 3.19.

$$\text{Min} \sum_{k=1}^{N} X_k \quad (3.18)$$

$$T_{PMU}\, X \geq b_{PMU} \quad (3.19)$$

Apabila didefinisikan suatu vektor Y = T_{PMU} X, maka elemen dari vektor Y, persamaan 3.20

$$y_i = T_{PMU,i}\, X \quad (3.20)$$

menunjukkan berapa kali bus ke-i diobservasi oleh *PMU*. T_{PMU}, i adalah baris ke-i dari matrik T_{PMU}. Jumlah total dari y_i dipengaruhi oleh dua bus yang terhubung oleh suatu line yang paling tidak satu diantaranya terobservasi oleh *PMU*. Batasan untuk permasalahan ini menjadi pada persamaan 3.21

$$b_1 = [1\ 1\ \ldots\ 1]{M_{1x1}}^T \quad (3.21)$$

Variabel M_1 adalah jumlah line pada sistem. Perumusannya menjadi seperti berikut persamaan 3.22 dan 3.23

$$\text{Min} \sum_{k=1}^{N} xd_k \quad (3.22)$$

$$A\,T_{PMU}\,Xd \geq b_1 \quad (3.23)$$

$$X_d = [xd_1\ xd_2\ \ldots\ xd_N]^T$$

$$x_i \in \{0,1\}$$

Dimana:

T_{PMU} : matrik hubungan antar bus

Xd : penempatan *PMU*

b_1 : batasan untuk permasalahan

N : banyaknya bus

A : matrik hubungan antara dua bus oleh suatu line

Dengan metode *Linear Integer Programming* menunjukkan hasil yang cukup baik. Namun ada kelemahan jika input data untuk beberapa parameter yang digunakan bukan nilai integer. Jika data yang di olah bukan integer maka tidak akurat. Untuk Menggunakan metode *K-means* lebih akurat, jika data terlalu jauh dari pusat, nilai batas maksimum dan minimum diperlukan. Oleh karena itu kombinasi kedua metode ke dalam metode *Integer Linear K-Means Clustering* menjadi sangat akurat.

Penelitian pada bab ini membahas penempatan optimal dan pengurangan *PMU* di Jaringan distribusi radial di Bendul Merisi sebagai obyek penelitian ini memiliki 11 bus dan jaringan Transmisi 19 bus Jawa-Bali. Untuk mengoptimalkan penempatan *PMU* di lakukan perbandingan beberapa metode dilakukan hasil yang di tunjukkan pada Tabel 3.42.

Tabel 3.42. Perbandingan Optimasi penempatan *PMU* untuk 11 dan 19 bus

Tanpa optimasi	*Integer Linear Programming*	Jumlah *Cluster Integer Linear K-Means*	Jumlah reduksi *PMU*	Pusat *PMU*
11	5	3	8	3,6,10
19	12	3	16	6,9,13

Kesimpulan pada pengujian menggunakan metode *Integer Linear K-Means Clustering* bahwa gabungan metode ini memperbaiki metode *Integer Linear Programming* karena lebih teliti . Selain itu *K_Means* di gunakan untuk data yang *floting point* (pecahan*).* Untuk mendapatkan optimasi di gunakan metode *Elbow*. Dengan metode *Elbow* di dapatkan 3 *cluster* yang optimal yang masing – masing di wakili 1 *PMU*. Pada pengujian jaringan distribusi Bendul Merisi 20 kV dengan parameter tegangan, arus dan impedansi, di dapatkan hasi reduksi *PMU* sebanyak 8 buah atau 73 %, dengan *center* pada bus, 3 dengan anggota *cluster* 2, dan 4. *Center* 6 anggota *cluster* 7 dan 8. Sedangkan center 10 anggota *cluster* 9 dan 11. Pengujian pada Jaringan Transmisi 19 bus Jawa-Bali 500 kV dengan parameter tegangan dan sudut fasa di dapatkatkan 3 *cluster* dengan reduksi 84%.

(Publikasi pada jurnal internasional ***Przeglad Elektrotechniczny***, quartile Q3).

<Halaman ini sengaja dikosongkan>

BAB 4 ESTIMASI DAYA DAN TEGANGAN YANG TIDAK TERPASANG *PMU* DI JARINGAN

Estimasi daya dan tegangan adalah prosedur untuk memperkirakan daya dan tegangan semua bus dalam sistem tenaga listrik berdasarkan pengukuran yang dilakukan pada beberapa bus. Tegangan dan arus pada bus diukur dengan alat ukur. Jaringan distribusi Bendul-Merisi Surabaya-Indonesia telah menjadi objek penelitian untuk mengoptimalkan penempatan *PMU* dengan membagi menjadi 3 *cluster* dan menempatkan 3 *PMU* pada bus-bus terpilih sebagai bus yang terpasang *PMU*. Dalam studi ini, estimasi daya dan tegangan pada bus yang tidak terpasang *PMU* menggunakan algoritma *Artificial Intelegent (AI).* Algoritma ini digunakan karena kemampuannya untuk mendapatkan nilai prediksi dengan akurasi dan kecepatan tinggi dalam mencapai *convergent*.

4.1 Algoritma Estimasi pada sistem Jaringan Distribusi

Sistem distribusi 20 kV memiliki jumlah bus yang bervariasi. Untuk memantau tegangan pada sistem, sensor pengukuran dipasang di setiap bus. Sensor pengukuran hanya dipasang pada bus tertentu berdasarkan pertimbangan biaya menggunakan teknik optimisasi *PMU*. Untuk bus yang tidak terpasang *PMU* di gunakan suatu algoritma hibrid *ANFIS PSO-GA* untuk memprediksi nilai parameter tegangan dan daya pada bus dalam sistem jaringan.

4.2 Perencanaan Estimasi Daya dan Tegangan menggunakan *hybrid ANFIS PSO-GA*

Ada dua langkah persiapan untuk estimasi daya & tegangan. Langkah pertama adalah mengalisa jaringan distribusi radial tiga fasa di Bendul Merisi Surabaya Selatan-Indonesia menggunakan metode *Integer linear K-Means* untuk mengoptimalkan penempatan *PMU* dengan parameter tegangan, arus, dan impedansi. Penelitian ini dilakukan dan hasil penelitian ditunjukkan pada Gambar 4.1 dan Tabel 4.1. Gambar 4.1 memperlihatkan diagram *single line* jaringan distribusi Bendul Merisi . Dalam Gambar tersebut dapat dilihat bahwa bus 3, 6, dan 9 dipasangkan dengan *PMU* sebagai bus yang diamati dan bus lainnya tidak dipasangkan dengan *PMU* sebagai bus yang tidak teramati . Tabel 3.1

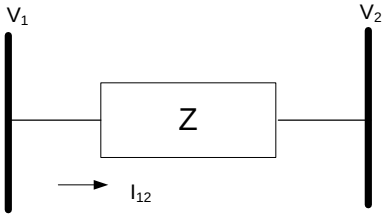
menunjukkan hasil optimasi penempatan *PMU* dengan 3 *cluster*. *Cluster*1 terdiri bus 2, 3, dan 4. Bus 3 dipilih sebagai bus terpasang *PMU*, sedangkan bus 2 dan bus 4 adalah bus yang tidak di pasang *PMU*. Dalam *cluster* 2, terdiri bus 6 adalah bus yang terpsang *PMU*, sedangkan bus 5, 7, dan 8 bus yang tidak terpasang *PMU*. Dalam *cluster* 3, bus 9 adalah bus di pasang *PMU*, untuk bus 10, dan 11 bus yang tidak *PMU*. Hasil parameter tegangan , arus dan impedansi bus 3, 6, dan 9 yang terpasang *PMU*.

Tabel 4.1 Hasil penempatan *PMU* pada jaringan distribusi Bendul Merisi

Cluster	Nomor Bus	Bus yang terpasang *PMU*	Bus yang tidak terpasang *PMU*
1	2, 3, and 4	3	2 and 4
2	5, 6, 7, and 8	6	5, 7, and 8
3	9, 10 and 11	10	10 and 11

Langkah kedua estimasi pada bus yang tidak di pasang *PMU* dilakukan dengan menggunakan *ANFIS* dengan algoritma modifikasi *hybrid PSO-GA* sehingga nilai daya dan tegangan dapat diprediksi. Bus 1 adalah bus utama sehingga tidak masuk *cluster*. *Single line* yang di gunakan ditunjukkan pada bab 3 Gambar 3.6.

Konsep untuk memperkirakan tegangan dan aliran bus pada bus yang tidak memiliki *PMU* (tidak teramati) dijelaskan di bawah ini. Tegangan dan arus yang tidak diketahui tertutup hanya dalam satu saluran, dapat diilustrasikan seperti yang ditunjukkan pada Gambar 4.1 .



Gambar 4.1 Rangkaian pengganti antar bus

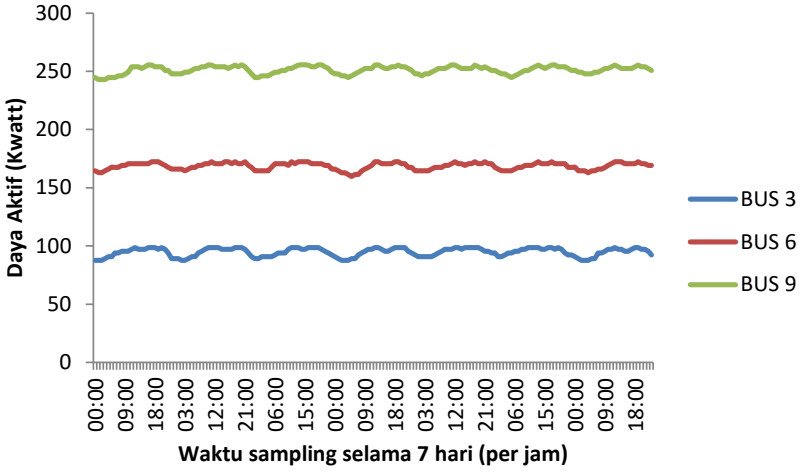
Gambar 4.1, V_1 adalah tegangan pada bus yang diamati (ada penempatan *PMU*), V_2 adalah tegangan bus yang tidak diketahui, dan I_{12} adalah arus yang keluar dari bus yang diamati , yang diketahui , ke bus yang tidak teramati itu tidak

diketahui. Untuk menemukan tegangan pada bus yang tidak dikenal, persamaan rangkaian listrik sederhana digunakan seperti yang ditunjukkan pada persamaan 4.1 .

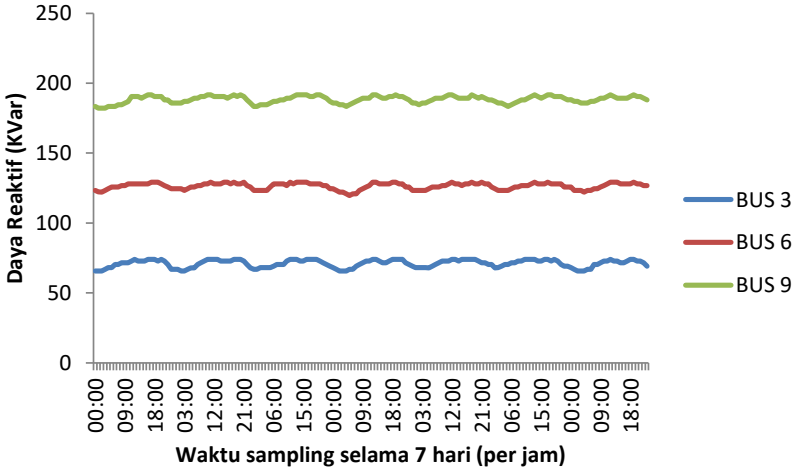
$$
\left.
\begin{aligned}
&\mathrm{Vbus}_2 = V_1 - B_1 \times Z_{12} \\
&\mathrm{Vbus}_3 = V_1 - B_1 \times Z_{12} - B_2 \times Z_{23} \\
&\mathrm{Vbus}_4 = V_1 - B_1 \times Z_{12} - B_3 \times Z_{24} \\
&\mathrm{Vbus}_5 = V_1 - B_1 \times Z_{12} - B_3 \times Z_{24} - B_4 \times Z_{45} \\
&\mathrm{Vbus}_6 = V_1 - B_1 \times Z_{12} - B_3 \times Z_{24} - B_4 \times Z_{45} - \\
&B_5 \times Z_{56} \\
&\mathrm{Vbus}_7 = V_1 - B_1 \times Z_{12} - B_3 \times Z_{24} - B_4 \times Z_{45} - \\
&B_5 \times Z_{56} - B_6 \times Z_{67} \\
&\mathrm{Vbus}_8 = V_1 - B_1 \times Z_{12} - B_3 \times Z_{24} - B_4 \times Z_{45} - \\
&B_5 \times Z_{56} - B_7 \times Z_{68} \\
&\mathrm{Vbus}_9 = V_1 - B_1 \times Z_{12} - B_3 \times Z_{24} - B_4 \times Z_{45} - \\
&B_5 \times Z_{56} - B_7 \times Z_{68} - B_8 \times Z_{89} \\
&\mathrm{Vbus}_{10} = V_1 - B_1 \times Z_{12} - B_3 \times Z_{24} - B_4 \times Z_{45} - \\
&B_5 \times Z_{56} - B_7 \times Z_{68} - B_8 \times Z_{89} - B_9 \times Z_{910} \\
&\mathrm{Vbus}_{11} = V_1 - B_1 \times Z_{12} - B_3 \times Z_{24} - B_4 \times Z_{45} - \\
&B_5 \times Z_{56} - B_7 \times Z_{68} - B_8 \times Z_{89} - B_9 \times Z_{910} - B_{10} \times Z_{1011}
\end{aligned}
\right\} \qquad (4.1)
$$

Bus 1 dan 2 pada Gambar 4. 2 dihitung dengan mempertimbangkan impedansi transformator, pergeseran fasa, dan konfigurasi dengan data pada jaringan distribusi Bendul Merisi yang dihitung menggunakan persamaan 4.1. Data dari setiap bus digunakan untuk menghasilkan daya aktif, faktor daya, dan impedansi beban yang pada akhirnya dapat menentukan perbedaan antara pengukuran dan estimasi.

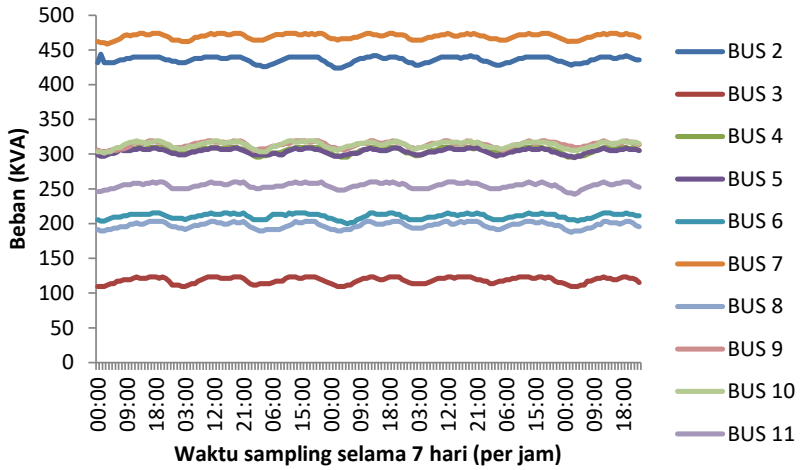
Data untuk penelitian ini dikumpulkan dari pengukuran pada setiap bus untuk parameter tegangan, arus, dan impedansi. Setiap pengukuran dikumpulkan setiap jam selama 10 hari. Jadi per hari data dikumpulkan 24 kali jadi kami mendapatkan 10 set data. Sebanyak 7 set data digunakan untuk data pelatihan dan 3 set data digunakan untuk menguji data. Data tegangan, arus, dan impedansi digunakan untuk menghitung daya aktif dan reaktif. Gambar 4.2 dan Gambar 4.3 menunjukkan daya aktif dan reaktif selama 7 hari untuk bus yang diamati (bus 3, 6, dan 9).



Gambar 4.2 Daya aktif selama 24 jam dalam 7 hari



Gambar 4.3 Daya reaktif selama 24 jam dalam 7 hari



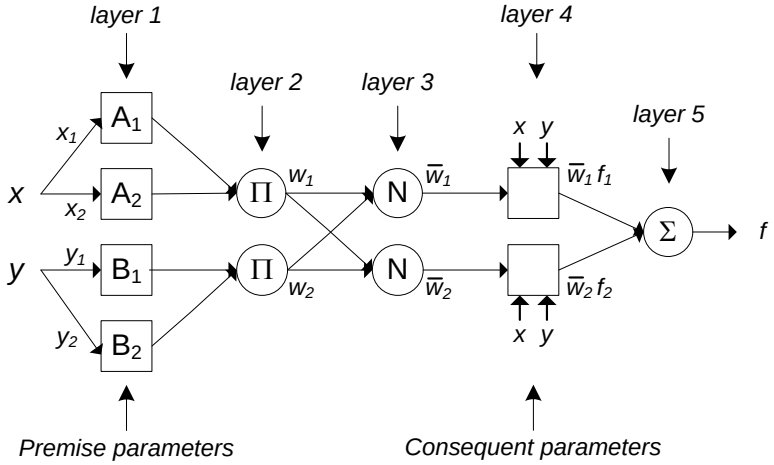
Gambar 4.4 Beban (*Load*) selama 24 jam di 7 hari

Sedangkan pada Gambar 4.4 menunjukkan nilai beban S (VA) 24 jam selama 7 hari di semua bus yang nilainya dihitung dari perkalian tegangan dan arus yang terukur . Nilai beban (*load*) untuk bus 4 dan 5 mendekati sama, begitu juga untuk nilai load pada bus 9 dan 10, jadi tampak berimpit.

4.3 Algoritma *ANFIS* dengan modifikasi *hybrid PSO-GA*

Algoritma *ANFIS* dengan *hybrid PSO-GA* yang dimodifikasi adalah kombinasi dari sistem jaringan saraf tiruan *ANFIS* dan *PSO-GA.* Metode ini dalam prosesnya ada 2 tahap yaitu *forward* dan *backward.* Arsitektur *ANFIS* ditunjukkan pada Gambar 4.5 dan persamaan 4.2.

Proses *forward* merupakan proses mulai dari data input tegangan premis a_1 , μ_1 dan arus a_1 , μ_1 di sebut sebagai premis membentuk fungsi keanggotaan untuk membentuk diagram terdapat mean yang di tunjukkan pada persamaan 4.3. Selanjutnya konsequen adalah parameter tujuan. Selanjutnya a_1,μ_1 yang di hitung bobot-bobot pada layer-1, menghitung semua nilai bobot pada masing-masing layer sampai layer output pada layer 5. Sedangkan prosesnya di hitung dengan persamaan 4.4 sampai dengan 4.7.



Gambar 4.5 Arsitektur *ANFIS*

$$\begin{aligned}&\text{Rule 1}:\text{if}(x_1 \text{ is } A_1) \text{ and } (y_1 \text{ is } B_1) \text{ then}\\&:\text{premise}(f_1 = p_1x_1 + q_1y_1 + r_1):\text{consequent}\\&\text{Rule 2}:\text{if}(x_2 \text{ is } A_2) \text{ and } (y_2 \text{ is } B_2) \text{ then}\\&:\text{premise}(f_2 = p_2x_2 + q_2y_2 + r_2):\text{consequent}\end{aligned} \tag{4.2}$$

Arsitektur *ANFIS* memiliki lima lapisan. Pada lapisan pertama, setiap neuron dihitung menggunakan fungsi aktivasi yang ditunjukkan pada persamaan 4.3. Parameter a, b, dan c adalah parameter premis yang nilainya dapat disesuaikan. Simpangan baku dan rata-rata digunakan sebagai nilai awal parameter *premise* a dan c, sedangkan nilai awal untuk parameter b ditentukan oleh nilai 1.

$$\mu(x) = \frac{1}{1 + \left| \frac{x - c}{a} \right|^{2b}} \tag{4.3}$$

Pada lapisan kedua, setiap neuron adalah neuron permanen. Setiap node mewakili predikat w (bobot) dari aturan ke-i, yang ditunjukkan pada persamaan 4.4.

$$w_i = \mu_{A_i}(x)\mu_{B_i}(y), i = 1,2 \tag{4.4}$$

Pada lapisan ketiga, setiap neuron adalah simpul yang dihitung sebagai rasio antara predikat w dari aturan ke-i dan jumlah semua predikat w, ditunjukkan oleh persamaan 4.5. *Output* dari lapisan ini dikenal sebagai bobot yang di normalisasi.

$$\overline{w}_i = \frac{w_i}{w_1 + w_2}, i = 1,2 \tag{4.5}$$

Pada lapisan keempat, setiap neuron dihitung menggunakan persamaan 4.6. Parameter {p, q, r} adalah parameter neuron yang disebut parameter konsekuensi.

$$\overline{w}_i f_i = \overline{w}_i (p_i x + q_i y + r_i), i = 1,2 \tag{4.6}$$

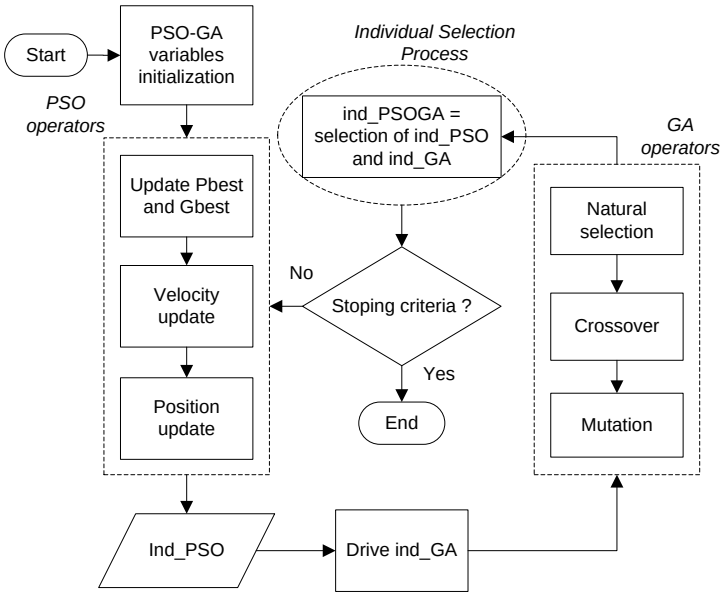
Di lapisan kelima adalah *output* keseluruhan. *Output* ini dihitung sebagai jumlah dari semua input, seperti yang ditunjukkan pada persamaan 4.7.

$$f_{out} = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{4.7}$$

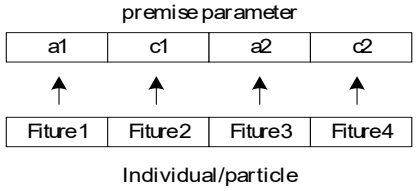
Nilai keluaran dari fungsi nilai di tunjuukan pada persamaan 4.8.

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (y_i - fout_i)^2 \tag{4.8}$$

Algoritma *ANFIS* dengan *hybrid PSO-GA* yang dimodifikasi yang digunakan dalam penelitian ini ditunjukkan pada Gambar 4.6. Gambar 4.7 menunjukkan proses transformasi dari *PSO* ke *GA*.



Gambar 4.6 Modifikasi algoritma *ANFIS* dengan *hybrid PSO*-GA



Gambar 4.7 Transformasi fitur dari parameter individual ke premis

$$k = \frac{n_{ind}}{Max_{iter}} \tag{4.9}$$

$$n_{indGA} = n_{ind} - k * iter \tag{4.10}$$

Langkah – langkah dari algoritma *ANFIS* dengan *hybrid PSO-GA* yang dimodifikasi dijelaskan pada bagian berikut:

Fungsi nilai *output* yang digunakan dalam penelitian ini adalah nilai *Mean Square Error (MSE)* yang diperoleh dari *output* metode *ANFIS* ***forward.*** Psudocode untuk menghitung fungsi nilai *output* adalah sebagai berikut.

Metode yang di gunakan algoritma *ANFIS* dengan pembelajaran arah maju *(forward)* dan arah mundur *(back).* Pertama dimasukkan nilai 2 input yaitu tegangan (X) dan input kedua arus (Y) pada *ANFIS*. Pada layer pertama di lakukan perhitungan menentukan parameter a,b,c pada rumus persamaan 4.3. Selanjutnya pada layer kedua tentukan perhitungan bobot (w) menggunakan

Persamaan 4.4. Sedangkan pada layer 3 proses perhitungan normalisasi bobot (w) menggunakan persamaan 4.5. *Output* dari lapisan ini adalah $\bar{w}$. Selanjutnya layer keempat di tentukan *consequent* nilai variabel p,q,r dengan metode *Least Square Error (LSE)* menggunakan persamaan 4.6. Keluaran akhir *ANFIS* menggunakan Persamaan 4.7. Pada persamaan 4.8 adalan nilai *MSE* ini merupakan keluaran dari fungsi nilai yang di gunakan untuk menentukan *output* dan selanjutkan bila masih *error* di back lagi sampai nilai *MSE* tidak berubah.

Setelah di dapatkan nilai *output* dengan metode *ANFIS* di lakukan proses ***ke backward ANFIS*** optimasi penggabungan antara *PSO* dan *GA*. Dimulai dengan inisialisasi *GA* parameter *PSO*, parameter *GA* , nilai-nilai untuk individu(n- ind), dan iterasi maksimum(maxiter) diilakukan pada layer bobot. Hasil acak semua ind_*PSO-GA* secara individual dan hitung fungsi nilai. Proses fungsi nilai melibatkan transformasi fitur individual dari *PSO-GA* ke parameter premise seperti yang ditunjukkan pada Gambar 4.7. Selanjutnya adalah evaluasi setiap individu ind_*PSO-GA* dengan operator *PSO*. Evaluasi ini mencakup pembaruan *P best* dan *G best*, menghitung kecepatan masing-masing individu dan menghitung posisi setiap perubahan individu.

Kemudian dilanjutkan untuk menghitung jumlah individu yang akan diproses oleh operator *GA* menggunakan persamaan 4.8 dan 4.9. Nilai individual ditentukan untuk setiap sebagai *ind_GA*. Kemudian evaluasi dilakukan untuk semua individu *GA* menggunakan operator *GA* dan hasilnya disimpan di *ind_GA*. Evaluasi ini mencakup seleksi alam, *Crossover*, dan pindah silang. Seleksi alam adalah upaya mempertahankan individu terbaik dari satu generasi ke generasi berikutnya. Operator *Crossover* digunakan untuk masing-masing individu dari dua orang tua.

Teknik *Crossover* dioperasikan menggunakan persamaan 4.11 dan 4.12. C_1 dan C_2 adalah individu baru. λ_1 adalah nilai acak dengan kisaran antara 0 dan 1, λ_2 adalah $1-\lambda_1$. X adalah orang tua pertama dan Y adalah orang tua kedua. Pindah silang adalah proses penggantian orangtua terburuk dengan orangtua baru yang ditentukan oleh nilai acak. Setelah proses evaluasi, lanjutkan dengan modifikasi *hybrid PSO-GA* asli.

Modifikasi dilakukan dengan mengurutkan ind_*PSO* dan ind_*GA* sesuai dengan nilai fungsi biaya. Ambil individu dengan jumlah yang sama dengan jumlah populasi dari peringkat atas. Individu-individu itu disebut *ind_PSO-GA*. Akhirnya, menentukan kondisi berdasarkan kriteria berhenti. Jika kriteria berhenti ditemukan maka individu terbaik, tetapi jika tidak maka kembali ke langkah evaluasi oleh setiap individu *ind_PSO*-GA dengan operator *PSO* .

$$C_1 = \lambda_1 X + \lambda_2 Y \tag{4.11}$$

$$C_2 = \lambda_2 X + \lambda_1 Y \tag{4.12}$$

Setelah proses pelatihan set-data-pelatihan dan pengujian set-data-tes menggunakan algoritma *ANFIS* dengan *hybrid PSO-GA* yang dimodifikasi, set data estimasi daya dan estimasi tegangan di peroleh, untuk bus yang tidak terpasang *PMU*. Untuk mengetahui keakuratan hasil estimasi, di gunakan rumus dalam persamaan 4.13 dan 4.14.

$$\delta_i = \frac{\left| d_m - d_e \right|}{d_e} \tag{4.13}$$

$$Accuracy = \left(1 - \frac{\sum_{i=1}^{N} \delta_i}{N} \right) x100\% \tag{4.14}$$

di mana δ_i = kesalahan, d_m = data pengukuran, d_e = data estimasi, N = jumlah data.

Berdasarkan percobaan dengan penentuan parameter *PSO-GA*, yang ditentukan sebagai berikut: Koefisien C1 dan C2 = 1; berat inersia = 1; persentase lintas = 0,7; persentase pindah silang = 0,2; populasi = 30; iterasi maksimum = 50. Proses pembelajaran 7 set data menggunakan algoritma *ANFIS* dengan *hybrid PSO-GA* yang dimodifikasi. Proses berlangsung hingga mencapai konvergen atau iterasi maksimum = 50. Setelah proses pembelajaran selesai, proses pengujian berlanjut pada 3 set data. Hasil pengujian adalah nilai prediksi atau estimasi dari set data yang diuji. Setelah kalkulasi, data yang diperoleh diperkirakan set daya dan tegangan untuk bus yang tidak terpasang *PMU*.

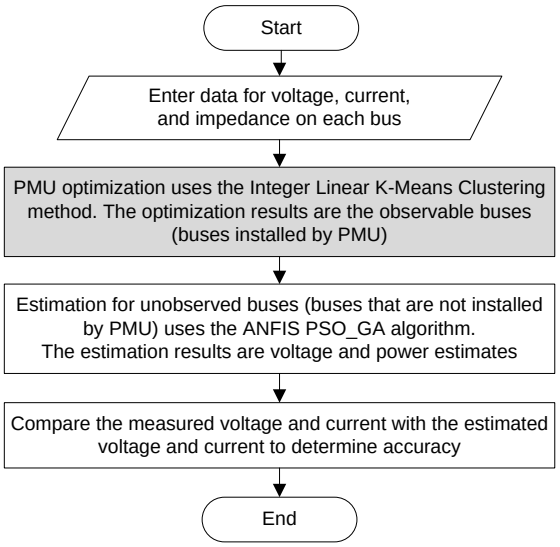
4.4 Algoritma Implementasi modifikasi *ANFIS* dengan *Hybrid PSO-GA* pada jaringan Distribusi di Bendul-Merisi

Untuk implementasi algoritma *ANFIS*, penelitian [33] telah dilakukan untuk mengoptimalkan penempatan *PMU* menggunakan metode *Integer Linear K-Means* yang diterapkan pada jaringan distribusi Bendul Merisi. Jaringan distribusi ini dibagi menjadi 3 *cluster* dengan 1 *PMU* di setiap *cluster*. Penentuan bus yang diamati dan bus yang tidak teramati telah dibahas dalam bagian 4.3.

Dalam studi ini, estimasi daya dan tegangan untuk bus yang tidak terpasang *PMU*. Urutan langkah-langkah ditunjukkan pada Gambar 4. 8. Kotak abu-abu menunjukkan studi sebelumnya yang tidak sepenuhnya dibahas dalam bagian ini.

Data yang digunakan untuk studi estimasi daya dan tegangan menggunakan algoritma modifikasi *ANFIS* dengan *hybrid PSO-GA* yang dimodifikasi diambil dari Perusahaan Listrik Negara di Bendul Merisi Surabaya Selatan-Indonesia.

Pada penelitian ini telah dikumpulkan data tentang pengukuran tegangan dan arus pada 11 bus dalam jaringan. Data diambil selama 10 hari, dan pengukuran harian dilakukan setiap jam dari pukul 00:00 hingga 23:00.



Gambar 4.8 Diagram Alur Penelitian

Penelitian yang dijelaskan merupakan kelanjutan dari penelitian tentang mengoptimalkan penempatan *PMU* untuk menentukan bus mana yang terpasang *PMU* dan bus mana yang merupakan bus yang tidak terpasang *PMU* yang

sesuai dengan jaringan distribusi Bendul Merisi. Hasilnya telah dijelaskan dan ditunjukkan pada Tabel 4.1. Bagian ini menjelaskan hasil perkiraan daya dan tegangan untuk bus yang tidak dipasang *PMU*. Algoritma estimasi yang digunakan adalah algoritma *ANFIS* dengan *hybrid PSO-GA* yang dimodifikasi.

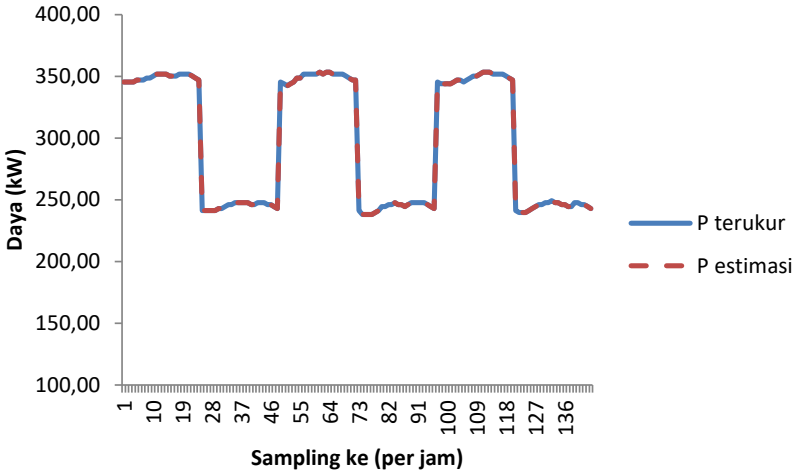
Data yang dikumpulkan adalah data tegangan (V) dan arus (I) di semua bus, yang diambil sampelnya per jam, per hari sampai 24 kali sehari selama 10 hari. Dari pengumpulan data, 10 set data diperoleh. Satu set data terdiri dari data tegangan dan arus dari 10 bus yaitu bus 2 hingga bus 11, selama 1 hari pengumpulan data. Dari 10 set data, 7 set data hari pertama hingga hari ke 7, digunakan untuk proses pembelajaran. Kumpulan data dari 8 hingga 10 digunakan untuk proses pengujian, dengan merujuk pada hasil proses pembelajaran *ANFIS*. Hasilnya ditunjukkan dalam Tabel 4.2 dan Gambar 4.9.a serta 4.9.b.

Tabel 4.2 Perbandingan pengukuran dan estimasi daya dan tegangan *cluster*1

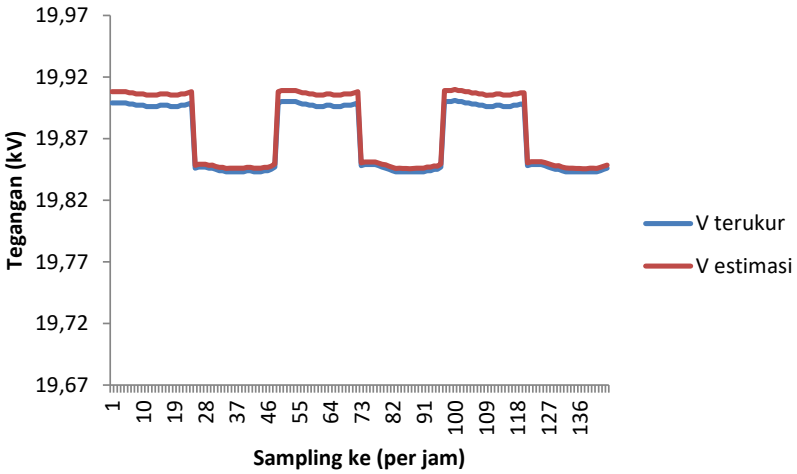
Cluster 1 (*Unobserved* nomor bus 2 dan 4)				
Jam	Daya P (kW)		Tegangan V (kV)	
	P - terukur	P - estimasi	V - terukur	V - estimasi
00:00	345,45	345,43	19,90	19,91
01:00	347,04	347,02	19,90	19,91
02:00	347,02	347,01	19,90	19,91
:	:	:	:	:
21:00	350,19	350,17	19,90	19,91
22:00	348,61	348,60	19,90	19,91
23:00	347,04	347,02	19,90	19,91
00:00	241,33	241,33	19,85	19,85
01:00	241,34	241,34	19,85	19,85
02:00	241,34	241,34	19,85	19,85
:	:	:	:	:
21:00	246,07	246,07	19,84	19,85
22:00	244,50	244,49	19,85	19,85
23:00	242,93	242,93	19,85	19,85

Tabel 4.2 menunjukkan perbandingan antara daya pengukuran terhadap daya yang diperkirakan dan tegangan pengukuran terhadap tegangan yang diperkirakan untuk *cluster* 1. Dalam *cluster* ini, bus yang diuji adalah bus 2 dan 4 (bus tidak teramati). Mempertimbangkan ukuran tabel, tabel ini hanya

menampilkan kumpulan data 1 hari sebanyak 24 data pengukuran dan estimasi. Perbandingan set data daya dan tegangan lengkap selama 3 hari untuk *cluster*1, disajikan menggunakan grafik pada Gambar 4.9.a dan 4.9.b.



Gambar 4.9.a Perbandingan hasil estimasi dan pengukuran daya di *cluster* 1



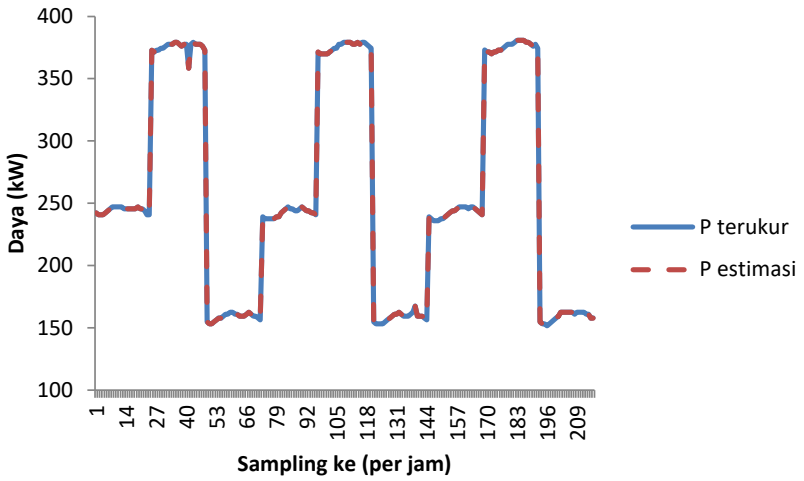
Gambar 4.9.b Perbandingan estimasi dan pengukuran tegangan di *cluster* 1

Tabel 4.3 menunjukkan perbandingan antara daya pengukuran terhadap daya yang diperkirakan dan tegangan pengukuran terhadap tegangan yang diperkirakan untuk *cluster* 2. Dalam *cluster* ini, bus yang diuji adalah bus 5, 7, dan 8 (bus tidak teramati).

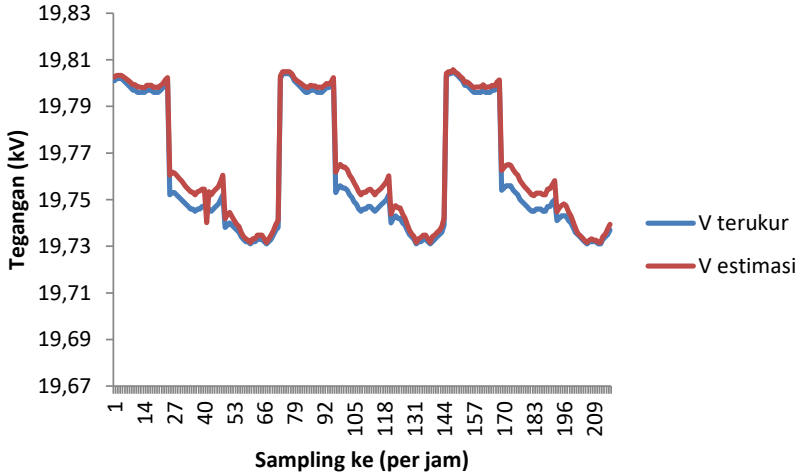
Tabel 4.3 Perbandingan pengukuran dan estimasi daya dan tegangan *Cluster* 2

Cluster 2 (*Unobserved* nomor bus 5, 7 dan 8)				
Jam	Daya P (kW)		Tegangan V (kV)	
	P - terukur	P - estimasi	V - terukur	V - estimasi
00:00	240,79	240,79	19,80	19,80
01:00	240,79	240,79	19,80	19,80
02:00	242,36	242,37	19,80	19,80
:	:	:	:	:
21:00	243,91	243,91	19,80	19,80
22:00	240,77	240,77	19,80	19,80
23:00	240,78	240,78	19,80	19,80
00:00	372,92	372,94	19,75	19,76
01:00	371,36	371,38	19,75	19,76
02:00	372,94	372,96	19,75	19,76
:	:	:	:	:
21:00	377,58	377,61	19,75	19,76
22:00	376,04	376,06	19,75	19,76
23:00	372,92	372,94	19,75	19,76
00:00	154,75	154,73	19,74	19,74
01:00	153,18	153,16	19,74	19,74
02:00	153,18	153,17	19,74	19,75
:	:	:	:	
21:00	159,46	159,45	19,74	19,74
22:00	157,90	157,89	19,74	19,74
23:00	156,33	156,31	19,74	19,74

Perbandingan set data daya dan tegangan lengkap selama 3 hari untuk *Cluster* 2, disajikan menggunakan grafik pada Gambar 4.10.a dan 4.10.b.



Gambar 4.10.a. Perbandingan hasil estimasi dan pengukuran daya di *Cluster* 2

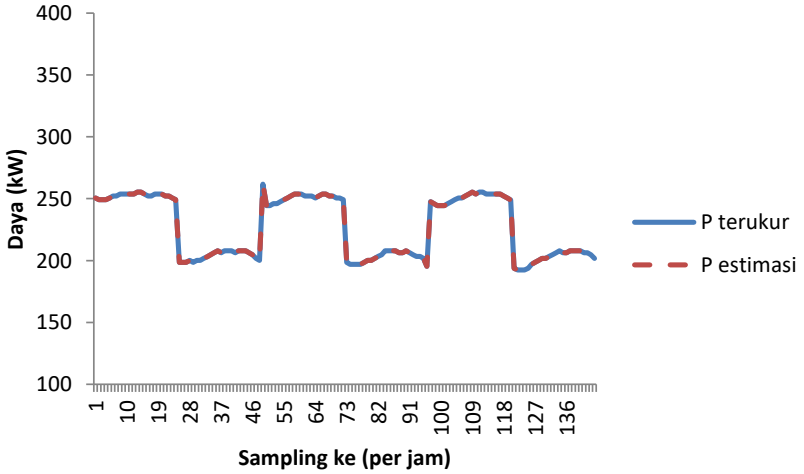


Gambar 4.10.b. Perbandingan hasil estimasi dan pengukuran tegangan di*cluster* 2

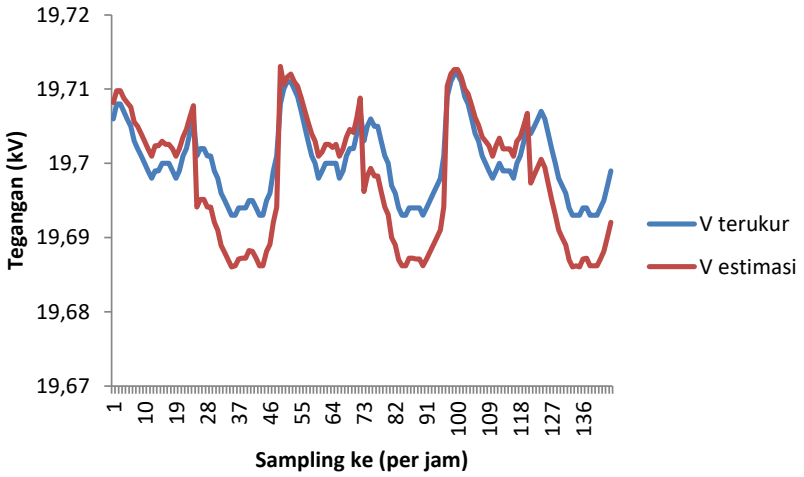
Tabel 4.4 menunjukkan perbandingan antara daya pengukuran terhadap estimasi dan tegangan yang terukur maupun perkiraan untuk *cluster* 3. Dalam *cluster* ini, bus yang diuji adalah bus 10 dan 11 (bus tidak ada *PMU*). Perbandingan set data daya dan tegangan lengkap selama 3 hari untuk *cluster* 3, disajikan dengan menggunakan grafik pada Gambar 4.11.a dan 4.11.b.

Tabel 4.4 Perbandingan pengukuran dan estimasi daya dan tegangan *Cluster*3

Cluster 3 (*Unobserved* nomor bus 10 dan 11)				
Jam	Daya P (kW)		Tegangan V (kV)	
	P - terukur	P - estimasi	V - terukur	V - estimasi
00:00	250,66	250,67	19,71	19,71
01:00	249,11	249,11	19,71	19,71
02:00	249,11	249,11	19,71	19,71
:				
21:00	252,19	252,19	19,70	19,71
22:00	250,64	250,63	19,70	19,71
23:00	249,08	249,09	19,71	19,71
00:00	198,59	198,58	19,70	19,70
01:00	198,60	198,59	19,70	19,70
02:00	198,60	198,59	19,70	19,70
:				
21:00	204,84	204,83	19,70	19,69
22:00	201,72	201,71	19,70	19,69
23:00	200,16	200,16	19,70	19,69



Gambar 4.11. a . Perbandingan hasil pengukuran dan estimasi daya di *Cluster* 3



Gambar 4.11. b Perbandingan hasil pengukuran dan estimasi tegangan *Cluster* 3

Penelitian serupa dari estimasi tegangan untuk jaringan distribusi radial menggunakan *ANFIS PSO-GA*, diperoleh untuk fungsi perkembangan metode yang diusulkan [24]. Penelitian metode *ANFIS hybrid PSO-GA* lebih direkomendasikan dengan akurasi rata-rata 99,863 % seperti yang ditunjukkan pada Tabel 4.5.

Algoritma ini telah menunjukkan kinerja yang baik dengan mendapatkan akurasi yang tinggi . Dengan tegangan input dan set data saat ini pada setiap bus di setiap *cluster*, nilai estimasi daya dan tegangan dalam bus yang tidak teramati dapat diperoleh.

Tabel 4.5 Akurasi hasil simulasi dengan *ANFIS* Hybrid *PSO-GA*

P/V	*Cluster*			Akurasi
	No	*Observed*	*Unobserved*	(%)
Daya (W)	1	Bus 3	Bus 2 and 4	99.976
	2	Bus 6	Bus 5, 7, and 8	99.954
	3	Bus 9	Bus 10 and 11	99.975
	Rata-rata accuracy 3 *cluster*			99.968
Tegangan (V)	1	Bus 3	Bus 2 and 4	99.704
	2	Bus 6	Bus 5, 7, and 8	99.804
	3	Bus 9	Bus 10 and 11	99.766
	Rata-rata 3 *cluster*			99.758

Hasil perbandingan antara daya yang diukur dengan daya yang diestimasi, nilai akurasi yang diperoleh di setiap *cluster* 1, 2, dan 3 sama dengan 99,976 %, 99,954 %, dan 99,975 %. Hasil rata – rata *error* untuk estimasi daya sebesar 0,233%.

Hasil perbandingan antara tegangan yang diukur dengan tegangan yang diestimasi, diperoleh nilai akurasi masing-masing dalam *cluster* 1, 2, dan 3 adalah sama dengan 99, 704 %, 99,604 %, dan 99,766 %. *Error* rata –rata untuk estimasi tegangan sebesar 0,232%.

(Publikasi pada jurnal internasional ***I.R.E.E*** quartile Q2 Scopus).

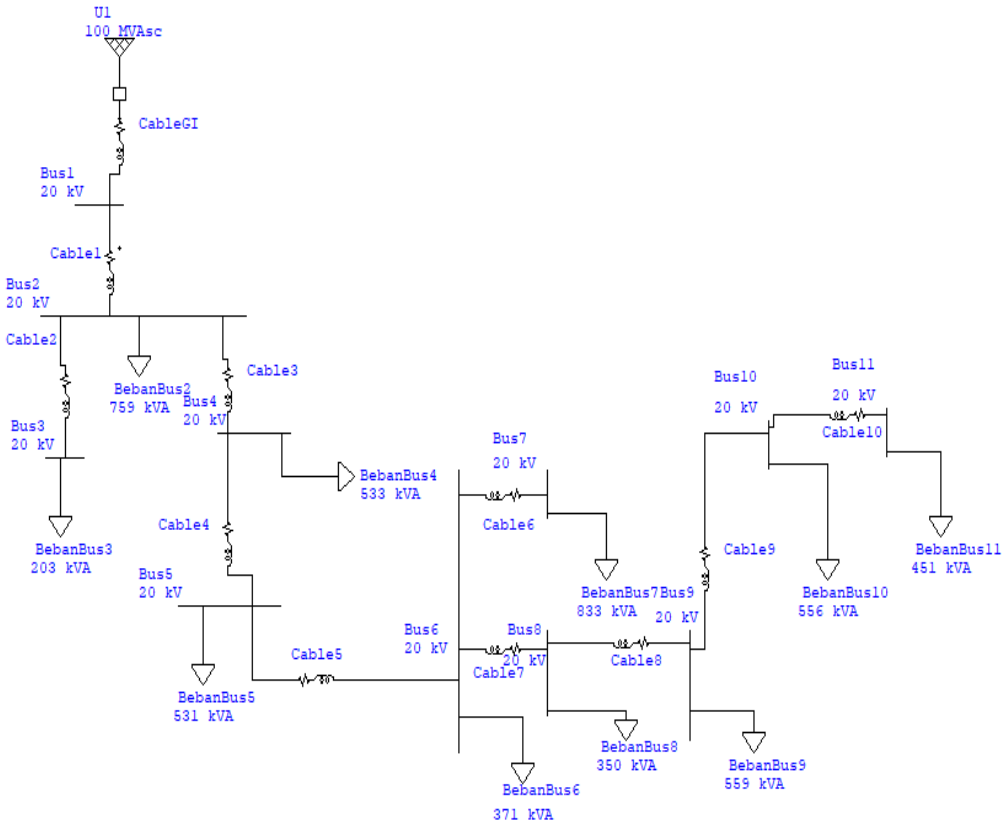
BAB 5 VERIFIKASI HASIL ESTIMASI PADA JARINGAN DISTRIBUSI SURABAYA SELATAN DI BENDUL MERISI

Pada bab ini akan dijelaskan beberapa hal terkait dengan simulasi hibrid data *offline* dan *online* dengan menggunakan *software Simulink* yang tersedia di Matlab, untuk mensimulasikan daya aktif dan daya reaktif pada sistem jaringan di Bendul Merisi yang dapat di monitoring secara *realtime* menggunakan *Internet of Think (IoT)* .

Pada penelitian ini bertujuan untuk membuat suatu program validasi bagaimana penempatan *PMU* yang optimal dalam sistem distribusi Surabaya Selatan di Bendul Merisi. Kemudian hasilnya dapat di monitoring menggunakan *smartphone* secara *realtime.* Metode *Integer Linear K-Means Clustering* didapatkan dari data PLN lalu diolah dengan menggunakan metode tersebut hingga ditemukan bagaimana penempatan *PMU* secara optimal yang di bahas pada bab 3. Selanjutnya di estimasi daya dan tegangan dengan modifikasi *ANFIS* dengan *hybrid PSO-GA yang dibahas* pada bab 4.

5.1 Analisa Simulasi Aliran Daya

Menurut data di Bendul Merisi di dapatkan *single line* diagram seperti pada Gambar 5.1 maka analisa penelitian dari simulasi ini akan diawali dengan menggunakan *software)* untuk menganalisa aliran daya pada jaringan distribusi Bendul Merisi, berikut hasil *running* di *software.* Dari Gambar 5.1 *single line* hasil aliran daya dari data awal yang di dapatkan dengan kondisi dimana tegangan di asumsukan dalam keadaan seimbang dimana tanpa adanya beban maka tegangannya yang digunakan yakni sebesar 20 kV dan Cos φ=0.8 (beban setimbang). Berikut data berupa nilai saluran impedansi dari jaringan di Bendul Merisi. Selanjutnya yaitu data beban puncak beserta saluran impedansi yang di analisa pertama adalah data beban listrik dan saluran impedansi yang akan dianalisa dan disimulasikan untuk di ketahui daya aktif (P) dan daya reaktif (Q) di setiap busnya dan hasilnya di tunjukkan pada Tabel 5.1.



Gambar 5.1 *Single line* diagram di Bendul Merisi dengan *Simulasi*

Tabel 5.1 Tabel nilai beban puncak berdasarkan pengukuran di Bendul Merisi

Bus m ke n	Impedansi	S (kW)	V (Volt)	I (Ampere)	Cos φ	Sin φ
Z1 ke Z2	0.07861+0.02931i	745	19913	21.4	0.8	0.6
Z2 ke Z3	0.05553+0.02071i	208	19910	6	0.8	0.6
Z2ke Z4	0.19424+0.07244i	496	19869	14.2	0.8	0.6
Z4 ke Z5	0.09788+0.03650i	483	19832	13.8	0.8	0.6
Z5 ke Z6	0.23836+0.08889i	319	19801	9.1	0.8	0.6
Z6 ke Z7	0.39681+0.14724i	689	19793	19.7	0.8	0.6
Z6 ke Z8	0.2776+0.10355i	290	19783	8.3	0.8	0.6
Z8 ke Z9	0.04869+0.01816i	455	19768	13	0.8	0.6
Z9 ke Z 10	0.13845+0.05163	350	19760	10	0.8	0.6
Z10 ke Z11	0.27005+0.10071i	363	19755	10.4	0.8	0.6

Pada Tabel 5.1 terdapat nilai sin φ sebesar 0.6 . Nilai sin φ tersebut didapatkan dengan langkah – langkah perhitungan seperti di bawah ini :

Cos φ = 0.8

φ = Arccos 0.8 = 36.87°

Sin φ = Sin 36.87° = 0.6

5.2 Perancangan Sistem *Hardware* dan *Software)*

Pada penelitian ini akan digunakan perangkat keras (*Hardware)* dan juga akan digunakan perangkat lunak (*software)*). Spesifikasi perangkat keras yang akan digunakan untuk membuat sistem simulasi ini adalah sebagai berikut :

1. Sony Vaio, Intel Core i5
2. RAM 4 Gb
3. Hard disk 500 Gb
4. Keyboard, mouse, dan touchpad
5. Mikrokontroller Arduino Mega 2560 dengan total jumlah input dan *output* sebanyak 54 buah dilengkapi dengan *clock speed* sebesar 16 Mhz.

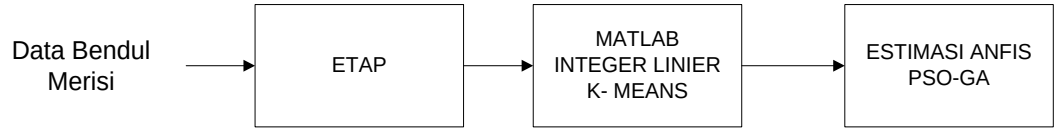
Spesifikasi dari perangkat lunak (*software)* yang digunakan untuk membuat sistem ini yaitu sebagai berikut :

1. Sistem Operasi Windows 10.0 Enterprise 64 bit
2. Matlab
3. *Simulasi*

Sedangkan, untuk minimum requirement untuk dapat menggunakan dan menjalan sistem ini yaknni haruslah sudah melakukan instalasi *software* Matlab dan juga support packages untuk Arduino mega 2560 pada *software* Matlab tersebut dan juga pada perangkat yang akan digunakan.

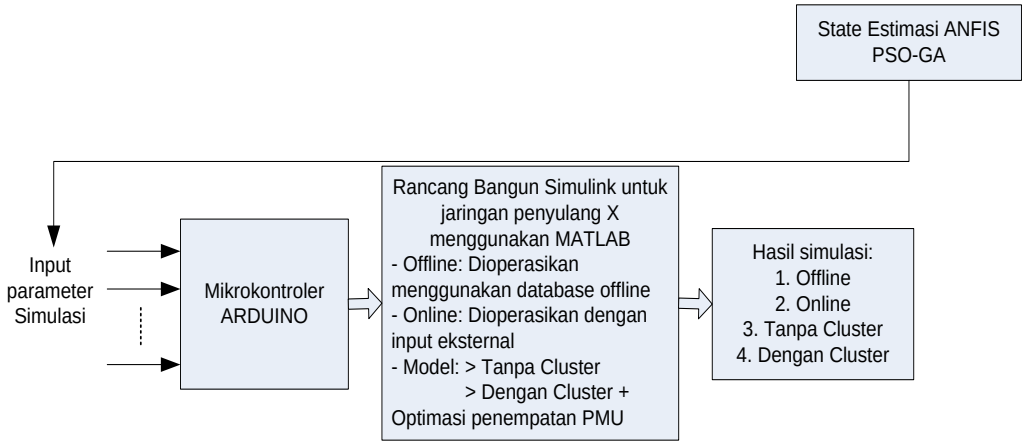
5.2.1 Blok Diagram *Hybrid* Sistem *Online dan Offline*

Tahap pertama mengumpulkan data pada jaringan distribusi Bendul Merisi. Kemudian analisa aliran daya menggunakan simulasi. Tujuannya untuk mengetahui besarnya V, I, P, Q dan PF. Selanjutnya *cluster* dengan menggunakan himpunan data dari simulasi, untuk mendapatkan titik optimum lokasi pemasangan sensor dan jumlah minimum sensor menggunakan teknik *Integer Linear K-Means Clustering* . Proses tahap pengujian *hardware* dapat di lihat pada Gambar 5.2.



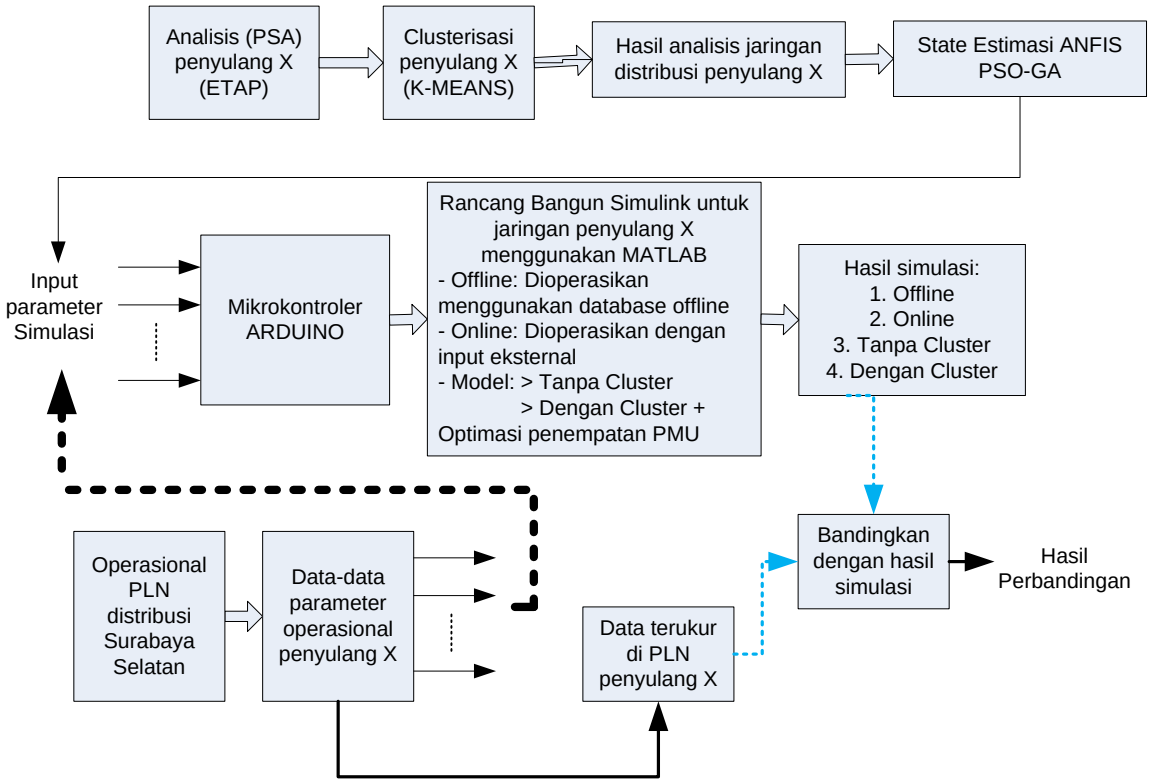
Gambar 5.2 Proses Tahap Awal Pengujian *Hardware*

Tahap kedua, mendesain distribusi Bendul Merisi menggunakan *simulink* dengan program matlab. Hasil dari desain *simulink* akan di gunakan sebagai simulator *online* jaringan distribusi Surabaya Selatan. Pada desain *simulink* di tempatkan sensor pada lokasi yang sudah di tentukan dan disediakan beberapa saluran input. Selanjutnya mensimulasikan data distribusi dengan mengubah beberapa nilai pada titik-titik *trigger*. Nilai yang di*trigger* dari luar berupa beban *(load)*. Beban disimulasikan menggunakan rangkaian resistor, rangkaian resistor akan mengirimkan tegangan pada mikrokontroller yang akan mengubah menjadi data digital dan akan dikirim melalui saluran usb ke komputer. Data digital tersebut merupakan nilai input beban pada simulator. Input beban simulator ada 10 titik. Karena banyak input beban yang disimulasikan, maka mikrokontroller berfungsi sebagai *multiplexer* yang akan mengirimkan data-data input beban secara bergantian ke simulator. Selanjutnya dilakukan *training* terhadap semua dari masing-masing bus dengan *Integer Linear K-Means* secara *offline* kemudian dilakukan *testing* dengan sampling beberapa bus saja secara *online* dari data eksternal. Proses tahap ke dua pengujian *hardware* dapat di lihat pada Gambar 5.3.



Gambar 5.3 Tahap Dua Pengujian *Hardware*

Tahap ketiga adalah proses simulasi, di monitoring nilai tegangan magnitude, arus dan daya pada *output* sensor-sensor tersebut. Simulasi analisa sistem tenaga *semi realtime* ini dibandingkan perubahan daya hasil di komputer dengan kondisi riil di lapangan. Tahap pengujian *hardware* secara keseluruhan dapat dilihat pada Gambar 5.4



Gambar 5.4 *Simulink On Line* dan *Off line Monitoring*

5.2.2 Diagram Alir Pengujian dan Perancangan sistem

Pengujian dan perancangan data yang telah didapatkan yaitu dengan menggunakan *simulasi* (*software)*. Perancangan ini dapat diketahui besar nilai daya aktif, daya reaktif, arus beserta tegangan yang mengalir pada setiap bus yang telah ditentukan berdasarkan *single line* diagram dari Jaringan Bendul Merisi.

Langkah pengujian ditunjukkan pada diagram alir pada Gambar 5.5.

Tahapan – tahapan analisa sebagai berikut :

Langkah 1: Merangkai single line diagram di *software* lengkap dengan saluran impedansi dan beban.

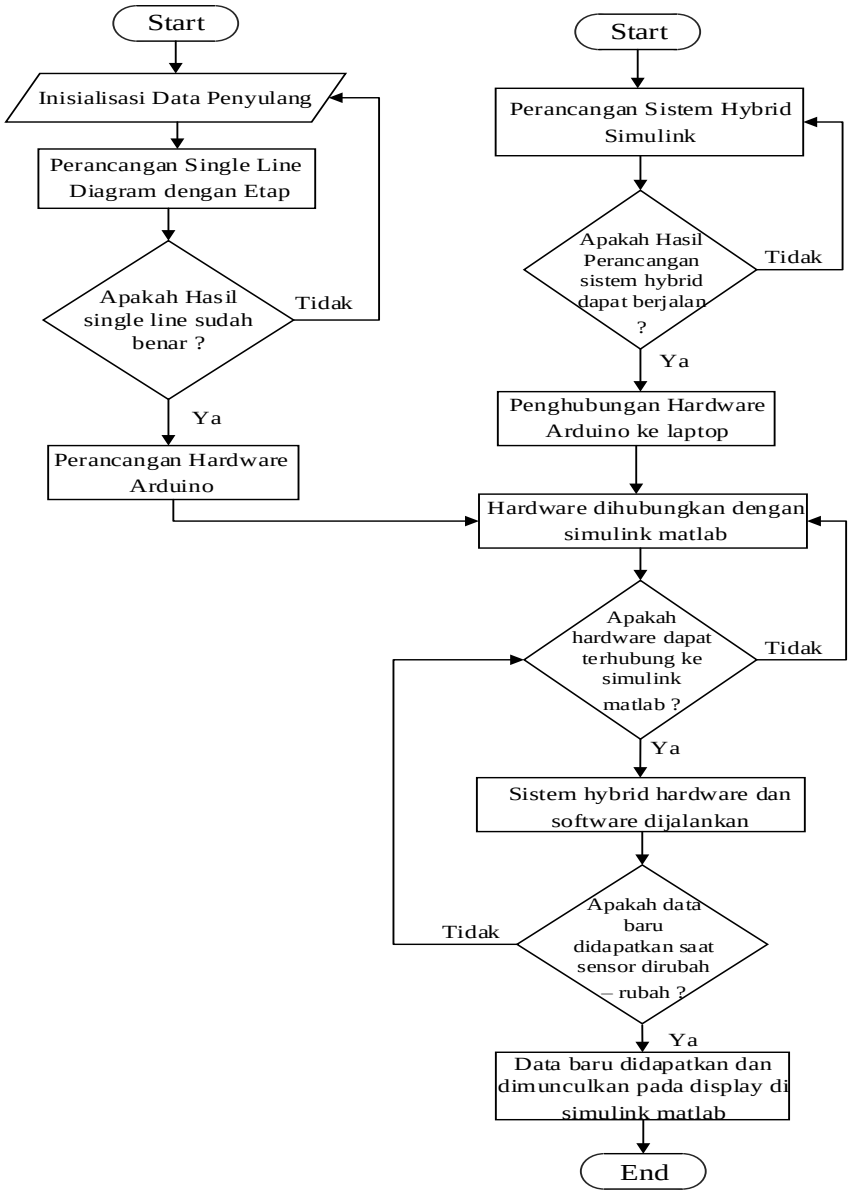
Langkah 2: Mencatat data tegangan,arus, daya aktif, dan daya reaktif yang didapatkan dari hasil *running software.*

Langkah 3: Melakukan perhitungan menggunakan rumus daya listrik untuk mencari daya aktif (P) dan daya reaktif (Q) menggunakan data-data yang dihasilkan dari langkah 1 dan langkah 2.

Langkah 4: Menghitung total daya listrik yang di dapatkan dari langkah 3 dengan perhitungan total daya listrik setiap *cluster*.

Langkah 5: Selanjutnya dilakukan perhitungan drop tegangan yang terjadi pada sistem distribusi Bendul Merisi dengan data beban dan nilai impedansi sesuai dengan langkah 2.

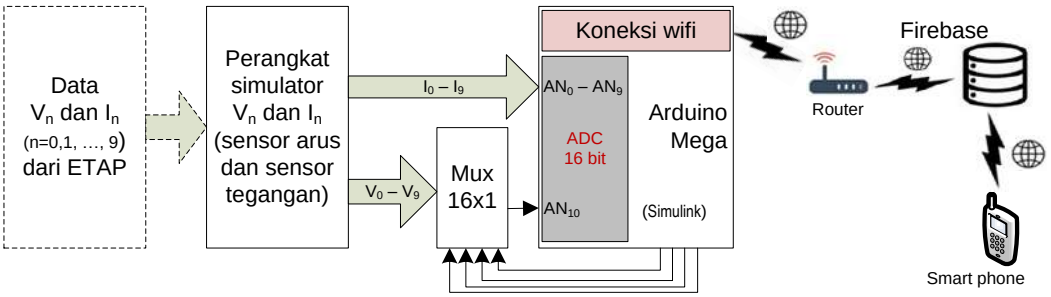
Langkah 6: Mensimulasikan dan estimasi penggunaan simulasi *simulink* yang telah dibuat untuk mengetahui daya listrik di setiap jam pada tiap bus dari data terbaru yang telah di dapat selama 6 jam.



Gambar 5.5 Diagram alir perancangan *hardware* dan pengujian sistem

5.3 Perancangan *Software Simulink*

Perancangan dan pembuatan sistem hybrid *offline* dan *online* pada jaringan distribusi Bendul Merisi di tunjukkan pada Gambar 5.6.



Gambar 5.6 Tampilan proses simulasi *Simulink*

Data tegangan (Vn) dan arus (In) diambil dari simulasi untuk jaringan distribusi Bendul Merisi dengan 10 bus. Data ini disimulasikan menggunakan 10 sensor tegangan (Vn) berupa resistor variable dan 10 sensor arus (In). Nilai tegangan dan arus dapat diubah-ubah dengan mengubah nilai resistansi pada resistor variable. Saluran-saluran arus listrik (I_0-I_9) dihubungkan ke pin AN_0 – AN_9 di blok ADC 16 bit pada Arduino Mega dan saluran-saluran tegangan listrik (V_0-V_9) dihubungkan ke pin AN10 di blok ADC 16 bit setelah melalui *multiplexer* 16x1 bit.

Data tegangan (V_0-V_9) dan arus (I_0-I_9) yang masuk di Arduino diproses menggunakan program yang menyerupai proses pada *Simulink* untuk mendapatkan nilai daya aktif dan daya reaktif. Data-data tegangan, arus dan daya dikirimkan ke internet menggunakan *wifi* dan *platform IoT firebase*. Dengan menggunakan platform ini maka data-data tersebut dapat diakses menggunakan *smartphone* dari mana saja.

Dalam simulasi, nilai tegangan dan arus dapat diubah-ubah dengan mengubah nilai resistor variable, sehingga secara *real time* data-data tegangan, arus dan daya pada *smartphone* juga berubah.

5.3.1 Data Pengujian Simulasi pada Sistem

Untuk mengetahui sistem simulasi telah berhasil atau tidak, maka simulasi sistem akan diuji untuk menghitung daya aktif (P) dan daya reaktif (Q) pada sistem distribusi Bendul Merisi. Pengujian ini dilakukan dengan menghitung nilai daya aktif dan daya reaktif dari data yang telah didapat yaitu data tegangan dan arus listrik selama 6 jam, dan tujuan simulasi ini yaitu untuk menghitung nilai daya aktif dan daya reaktif tiap jam dari data yang telah didapatkan. Tabel 5.2 dan

Tabel 5.3 data tegangan dan arus yang akan dijadikan bahan pengujian. Hasil simulasi terdapat pada Tabel 5.3 di lampiran 1.

Tabel 5.2 Tabel data tegangan dan arus jam 8:00 – 10:00

Bus	Waktu								
	8:00			9:00			10:00		
	kVA	V	I	kVA	V	I	kVA	V	I
2	762	19898	21.9	765	19898	22	766	19897	22
3	208	19895	6	208	19895	6	209	19895	6
4	539	19845	15.4	539	19845	15.4	541	19844	15.5
5	539	19799	15.4	539	19799	15.4	540	19798	15.4
6	375	19760	10.7	375	19759	10.7	377	19759	10.8
7	834	19750	23.8	837	19749	23.9	836	19748	23.8
8	349	19736	9.9	349	19735	9.9	353	19734	10.1
9	549	19716	15.6	551	19716	15.7	554	19715	15.8
10	558	19704	15.9	563	19703	16	567	19702	16.1
11	458	19698	13	457	19697	13	457	19696	13

Tabel 5.3 Tabel data tegangan dan arus jam 11:00 – 13:00

Bus	Waktu								
	11:00			12:00			13:00		
	kVA	V	I	kVA	V	I	KVA	V	I
2	768	19897	22.1	768	19896	22.1	770	19896	22.1
3	213	19894	6.1	215	19894	6.2	213	19894	6.1
4	542	19843	15.5	545	19843	15.6	545	19843	15.6
5	543	19797	15.5	543	19796	15.5	545	19797	15.6
6	379	19757	10.8	380	19756	10.8	379	19757	10.8
7	837	19747	23.9	840	19746	23.9	842	19746	24
8	356	19733	10.1	358	19732	10.2	355	19733	10.1
9	565	19713	16.1	567	19712	16.1	565	19713	16.1
10	567	19700	16.1	568	19699	16.2	565	19700	16.1
11	460	19694	13.1	462	19693	13.1	460	19694	13.1

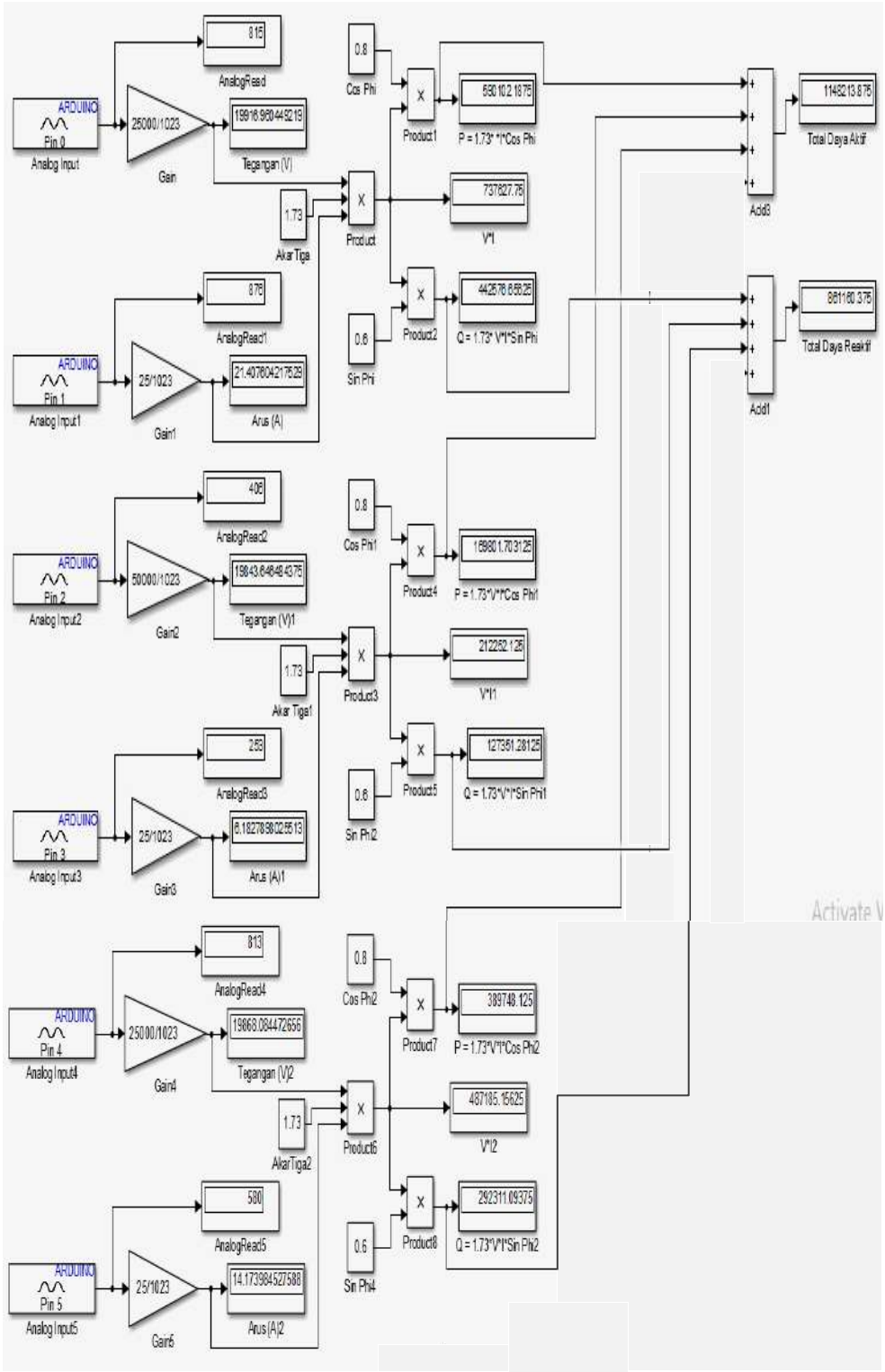
5.3.2 Pengujian Simulasi pada Sistem

Setelah perhitungan total daya aktif dan total daya reaktif di setiap *cluster* maka langkah kemudian yaitu menerapkan sistem simulasi yang telah dibuat pada *simulink* matlab untuk mencari nilai daya aktif dan daya reaktif di setiap bus dan total daya aktif dan daya reaktif di masing – masing *cluster*. Berikut hasil *running* simulasi sistemnya pada Gambar 5.7 dan 5.8.

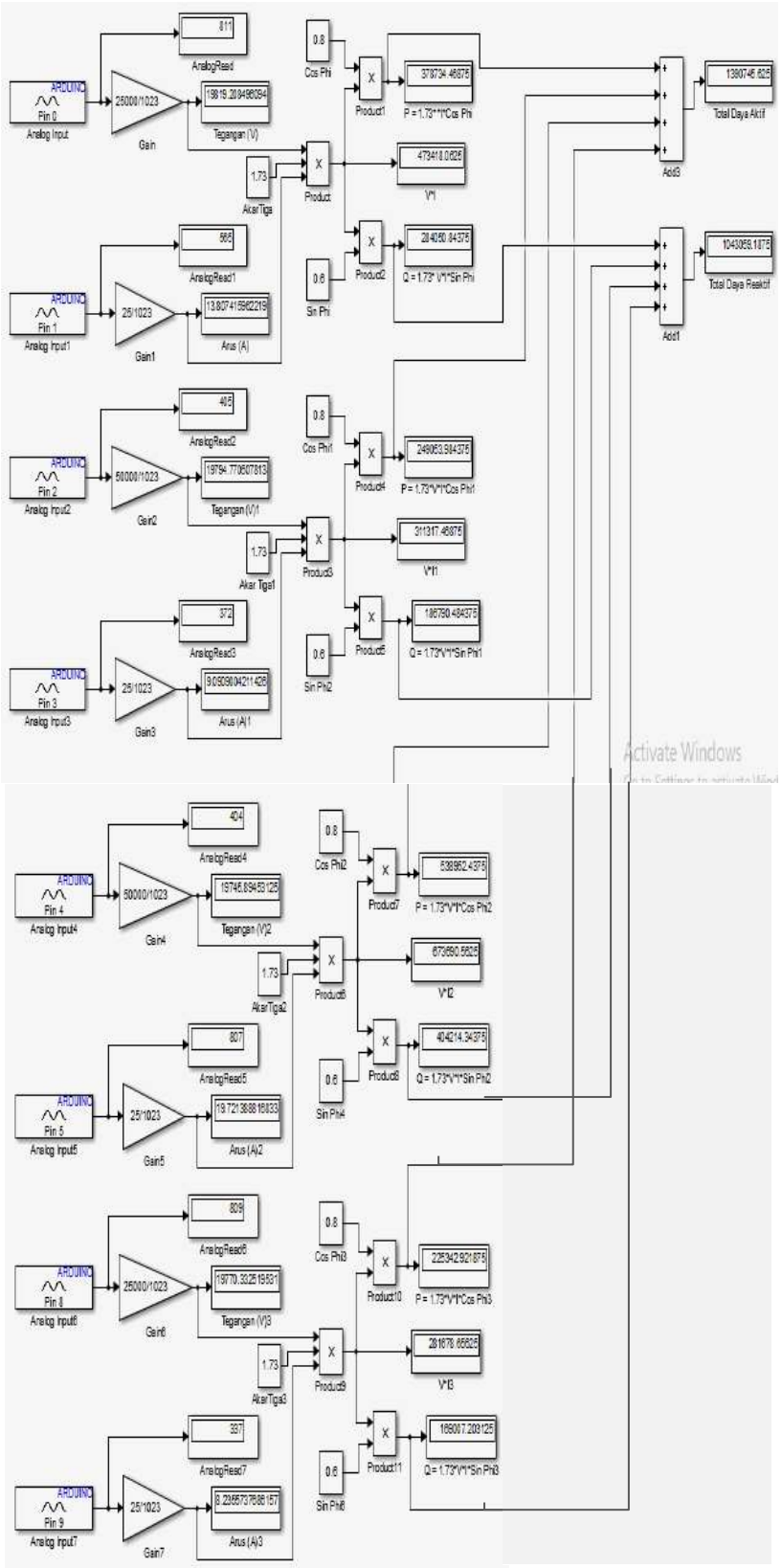
Pada Gambar 5.7 Analog input pada Pin_0 dan Pin_1 merupakan saluran input untuk tegangan dan arus pada bus 2. Analog input pada Pin_2 dan Pin_3

merupakan saluran input untuk tegangan dan arus pada bus 3. Sedangkan untuk analog input pada Pin_4 dan Pin_5 merupakan saluran input untuk tegangan dan Arus pada bus 4. Analog input pada Pin_0 (tegangan) dikalikan dengan faktor penguatan sebesar 25000/1023. Sedangkan analog input Pin_1 (arus) dikalikan dengan faktor nilai 25/1023. Kemudian setelah dikalikan faktor masing-masing, maka kedua nilai tegangan dan arus ini dikalikan dengan $\sqrt{3}$=1,73 karena jaringan 3 fasa. *Output* dari perkalian ini dikalikan dengan cos Phi = 0,8 untuk mendapatkan nilai gaya aktif, dan dikalikan dengan sin Phi = 0,6 untuk mendapatkan daya reaktif. Dengan demikian daya aktif P_2 dan daya reaktif Q_2 untuk bus 2 didapatkan nilainya. Adapun tegangan dan arus input pada bus 3 melalui Pin_2 dan Pin_3 diproses dengan cara yang sama sehingga didapat nilai untuk daya aktif P_3 dan daya reaktif Q_3 untuk bus 3. Begitu pula untuk tegangan dan arus input pada bus 4 melalui Pin_4 dan Pin_5, setelah diproses dengan cara yang sama akan didapatkan nilai daya aktif P_4 dan daya reaktif Q_4 untuk bus 4. Penambahan (*sum*) dari ketiga daya aktif (P_2, P_3 dan P_4) dan ketiga daya aktif (Q_2, Q_3 dan Q_4) didapat total daya aktif P_I dan daya reaktif Q_I untuk *cluster* I pada jaringan distribusi.

Pada Gambar 5.8, analog input pada Pin_0 dan Pin_1 merupakan saluran input untuk tegangan dan arus pada bus 5. Analog input pada Pin_2 dan Pin_3 merupakan saluran input untuk tegangan dan arus pada bus 6. Analog input pada Pin_4 dan Pin_5 merupakan saluran input untuk tegangan dan arus pada bus 7. Sedangkan untuk analog input pada Pin_8 dan Pin_9 merupakan saluran input untuk tegangan dan arus pada bus 8. Setelah dikalikan dengan faktor gain untuk tegangan dan arus maka daya aktif (P) dan reaktif (Q) pada masing-masing bus dapat dihitung Pn=V.I. $\sqrt{3}$.cos(Phi).

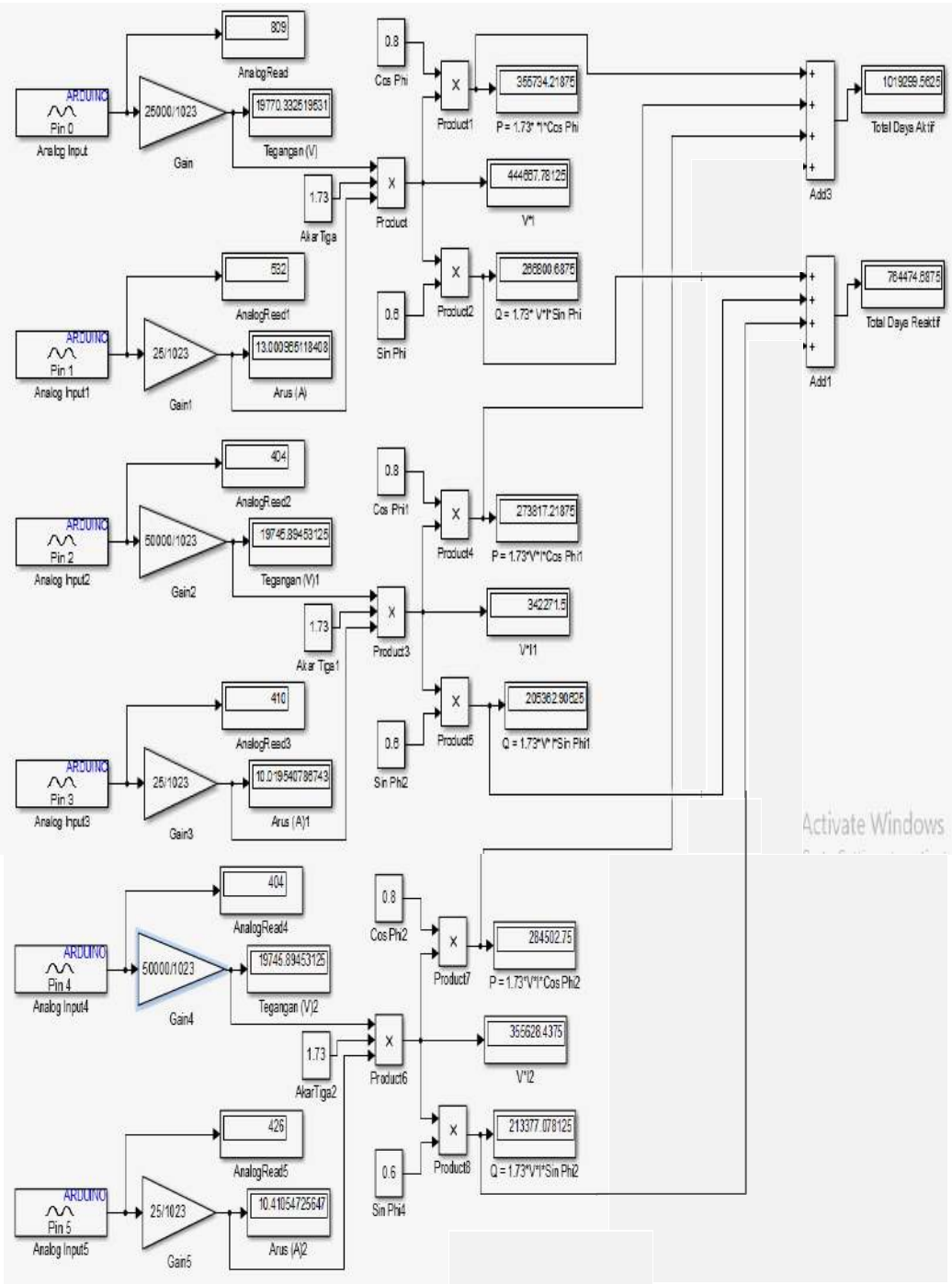


Gambar 5.7 Hasil simulasi untuk *Cluster* I pada bus 2, 3 dan 4



Gambar 5.8 Hasil simulasi untuk *Cluster* II pada bus 5, 6, 7 dan 8

dan Qn=V.I. $\sqrt{3}$.sin(Phi) dengan n=5, 6, 7 dan 8. Penambahan *(sum)* dari keempat daya aktif (P_5, P_6 , P_7 dan P_8) dan keempat daya Reaktif (Q_5, Q_6 , Q_7 dan Q_8) didapat total daya aktif P_{II} dan daya reaktif Q_{II} untuk *cluster* II pada jaringan distribusi.



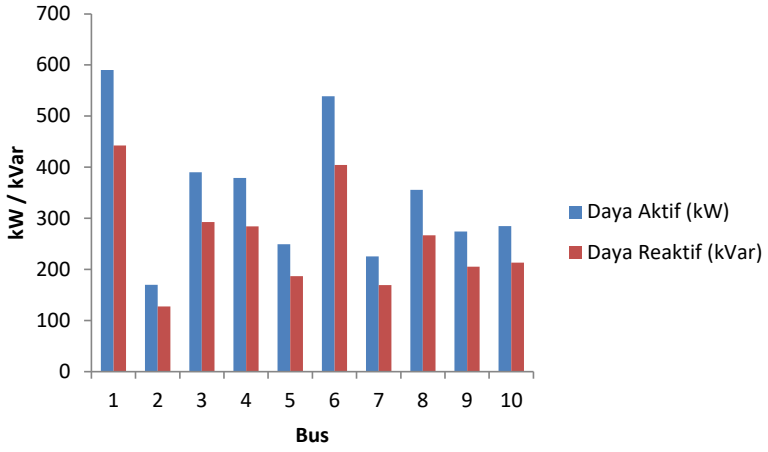
Gambar 5.9 Hasil simulasi untuk *Cluster* III pada bus 9, 10 dan 11

Pada Gambar 5.9 Analog input pada Pin_0 dan Pin_1 merupakan saluran input untuk Tegangan dan Arus pada bus 9. Analog input pada Pin_2 dan Pin_3 merupakan saluran input untuk Tegangan dan Arus pada bus 10. Sedangkan untuk analog input pada Pin_8 dan Pin_9 merupakan saluran input untuk Tegangan dan arus pada bus 11. Setelah dikalikan dengan faktor Gain untuk tegangan dan Arus maka daya aktif (P) dan reaktif (Q) pada masing-masing bus dapat dihitung Pn=V.I. $\sqrt{3}$.cos(Phi) dan Qn=VI. $\sqrt{3}$.sin(Phi) dengan n=9, 10 dan 11. Penambahan (*sum)* dari ketiga daya aktif (P_9, P_{10} dan P_{11}) dan ketiga daya reaktif (Q_9, Q_{10} dan Q_{11}) didapat total daya aktif P_{III} dan daya reaktif Q_{III} untuk *cluster* III pada jaringan distribusi.

Pada Tabel 5.4 berisi hasil perhitungan menggunakan simulasi sehingga menunjukkan nilai daya aktif (P) dan daya reaktif (Q) pada masing-masing bus di setiap *cluster*. Gambar Grafik 5.10 perbandingan daya aktif dan daya reaktif.

Tabel 5.4 Tabel hasil simulasi pada *Simulink* Matlab

No. Bus	Beban (S) (kVA)	Tegangan (kV)	Arus (Ampere)	Daya Aktif (P) (kW)	Daya Reaktif (Q) (kVar)
2	745	19,92	21.410	590,10	442,58
3	208	19,84	6.183	169,80	127,35
4	496	19,87	14.174	389,75	292,31
5	483	19,82	13.810	378,73	284,05
6	319	19,80	9.091	249,05	186,79
7	689	19,75	19.721	538,95	404,21
8	290	19,77	8.235	225,34	169,01
9	455	19,77	13.009	355,73	266,80
10	350	19,75	10.019	273,82	205,36
11	363	19,75	10.410	284,50	213,38



Gambar 5.10 Grafik Perbandingan daya aktif dan daya reaktif

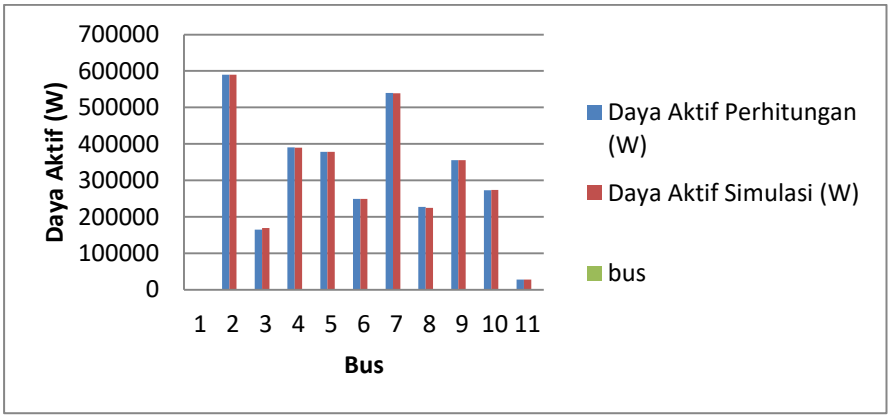
5.3.3 Analisa Akurasi Simulasi pada Sistem

Pada analisa akurasi ini dilakukan agar dapat diketahui tingkat persen *error* dari simulasi yang telah dibuat. Tujuannya apakah simulasi telah berhasil atau tidak. Pada Tabel 5.5 dan 5.6 adalah hasil perhitungan dari *error* antara perhitungan dengan dengan simulasi

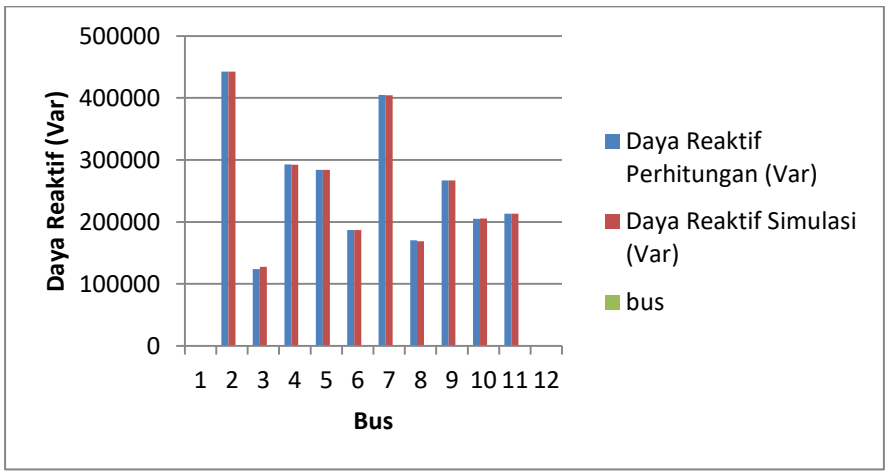
Tabel 5.5 Tabel hasil *error* perbandingan antara perhitungan dengan segi daya dengan simulasi *simulink*

Bus	Beban (kVA)	Perhitungan Segitiga Daya					
		Daya Aktif (kW)		*Error* (%)	Daya Reaktif(QVar)		% *Error*
		Perhitungan	Simulasi		Perhitungan	Simulasi	
2	745	589775	590102.187	0.055	442331	442576.66	0.056
3	208	165333	169801.703	2.703	123999	127351.281	2.703
4	496	390841	389748.125	0.280	292861	292311.094	0.188
5	483	378775	378734.47	0.011	284082	284050.844	0.011
6	319	249382	249053.984	0.132	187036	186790.484	0.131
7	689	539652	538952.437	0.130	404739	404214.344	0.130
8	290	227251	225342.922	0.840	170438	169007.203	0.839
9	455	355666	355734.219	0.019	266749	266800.687	0.019
10	350	273478	273817.219	0.124	205109	205362.906	0.124
11	363	284346	284502.75	0.055	213259	213377.078	0.055
		Error Rata - Rata (%)		0.435	*Error* Rata - Rata (%)		0.426

Hasil perbandingan *error* antara perhitungan segitiga daya dan *simulink* untuk daya aktif (P) dan daya reaktif (Q) di tunjukkan pada Gambar grafik 5.11.a dan 5.11.b.



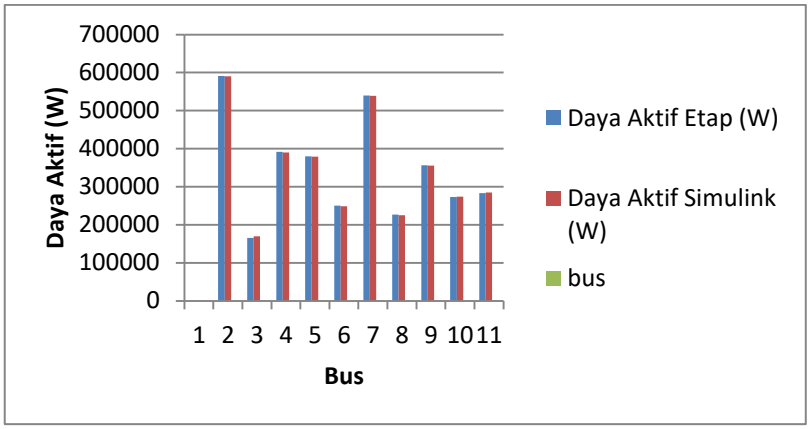
Gambar 5.11. a. Grafik Perbandingan *error* daya aktif



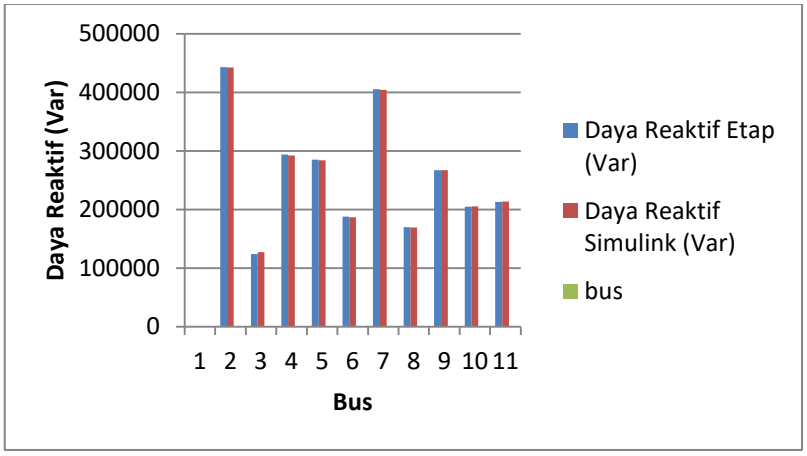
Gambar 5.11. b. Grafik Perbandingan *error* daya reaktif

Tabel 5.6 Perbandingan *error* dengan *Running Etap* dengan *simulink*

Bus	Beban (kVa)	Hasil *Running Etap* dengan *simulink*					
		Daya aktif		*Error* (%)	Daya Reaktif		*Error* (%)
		Etap	*Simulink*		*Etap*	*Simulink*	
2	745	591000	590102.187	0.152	443000	442576.66	0.096
3	208	165000	169801.703	2.910	124000	127351.281	2.703
4	496	392000	389748.125	0.574	294000	292311.094	0.574
5	483	380000	378734.47	0.333	285000	284050.844	0.333
6	319	250000	249053.984	0.378	188000	186790.484	0.643
7	689	540000	538952.437	0.194	405000	404214.344	0.194
8	290	227000	225342.922	0.730	170000	169007.203	0.584
9	455	356000	355734.219	0.075	267000	266800.687	0.075
10	350	273000	273817.219	0.299	205000	205362.906	0.177
11	363	283000	284502.75	0.531	213000	213377.078	0.177
		Error Rata - Rata (%)		0.618	*Error* Rata - Rata (%)		0.556



Gambar 5.12. a. Grafik hasil *error* perbandingan daya aktif antara perhitungan dengan *running Simulasi* dengan *Simulink*

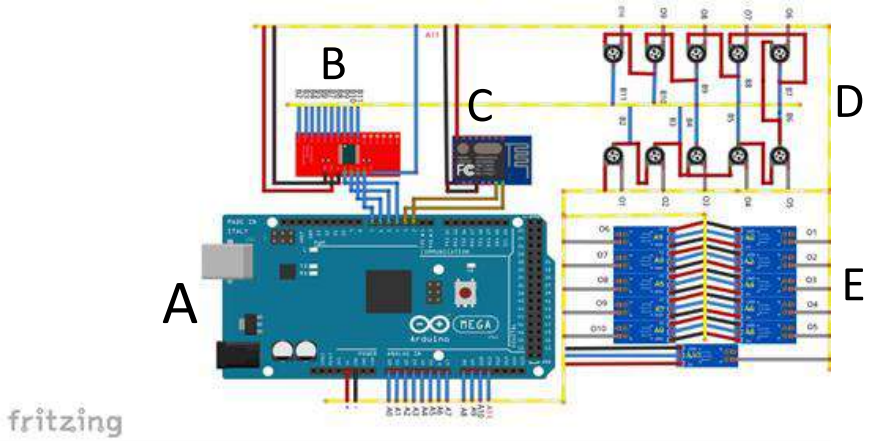


Gambar 5.12.b. Grafik hasil *error* perbandingan daya reaktif antara perhitungan dengan *Simulasi* dengan *Simulink*

Dari Tabel 5.5 dan 5.6 dapat disimpulkan bahwa *error* rata – rata antara perhitungan matematis dengan metode segitiga daya dan simulasi untuk daya aktif sebesar 0.435% dan untuk daya reaktif sebesar 0.426% lalu *error* rata – rata antara hasil *running* simulasi dan hasil simulasi yakni sebesar 0.618% untuk daya aktif dan 0.556% untuk daya reaktif yang di lihat pada Gambar Grafik 5.12.a dan 5.12.b. Ini menunjukkan bahwa sistem simulasi yang telah dibuat berhasil untuk melakukan simulasi untuk mencari nilai daya aktif dan daya reaktif di setiap bus pada jaringan Distribusi Bendul Merisi

5.4 Perancangan *Hardware* dan Monitoring *Internet of Thing (IoT)*

Pada bagian ini membahas dua tahap yaitu desain *Hardware* dan monitoring *Internet of Things (IoT).* Tahap pertama desain *Hardware* di tunjukkan pada Gambar 5.16.



Gambar 5.13 Perancangan *Hardware Internet of Things (IoT)*

Pada blok diagram 5.13 terdapat lima komponen yang memiliki fungsi masing-masing. Komponen tersebut antara lain :

Bagian A: Arduino Mega 2560

Bagian B: Analog *Multiplexer* 16 *channel*

Bagian C: Modul *wifi* ESP8266

Bagian D: Sensor Arus ACS712

Bagian E : Variabel resistor.

Prinsip kerja arduino mega 2560 sebagai pembacaan data sensor arus beserta tegangan, mengontrol selektor *multiplexer* serta mengirim data secara nirkabel melalui perantara wifi. Arduino Mega 2560 akan mengonversi sinyal analog dari 10 sensor ACS712 dan 10 variabel resistor dengan kedalam bentuk digital melalui fungsi ADC yang ada pada mikrokontroller ini. Data ADC dari hasil pembacaan sensor ACS712 akan diubah kedalam bentuk ampere sedangkan variabel resistor akan diubah ke bentuk volt. Dikarenakan jumlah pin ADC pada arduino mega 2560 tidak mencukupi dari total jumlah sensor sehingga di gunakan analog *multiplexer* 16 *channel* sebagai selektor sehingga sinyal analog dari *VR (Variabel Resistor)* dapat terbaca oleh mikrokontroller Arduino Mega 2560. Pada konfigurasi pin-nya selektor 0-3 akan masuk ke pin 4-7 secara berurutan pada

mikrokontroller sebagai pengatur posisi *channel* yang akan dibaca. Proses ini menggunakan perhitungan bilangan biner dalam penentuan lokasi *channel* yang mana nilai decimal dari *channel* akan diubah ke bentuk biner (1 atau 0). *Multiplexer* ini di-supply 5V yang berasal dari pin vcc arduino mega 2560.

Modul *wifi* ESP8266 merupakan perangkat/alat yang digunakan sebagai perantara komunikasi data secara nirkabel antara *client* dan *server*. Dalam penelitian ini penggunaan ESP8266 memiliki fungsi sebagai pengirim data arus dan tegangan pada bus 2 – 11 yang dapat di monitoring melalui *smartphone*. Prinsip kerja dari modul ini berbentuk komunikasi serial sehingga pada konfigurasinya terdapat pin *Receiver (RX)* dan *TX (Transmitter*) yang mana pin *RX* dihubungkan pada pin *TX* pada arduino mega 2560 dan pin *TX* akan dihubungkan pada pin *RX* pada arduino mega 2560.

Prinsip kerja dari sensor *ACS712* yaitu membaca arus yang diterima oleh sensor. Pada penelitian ini menggunakan jenis sensor *ACS712* sebanyak 10 unit yang memiliki akurasi pembacaan terhadap arus maksimal 30 a*mpere*. Sensor ini pada konfigurasi *input*nya akan di rangkai seri terhadap beban yang akan di deteksi serta di-*supply* 5V. Dari ke-10 sensor *ACS712* pin *output* dari sensor akan masuk ke pin ADC A0 – A10 secara berurutan pada mikrokontroller arduino mega 2560.

Prinsip kerja dari *VR (variabel resistor)* sebagai pengatur tegangan yang dilewati oleh *VR).* Pada penelitian ini menggunakan 10 unit VR sebagai permisalan terhadap tegangan yang disupply 5V. *Output* dari VR akan masuk ke pin 0 – 10 analog *multiplexer* 16 *Channel* secara berurutan.

5.5 Pengujian *Hardware*

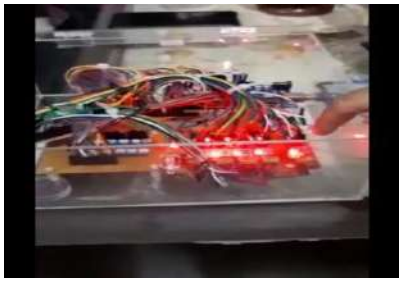
Dalam konfigurasi sistem menggunakan sumber tegangan dan arus. Selanjutnya luaran hasil simulasi di tampilkan menggunakan aplikasi internet sebagai media monitoring dengan *smartphone* atau ponsel hp. Kemudian pada rangkaian dapat di monitoring secara realtime menggunakan internet sehingga hasil yang tampilkan sesuai dengan kondisi lapangan dengan waktu yang relatif cepat.

Secara garis besar proses pengujian sistem *Hardware IoT* di fungsi masing-masing sebagai berikut :

1. Energi listrik yang dihasilkan berupa tegangan dan arus. Sensor ini pada konfigurasi input-nya akan di rangkai seri terhadap beban yang akan di deteksi serta di supply 5V. Dari ke-10 sensor ACS712 pin *output* dari sensor akan masuk ke pin ADC A0 – A9 secara berurutan pada mikrokontroller arduino mega 2560 di lihat pada Gambar 5.14.

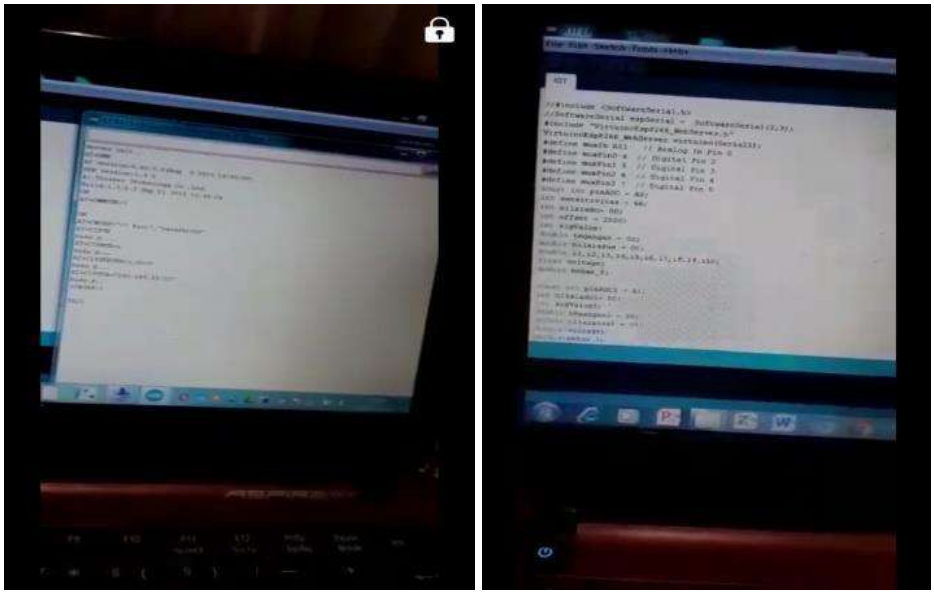
Gambar 5.14 Perangkat *Hardware*

2. *Output* dari VR akan masuk ke pin 0 – 9 analog *multiplexer* 16 Channel secara berurutan di tunjukkan pada Gambar 5.15.



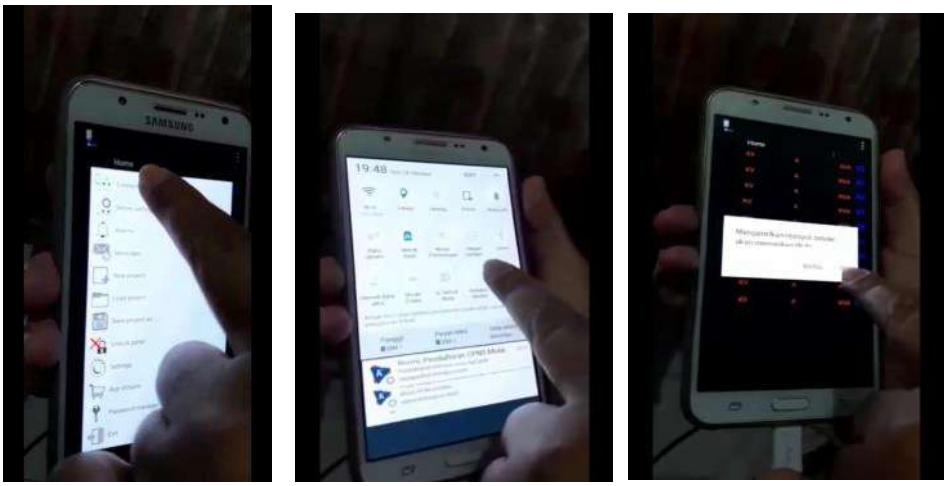
Gambar 5.15 *Output* dari sensor tegangan dan arus

3. Selanjutnya mikrokontroller arduino mega 2560, pada konfigurasi pin-nya selektor 0-3 akan masuk ke pin 4-7 secara berurutan pada mikrokontroller sebagai pengatur posisi channel yang akan dibaca dapat di lihat Gambar 5.16. Data program di lampiran I B.



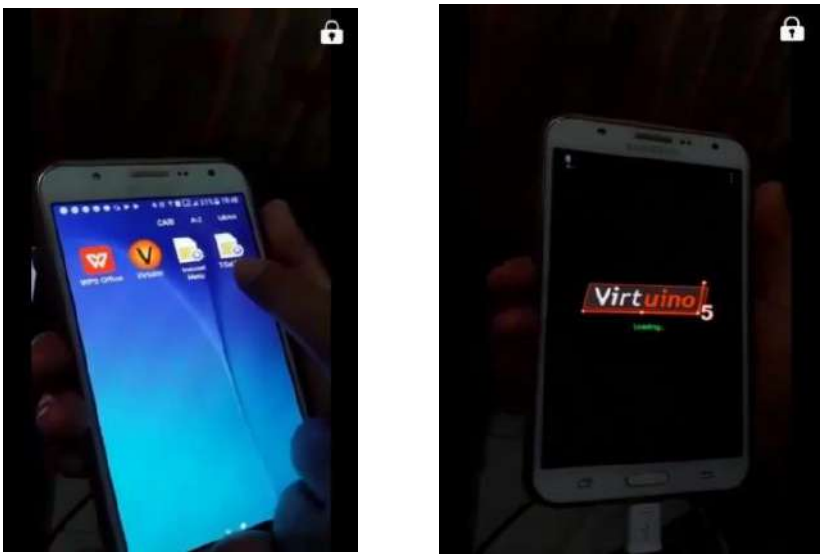
Gambar 5.16 Program tampilan di server yaitu komputer

4. *Multiplexer* mengontrol selektor serta mengirim data secara nirkabel melalui perantara *wifi*. Proses penerima data melalui android di tunjukkan pada Gambar 5.17.



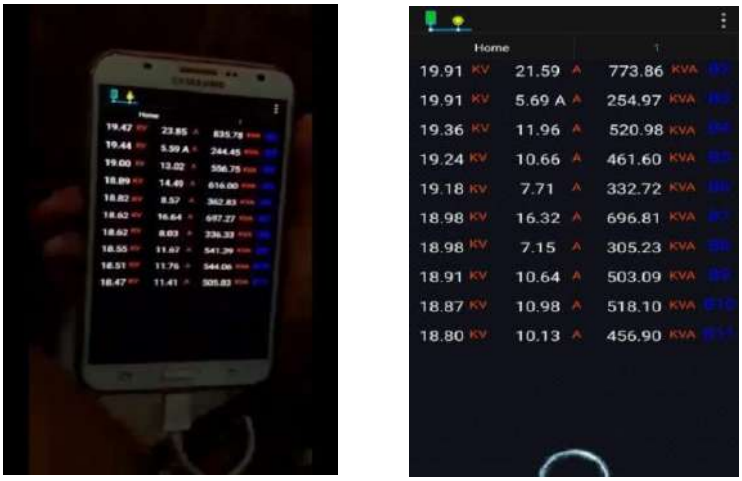
Gambar 5.17 Proses penerima data melalui di *smartphone*

5. Modul *wifi* ESP8266 merupakan perangkat/alat yang digunakan sebagai perantara komunikasi data secara nirkabel antara client dan server. Di tunjukkan pada Gambar 5.18.



Gambar 5.18 Tampilan di *smartphone*

6. *Smartphon* adalah menampilkan hasil perhitungan diperoleh data arus beserta tegangan melalui perantara modul *wifi* ESP8266 di lihat pada Gambar 5.19.

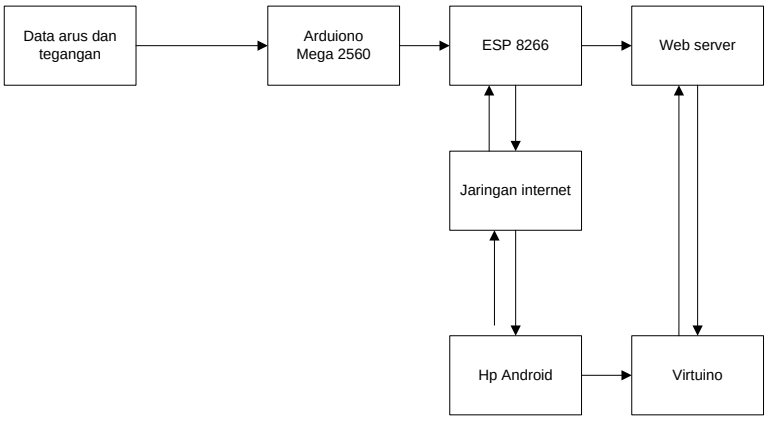


Gambar 5.19 Tampilan data di *smartphone*

5.6 Pengujian *Internet of Things (IoT)*

Pada Gambar 5.20 dapat di jelaskan proses monitoring sebagai berikut : Dari sensor arus dan tegangan masuk ke Arduino Uno, data tegangan dan arus diproses untuk mendapatkan informasi yg diperlukan user. Informasi dikrim ke *Firebase* sebagai media penyimpan data menggunakan *wifi (ESP8266)* adalah perangkat yg dapat menghubungkan Arduino dengan router). Router adalah penghubung antara wifi dengan internet. Dengan demikian informasi dari arduino dapat tersimpan di *Firebase*. *Fire base*

merupakan media penyimpan data yang dapat diakses oleh *smartphone* kapan saja dan dimana saja menggunakan media internet.



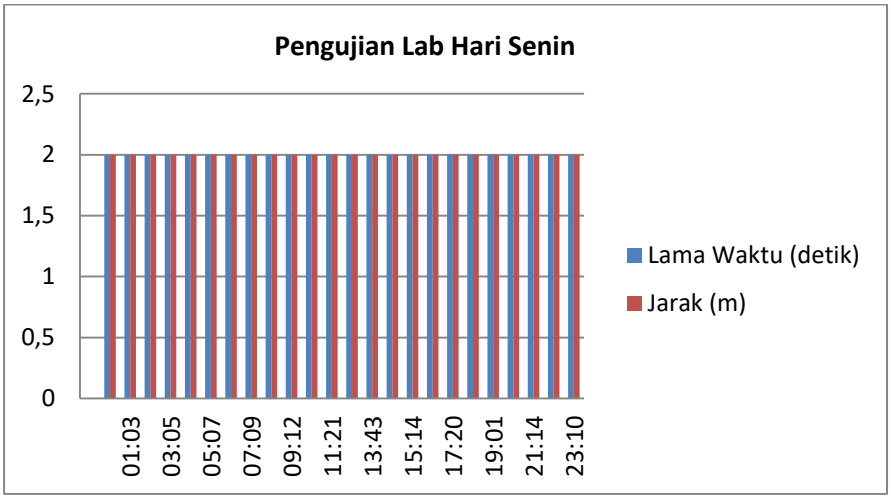
Gambar 5.20 Diagram Alur Sistem Monitoring *Smartphon*

Pengujian monitoring dilakukan dua tahap yaitu pertama tahap pengujian laboratorium dan tahap pengujian lapangan. Pengujian dilakukan selama 7 hari dari hari Senin sampai Sabtu (tanggal 27 Juli sampai dengan tanggal 1 Agustus 2020).

Tabel 5.7 Hasil pengujian laboratorum pada hari Senin

Waktu	**Jarak (m)**	**Data Masuk**		**Lama Waktu (detik)**
		Ada	**Tidak**	
00:01	2	√	-	2
01:03	2	√	-	2
02:02	2	√	-	2
03:05	2	√	-	2
04:03	2	√	-	2
05:07	2	√	-	2
06:11	2	√	-	2
07:09	2	√	-	2
08:05	2	√	-	2
09:12	2	√	-	2
10:24	2	√	-	2
11:21	2	√	-	2
12:01	2	√	-	2
13:43	2	√	-	2
14:12	2	√	-	2
15:14	2	√	-	2
16:18	2	√	-	2
17:20	2	√	-	2
18:21	2	√	-	2
19:01	2	√	-	2
20:08	2	√	-	2
21:14	2	√	-	2
22:18	2	√	-	2
23:10	2	√	-	2

Hasil pengujian sebagian di tunjukkan pada Tabel 5.7 sampai dengan Tabel 4.8. Sedangkan grafik ditunjukkan pada Gambar 5.23 sampai dengan 5.25, dan lainnya di lampiran 5.

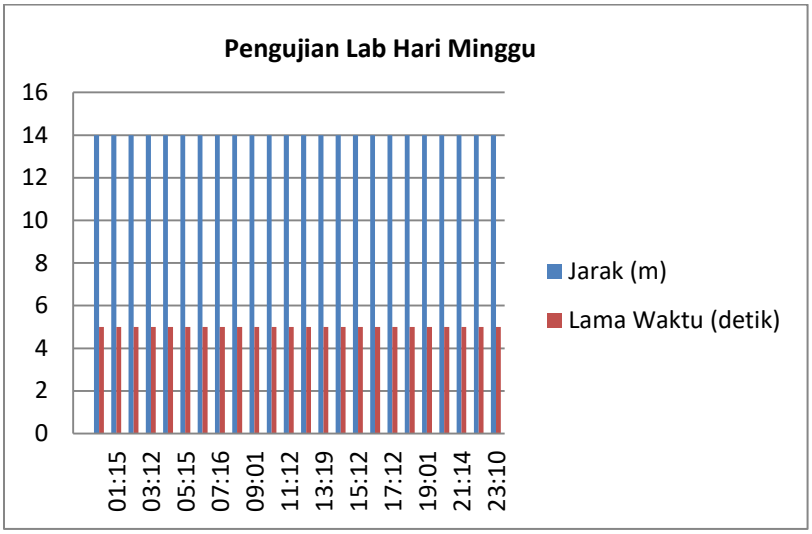


Gambar 5.21 Grafik hasil monitoring menggunakan *IoT*

Tabel 5.8 Hasil pengujian laboratorium pada hari Minggu

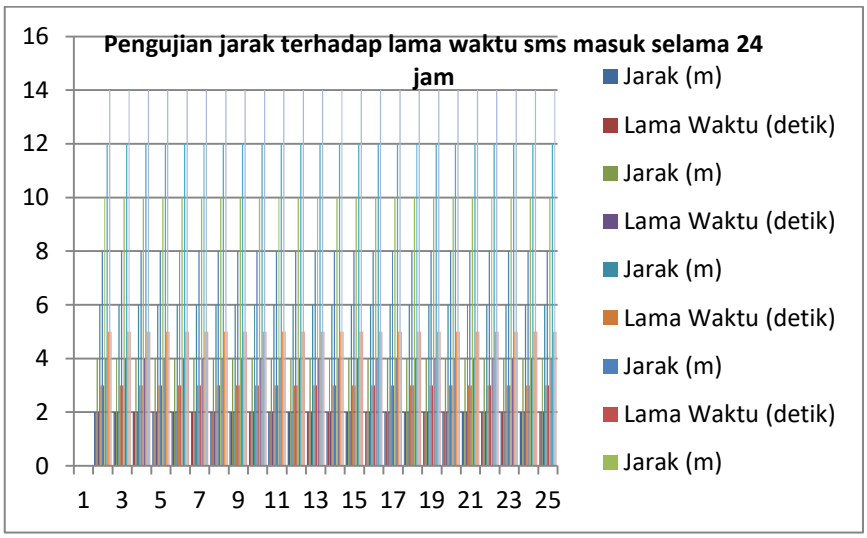
Waktu	Jarak (m)	Data Masuk		Lama Waktu (detik)
		Ada	Tidak	
00:10	14	√	-	5
01:15	14	√	-	5
02:23	14	√	-	5
03:12	14	√	-	5
04:14	14	√	-	5
05:15	14	√	-	5
06:12	14	√	-	5
07:16	14	√	-	5
08:02	14	√	-	5
09:01	14	√	-	5
10:10	14	√	-	5
11:12	14	√	-	5
12:02	14	√	-	5
13:19	14	√	-	5
14:15	14	√	-	5
15:12	14	√	-	5
16:10	14	√	-	5
17:12	14	√	-	5
18:19	14	√	-	5
19:01	14	√	-	5
20:08	14	√	-	5
21:14	14	√	-	5
22:18	14	√	-	5
23:10	14	√	-	5

Dari hasil pengujian di laboratorium bahwa jarak 2 sampai dengan 4 meter menggunakan wifi di butuhkan 2 detik lamanya data di terima oleh *smartphone*. Alat di uji tiap jam selama 24 jam pada hari Senin dan Selasa. Sedangkan untuk jarak 6 sampai dengan 8 meter di butuhkan waktu 3 detik, di ujikan pada hari Rabu dan Kamis. Untuk jarak 10 sampai dengan 12 meter diujikan pada hari Jumat dan Sabtu dan dibutuhkan waktu 4 detik. Pada hari Minggu di ujikan untuk jarak 14 meter dan waktunya 5 detik. Pada Tabel 4.7 dan 4.8 adalah hasil sampling pengujian dari 7 hari.



Gambar 5.22 Grafik hasil monitoring menggunakan *IOT*

Hasil pengujian jarak terhadap waktu selama 1 minggu di tunjukkan pada grafik Gambar 54.27 dengan rata- rata waktu 2,8 detik.



Gambar 5.23 Grafik hasil monitoring skala laboratorium selama 7 hari

Gambar 5.24 Peta jarak lokasi transmitter (alat) terhadap receiver *smartphone*

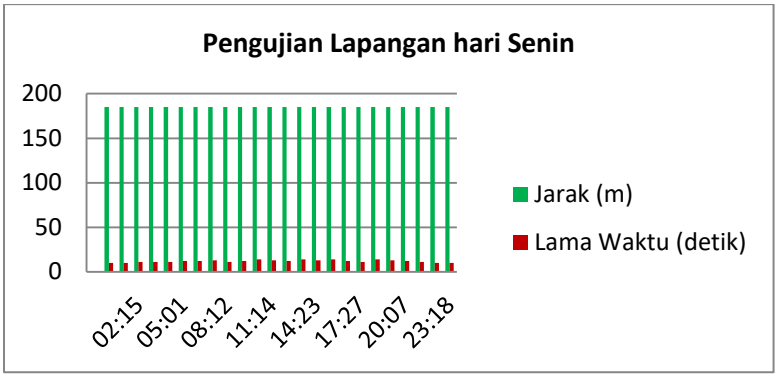
Selanjutnya di lakukan pengujian alat di lapangan dengan lokasi di Madiun dari start Jl Wiguna Selatan (Gunung Anyar) di terima di wilayah desa Petung Rejo Magetan dengan jarak 185 km yang dilihat pada Gambar 5.29.

Hasil sampling pengujian pada hari Seni dan Minggu ditunjukkan pada Tabel 5.9 dan 5.10 selama 24 jam dengan menggunakan internet (modem). Untuk hari Selasa sampai dengan Sabtu di lampiran.

Sedangkan grafik di tunjukkan pada Gambar 5.30 dan 5.31. Terjadi perubahan disebabkan karena sinyal yang diterima dari *Base Tranreceiver Station (BTS).*

Tabel 5.9 Hasil pengujian lapangan pada hari Senin

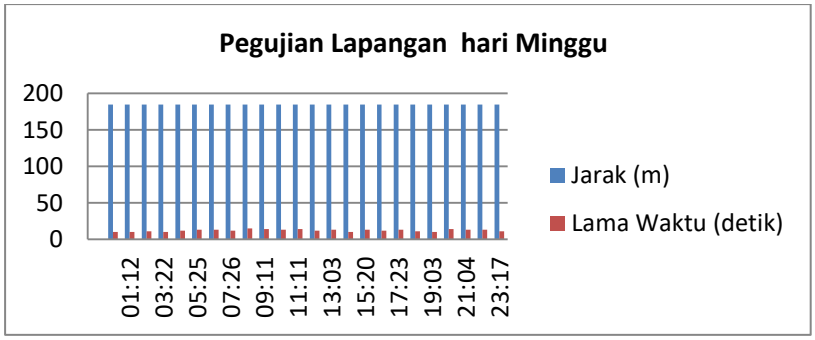
Waktu	Jarak (m)	Lama Waktu (detik)
00:07	185	10
01:13	185	10
02:15	185	11
03:11	185	11
04:06	185	11
05:01	185	12
06:13	185	12
07:20	185	13
08:12	185	11
09:21	185	12
10:10	185	14
11:14	185	13
12:27	185	12
13:15	185	14
14:23	185	13
15:06	185	14
16:18	185	12
17:27	185	11
18:17	185	14
19:19	185	13
20:07	185	12
21:10	185	11
22:13	185	10
23:18	185	10



Gambar 5.25 Grafik hasil monitoring

Tabel 5.10 Hasil pengujian laboratorum pada hari Minggu

Waktu	Jarak (m)	Lama Waktu (detik)
00:30	185	10
01:12	185	10
02:13	185	11
03:22	185	10
04:34	185	12
05:25	185	13
06:10	185	13
07:26	185	12
08:02	185	15
09:11	185	14
10:20	185	13
11:11	185	14
12:22	185	12
13:03	185	13
14:30	185	10
15:20	185	13
16:12	185	12
17:23	185	13
18:29	185	11
19:03	185	10
20:03	185	14
21:04	185	13
22:08	185	13
23:17	185	11

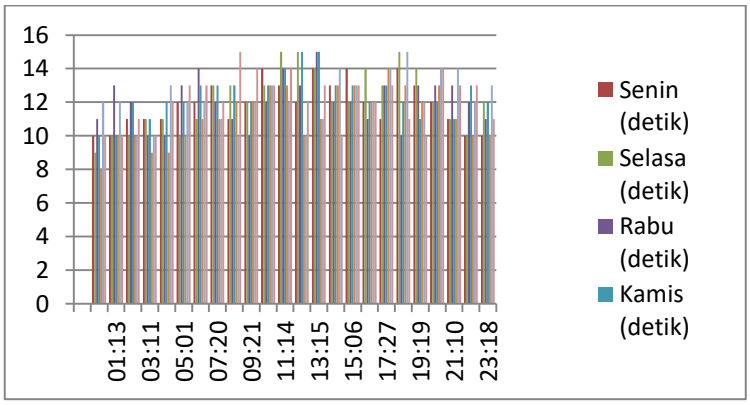


Gambar 5.26 Grafik hasil monitoring uji lapangan selama 24 jam

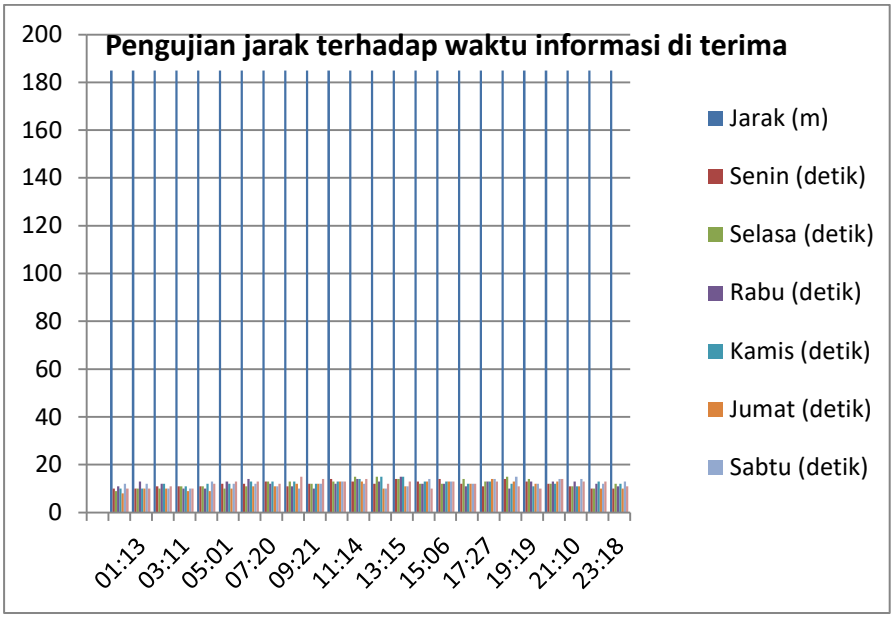
Pada Tabel 5.11 hasil pengujian lapangan selama 7 hari. Sedangkan grafik Gambar 5.30 dan 5.31 hasil tampilan jarak terhadap waktu data diterima oleh *smartphone.* Untuk lama pengiriman selama 24 jam sekitar 10,416 detik.

Tabel 5.11 Hasil pengujian lapangan selama 7 hari

Waktu	**Jarak (m)**	**Senin (detik)**	**Selasa (detik)**	**Rabu (detik)**	**Kamis (detik)**	**Jumat (detik)**	**Sabtu (detik)**	**Minggu (detik)**
00:07	185	10	9	11	10	8	12	10
01:13	185	10	10	13	10	10	12	10
02:15	185	11	10	12	12	10	10	11
03:11	185	11	11	10	11	9	10	10
04:06	185	11	11	10	12	9	13	12
05:01	185	12	10	13	12	10	12	13
06:13	185	12	11	14	13	11	12	13
07:20	185	13	13	12	13	11	11	12
08:12	185	11	13	11	13	12	10	15
09:21	185	12	12	10	12	12	12	14
10:10	185	14	13	12	13	13	13	13
11:14	185	13	15	14	14	13	12	14
12:27	185	12	15	13	15	10	10	12
13:15	185	14	14	15	15	11	11	13
14:23	185	13	12	12	13	13	14	10
15:06	185	14	12	12	13	13	13	13
16:18	185	12	14	11	12	12	12	12
17:27	185	11	13	13	13	14	14	13
18:17	185	14	15	10	12	13	15	11
19:19	185	13	14	13	11	12	12	10
20:07	185	12	12	13	12	13	14	14
21:10	185	11	11	13	11	11	14	13
22:13	185	10	10	12	13	10	12	13
23:18	185	10	12	11	12	10	13	11



Gambar 5.27 Grafik hasil monitoring uji lapangan selama 7 hari



Gambar 5.28 Grafik hasil monitoring uji lapangan jarak terhadap data di terima *smartphone* selama 7 hari

Berdasarkan hasil percobaan dari simulasi yang telah dibuat dan perhitungan serta analisa sistem *hybrid online* dan *offline* dengan menggunakan *simulink* matlab beserta arduino maka dapat disimpulkan sebagai berikut :

1. Hasil dari simulasi ini bisa di terapkan dengan penyulang lain apabila ingin mencari nilai daya aktif dan daya reaktif dari setiap bus dikarenakan hasil *error* perbandingan antara hasil simulasi dengan perhitungan matematis dan juga hasil *running* simulasi menghasilkan nilai *error* <1%.
2. Pada percobaan simulasi terdapat noise yang membuat hasil nilai tidak bisa akurat 100%
3. Simulasi yang dibuat dapat menghitung nilai daya aktif dan daya reaktif pada masing-masing bus di tiap-tiap *cluster* sehingga dapat juga diketahui total daya aktif dan daya reaktif pada tiap-tiap *cluster*.
4. Penggunaan potensiometer sebagai komponen untuk mengatur nilai input tegangan dan arus dari data Bendul Merisi memberikan nilai yang belum 100% akurat dengan nilai input yang diinginkan sehingga nilai input terkadang melebihi nilai asli dan juga kadang kurang dari nilai input sesungguhnya.

5. Pengujian alat pada skala laboratorium selama 7 hari memiliki waktu rata-rata 2,8 detik. Sedangkan untuk pengujian lapangan di butuhkan waktu rata – rata sebesar10,416 detik selama 24 jam.

(Paten *Hardware* no. : P00202000082 dan *software* no. pencatatan 000174886).

<Halaman ini sengaja dikosongkan>

BAB 6 PENUTUP

6.1 Kesimpulan

Penelitian ini memberikan kontribusi berupa metode untuk mereduksi jumlah *PMU* dan optimalisasi penempatan *PMU* menggunakan suatu penggabungan yang disebut *Integer Linear K-Means*. Jaringan distribusi radial Bendul Merisi yang menjadi objek penelitian ini memiliki 11 bus, sehingga perlu dilakukan optimalisasi penempatan *PMU*. Penggunaan metode *Integer Linear K-Means Clustering* untuk optimasi penempatan *PMU* menunjukkan hasil yang baik. Dengan jaringan sebanyak 11 bus, hanya dibutuhkan 3 *PMU* sehingga terjadi penurunan jumlah *PMU* sebesar 73%. Data parameter tegangan, arus, dan impedansi merupakan data hasil pengukuran di lapangan dan memiliki nilai pecahan. Hasil penempatan *PMU* pada bus 3, 6, dan 10 merupakan penempatan *PMU* yang paling optimal. Untuk data yang di ujikan pada jaringan Transmisi dengan parameter tegangan dan sudut, dengan 19 bus Jawali 500 kV dengan metode ini di dapatkan 3 *cluster* di pasang 3 *PMU* di bus 6, 9 atau 10 dan bus 13, dengan penurunan sebanyak 84% pemasangan *PMU*.

Jaringan terdistribusi Bendul Merisi Surabaya memiliki 11 bus dan terbagi dalam 3 *cluster* seperti yang di dapatkan pada kesimpulan diatas. Penelitian selanjutnya memperkirakan daya dan tegangan bus terhadap bus yang tidak dipasang oleh *PMU* (bus tak teramati). Untuk proses estimasi ini, digunakan algoritma *ANFIS* dengan *hybrid PSO-GA* yang dimodifikasi. Algoritma ini telah menunjukkan kinerja yang baik dengan memperoleh nilai akurasi yang tinggi. Dengan set data tegangan dan arus masukan pada setiap bus di setiap *cluster*, perkiraan nilai daya dan tegangan pada bus yang tidak teramati dapat diperoleh. Hasil perbandingan antara daya terukur dengan daya estimasi diperoleh nilai akurasi pada masing-masing *cluster*1, 2, dan 3 adalah sebesar 99,976%, 99,954%, dan 99,975%. Hasil perbandingan antara tegangan yang diukur dengan tegangan estimasi, diperoleh nilai akurasi masing-masing pada *cluster* 1, 2, dan 3 adalah sebesar 99,704%, 99,604%, dan 99,766%.

Hasil dari validasi alat dan dari simulasi ini bisa di terapkan dengan penyulang lain apabila ingin mencari nilai daya aktif dan daya reaktif dari setiap bus dikarenakan hasil *error* perbandingan antara hasil simulasi dengan

perhitungan matematis dan juga hasil *running* simulasi menghasilkan nilai *error* <1%. Simulasi yang dibuat dapat menghitung nilai daya aktif dan daya reaktif pada masing – masing bus di setiap *cluster* sehingga dapat juga diketahui total daya aktif dan daya reaktif pada tiap – tiap *cluster*. Pengujian alat untuk mengirim data menggunakan *Internet of Things (IoT)* pada skala laboratorium selama 7 hari memiliki waktu rata- rata 2,8 detik. Sedangkan untuk pengujian lapangan di butuhkan waktu rata –rata sebesar10,416 detik selama 24 jam.

6.2 Saran

Saran pada penelitian untuk hasil yang lebih optimal di butuhkan data yang lengkap agar dapat di dapatkan pola yang jelas hasil yang semakin akurat.

DAFTAR PUSTAKA

[1]. U.S. Department of Energy, National Energy Technology Laboratory, Modern Grid Initiative (2007), http://www.netl.doe.gov/moderngrid/ opportunity/ vision_technologies.html. Archived July 11, 2007, at the Wayback Machine .

[2]. Berger, Lars T.; Iniewski, Krzysztof, eds.April " Smart Grid - Applicacions, Communications and Security". John Wiley and Sons. ISBN 978-1-1180-0439-5, 2012.

[3]. F.R. Yu, P. Zhang, W. Xiao, and P. Choudhury,), "Communication Systems for Grid Integration of Renewable Energy Resources," IEEE Network, vol. 25, no. 5, pp. 22-29, 2011.

[4]. Singh, S.P. and Singh, S "Optimal *PMU* placement in power system considering the measurement redundancy", Advance in Electronic and Electric Engineering, Vol. 4, 2014.

[5]. Yunan Helmy Amrulloh, Rony Seto Wibowo, dan Sjamsjul Anam, (2013), "Penempatan Optimal Phasor Measurement Unit (*PMU*) dengan Integer Programming", Jurnal Teknik POMITS Vol. 2, No. 2, ISSN: 2337-3539, 2013.

[6]. Wira Candra Nur Pratama, Ontoseno Penangsang, Rony Seto Wibowo, "Penentuan Lokasi Pengukuran Untuk State Estimasi Pada Jaringan Distribusi menggunakan Teknik Dekomposisi", Jurnal Teknik POMITS, Vol. 1, No.1,2014.

[7]. M. Pau, P. A. Poegoraro, and S. Sulis, , "Branch current state estimator for distribution system based on synchronized measurements" Proc. IEEE Int. Workshop AMPS, Sep. 2012, pp. 53-58.

[8]. M. Pau, P. A. Poegoraro, and S. Sulis,"Efficient Branch-Current-Based Distribution System State Estimation Including Synchronized Measurements", IEEE Transactions on Intrumentation and Measurement, Vol. 62, No. 9, 2013.

[9]. M. Pau, P. A. Poegoraro, and S. Sulis," WLS Distribution System State Estimator Based on Voltage or Branch-currents: Accuracy and Perfomance Comparison" IEEE Conference Publications, Italy, 2013.

[10]. Xu B, Abur A. Observability analysis and measurement placement for systems with *PMU*s. In: Proceedings of the IEEE PES power systems conference and exposition 2004;2: p. 943–6.

[11] Xu B, Abur A. Optimal placement of phasor measurement units for state estimation. Final project report, PSERC 2005.

[12]. Manousakis, N.M. and Korres, G.N , "A weighted least squares algorithm for optimal *PMU* placement", IEEE Transactions on Power Systems, Vol. 28, No. 3, pp.3499–3500, 2013.

[13]. Mahari, A. and Seyedi, H,"Optimal *PMU* placement for power system observability using BICA, onsidering measurement redundancy", Electric Power Systems Research, Vol. 103, pp.78–85, 2013.

[14]. Tai, X., Marelli, D., Rohr, E. and Fu, M, "Optimal *PMU* placement for power system state estimation with random component outages", International Journal of Electrical Power & Energy Systems, Vol. 51, 2013, pp.35–42.

[15]. Sodhi, R., Srivastava, S.C. and Singh, S.N, "Optimal *PMU* placement method for complete topological and numerical observability of power systems", Electric Power Systems Research, Vol. 80, No. 9, 2010, pp.1154–1159.

[16]. Azizi, S., Dobakhshari, A.S., Nezam Sarmadi, S.A. and Ranjbar, A.M, "Optimal *PMU* placement by an equivalent linear formulation for exhaustive search", IEEE Transactions on Smart Grid, Vol. 3, No. 1, 2012, pp.174–182 .

[17]. Fei, Z., Hao, X., Daonong, Z., Xiaoyi, Z. and Yubo, Y. (2014) "A new strategy for optimal *PMU* placement based on limited exhaustive approach", in 2014 International Conference on Power System Technology (POWERCON), pp.67–74.

[18]. Hajian, M., Ranjbar, A.M., Amraee, T. and Mozafari, B, "Optimal placement of *PMU* to maintain network observability using a modified B*PSO* algorithm", International Journal of Electrical Power & Energy Systems, Vol. 33, No. 1, 2011, pp.28–34.9.

[19]. Sharma, C. and Tyagi, B, "An approach for optimal *PMU* placement using binary particle swarm optimization with conventional measurements",

International Journal of Engineering, Science and Technology, Vol. 3, No. 3, 2011, pp.56–63.

[20]. El-Zonkoly, A., El-Safty, S. and Maher, R, "Optimal placement of *PMU* using improved tabu search for complete observability and out-of-step prediction", Turkish Journal of Electrical Engineering & Computer Sciences, Vol. 21, No. 5,2013, pp.1376–1393.

[21]. Jamuna, K. and Swarup, K.S , "Multi-objective biogeography based optimization for optimal *PMU* placement", Applied Soft Computing, Vol. 12, No. 5, 2012 , pp.1503–1510.

[22]. Ghosh, D., Ghose, T. and Mohanta, D, "Reliability analysis of geographic information system (GIS) aided optimal phasor measurement unit location for smart grid operation", Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability, p.1748006X13482853, 2013.

[23]. Golberg, D.E. (Addison-Wesley Reading, MA, "Genetic algorithms in search, optimization and machine learning ", 1989.

[24]. A.Z. GAmm, Senior Member, IEEE I.N.Kolosok, Senior Member , IEEE, A.M. Glazunova, E.S. Korkina " *PMU* placement criteria for EPS state estimation", 2008.

[25]. Patirikkat gopakumar,Maddikara Jaya Bharata Reddy and Dusmanta kumar Mohanta. Proceding "A Novel Topological Genetic Algorithm-Based Phasor Measurement Unit Placement and Scheduling Methodology for Enhanced State Estimatio)", 2015.

[26]. Muhammad Nizam "Kohonen Neural Network *Cluster*ing For Voltage Control In Power Systems, Telkomnika ISSN: 1693-6930", 2010.

[27]. Yang Wei-hong, Dai Ai-ying, Zhang Hong-bin, " Dynamic Characteristics *Cluster*ing of Electric Loads Based on Kohonen Neural Network " , North China Electric Power University, Changping District, Beijing, China IEEE, 2010.

[28]. Gheorghe Grigoras, Gheorghe Cartina, Florina Rotaru," Using *K-Means Cluster*ing Method in Determination of the Energy Lo*SEE*s Levels from Electric Distribution Systems" , Athematical Methods and Computational Techniques', 2015.

[29]. Saran Satsangi, Ashish Saini and Amit Saraswat, (SEICON) “Voltage Control Areas for Reactive Power Management using *Cluster*ing Approach in Deregulated Power System”, Chennai and Dr.MGR University Second International Conference on Sustainable Energy and Intelligent System “, 2011.

[30]. F.J. Marı´n, F. GArcı´a-Lagos, G. Joya and F Sandoval ,” Genetic algorithms for optimal placement of phasor measurement units in electrical networks, IEEE”,2003.

[31]. Rejina Parvin , Assistant Professor, Vasanthanayaki C, Associate Professor Rejina And Vasanthanayaki ” Particle Swarm Optimization Based *Cluster*ing”, IEEE.BPPT, *Outlook Energi Indonesia* . BPPT, 2015.

[32] Riny Sulistyowati, Mochamad Ashari, Dedet Candra Riawan, Rony Seto Wibowo, “Optimum Placement of Measurement Devices on Distribution Networks using Integer Linear *K-Means Cluster*ing Method,” Vol. 96 No. 10 DOI:10.15199/48.2020.10.23, *Przglad Elektrotechnizy*, 2020.

[33] K. Kiran Kumar, N. Venkata Ramana, S. Kamakshaiah., “Design and Development of Modified *PSO* Algorithm for Network Reconfiguration”, International Review of Electrical Engineering (IREE), Vol. 8, No. 5, pp. 1586-1593, 2013.

[34] Wang, Hui; Sun, Hui; Li, Changhe; Rahnamayan, Shahryar; Pan, Jeng-shyang, “Diversity enhanced particle swarm optimization with neighborhood search”, Information Sciences, Vol. 223, pp. 119–135, ELSEVIER, 2013.

[35] GAng,Ma., Wei, Zhou., “A novel particle swarm optimization algorithm based on particle migrasion”, Applied Mathematics and Computation, Vol. 218, pp. 6620–6626, ELSEVIER, 2012.

[36] Ramadoni Syahputra, Imam Robandi, Mochamad Ashari, ” Performance Improvement of Radial Distribution Network with Distributed Generation Integration Using Extended Particle Swarm Optimization Algorithm”, International Review of Electrical Engineering (I.R.E.E.), Vol. 10, No. 2, ISSN 1827-6660, March – April 2015.

[37] Sulistyowati, Riny; Riawan, Dedet Candra; Ashari, Mochamad “*Cluster*ing Based Optimal Sizing and Placement of PV-DG Using Neural Network”,

Advanced Science Letters, Vol 23, No. 3, pp. 2373-2375, American Scientific Publisher, March 2017.

[38] Majed Jabri, Helmi Aloui., "Genetic Lagrangian Relaxation Selection Method for the Solution of Unit Commitment Problem", International Journal on Engineering Applications (IREA), Vol 7, No. 2, 2019.

[39[Ariani Indrawati, Abba SuGAnda Girsang, "Electricity Demand Forecasting Using Adaptive Neuro-*Fuzzy* Inference System and Particle Swarm Optimization", International Review of Automatic Control (IREACO), Vol. 9, No. 6, 2016.

[40] Sabzi, Hamed Zamani; Humbersona, Delbert; Abuduc, Shalamu; King, James Phillip, "Optimization of adaptive *fuzzy* logic controller using novel combine devolutionary algorithms, and its application in Diez Lagos flood controlling system, Southern New Mexico", Expert Systems With Applications, Vol. 43, pp. 154–164, ELSEVIER , 2016.

[41] Dervis KaraboGA, Ebubekir Kaya, "Adaptive network based *fuzzy* inference system (*ANFIS*) training approaches: a comprehensive survey", Artificial Intelligence Review,Vol. 52, pp. 2263–2293, Springer, 2019.

[42] Irvan Santoso, Robin Solala Gulo, Abba SuGAnda Girsang, "An Efficient Method for *Software)* Reliability Growth Model Selection Using Modified Particle Swarm Optimization Technique", International Review on Computers and *Software)* (IRECOS), Vol. 10, No. 12, 2015.

[43] Ba*SEE*ra, Ho*SEE*in., Karami, Hojat, et al., "Hybrid *ANFIS–PSO* Approach For Predicting Optimum Parameters Of A Protective Spur Dike", Applied Soft Computing, Vol. 30, pp. 642–649, ELSEVIER, 2015.

[44] Yue-Jiao Gong, Jing-Jing Li, Yicong Zhou, Yun Li, Henry Shu-Hung Chung, Yu-Hui Shi, Jun Zhang, "Genetic Learning Particle Swarm Optimization", IEEE Transactions on Cybernetics, Vol 46, No. 10, pp. 2277 - 2290, October 2016.

[45] Irvan Santoso, Robin Solala Gulo, Abba SuGAnda Girsang, "An Adaptive Cat Swarm Optimization Based on Particle Swarm Optimization Approach (AC*PSO*) for *Cluster*ing", International Review on Computers and *Software)* (IRECOS) journal, Vol 11, Number 1 2016.

[46] P*IoT*r Dziwinski, Łukasz Bartczuk, and P*IoT*r Goet-zen, “A new hybrid particle swarm optimization and evolutionary algorithm”, International Conference on Artificial Intelligence and Soft Computing, pp. 432–444, Springer, 2019.

[47] Yanhui Xi, Xin Tang, Zewen Li, Yonglin Cui, Ting Zhao, Xiangjun Zeng, Jun Guo, Wei Duan, “Harmonic estimation in power systems using an optimised adaptive Kalman filter based on *PSO*-GA”, IET Generation, Transmission & Distribution, Vol. 13, No. 17, pp. 3968 – 3979, 2019.

[48] Sabzi, Hamed Zamani., Humbersona, Delbert., Abuduc, Shalamu., KinGA, , James Phillip., “Optimization of adaptive *fuzzy* logic controller using novel combine devolutionary algorithms, and its application in Diez Lagos floodcontrolling system, Southern New Mexico”, Expert Systems with Applications: An International Journal, Vol 43, pp.154–164., ELSEVIER, 2016.

[49] Sarkheyli, Arezoo., Zain, Azlan Mohd., Sarif, Safian., “Robust Optimation of *ANFIS* based on a new Modified GA”, Neuro Computing, Vol. 166, pp. 357-366, ELSEVIER, 2015.

[50] Rini, Dian Palupi; Shamsuddin, Siti Mariyam; Yuhanis, Sophiayati Yuhanis, “Particle Swarm Optimization for *ANFIS* Interpretability and Accuracy”, Methodologies and Application: Soft Computing, Vol. 20, pp. 251–262, Springer, 2016.

[51] M. Pau, P.A. Pegoraro, S. Sulis, “ Efficient branch-current-based distribution system state estimation including synchronized measurements,” IEEE Transactions on Instrumentation and Measurement, Vol. 62, No. 9, pp. 2419–2429, 2013.

[52] Antti Mutanen, Pertti Jarventausta, Sami Repo,”Smart Meter Data-Based Load Profiles and Their Effect on Distribution System State Estimation Accuracy”, International Review of Electrical Engineering (I.R.E.E), Vol, 12, No. 6, 2017.

[53] E. Mantisas, R. Singh, B.C. Pal, & G. Strbac. "Distribution System State Estimation Using an Artificial Neural Network Approach for Pseudo Measurement Modelling", IEEE Transactions on Power Systems, Vol. 27, No. 4, pp. 1888-1896, 2012.

[54] M. Biserica, Y. Besanger, R. Caire, et al., " Neural networks to improve distribution state estimation – Volt Var control performances, IEEE Transactions on Smart Grid journal, Vol. 3, No. 3, pp. 1137–1144, 2012.

[55] P*IoT*r Dziwińsk, Łukasz Bartczuk, "A New Hybrid Particle Swarm Optimization and Genetic Algorithm Method Controlled by *Fuzzy* Logic", IEEE Transactions on *Fuzzy* Systems journal, Vol. 28, No. 6, pp. 1140 – 1154, 2020

[56] Riny Sulistyowati, Mochamad Ashari, Dedet Candra Riawan, Rony Seto Wibowo, "Power and Voltage Estimation Of *Unobserved* Bus On Distribution Network Using *ANFIS* Algorithm With The Modified *PSO*-GA Hybrid" International Review of Electrical Engineering (I.R.E.E., 2020.

[57] Fan, N. and Watson, J-P, "On Integer Programming models for the multi-channel *PMU* placement problem and their solution", EnergySystems, Vol. 6, No. 1, pp.1–19, 2014.

[58] Roy, B.S., Sinha, A. and Pradhan, A, "An optimal *PMU* placement technique for power system observability", International Journal of Electrical Power & Energy Systems, Vol. 42,No. 1, 2012, pp.71–77.

[59] Tan, P.N., Steinbach, M., Kumar, V. INTRODUCTION TO DATA MINING. Boston:Pearson Education, 2006.

[60] Jain. A.K. Data *Cluster*ing: 50 Years Beyond *K-Means*. Pattern Recognition Letters, 2009.

[61] Han, J., Kamber, M., Pei, J.: Data Mining Concept and Techniques, 3rd ed. Morgan Kaufmann-Elsevier, Amsterdam, 2012.

[62] Michael Jonatan, Imam Robandi, Marwan Rosyadi," Optimal Reactive Power Dispatch using Genetic Algorithms", IES – Politeknik Elektronika Negeri Surabaya - ITS, 2005.

[63] V. Bhuvaneswari, R Porkodi, "The Internet of Things (*IoT*) Applications and Communication Enabling Technology Standards: An Overview", International Conference on Intelligent Computing Applications, 2014

[64] "Internet of Things: Wireless Sensor Networks", White Paper, IEC Publication

[65] B.N. Karthik, L. DurGAParameswari, R. Harshini, A.Akshaya, “Survey on *IOT* & Arduino Based Patient Health Monitoring System”, *International Journal of Scientific Research in Computer Science, Engineering and Information Technology* ,Vol. 3 ,No.1, 2018

[66] https://www.kelasplc.com, May, 25, 2020

[67] MourvikaShirode, Monika Adaling, JyotiBiradar, Trupti Mate,” *IOT* Based Water Quality Monitoring System”,International Journal of Scientific Research inComputer Science, Engineering and Information Technology (IJSRCSEIT) | Vol. 3, No.1, 2018

[68] M.U. Farooq , Muhammad Waseem , Sadia Mazhar , AnjumKhairi , Talha Kamal ,”A Review on Internet of Things (*IoT*) “,International Journal of Computer Applications , Vol 113 , No. 1, March 2015.

[69] Andrea Zanella Nicola Bui, Angelo Castellani,LorenzoVangelista, and Michele Zorzi, „Internet of Things for Smart Cities “,*IEEE Internet Of Things Journal* Vol.1, No.1, February 2014.

LAMPIRAN 1

Tabel 4.4 Tabel hasil pengujian simulasi

Cluster	Bus	1			1			1			1			1			1		
		8:00			9:00			10:00			11:00			12:00			13:00		
		kVA	P	Q	kVA	P	Q	kVA	P	Q	kVA	P	Q	kVA	P	Q	kVA	P	Q
1	2	762	602834.3	452125.75	765	606871.19	454127.67	766	607543.94	455657.97	768	608216.75	456162.59	768	608889.5	456667.16	770	608316.66	456754.47
	3	208	166183.11	124637.34	208	167675.83	125756.87	209	166183.11	124637.33	213	168201.55	126151.16	215	170892.76	128169.58	213	170792.76	128149.58
	4	539	423497.56	317623.19	539	423497.56	317623.19	541	424839.84	318629.91	542	426853.31	320140	545	428195.59	321146.72	545	428395.59	323146.72
2	5	539	421034.47	315775.84	539	422454.47	316840.84	540	421784.97	316338.72	543	425801.97	319351.47	543	425132.47	318849.34	545	423132.47	318549.34
	6	375	294216.28	220662.22	375	292878.97	219659.22	377	295553.625	221665.23	379	296222.31	222166.73	380	291849.09	218886.81	379	294849.09	218486.81
	7	834	650482.8	487862.12	837	653154.19	489865.66	836	650482.81	487862.12	837	653822.12	490366.59	840	653962.56	490471.94	842	653662.56	490571.94
	8	349	270477.97	202858.48	349	270477.97	202858.48	353	275820.75	206865.58	356	277156.47	207867.34	358	279159.97	209369.98	355	273159.97	203369.98
3	9	549	426384	319773	551	427964.16	320973.12	554	431562.16	323671.62	565	440233.41	330175.06	567	439021.72	329266.28	565	439521.72	329666.28
	10	558	435024.56	328268.44	563	439566.41	329674.81	567	440354.06	330265.56	567	439687.91	329765.94	568	441138.47	330853.84	565	440138.47	333853.84
	11	458	356413.69	267310.28	457	357079.84	267809.91	457	357298.06	267973.56	460	357079.84	267809.91	462	357740.25	268305.19	460	355740.25	268205.19

LAMPIRAN II

A. *Program Integer Linear K-Mean*

```
% Program: kmeans
% Oleh   : Riny Sulistyowati -  mhs s3 Teknik Elektro
clc
clear
kelas=3;
%data =[1 1;4 1;6 1;1 2;2 3;5 3;2 5;3 5;2 6;3 8];
load('data.mat')
%data=data2;
[a,b]=size(data);
yawal=randi([1 kelas],1,a);


for loop=1:2000
  cent=cari_centroid(kelas,data,yawal);

  obj=cari_nilaiobj(kelas,data,cent);

  [obj2,kls]=cari_kelas(kelas,data,obj);

  SEE(loop)=sum(sum(obj2));
  for i=1:kelas
     [val,dekat(i)]=min(obj(:,i));
  end

   if yawal==kls
       break
  end
  yawal=kls;
clc
clear
kelas=3;
%data =[1 1;4 1;6 1;1 2;2 3;5 3;2 5;3 5;2 6;3 8];
load('data.mat')

for k=3:6
    [SEE(:,k),kls(k),dekat(k),cent(k)]=kmeanku(data,kelas)
end
```

LAMPIRAN III

Proses KMeans:

A. Sebelum proses

Bus	Tegangan	Sudut
9	0,844	19,418
11	0,858	1,817
18	0,927	6,021
5	1	39,737
6	1	36,121
8	0,961	32,099
13	0,965	3,524
1	1	0
10	0,853	17,821
16	0,94	5,137
14	0,995	0,428
19	0,928	6,159
12	0,921	2,399
7	0,97	32,513
3	0,96	3,159
2	0,98	3,056
4	0,95	2,97
15	0,945	4,499
17	0,941	4,619

B. Proses *K-Means*

Iterasi 1.

Bus	jarak 1	jarak 2	jarak 3	Min	Cluster
9	20,3196	19,41863	16,36257	16,36257	3
11	37,92027	1,82254	1,244992	1,244992	3
18	45,75806	6,021443	2,965474	2,965474	3
5	0	39,737	36,68101	0	1
6	3,616	36,121	33,06501	3,616	1
8	7,6381	32,09902	29,04301	7,6381	1
13	43,26101	3,524174	0,46824	0,46824	3
1	39,737	0	3,056065	0	2
10	21,91649	17,82161	14,76555	14,76555	3
16	44,87404	5,13735	2,081384	2,081384	3
14	40,165	0,428029	2,628043	0,428029	2
19	45,89606	6,159421	3,103436	3,103436	3
12	42,13607	2,4003	0,659644	0,659644	3
7	7,224062	32,51301	29,457	7,224062	1
3	42,89602	3,159253	0,104924	0,104924	3
2	42,793	3,056065	0	0	3
4	42,70703	2,970421	0,091082	0,091082	3
15	44,23603	4,499336	1,443424	1,443424	3
17	44,35604	4,619377	1,563486	1,563486	3
			SSE	63,76039	

Hasil penentuan anggota cluster			
Bus	Tegangan	Sudut	cluster
5	1	39,737	1
6	1	36,121	1
8	0,961	32,099	1
7	0,97	32,513	1
1	1	0	2
14	0,995	0,428	2
9	0,844	19,418	3
11	0,858	1,817	3
18	0,927	6,021	3
13	0,965	3,524	3
10	0,853	17,821	3
16	0,94	5,137	3
19	0,928	6,159	3
12	0,921	2,399	3
3	0,96	3,159	3
2	0,98	3,056	3
4	0,95	2,97	3
15	0,945	4,499	3
17	0,941	4,619	3

Iterasi 2.

Bus	jarak 1	jarak 2	jarak 3	Min	Cluster
5	4,619532	39,523	33,53716	4,619532	1
6	1,003648	35,907	29,92117	1,003648	1
8	3,018578	31,88502	25,8991	3,018578	1
7	2,604531	32,29901	26,31312	2,604531	1
1	35,1175	0,214015	6,200389	0,214015	2
14	34,6895	0,214015	5,77236	0,214015	2
9	15,70011	19,20461	13,21832	13,21832	3
11	33,30073	1,609058	4,38342	1,609058	2
18	29,09655	5,807428	0,178948	0,178948	3
13	31,5935	3,31016	2,676237	2,676237	3
10	17,29699	17,60759	11,62129	11,62129	3
16	29,98053	4,923336	1,063043	1,063043	3
19	28,95855	5,945406	0,041118	0,041118	3
12	32,71856	2,186339	3,800924	2,186339	2
3	31,95851	2,945239	3,041136	2,945239	2
2	32,0615	2,842054	3,144422	2,842054	2
4	32,14752	2,756409	3,230028	2,756409	2
15	30,61852	4,285322	1,701053	1,701053	3
17	30,49853	4,405362	1,581014	1,581014	3
			SSE	56,09445	

Hasil penentuan anggota cluster			
Bus	Tegangan	Sudut	cluster
5	1	39,737	1
6	1	36,121	1
8	0,961	32,099	1
7	0,97	32,513	1
1	1	0	2
14	0,995	0,428	2
11	0,858	1,817	2
12	0,921	2,399	2
3	0,96	3,159	2
2	0,98	3,056	2
4	0,95	2,97	2
9	0,844	19,418	3
18	0,927	6,021	3
13	0,965	3,524	3
10	0,853	17,821	3
16	0,94	5,137	3
19	0,928	6,159	3
15	0,945	4,499	3
17	0,941	4,619	3

Iterasi 3.

Bus	jarak 1	jarak 2	jarak 3	Min	Cluster
5	4,619532	37,76146	31,33736	4,619532	1
6	1,003648	34,14546	27,72137	1,003648	1
8	3,018578	30,12343	23,69929	3,018578	1
7	2,604531	30,53743	24,11331	2,604531	1
1	35,1175	1,976154	8,400151	1,976154	2
14	34,6895	1,548169	7,972123	1,548169	2
11	33,30073	0,184339	6,583022	0,184339	2
12	32,71856	0,424562	6,000751	0,424562	2
3	31,95851	1,183456	5,240919	1,183456	2
2	32,0615	1,080791	5,344111	1,080791	2
4	32,14752	0,994431	5,429845	0,994431	2
9	15,70011	17,44276	11,0185	11,0185	3
18	29,09655	4,045506	2,378768	2,378768	3
13	31,5935	1,548483	4,875978	1,548483	2
10	17,29699	15,84574	9,421473	9,421473	3
16	29,98053	3,161451	3,262825	3,161451	2
19	28,95855	4,183497	2,240773	2,240773	3
15	30,61852	2,523438	3,900844	2,523438	2
17	30,49853	2,643451	3,780821	2,643451	2
			SSE	53,57453	

Hasil penentuan anggota cluster			
Bus	Tegangan	Sudut	cluster
5	1	39,737	1
6	1	36,121	1
8	0,961	32,099	1
7	0,97	32,513	1
1	1	0	2
14	0,995	0,428	2
11	0,858	1,817	2
12	0,921	2,399	2
3	0,96	3,159	2
2	0,98	3,056	2
4	0,95	2,97	2
13	0,965	3,524	2
16	0,94	5,137	2
15	0,945	4,499	2
17	0,941	4,619	2
9	0,844	19,418	3
18	0,927	6,021	3
10	0,853	17,821	3
19	0,928	6,159	3

Iterasi 4.

Bus	jarak 1	jarak 2	jarak 3	Min	Cluster
5	4,619532	36,86358	27,38248	4,619532	1
6	1,003648	33,24758	23,76651	1,003648	1
8	3,018578	29,22555	19,74438	3,018578	1
7	2,604531	29,63955	20,15842	2,604531	1
1	35,1175	2,873882	12,35526	2,873882	2
14	34,6895	2,44586	11,92723	2,44586	2
11	33,30073	1,060492	10,53779	1,060492	2
12	32,71856	0,475368	9,955805	0,475368	2
3	31,95851	0,285705	9,196032	0,285705	2
2	32,0615	0,184921	9,299205	0,184921	2
4	32,14752	0,096547	9,384955	0,096547	2
13	31,5935	0,650708	8,831086	0,650708	2
16	29,98053	2,26357	7,217937	2,26357	2
15	30,61852	1,625555	7,855957	1,625555	2
17	30,49853	1,745571	7,735932	1,745571	2
9	15,70011	16,54489	7,063387	7,063387	3
18	29,09655	3,147633	6,33387	3,147633	2
10	17,29699	14,94786	5,466362	5,466362	3
19	28,95855	3,285622	6,195879	3,285622	2
			SSE	43,91747	

Hasil penentuan anggota cluster			
Bus	Tegangan	Sudut	cluster
5	1	39,737	1
6	1	36,121	1
8	0,961	32,099	1
7	0,97	32,513	1
1	1	0	2
14	0,995	0,428	2
11	0,858	1,817	2
12	0,921	2,399	2
3	0,96	3,159	2
2	0,98	3,056	2
4	0,95	2,97	2
13	0,965	3,524	2
16	0,94	5,137	2
15	0,945	4,499	2
17	0,941	4,619	2
18	0,927	6,021	2
19	0,928	6,159	2
9	0,844	19,418	3
10	0,853	17,821	3

Iterasi 5.

Bus	jarak 1	jarak 2	jarak 3	Min	Cluster
5	4,619532	36,36873	21,11804	4,619532	1
6	1,003648	32,75274	17,50216	1,003648	1
8	3,018578	28,7307	13,47997	3,018578	1
7	2,604531	29,1447	13,89403	2,604531	1
1	35,1175	3,368726	18,62012	3,368726	2
14	34,6895	2,940701	18,19209	2,940701	2
11	33,30073	1,553854	16,8025	1,553854	2
12	32,71856	0,969654	16,22066	0,969654	2
3	31,95851	0,209716	15,4609	0,209716	2
2	32,0615	0,314054	15,56406	0,314054	2
4	32,14752	0,39832	15,64983	0,39832	2
13	31,5935	0,156738	15,09595	0,156738	2
16	29,98053	1,768706	13,48281	1,768706	2
15	30,61852	1,130694	14,12083	1,130694	2
17	30,49853	1,250706	14,00081	1,250706	2
18	29,09655	2,652767	12,59874	2,652767	2
19	28,95855	2,790756	12,46075	2,790756	2
9	15,70011	16,05002	0,798513	0,798513	3
10	17,29699	14,453	0,798513	0,798513	3
			SSE	32,34871	

Hasil penentuan anggota cluster			
Bus	Tegangan	Sudut	cluster
5	1	39,737	1
6	1	36,121	1
8	0,961	32,099	1
7	0,97	32,513	1
1	1	0	2
14	0,995	0,428	2
11	0,858	1,817	2
12	0,921	2,399	2
3	0,96	3,159	2
2	0,98	3,056	2
4	0,95	2,97	2
13	0,965	3,524	2
16	0,94	5,137	2
15	0,945	4,499	2
17	0,941	4,619	2
18	0,927	6,021	2
19	0,928	6,159	2
9	0,844	19,418	3
10	0,853	17,821	3

LAMPIRAN IV
PROGRAM *ANFIS PSO*-GA

```
function Pbest_GA=GA_FUNGSI(X,y,Kelas,Pbest_posisi_awal,n,Psilang,Pp
indah silang,fitness_sekarang,ty)
Pbest_posisi=Pbest_posisi_awal;
[baris,kolom]=size(X);
%f=kolom^2;
for t=1:n
fitness_skr(t)=fitness_sekarang(t);
end
fitness_sekarang=fitness_skr;
%% tentukan Pbest dan Gbest
Pbest_fitness = fitness_sekarang;
[Gbest_fitness,g] = min(Pbest_fitness);

%% proses seleksi
[Pbest_copy,iterasi_mulai]=fungsiGA_seleksi(n,g,Pbest_posisi);

%% proses pemilihan induk
for i=1:n
    FIT(i)=1/(1+fitness_sekarang(i));

end
[mim,in]=min(FIT);
for i=1:n
fn(i)=FIT(i)-mim;
end
fkn(1)=fn(1);
for k=2:n
         fkn(k)= fkn(k-1)+fn(k);
end
fknmax=max(fkn);
        for a=1:n
            nfnk(a)=fkn(a)/fknmax;
        end
bapak=fungsiGA_kolas3(nfnk,n);
ibu=fungsiGA_kolas3(nfnk,n);
%kawin silang
Pbest_copy=fungsiGA_kawinsilang(Pbest_copy,Pbest_posisi,iterasi_mula
i,Psilang,n,bapak,ibu);

%Pindah silang
Pbest_copy=fungsiGA_pindah silang(Pbest_copy,n,Ppindah silang,ty);

Pbest_GA=Pbest_copy;
```

```
hasil=Gbest_fitness;
Gbest_fitnessGA=Gbest_fitness;
[Jbest_min,I] = min(fitness_sekarang); % minimum fitness
                Pbest_GA(:,I); % best solution
individu_baikGA=Pbest_GA(:,I);
%% end program
end
```

TESTING PROGRAM *ANFIS*

```
%% in ANFIS
p=an;

 nOutput=numel(fis.output);
    for i=1:nOutput
        nMF=numel(fis.output(i).mf);
        for j=1:nMF
            k=numel(fis.output(i).mf(j).params);
            fis.output(i).mf(j).params=p(1:k);
            p(1:k)=[];
        end
    end

%}
%% ----------------------------

[baris,kolom]=size(target_testing);

   out=evalfis(data_testing,fis);
   angka=scan(target_testing);

%%
SEE=0;
for k=1:baris
    SEE=SEE+(abs(target_testing(k)-out(k)));
end
Hasil=[out target_testing SEE];
End
```

Program Main *ANFIS PSO*-GA

```
load('hasil_running_final.mat')
[n,m]=size(X);
porsen_training=0.7*n;
data_training=X(1:porsen_training,:);
```

```
data_testing=X(porsen_training+1:end,:);
target_training=Y_P(1:porsen_training,:);
target_testing1=Y_P(porsen_training+1:end,:);
target_testing2=Y_V(porsen_training+1:end,:);


[SEE,out]=testing2(data_testing,target_testing1,fis,ind_terbaik)

matrik1=[out target_testing1 abs(target_testing1-out)]
%p=Y*X(1,:);
%pest=out*X(1,:);
%matrik2=[p pest abs(pest-p)];
```

Uji Program

```
function RR=diuji(angka,yr)
[~,baris]=size(angka);
for i=1:baris
    if i<baris
    bts= angka(i)+(angka(i+1)-angka(i))/2;
       if yr < bts
        RR=angka(i);
        break
    end
    else
        RR=angka(i);
    end
end
end
```

TESTING 2

```
function [SEE,out]=testing2(data_testing,target_testing,fis,ind_terb
aik)
warning off;
p=ind_terbaik;
test_data=[data_testing target_testing];
%% get AN
 nInput=numel(fis.input);
    for i=1:nInput
        nMF=numel(fis.input(i).mf);
        for j=1:nMF
            k=numel(fis.input(i).mf(j).params);
            fis.input(i).mf(j).params=p(1:k);
            p(1:k)=[];
        end
    end
```

```
an=[];
asem=[0,0,0,0];
AN = ANFIS(test_data,fis,1,asem);

nOutput=numel(AN.output);
    for i=1:nOutput
        nMF=numel(AN.output(i).mf);
        for j=1:nMF
            an=[an AN.output(i).mf(j).params];
        end
    end
    an;

%% in ANFIS
p=an;

 nOutput=numel(fis.output);
    for i=1:nOutput
        nMF=numel(fis.output(i).mf);
        for j=1:nMF
            k=numel(fis.output(i).mf(j).params);
            fis.output(i).mf(j).params=p(1:k);
            p(1:k)=[];
        end
    end


%% ----------------------------

[baris,kolom]=size(target_testing);

   out=evalfis(data_testing,fis);

   angka=scan(target_testing);
SEE=0;
for k=1:baris
    SEE=SEE+(abs(target_testing(k)-out(k)));
end
% %%
% TP=0;
% FP=0;
% TN=0;
% FN=0;
% y=target_testing;
% angka=scan(y);
% benar=0;
```

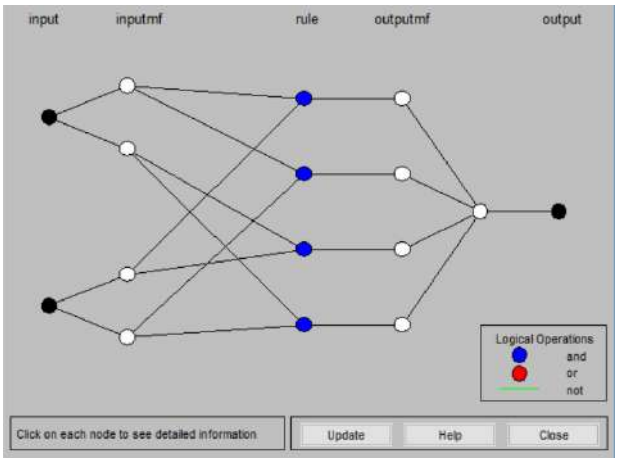
```
% for i=1:baris
% yyr(i)=diuji(angka,out(i));
% if yyr(i)==target_testing(i);
%     benar=benar+1;
% end
% end
%
% salah=baris-benar;
% [out yyr' target_testing];
% plus=1-angka(1);
% %
% if angka(1)<=0
%     yyr=yyr+plus;
%     y=y+plus;
%     angka=angka+plus;
% end
%
% [q,w]=size(angka);
% tabel=[];
% for k=1:angka(w)
%     total1=0;
%     total0=0;
%     A=0;
%     B=0;
%         for i=1:baris
%             if yyr(i)==k
%                 total0=total0+1;
%                 if yyr(i)==y(i)
%                     A=A+1;
%                 else
%                     B=B+1;
%                 end
%             end
%         end
%         tabel(k,1)=A;
%         tabel(k,2)=B;
% end
% TP=tabel(1,1);
% FP=tabel(1,2);
% for l=2:angka(w)
%     TN=TN+tabel(l,1);
%     FN=FN+tabel(l,2);
% end
% MSE=(TP+FP+TN+FN)-(TP+TN);
SEE;
end
```

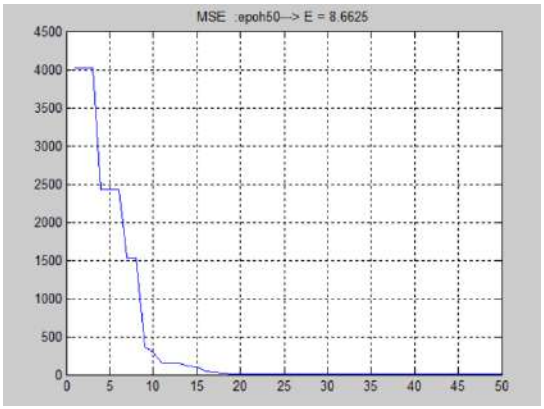
PROGRAM ESTIMASI *PSO*GA

Struktur FIS

Number of inputs = 2

Grafik training data (7 hari)



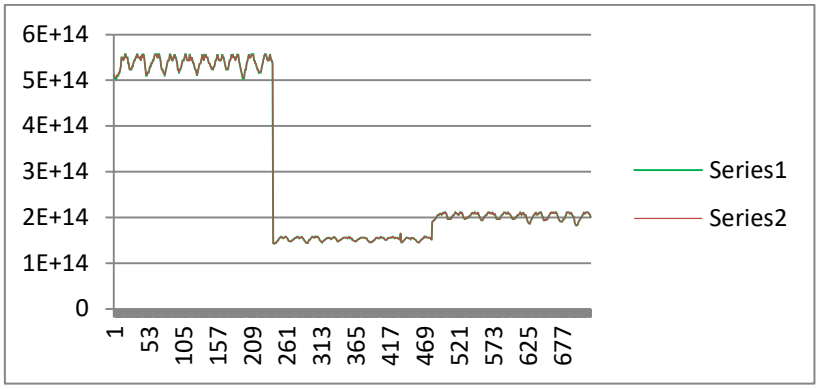


Parameter *PSO*GA *Error SEE* Training = 8.6625

```
Coefesien C1=1;
Coefesien C2=1;
Bobot inersia=1;
Porsentase kawin silang=0.7;
Persontase pindah silang=0.2;
populasi=30;
maximum iterasi=50;
```

MODEL TRANING *ANFIS-PSO*GA

A11	57.4959765558178
C11	-63.0200816421904
A12	-140.017243496395
C12	-28.7892467232851
A21	-583.179856356613
C21	-300.868594072086
A22	5339.40792767141
C22	568.252906928094



Series 1 = V estimasi

Series 2 = V actual

LAMPIRAN V
PROGRAM INTERNET oF THINK (*IoT*)

A. Program menampilkan data secara *online*

```
//#include <Software) Serial.h>
//Software) Serial espSerial =  Software) Serial(2,3);
#include "VirtuinoEsp8266_WebServer.h"
VirtuinoEsp8266_WebServer virtuino(Serial1);
const int pinADC = A5;
int sensitivitas = 66;
int nilaiadc= 00;
int teganganoffset = 2500;
double tegangan = 00;
double nilaiarus = 00;
double teganganV;
double daya;

const int pinADC1 = A6;
int sensitivitas1 = 66;
int nilaiadc1= 00;
int teganganoffset1 = 2500;
double tegangan1 = 00;
double nilaiarus1 = 00;
double teganganV1;
double daya1;

const int pinADC2 = A7;
int sensitivitas2 = 66;
int nilaiadc2= 00;
int teganganoffset2 = 2500;
double tegangan2 = 00;
double nilaiarus2 = 00;
double teganganV2;
double daya2;

const int pinADC3 = A8;
int sensitivitas3 = 66;
int nilaiadc3= 00;
int teganganoffset3 = 2500;
double tegangan3 = 00;
double nilaiarus3 = 00;
double teganganV3;
double daya3;

const int pinADC4 = A9;
int sensitivitas4 = 66;
int nilaiadc4= 00;
```

```
int teganganoffset4 = 2500;
double tegangan4 = 00;
double nilaiarus4 = 00;
double teganganV4;
double daya4;

const int pinADC5 = A10;
int sensitivitas5 = 66;
int nilaiadc5= 00;
int teganganoffset5 = 2500;
double tegangan5 = 00;
double nilaiarus5 = 00;
double teganganV5;
double daya5;

const int pinADC6 = A11;
int sensitivitas6 = 66;
int nilaiadc6= 00;
int teganganoffset6 = 2500;
double tegangan6 = 00;
double nilaiarus6 = 00;
double teganganV6;
double daya6;

const int pinADC7 = A12;
int sensitivitas7 = 66;
int nilaiadc7= 00;
int teganganoffset7 = 2500;
double tegangan7 = 00;
double nilaiarus7 = 00;
double teganganV7;
double daya7;

const int pinADC8 = A13;
int sensitivitas8 = 66;
int nilaiadc8= 00;
int teganganoffset8 = 2500;
double tegangan8 = 00;
double nilaiarus8 = 00;
double teganganV8;
double daya8;

const int pinADC9 = A14;
int sensitivitas9 = 66;
int nilaiadc9= 00;
int teganganoffset9 = 2500;
double tegangan9 = 00;
double nilaiarus9 = 00;
```

```
double teganganV9;
double daya9;

const int pinADC10 = A15;
int sensitivitas10 = 66;
int nilaiadc10= 00;
int teganganoffset10 = 2500;
double tegangan10 = 00;
double nilaiarus10 = 00;
double teganganV10;
double daya10;
void setup()
{
  virtuino.DEBUG=true;                                  // set this value TRUE to enable the serial monitor status.It is neccesary to get your esp8266 local ip
  Serial.begin(9600);
  Serial1.begin(9600);
  //espSerial.begin(9600);
  virtuino.connectESP8266_toInternet("Emon","qwertyuiop",8000);    // Set your home wifi router SSID and PASSWORD. ESP8266 will connect to Internet. Port=80
  virtuino.esp8266_setIP(192,168,43,10);
   virtuino.password="1234";
   pinMode(13,OUTPUT);
}

void data_olah(){
  nilaiadc = analogRead(pinADC);
  tegangan = (nilaiadc / 1024.0) * 5000;
  nilaiarus = ((tegangan - teganganoffset) / sensitivitas);
  teganganV= tegangan / 1000;
  daya=teganganV*nilaiarus;
  virtuino.vMemoryWrite(0,teganganV);
  virtuino.vMemoryWrite(1,nilaiarus);
  virtuino.vMemoryWrite(2,daya);
}

  void data_olah1(){
  nilaiadc1 = analogRead(pinADC1);
  tegangan1 = (nilaiadc1 / 1024.0) * 5000;
  nilaiarus1 = ((tegangan1 - teganganoffset1) / sensitivitas1);
  teganganV1= tegangan1 / 1000;
  daya1=teganganV1*nilaiarus1;
  virtuino.vMemoryWrite(3,teganganV1);
  virtuino.vMemoryWrite(4,nilaiarus1);
  virtuino.vMemoryWrite(5,daya1);
  }
```

```
void data_olah2(){
  nilaiadc2 = analogRead(pinADC2);
  tegangan2 = (nilaiadc2 / 1024.0) * 5000;
  nilaiarus2 = ((tegangan2 - teganganoffset2) / sensitivitas2);
  teganganV2= tegangan2 / 1000;
  daya2=teganganV2*nilaiarus2;
  virtuino.vMemoryWrite(6,teganganV2);
  virtuino.vMemoryWrite(7,nilaiarus2);
  virtuino.vMemoryWrite(8,daya2);
  }


void data_olah3(){
  nilaiadc3 = analogRead(pinADC3);
  tegangan3 = (nilaiadc3 / 1024.0) * 5000;
  nilaiarus3 = ((tegangan3 - teganganoffset3) / sensitivitas3);
  teganganV3= tegangan3 / 1000;
  daya3=teganganV3*nilaiarus3;
  virtuino.vMemoryWrite(9,teganganV3);
  virtuino.vMemoryWrite(10,nilaiarus3);
  virtuino.vMemoryWrite(11,daya3);
}

void data_olah4(){
  nilaiadc4 = analogRead(pinADC4);
  tegangan4 = (nilaiadc4 / 1024.0) * 5000;
  nilaiarus4 = ((tegangan4 - teganganoffset4) / sensitivitas4);
  teganganV4= tegangan4 / 1000;
  daya4=teganganV4*nilaiarus4;
  virtuino.vMemoryWrite(12,teganganV4);
  virtuino.vMemoryWrite(13,nilaiarus4);
  virtuino.vMemoryWrite(14,daya4);
}

void data_olah5(){
  nilaiadc5 = analogRead(pinADC5);
  tegangan5 = (nilaiadc5 / 1024.0) * 5000;
  nilaiarus5 = ((tegangan5 - teganganoffset5) / sensitivitas5);
  teganganV5= tegangan5 / 1000;
  daya5=teganganV5*nilaiarus5;
  virtuino.vMemoryWrite(15,teganganV5);
  virtuino.vMemoryWrite(16,nilaiarus5);
  virtuino.vMemoryWrite(17,daya5);

}
void data_olah6(){
  nilaiadc6 = analogRead(pinADC6);
  tegangan6 = (nilaiadc6 / 1024.0) * 5000;
```

```
  nilaiarus6 = ((tegangan6 - teganganoffset6) / sensitivitas6);
  teganganV6= tegangan6 / 1000;
  daya6=teganganV6*nilaiarus6;
  virtuino.vMemoryWrite(18,teganganV6);
  virtuino.vMemoryWrite(19,nilaiarus6);
  virtuino.vMemoryWrite(20,daya6);
}

void data_olah7(){
  nilaiadc7 = analogRead(pinADC7);
  tegangan7 = (nilaiadc7 / 1024.0) * 5000;
  nilaiarus7 = ((tegangan7 - teganganoffset7) / sensitivitas7);
  teganganV7= tegangan / 1000;
  daya7=teganganV7*nilaiarus7;
  virtuino.vMemoryWrite(21,teganganV7);
  virtuino.vMemoryWrite(22,nilaiarus7);
  virtuino.vMemoryWrite(23,daya7);
}

void data_olah8(){
  nilaiadc8 = analogRead(pinADC8);
  tegangan8 = (nilaiadc8 / 1024.0) * 5000;
  nilaiarus8 = ((tegangan8 - teganganoffset8) / sensitivitas8);
  teganganV8= tegangan8 / 1000;
  daya8=teganganV8*nilaiarus8;
  virtuino.vMemoryWrite(24,teganganV8);
  virtuino.vMemoryWrite(25,nilaiarus8);
  virtuino.vMemoryWrite(26,daya8);
}
void data_olah9(){
  nilaiadc9 = analogRead(pinADC9);
  tegangan9 = (nilaiadc9 / 1024.0) * 5000;
  nilaiarus9 = ((tegangan9 - teganganoffset9) / sensitivitas9);
  teganganV9= tegangan9 / 1000;
  daya9=teganganV9*nilaiarus9;
  virtuino.vMemoryWrite(27,teganganV9);
  virtuino.vMemoryWrite(28,nilaiarus9);
  virtuino.vMemoryWrite(29,daya9);
}
void data_olah10(){
  nilaiadc10 = analogRead(pinADC10);
  tegangan10 = (nilaiadc10 / 1024.0) * 5000;
  nilaiarus10 = ((tegangan10 - teganganoffset10) / sensitivitas10);
  teganganV10= tegangan10 / 1000;
  daya10=teganganV10*nilaiarus10;
  virtuino.vMemoryWrite(30,teganganV10);
  virtuino.vMemoryWrite(31,nilaiarus10);
  virtuino.vMemoryWrite(32,daya10);
```

```
}
void loop(){
  virtuino.run();
  data_olah();
  data_olah1();
  data_olah2();
  data_olah3();
  data_olah4();
  data_olah5();
  data_olah6();
  data_olah7();
  data_olah8();
  data_olah9();
  data_olah10();

  virtuino.vDelay(1000);
}
```

B. lampiran Dokumen Paten dan hak cipta *software)*

HAKI:

REPUBLIK INDONESIA
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

SURAT PENCATATAN
CIPTAAN

Dalam rangka pelindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan	:	EC00202000406, 6 Januari 2020
Pencipta		
Nama	:	**Rony Sulistyowati, S.T., M.T, Prof. Ir.Mochamad Ashari, M.Eng., Ph.D, , dkk**
Alamat	:	Jl Wiguna Selatan XV / 2, Gunung Anyar Tambak, Gunung Anyar, Surabaya, Jawa Timur, -
Kewarganegaraan	:	Indonesia
Pemegang Hak Cipta		
Nama	:	**Institut Teknologi Sepuluh Nopember**
Alamat	:	Kampus ITS Sukolilo, Surabaya, Jawa Timur, 60111
Kewarganegaraan	:	Indonesia
Jenis Ciptaan	:	**Program Komputer**
Judul Ciptaan	:	**Aplikasi Kontrol Menggunakan Arduuno Untuk Internet Of Things (IOT)**
Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia	:	5 Januari 2020, di Surabaya
Jangka waktu pelindungan	:	Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.
Nomor pencatatan	:	000174886

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.
Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

a.n. MENTERI HUKUM DAN HAK ASASI MANUSIA
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL

Dr. Freddy Harris, S.H., LL.M., ACCS.
NIP. 196611181994031001

FORMULIR PERMOHONAN PENDAFTARAN PATEN INDONESIA

APPLICATION FORM OF PATENT REGISTRATION OF INDONESIA

Data Permohonan (*Application*)

Nomor Permohonan *Number of Application*	: P00202000082	Tanggal Permohonan *Date of Submission*	: 06-JAN-20
Jenis Permohonan *Type of Application*	: PATEN	Jumlah Klaim *Total Claim*	: 5
		Jumlah halaman *Total page*	: 5
Judul *Title*	: HARDWARE MONITORING INTERNET OF THINGS		
Abstrak *Abstract*	: HARDDWARE MONITORING INTERNET OF THINGS(IOT) Internet Of Things (IOT) merupakan peralatan monitoring yang di gunakan medeteksi besaran nilaitegangan,arus,dan daya di dalam jaringan listrik. Pada desain hardware di tempatkan sensor pada lokasi yang sudah di tentukan dan disediakan beberapa saluran input. Selanjutnya mensimulasikan data distribusi penyulang dengan mengubah beberapa nilai pada titik – titik trigger. Nilai yang di trigger dari luar berupa beban (load) . Beban di simulasikan menggunakan rangkaian resistor, rangkaian resistor akan mengirimkan tegangan pada mikrokontroller yang akan mengubah menjadi data digital dan akan di kirim melalui saluran usb ke komputer. Data digital tersebut merupakan nilai input beban pada simulator. Input beban simulator ada 11 titik. Karena banyak input beban yang disimulasikan, maka mikrokontroller berfungsi sebagai multiplexer yang akan mengirimkan data – data input beban secara bergantian ke simulator. Pada Proses simulasi, di monitoring nilai tegangan magnitude dan arus dan daya pada output sensor-sensor tersebut . Simulasi analisa sistem tenaga semi realtime ini dibandingkan perubahan daya hasil di komputer dengan kondisi riil di lapangan.		

Permohonan PCT (*PCT Application*)

Nomor PCT *PCT Number*	:	Nomor Publikasi *Publication Number*	:
Tanggal PCT *PCT Date*	:	Tanggal Publikasi *Publication Date*	:

Pemohon (*Applicant*)

Name ***(Name)***	**Alamat** ***(Addresss)***	**Surel/Telp** ***(Email/Phone)***
Institut Teknologi Sepuluh Nopember	Kampus ITS Sukolilo, Surabaya 60111	0315994251 humas@its.ac.id

Penemu (*Inventor*)

Nama ***(Name)***	**Warganegara** ***(Nationality)***	**Alamat** ***(Address)***	**Surel/Telp.** ***(Email/Phone)***
Riny Sulistyowati. S.T., M.T	Indonesia	Jl Wiguna Selatan XV / 2, Gunung Anyar Tambak, Gunung Anyar – Kota Surabaya	shaffer.ricky@gmail.com 0
Prof. Ir.Mochamad Ashari, M.Eng., Ph.D	Indonesia	Dusun Pilang RT 04, RW 02, Pilang, Wonoayu – Sidoarjo	shaffer.ricky@gmail.com 0
Dedet Candra Riawan, ST, M.Eng, Ph.D	Indonesia	Perum Sukolilo Park Regency A-39, Keputih, Sukolilo – Surabaya	shaffer.ricky@gmail.com 0

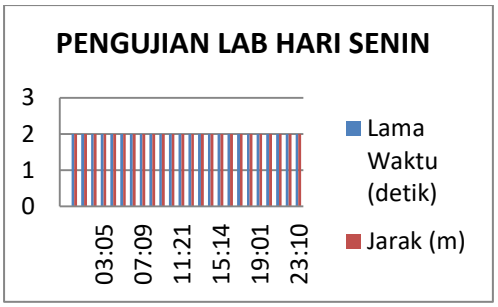
LAMPIRAN VI

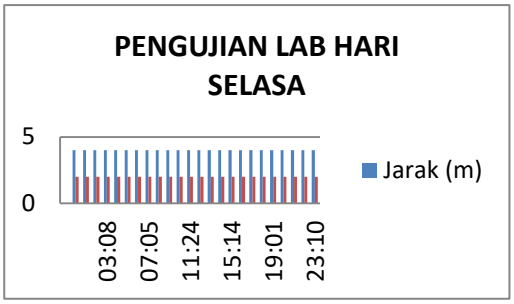
Data Monitoring SMS GAteway

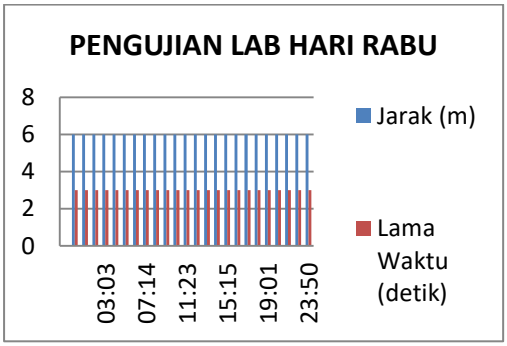
Waktu	Lama Waktu (detik)	Jarak (m)	Waktu	Jarak (m)	Lama Waktu (detik)	Waktu	Jarak (m)	Lama Waktu (detik)	Waktu	Jarak (m)	Lama Waktu (detik)
00:01	2	2	00:02	4	2	00:05	6	3	00:08	8	4
01:03	2	2	01:02	4	2	01:01	6	3	01:03	8	4
02:02	2	2	02:04	4	2	02:05	6	3	02:02	8	4
03:05	2	2	03:08	4	2	03:03	6	3	03:05	8	4
04:03	2	2	04:03	4	2	04:06	6	3	04:03	8	4
05:07	2	2	05:07	4	2	05:08	6	3	05:07	8	4
06:11	2	2	06:01	4	2	06:10	6	3	06:11	8	4
07:09	2	2	07:05	4	2	07:14	6	3	07:09	8	4
08:05	2	2	08:02	4	2	08:05	6	3	08:05	8	4
09:12	2	2	09:11	4	2	09:11	6	3	09:12	8	4
10:24	2	2	10:10	4	2	10:10	6	3	10:24	8	4
11:21	2	2	11:24	4	2	11:23	6	3	11:21	8	4
12:01	2	2	12:01	4	2	12:30	6	3	12:01	8	4
13:43	2	2	13:43	4	2	13:31	6	3	13:43	8	4
14:12	2	2	14:12	4	2	14:30	6	3	14:12	8	4
15:14	2	2	15:14	4	2	15:15	6	3	15:14	8	4
16:18	2	2	16:18	4	2	16:18	6	3	16:18	8	4
17:20	2	2	17:20	4	2	17:10	6	3	17:20	8	4
18:21	2	2	18:21	4	2	18:27	6	3	18:21	8	4
19:01	2	2	19:01	4	2	19:01	6	3	19:01	8	4
20:01	2	2	20:08	4	2	20:10	6	3	20:08	8	4
21:14	2	2	21:14	4	2	21:09	6	3	21:14	8	4
22:18	2	2	22:18	4	2	22:39	6	3	22:18	8	4
23:10	2	2	23:10	4	2	23:50	6	3	23:10	8	4

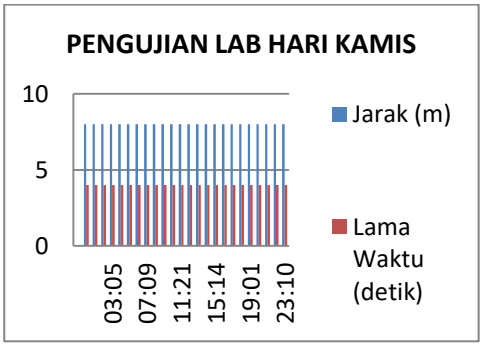
\Waktu	Jarak (m)	Lama Waktu (detik)	Waktu	Jarak (m)	Lama Waktu (detik)	Waktu	Jarak (m)	Lama Waktu (detik)
00:02	10	4	00:11	12	5	00:10	14	5
01:04	10	4	01:10	12	5	01:15	14	5
02:09	10	4	02:22	12	5	02:23	14	5
03:15	10	4	03:25	12	5	03:12	14	5
04:13	10	4	04:23	12	5	04:14	14	5
05:17	10	4	05:27	12	5	05:15	14	5
06:01	10	4	06:21	12	5	06:12	14	5
07:19	10	4	07:29	12	5	07:16	14	5
08:15	10	4	08:25	12	5	08:02	14	5
09:02	10	4	09:22	12	5	09:01	14	5
10:14	10	4	10:13	12	5	10:10	14	5
11:01	10	4	11:11	12	5	11:12	14	5
12:11	10	4	12:21	12	5	12:02	14	5
13:03	10	4	13:23	12	5	13:19	14	5
14:02	10	4	14:22	12	5	14:15	14	5

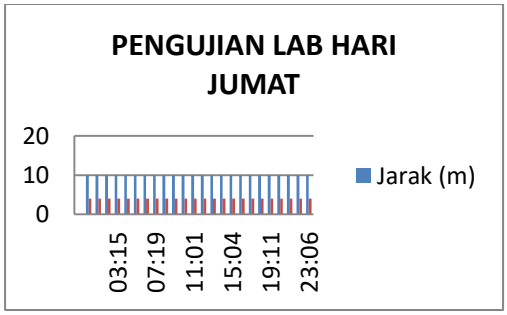
15:04	10	4	15:24	12	5	15:12	14	5
16:08	10	4	16:28	12	5	16:10	14	5
17:10	10	4	17:20	12	5	17:12	14	5
18:11	10	4	18:21	12	5	18:19	14	5
19:11	10	4	19:21	12	5	19:01	14	5
20:18	10	4	20:28	12	5	20:08	14	5
21:04	10	4	21:24	12	5	21:14	14	5
22:28	10	4	22:28	12	5	22:18	14	5
23:06	10	4	23:20	12	5	23:10	14	5

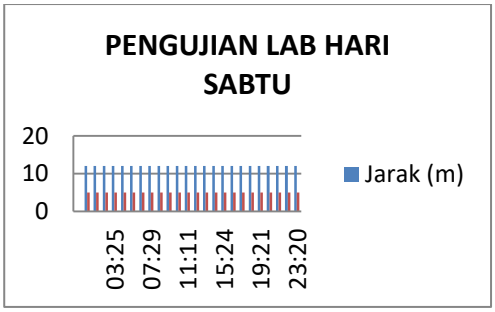


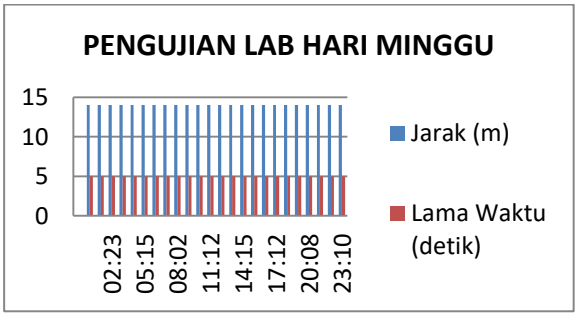






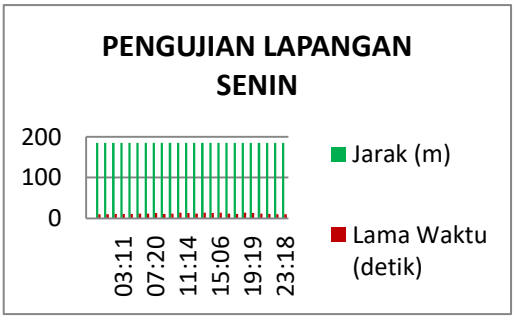


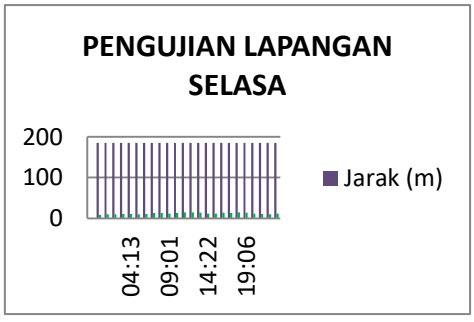


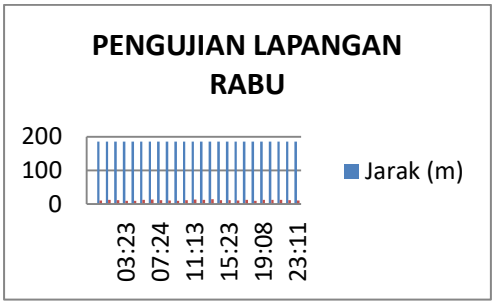


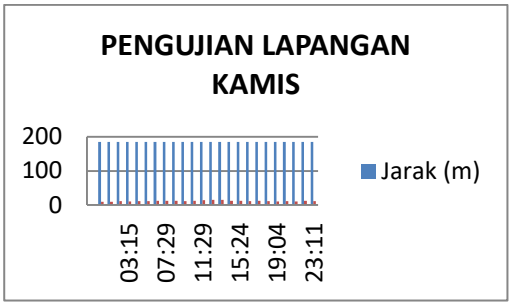
C. Data skala Uji lapangan

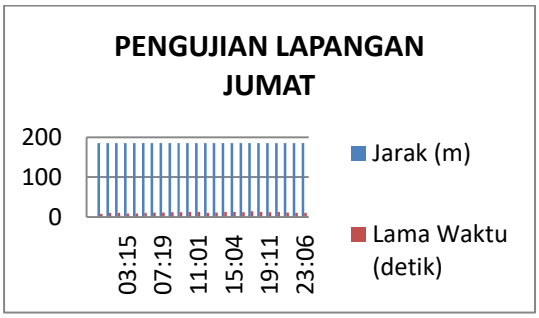
Waktu	Jarak (m)	Lama Waktu (detik)	Waktu	Lama Waktu (detik)	Waktu	Lama Waktu (detik)	Waktu	Lama Waktu (detik)	Waktu	Lama Waktu (detik)	Waktu	Lama Waktu (detik)	Waktu	Lama Waktu (detik)
00:07	185	10	00:12	9	00:15	11	00:10	10	00:10	10	00:10	12	00:30	10
01:13	185	10	01:12	10	01:11	13	01:13	10	01:13	10	01:12	12	01:12	10
02:15	185	11	02:14	10	02:25	12	02:22	12	02:22	12	02:24	10	02:13	11
03:11	185	11	03:18	11	03:23	10	03:15	11	03:15	11	03:21	10	03:22	10
04:06	185	11	04:13	11	04:16	10	04:23	12	04:23	12	04:14	13	04:34	12
05:01	185	12	05:17	10	05:28	13	05:27	12	05:27	12	05:07	12	05:25	13
06:13	185	12	06:11	11	06:30	14	06:31	13	06:31	13	06:01	12	06:10	13
07:20	185	13	07:15	13	07:24	12	07:29	13	07:29	13	07:19	11	07:26	12
08:12	185	11	08:12	13	08:25	11	08:15	13	08:15	13	08:25	10	08:02	15
09:21	185	12	09:01	12	09:21	10	09:32	12	09:32	12	09:32	12	09:11	14
10:10	185	14	10:01	13	10:20	12	10:21	13	10:21	13	10:33	13	10:20	13
11:14	185	13	11:14	15	11:13	14	11:29	14	11:29	14	11:01	12	11:11	14
12:27	185	12	12:11	15	12:20	13	12:31	15	12:31	15	12:03	10	12:22	12
13:15	185	14	13:13	14	13:21	15	13:13	15	13:13	15	13:01	11	13:03	13
14:23	185	13	14:22	12	14:21	12	14:22	13	14:22	13	14:10	14	14:30	10
15:06	185	14	15:24	12	15:23	12	15:24	13	15:24	13	15:15	13	15:20	13
16:18	185	12	16:28	14	16:11	11	16:18	12	16:18	12	16:31	12	16:12	12
17:27	185	11	17:11	13	17:13	13	17:22	13	17:22	13	17:21	14	17:23	13
18:17	185	14	18:08	15	18:17	10	18:17	12	18:17	12	18:30	15	18:29	11
19:19	185	13	19:06	14	19:08	13	19:04	11	19:04	11	19:21	12	19:03	10
20:07	185	12	20:05	12	20:13	13	20:02	12	20:02	12	20:29	14	20:03	14
21:10	185	11	21:11	11	21:14	13	21:15	11	21:15	11	21:22	14	21:04	13
22:13	185	10	22:13	10	22:21	12	22:12	13	22:12	13	22:15	12	22:08	13
23:18	185	10	23:14	12	23:11	11	23:11	12	23:11	12	23:12	13	23:17	11

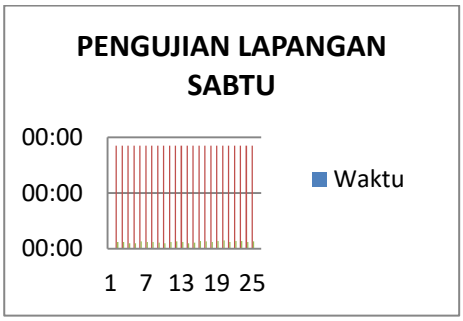


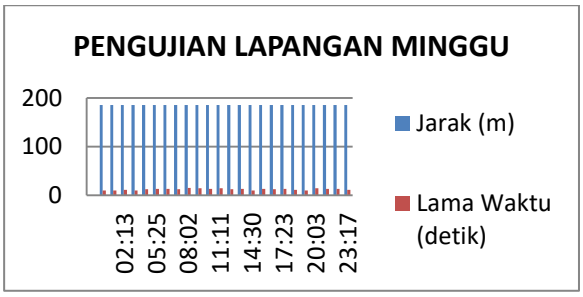



PENGUJIAN LAPANGAN RABU
200
100
0
03:23
07:24
11:13
15:23
19:08
23:11
Jarak (m)


PENGUJIAN LAPANGAN KAMIS
200
100
0
03:15
07:29
11:29
15:24
19:04
23:11
Jarak (m)


PENGUJIAN LAPANGAN JUMAT
200
100
0
03:15
07:19
11:01
15:04
19:11
23:06
Jarak (m)
Lama Waktu (detik)


PENGUJIAN LAPANGAN SABTU
00:00
00:00
00:00
1 7 13 19 25
Waktu


PENGUJIAN LAPANGAN MINGGU
200
100
0
02:13
05:25
08:02
11:11
14:30
17:23
20:03
23:17
Jarak (m)
Lama Waktu (detik)