

**IMPLEMENTASI KONTROL POSISI BERBASIS WEB PADA PANEL SURYA MENGGUNAKAN METODE *FUZYY PROPORTIONAL INTEGRAL AND DERIVATIVE  
(FPID*) CONTROLLER**

**SKRIPSI**

oleh

**Nila Choirotun Nisa’**

**NIM 152410101153**

**PROGRAM STUDI SISTEM INFORMASI**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS JEMBER**

**2019**



**IMPLEMENTASI KONTROL POSISI BERBASIS WEB PADA PANEL SURYA MENGGUNAKAN METODE *FUZYY PROPORTIONAL INTEGRAL AND DERIVATIVE  
(FPID*) CONTROLLER**

**SKRIPSI**

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat

untuk menyelesaikan Pendidikan Sarjana (S1) Program Studi Sistem Informasi

Universitas Jember dan mencapai gelar Sarjana Komputer

oleh

**Nila Choirotun Nisa’**

**NIM 152410101153**

**PROGRAM STUDI SISTEM INFORMASI**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS JEMBER**

**2019**

# DAFTAR TABEL

[Tabel 2.1 Karakteristik PID 10](#_Toc23495954)

[Tabel 4.1 Kebutuhan Fungsional 24](#_Toc23495955)

[Tabel 4.2 Kebutuhan Non Fungsional 25](#_Toc23495956)

[Tabel 4.3 Definisi Tugas 26](#_Toc23495957)

[Tabel 4.4 Deskripsi Usecase 26](#_Toc23495958)

# DAFTAR GAMBAR

[Gambar 2.1 skema PID 9](#_Toc23496124)

[Gambar 3.1 Tahapan Penelitian 12](#_Toc23496125)

[Gambar 3.2 Penempatan Sensor LDR 13](file:///D:\1.%20SKRIPSIIIIII\301019%20SKRIPSI%20Nila%2015153(2)%20(2).docx#_Toc23496126)

[Gambar 3.3 Range error V 15](#_Toc23496127)

[Gambar 3.4 Range error H 15](file:///D:\1.%20SKRIPSIIIIII\301019%20SKRIPSI%20Nila%2015153(2)%20(2).docx#_Toc23496128)

[Gambar 3.5 Flowchart 16](#_Toc23496129)

[Gambar 3.6 Drajat Keanggotaan 17](#_Toc23496130)

[Gambar 3.7 Fuzzifikasi 20](#_Toc23496131)

[Gambar 4.1 Business Process 25](#_Toc23496132)

[Gambar 4.2 Usecase Diagram 27](#_Toc23496133)

[Gambar 4.25 ERD 36](#_Toc23496134)

[Gambar 4.26 Class Diagram 37](#_Toc23496135)

[Gambar 6.1 Tahapan Penelitian 46](file:///D:\1.%20SKRIPSIIIIII\301019%20SKRIPSI%20Nila%2015153(2)%20(2).docx#_Toc23496136)

[Gambar 6.2 Range error H 46](file:///D:\1.%20SKRIPSIIIIII\301019%20SKRIPSI%20Nila%2015153(2)%20(2).docx#_Toc23496137)

[Gambar 6.3. tampak atas 46](file:///D:\1.%20SKRIPSIIIIII\301019%20SKRIPSI%20Nila%2015153(2)%20(2).docx#_Toc23496138)

[Gambar 6.4. tampak samping 46](file:///D:\1.%20SKRIPSIIIIII\301019%20SKRIPSI%20Nila%2015153(2)%20(2).docx#_Toc23496139)

[Gambar 6.5 Grafik fuzzy 46](file:///D:\1.%20SKRIPSIIIIII\301019%20SKRIPSI%20Nila%2015153(2)%20(2).docx#_Toc23496140)

# DAFTAR ISI

[DAFTAR TABEL ii](#_Toc23496141)

[DAFTAR GAMBAR iii](#_Toc23496142)

[DAFTAR ISI iv](#_Toc23496143)

[BAB 1. PENDAHULUAN 1](#_Toc23496144)

[1.1 Latar Belakang 1](#_Toc23496145)

[1.2 Rumusan Masalah 3](#_Toc23496146)

[1.3 Tujuan Penelitian 3](#_Toc23496147)

[1.4 Manfaat Penelitian 3](#_Toc23496148)

[1.5 Batasan Masalah 4](#_Toc23496149)

[BAB 2. TINJAUAN PUSTAKA 5](#_Toc23496150)

[2.1 Penelitian Terdahulu 5](#_Toc23496151)

[2.2 Panel Surya 7](#_Toc23496152)

[2.3 Light Dependent Resistor (LDR) 7](#_Toc23496153)

[2.4 *Solar Tracker* 8](#_Toc23496154)

[2.5 *Fuzyy Logic Control* 8](#_Toc23496155)

[2.5.1 *Fuzzifikasi* 8](#_Toc23496156)

[2.5.2 *Control Rule Base* 8](#_Toc23496157)

[2.5.3 *Defuzzifikasi* 9](#_Toc23496158)

[2.6 Proportional Integral Derivative (PID) 9](#_Toc23496159)

[2.7 Database 10](#_Toc23496160)

[2.8 Bahasa Pemrograman 10](#_Toc23496161)

[2.9 Internet of Things (IOT) 11](#_Toc23496162)

[BAB 3. METODOLOGI PENELITIAN 12](#_Toc23496163)

[3.1 Objek Penelitian 12](#_Toc23496164)

[3.2 Tempat dan Waktu Penelitian 12](#_Toc23496165)

[3.3 Tahapan Penelitian 12](#_Toc23496166)

[3.3.1 Analisa Kebutuhan 13](#_Toc23496167)

[3.3.2 Desain sistem 13](#_Toc23496168)

[3.3.3 Impelementasi Desain *Hardware* 13](#_Toc23496169)

[3.3.4 Pengambilan Data Sensor 13](#_Toc23496170)

[3.3.5 Implementasi Metode *Fuzyy* 14](#_Toc23496171)

[3.3.6 Implementasi *Solar* *Tracker* Tanpa Metode *Fuzyy* 21](#_Toc23496172)

[3.3.7 Implementasi *Solar* *Tracker* Dengan Metode *Fuzyy* 22](#_Toc23496173)

[3.3.8 Implementasi Metode PID 22](#_Toc23496174)

[3.3.9 Pembuatan Web Sistem 23](#_Toc23496175)

[BAB 4. ANALISA DAN PERANCANGAN SISTEM 24](#_Toc23496176)

[4.1 Analisa Kebutuhan 24](#_Toc23496177)

[4.1.1 Kebutuhan Fumgsional 24](#_Toc23496178)

[4.1.2 Kebutuhan Non Fumgsional 24](#_Toc23496179)

[4.2 *Business Process* 25](#_Toc23496180)

[4.3 *Usecase Diagram* 26](#_Toc23496181)

[4.4 Skenario 28](#_Toc23496182)

[4.4.1 Skenario *Log In* 28](#_Toc23496183)

[4.4.2 Skenario Tambah *User* 28](#_Toc23496184)

[4.4.3 Edit Us*er* 28](#_Toc23496185)

[4.4.4 History Log In 28](#_Toc23496186)

[4.4.5 Lihat Data *History Tracker* 29](#_Toc23496187)

[4.4.6 Lihat Data History Aktuator 29](#_Toc23496188)

[4.4.7 Lihat *Grafik Sensor* 29](#_Toc23496189)

[4.4.8 Lihat Data Nilai *Setpoint* 29](#_Toc23496190)

[4.4.9 Lihat Grafik *Tracker* 29](#_Toc23496191)

[4.4.10 Lihat Grafik Aktuator 30](#_Toc23496192)

[4.4.11 Log out 30](#_Toc23496193)

[4.5 Activity Diagram 30](#_Toc23496194)

[4.5.1 Log in 30](#_Toc23496195)

[4.5.2 Tambah user 30](#_Toc23496196)

[4.5.3 Edit *user* 31](#_Toc23496197)

[4.5.4 Lihat Data History Aktuator 31](#_Toc23496198)

[4.5.5 Lihat Data *History Tracker* 31](#_Toc23496199)

[4.5.6 Lihat Grafik Sensor 31](#_Toc23496200)

[4.5.7 Lihat Data Nilai Setpoint 32](#_Toc23496201)

[4.5.8 Lihat Grafik *Tracker* 32](#_Toc23496202)

[4.5.9 Lihat Grafik Aktuator 32](#_Toc23496203)

[4.5.10 Log out 32](#_Toc23496204)

[4.6 Sequence Diagram 32](#_Toc23496205)

[4.6.1 Log In 32](#_Toc23496206)

[4.6.2 Tambah User 33](#_Toc23496207)

[4.6.3 Edit user 33](#_Toc23496208)

[4.6.4 History Login 33](#_Toc23496209)

[4.6.5 Lihat Data *History* Aktuator 33](#_Toc23496210)

[4.6.6 Lihat Data *History Tracker* 34](#_Toc23496211)

[4.6.7 Lihat Grafik Sensor 34](#_Toc23496212)

[4.6.8 Lihat Data Nilai *Setpoint* 34](#_Toc23496213)

[4.6.9 Lihat Grafik *Tracker* 35](#_Toc23496214)

[4.6.10 Lihat GrafikAktuator 35](#_Toc23496215)

[4.6.11 Log out 35](#_Toc23496216)

[4.7 *Entity Relationship Diagram* (ERD) 36](#_Toc23496217)

[4.8 *Class Diagram* 36](#_Toc23496218)

[4.9 Desain *User Interface (UI)* 37](#_Toc23496219)

[4.9.1 User Interface Log In 37](#_Toc23496220)

[4.9.2 User Interface Tambah User 37](#_Toc23496221)

[4.9.3 Edit User 37](#_Toc23496222)

[4.9.4 History Login 37](#_Toc23496223)

[4.10 Pengujian Sistem 38](#_Toc23496224)

[4.10.1 Pengujian *Solar Tracker* Tanpa Metode *Fuzzy* 38](#_Toc23496225)

[4.10.2 Pengujian *Solar Tracker* dengan Metode *Fuzzy* 39](#_Toc23496226)

[4.10.3 Pengujian Aktuator dengan MetodePID 40](#_Toc23496227)

[4.10.4 Pengujian Aktuator dengan MetodePID 41](#_Toc23496228)

[BAB 5. HASIL DAN PEMBAHASAN 44](#_Toc23496229)

[5.1 Implementasi Desain *User Interface* 44](#_Toc23496230)

[5.1.1 *User Interface* Log In 45](#_Toc23496231)

[5.2 Hasil Implementasi Solar Tracker dengan Metode *Fuzzy* 45](#_Toc23496232)

[5.3 Hasil Implementasi Aktuator dengan Metode PID 45](#_Toc23496233)

[5.4 Hasil Implementasi Web *Solar Tracker System* 45](#_Toc23496234)

[BAB 6. KESIMPULAN 46](#_Toc23496235)

[6.1 Kesimpulan 46](#_Toc23497518)

[6.2 Saran 47](#_Toc23497519)

[Daftar Pustaka 49](#_Toc23497520)

[LAMPIRAN 51](#_Toc23497521)

[Lihat data *history tracker* 53](#_Toc23497522)

# PENDAHULUAN

Bab ini merupakan langkah awal dari penulisan tugas akhir. Bab ini berisi latar belakang, rumusan masalah, tujuan, manfaat, dan batasan masalah.

## Latar Belakang

Sumber energi baru dan yang terbarukan di masa mendatang akan semakin mempunyai peran yang sangat penting dalam memenuhi kebutuhan energi. Hal ini disebabkan oleh penggunaan bahan bakar fosil sebagai sumber daya pembangkit listrik konvensional dalam jangka panjang akan menguras sumber minyak bumi, gas dan batu bara yang cadangannya semakin lama semakin menipis (Anggara, Kumara, & Giriantari, 2014). Energi terbarukan juga lebih ramah lingkungan dan lebih aman daripada energi tidak terbarukan seperti energi fosil.

Tenaga listrik sangat di butuhkan dalam era modern saat ini. Berbagai metode telah ditemukan dalam hal membangkitkan tenaga listrik mulai dari yang memakai energi terbarukan sampai nonterbarukan. Salah satu cara membangkitkan energi listrik adalah dengan menggunakan tenaga matahari.

Pembangkit listrik tenaga matahari banyak digunakan karena pemasangan dan perawatannya relatif lebih mudah. Selain itu teknologi ini juga ramah lingkungan dan jauh lebih aman daripada metode yang lain. pembangkit listrik tanaga surya juga bersifat fleksibel dan modular sehingga dapat di pasangkan pada benda bergerak seperi perahu atau mobil.

Intensistas radiasi yang tinggi di Indonesia dapat membangkitkan energy listrik yang besar dengan rata-rata radiasi sebesar 1000 Watt/m2  (Rahardjo & Fitriana, 2014). Kondisi ini sangat cocok sebagai acuan untuk mengembangkan pembangkit listrik tenaga surya di Indonesia. Hal ini dapat dilakukan dengan bantuan panel surya yaitu satu kesatuan modul yang didalamnya terdapat sel surya dan komponen pendukung lainnya.

Panel surya memiliki beberapa kelemahan yaitu ketersediaan sinar matahari yang di perlukan untuk membangkitkan listrik. Mulai dari topografi, lokasi, dan musim. Salah satu cara mengatasi kelemahan itu adalah dengan menggerakan posisi panel surya ke arah yang tepat. Menggerakkan posisi panel surya dapat membantu memaksimalkan penerimaan sinar matahari sehingga produtivitas panel menjadi lebih baik. Pengaruh sudut kemiringan panel surya terhadap radiasi matahari yang diterima oleh panel surya per bulan cenderung tidak tetap. Hal ini disebabkan radiasi yang diterima panel surya tidak hanya bergantung pada besar sudut kemiringan panel tetapi juga diakibatkan beberapa faktor lain seperti gerak semu harian dan tahunan matahari serta indeks kecerahan per bulan (Pangestuningtyas, 2013).

Penelitian ini akan menggunakan metode *Fuzyy* yaitu suatu metode yang digunakan untuk menentukan nilai yang tidak pasti diantara nilai 0 dan 1. Pada penelitian ini metode *Fuzyy* akan digunakan untuk menentukan posisi sudut optimal (*setpoint*) dalam menentukan arah pengambilan sinar matahari .Selain menggunakan metode *Fuzyy* penelitian ini juga menggunakan metode *Proportional Integral Derivative* (PID) yaitu suatu metode yang digunakan untuk menjaga presisi suatu benda agar selalu berada pada posisi yang telah ditentukan.

Memaksimalkan penerimaan sumber cahaya dapat dilakukan dengan menggabungkan *Fuzyy* dan PID. Pengaturan sudut panel surya ditentukan menggunakan *solar tracker* dengan menerapkan metode *Fuzyy Logic Controller* dengan menilai sudut yang paling optimal untuk penerimaan sinar matahari yang didapat melalui data sensor *Light Dependant Resistor* (LDR). Sedangkan untuk PID berfungsi sebagai penyesuaian motor untuk mendekati sudut optimal. Penggabungan kedua metode tersebut juga bertujuan untuk mempercepat kinerja *solar tracker*.

Penelitian ini bertujuan untuk mengimplementasikan metode *Fuzyy* PID pada *solar tracker* dengan cara menentukan sudut terbaik dalam penerimaan cahaya yang didapat melalui perhitungan *Fuzyy.* Hasil perhitungan tersebut digunakan untuk mengarahkan posisi panel menuju sudut matahari paling optimal dengan bantuan PID*.* Kontrol posisi selain menggunakan dua metode tersebutjuga berbasis web untuk mengimplemntasikan konsep Internet Of Things (IOT) pada komunikasi antara web dengan *solar tracker.*

## Rumusan Masalah

Berdasarkan latar belakang masalah penelitian, maka muncul perumusan masalah sebagai berikut:

1. Apa pengaruh sudut penerimaan terhadap produktivitas pembangkit listrik tenaga surya?
2. Bagaimana implementasi logika *Fuzyy* dalam memilih *setpoint*?
3. Bagaimanakah perbandingan kinerja *solar tracker* yang menggunakan metode *Fuzyy* dan PID dengan *solar tracker* tanpa *Fuzyy dan PID*?
4. Bagaimana cara user berinteraksi dengan sistem?

## Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini adalah:

1. Menganalisis pengaruh sudut penerimaan cahaya pada panel surya terhadap matahari
2. Mengoptimalkan penerimaan cahaya matahari untuk mendapatkan sumber energi yang maksimal
3. Membandingkan kinerja dan *response time* dari 2 buah panel surya yang menggunakan metode *Fuzyy* dan *PID* dengan panel surya tanpa *Fuzyy dan PID*
4. Memonitoring *solar tracker* menggunakan web

## Manfaat Penelitian

Manfaat dari penelitian ini adalah:

1. Bagi Akademis

Penelitian yang dilakukan diharapkan memberikan hasil yang mampu memberikan masukan informasi yang terkait dengan judul penelitian kepada pembaca pada umumnya dan pada Program Studi Sistem Informasi Universitas Jember pada khususnya.

1. Bagi Peneliti

Mengetahui bagaimana proses penerapan metode *Fuzyy PID* untuk mengetahui seberapa besar pengaruhnya terhadap sistem kontrol posisi pada panel surya.

1. Bagi Objek Penelitian

Memberikan manfaat berupa pemanfaatan sinar matahari agar lebih optimal

## Batasan Masalah

Peneliti memberikan batasan masalah untuk objek dan tema yang dibahas sehingga tidak terjadi penyimpangan dalam proses penelitian dan menganalisis

1. Pencarian sudut digunakan untuk mendapatkan cahaya matahari paling maksimal
2. Bahasa pemrograman yang digunakan pada mikrokontroller adalah C, sedangkan untuk menampilkan hasil visual menggunakan PHP, CSS, dan *Javascript.*
3. Uji coba pada penelitian ini dilakukan dengan bantuan senter lampu pada sudut 80’. Uji coba ini digunakan untuk membandingkan respon  *solar tracker* yang dilengkapi *Fuzyy* PID dengan *solar tracker* tanpa dilengkapi *Fuzyy* PID.
4. Tidak membahas tentang jaringan komputerisasi dan keamanan data pada sistem yang dibuat.
5. Web sistem hanya digunakan untuk menampilkan hasil perhitungan.

# TINJAUAN PUSTAKA

Bagian ini memaparkan tinjauan yang berkaitan dengan masalah yang dibahas, serta kajian teori yang dikaitkan dengan permasalahan yang dihadapi. Teori yang di dapatkan berupa perhitungan *Fuzyy* PID dan penerapanya yang dapat membantu peneliti dalam penelitian ini. Perhitungan yang di dapatkan akan membantu peneliti dalam menghitung nilai yang akan berpengaruh ke posisi *tracker.*

## **Penelitian Terdahulu**

Penelitian dengan judul “Sistem Kontrol Cerdas Pelacak Sumber Cahaya Menggunakan Kontrol Proporsional Integral Deverative (PID)” (Prahara, 2018) melakukan penelitian tentang perhitungan untuk mencari titik opimal cahaya matahari dengan komputasi menggunakan program Matlab dan mikrokontroler ATMega328 sebagai *hardware*. Data sensor digunakan sebagai acuan untuk perhitungan kontrol PID dan *Fuzyy* digunakan sebagai penyelaras hasilnya. Data sensor selanjutnya dikirim ke komputer untuk diproses oleh Matlab. Hasil komputasi Matlab yang berupa data pengamatan ditampilkan pada layar komputer melalui GUI, sedangkan hasil komputasi yang berupa perintah menggerakkan motor servo dikirim ke mikrokontroler melalui serial komunikasi. Dari penelitian tersebut,diperoleh hasil yang mampu memberikan kecepatan pelacakan dengan tingkat akurasi yang baik.

Penelitian dengan judul “Perbandingan Pemodelan Kontrol *Fuzyy* dan PID

pada Pemanas *Fuel Gas*” (ZA & Maulinda, 2015) melakukan penelitian tentang sistem feed reject di PT. Arun NGL yang digunakan untuk mengatur temperatur aliran gas yang menuju kompresor aliran gas menuju kompresor *bahan bakar* gas. Tujuan dari perbandingan metode *Fuzyy* dan PID untuk menjaga terjadiya *surging* pada kompresor bahan bakar gas. Hasil simulasi menunjukkan bahwa, pada pengujian model kontrol dengan variasi setpoint pada range temperatur yang kecil model kontrol *Fuzyy* lebih cepat 1.05 menit/oC untuk mencapai setpoint dibandingkan model kontrol PID. Kenaikan dan penurunan temperatur setpoint secara bertahap pada range temperatur yang besar model kontrol *Fuzyy* lebih cepat mencapai setpoint 0.164 menit/oC dibandingkan model kontrol PID. *Fuzyy* lebih cepat 0.13 menit mencapai setpoint dibandingkan kontrol PID pada kenaikan dan penurunan temperature yang drastis. Pada posisi error yang besar dan laju alir yang besar *Fuzyy* memiliki kinerja yang lebih baik, sebaliknya kontrol PID pada posisi error yang kecil dan laju alir yang kecil memiliki kinerja yang sangat baik.

Penelitian dengan judul “Analisis Pengaruh Sudut Kemiringan Panel Surya Terhadap Radiasi Matahari yang Diterima Oleh Panel Surya Tipe Larik Tetap” (Pangestuningtyas, 2013) melakukan penelitia tentang bagaimana cara menstabilkan daya yang dihasilkan dari panel surya tipe array yang di terapkan di Kota Semarang. Program dibuat dengan menggunakan GUI (*Graphic User Interface*) pada *Software* MATLAB 2008a. Penggunaan GUI bertujuan untuk memudahkan dalam hal pengoperasian program serta melihat pengaruh sudut kemiringan modul dari 1o hingga 90o terhadap radiasi matahari yang diterima panel surya per bulan. Data perhitungan radiasi yang telah didapat di validasi dengan *software* *RETScreen.* Hasil yang diperoleh dari penelitian ini berupa sudut kemiringan yang paling tepat untuk menerima radiasi matahari setiap bulannya sebesar 3,965 kWh/m2/hari.Perbedaan hasil antara perhitungan, pengukuran dan *software* RETScreen diakibatkan adanya gerak semu harian dan tahunan matahari serta indeks kecerahan yang berbeda serta besar radiasi matahari terukur yang digunakan sebagai acuan dalam menghitung radiasi matahari yang dapat diterima oleh panel surya.

Penelitian dengan judul “Pengaturan Pitch Angle Turbin Angin Berbasis Kendali Logika *Fuzyy* (Aplikasi Pada Data Angin Daerah Medan Tuntungan Dan Sekitarnya)” (Pahlevi & Yana, 2016) melakukan penelitian tentang pengembangan pemanfaatkan energi terbarukan untuk meningkatkan efisiensi kerja turbin angin menggunakan *Pitch Angle Control.* kontrol ini untuk memaksimalkan daya yang terdapat pada energi angin selain itu juga merupakan pengaruh yang ditimbulkan dengan menambahkan kontrol *pitch angle*. Pengaturan *pitch angle* turbin angin menggunakan *Fuzyy Logic Controller* (FLC) yang mana kontrol ini mempengaruhi koefisien performansi (*Performance Coefficient*) pada turbin angin. Hasil yang diperoleh dari penelitian ini berupa penambahan kendali logika *Fuzyy* pada pengaturan sudut baling-baling turbin mampu meningkatkan efisiensi kerja turbin angin sebesar 11,9% dibandingkan saat menggunakan sudut tetap 10o, dimana sudut tetap sebesar 10o dianggap paling optimal untuk kondisi angin kota Medan.

Penelitian dengan judul “Desain dan Purwarupa *Fuzyy* Logic Control untuk Pengendalian Suhu Ruangan” (Wahab, Sumardiono, & Tahtawi, 2017) melakukan penelitian tentang pengendalian suhu ruangan yang menggunakan *Fuzyy* dan yang tidak. Simulasi penelitian ini dibantu oleh *Fuzyy* Logic Toolbox yang ada pada Matlab. Dua unit purwarupa dirancang dengan perbedaan pada sensor masukannya. Purwarupa pertama menggunakan sensor yang masih diasumsikan, sedangkan yang kedua menggunakan sensor sebenarnya. Hasil pengujian menunjukkan bahwa purwarupa sistem pertama dan kedua mampu mengendalikan suhu ruangan dengan rata-rata kesalahan berturut-turut 1,31% dan 4,06% jika dibandingkan dengan simulasi Matlab.

Penelitian-penelitian tersebut disimpulkan bahwa PID sesuai untuk perhitungan penyesuaian sudut optimal dan *Fuzyy* digunakan sebagai penyelaras dari hasil PID yang didapatkan. Meskipun nilai *setpoint* bisa berubah-ubah *Fuzyy* mampu menyesuikan hasil sudut optimalnya.

## Panel Surya

Sel surya adalah suatu peralatan yang merupakan implementasi dari efek *fotovoltaik* yaitu mengkonversi cahaya matahari menjadi energi listrik. Panel surya adalah satu kesatuan modul yang didalamnya terdapat sel surya dan peralatan pendukung lainnya (Saputra, 2014)*.*

## Light Dependent Resistor (LDR)

*Light Dependent Resistor* (LDR) meru *Light Dependent Resistor* (LDR)akan Sensor cahaya yang digunakan untuk mengukur intensitas cahaya. Sensor LDR merupakan alat yang memiliki variasi nilau resistansi tergantung pada intensitas cahaya yang mengenai nya. semakin besar intensitas cahaya yang diberikan maka akan semakin kecil resistansi yang ada pada sensor (Suoth, Mosey, & Telleng, 2018).

## *Solar Tracker*

*Solar Tracker* atau Penjejak Matahari adalah komponen tambahan pada panel surya yang terdiri dari motor dan *driver* guna menggerakkan panel surya sesuai dengan titik maksimal radiasi matahari atau dengan kata lain menjaga posisi panel surya tetap 90 derajat dengan matahari. Banyak tipe dari Penjejak matahari ini, diantaranya adalah penjejak matahari dengan sensor radiasi dan penjejak matahari berdasarkan waktu (Saputra, 2014).

## *Fuzyy Logic Control*

Kendali logika *Fuzyy* dalam perancangannya. Terdapat tiga tahapan yaitu fuzzifikasi, mekanisme inferensi dan defuzzifikasi. Fuzzifikasi merupakan tahap pertama yang mengubah nilai tegas dari suatu variable menjadi nilai *Fuzyy*. Nilai *Fuzyy* ini selanjutnya digunakan sebagai masukan dari mekanisme inferensi. Pada tahap ini, akan dilakukan pengambilan keputusan dari masukan yang ada berdasarkan basis aturan logika yang dirancang. Terakhir, nilai keluaran dari mekanisme inferensi yang berbentuk *Fuzyy* selanjutnya diubah kembali kedalam bentuk tegas melalui proses *defuzzifikasi* (Wahab, Sumardiono, & Tahtawi, 2017).

Perhitungan *Fuzyy* terbagi menjadi beberapa langkah yang pertama yaitu tahap fuzzifikasi, *control rule base* dan defuzzifikasi

### *Fuzzifikasi*

Prosedur fuzzifikasi merupakan proses yang berfungsi mengubah variabel numerik menjadi variabel linguistik (variabel *Fuzyy*). Langkah pertama yang harus dilakukan adalah mencari nilai error yaitu nilai yang didapat dari selisih pembacaan sensor. Selanjutnya dibutuhkan nilai delta error yaitu selisih nilai error saat ini dengan error sebelumnya.Nilai error dan delta error yang dikuantisasi sebelumnya diolah oleh kontroler logika *Fuzyy*, kemudian diubah terlebih dahulu ke dalam variabel *Fuzyy*. Melalui fungsi keanggotaan yang telah disusun, maka dari nilai error dan delta error kuantisasi akan didapatkan derajat keanggotaan bagi masing-masing nilai error dan delta error.

### *Control Rule Base*

Pengetahuan *Fuzyy* terdiri dari beberapa aturan *Fuzyy* yang dikelompokkan kedalam suatu basis aturan. *Rule base* merupakan dasar dari pengambilan keputusan atau *inference  process*, untuk mendapatkan aksi keluaran sinyal kontrol dari suatu kondisi masukan yaitu error dan delta error, dengan berdasarkan rule-rule yang telah ditetapkan. Proses inferensi menghasilkan sinyal keluaran yang masih dalam bentuk bilangan *Fuzyy*, yaitu derajat keanggotaan dari sinyal kontrol.

### *Defuzzifikasi*

Defuzzifikasi merupakan kebalikan dari proses transformasi sebuah himpunan *Fuzyy* kedalam himpunan tegas. Metode defuzzifikasi yang umum digunakan ada 2 macam yaitu metode defuzzifikasi versi sugeno dan mamdani. Metode yang cocok pada penelitian ini menggunakan metode mamdani *the center of gravity method.* Persamaan metode ini dapat dihitung melalui rumus (1).

(1)

x\* adalah output defuzzified, •• (x) adalah fungsi keanggotaan agregat dan x adalah variabel output.

## Proportional Integral Derivative (PID)

Pengendali PID merupakan gabungan dari tiga macam pengendali, yaitu pengendali proporsional, pengendali integral, dan pengendali turunan. Tujuan dari penggabungan ketiga macam pengendali tersebut adalah untuk memperbaiki kinerja sistem di mana masing-masing pengendali akan saling melengkapi dan menutupi dengan kelemahan dan kelebihan masing-masing sesuai dengan skema proses PID pada Gambar 2.1.



Gambar 2.1 skema PID

Gambar 2.1 menjelaskan nilai *setpoint* yaitu nilai setpoint diproses dengan langkah awal mencari nilai error dari selisih *setpoint* dengan *Present Value*(PV). Struktur PID parallel merupakan struktur PID yang sering digunakan. Persamaan pada Persamaan 2 memperlihatkan bentuk umum dari kontrol PID.

(2)

Dengan CO(t) = output kontroler, e(t) = *error* (e = SP - PV), Kp = gain proporsional, Ti = *time* *integral*, Td = *time derivative*. Dalam kawasan *laplace* dapat ditulis sesuai rumus (3).

(3)

Parameter PID perlu ditentukan terlebih dahulu untuk menentukan masing-masing koefisien dari parameter proporsional, integral dan derivatif pada plan. Proses ini dapat dilakukan dengan cara *trial and error* untuk memilih koefisien yang pas. Proses trial and error dilakukan sampai diperoleh hasil yang diinginkan dengan mengacu pada karakteristik masing-masing kontrol proporsional, integral, dan derivatif yang sesuai pada Tabel 2.1.

Tabel 2.6.A Karakteristik PID

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kontrol | Rise time | Overshoot | Settling time | Error steady state |
| Kp | Mengurangi | Meningkatkan | Mengurangi | Mengurangi |
| Ki | Mengurangi | Meningkatkan | Meningkatkan | Menghilangkan |
| Kd | Perubahan kecil | Mengurangi | Menurun | Perubahan kecil |

## Database

Database adalah kumpulan data yang disimpan secara sistematis didalam computer yang dapat dimanipulasi menggunakan perangkat lunak untuk menghasilkan informasi. Database merupakan aspek yang sangat penting dalam mengelola informasi karena digunakan sebagai media penyimpanan data yang akan dioleh lebih lanjut (Febrianto, 2015).

## Bahasa Pemrograman

Bahasa pemrograman merupakan bahasa yang digunakan untuk membangun suatu sistem. Contoh dari Bahasa pemrograman yaitu PHP, C, Ruby, dan lain lain. Pada penelitian ini peneliti menggunakan Bahasa pemrograman PHP dan C untuk backend. Sedangkan untuk frontend menggunakan HTML, CSS dan javascript

## Internet of Things (IOT)

IOT merupakan segala aktifitas yang pelakunya saling berinteraksi dan dilakukan dengan memanfaatkan internet . Dalam penggunaan nya *Internet of Things* banyak ditemui dalam kehidupan sehari-hari sebagai media untuk melakukannya (Sulaiman, 2017).

Media penyimpanan pada IOT dapat dilakukan dengan memanfaatkan *cloud computing.* Data yang telah disimpan pada *cloud* dapat diambil kapan saja dan dimana saja. Namun media ini memiliki kekurangan pada kecepatan akses tergantung pada *bandwith* internet dan memerlukan ruang penyimpanan yang besar (Sulaiman, 2017).

# METODOLOGI PENELITIAN

Bab ini menggambarkan tentang penelitian yang akan dilakukan untuk menjawab rumusan masalah sehingga dapat mewujudkan tujuan sebenarnya dari penelitian. Pada metodologi penelitian akan dijelaskan tentang tempat dan waktu penelitian serta tahapan dari penelitian.

## Objek Penelitian

Objek penelitian merupakan *solar tracker* yang dilengkapi metode *Fuzyy* PID. Penelitian ini menggunakan *prototype* yang mewakili sistem *solar tracker* yang asli untuk menganalisa pengaruh respon terhadap penerimaan cahaya matahari jika menggunakan metode *Fuzyy* dan PID yang dapat di monitoring melalui web.

## Tempat dan Waktu Penelitian

Penelitian ini dilakukan di Universitas Jember. Penelitian dilakukan selama satu hari.

## Tahapan Penelitian

Tahapan penelitian merupakan tahapan yang digunakan untuk menyelesaikan penelitian ini. Tahapan penelitian ini dimulai dari menganalisa kebutuhan sampai dengan membuat kesimpulan dari uji simulasi sistem yang telah dikerjakan sesuai dengan Gambar 3.1.

Analisa Kebutuhan

Desain Sistem

Pengambilan data sensor

Implementasi metode Fuzzy

Implementasi desain

Implementasi metode PID

Pembuatan web sistem

Implementasi desain *hardware*

Uji Coba

Mulai

Analisa hasil uji coba

Selesai

Kesimpulam

Gambar 3.1 Tahapan Penelitian

### Analisa Kebutuhan

Analisa Kebutuhan dalam penelitian ini ada dua yaitu kebutuhan secara fungsional dan non fungsional. Jenis kebutuhan berisi tentang apa saja yang dibutuhkan oleh sistem serta berbagai informasi yang dihasilkan oleh sistem. Berikut merupakan jenis kebutuhan sistem yang telah dibangun oleh pengembang.

### Desain sistem

Desain sistem merupakan tahapan yang harus dilakukan setelah menganalisa kebutuhan. Hal ini berkaitan dengan menggambarkan bagaimana suatu sistem akan dibangun. Pada penelitian ini desain sistem dimulai dengan pembuatan skenario sampai dengan desain *user interface* (UI).

### Impelementasi Desain *Hardware*

Desain  *Hardware*

**

### Pengambilan Data Sensor

Pengambilan data sensor dilakukan pada saat ada sinar matahri. Pengambilan dilakukan dengan bantuan 4 sensor LDR dengan penempatan sesuai Gambar 3.3.

Gambar 3.2 Penempatan Sensor LDR

Penempatan tersebut dimaksudkan agar dapat membaca dari 4 arah atas, bawah, kiri, dan kanan dengan membandingkan nilai resistansi dari keempat sensor. Dari keempat sensor tersebut dapat menghasilkan dua buah variabel yaitu error vertikal dan error horizontal. Kedua variabel tersebut didapat melalui perhitungan di bawah ini:

Error vertikal

Error\_v = ((tl-tr)/2)- ((dl-dr)/2) (4)

Error horizontal

Error\_h = ((tl-dl)/2)- ((tr-dr)/2) (5)

Ket :

error\_v = error vertikal

error\_h = error horizontal

tl = ldr top left (penempatan di kiri atas)

tr = ldr top right (penempatan di kanan atas)

dl = ldr down left (penempatan di kiri bawah)

dr = ldr down right (penempatan di kanan bawah)

nilai error dapat berupa bilai bulat (negatif dan positif) dengan aturan sebagai berikut

1. *Solar* *tracker* akan bergerak ke arah kiri jika nilau error horizontal bernilai positif
2. *Solar* *tracker* akan bergerak ke arah kanan jika nilau error horizontal bernilai negatif
3. *Solar* *tracker* akan bergerak ke arah atas jika nilau error vertikal bernilai positif
4. *Solar* *tracker* akan bergerak ke arah bawah jika nilau error vertikal bernilai negatif

### Implementasi Metode *Fuzyy*

Proses perhitungan metode *Fuzyy* pada *tracker* diawali dengan melakukan pembacaan sensor *Light Dependent Resistor* (LDR) yang terdapat pada *tracker*. Kemudian sistem akan menghitung nilai error pada masing-masing posisi vertikal dan horizontal. Hasil perhitungan dari nilai error tersebut akan diolah oleh metode *Fuzyy* sampai menemukan keputusan apa yang harus dilakukan berdasarkan keluaran *Fuzyy* sesuai pada *flowchar* Gambar 3.5. Apabila keluaran *Fuzyy* masih belum mencapai nilai 0 atau sesuai dengan Gambar 3.5 drajat keanggotaan pada variabel linguistik ZE, maka akan dilakukan pembacaan sensor lagi.

*Range* pada drajat keanggotaan *Fuzyy* didapat melaui uji coba pembacaan sensor LDR dengan mencari rata-rata pada masing-masing sisi. Setelah mendapat rata-rata maka dilakukan perhitungan nilai error. Dari nilai error tersebut yang akan digunakan sebagai parameter drajat keanggotaan *Fuzyy* dengan *range* -275 sampai 275 sesuai dengan Gambar 3.3 dan 3.4.



Gambar 3.3 Range error V



Gambar 3.4 Range error H



Gambar 3.5 Flowchart

Hasil uji coba pembacaan sensor akan digunakan pada range drajat keanggotaan *Fuzyy*. Berikut ini adalah langkah perhitungan metode *Fuzyy* control yang dimulai dengan mencari nilai error sampai dengan deffuzifikasi.

1. *Error*

Nilai error yang akan digunakan dalam penelitian ini ada dua yaitu error vertical dan error horizontal. *Error* horizontalmerupakan hasil yang didapatkan dengan menghitung selisih dari rata-rata hasil sensor LDR kanan dengan kiri sedangkan error vertikal didapat dari selisih antara rata-rata atas dengan bawah. Nilai *error* dibagi menjadi 5 fungsi keanggotaan. Fungsi keanggotaan serta derajat keanggotaan untuk *error* adalah sebagai berikut:



Gambar 3.6 Drajat Keanggotaan

1. *Negative Big (NB)*

*Negative Big yang* bernilai < -125

,

1. *Negative Medium (NM)*

*Negative Medium yang* bernilai < -10 sampai -275

,

,

1. *Negative Small (NS)*

*Negative Small yang* bernilai < -125 sampai -0

,

,

1. *Zero Error (ZE)*

*Zero Error yang* bernilai < -10 sampai 10

,

,

1. *Positive Small (PS)*

*Positive Small yang* bernilai >10 sampai 125

,

,

1. *Positive Medium (PM)*

*Positive Medium yang* bernilai >10 sampai 275

,

,

1. *Positive Big (PB)*

*Positive Big yang* bernilai >125

,

,

1. Delta(*Δ*) Error

*Delta(Δ) error* merupakan selisih antara nilai *error* saat ini dengan nilai *error* yang ada sebelumnya. Cara mendapatkan nilai variabel *delta error* harus diketahui nilai *error* sebelumnya atau jika nilai error sebelumnya tidak ada maka dinyatakan dengan 0, untuk lebih jelasnya dapat diperhatikan persamaan berikut:

(6)

Keterangan:

en= *error* sekarang

en-1= *error* sebelumnya

Fungsi keanggotaan *delta error* ditentukan dengan melihat perubahan *error* dari yang terkecil hingga *error* terbesar. Selanjutnya nilai *delta error* tersebut dibagi menjadi 7 fungsi keanggotaan seperti fungsi keanggotaan *error* sebagai berikut:

1. *Negative Big (NB)*

*Negative Big yang* bernilai < -125

,

1. *Negative Medium (NM)*

*Negative Medium yang* bernilai < -10 sampai -275

,

,

1. *Negative Small (NS)*

*Negative Small yang* bernilai < -125 sampai -0

,

,

1. *Zero Error (ZE)*

*Zero Error yang* bernilai < -10 sampai 10

,

,

1. *Positive Small (PS)*

*Positive Small yang* bernilai >10 sampai 125

,

,

1. *Positive Medium (PM)*

*Positive Medium yang* bernilai >10 sampai 275

,

,

1. *Positive Big (PB)*

*Positive Big yang* bernilai >125

,

,

1. Fuzzifikasi

Pada proses fuzzifikasi, nilai tegas akan diubah menjadi variable liguistik yang berbentuk kurva sebelum diproses pada tahapan selanjutnya (Aisuwarya & Annafi, 2017). Jika nilai tegas memiliki input keanggotaan 1, maka sudah jelas masuk dalam anggota variabel linguistik sesuai pada Gambar 3.40 dan penjelasan pada Tabel 3.1, hal sebaliknya akan terjadi jika nilai tegas yang didapay adalah 0.



**1**

**0**

Gambar 3.7 Fuzzifikasi

Table 1 Tabel Istilah Linguistik

|  |  |
| --- | --- |
| **Istilah Linguistic** | **Label** |
| Negative Big | NB |
| Negative Medium | NM |
| Negative Small | NS |
| Zero | Z |
| Positive Small | PS |
| Positive Medium | PM |
| Positive Big | PB |
| Error | E |
| Change in Error | DE |

1. *Rule base*

*Rule Fuzyy* untuk kontrol menentuka sudut optimal agar berjalan stabil. Pada proses pembentukan aturan (*rule*) ini menggunakan operator “*and*” untuk mengkombinasikan nilai variabel *error* dengan variable delta error-nya maka didapatkanlah aturan (*rule*) sebagai Tabel 3.2.

Table 2 Rule Base Fuzzy

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **E/DE** | **NB** | **NM** | **NS** | **ZE** | **PS** | **PM** | **PB** |
| **NB** | NB | NB | NB | NB | NM | NS | ZE |
| **NM** | NB | NB | NM | NM | NS | ZE | PS |
| **NS** | NB | NM | NS | NS | ZE | PS | PM |
| **ZE** | NB | NM | NS | ZE | PS | PM | PB |
| **PS** | NM | NS | ZE | PS | PS | PB | PB |
| **PM** | NS | ZE | PS | PM | PB | PB | PB |
| **PB** | ZE | PS | PM | PB | PB | PB | PB |

1. Defuzzifikasi

Defuzzifikasi adalah suatu himpunan *Fuzyy* yang diperoleh dari perhitungan kombinasi *control rule base*, sedangkan output yang dihasilkan merupakan suatu bilangan pada domain himpunan *Fuzyy* tersebut, sehingga jika diberikan suatu himpunan *Fuzyy* dalam range tertentu, maka harus dapat diambil suatu nilai crisp tertentu sebagai keluarannya.

(7)

### Implementasi *Solar* *Tracker* Tanpa Metode *Fuzyy*

Dua pasang LDR yang digunakan sebagai sensor digunakan sebagai acuan penentu arah pergerakan *solar* *tracker*, kedua sensor tersebut akan menerima intensitas cahaya yang berbeda dan akan selalu naik turun sesuai dengan besar cahaya yang masuk (Mustafa & Ahmed, 2017). Jika salah satu memiliki nilai intensitas yang berbeda maka motor penggerak akan bergerak. Misalkan terdapat dua pasang sensor atas dan bawah jika sensor atas memiliki nilai intensitas cahaya yang lebih tinggi, maka motor akan bergerak ke arah atas sampai nilai kedua sensor (sensor atas dan sensor bawah) memiliki nilai intensitas cahaya yang sama atau memiliki nilai perbedaan bernilai nol.

Pergerakan dari *solar* *tracker* ini lebih sederhana, dan memiliki kecepatan pergerakan yang tetap, pergerakan yang dimaksud adalah saat intensitas cahaya yang diterima tidak sama maka akan bergerak dengan kecepatan yang sama sampai kedua sensor mendapatkan nilai yang sama. Implementasi dari *solar* *tracker* tanpa metode *Fuzyy* juga tidak stabil dikarenakan motor penggerak akan selalu bergerak sampai kedua sensor memiliki nilai intensitas yang sama, dan nilai tersebut selalu berubah-ubah serta sulit mencapai nilai perbedaan yang bernilai nol.

### Implementasi *Solar* *Tracker* Dengan Metode *Fuzyy*

*Solar* *tracker* dengan metode *Fuzyy* memiliki beberapa perbedaan dengan *solar* *tracker* pada desain sebelum nya. Perbedaan nya antara lain pada saat proses pembacaan sensor dan pengolahan data sensor. Pada *solar* *tracker* tanpa metode *Fuzyy* hanya  membandingkan perbedaan intensitas cahaya pada sisi yang berlawanan (sisi atas dengan sisi bawah atau sisi kanan dengan sisi kiri). Sedangkan *solar* *tracker* dengan metode *Fuzyy* memiliki perhitungan setelah mendapatkan nilai sensor.

Nilai sensor akan di hitung dan diolah untuk menentukan intensitas cahaya yang berupa variabel numerik menjadi variabel lingusitik. Hal ini bertujuan agar sistem mengetahui kondisi intensitas cahaya yang sebelumnya kurang spesifik antara gelap dan terang menjadi lebih spesifik dengan memiliki status terang , agak terang , redup dan gelap.

*Solar* *tracker* tanpa metode *Fuzyy* yang memiliki dua status yaitu gelap dan terang mengakibatkan sistem hanya mengirim perintah ke motor berupa bergerak saat terang dan berhenti saat gelap. Sedangan dengan penambahan metode guzzy akan memiliki banyak kemungkinan dan kemnungkinan tersebut membantu mengatur pergerakan motor dari cepat ke lambat sampai berhebti di status cahaya matahari paling optimal di range tertentu.

### Implementasi Metode PID

Metode PID digunakan untuk membantu pergerakan dan kestabilan aktuator / penggerak panel surya. Metode PID membantu pergerakan karena metode tersebut membantu arah pergerakan motor penggerak agar lebih akurat untuk menuju setpoint yang sudah ditetukan oleh *tracker*. Metode PID juga membantu menstabilkan motor penggerak, dengan cara panel surya menjaga posisi panel surya pada sudut tertentu sesuai dengan setpoint yang diperoleh oleh *tracker*.

### Pembuatan Web Sistem

Web sistem dibangun bertujuan untuk membantu memonitor dan menampilkan data kinerja *solar* *tracker* dan aktuator . Data yang di tampilkan antara lain data realtime berupa sudut sekarang, histori atau data sudut terdahulu, simulasi perhitungan *Fuzyy* dan PID, simulasi penggunaan kebutuhan energi.

Web sistem juga sebagai perantara penyimpanan data setpoint yang dihasilkan oleh *tracker* dan data setpoint tersebut akan diambil / di request sebagai data acuan PID. Proses pengiriman dan pengambilan data melalui koneksi internet ataupun lokal. Proses tersebut bisa dikatan merupakan IOT dikarenakan adanya komunikasi mechine to machine atau dari *solar* *tracker* ke aktuator tanpa adany campur tangan operator atau manusia di dalamnya.

# ANALISA DAN PERANCANGAN SISTEM

Bab ini merupakan bagian yang membahas tentang pengembangan sistem kontrol posisi berbasis web pada panel surya menggunakan metode Fuzzy PIDPengembangan sistem dilakukan dengan menggunakan model waterfall, dengan tahapan yakni analisis kebutuhan fungsional dan non-fungsional sistem, pembuatan desain sistem, penulisan kode program dan pengujian sistem.

## Analisa Kebutuhan

Analisa Kebutuhan dalam penelitian ini ada dua yaitu kebutuhan secara fungsional dan non fungsional. Jenis kebutuhan berisi tentang apa saja yang dibutuhkan oleh sistem serta berbagai informasi yang dihasilkan oleh sistem. Berikut merupakan jenis kebutuhan sistem yang telah dibangun oleh pengembang.

### Kebutuhan Fumgsional

Kebutuhan fungsional adalah kebutuhan utama yang harus dilakukan oleh sistem. Kebutuhan yang berkaitan dengan fungsi sistem pada kebutuhan fungsional atau *Software Requirement Spesification Functional* untuk dapat menghasilkan keluaran yang diinginkan oleh pengembang (Febrianto, 2015). Kebutuhan ini dapat dilihat pada Tabel 3.1.

Tabel 4.1.A Kebutuhan Fungsional

|  |  |
| --- | --- |
| SRSF ID | Identifikasi |
| SRSF\_1 | Dapat memproses metode *Fuzyy* yang digunakan pada *tracker* |
| SRSF\_2 | Dapat memproses PID berdasarkan *setpoint* yang didapatkan oleh *tracker.* |
| SRSF\_3 | Membaca hasil pembangkitan energi berupa *voltage* dan *ampere* |

### Kebutuhan Non Fumgsional

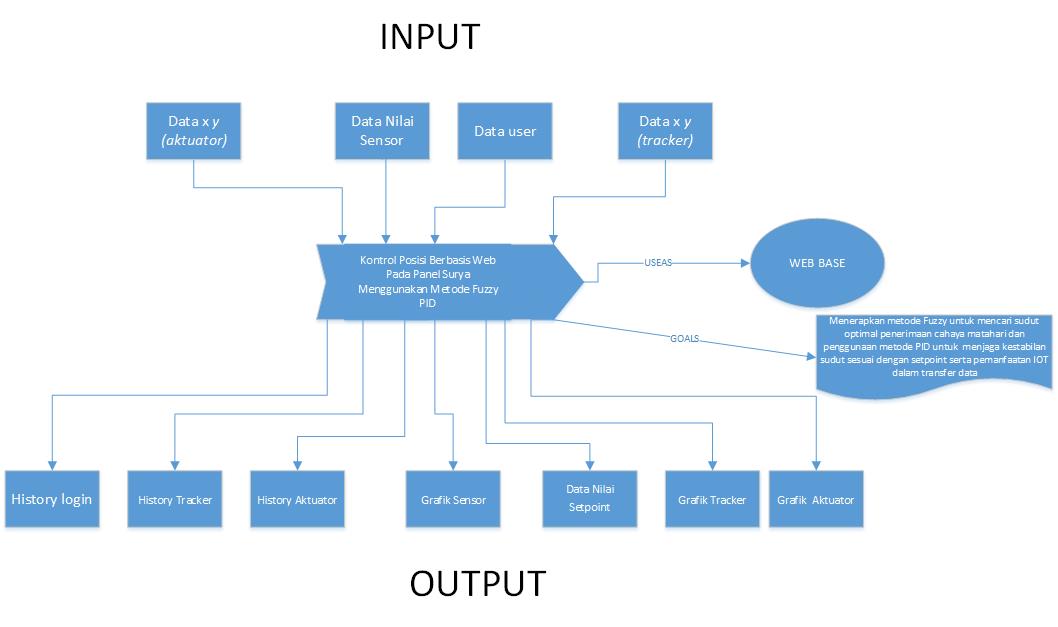
Kebutuhan non fungsional adalah kebutuhan sekunder yang dimiliki sistem yang secara tidak langsung berkaitan dengan sistem yang dibangun (Febrianto, 2015). Kebutuhan non fungsional ini akan dijelaskan pada Tabel 3.2.

Tabel 4.1.B Kebutuhan Non Fungsional

|  |  |
| --- | --- |
| SRSNF ID | Identifikasi |
| SRSNF\_1 | *Availability,* sistem harus dapat diakses selama proses penelitian ini yaitu mulai dari jam 5 pagi sampai jam 6 malam selama satu hari. |
| SRSNF\_2 | *Response Time*, sistem harus dapat memproses request data secara real time. |
| SRSNF\_3 | Security, sistem harus memiliki hak akses dalam pengaksesan sistem. |
| SRSNF\_4 | *User Friendly,* sistem harus didesain untuk memudahkan pengguna. |
| SRSNF\_5 | *Realibility.* sistem harus berjalan sesuai kebutuhan pengguna |

## *Business Process*

*Business process* merupakan kumpulan proses yang berisi aktifitas yang saling berelasi yang menghasilkan keluaran dari suatu proses dengan adanya data masukan guna mencapai suatu tujuan (Febrianto, 2015). Pada Gambar 4.1 *business process* penelitian ini memiliki masukan data berupa data user, data sudut x dan y pada *actuator*, data sudut x dan y pada *tracker*, dan data nilai sensor. Sedangkan untuk keluaran nya berupa data *nilai setpoint*, history *login,* history *tracker* dan aktuator, grafik sensor, dan grafik aktuator serta tracker.



Gambar 4.1 Business Process

## *Usecase Diagram*

*Usecase diagram* pada sistem berguna untuk menggambarkan fitur-fitur yang akan dibuat pada sistem. Selain itu juga berguna untuk menggambarkan interaksi antara aktor dengan sistem. Usecase pada sistem ini sesuai pada Gambar 4.2. *Usecase diagram* pada penelitian ini memiliki 2 aktor yaitu admin dan user. Definisi untuk setiap aktor dapat dilihat pada Tabel 4.3.

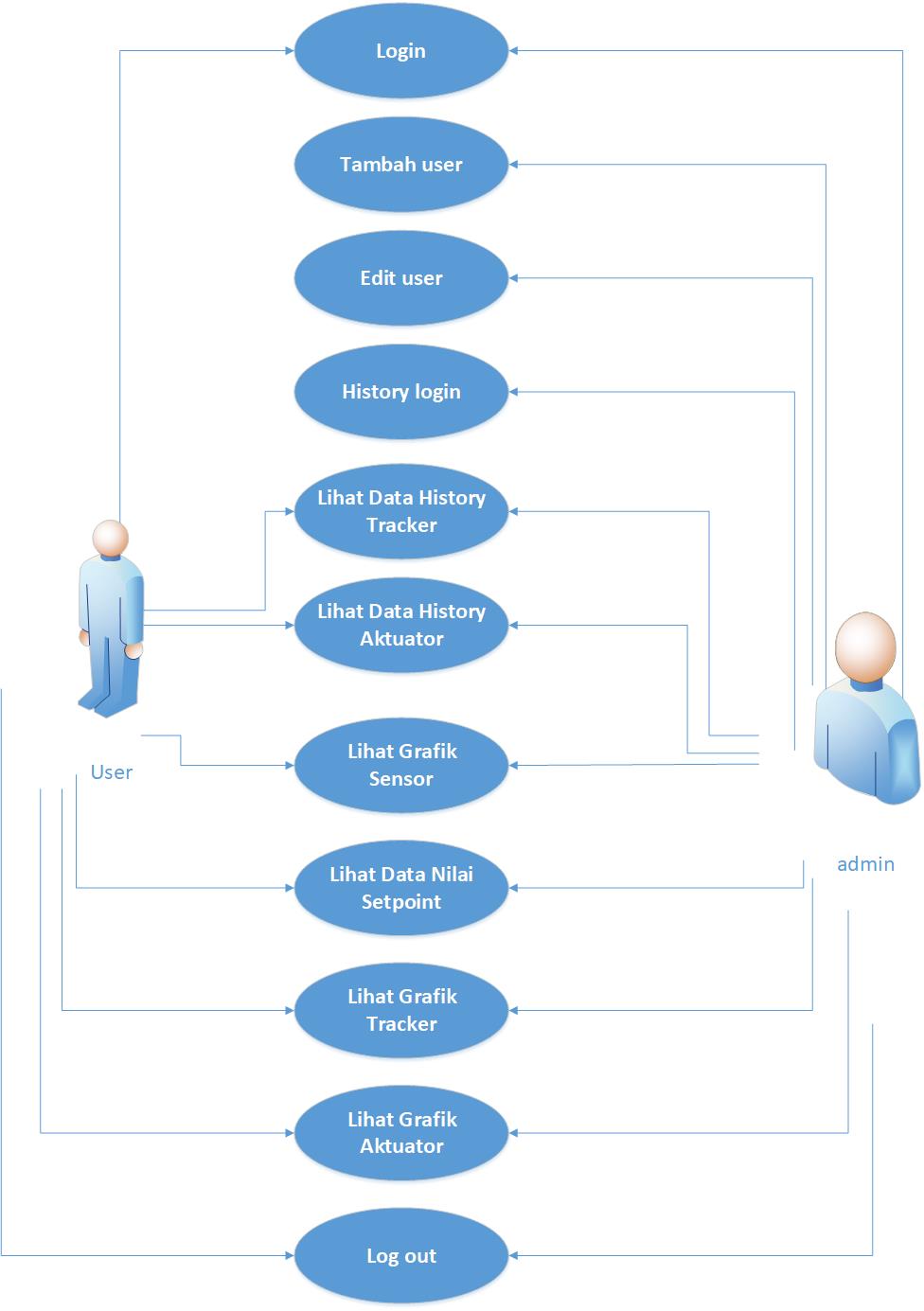
Tabel 4.3.A Definisi Tugas

|  |  |  |
| --- | --- | --- |
| No | Aktor | Definisi Tugas |
| 1 | Admin | Dapat melakukan semua proses mulai dari penambahan user sampai simulasi |
| 2 | User | Memantau grafik perolehan energi (arus dan tegangan), posisi *tracker*, posisi aktuator dan melakukan simulasi |

*Usecase Diagam* memiliki fitur-fitur pada sistem yang akan diakses oleh aktor dengan hak akses masing-masing. Berikut adalah definisi dari masing-masing usecase yang dapat dilihat pada Tabel 4.4

Tabel 4.3.B Deskripsi Usecase

|  |  |  |
| --- | --- | --- |
| No | Nama | Definisi |
| 1 | Login | Fitur untuk user login yang bisa diakses oleh semua aktor |
| 2 | Tambah user | Fitur untuk menambah data user yang bisa dilakukan oleh admin |
| 3 | Edit user | Fitur ubah data user yang bias dilakukan oleh admin |
| 4 | History login | Fitur yang berisi detail login (nama aktor dan tanggal login) yang bisa diakses oleh admin |
| 5 | Lihat data *History Tracker* | Fitur yang berisi data *history* sudut pada *tracker* yang dapat diakses oleh semua aktor |
| 6 | Lihat data *History* Aktuator | Fitur yang berisi data *history* sudut pada aktuator yang dapat diakses oleh semua aktor |
| 7 | Lihat grafik sensor | Fitur yang menampilkan data grafik sensor yang telah didapat dari 4 sensor LDR. Fitur ini dapat dilihat oleh semua aktor |
| 8 | Lihat data nilai *setpoint* | Fitur yang menampilkan nilai *setpoint* yang didapat dari perhitungan dari nilai sensor dengan metode fuzzy. |
| 9 | Lihat grafik *tracker* | Fitur yang menampilkan grafik dari data *tracker* yang dapat dilihat oleh semua aktor |
| 10 | Lihat grafik aktuator | Fitur yang menampilkan grafik yang dapat dilihat oleh semua aktor |
| 11 | Logout | Fitur untuk keluar dari sistem. Fitur ini dapat diakses oleh semua aktor |



Gambar 4.2 Usecase Diagram

## Skenario

Skenario merupakan alur yang menjelaskan proses pada setiap usecase. Skenario berisi nama usecase, aktor, pre-condition. Post condition, scenario utama dan scenario alternatif.

### Skenario *Log In*

Skenario ini menjelaskan alur masuk ke dalam sistem. Seluruh aktor yang akan memasuki sistem harus memasukkan *username* dan *password* yang dimiliki. Jika *usernam*e dan *password* yang dimasukkan benar maka akan berhasil masuk ke sistem, sebaliknya jika salah maka akan muncul pemberitahuan jika *username* atau *password* salah. Skenario login dapat dilihat pada Tabel 4.5.

### Skenario Tambah *User*

Skenario ini menjelaskan alur menambahkan pengguna yang bisa mengakses sistem. Fitur untuk menambah data pengguna hanya dapat dilakukan oleh admin. Admin harus masuk pada menu tambah pengguna terlebih dahulu. Setelah itu mengisikan dan *password.* Data pengguna berhasil ditambahkan jika inputan sudah benar. Sebaliknya jika inputan salah maka akan muncul peringatan bahwa data yang dimasukkan salah atau tidak sesuai. Skenario tambah user dapat dilihat pada Tabel 3.6.

### Edit Us*er*

Skenario ini menjelaskan alur untuk mengubah data *user*. Fitur untuk mengubah data *user* bisa dilakukan oleh admin. Mengubah data *user* dapat dilakukan dengan menekan tombol edit pada masing-masing baris data yang diinginkan. Setelah itu ubah data *username* maupun *password* pada form yang telah disediakan. Klik tombol *update* untuk menyimpan perubahan data. Jika data yang dimasukkan benar maka data akan tersimpan, jika salah maka akan muncul *pop up* pemberitahuan. Skenario edit user dapat dilihat pada Tabel 3.7.

### History Log In

Skenario ini menjelaskan alur untuk melihat *history log in* dari user yang telah mengakses sistem. Fitur untuk melihat history log in bisa dilakukan oleh admin. Admin harus masuk kedalam sistem terlebih dahulu. Setelah itu klik menu *History* dan pilih menu *History Log in*, maka akan menampilkan tabel siapa saja yang pernah mengakses sistem pada waktu tertentu. Skenario history login dapat dilihat pada Tabel 3.8.

### Lihat Data *History Tracker*

Skenario ini menjelaskan alur untuk melihat data *history tracker*. Fitur ini bisa dilakukan oleh semua *user*. Setiap *user* harus masuk ke dalam sistem terlebih dahulu, lalu memilih menu *history tracker*. Maka, sistem akan menampilkan data sudut *tracker*. Skenario lihat data history *tracker* dapat dilihat pada Tabel 4.9.

### Lihat Data History Aktuator

Skenario ini menjelaskan alur untuk melihat *history* aktuator pada aktuator. Fitur ini bisa dilakukan oleh semua *user*. Setiap *user* harus masuk ke dalam sistem terlebih dahulu, lalu memilih menu *history* aktuator. Maka, sistem akan menampilkan data sudut aktuator. Skenario lihat data history aktuator dapat dilihat pada Tabel 4.10.

### Lihat *Grafik Sensor*

Skenario ini menjelaskan alur untuk melihat grafik sensor secara *realtime*. Fitur ini bisa dilakukan oleh semua *user*. Setiap *user* harus masuk ke dalam sistem terlebih dahulu, lalu memilih menu grafik sensor. Maka, sistem akan menampilkan grafik sensor dalam rentang waktu tertentu. Skenario lihat grafik sensor dapat dilihat pada Tabel 4.11.

### Lihat Nilai *Setpoint*

Skenario ini menjelaskan alur untuk melihat nilai setpoint. Fitur ini bisa dilakukan oleh semua *user*. Setiap *user* harus masuk ke dalam sistem terlebih dahulu, lalu memilih menu nilai sensor Maka, sistem akan menampilkan nilai sensor dalam rentang waktu tertentu. Skenario lihat data nilai snsor dapat dilihat pada Tabel 4.12.

### Lihat Grafik *Tracker*

Skenario ini menjelaskan alur untuk melihat grafik *tracker* secara *realtime*. Fitur ini bisa dilakukan oleh semua *user*. Setiap *user* harus masuk ke dalam sistem terlebih dahulu, lalu memilih menu grafik *tracker*. Maka, sistem akan menampilkan grafik *tracker* dalam rentang waktu tertentu. Skenario lihat grafik *tracker* dapat dilihat pada Tabel 4.13.

### Lihat Grafik Aktuator

Skenario ini menjelaskan alur untuk melihat grafik aktuator secara *realtime*. Fitur ini bisa dilakukan oleh semua *user*. Setiap *user* harus masuk ke dalam sistem terlebih dahulu, lalu memilih menu grafik aktuator. Maka, sistem akan menampilkan grafik aktuator dalam rentang waktu tertentu. Skenario lihat grafik aktuator dapat dilihat pada Tabel 4.14.

### Log out

Skenario ini menjelaskan alur untuk keluar dari sistem. Fitur ini bisa dilakukan oleh semua *user*. *Use*r yang ingin keluar dari sistem hanya perlu memilih menu log out maka akan langsung kembali menampilkan halaman log in. Skenario *log out* dapat dilihat pada Tabel 4.15.

## Activity Diagram

*Activity Diagram* merupakan bentuk visual dari alur kerja sistem yang berisi aktivitas dan tindakan, yang juga dapat berisi pilihan maupun pengulangan. *Activity Diagram* dibuat untuk menjelaskan aktivitas sistem.

### Log in

*Aktivity diagram* *log in* sistem dapat dilihat pada Gambar 3.4. Pengguna mengisi username dan password pada halaman awal sistem ketika ingin masuk ke dalam sistem. Setelah itu sisitem akan mengecek apakah username dan password salah. Ketika username dan password benar maka pengguna akan diarahkan untuk masuk ke halaman dashboard. Tetapi ketika salah maka akan muncul pop up dan ketika klik ok, maka akan kembali ke halaman login.

### Tambah user

*Aktivity diagram* tambah user dapat dilihat pada Gambar 3.5. Pengguna mengisi nama, username dan password pada form tambah user . Setelah itu sisitem akan mengecek apakah data yang diinputkan sudah lengkap. Ketika datayang diisikan benar, maka pengguna akan diarahkan pada tampilan data user. Tetapi ketika data salah maka akan muncul pop up setelah klik ok, maka akan kembali ke form tambah user.

### Edit *user*

*Aktivity diagram* edit *user* dapat dilihat pada Gambar 3.6. Pengguna mengisi nama, username dan password pada form edit *user*. Setelah itu sistem akan mengecek apakah data yang di masukkan sudah lengkap. Ketika data yang di masukkan benar, maka pengguna akan diarahkan pada tampilan data user. Tetapi ketika data salah maka akan muncul pop up peringatan bahwa data yang dimasukkan tidak lengkap. Jika pada pop up memilih klik ok, maka akan kembali ke form edit *user*.

History login

*Aktivity diagram* history login dapat dilihat pada Gambar 3.7. Pengguna dapat melihat data siapa saja yang memasuki sistem pada waktu tertentu dengan memilih menu *history log in*. Maka, sistem akan menampilkan data siapa saja yang telah memasuki sistem pada waktu tertentu

### Lihat Data History Aktuator

*Aktivity diagram* lihat data sudut aktuator dapat dilihat pada Gambar 3.8. Pengguna dapat melihat sudut posisi aktuator pada waktu tertentu dengan memilih menu lihat sudut aktuator. Maka, sistem akan menampilkan data sudut aktuator.

### Lihat Data *History Tracker*

*Aktivity diagram* lihat sudut *tracker* dapat dilihat pada Gambar 3.9. Pengguna dapat melihat sudut posisi *tracker* pada waktu tertentu dengan memilih menu lihat sudut *tracker*. Maka, sistem akan menampilkan data sudut *tracker*.

### Lihat Grafik Sensor

*Aktivity diagram* lihat Lihat Grafik Sensor dapat dilihat pada Gambar 3.10. Pengguna dapat melihat Lihat Grafik Sensor sesuai perubahan yang terjadi setiap periode waktu dengan memilih menu lihat *Lihat Grafik Sensor*. Maka, sistem akan menampilkan data *Lihat Grafik Sensor* .

### Lihat Nilai Setpoint

*Aktivity diagram* lihat nilai *setpoint* dapat dilihat pada Gambar 3.11. Pengguna dapat melihat nilai *setpoint* yang terjadi setiap periode waktu dengan memilih menu lihat nilai *setpoint*. Maka, sistem akan menampilkan nilai *setpoint*.

### Lihat Grafik *Tracker*

*Aktivity diagram* lihat grafik *tracker* dapat dilihat pada Gambar 3.12. Pengguna dapat melihat grafik *tracker* yang masuksecara *realtime* dengan memilih menu grafik dan sub menu lihat grafiktracker. Maka, sistem akan menampilkan grafik *tracker.*

### Lihat Grafik Aktuator

*Aktivity diagram* lihat grafik aktuator dapat dilihat pada Gambar 3.12. Pengguna dapat melihat grafik aktuator yang masuksecara *realtime* dengan memilih menu grafik dan sub menu lihat grafik aktuator. Maka, sistem akan menampilkan grafikaktuator*.*

### Log out

*Aktivity diagram* log out dapat dilihat pada gambar 3.15. Pengguna dapat keluar dari sistem dengan memilih menu *log out.* Maka sistem akan kembali menampilkan halaman *log in.*

## Sequence Diagram

Diagram yang menggambarkan interaksi antar objek yang dapat berkomunikasi. Diagram ini juga berisi pesan yang terjadi antara objek-objek untuk melakukan suatu tugas atau aksi tertentu.

### Log In

*Sequence diagram* *log in* sistem menggambarkan proses interaksi objek pada proses memasuki sistem. Ketika pengguna memilih tombol *log in* maka *view*  mengirimkan data *username* dan *password* pada *controller* untuk memproses kebenaran data. Jika data yang dikirimkan sesuai maka tampilan akan berpindah ke *dashboard* masing-masing pengguna. Sebaliknya jika salah maka akan kembali lagi ke halaman *log in.* Proses ini dapat dilihat pada Gambar 3.16.

### Tambah User

*Sequence diagram* tambah user menggambarkan proses interaksi objek pada proses penambahan data user yang bisa mengakses sistem. Ketika pengguna memilih tombol tambah user maka *view*  mengirimkan data *username* dan *password* pada *controller* untuk memproses kebenaran data. Jika data yang dikirimkan sesuai maka akan menampilkan *pop up* “berhasil simpan”. Sebaliknya jika salah maka akan muncul *pop up* “inputan salah”*.* Proses ini dapat dilihat pada Gambar 3.17.

### Edit user

*Sequence diagram* edit user menggambarkan proses interaksi objek pada proses mengubah data user yang bisa mengakses sistem. Ketika pengguna memilih tombol edit user maka *view*  mengirimkan data *username* dan *password* pada *controller* untuk memproses kebenaran data. Jika data yang dikirimkan sesuai maka akan menampilkan *pop up* “berhasil simpan”. Sebaliknya jika salah maka akan muncul *pop up* “inputan salah”*.* Proses ini dapat dilihat pada Gambar 3.18.

### History Login

*Sequence diagram* lihat data *history login* menggambarkan interaksi objek pada proses menampilkan datanya. Ketika pengguna memilih menu lihat *history login* maka *view* meminta data pada *controller* untuk menampilkan halaman lihat history sudut aktuator. Saat itu juga *controller* mengambil data dari database melalui *model.* Data tersebut dari model yang beruapa *array* dikirim melalui *controller* ke *view*. Langkah terakhir view mengubah data *array* tersebut dalam bentuk tabel sehingga bisa dilihat oleh pengguna. Proses ini dapat dilihat pada gambar 3.20.

### Lihat Data *History* Aktuator

*Sequence diagram* lihat data *history* aktuator menggambarkan interaksi objek pada proses menampilkan data sudut aktuator. Ketika pengguna memilih menu lihat history sudut aktuator maka *view* meminta data pada *controller* untuk menampilkan halaman lihat *history* aktuator. Saat itu juga *controller* mengambil data dari database melalui *model.* Data tersebut dari model yang beruapa *array* dikirim melalui *controller* ke *view*. Langkah terakhir view mengubah data *array* tersebut dalam bentuk tabel sehingga bisa dilihat oleh pengguna. Proses ini dapat dilihat pada gambar 3.20.

### Lihat Data *History Tracker*

*Sequence diagram* lihat data *history* *tracker* menggambarkan interaksi objek pada proses menampilkan data sudut aktuator. Ketika pengguna memilih menu lihat history sudut *tracker* maka *view* meminta data pada *controller* untuk menampilkan halaman lihat history sudut aktuator. Saat itu juga *controller* mengambil data dari database melalui *model.* Data tersebut dari model yang beruapa *array* dikirim melalui *controller* ke *view*. Langkah terakhir view mengubah data *array* tersebut dalam bentuk tabel sehingga bisa dilihat oleh pengguna. Proses ini dapat dilihat pada gambar 3.20.

### Lihat Grafik Sensor

*Sequence diagram* lihat grafik sensor menggambarkan interaksi objek pada proses menampilkan grafik sensor. Ketika pengguna memilih menu lihat grafik sensor maka *view* meminta data pada *controller* untuk menampilkan halaman lihat grafik sensor. Saat itu juga *controller* mengambil data dari database melalui *model.* Data tersebut dari model yang beruapa *array* dikirim melalui *controller* ke *view*. Langkah terakhir view mengubah data *array* tersebut dalam bentuk grafik sehingga bisa dilihat oleh pengguna. Proses ini dapat dilihat pada gambar 3.20.

### Lihat Nilai *Setpoint*

*Sequence diagram* nilai *setpoint* menggambarkan interaksi objek pada proses menampilkannya. Ketika pengguna memilih menu lihat nilai *setpoint* maka *view* meminta data pada *controller* untuk menampilkan halaman lihat grafik sensor. Saat itu juga *controller* mengambil data dari database melalui *model.* Data tersebut dari model yang beruapa *array* dikirim melalui *controller* ke *view*. Langkah terakhir view mengubah data *array* tersebut dalam bentuk tabel sehingga bisa dilihat oleh pengguna. Proses ini dapat dilihat pada gambar 3.20.

### Lihat Grafik *Tracker*

*Sequence diagram* lihat grafik *tracker* menggambarkan interaksi objek pada proses menampilkannya secara *realtime*. Ketika pengguna memilih menu lihat grafik *tracker* maka *view* meminta data pada *controller* untuk menampilkan halaman lihat grafik *tracker*. Saat itu juga *controller* mengambil data dari database melalui *model.* Data tersebut dari model yang beruapa *array* dikirim melalui *controller* ke *view*. Pada interval tertentu *controller* mengirim data baru ke view agar data dapat berubah secara *realtime.* Data tersebut berasal dari sensor yang berada pada *tracker*. Langkah terakhir view mengubah data *array* tersebut dalam bentuk grafik sehingga bisa dilihat oleh pengguna. Proses ini dapat dilihat pada gambar 3.20.

### Lihat GrafikAktuator

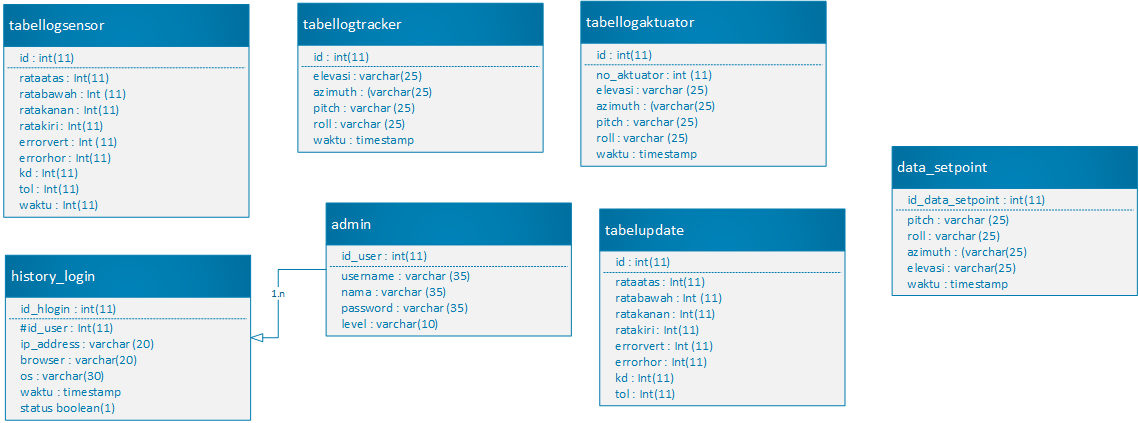
*Sequence diagram* lihat grafik aktuator menggambarkan interaksi objek pada proses menampilkannya secara *realtime*. Ketika pengguna memilih menu lihat grafik aktuator maka *view* meminta data pada *controller* untuk menampilkan halaman lihat grafik aktuator. Saat itu juga *controller* mengambil data dari database melalui *model.* Data tersebut beruapa *array* yang dikirim melalui *controller* ke *view*. Pada interval tertentu *controller* mengirim data baru ke view agar data dapat berubah secara *realtime.* Data tersebut berasal dari aktuator. Langkah terakhir view mengubah data *array* tersebut dalam bentuk grafik sehingga bisa dilihat oleh pengguna. Proses ini dapat dilihat pada gambar 3.20.

### Log out

*Sequence diagram* logout menggambarkan interaksi objek pada proses menampilkan perubahan-perubahan sudut y secara realtime. Ketika pengguna memilih menu *log out. Controller*  akan menampilkan kembali halaman *log in.* Proses ini dapat dilihat pada gambar 3.25.

## *Entity Relationship Diagram* (ERD)

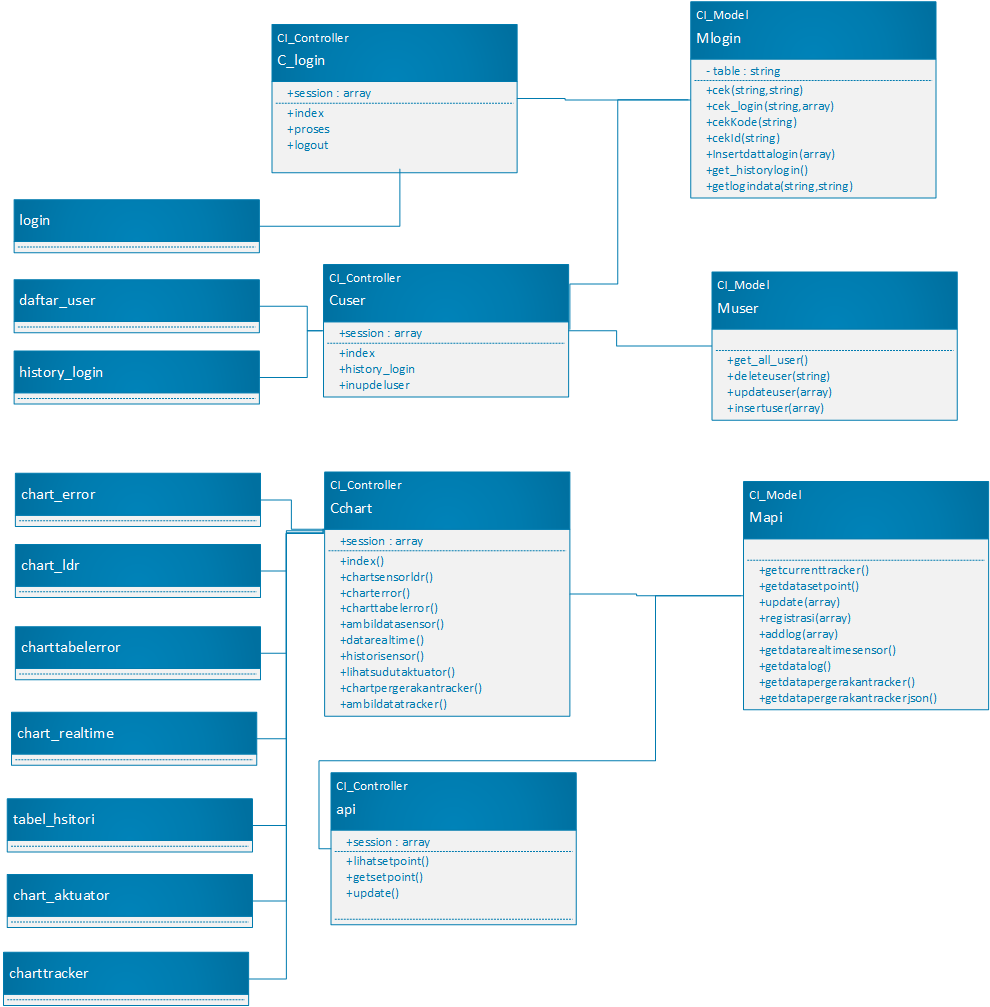
Diagram yang menggambarkan interaksi antar tabel yang dapat berkomunikasi. Diagram ini juga berisi ketergantungan antara tabel satu dengan tabel yang lainnya untuk kebutuhan data. Pada penelitian ini terdapat tujuh tabel yang digunakan untuk menyimpan data. Dari keseluruhan tabel, hanya ada dua tabel yang memiliki ketergantungan data yaitu tabel admin dan history\_login. Sedangkan untuk tabel lain nya berfungsi sebagai tempat penyimpanan data tanpa terhubung satu dengan yang lain nya.



Gambar 4.3 ERD

## *Class Diagram*

Diagram yang menggambarkan interaksi antar *class* pada sistem seperti *class model, view* dan *controller*. *Class Diagram* memiliki 3 bagian utama yaitu attribute, operation, dan name. *Class* yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem.



Gambar 4.4 Class Diagram

## Desain *User Interface* (UI)

Desain *User Interface* (UI) adalah desain yang digunakan sebagai acuan tampilan yang diterapkan pada sistem. Tujuan dari penggunaan desain ini adalah untuk membuat interaksi pengguna menjadi sederhana dan efisien. Hal ini sangat penting karena akan mempengaruhi pengguna dalam menggunakan sistem yang telah dibangun. Apabila suatu program sulit untuk digunakan, maka akan mengakibatkan pengguna untuk melakukan suatu kesalahan saat menggunakan program tersebut.

### *User Interface Log In*

*User Interface Log In* menggambarkan desain tampilan untuk masuk ke dalam sistem. Seluruh aktor yang akan memasuki sistem harus memasukkan *username* dan *password* yang dimiliki. Jika *usernam*e dan *password* yang dimasukkan benar maka akan berhasil masuk ke sistem, sebaliknya jika salah maka akan muncul pemberitahuan jika *username* atau *password* salah. *User Interface Log In* dapat dilihat pada

### *User Interface* Tambah User

*User Interface* tambah user menggambarkan desain tampilan untuk menambahkan pengguna yang bisa mengakses sistem. Fitur untuk menambah data pengguna hanya dapat dilakukan oleh admin. Admin harus masuk pada menu tambah pengguna terlebih dahulu. Setelah itu mengisikan dan *password.* Data pengguna berhasil ditambahkan jika inputan sudah benar. Sebaliknya jika inputan salah maka akan muncul peringatan bahwa data yang dimasukkan salah atau tidak sesuai. *User interface* tambah user dapat dilihat pada

### *User Interface* Edit User

*User Interface* edit user menggambarkan desain tampilan untuk mengubah data *user* bisa dilakukan oleh admin. Mengubah data *user* dapat dilakukan dengan menekan tombol edit pada masing-masing baris data yang diinginkan. Setelah itu ubah data *username* maupun *password* pada form yang telah disediakan. Klik tombol *update* untuk menyimpan perubahan data. Jika data yang dimasukkan benar maka data akan tersimpan, jika salah maka akan muncul *pop up* pemberitahuan. *User interface* edit user dapat dilihat pada

### *User Interface History Log in*

*User Interface* *history log in* menggambarkan desain tampilan untuk menampilkan data *history log in*. Fitur untuk melihat data history log in bisa dilakukan oleh admin. Admin harus masuk kedalam sistem terlebih dahulu. Setelah itu klik menu *History* dan pilih menu *History Log in*, maka akan menampilkan tabel siapa saja yang pernah mengakses sistem pada waktu tertentu. *User interface history log in* dapat dilihat pada

### *User Interface History Tracker*

*User Interface* *tracker* menggambarkan desain tampilan untuk menampilkan data *history log in*. Fitur untuk melihat data *history tracker*  bisa dilakukan oleh semua pengguna. Pengguna harus masuk kedalam sistem terlebih dahulu. Setelah itu klik menu *History Tracker*, maka akan menampilakn tabel sudut yang pernah dilalui *tracker* pada waktu tertentu. *User interface history tracker* dapat dilihat pada

### *User Interface History* Aktuator

*User Interface* *history log in* menggambarkan desain tampilan untuk menampilkan data *history log in*. Fitur untuk melihat data *history* aktuator bisa dilakukan oleh semua pengguna. Pengguna harus masuk kedalam sistem terlebih dahulu. Setelah itu klik menu *history* aktuator, maka akan menampilakn tabel sudut yang pernah dilalui aktuator pada waktu tertentu. *User interface history* aktuatordapat dilihat pada

### *User Interface* Grafik Sensor

*User Interface* grafik sensor menggambarkan desain tampilan untuk menampilkan data grafik sensor Fitur untuk melihat data grafik sensor bisa dilakukan oleh semua pengguna. Pengguna harus masuk kedalam sistem terlebih dahulu. Setelah itu klik menu grafik pilih grafik sensor, maka akan menampilakn grafik sensor secara *realtime*. *User interface* grafik sensor dapat dilihat pada

### *User Interface* Nilai *Setpoint*

*User Interface* ini menggambarkan desain tampilan untuk menampilkan data nilai *setpoint*. Fitur untuk melihat data nilai *setpoint* bisa dilakukan oleh semua pengguna. Pengguna harus masuk kedalam sistem terlebih dahulu. Setelah itu klik menu nilai *setpoint*, maka akan menampilakan nilai setpoint yang digunakan saat itu. *User interface* niali *setpoint* dapat dilihat pada

### *User Interface* Grafik *Tracker*

*User Interface* grafik *tracker* menggambarkan desain tampilan untuk menampilkan data grafik sensor Fitur untuk melihat data grafik *tracker* bisa dilakukan oleh semua pengguna. Pengguna harus masuk kedalam sistem terlebih dahulu. Setelah itu klik menu grafik pilih grafik *tracker*, maka akan menampilakn grafik *tracker* secara *realtime*. *User interface* grafik *tracker* dapat dilihat pada

### *User Interface* Grafik Aktuator

*User Interface* grafik aktuator menggambarkan desain tampilan untuk menampilkan data grafik aktuator Fitur untuk melihat data grafik aktuator bisa dilakukan oleh semua pengguna. Pengguna harus masuk kedalam sistem terlebih dahulu. Setelah itu klik menu grafik pilih grafik aktuator, maka akan menampilakn grafik aktuator secara *realtime*. *User interface* grafik aktuator dapat dilihat pada

### *User Interface Log Out*

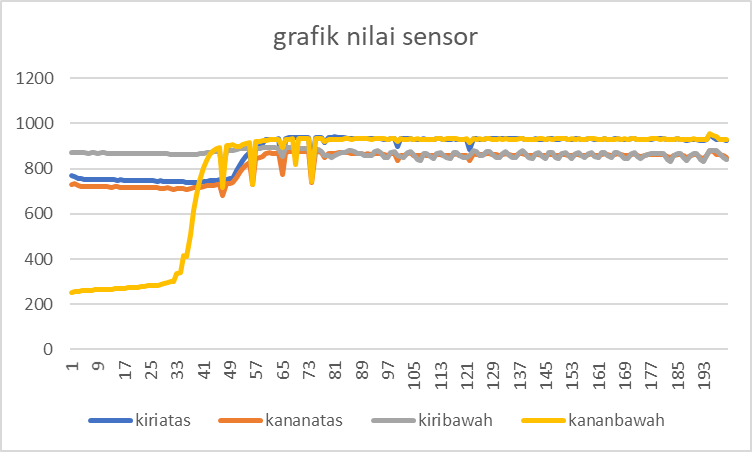
Fitur ini menggambakarkan desain tampilan untuk keluar dari sistem. Fitur ini bisa dilakukan oleh semua *user*. *Use*r yang ingin keluar dari sistem hanya perlu memilih menu log out maka sistem akan langsung kembali menampilkan halaman log in. *User interface* *log out* dapat dilihat pada .

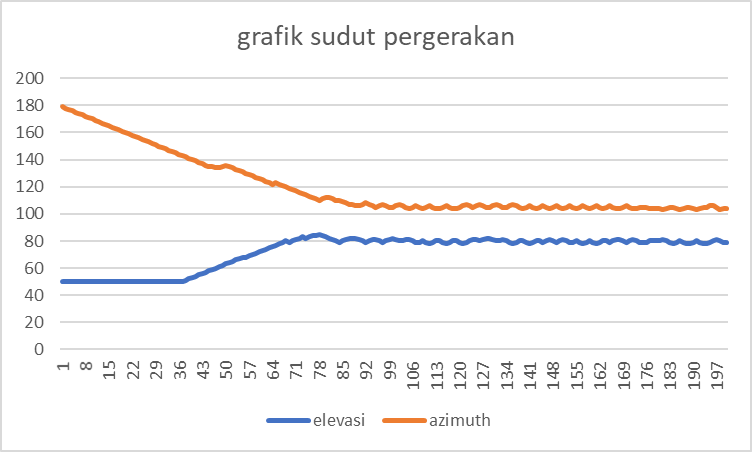
## Pengujian Sistem

Pada Penelitian ini pengujian sistem dilakukan dengan metode *black box testing.* Metode ini dilakukan dengan cara menguji fitur – fitur maupun rancangan yang telah dibuat untuk dibandingkan dengan yang sudah diimplementasikan.

### Pengujian *Solar Tracker* Tanpa Metode *Fuzzy*

Pengujian dilakukan dengan melihat langkah/iterasi dari proses menuju arah penerimaan sinar matahari paling optimal.Percobaan *solar* *tracker* menggunakan cahaya lampu sebagai pengganti sinar matahari dengan diatur pada sudut tertentu. Hasil percobaan *solar* *tracker* tanpa menggunakan metode *Fuzyy* dengan kondisi lampu di sudut 80 derajat.



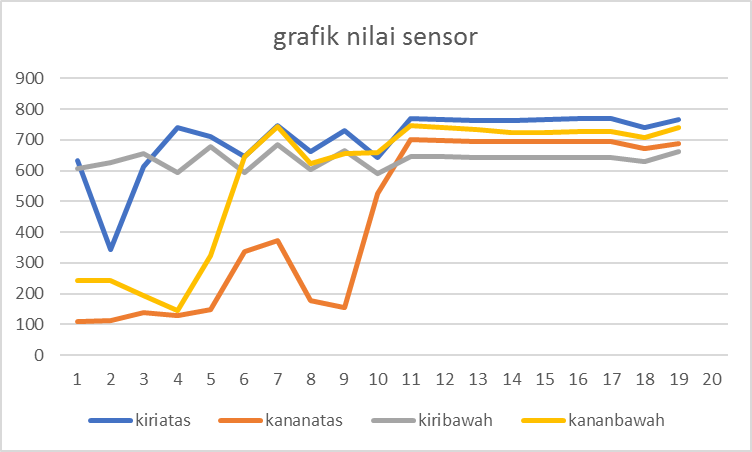


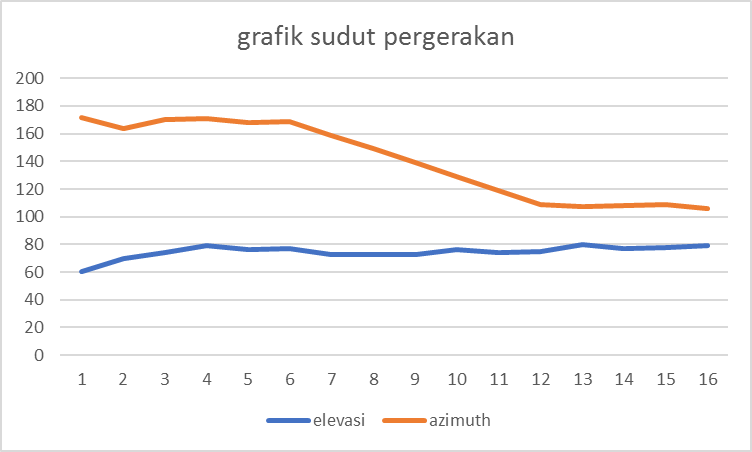


Dari hasil tersebut menunjukan bahwa untuk menstabilkan nilai perbedaan di angka nol sangat sulit dikarenakan nilai intensitas yang diterima sensor sangat variatif dan selalu berubah-ubah.

### Pengujian *Solar Tracker* dengan Metode *Fuzzy*

kondisi percobaan *solar* *tracker* dengan metode *Fuzyy* sama seperti sebelum nya yaitu menggunakan sinar lampu pada sudut 80 derajat. Hasil pengujian *Fuzyy* dapat dilihat pada Tabel 4.1.





Dari hasil tersebut menunjukan bahwa metode *Fuzyy* membantu pergerakan *tracker* lebih optimal dengan mengubah kecepatan motor saat nilai perbedaan tinggi dan semakin lambat saat mendekati antara nilai -10 sampai 10 (himpunan keanggotaan *zero*). Hal ini berdampak pada waktu tempuh mencapai tujuan dan akan stabil di range tersebut.

### Pengujian Aktuator dengan MetodePID

Pengujian aktuator diuji dengan cara mengkoneksikan modul aktuator dengan jaringan yang terdapat server web sistem atau web sistem online yang sudah ada sebelumnya. Setelah terhubung maka aktuator akan merequest atau meminta nilai setpoint ke server. Interval request peneliti set setiap 10 menit sekali . Setelah mendapatkan nilai setpoint maka aktuator akan menggerakkan motor penggerak dengan acuan setpoint yang di dapat. Berikut flowchart alur pengiriman dan permintaan setpoint.

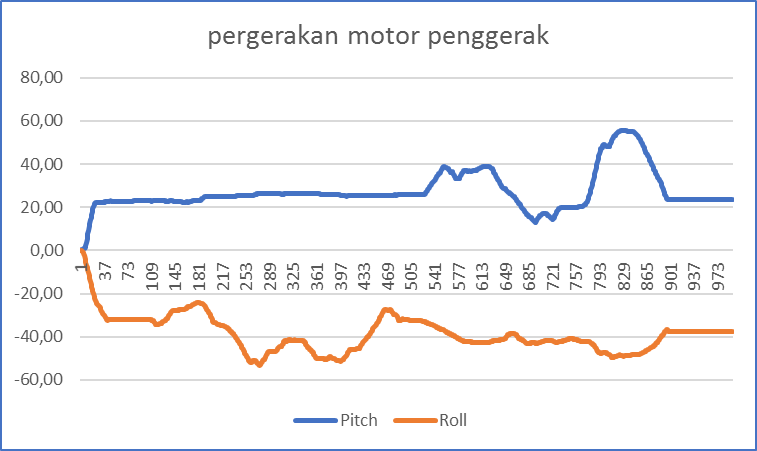
Setelah setpoint di dapatkan maka aktuaotr akan menggerakan motor pengggerak dengan hasil pergerakan berupa sudut dan posisi motor dibawah ini

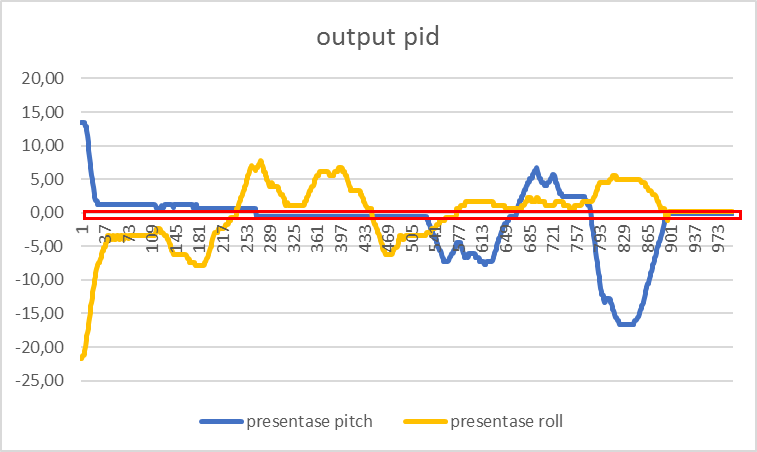
### Pengujian Aktuator dengan MetodePID

Metode PID diuji dengan cara mengkoneksikan modul aktuator dengan jaringan yang terdapat server web sistem yang sudah ada sebelumnya. Setelah terhubung maka aktuator akan merequest atau meminta nilai setpoint ke server yang digunakan dalam perhitungan PID. Interval request peneliti set setiap 10 menit sekali . Setelah mendapatkan nilai setpoint maka PID akan membantu pergerakan motor penggerak dengan acuan setpoint yang di dapat. Berikut flowchart alur pengiriman dan permintaan setpoint.



Setelah setpoint di dapatkan metode PID akan menggerakan motor pengggerak dengan hasil pergerakan berupa sudut dan waktu tempuh dibawah ini

****

****

# HASIL DAN PEMBAHASAN

Bab ini menggambarkan tentang hasil dari penelitian yang telah dilakukan. untuk menjawab rumusan masalah sehingga dapat mewujudkan tujuan sebenarnya dari penelitian. Pada bab ini akan dijelaskan tentang pengujian dan hasil dari pengujian yang telah dilakukan dengan mengimplementasikan metode *Fuzyy* dan PID.

## Hasil Pembangunan Sistem

Implementasi pembangunan sistem harus sesuai dengan desain yang telah dibuat sebelumnya. Hal ini digunakan sebagai acuan tampilan maupun alur yang diterapkan pada sistem untuk membuat interaksi pengguna menjadi sederhana dan efisien. Berikut ini adalah desain yang telah berhasil diterapkan sesuai pada keterangan Tabel 5.1.

Tabel 5.1.1 Implementasi Desain User Interface

|  |  |  |  |
| --- | --- | --- | --- |
| No | Nama Fitur | Definisi | Keterangan |
| 1 | Login | Fitur untuk user login yang bisa diakses oleh semua aktor | Berhasil diimplemantasikan |
| 2 | Tambah user | Fitur untuk menambah data user yang bisa dilakukan oleh admin | Berhasil diimplemantasikan |
| 3 | Edit user | Fitur ubah data user yang bias dilakukan oleh admin | Berhasil diimplemantasikan |
| 4 | History login | Fitur yang berisi detail login (nama aktor dan tanggal login) yang bisa diakses oleh admin | Berhasil diimplemantasikan |
| 5 | Lihat data *History Tracker* | Fitur yang berisi data *history* sudut pada *tracker* yang dapat diakses oleh semua aktor | Berhasil diimplemantasikan |
| 6 | Lihat data *History* Aktuator | Fitur yang berisi data *history* sudut pada aktuator yang dapat diakses oleh semua aktor | Berhasil diimplemantasikan |
| 7 | Lihat grafik sensor | Fitur yang menampilkan data grafik sensor yang telah didapat dari 4 sensor LDR. Fitur ini dapat dilihat oleh semua aktor | Berhasil diimplemantasikan |
| 8 | Lihat data nilai *setpoint* | Fitur yang menampilkan nilai *setpoint* yang didapat dari perhitungan dari nilai sensor dengan metode fuzzy. | Berhasil diimplemantasikan |
| 9 | Lihat grafik *tracker* | Fitur yang menampilkan grafik dari data *tracker* yang dapat dilihat oleh semua aktor | Berhasil diimplemantasikan |
| 10 | Lihat grafik aktuator | Fitur yang menampilkan grafik yang dapat dilihat oleh semua aktor | Berhasil diimplemantasikan |
| 11 | Logout | Fitur untuk keluar dari sistem. Fitur ini dapat diakses oleh semua aktor | Berhasil diimplemantasikan |

### Fitur *Log In*

Fitur *Log In* merupakan fitur yang digunakan sebagai pintu untuk masuk ke dalam sistem. Hal ini dpat dilakukan dengan memasukkan username dan *password*  pada form yang telah disediakan. Setelah itu klik tombol *log in* sesuai pada Gambar 5.1.

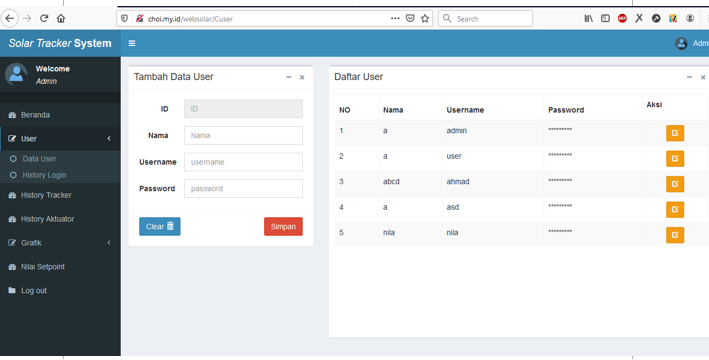


Gambar 5.1 Halaman Log In

Fitur ini telah berhasil mengiplementsikan desain yang telah dibuat sebelumnya. Hal ini dikarenakan sistem mampu membuat pengguna masuk ke halaman beranda jika *username* dan password yang dimasukkan benar sesuai Gambar 5.2. Apabila salah maka akan muncul tulisan *username* dan *password* salah.

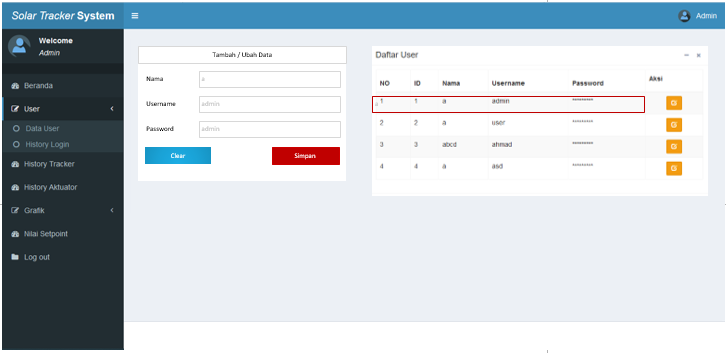
### Fitur Tambah User

Fitur ini merupakan fitur yang digunakan untuk menambah pengguna yang bisa mengakses sistem. Fitur untuk menambah data pengguna hanya dapat dilakukan oleh admin. Admin harus masuk pada menu tambah pengguna terlebih dahulu. Setelah itu mengisikan dan *password.* Data pengguna berhasil ditambahkan jika inputan sudah benar. Fitur ini sesuai pada Gambar 5.2



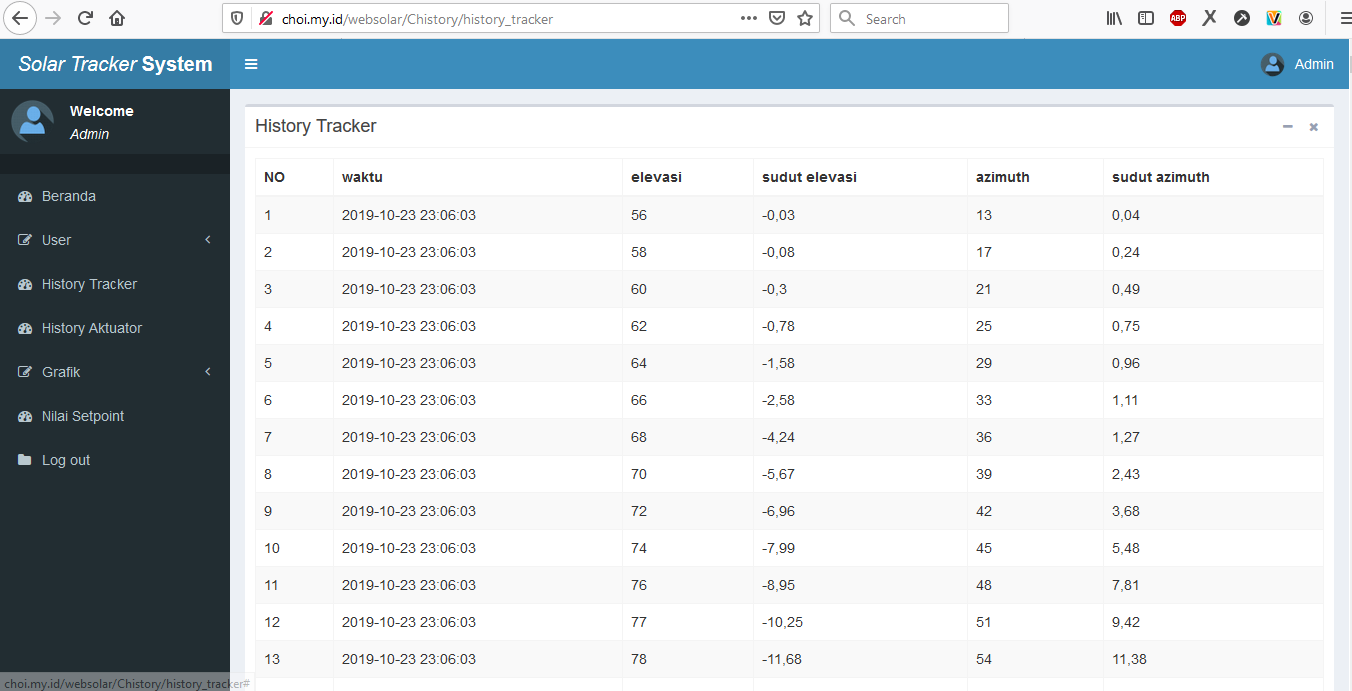
### Fitur Edit User

Fitur ini merupakan fitur yang digunakan untuk mengubah data pengguna yang bisa mengakses sistem. Fitur untuk mengubah data pengguna hanya dapat dilakukan oleh admin. Admin harus masuk pada menu tambah pengguna terlebih dahulu. Setelah itu mengisikan data yang akn diuabh pada *form* yang telah disediakan*.* Data pengguna berhasil ditambahkan jika inputan sudah benar. Fitur ini sesuai pada Gambar 5.3.



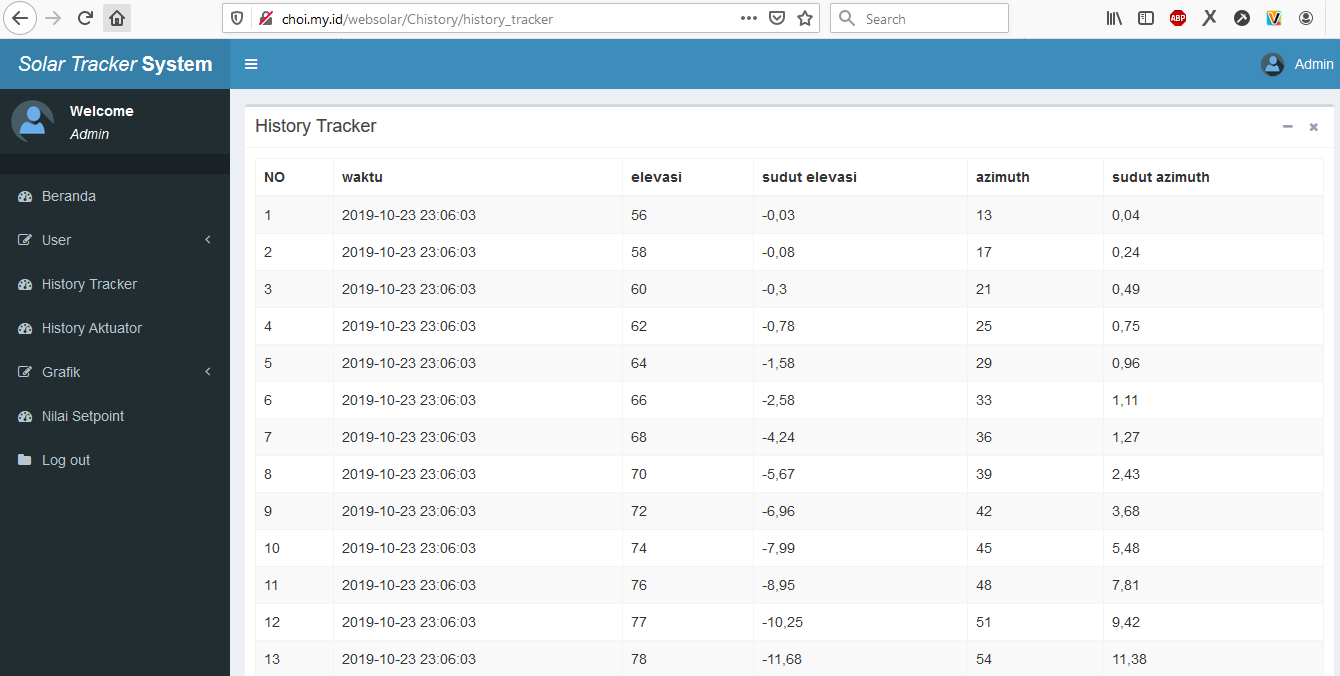
### Fitur *History Log in*

Fitur ini merupakan fitur yang digunakan untuk menampilkan data *history log in*. Fitur untuk melihat data history log in bisa dilakukan oleh admin. Admin harus masuk kedalam sistem terlebih dahulu. Setelah itu klik menu *History* dan pilih menu *History Log in*, maka akan menampilkan tabel siapa saja yang pernah mengakses sistem pada waktu tertentu. Fitur ini dapat dilihat pada Gambar 5.4.



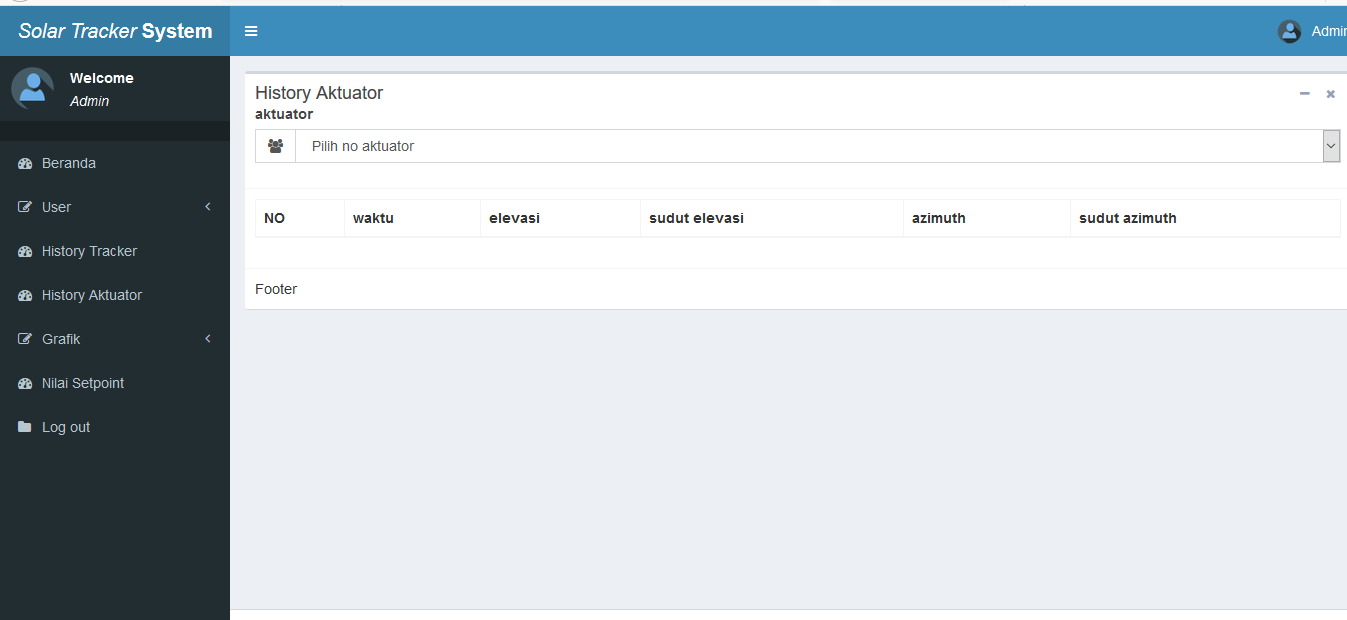
### Fitur *History Tracker*

Fitur ini merupakan fitur yang digunakan untuk menampilkan data *history log in*. Fitur untuk melihat data *history tracker*  bisa dilakukan oleh semua pengguna. Pengguna harus masuk ke dalam sistem terlebih dahulu. Setelah itu klik menu *History Tracker*, maka akan menampilakn tabel sudut yang pernah dilalui *tracker* pada waktu tertentu. Fitur ini dapat dilihat pada Gambar 5.5.



### Fitur *History* Aktuator

Fitur ini merupakan fitur yang digunakan untuk menampilkan data *history log in*. Fitur untuk melihat data *history* aktuator bisa dilakukan oleh semua pengguna. Pengguna harus masuk kedalam sistem terlebih dahulu. Setelah itu klik menu *history* aktuator. Jika aktuator yang dimiliki lebih dari satu maka pengguna harus memilih aktuator mana yang ingin ditampilkan. Hal ini bisa dilakukan denga memilih pada *dropdown* aktuator yang telah disediakan. Maka akan menampilakn tabel sudut yang pernah dilalui aktuator pada waktu tertentu. Fitur ini dapat dilihat pada Gambar 5.6.

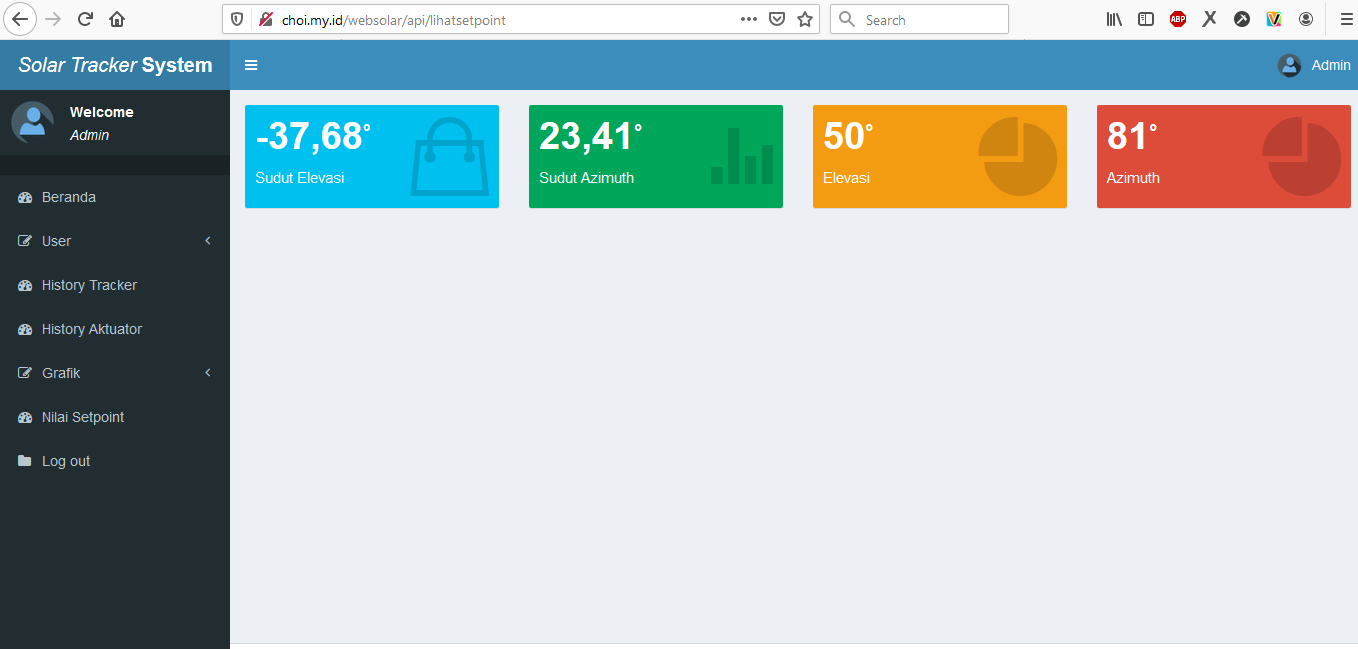


### Fitur Grafik Sensor

Fitur ini merupakan fitur yang digunakan untuk menampilkan data grafik sensor Fitur untuk melihat data grafik sensor bisa dilakukan oleh semua pengguna. Pengguna harus masuk kedalam sistem terlebih dahulu. Setelah itu klik menu grafik pilih grafik sensor, maka akan menampilakn grafik sensor secara *realtime*. Fitur ini dapat dilihat pada Gambar 5.7

### Fitur Nilai *Setpoint*

Fitur ini merupakan fitur yang digunakan untuk menampilkan data nilai *setpoint*. Fitur untuk melihat data nilai *setpoint* bisa dilakukan oleh semua pengguna. Pengguna harus masuk kedalam sistem terlebih dahulu. Setelah itu klik menu nilai *setpoint*, maka akan menampilakan nilai setpoint yang digunakan saat itu. Fitur ini dapat dilihat pada Gambar 5.8.



### Fitur Grafik *Tracker*

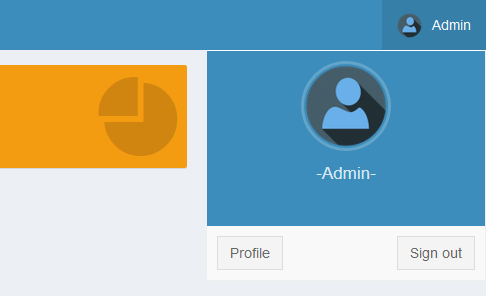
Fitur ini merupakan fitur yang digunakan untuk menampilkan data grafik sensor Fitur untuk melihat data grafik *tracker* bisa dilakukan oleh semua pengguna. Pengguna harus masuk kedalam sistem terlebih dahulu. Setelah itu klik menu grafik pilih grafik *tracker*, maka akan menampilakn grafik *tracker* secara *realtime*. Fitur ini dapat dilihat pada Gambar 5.9.

### Fitur Grafik Aktuator

Fitur ini merupakan fitur yang digunakan untuk menampilkan data grafik aktuator. Fitur untuk melihat data grafik aktuator bisa dilakukan oleh semua pengguna. Pengguna harus masuk kedalam sistem terlebih dahulu. Setelah itu klik menu grafik pilih grafik aktuator, maka akan menampilakn grafik aktuator secara *realtime*. Fitur ini dapat dilihat pada Gambar 5.10.

### Fitur *Log Out*

Fitur ini berfungsi untuk membuat pengguna keluar dari sistem dengan mengahapus *session* *log in* yang dimilikinya. Fitur ini bisa dilakukan oleh semua *user*. *Use*r yang ingin keluar dari sistem hanya perlu memilih menu log out maka sistem akan langsung kembali menampilkan halaman log in. Fitur ini dapat dilihat pada Gambar 5.11



## Hasil Implementasi Solar Tracker dengan Metode *Fuzzy*

## Hasil Implementasi Aktuator dengan Metode PID

## Hasil Implementasi Web *Solar Tracker System*

# KESIMPULAN























































































## Kesimpulan

Berdasarkan penelitian yang telah dilakukan , maka dapat ditarik kesimpulan sebagai berikut :

1. Penempatan sudut yang tepat berpengaruh terhadap produktivitas panel surya,

maka panel surya perlu ditempatkan sesuai dengan arah perpindahan matahari yang digunakan sebagai *setpoint*. Tetapi, pada penelitian ini matahri digantikan dengan senter untuk pengujian nya. Pengujian ini dilakukan dengan menempatkan senter diatas *solar tracker* pada posisi sudut 80 derajat. Pengambilan sudut dibantu dengan menempatkan 4 sensor LDR pada *solar tracker* agar bisa menangkap cahaya dengan lebih optimal pada posisi atas dan bawah maupun kanan dan kiri.

1. Metode *Fuzzy* diterapkan pada solar tracker untuk menentukan sudut optimal atau *setpoint.* Solar tracker menentukan setpoint setiap 10 menit sekali.  *Metode ini memerlukan* hasil perhitungan yang didapat dari sensor LDR. Perhitungan yang diperlukan yaitu nilai error vertical dan error horizontal . Error Vertikal yang didapat dari selisih rata atas dan rata bawah dari output sensor LDR. Sedagkan Error Horizontal didapat dari selisih rata kanan dengan rata kiri dari output sensor LDR. Setelah mendaptkan kedua nilai tersebut langkah yang harus dilakukan adalah fuzzifikasi. Fuzzifikasi merupkan proses yang berfungsi mengubah variabel numerik menjadi variabel linguistik (variabel *Fuzyy*). Variable numerik pada penelitian ini didapat dari nilai error vertical dan error horizontal. Selanjutnya dibutuhkan nilai delta error yaitu selisih nilai error saat ini dengan error sebelumnya.Nilai error dan delta error yang dikuantisasi sebelumnya diolah oleh kontroler logika *Fuzyy*, kemudian diubah terlebih dahulu ke dalam variabel *Fuzyy*. Melalui fungsi keanggotaan yang telah disusun, maka dari nilai error dan delta error kuantisasi akan didapatkan derajat keanggotaan bagi masing-masing nilai error dan delta error. Setelah tahap fuzzifikasi selesai, langkah selanjutnya yaitu *control rule base*. Tahap ini merupakan aturan yang diterapkan sebagai output akhir dari fuzzy. Pada penelitian ini terdapat 49 aturan yang diperoleh dari pnggabungan *variable linguistic* dari error dengan *delta error*. Langkah terakhir adalah deffuzifikasi. Tahapan ini merupakan kebalikan dari fuzzifikasi, variable linguistic yang didapat dari tahap sebelumnya harus diubah menjadi variable numerik agar bisa diolah oleh sistem untuk menjalankan solar tracker ke arah *setpoint* sesuai dengan besarnya jarak antara setpoint dengan sudut sebenarnya. Perpindahan tersebut terbagi menjadi 4 yaitu nol,lima,delapan dan 10 tahap. Setiap tahapan bernilai satu sudut.

Sedangkan metode PID diterapkan pada aktuator untuk menstabilkan aktuator agar posisinya tidak berubah dari *setpoint* apabila terjadi gangguan dari faktor luar seperti angin.

1. *Solar tracker* yang menggunakan metode *Fuzzy* lebih efisien waktu untuk berpindah dari sudut awal ke arah *setpoint.* Hal ini dikarenakan pada tiap proses nya *Fuzzy* mampu memberikan perpindahan sudut sampai sepuluh derajat sekaligus bahkan lebih sesuai aturan yang diterapkan pada *control rule base.* Sedangkan *solar tracker* jika tanpa menggunakan metode *Fuzzy* hanya mampu berpindah satu derajat tiap tahapannya. Selain itu *setpoint* hanya ditentukan melalui output sensor LDR yang harus bernilai nol. Hal ini mengakibatkan pergerakan *solar tracker* menjadi kurang efisien karena terus bergerak mencari nilai nol.
2. User Berinteraksi dengan sistem dapat dilakukan melalui web. Web ini berfungsi untuk menampilkan hasil dari penerapan metode *Fuzzy* pada *solar tracker* dan PID pada aktuator. Web ini juga sebagai media pengirim data dari tracker ke aktuator. Pengiriman tersebut tanpa ada campur tangan pengguna sesuai dengan kosep IOT karena pengguna hanya bertugas memonitoring sistem. Akses web dapat dilakukan dengan masuk ke halaman *log in* untuk mengisi username dan password yang dimiliki. Web sistem ini memiliki dua jenis pengguna yaitu user dan admin. Perbedaan dari dua jenis pengguna ini terdapat pada akses fitur data user dan *history log in*  yang hanya dapat diakses oleh admin sistem.

## Saran

# Daftar Pustaka

Ahmad, K. (2011). Pembangkit Listrik tenaga Surya dan Penerapan Untuk Daerah Terpencil. *Pusat Pengkajian dan Penerapan Teknologi Konversi dan Konservasi Energi, BPP-Teknologi*, 2.

Aisuwarya, R., & Annafi, R. (2017). Pengendali Fuzzy Logic Controller untuk Pengendalian Kecepatan Roda Pada Mobile Robot Pada Variasi nilai SetPoint. *Pengendali Fuzzy Logic Controller untuk Pengendalian Kecepatan Roda Pada Mobile Robot Pada Variasi nilai SetPoint*, 1.

Anggara, Kumara, I. N., & Giriantari, I. (2014). Studi Terhadap Unjuk kerja Pembangkit Listrik Tenaga Surya 1,9 KW dI UNIVERSITAS UDAYANA BUKIT JIMBARAN. *ResearchGate*, 2-3.

Febrianto, D. A. (2015). *Perancangan dan Pembuatan Sistem Informasi Simpan Pinjam Pada Koperasi Pegawai Republik Indonesia (KP-RI) "DIAN PATIRANA" Berbasis Web.* Jember: Universitas Jember.

Irkhos, & Suprijadi. (n.d.). Simulasi Kontrol Temperatur Berbasis Fuzzy Logic untuk Tabung Sampel Minyak Bumi pada Metode Direct Subsurface Sampling.

Madyanto, T. D. (2010). Pengontrolan Suhu Menggunakan Fuzzy PID Pada Model Sistem Hipertemia. *ResearchGate*, 2.

Mustafa, F. M., & Ahmed, S. A. (2017). Design and Implementation of Dual-Axis Solar Tracking System. *IARJSET*, 2.

Pahlevi, E. L., & Yana, S. (2016). Pengaturan Pitch Angle Turbin Angin Berbasis Kendali Logika Fuzzy (Aplikasi Pada Data Angin Daerah Medan Tuntungan dan sekitarnya. *Singuda Ensikom*, 1.

Pangestuningtyas. (2013). Analisis Pengaruh Sudut Kemiringan Panel Surya Terhadap Radiasi Matahari Yang Diterima oleh Panel Surya Tipe Array Tetap. *Transient*, 3.

Prahara, T. (2018). Sistem Kontrol Cerdas Pelacak Sumber Cahaya Menggunakan Kontrol Proportional Integral Derivative (PID). *Journal of Information Education*, 1.

Rahardjo, I., & Fitriana, I. (2014). Analisis Potensi Pembangkit Listrik Tenaga Surya di Indonesia . *Analisis Potensi Pembangkit Listrik Tenaga Surya di Indonesia* , 4.

Saputra, M. A. (2014). Inovasi Peningkatan Efisiensi Panel Surya Berbasis Fresnel Solar Concentrator dan Solar Tracker. *Journal Electro*, 2.

Sulaiman, O. K. (2017). Sistem Internet Of Things (IoT) Berbasis Cload Computing Dalam Campus Area Network. *Information System*, 2.

Suoth, V. A., Mosey, H. I., & Telleng, R. C. (2018). *Rancang bangun alat pendeteksi intensitas cahaya berbasis Sensor Light Dependent Resistance (LDR).* Manado: Universitas Sam Ratulangi.

Wahab, F., Sumardiono, A., & Tahtawi, A. A. (2017). Desain dan Purwarupa Fuzzy Logic Controluntuk Pengendalian Suhu Ruangan. *Jurnal Teknologi Rekayasa*, 1-8.

ZA, N., & Maulinda, L. (2015). Perbandingan Kontrol Fuzzy dan PID pada Pemanas Fuel Gas. *Jurnal Teknologi Kimia Unimal*, 1-18.

# LAMPIRAN

## Skenario

|  |  |  |
| --- | --- | --- |
| **Nama Usecase** | Log In | |
| **Aktor** | Seluruh aktor | |
| **Deskripsi Singkat** | Aktor memasukkan username dan password | |
| **Prekondisi** | Aktor masuk halaman utama Login | |
| **Pascakondisi** | Aktor masuk halaman utama sesuai aktor | |
| **Flow Event** | | |
| **Normal Flow : Log In** | | |
| Aksi Aktor | | Reaksi Sistem |
| 1. Masuk ke halaman login | |  |
|  | | 1. Sistem menampilkan halaman login yang berisi form, sebagai berikut :   a. Username(varchar 20)  b. Password (varchar 20 ) |
| 1. Aktor mengisi username dan password | |  |
| 1. Klik ‘Login’ | |  |
|  | | 1. Sistem mengecek inputan dan mencocokkan dengan data yang ada di database |
|  | | 1. Sistem menampilkan dashboard sesuai level user |
| **Flow Event** | | |
| Alternatif Flow : Nama Pengguna atau Password salah | | |
| 4. Klik ‘Login’ | |  |
|  | | 1. Menampilkan pop-up “Username atau password salah” |
| 1. Klik ‘oke’ | |  |
|  | | 7. Sistem menampilkan halaman login yang berisi form, sebagai berikut :  a. Username (varchar 20)  b. Password (varchar 20 ) |

|  |  |  |
| --- | --- | --- |
| **Nama Usecase** | Tambah *user* | |
| **Aktor** | Admin | |
| **Deskripsi Singkat** | Aktor menambahkan user baru | |
| **Prekondisi** | Aktor masuk halaman dashboard admin | |
| **Pascakondisi** | Data user bertambah | |
| **Flow Event** | | |
| **Normal Flow : Tambah user** | | |
| Aksi Aktor | | Reaksi Sistem |
| * + 1. Klik menu user pilih tambah user | |  |
|  | | * + 1. Menampilkan form tambah user  1. Nama (varchar 20) 2. Username (varchar 20) 3. Password (varchar 20) |
| * + 1. Aktor mengisi nama, username, dan password | |  |
| * + 1. Klik ‘Simpan’ | |  |
|  | | * + 1. Sistem mengecek inputan |
|  | | * + 1. Sistem menyimpan data ke database |
| **Flow Event** | | |
| Alternatif Flow : Inputan salah | | |
| 1. Klik Simpan | |  |
|  | | 1. Menampilkan pop-up “Inputan salah” |
| 1. Klik ‘oke’ | |  |
|  | | 1. Sistem menampilkan halaman form tambah data user 2. Nama (varchar 20) 3. Username (varchar 20) 4. Password (varchar 20) |

|  |  |  |
| --- | --- | --- |
| **Nama Usecase** | Edit User | |
| **Aktor** | Admin | |
| **Deskripsi Singkat** | Aktor mengubah data user | |
| **Prekondisi** | Aktor masuk halaman dashboard admin | |
| **Pascakondisi** | Data user berhasil diubah | |
| **Flow Event** | | |
| **Normal Flow : Edit user** | | |
| Aksi Aktor | | Reaksi Sistem |
| 1. Klik tombol edit pada user yang akan diubah data nya | |  |
|  | | 1. Menampilkan form edit user |
| 1. Aktor mengisi nama, username, atau password yang akan di ganti | |  |
| 1. Klik ‘Update’ | |  |
|  | | 1. Sistem mengecek inputan |
|  | | 1. Sistem menyimpan data ke database |
| **Flow Event** | | |
| Alternatif Flow : Inputan salah | | |
| 1. Klik Update | |  |
|  | | 1. Menampilkan pop-up “Inputan Salah” |
| 1. Klik ‘oke’ | |  |
|  | | 1. Sistem menampilkan halaman form edit data user 2. Nama (varchar 20) 3. Username (varchar 20) 4. Password (varchar 20) |

|  |  |  |
| --- | --- | --- |
| **Nama Usecase** | History login | |
| **Aktor** | Admin | |
| **Deskripsi Singkat** | Aktor melihat history login | |
| **Prekondisi** | Aktor masuk halaman dashboard admin | |
| **Pascakondisi** | History login ditampilkan | |
| **Flow Event** | | |
| **Normal Flow : History login** | | |
| Aksi Aktor | | Reaksi Sistem |
| 1. Klik menu user pilih History login | |  |
|  | | 1. Menampilkan data history login |

|  |  |  |
| --- | --- | --- |
| **Nama Usecase** | Lihat data history tracker | |
| **Aktor** | Senua aktor | |
| **Deskripsi Singkat** | Aktor melihat **data *history tracker*** | |
| **Prekondisi** | Aktor masuk halaman dashboard masing-masing | |
| **Pascakondisi** | Aktor dapat melihat **data *history tracker*** | |
| **Flow Event** | | |
| **Normal Flow : data *history*** *tracker* | | |
| Aksi Aktor | | Reaksi Sistem |
| 1. Klik ***history tracker*** | |  |
|  | | 1. Menampilkan **data *history tracker*** |

|  |  |  |
| --- | --- | --- |
| **Nama Usecase** | **Lihat data *history*** aktuator | |
| **Aktor** | Senua aktor | |
| **Deskripsi Singkat** | Aktor melihat **data *history*** aktuator | |
| **Prekondisi** | Aktor masuk halaman dashboard masing-masing | |
| **Pascakondisi** | Aktor dapat melihat **data *history*** aktuator | |
| **Flow Event** | | |
| **Normal Flow : Lihat data *history*** aktuator | | |
| Aksi Aktor | | Reaksi Sistem |
| 1. Klik ***history* aktuator** | |  |
|  | | 1. Menampilkan **data *history* aktuator** |

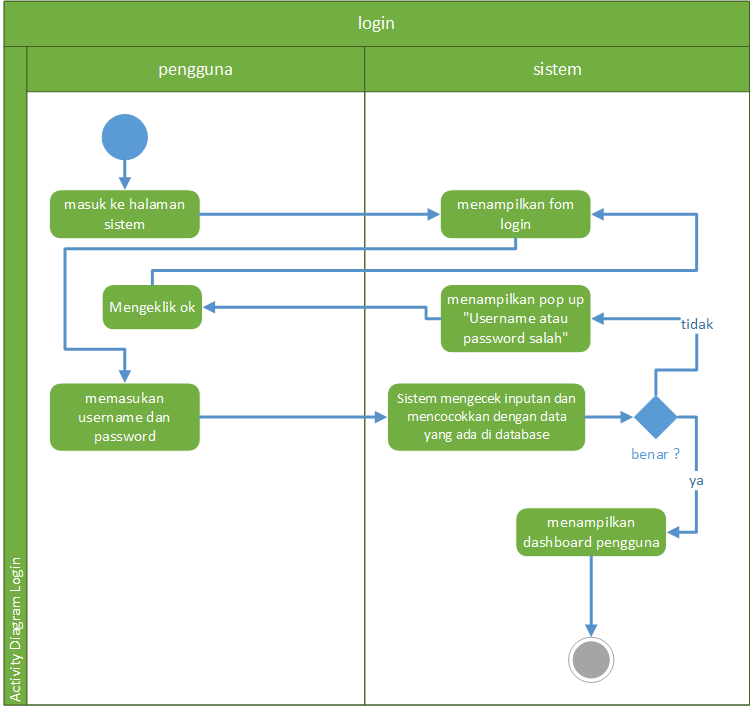
|  |  |  |
| --- | --- | --- |
| **Nama Usecase** | Lihat *Lihat Grafik Sensor* | |
| **Aktor** | Senua aktor | |
| **Deskripsi Singkat** | Aktor melihat Lihat Grafik Sensor | |
| **Prekondisi** | Aktor masuk halaman dashboard masing-masing | |
| **Pascakondisi** | Aktor dapat melihat Lihat Grafik Sensor | |
| **Flow Event** | | |
| **Normal Flow : Lihat Lihat Grafik Sensor** | | |
| Aksi Aktor | | Reaksi Sistem |
| 1. Klik menu Grafik pilih menu grafik sensor | |  |
|  | | 1. Menampilkan Lihat Grafik Sensor |

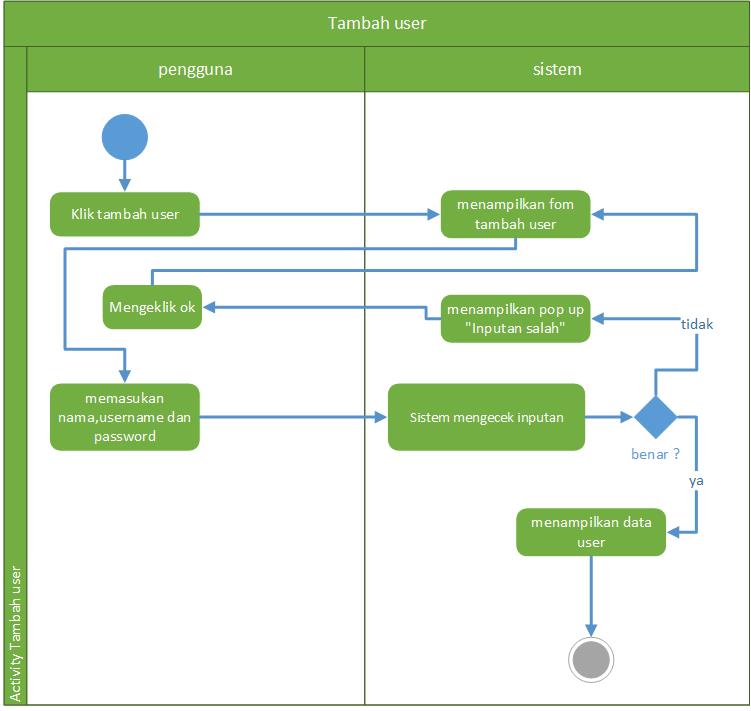
|  |  |  |
| --- | --- | --- |
| **Nama Usecase** | Lihat Data Nilai *Setpoint* | |
| **Aktor** | Senua aktor | |
| **Deskripsi Singkat** | Aktor melihat nilai *setpoint* | |
| **Prekondisi** | Aktor masuk halaman dashboard masing-masing | |
| **Pascakondisi** | Aktor dapat melihat data nilai *setpoint* | |
| **Flow Event** | | |
| **Normal Flow : Lihat history sudut y** | | |
| Aksi Aktor | | Reaksi Sistem |
| 1. Klik menu nilai *setpoint* | |  |
|  | | 1. Menampilkan data nilai *setpoint* |

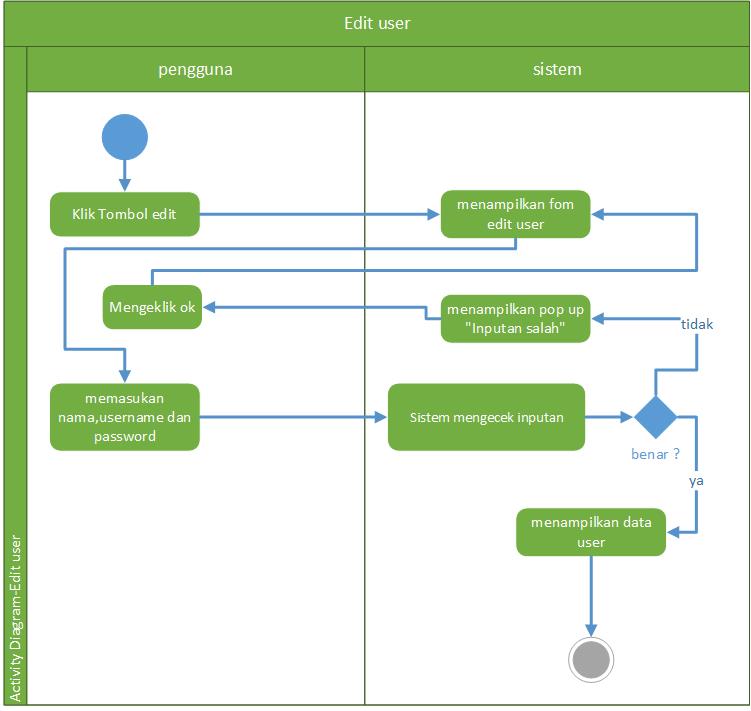
|  |  |  |
| --- | --- | --- |
| **Nama Usecase** | Lihat grafik *tracker* | |
| **Aktor** | Senua aktor | |
| **Deskripsi Singkat** | Aktor melihat grafik *tracker* | |
| **Prekondisi** | Aktor masuk halaman dashboard masing-masing | |
| **Pascakondisi** | Aktor dapat melihat grafik *tracker* | |
| **Flow Event** | | |
| **Normal Flow : Lihat grafik sudut y** | | |
| Aksi Aktor | | Reaksi Sistem |
| 1. Klik menu grafik pilih menu grafik *tracker* | |  |
|  | | 1. Menampilkan grafik *tracker* |

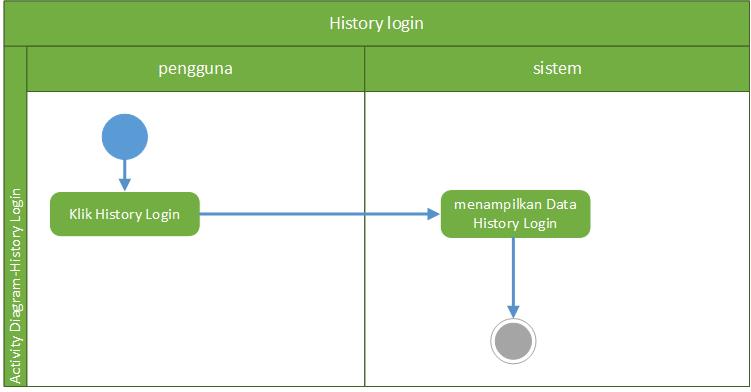
|  |  |  |
| --- | --- | --- |
| **Nama Usecase** | Lihat grafik aktuator | |
| **Aktor** | Senua aktor | |
| **Deskripsi Singkat** | Aktor melihat grafik aktuator | |
| **Prekondisi** | Aktor masuk halaman dashboard masing-masing | |
| **Pascakondisi** | Aktor dapat melihat grafik aktuator | |
| **Flow Event** | | |
| **Normal Flow : Lihat grafik sudut y** | | |
| Aksi Aktor | | Reaksi Sistem |
| 1. Klik menu grafik pilih menu grafik aktuator | |  |
|  | | 1. Menampilkan grafik aktuator |

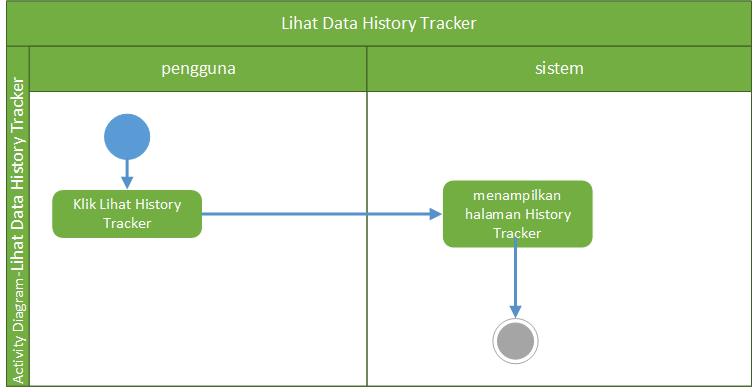
|  |  |  |
| --- | --- | --- |
| **Nama Usecase** | *Log out* | |
| **Aktor** | Senua aktor | |
| **Deskripsi Singkat** | Aktor keluar dari sistem | |
| **Prekondisi** | Aktor masuk halaman dashboard masing-masing | |
| **Pascakondisi** | Aktor keluar dari sistem | |
| **Flow Event** | | |
| **Normal Flow : Log out** | | |
| Aksi Aktor | | Reaksi Sistem |
| 1. Klik menu logout | |  |
|  | | 1. Keluar sistem, Menampilkan halaman *log in* |

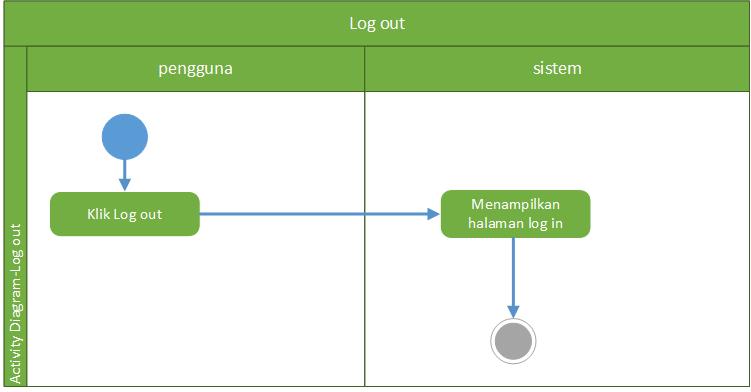
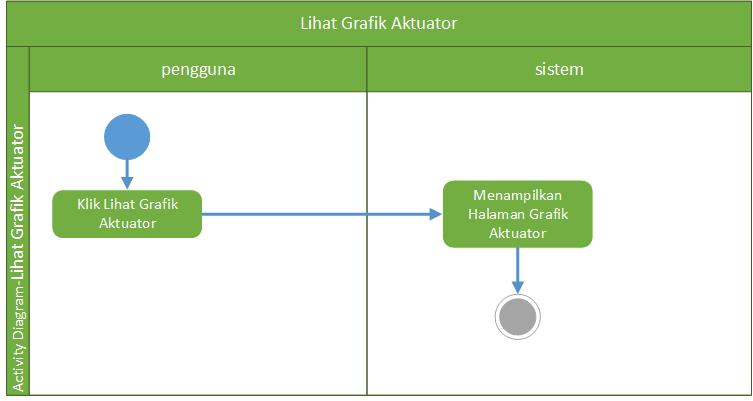
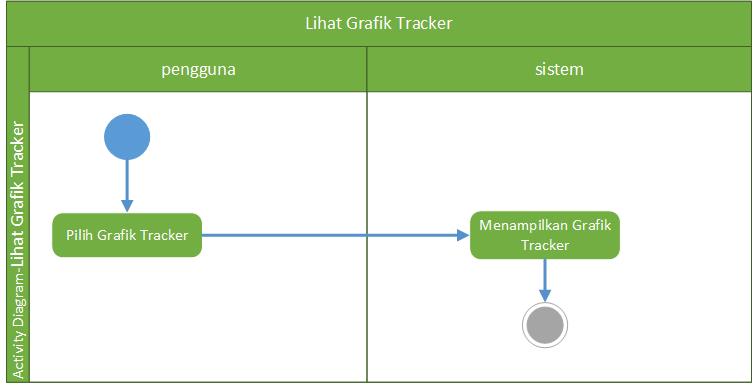
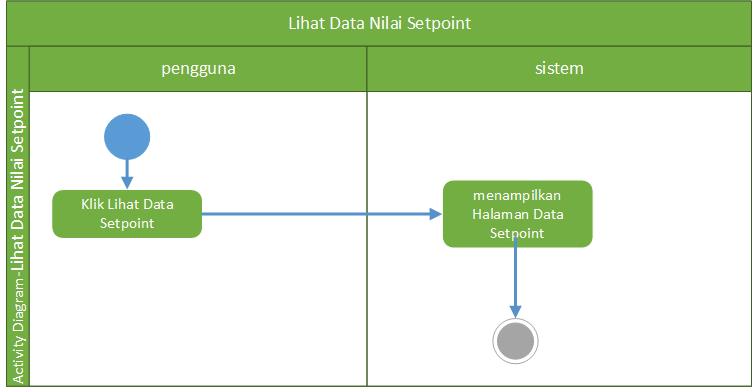
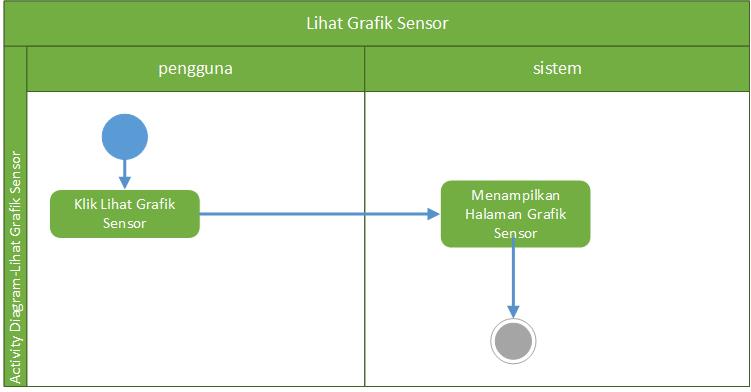
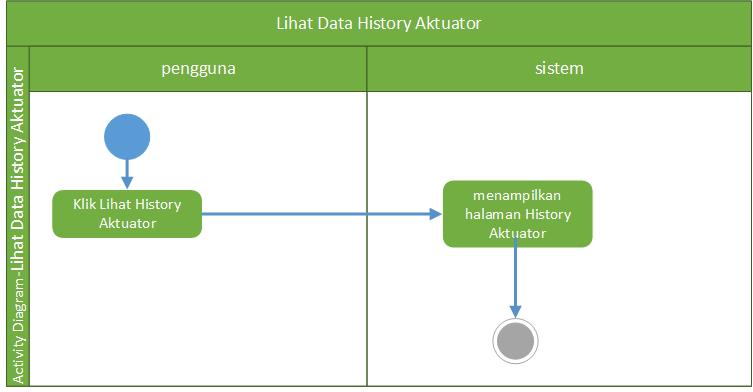




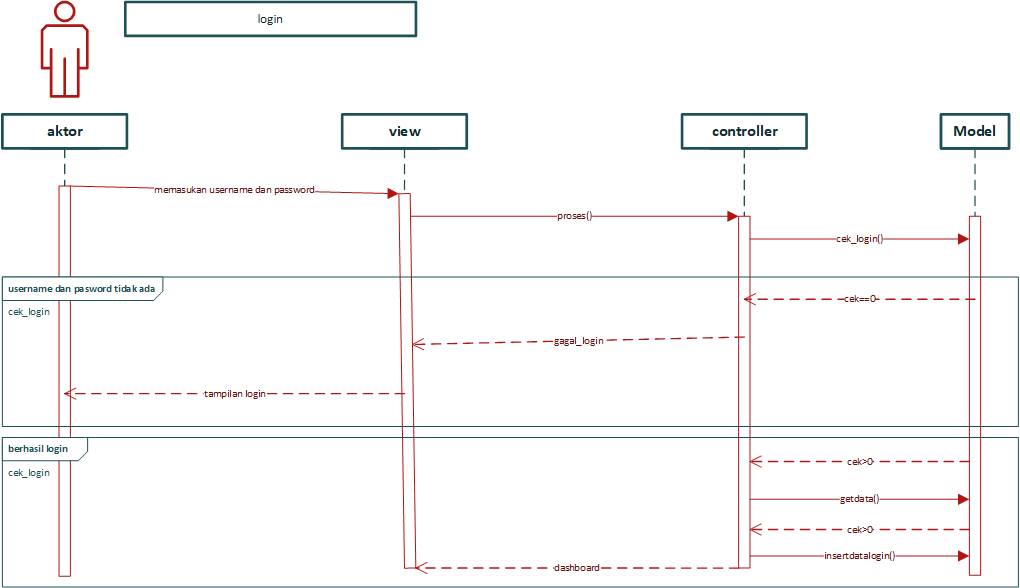


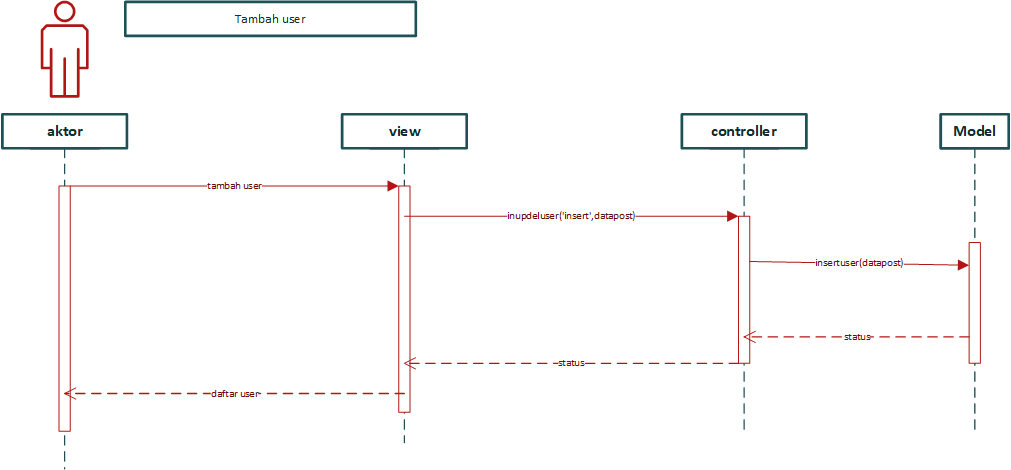


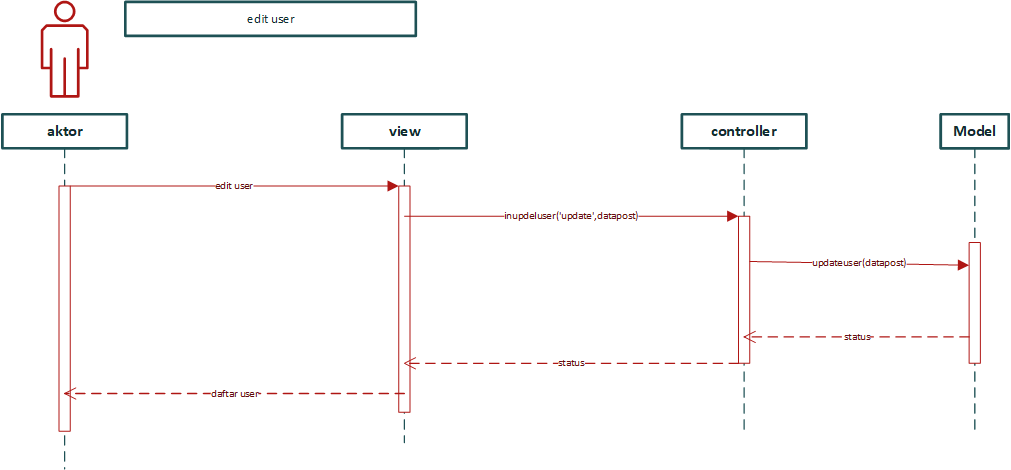


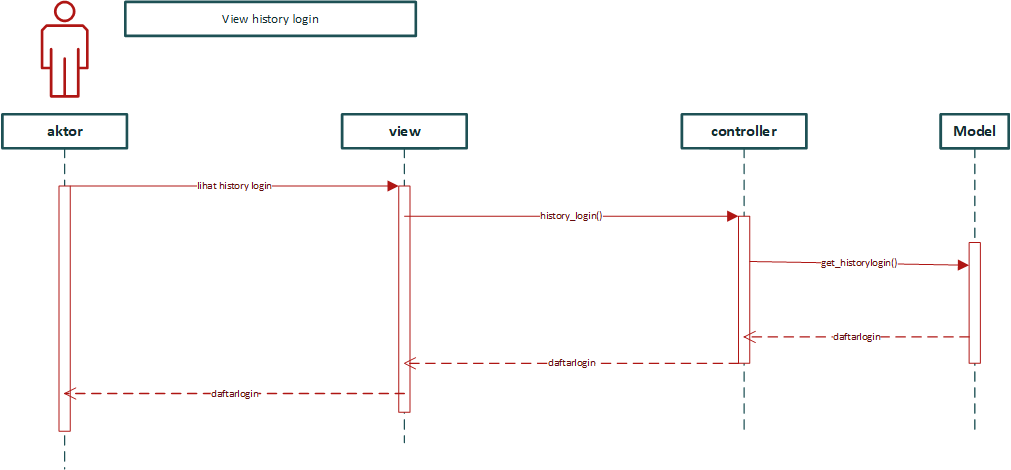


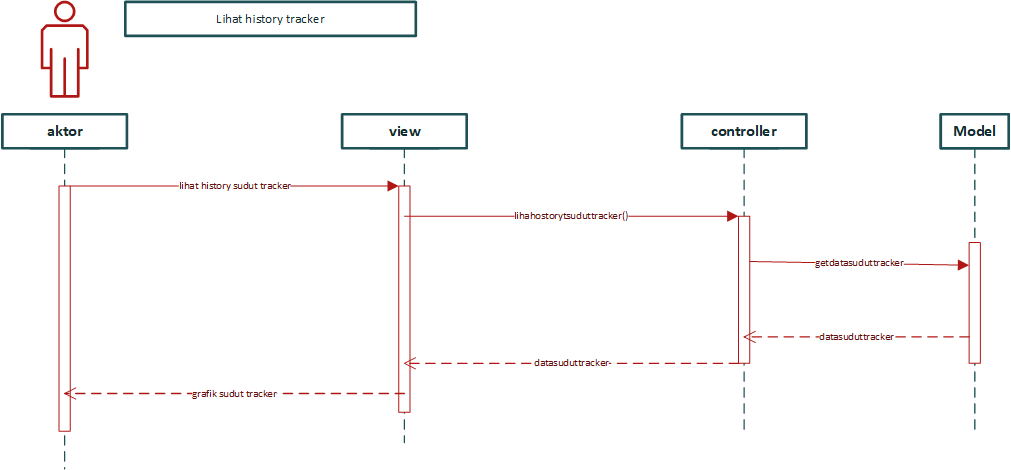
## *Sequence Diagram*

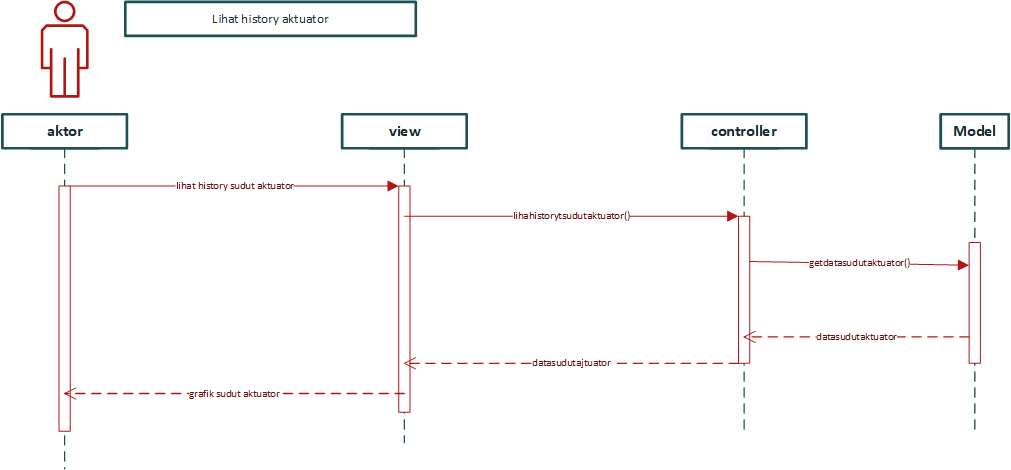


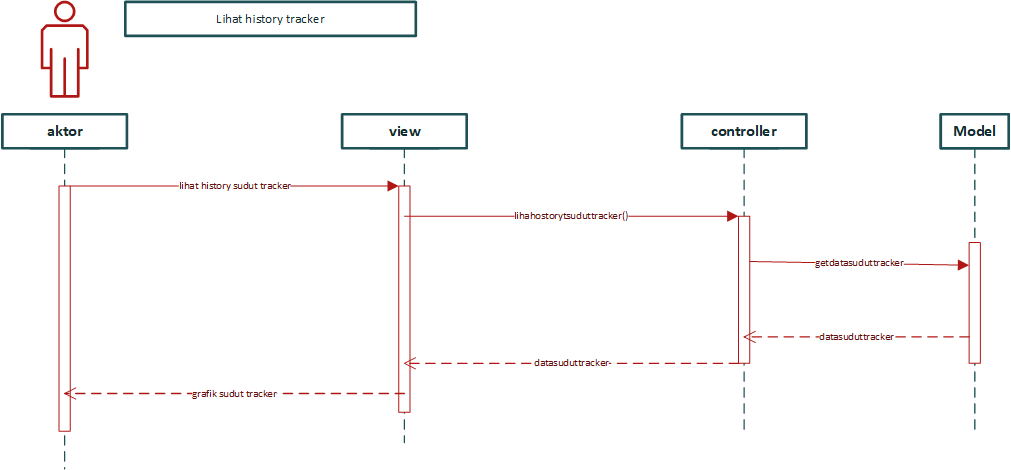


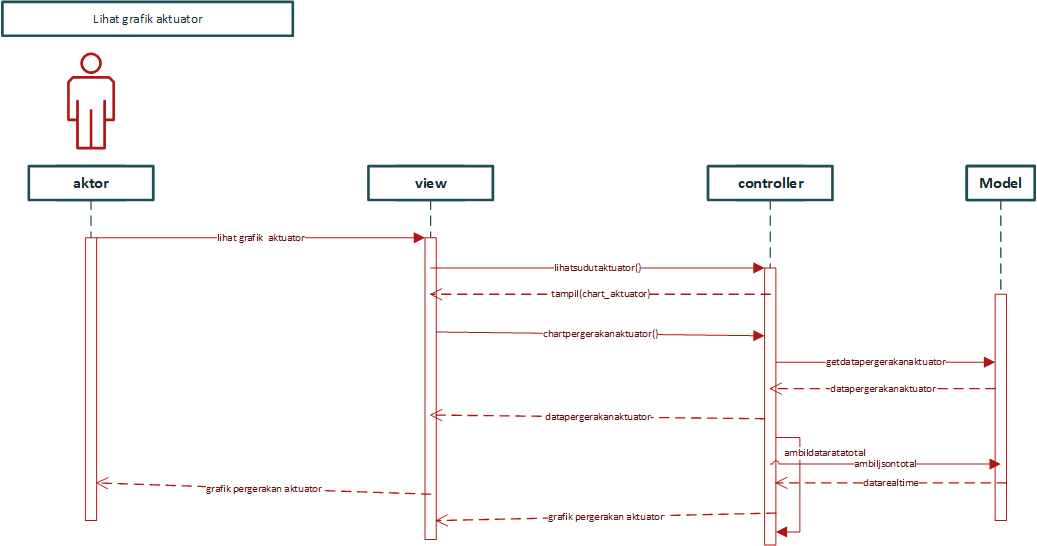


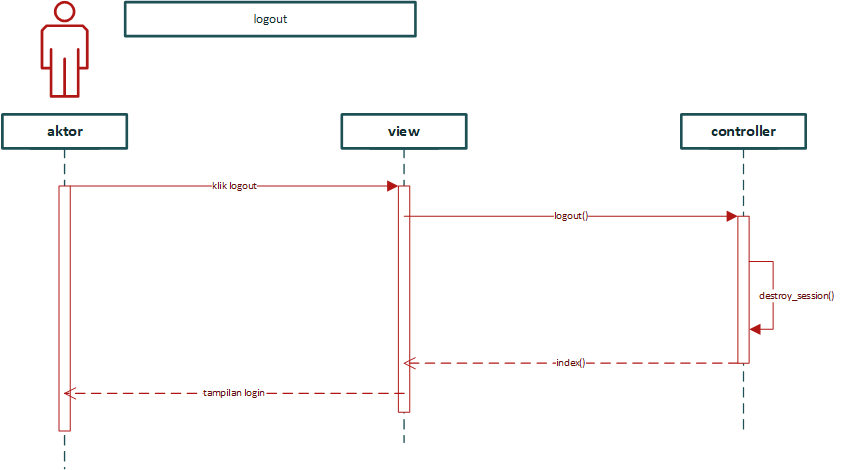












1. ***Desain User Interface***

