

PERANCANGAN DASHBOARD DAN QUERY UNTUK
MONITORING DATA SENSOR PADA MINIATUR STASIUN
CUACA MENGGUNAKAN INFLUXDB DAN GRAFANA



SKRIPSI

Diajukan sebagai salah satu syarat untuk menyelesaikan
pendidikan Diploma Empat (D-4) Program Studi Teknik Komputer dan Jaringan
Jurusan Teknik Elektro
Politeknik Negeri Ujung Pandang

TITIN NURFADHILA SUDIRMAN
425 15 030

PROGRAM STUDI D-4 TEKNIK KOMPUTER DAN JARINGAN
JURUSAN TEKNIK ELEKTRO
POLITEKNIK NEGERI UJUNG PANDANG
MAKASSAR
2019

HALAMAN PENGESAHAN

Skripsi dengan judul “Perancangan Dashboard dan Query untuk Monitoring Data Sensor pada Miniatur Stasiun Cuaca menggunakan InfluxDB dan Grafana” oleh Titin Nurfadhila Sudirman 425 15 030 telah diterima dan disahkan sebagai salah satu syarat untuk memperoleh gelar Diploma IV (D-4/SI Terapan) pada Program Studi Teknik Komputer dan Jaringan Jurusan Teknik Elektro Politeknik Negeri Ujung Pandang.

Makassar, 2019

Mengesahkan,

Pembimbing I



Drs. Kasim, M.T.
NIP.19630620 199103 1 002

Pembimbing II



Dr. Ir. Hafsah Nirwana, M.T.
NIP.19640405 199003 2 002

Mengetahui,

Ketua Program Studi
Teknik Komputer dan Jaringan
Politeknik Negeri Ujung Pandang






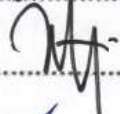


Rini Nur, S.T., M.T.
NIP.19730713 200912 2 001

HALAMAN PENERIMAAN

Pada hari ini, Jumat tanggal 02 Agustus 2019, Tim Penguji Sidang Tugas Akhir, telah menerima dengan baik hasil skripsi oleh mahasiswa: **Titin Nurfadhila Sudirman (425 15 030)** dengan judul **“Perancangan Dashboard Dan Query Untuk Monitoring Data Sensor Pada Miniatur Stasiun Cuaca Menggunakan InfluxDB Dan Grafana”**.

Makassar, 2019

Tim Penguji Ujian Skripsi:

- | | | |
|--|------------|--|
| 1. Eddy Tungadi, ST., MT. | Ketua | (..... ) |
| 2. Sahbuddin Abdul Kadir, ST., MT. | Sekretaris | (..... ) |
| 3. Iin Karmila Yusri, S.ST., M.Eng.,PhD. | Anggota | (..... ) |
| 4. Kartika Dewi, ST., MT. | Anggota | (..... ) |
| 5. Drs. Kasim, MT. | Anggota | (..... ) |
| 6. Dr. Ir. Hafsah Nirwana, MT. | Anggota | (..... ) |

KATA PENGANTAR

Puji syukur senantiasa penulis panjatkan ke hadirat Allah SWT, karena berkat rahmat dan karunia-Nya yang telah memberikan kesehatan dan keselamatan kepada penulis sehingga penulisan skripsi yang berjudul “Perancangan Dashboard Dan Query Untuk Monitoring Data Sensor Pada Miniatur Stasiun Cuaca Menggunakan InfluxDB Dan Grafana” dapat terselesaikan. Shalawat dan salam kepada baginda rasul Muhammad SAW, sebagai sebaik-baik panutan bagi seluruh manusia.

Skripsi ini disusun untuk memenuhi salah satu persyaratan untuk menyelesaikan studi pada Program Studi D-IV Teknik Komputer dan Jaringan di Politeknik Negeri Ujung Pandang.

Dalam penyelesaian skripsi ini tidak sedikit hambatan yang penulis alami. Namun, berkat bantuan dari berbagai pihak sehingga penyusunan skripsi ini dapat terselesaikan. Oleh karena itu, dengan rendah hati penulis mengucapkan terima kasih kepada:

1. Orang tua penulis, ayah penulis Sudirman dan Ibunda tercinta Suriani yang senantiasa memberikan semangat, kesabaran mendidik, motivasi, dukungan, bimbingan dan doa restu kepada penulis serta seluruh keluarga besar.
2. Bapak Prof. Ir. Muhammad Anshar, M.Si, Ph.D. selaku Direktur Politeknik Negeri Ujung Pandang.
3. Ibu Dr. Ir. Hafsah Nirwana, M.T. selaku Ketua Jurusan Elektro Politeknik Negeri Ujung Pandang sekaligus selaku pembimbing II yang berkenan

menyempatkan waktunya untuk membantu, mengarahkan, dan memberikan masukan dalam membimbing penulis hingga selesainya penelitian ini.

4. Ibu Rini Nur, S.T., M.T. selaku Ketua Program Studi Teknik Komputer dan Jaringan.
5. Bapak Drs. Kasim, M.T. selaku pembimbing I atas segala ilmu, motivasi, nasehat, arahan, bantuan dan kesedian waktu dan kesabarannya dalam membimbing penulis hingga terselesaikannya penelitian ini.
6. Seluruh dosen dan Staf Jurusan Teknik Elektro, Khususnya Prodi Teknik Komputer dan Jaringan.
7. Teman-teman seperjuangan di Jurusan Teknik Elektro dan khususnya teman teman angkatan 2015 Teknik Komputer dan Jaringan.
8. Anggota team IoT khususnya team stasiun cuaca Hermansyah dan Samsinar yang selalu memberikan arahan dan bantuan dalam penyelesaian skripsi.
9. Sahabat – sahabatku Misna Syari, Ferra Andriani dan Nopilyn jaya yang selalu memberikan dukungan, semangat, serta doa.
10. Semua pihak yang telah memberikan bantuan moril maupun materil yang tidak dapat disebutkan satu per satu.

Penulis menyadari bahwa skripsi ini masih terdapat banyak kekurangan. Oleh karena itu, penulis memohon maaf dan mengharapkan kritik membangun serta saran terhadap laporan skripsi ini. Semoga skripsi ini bermanfaat bagi pembacanya.

Makassar, 02 Agustus 2019

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PENGESAHAN	ii
HALAMAN PENERIMAAN	iii
KATA PENGANTAR	iv
DAFTAR ISI.....	vi
DAFTAR GAMBAR.....	ix
DAFTAR TABEL.....	xi
DAFTAR LAMPIRAN.....	xii
SURAT PERNYATAAN	xiii
RINGKASAN	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Ruang Lingkup Penelitian.....	2
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
BAB II TINJAUAN PUSTAKA	4
2.1 Monitoring Data Stasiun Cuaca	4
2.1.1. <i>Monitoring</i>	4
2.1.2. <i>Miniatur Stasiun Cuaca</i>	4
2.2 <i>Time Series</i>	5
2.3 Penyimpanan Data <i>Time Series</i>	6
2.3.1 <i>Time Series Database (TSDB)</i>	6
2.3.2 <i>Pengertian InfluxDB</i>	7

2.3.3	Fitur – Fitur InfluxDB.....	8
2.4	Aplikasi Penerimaan Data.....	9
2.4.1	<i>Message Queue Telemetry Transport</i> (MQTT).....	9
2.4.1.	Python.....	11
2.5	<i>Dashboard</i>	11
2.6	Visualisasi Data.....	11
2.6.1.	Grafana.....	12
2.6.2.	Kelebihan Grafana.....	13
2.6.3.	Fitur – Fitur Grafana.....	14
BAB III METODOLOGI PENELITIAN.....		16
3.1	Waktu dan Tempat Penelitian.....	16
3.2	Alat dan Bahan.....	16
3.3	Metode Penelitian.....	17
3.3.1	Identifikasi Masalah.....	17
3.3.2	Analisis Kebutuhan Data.....	17
3.3.3	Perancangan Sistem.....	18
3.3.4	Instalasi Dan Konfigurasi.....	24
3.3.5	Perancangan <i>Dashboard</i> dan <i>Query</i>	26
3.3.6	Pengujian.....	28
3.3.7	Analisis dan Kesimpulan.....	29
BAB IV HASIL DAN PEMBAHASAN.....		30
4.1	Implementasi.....	30
4.1.1	Konfigurasi Penerimaan Data.....	30
4.1.2	Data Sensor pada Server InfluxDB.....	32
4.1.3	Proses Visualisasi Data.....	34
4.2	Pengujian.....	40
4.2.1.	Pengujian Penerimaan dan Verifikasi Data.....	40
4.2.2.	Pengujian <i>Query</i> Data.....	44
BAB V KESIMPULAN DAN SARAN.....		51

5.1	Kesimpulan.....	51
5.2	Saran.....	51
DAFTAR PUSTAKA.....		52
LAMPIRAN.....		54



DAFTAR GAMBAR

	Halaman
Gambar 2. 1 <i>DB-Engines Ranking of Time Series DBMS</i>	7
Gambar 2. 2 InfluxDB	8
Gambar 2. 3 Skema <i>Publish</i> Dan <i>Subscribe</i> Pada MQTT	10
Gambar 2. 4 <i>Dashboard</i> Grafana	13
Gambar 3. 1 Tahapan Penelitian	17
Gambar 3. 2 Diagram Blok Miniatur Stasiun Cuaca	19
Gambar 3. 3 Diagram Alir Sistem	22
Gambar 3. 4 Diagram Alir Visualisasi Data di Grafana	23
Gambar 3. 5 Diagram Alir Instalasi dan Konfigurasi	25
Gambar 3. 6 Desain <i>Layout Dashboard</i> Stasiun Cuaca	27
Gambar 3. 7 Desain <i>Layout dashboard</i> Panel Surya	28
Gambar 4. 1 Inisialisasi Tujuan Penerimaan Data di Server	31
Gambar 4. 2 Inisialisasi Client <i>Subscribe</i> Miniatur Stasiun Cuaca	31
Gambar 4. 3 Inisialisasi <i>Client Subscribe</i> Panel Surya	32
Gambar 4. 4 Data <i>Measurements</i> Pada InfluxDB	33
Gambar 4. 5 Data Sensor Kecepatan Angin pada InfluxDB	34
Gambar 4. 6 Menampilkan Nilai <i>Min, Max, Mean, Last</i>	34
Gambar 4. 7 <i>Plugin Panel</i>	35
Gambar 4. 8 <i>Query Data</i> pada Panel <i>Graph</i>	36
Gambar 4. 9 <i>Data Source</i>	37
Gambar 4. 10 Tampilan <i>Dashboard</i> Miniatur Stasiun Cuaca	38

Gambar 4. 11 Tampilan <i>Dashboard</i> Panel Surya	39
Gambar 4. 12 Proses Penerimaan Data Miniatur Stasiun Cuaca	40
Gambar 4. 13 Proses Penerimaan Data Panel Surya.....	41
Gambar 4. 14 Visualisasi Data Suhu dan Kelembaban	45
Gambar 4. 15 Grafik Data Suhu dan Kelembaban	46
Gambar 4. 16 Visualisasi Data Tekanan Udara.....	46
Gambar 4. 17 Visualisasi <i>Wind Direction</i>	48
Gambar 4. 18 Visualisasi Data <i>Power</i>	49

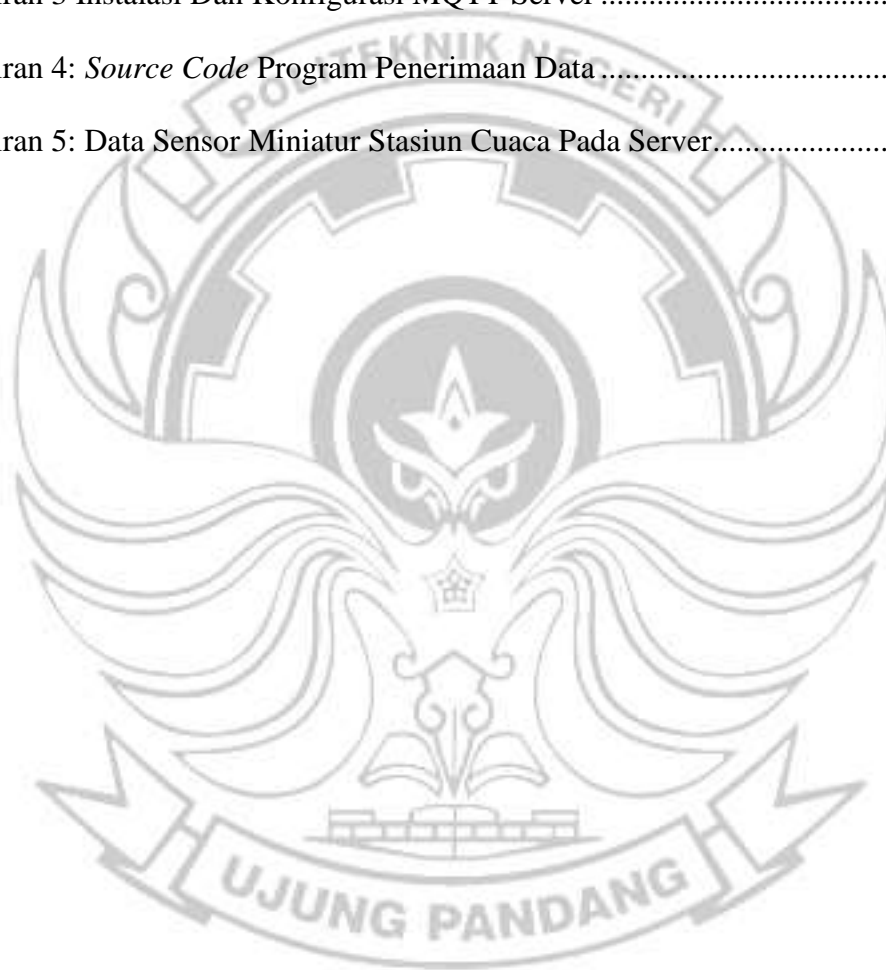


DAFTAR TABEL

	Halaman
Tabel 3. 1 Spesifikasi Perangkat Keras.....	16
Tabel 3. 2 Spesifikasi Perangkat Lunak.....	16
Tabel 4. 1 Data Sensor Miniatur Stasiun Cuaca Yang Dikirim Dan Diterima Server	42
Tabel 4. 2 Data Sensor Panel Surya Yang Dikirim Dan Diterima Di Server	43
Tabel 4. 3 Data Sensor BME280 Pada InfluxDB	45
Tabel 4. 4 Keterangan Data Arah Angin	47
Tabel 4. 5 Data <i>Wind Vane Direction</i>	47
Tabel 4. 6 Data Sensor <i>Power</i> Pada InfluxDB (mW)	49
Tabel 4. 7 Hasil Pengujian <i>Query</i>	50

DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Instalasi Dan Konfigurasi InfluxDB.....	55
Lampiran 2 Instalasi Dan Konfigurasi Grafana	58
Lampiran 3 Instalasi Dan Konfigurasi MQTT Server	59
Lampiran 4: <i>Source Code</i> Program Penerimaan Data	61
Lampiran 5: Data Sensor Miniatur Stasiun Cuaca Pada Server.....	64



SURAT PERNYATAAN

Saya bertanda tangan dibawah ini:

Nama: Titin Nurfadhila Sudirman

Nim: 425 15 030

Menyatakan dengan sebenar-benarnya bahwa segala pernyataan dalam skripsi ini yang berjudul **“Perancangan Dashboard Dan Query Untuk Monitoring Data Sensor Pada Miniatur Stasiun Cuaca Menggunakan InfluxDB Dan Grafana”** merupakan gagasan dan hasil karya sendiri dengan arahan komisi pembimbing, dan belum pernah diajukan dalam bentuk apapun pada perguruan tinggi dan instansi manapun.

Semua data dan informasi yang digunakan telah dinyatakan secara jelas dan dapat diperiksa kebenarannya. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan dari penulis lain telah disebutkan dalam naskah dan dicantumkan dalam skripsi ini.

Jika pernyataan saya tersebut diatas tidak benar, saya siap menanggung resiko yang ditetapkan oleh Politeknik Negeri Ujung Pandang



Makassar, 02 Agustus 2019

Titin Nurfadhila Sudirman
NIM: 425 15 030

PERANCANGAN DASHBOARD DAN QUERY UNTUK MONITORING DATA SENSOR PADA MINIATUR STASIUN CUACA MENGGUNAKAN INFLUXDB DAN GRAFANA

RINGKASAN

Informasi tentang prakiraan cuaca yang cepat dan tepat menjadi suatu hal yang penting karena cuaca menjadi bagian yang tidak dapat dipisahkan dari aktivitas manusia. Dalam proses pengamatan cuaca diperlukan alat untuk mengukur keadaan cuaca yaitu stasiun cuaca. Stasiun cuaca merupakan suatu fasilitas dengan instrumen dan peralatan untuk mengamati kondisi fisik atmosfer pada suatu lokasi atau wilayah pada saat tertentu atau dalam periode jangka pendek. Data hasil pengukuran stasiun cuaca perlu disimpan dalam *database* dan ditampilkan secara *real time* agar dapat di analisa. Salah satu *platform open source* untuk penyimpanan dan visualisasi data sensor yaitu InfluxDB sebagai *database* untuk pengelolaan data sensor dan Grafana untuk visualisasi data sensor. Tujuan dilakukannya penelitian ini adalah merancang *dashboard* dan *query* untuk memantau data sensor pada miniatur stasiun cuaca dalam bentuk visualisasi grafis menggunakan InfluxDB dan Grafana.

Berdasarkan hasil penelitian menunjukkan bahwa data sensor yang dikirim oleh miniatur stasiun cuaca dan panel surya dapat diterima di server dan disimpan ke *database* influxDB. Data sensor yang tersimpan pada influxDB selanjutnya ditampilkan dalam bentuk visualisasi *dashboard* menggunakan Grafana. Hasil rancangan *dashboard* yaitu *dashboard* miniatur stasiun cuaca menampilkan parameter cuaca yaitu suhu, kelembaban, tekanan udara, arah angin, kecepatan angin, intensitas cahaya matahari sedangkan pada panel surya yaitu arus, tegangan, daya dari sensor INA219 dihubungkan dengan AKI dan Solar Panel, data suhu, dan relay. Visualisasi ditampilkan dalam bentuk panel *graph*, *gauge*, *singlestat*, *table* maupun *text*. Hasil rancangan *dashboard* dan *query* ini, mampu menampilkan informasi mengenai parameter cuaca dalam bentuk visualisasi grafis secara *real time*.

Kata Kunci: *Miniatur Stasiun Cuaca, InfluxDB, Grafana, Internet of Things*

BAB I PENDAHULUAN

1.1 Latar Belakang

Informasi tentang prakiraan cuaca yang cepat dan tepat menjadi suatu hal yang penting, karena cuaca menjadi bagian yang tidak dapat dipisahkan dari aktivitas manusia dan mempengaruhi berbagai bidang kehidupan, seperti penentuan masa tanam (pertanian), kelayakan keberangkatan pesawat udara (transportasi) dan bidang lainnya (Wardani, 2011). Dalam proses pengamatan cuaca diperlukan instrumen yang akan ditempatkan dalam suatu lokasi tertentu untuk mewakili kondisi lingkungan daerah sekitarnya yang disebut stasiun cuaca (Sucipto, Hartawan dan Setiawan, 2017). Ada beberapa parameter yang diukur pada stasiun cuaca dengan menggunakan sensor. Data hasil keluaran dari sensor ini akan diproses dan disimpan di data loger kemudian dikirimkan ke server.

Data sensor pada server selanjutnya diolah di *database* dan ditampilkan dalam bentuk visualisasi *dashboard*. Pada penelitian sebelumnya, data hasil rancang bangun miniatur stasiun cuaca disimpan dan divisualisasikan menggunakan *IoT cloud server thingspeak* dan belum berjalan secara *real time*. Sistem dapat ditambahkan agar dapat bekerja secara *real time* dan dapat menyimpan data yang terbaca pada sebuah memori atau komputer (Nurbaitil dan Cahyati, 2018).

Oleh karena itu dibutuhkan aplikasi untuk *monitoring* data sensor secara *real time*. Sehingga dibutuhkan sistem *database* khusus yang mampu menangani proses *write* secara *continue*, karena tingkat *write* yang tinggi tersebut basis data relasional biasa kurang cocok digunakan dan *Time Series Database* (TSDB) menjadi solusi yang lebih rasional untuk data sensor yang merupakan data berdasarkan waktu

(*time series/historis*) (Dix, 2016). Serta dibutuhkan aplikasi untuk menampilkan data sensor dengan interval waktu yang kecil dalam bentuk *dashboard* secara *real time* yang mudah digunakan sehingga memudahkan dalam pengambilan keputusan. Salah satu *tools open source* yang mudah digunakan untuk monitoring data secara *real time* yaitu InfluxDB dan Grafana.

InfluxDB yang secara teknis mengungguli Cassandra DB sebanyak 4.5 kali lipat dalam kecepatan *write throughput*, dan menggunakan *resource* 10.8 kali lebih hemat penggunaan *storage* berkat *compression* nya (Influxdata, 2018a). Konfigurasi InfluxDB dapat menghemat ruang untuk menyimpan data dalam jangka waktu tertentu, secara otomatis berakhir dan menghapus semua data yang tidak diperlukan dari sistem (Influxdata, 2018b). Grafana merupakan salah satu *tools* yang digunakan untuk visualisasi dalam bentuk *dashboard* secara *real time*. Grafana ditawarkan untuk visualisasi data penyajian data deret waktu baik untuk Graphite, ElasticSearch, Prometheus, InfluxDB, OpenTSDB, dan KairosDB (Grafana, 2019).

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, dapat dirumuskan permasalahan tugas akhir ini adalah bagaimana merancang *dashboard* dan *query* untuk memantau data sensor pada miniatur stasiun cuaca dalam bentuk visualisasi grafis menggunakan InfluxDB dan Grafana secara *real time*?

1.3 Ruang Lingkup Penelitian

Agar dalam penulisan tidak terdapat penyimpangan dari tujuan penulisan dan permasalahan yang dicakup tidak terlalu luas maka masalah akan dibatasi pada

perancangan *dashboard* dan *query* untuk visualisasi data sensor pada miniatur stasiun cuaca menggunakan InfluxDB dan Grafana secara *real time*.

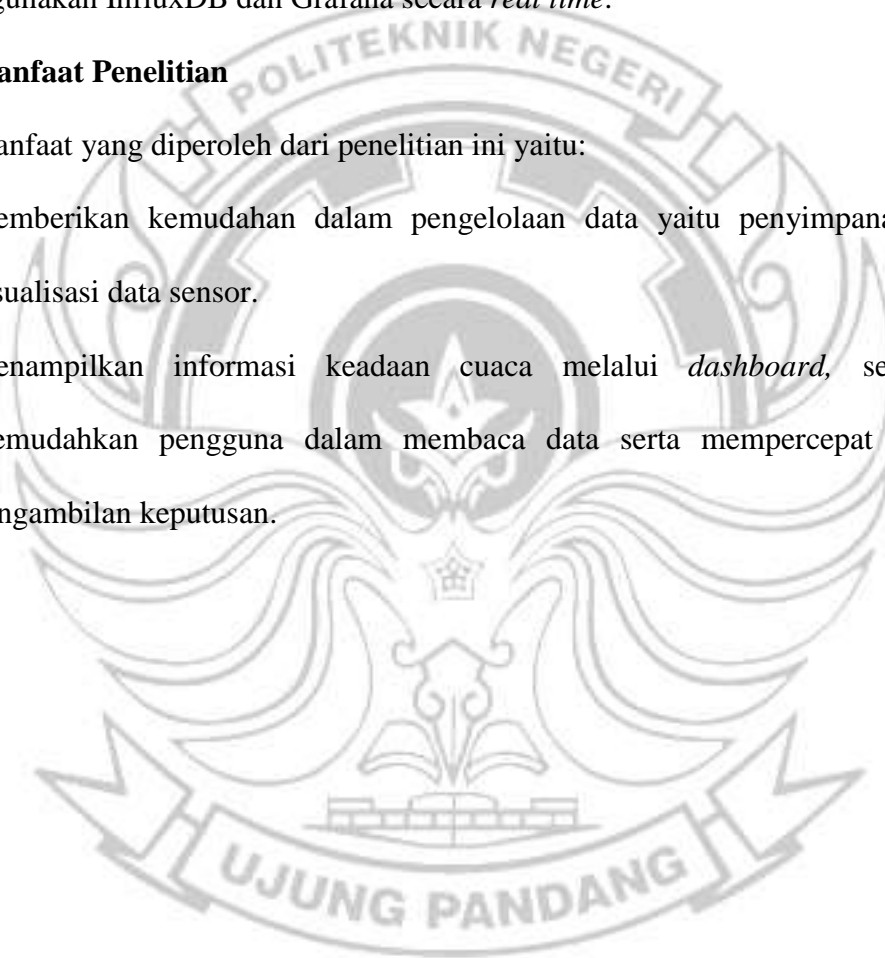
1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk merancang *dashboard* dan *query* untuk memantau data sensor pada miniatur stasiun cuaca dalam bentuk visualisasi grafis menggunakan InfluxDB dan Grafana secara *real time*.

1.5 Manfaat Penelitian

Manfaat yang diperoleh dari penelitian ini yaitu:

1. Memberikan kemudahan dalam pengelolaan data yaitu penyimpanan dan visualisasi data sensor.
2. Menampilkan informasi keadaan cuaca melalui *dashboard*, sehingga memudahkan pengguna dalam membaca data serta mempercepat proses pengambilan keputusan.



BAB II TINJAUAN PUSTAKA

2.1 Monitoring Data Stasiun Cuaca

2.1.1. *Monitoring*

Monitoring adalah proses rutin pengumpulan data dan pengukuran kemajuan atas objektif program. Memantau perubahan yang fokus pada proses dan keluaran. Monitoring menyediakan data dasar untuk menjawab permasalahan. Monitoring akan memberikan informasi tentang status dan kecenderungan bahwa pengukuran dan evaluasi yang diselesaikan berulang dari waktu ke waktu, pemantauan umumnya dilakukan untuk tujuan tertentu, untuk memeriksa terhadap proses suatu objek atau untuk mengevaluasi kondisi atau kemajuan menuju tujuan hasil manajemen atas efek tindakan dari beberapa jenis tindakan untuk mempertahankan manajemen yang sedang berjalan (Yahwe, Isnawaty dan Aksara, 2016).

2.1.2. Miniatur Stasiun Cuaca

Stasiun cuaca atau *Automated Weather Station* (AWS) merupakan suatu fasilitas dengan instrumen dan peralatan untuk mengamati kondisi fisik atmosfer pada suatu lokasi atau wilayah pada saat tertentu atau dalam periode jangka pendek (maksimum harian) (Krisnandi, 2011). Pada dasarnya miniatur stasiun cuaca memiliki sistem yang sama dengan AWS, namun tidak sempurna karena terdapat bagian tidak masuk dalam AWS. Hal ini yang mendasari mengapa disebut miniatur stasiun cuaca. Sebagian besar dari miniatur stasiun cuaca ini menggunakan sensor yang dirancang sedemikian rupa sehingga berfungsi seperti AWS.

2.2 Time Series

Time series adalah pengukuran atau peristiwa yang di lacak, di monitor, downsampled dan dikumpulkan dari waktu ke waktu seperti metrik Server, pemantauan kinerja aplikasi, jaringan data, data sensor, peristiwa, dan data analisis lainnya (Dix, 2016).

Time series digunakan dalam berbagai konteks, yang paling umum adalah (Dix, 2016):

- 1) Analisis Rangkaian Waktu (*time series*): Analisis Rangkaian Waktu digunakan untuk menyelidiki bagaimana suatu variabel berubah dari waktu ke waktu.
- 2) Analisis Regresi: Analisis Regresi dapat digunakan untuk memeriksa bagaimana perubahan yang terkait dengan variabel tertentu yang dapat menyebabkan pergeseran dalam variabel lain selama periode waktu yang sama.
- 3) Peramalan Seri Waktu: Peramalan Seri Waktu menggunakan informasi terkait nilai historis dan pola terkait untuk memprediksi aktivitas di masa mendatang. Misalnya, Peramalan Ekonomi, Peramalan Cuaca, Peramalan Gempa Bumi (*seismic time series*), Peramalan Penjualan, dll.

Menurut (Dedi, 2011:1), untuk dapat memahami pemodelan runtun waktu/*time series*, perlu diketahui beberapa jenis data menurut waktu, yang dapat dibedakan sebagai berikut:

- 1) Data *Time Series* (Runtun Waktu) yakni jenis data yang terdiri atas variabel - variabel yang dikumpulkan menurut urutan waktu dalam suatu rentang waktu tertentu. Jika waktu dipandang bersifat diskrit (waktu dapat dimodelkan bersifat kontinyu), frekuensi pengumpulan selalu sama (*equidistant*). Dalam kasus

diskrit, frekuensi dapat berupa misalnya detik, menit, jam, hari, minggu, bulan atau tahun dan lain-lain.

- 2) Data *Cross-Section* yakni jenis data yang terdiri atas variabel-variabel yang dikumpulkan pada sejumlah individu atau kategori pada suatu titik waktu tertentu. Model yang digunakan untuk memodelkan data tipe ini seperti model regresi (*Cross-Section*). Analisis data *cross-sectional* biasanya terdiri dari membandingkan perbedaan antara subyek.
- 3) Data Panel atau *Pooled* yakni tipe data yang terdiri atas variabel-variabel yang dikumpulkan menurut urutan waktu dalam suatu rentang waktu tertentu pada sejumlah individu atau kategori. Model yang digunakan untuk pemodelan data ini seperti model data panel dan model runtun waktu multivariat. Secara ekuivalen, dikenal juga tipe data longitudinal dengan frekuensi data tidak harus *equidistant*, namun analisis statistika yang dilakukan dalam model longitudinal berbeda tujuannya dengan analisis untuk model panel.

2.3 Penyimpanan Data *Time Series*

2.3.1 *Time Series Database* (TSDB)

Time Series Database (TSDB) adalah *database* yang dioptimalkan untuk data *time series*. TSDB memungkinkan penggunaanya untuk membuat, menghitung, memperbarui, dan mengatur berbagai rangkaian waktu dengan cara yang lebih efisien (Naqvi dan YFantidou, 2017).

Adapun manfaat penggunaan *Time Series Database* sebagai berikut (Naqvi dan YFantidou, 2017):

1. Skalabilitas dan kinerja tinggi

2. Mengurangi *downtime*: Dalam skenario kehidupan nyata ada beberapa situasi di mana ketersediaan sangat penting, arsitektur *database* yang dibangun untuk data deret waktu dapat menghindari *downtime* pada data bahkan jika terjadi partisi jaringan atau kegagalan perangkat keras
3. Biaya lebih rendah
4. Keputusan bisnis yang lebih baik: Karena TSDB memungkinkan organisasi untuk memonitor dan menganalisis data secara *real time*, TSDB membantu dalam membuat penyesuaian lebih cepat dan lebih akurat untuk perubahan infrastruktur, konsumsi energi, pemeliharaan perangkat, atau keputusan besar lainnya yang memengaruhi bisnis.

RANK	DBMS	SCORE		
		AUG 2019	24 MOS +	12 MOS +
1	InfluxDB	18.34	+10.24	+6.77
2	Kdb+	5.76	+4.09	+2.25
3	Prometheus	3.48	+2.88	+1.96
4	Graphite	3.4	+1.10	+0.80
5	RRDtool	3.19	+0.13	+0.72
6	OpenTSDB	2.66	+0.76	+1.51
7	Druid	1.82	+0.84	+0.63
8	TimescaleDB	1.73	+1.73	+1.52
9	KairosDB	0.66	+0.17	+0.17
10	GridDB	0.49	+0.00	+0.49

Source: DB-Engines
independent comparison of InfluxDB to other TSDBs, including OpenTSDB and Prometheus
23 SYSTEMS TRACKING, August 2019

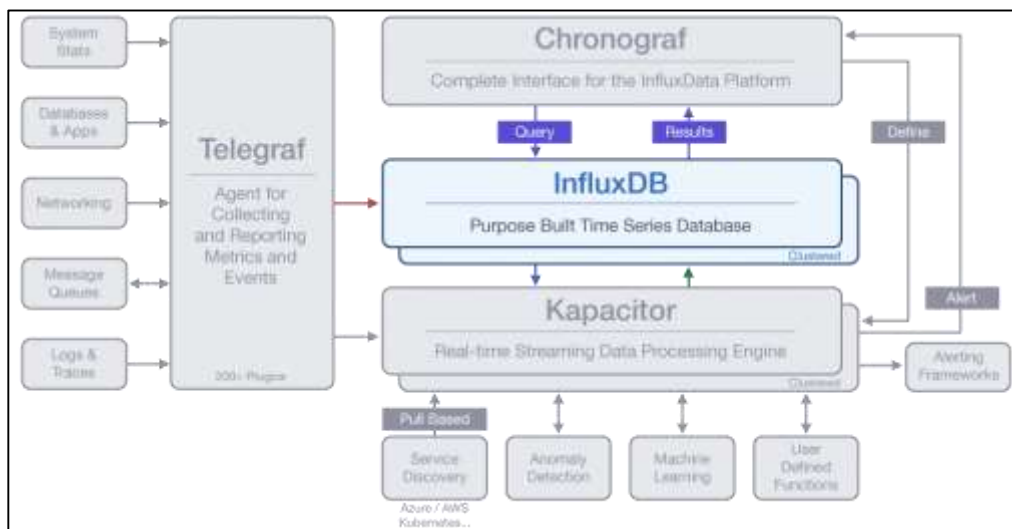
Sumber:(Influxdata, 2018b)

Gambar 2. 1 DB-Engines Ranking of Time Series DBMS

2.3.2 Pengertian InfluxDB

InfluxDB adalah sebuah *database time series open-source* yang dikembangkan oleh InfluxData. InfluxDB didukung oleh Bahasa Go (Naqvi dan YFantidou, 2017). InfluxDB digunakan sebagai penyimpanan data untuk setiap kasus yang melibatkan sejumlah besar data *time-stamped*, termasuk pemantauan DevOps, data log, metrik

aplikasi, data sensor IoT, dan analisis *real-time*. Konfigurasi InfluxDB dapat menghemat ruang untuk menyimpan data dalam jangka waktu tertentu, secara otomatis berakhir dan menghapus semua data yang tidak diperlukan dari sistem. InfluxDB juga menawarkan bahasa kueri seperti SQL untuk berinteraksi dengan data (Influxdata, 2018b).



Sumber: (Influxdata, 2018b).

Gambar 2. 2InfluxDB

2.3.3 Fitur – Fitur InfluxDB

Berikut adalah beberapa fitur yang saat ini didukung oleh InfluxDB (Influxdata, 2018b):

1) Kinerja Tinggi

InfluxDB adalah penyimpanan data berkinerja tinggi yang ditulis khusus untuk data deret waktu. Hal ini memungkinkan untuk mengolah *throughput* tinggi, kompresi, dan kueri *real-time* dari data yang sama. InfluxDB ditulis dalam Bahasa Go dan dikompilasi menjadi satu biner tanpa ketergantungan eksternal. Ini menyediakan kemampuan menulis dan kueri dengan antarmuka baris perintah, HTTP API *built-in*, satu set pustaka klien (seperti Go, Java, dan Javascript untuk

beberapa nama) dan dengan *plugin* untuk format data umum seperti Telegraf, Graphite, Collectd, dan OpenTSDB.

2) *SQL-Like Queries*

InfluxDB menyediakan InfluxQL sebagai bahasa kueri seperti SQL untuk berinteraksi dengan data. Menyediakan fitur khusus untuk menyimpan dan menganalisis data deret waktu. InfluxQL juga mendukung ekspresi reguler, ekspresi aritmatika, dan fungsi khusus deret waktu untuk mempercepat pemrosesan data.

3) Downsampling dan Penyimpanan Data

InfluxDB dapat menangani jutaan titik data per detik. Bekerja dengan banyak data selama jangka waktu yang panjang dapat menciptakan masalah penyimpanan. InfluxDB akan secara otomatis memadatkan data untuk meminimalkan ruang penyimpanan sistem. Selain itu, InfluxDB dapat menurunkan sampel data, menjaga data mentah presisi tinggi hanya untuk waktu yang terbatas, dan menyimpan presisi yang lebih rendah, meringkas data lebih lama atau selamanya. InfluxDB menawarkan dua fitur — *Continuous Queries (CQ)* dan *Retention Policies (RP)* — yang membantu mengotomatiskan proses downsampling data dan kadaluwarsa data lama.

2.4 Aplikasi Penerimaan Data

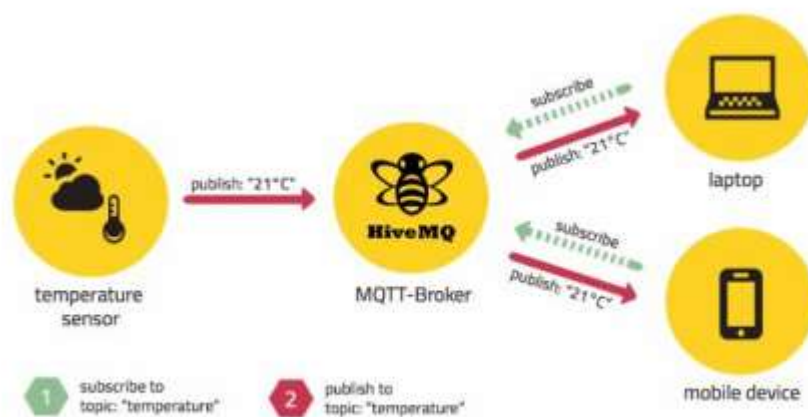
2.4.1 *Message Queue Telemetry Transport (MQTT)*

Data sensor pada miniatur stasiun cuaca dikirim ke server menggunakan protokol MQTT. MQTT merupakan protokol komunikasi *publish/subscribe topic based* yang sangat sederhana dan ringan, yang didesain untuk alat yang memiliki

kemampuan terbatas, *bandwidth* yang rendah, *latency* yang tinggi atau jaringan yang kurang dapat diandalkan. Prinsip dari desain ini adalah untuk meminimalkan penggunaan *bandwidth* jaringan dan kebutuhan sumber daya pada perangkat serta pada waktu yang sama juga berusaha untuk memastikan keandalan dan kepastian dari pengiriman data (MQTT, 2019).

Protokol MQTT didasarkan pada prinsip mempublikasikan pesan dan berlangganan topik, atau "pub/sub". Banyak klien yang terhubung ke *broker* dan berlangganan topik yang mereka minati. Klien juga terhubung ke broker dan mempublikasikan pesan ke topik. Klien dapat berlangganan ke topik yang sama dan melakukan dengan informasi sesuka mereka. *broker* dan MQTT bertindak sebagai antarmuka sederhana dan umum untuk semua yang terhubung (Eclipse Foundation, 2019).

Pola pesan *publish-subscribe* membutuhkan broker pesan terlihat seperti pada Gambar 2.3.



Sumber:(Satria, Satrya dan Herutomo, 2015)

Gambar 2. 3 Skema *Publish* Dan *Subscribe* Pada MQTT

2.4.1. Python

Python merupakan bahasa pemrograman yang digunakan untuk program penerimaan data di server. Python adalah bahasa pemrograman interpretatif multiguna. Tidak seperti bahasa lain yang susah untuk dibaca dan dipahami, python lebih menekankan pada keterbacaan kode agar lebih mudah untuk memahami sintaks. Hal ini membuat Python sangat mudah dipelajari baik untuk pemula maupun untuk yang sudah menguasai bahasa pemrograman lain dan paling baik digunakan secara luas. Python adalah bahasa yang sangat ekspresif, yang berarti bahwa python biasanya dapat menulis lebih sedikit baris kode daripada yang diperlukan untuk aplikasi yang setara yang ditulis dalam, seperti C ++ atau Java (Summerfiel, 2010).

2.5 Dashboard

Dashboard merupakan sebuah tampilan visual dari informasi penting yang dibutuhkan untuk tercapainya suatu tujuan, digabungkan dan diatur pada sebuah layar yang menghasilkan informasi yang dibutuhkan dan didapat secara sekilas. *Dashboard* merupakan sebuah tampilan pada satu monitor komputer penuh yang berisi informasi yang bersifat kritis, agar kita dapat mengetahui hal – hal yang perlu diketahui. Biasanya kombinasi teks dan grafik, tetapi lebih ditekankan pada grafik (Few, 2006).

2.6 Visualisasi Data

Visualisasi data berkaitan dengan desain, pengembangan, dan aplikasi representasi grafis data yang dihasilkan komputer. Ini memberikan representasi data yang efektif dari data yang berasal dari sumber yang berbeda. Ini

memungkinkan para pembuat keputusan untuk melihat analitik dalam bentuk visual dan membuatnya mudah bagi mereka untuk memahami data. Ini membantu mereka menemukan pola, memahami informasi, dan membentuk opini (Sadiku *dkk.*, 2016).

Ada beberapa aplikasi *open source* untuk visualisasi data seperti Thingspeak, MyDevices Cayenne, Ubidots, Tembo, Ganglia, Collectd, Grafana dan aplikasi lainnya. Pada penelitian ini digunakan Grafana untuk visualisasi data, karena grafana adalah salah satu alat visualisasi dan pemantauan yang populer, mendukung lebih dari 30 *open source*, salah satunya InfluxDB yang digunakan untuk penyimpanan data sensor secara *real time*.

2.6.1. Grafana

Grafana adalah perangkat *open source* untuk analisis dan visualisasi metrik. Grafana paling sering digunakan untuk memvisualisasikan data deret waktu untuk infrastruktur dan analitik aplikasi dan juga banyak digunakan di domain lain termasuk sensor industri, otomatisasi rumah, cuaca, dan kontrol proses. Grafana mendukung banyak *storage backends* yang berbeda untuk data *time series* (*Source Data*). Setiap sumber data memiliki kueri editor tertentu yang disesuaikan untuk fitur dan kemampuan tertentu (Grafana, 2019).



Sumber: (Grafana, 2019)

Gambar 2. 4 *Dashboard Grafana*

2.6.2. Kelebihan Grafana

Berikut beberapa kelebihan menggunakan grafana seperti:

1. Grafana adalah program interaktif, yang memungkinkan pengguna untuk menghasilkan gambar berdasarkan data yang disimpan pada file disk. Keuntungan utama dan orisinalitas sistem adalah bahwa pengguna dapat secara interaktif menentukan efek visual suatu gambar dan segera melihatnya pada data yang disimpan pada file disk. Selain itu, gambar apa pun yang dihasilkan oleh Grafana masih dapat diedit oleh editor grafis untuk keperluan umum. Sistem ini memungkinkan pengguna untuk mendapatkan gambar yang tepat untuk buku, kertas, atau laporan apa pun (Thalmann, 2013).
2. Grafana adalah perangkat lunak sumber terbuka, kaya fitur, analitik, dan pemantauan gratis yang dijalankan sebagai aplikasi web untuk membuat dasbor dan grafik. Grafana adalah salah satu alat visualisasi dan pemantauan yang populer, mendukung lebih dari 30 *open source* dari berbagai jenis sumber data

(*source data*) yang menyimpan data *time series* seperti Graphite, ElasticSearch, Prometheus, InfluxDB, OpenTSDB, dan KairosDB. Setiap *source data* memiliki *Query Editor* tertentu yang disesuaikan untuk fitur dan kemampuan tertentu (Yong, 2019).

3. Grafana menjadi salah satu alat visualisasi populer untuk memantau sejumlah data besar yang mudah diatur, dirawat, lebih mudah digunakan, dan menampilkan metrik dengan gaya yang baik (Yong, 2019).

2.6.3. Fitur – Fitur Grafana

Berikut beberapa fitur dari grafana (Grafana, 2019):

1) Panel

Panel adalah blok visualisasi dasar di Grafana. Setiap Panel menyediakan Editor Kueri (bergantung pada sumber data yang dipilih di panel) yang memungkinkan untuk mengekstrak visualisasi yang sempurna untuk ditampilkan pada Panel dengan memanfaatkan Editor Kueri.

2) *Feature dashboard*

Pada grafana menyediakan fitur dasbor seperti *annotations, playlist, search, sharing, time range, export and import, scripted Dashboards, Dashboard Version History, JSON Model.*

4) *Data source*

Grafana mendukung banyak *backend* penyimpanan yang berbeda untuk *data source* (Sumber Data). Setiap Sumber Data memiliki Editor Kueri khusus yang disesuaikan untuk fitur dan kemampuan yang diungkapkan oleh Sumber Data

tertentu. Sumber data berikut secara resmi didukung grafana: Graphite, InfluxDB, OpenTSDB, Prometheus, Elasticsearch, CloudWatch.

5) *Alerting*

Pada grafana memungkinkan untuk melampirkan aturan ke panel dasbor. Pada saat menyimpan dasbor, Grafana akan mengekstrak aturan *alerting*/peringatan ke penyimpanan aturan peringatan terpisah dan menjadwalkannya untuk evaluasi di tab *alert* pada panel grafik.

6) *Keyboard Shortcut*

Pada grafana menyediakan pintasan *keyboard* untuk menjalankan sebuah program.



BAB III METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

Penelitian ini dilaksanakan di Kampus 1 Politeknik Negeri Ujung Pandang, tepatnya di Jl. Perintis Kemerdekaan Km. 10 Tamalanrea, Makassar. Penelitian ini dimulai dari bulan November 2018 sampai Juli 2019.

3.2 Alat dan Bahan

Alat dan bahan yang digunakan pada penelitian ini adalah sebagai berikut:

1) Perangkat Keras (*Hardware*)

Perangkat keras yang dibutuhkan seperti pada tabel 3.1.

Tabel 3. 1 Spesifikasi Perangkat Keras

No	Hardware	Sistem Operasi	Spesifikasi	Jumlah
1	Personal Computer (PC)/Notebook/Laptop	Windows 10	Intel(R) Core(TM) i3 RAM 8192MB	1

2) Perangkat Lunak (*Software*)

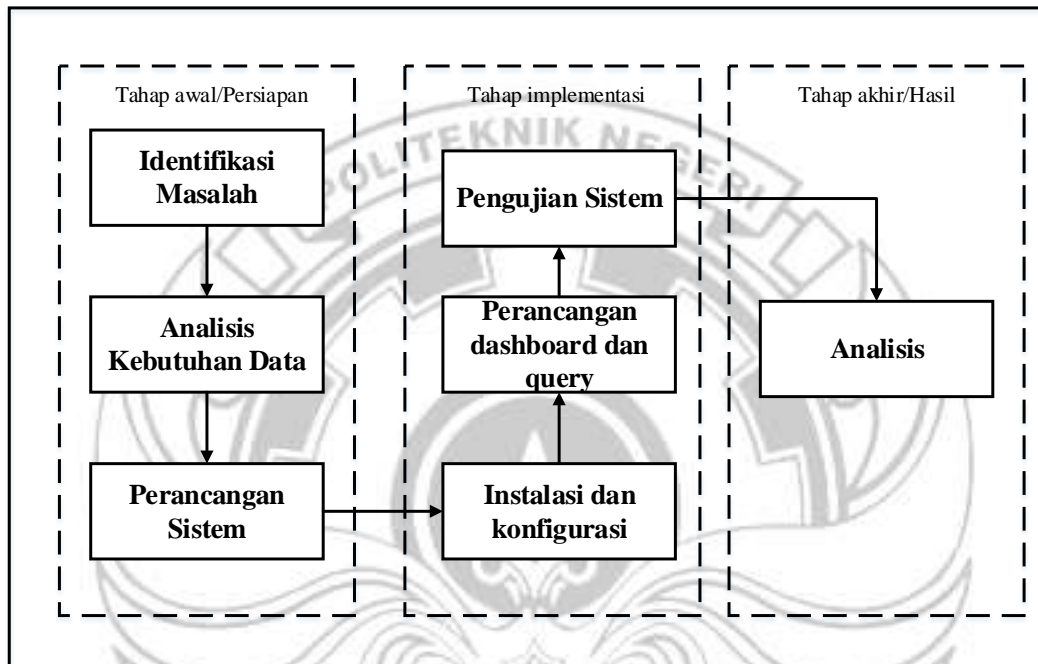
Perangkat lunak yang dibutuhkan seperti pada tabel 3.2.

Tabel 3. 2 Spesifikasi Perangkat Lunak

No	Keterangan	Perangkat Lunak
1	Sistem Operasi	Debian 8 (<i>Jessie</i>)
2	<i>Software</i>	influxDB_1.7.7_amd64.deb grafana_6.2.5_amd64.deb
3	<i>Web Browser</i>	Chrome, Mozilla firefox, Opera Browser
4	Bahasa pemrograman	Python 2.7.9
5	Aplikasi desain	Microsoft Visio 2016

3.3 Metode Penelitian

Agar penelitian yang dilakukan dapat berjalan dengan baik dan terstruktur maka perlu sebuah metode penelitian sehingga hasil yang diperoleh sesuai dengan tujuan penelitian. Adapun tahapan penelitian seperti pada Gambar 3. 1.



Gambar 3. 1 Tahapan Penelitian

3.3.1 Identifikasi Masalah

Tahap pertama yang dilakukan dalam penelitian ini adalah identifikasi masalah serta cakupan sistem yang akan dibuat. Proses identifikasi masalah dilakukan dengan melakukan studi literatur dengan mencari berbagai acuan baik melalui buku, jurnal, tugas akhir, maupun artikel dengan narasumber yang jelas dan terpercaya serta mencari kebutuhan yang diperlukan sistem.

3.3.2 Analisis Kebutuhan Data

Pada penelitian ini data yang diperlukan untuk ditampilkan pada *dashboard* monitoring di Grafana yaitu data dari miniatur stasiun cuaca dan data panel surya

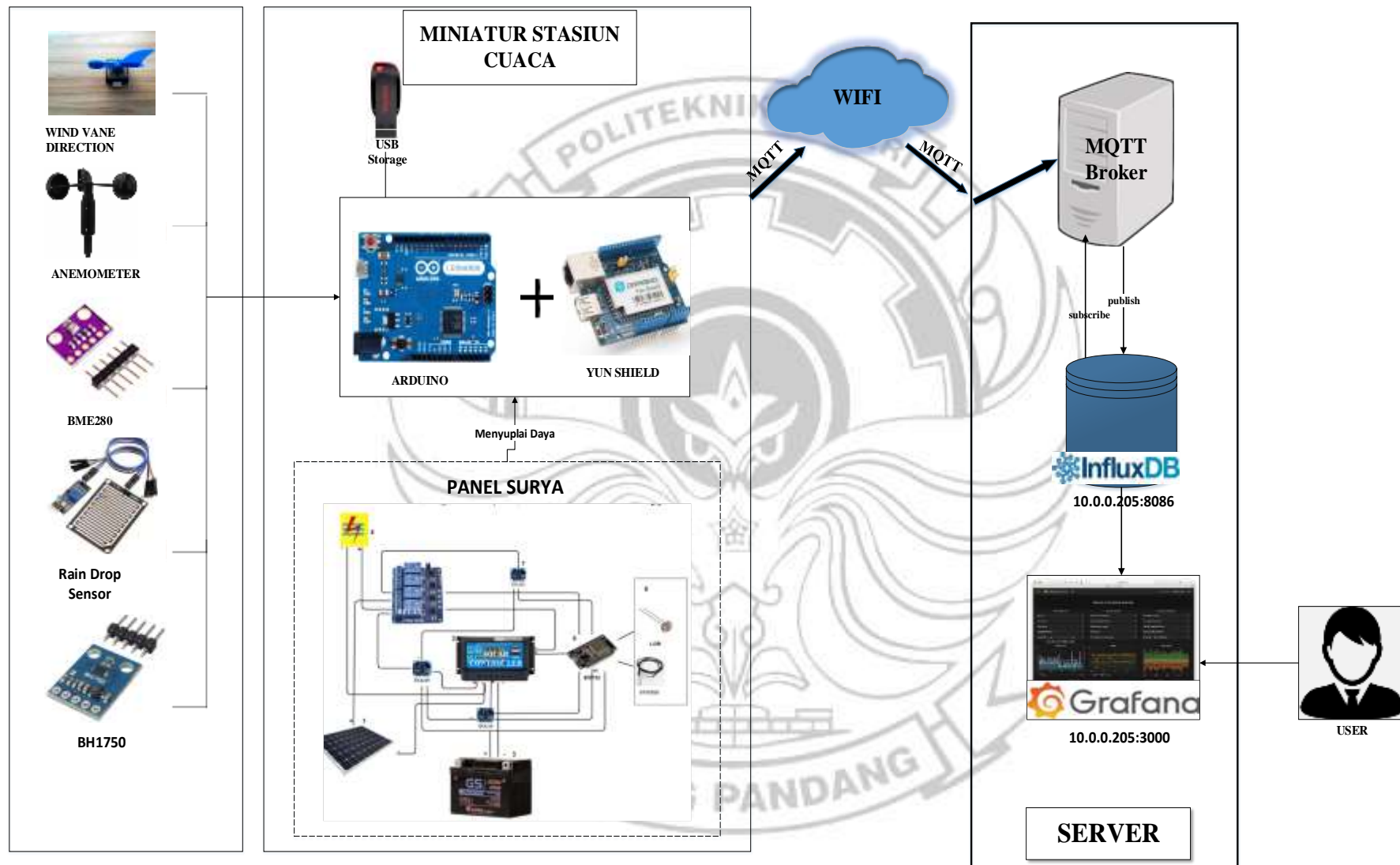
yang merupakan bagian dari penelitian yang dilakukan. Data pada minatur stasiun cuaca terdiri dari data suhu, kelembaban, tekanan udara, arah angin, kecepatan angin, intensitas cahaya matahari, sedangkan pada panel surya yaitu arus, tegangan, daya dari sensor INA219 dihubungkan dengan AKI dan Solar Panel, data suhu, dan relay. Data pada Miniatur stasiun cuaca dan panel surya dikirim ke server InfluxDB dan Grafana menggunakan protokol MQTT. Adapun rancangan sistem secara keseluruhan dari penelitian ini mulai dari rancangan miniatur stasiun cuaca dan panel surya serta server dapat dilihat pada Gambar 3.2.

3.3.3 Perancangan Sistem

Tahapan ketiga yang dilakukan pada penelitian ini adalah perancangan sistem. Tujuan dari tahap ini adalah memberikan gambaran tentang perancangan dari sistem sesuai dengan hasil identifikasi masalah dan analisis kebutuhan. Perancangan digambarkan dengan beberapa diagram yaitu diagram blok miniatur stasiun cuaca dan diagram alir.

1) Diagram Blok Miniatur Stasiun Cuaca

Diagram blok merupakan Gambaran umum sistem miniatur stasiun cuaca yang akan dibuat. Diagram sistem ini menggambarkan proses pengambilan data sensor pada miniatur stasiun cuaca sampai proses visualisasi di server. Diagram blok miniatur stasiun cuaca dapat dilihat pada Gambar 3.2.



Gambar 3. 2 Diagram Blok Miniatur Stasiun Cuaca

Dari Gambar 3.2, dapat diamati ada 2 bagian penyusun dari sistem miniatur stasiun cuaca yang dibuat, yaitu bagian miniatur stasiun cuaca dan server. Bagian stasiun terdapat bagian input berupa sensor sebagai penerima input dari parameter cuaca, Yun Shield dan Arduino sebagai pengontrol sistem, serta bagian panel surya sebagai cadangan sumber listrik. Selanjutnya bagian server terdiri dari Mosquitto server, InfluxDB, dan Grafana server.

Secara umum, cara kerja dari alat stasiun cuaca ini adalah bagian sensor yang digunakan untuk menangkap energi yang diukur pada masing-masing sensor, kemudian oleh sensor diubah menjadi suatu besaran tertentu, lalu diproses dan dikirim oleh Yun Shield dan Arduino. Pada Yun Shield data dari sensor disimpan pada *SD card* sebagai data logger sebelum dikirim ke server. Begitupun data dari panel surya sebagai penyuplai daya akan langsung dikirim ke server. Data dari miniatur stasiun cuaca ini dikirim ke server menggunakan protokol MQTT. Selanjutnya proses penerimaan data di server. Apabila data yang dikirim sesuai dengan format masukan yang ditentukan server, pesan akan diimpor ke InfluxDB. Data sensor kemudian divisualisasikan melalui Grafana *dashboard*. Visualisasi melalui Grafana ini dapat diakses oleh user yang membutuhkan data mengenai keadaan cuaca dengan menggunakan jaringan internet yang tersedia di kampus 1 Politeknik Negeri Ujung Pandang.

2) Diagram Alir

Dalam menggambarkan urutan proses dengan proses yang lainnya dalam suatu program maka digunakan diagram alir yang merupakan suatu bagan dengan

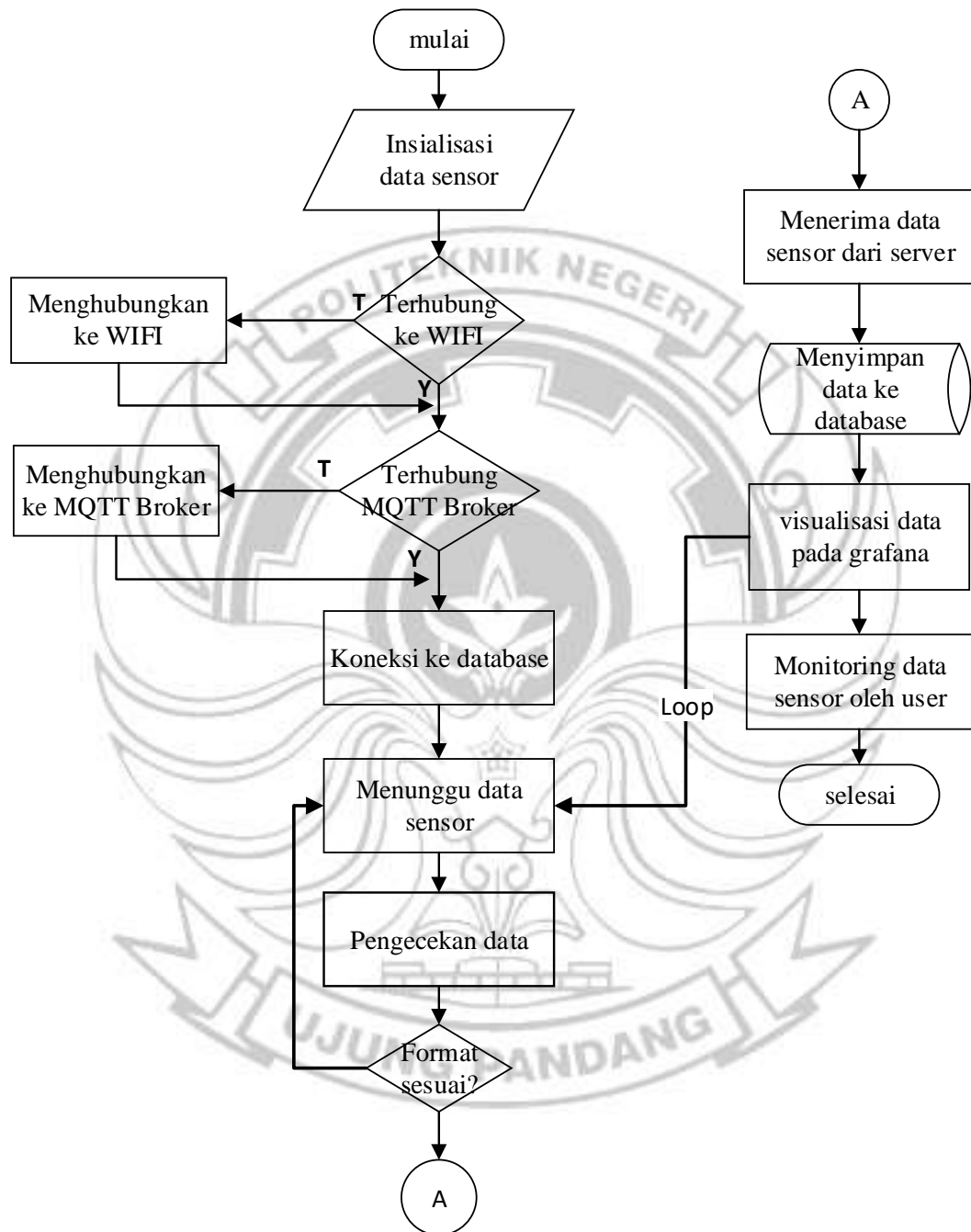
simbol-simbol tertentu yang menggambarkan urutan proses. Diagram alir pada sistem yang dibuat dapat dilihat pada Gambar 3.3.

Pada Gambar 3.3 menunjukkan bahwa sistem dapat berjalan apabila terkoneksi ke jaringan internet kampus 1 Politeknik Negeri Ujung Pandang. Apabila belum terhubung, maka harus dihubungkan ke jaringan internet karena proses penerimaan dan visualisasi pada server dapat berjalan apabila terkoneksi ke jaringan internet. Setelah itu, server juga harus terkoneksi dengan MQTT Broker untuk proses komunikasi antara server dengan miniatur stasiun cuaca dan panel surya. Selanjutnya terkoneksi dengan *database* yang telah dibuat pada *influx*. *Database* yang dibuat disesuaikan dengan *database* yang ditetapkan pada program pengiriman data dari miniatur stasiun cuaca maupun panel surya.

Selanjutnya server akan menunggu hingga data dikirim dari stasiun cuaca. Stasiun cuaca akan mengirim sebuah pesan/*topic* (*publish*) dengan tipe data tertentu ke broker. Broker MQTT akan melakukan pengecekan dengan memfilter pesan-pesan yang masuk dan mendistribusikannya kepada klien yang tertarik menerima pesan-pesan tersebut / *subscriber* yakni ke InfluxDB. influxDB sebelumnya telah mendaftarkan diri kepada broker.

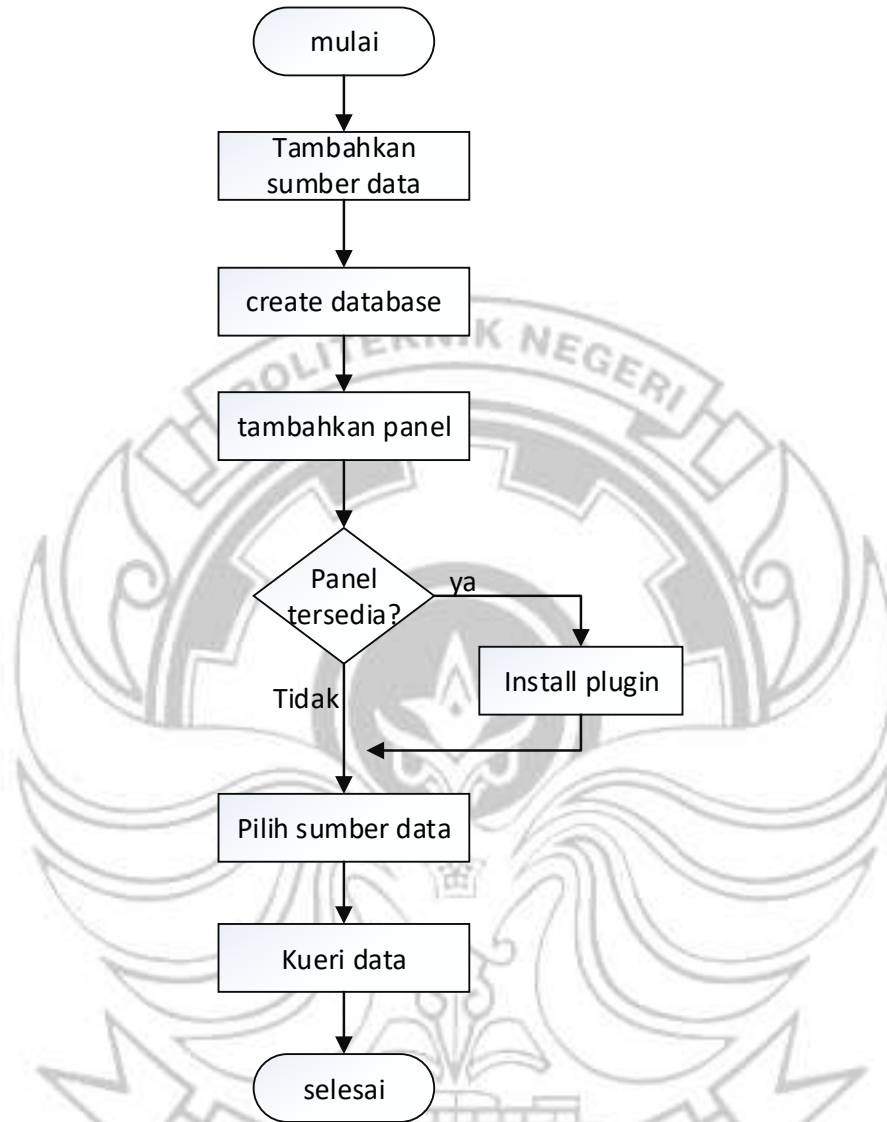
Apabila pesan/*topic* yang dikirim sesuai dengan format masukan yang ditentukan server, pesan akan diimpor ke InfluxDB. Apabila data yang dikirim tidak sesuai, maka akan kembali ke proses sebelumnya yaitu menunggu pengiriman data dari stasiun cuaca yang sesuai dengan format yang ditentukan pada server. Data pada server InfluxDB selanjutnya dirancang bentuk visualisasinya dengan grafana

server. Proses pembacaan hingga visualisasi data sensor ini dilakukan terus menerus (*Looping*) setiap data dikirim dari stasiun cuaca.



Gambar 3. 3 Diagram Alir Sistem

Adapun proses visualisasi data sensor pada grafana dapat dilihat pada Gambar 3.4.



Gambar 3. 4 Diagram Alir Visualisasi Data di Grafana

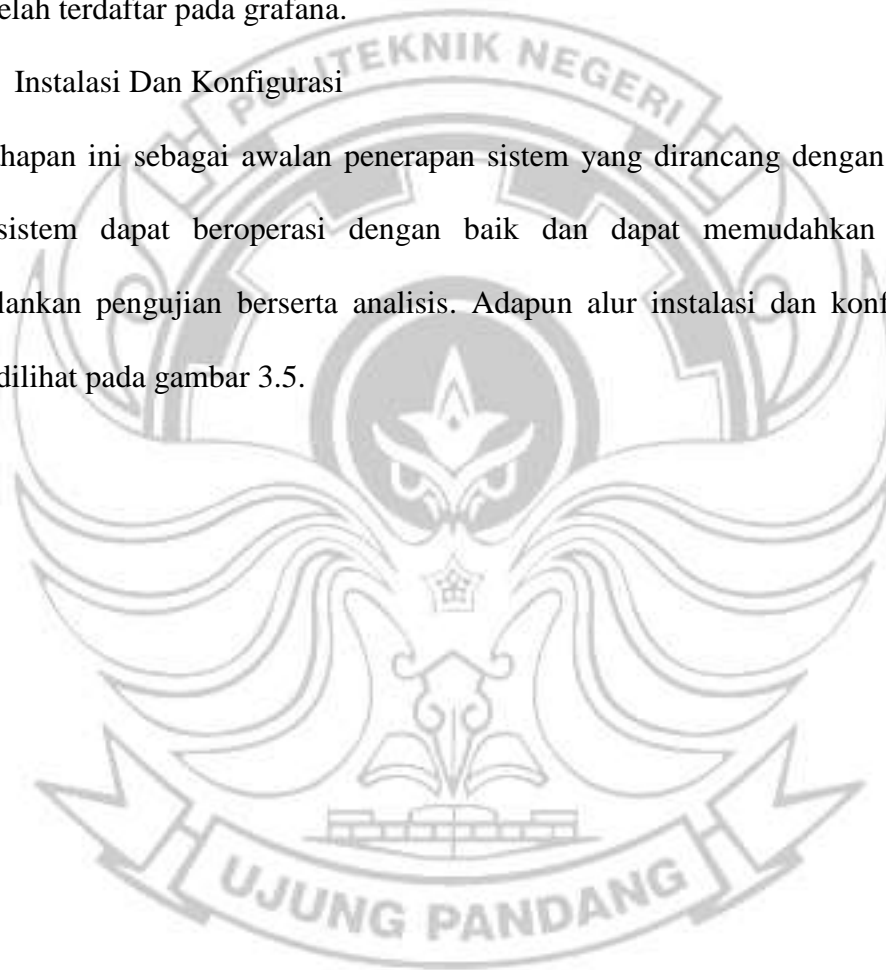
Gambar 3.4 menunjukkan proses visualisasi data sensor pada grafana.

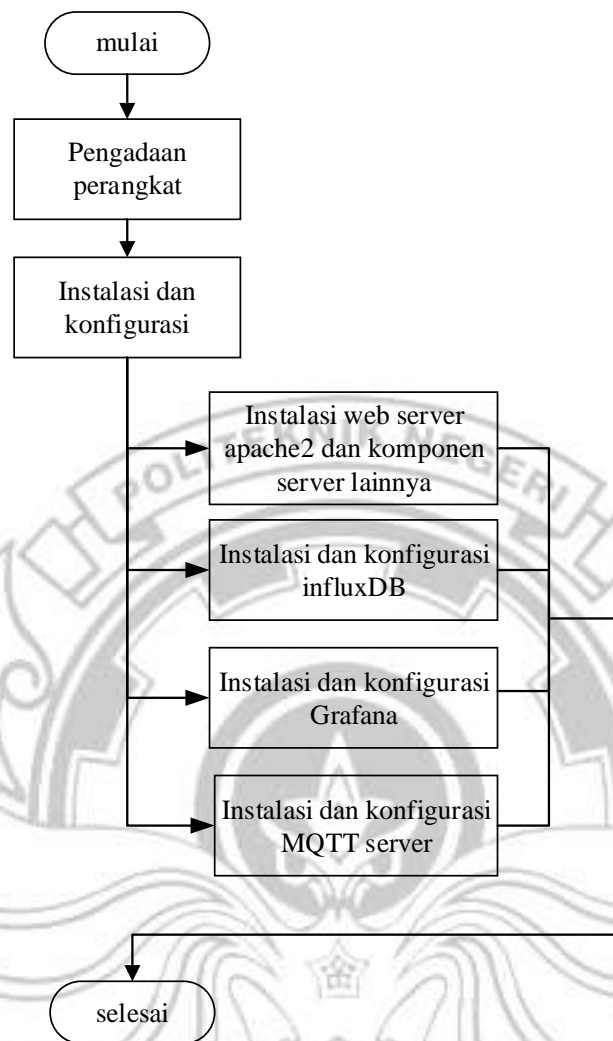
Dimulai dengan menambahkan sumber data dengan memilih InfluxDB sebagai tipe sumber data. Hal ini karena InfluxDB merupakan *database* yang digunakan untuk penyimpanan data sensor. Setelah itu membuat *dashboard*. Pada grafana menyediakan panel grafik seperti tabel, grafik, gauge, teks dan lainnya. Setiap data sensor yang tersimpan pada *database* influxDB memiliki bentuk panel yang

berbeda. Apabila jenis panel yang tersedia tidak sesuai dengan bentuk grafik yang akan dirancang, maka pada grafana telah disediakan plugin panel yang dapat diinstal sesuai dengan kebutuhan sistem yang akan dibuat. Panel memiliki kueri editor yang dapat diatur sehingga menampilkan informasi yang diinginkan. Selanjutnya *dashboard* yang telah dibuat dapat dimonitoring/dipantau oleh user yang telah terdaftar pada grafana.

3.3.4 Instalasi Dan Konfigurasi

Tahapan ini sebagai awalan penerapan sistem yang dirancang dengan tujuan agar sistem dapat beroperasi dengan baik dan dapat memudahkan dalam menjalankan pengujian berserta analisis. Adapun alur instalasi dan konfigurasi dapat dilihat pada gambar 3.5.





Gambar 3. 5 Diagram Alir Instalasi dan Konfigurasi

1) Instalasi Web Server Apache2 Dan Komponen Server

Penginstalan apache2 sebagai web server dari sistem yang dibuat. Serta menginstall komponen server yang dibutuhkan seperti curl, sudo dan komponen server lainnya.

2) Instalasi Dan Konfigurasi InfluxDB

Penginstalan InfluxDB dilakukan sebagai *database* untuk mengelola data sensor yang dikirim dari miniatur stasiun cuaca secara *time series*. Pada InfluxDB juga

diperlukan konfigurasi agar data dari miniatur stasiun cuaca dan panel surya dapat disimpan pada *database* dan divisualisasikan pada grafana.

Untuk proses instalasi dan konfigurasi influxDB dapat dilihat pada lampiran 1.

3) Instalasi Dan Konfigurasi Grafana

Penginstalan grafana dilakukan untuk menampilkan *dashboard* monitoring dalam bentuk visualisasi grafis dari data sensor yang dikirim dari miniatur stasiun cuaca dan panel surya. Visualisasi *dashboard* ini berupa tabel, grafik, angka maupun tulisan yang disesuaikan dengan kebutuhan sistem yang akan dibuat.

Untuk proses instalasi dan konfigurasi grafana dapat dilihat pada lampiran 2.

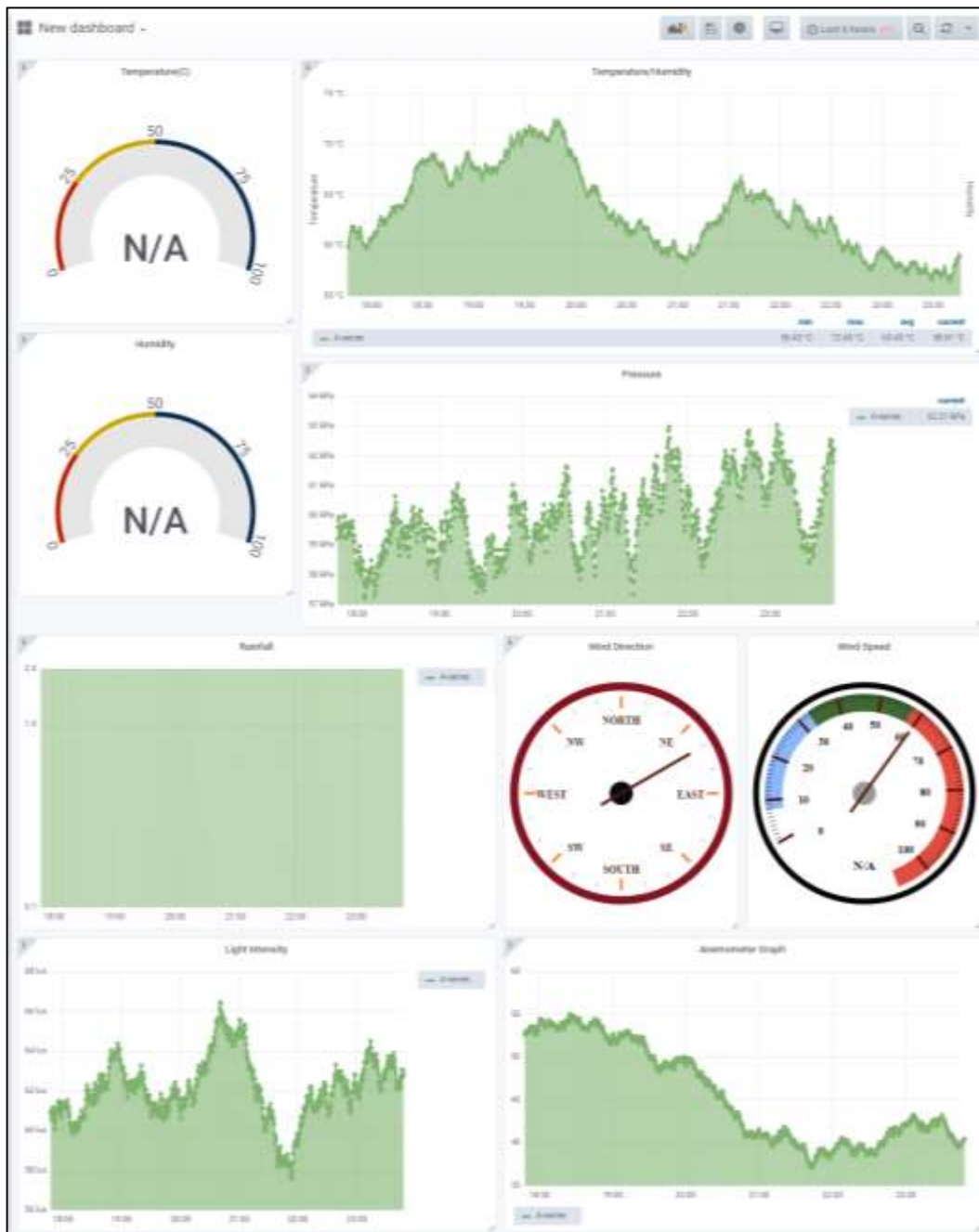
4) Mosquitto server

Penginstalan Mosquitto server dilakukan sebagai broker MQTT yang akan meneruskan data dari stasiun cuaca ke server InfluxDB, apabila data yang dikirim sesuai dengan format masukan dari server. Pada tahap ini juga diinstall paho mqtt dan juga perlu dilakukan konfigurasi user mqtt server.

Untuk proses instalasi dan konfigurasi Mosquitto server dapat dilihat pada lampiran 3.

3.3.5 Perancangan *Dashboard* dan *Query*

Pada tahap ini dilakukan perancangan *dashboard* dan *query* pada Grafana server sesuai dengan data sensor yang telah tersimpan pada server InfluxDB. Pada grafana sendiri telah menyediakan *plugin* panel yang dibutuhkan untuk visualisasi data sensor. *Plugin* tersebut dapat diinstal pada *command line* server sesuai dengan kebutuhan sistem yang dibuat. Pada Gambar 3.6 dan Gambar 3.7 merupakan desain layout *dashboard* yang akan dibuat.



Gambar 3. 6 Desain *Layout Dashboard Stasiun Cuaca*



Gambar 3. 7 Desain *Layout dashboard* Panel Surya

3.3.6 Pengujian

Tahap pengujian merupakan tahap yang dilakukan pada saat sistem telah dibuat dan dilakukan untuk memastikan sistem yang dibuat telah berjalan dengan benar. Pengujian dilakukan untuk mendapatkan hasil berupa data yang dapat digunakan untuk melakukan analisis terhadap kinerja dari sistem. Adapun pengujian yang

dilakukan yaitu pengujian penerimaan data, pengujian verifikasi data, pengujian *query*.

1) Pengujian Penerimaan dan Verifikasi Data

Pengujian penerimaan dan verifikasi data dilakukan untuk mengetahui data yang dikirim dari miniatur stasiun cuaca dan panel surya diterima dan sesuai dengan data yang diterima di server.

2) Pengujian *Query*

Pengujian *query* dilakukan untuk menguji kueri yang dirancang dapat berjalan dan menampilkan data yang sesuai dengan data pada *database* influxDB. Setiap panel pada *dashboard* grafana memiliki bentuk editor kueri yang berbeda yang dapat diatur sesuai dengan kebutuhan sistem yang akan dibuat.

3.3.7 Analisis dan Kesimpulan

Pada tahap ini dilakukan analisis terhadap kelayakan sistem. Akan dilakukan perbaikan program apabila terdapat error yang terjadi pada saat proses pengujian, serta melakukan proses perbaikan ulang jika terdapat fitur aplikasi yang belum sesuai dengan kebutuhan.

BAB IV HASIL DAN PEMBAHASAN

Hasil yang didapatkan dari penelitian ini berdasarkan rancangan, tahapan – tahapan pelaksanaan, dan pengujian yang sesuai dengan pokok permasalahan dan ruang lingkup penelitian. Penelitian ini berfokus pada perancangan *dashboard* dan *query* untuk visualisasi data sensor pada miniatur stasiun cuaca dan panel surya menggunakan InfluxDB dan Grafana. Visualisasi *dashboard* ditampilkan dalam bentuk panel *graph*, *table*, *singlestat*, *D3 gauge* maupun *text*. *Dashboard* ini berjalan secara *real time*. Hasil rancangan *dashboard* ini bisa diakses oleh user dengan menggunakan jaringan internet yang disediakan di kampus 1 Politeknik Negeri Ujung Pandang.

4.1 Implementasi

4.1.1 Konfigurasi Penerimaan Data

Data sensor pada miniatur stasiun cuaca dan panel surya dikirim/*publish* ke server menggunakan protokol MQTT dalam bentuk *message/topic*. Untuk dapat menerima data sensor yang dikirim, pada server diinstallan MQTT server yaitu mosquitto server sebagai *broker* antara server dengan miniatur stasiun cuaca dan panel surya.

MQTT broker sebagai penghubung transaksi data antara *publisher* dan *subscriber*. *Publisher* disini ialah miniatur stasiun cuaca dan panel surya sebagai pengirim pesan berupa topik. Sedangkan *subscriber* yaitu server influxDB yang melakukan *subscribe* sesuai dengan topik yang dikirim oleh *publisher* sehingga data dapat diterima oleh server InfluxDB.

Pada penelitian ini digunakan pemrograman *python* untuk merancang program menerima data sensor yang dikirim oleh miniatur stasiun cuaca dan panel surya.

Source code Program penerimaan data dapat dilihat pada lampiran 4.

```
#!/usr/bin/env python3
import paho.mqtt.client as mqtt
import requests
import StringIO
import time
from datetime import datetime
from influxdb import InfluxDBClient

# set influxdb configuration -----
dbhost = "10.0.0.205"
dbport = 8086
dbuser = "cuaca"
dbpassword = "123456"
dbname = "mydb"
#-----

# set mqtt configuration =====
mqtt_server = "10.0.0.205"
mqtt_port = 1883
mqtt_user = "cuaca"
mqtt_password = "123456"
```

Gambar 4. 1 Inisialisasi Tujuan Penerimaan Data di Server

Adapun *topic* pesan/*subscribe* yang dirancang pada server untuk penerimaan data sensor seperti pada gambar 4.2 dan Gambar 4.3. Topik *subscribe* dirancang sesuai dengan topik yang dikirim oleh *publisher* agar data yang dikirim dapat diterima dan disimpan ke server influxDB.

```
# set client subscriber -----
client.subscribe("rain/Rain")
client.subscribe("bme/Pressure")
client.subscribe("bme/Humidity")
client.subscribe("bme/Temperature_C")
client.subscribe("Wine/Value")
client.subscribe("anemometer/Counter")
client.subscribe("anemometer/RPM")
client.subscribe("anemometer/Speedwind")
```

Gambar 4. 2 Inisialisasi Client *Subscribe* Miniatur Stasiun Cuaca

```
client.subscribe("Cap/Bateray")
client.subscribe("BH1750/Light")
client.subscribe("LDR/ldr")
client.subscribe("RELAY/relay")
client.subscribe("TEMPERATUR/celcius")
client.subscribe("TEMPERATUR/fahrenheit")
client.subscribe("TEMPERATUR/kelvin")
client.subscribe("TEMPERATUR/rheamur")
client.subscribe("INA219/busvoltage_C")
client.subscribe("INA219/shuntvoltage_C")
client.subscribe("INA219/current_mA_C")
client.subscribe("INA219/power_mW_C")
client.subscribe("INA219/loadvoltage_C")
client.subscribe("INA219/busvoltage_B")
client.subscribe("INA219/shuntvoltage_B")
client.subscribe("INA219/current_mA_B")
client.subscribe("INA219/power_mW_B")
client.subscribe("INA219/loadvoltage_B")
```

Gambar 4. 3 Inisialisasi *Client Subscribe* Panel Surya

4.1.2 Data Sensor pada Server InfluxDB

Data sensor yang dikirim oleh miniatur stasiun cuaca dan panel surya disimpan ke *database* server InfluxDB yaitu **mydb**. Data yang tersimpan ini berupa waktu pengambilan data dengan *value*/nilai yang dihasilkan. Waktu penerimaan data pada influxDB otomatis disesuaikan dengan waktu pada saat data dikirim ke server. Untuk dapat mengakses/masuk ke *database* influxdb dengan perintah ***influx - precision rfc3339***.

Adapun data sensor yang tersimpan pada *database* **mydb** seperti pada gambar 4.4.

```

root@Percobaan2:/home/percobaan2# influx
Connected to http://localhost:8086 version 1.7.7
InfluxDB shell version: 1.7.7
> use mydb
Using database mydb
> show measurements
name: measurements
----
BH1750/Light
Cap/Bateray
INA219/busvoltage_B
INA219/busvoltage_C
INA219/current_mA_B
INA219/current_mA_C
INA219/loadvoltage_B
INA219/loadvoltage_C
INA219/power_mW_B
INA219/power_mW_C
INA219/shuntvoltage_B
INA219/shuntvoltage_C
LDR/ldr
RELAY/relay
TEMPERATUR/celcius
TEMPERATUR/fahrenheit
TEMPERATUR/kelvin
TEMPERATUR/rheamur
Wine/Value
anemometer/Counter
anemometer/RPM
anemometer/Speedwind
bme/Humidity
bme/Pressure
bme/Temperature_C
rain/Rain

```

Gambar 4. 4 Data *Measurements* Pada InfluxDB

pada gambar 4.4 menunjukkan tampilan *measurements* yang tersimpan pada server influxDB. Data *measurements* ini sesuai dengan topik pesan yang dikirim dari miniatur stasiun cuaca dan panel surya. Setiap *measurements* mewakili setiap data sensor yang dikirim ke server yang berisi *record* data dari masing – masing parameter.

Adapun *record* data sensor yang tersimpan pada *database* mydb pada salah satu *measurement* sebagai berikut:


```

> select * from "Wine/Value"
name: Wine/Value
time                value
-----
2019-07-21T09:35:49Z 135
2019-07-21T09:36:28Z 135
2019-07-21T09:37:54Z  90
2019-07-21T09:38:37Z  90
2019-07-21T09:40:03Z  45
2019-07-21T09:40:47Z  45
2019-07-21T09:42:13Z   0
2019-07-21T09:42:57Z 225
2019-07-21T09:44:23Z  45
2019-07-21T09:45:07Z  45
2019-07-21T09:46:33Z   0
2019-07-21T09:47:17Z  45
2019-07-21T09:48:43Z 180
2019-07-21T09:49:26Z 180
2019-07-21T09:50:53Z   0
2019-07-21T09:51:36Z  90
2019-07-21T09:53:03Z  90
2019-07-21T09:53:46Z 180

```

Gambar 4. 5 Data Sensor Kecepatan Angin pada InfluxDB

Untuk menampilkan data *min*, *max*, *last* dan lainnya dapat menggunakan perintah kueri pada *influx* seperti pada gambar 4.6.

```

> select min(*) from "Wine/Value"
name: Wine/Value
time                min_value
-----
2019-07-21T09:42:13Z 0
> select max(*) from "Wine/Value"
name: Wine/Value
time                max_value
-----
2019-07-21T13:03:44Z 315
> select mean(*) from "Wine/Value"
name: Wine/Value
time                mean_value
-----
1970-01-01T00:00:00Z 107.33369098712447
> select last(*) from "Wine/Value"
name: Wine/Value
time                last_value
-----
2019-07-23T14:40:17Z 180
> select (*) from "Wine/Value"
name: Wine/Value
time                value
-----
2019-07-21T09:35:49Z 135
2019-07-21T09:36:28Z 135

```

Gambar 4. 6 Menampilkan Nilai *Min*, *Max*, *Mean*, *Last*

4.1.3 Proses Visualisasi Data

1) Kueri Data Pada Panel

Pada penelitian ini untuk menampilkan data sensor ke *dashboard* tidak memerlukan program yang rumit dan panjang. Untuk menampilkan data dari

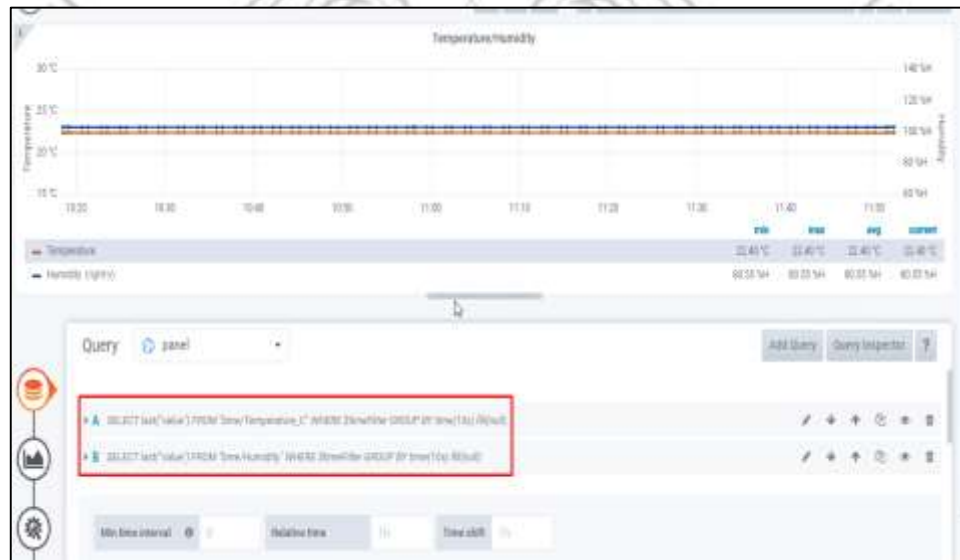
database influxDB ke *dashboard* grafana hanya diperlukan kueri data. Kueri data dilakukan pada setiap panel yang dirancang pada grafana.

Apabila panel yang diperlukan untuk merancang tidak tersedia, maka pada Grafana disediakan *plugin* yang dapat langsung diinstallkan pada *command line server*. Pada penelitian ini ada beberapa *plugin* panel yang diinstal untuk merancang *dashboard* miniatur stasiun cuaca dan panel surya yaitu *graph*, *singlestat*, *Table*, *D3 Gauge*, *bar gauge*, *text* dan *Clock*.



Gambar 4. 7 *Plugin Panel*

Adapun proses kueri data pada grafana seperti pada gambar 4.8 berikut:



a. 1 *Query Temperatur/Humidity* pada Panel *Graph*



a. 2 Pengaturan Tampilan Visualisasi Data



a. 3 Tab General

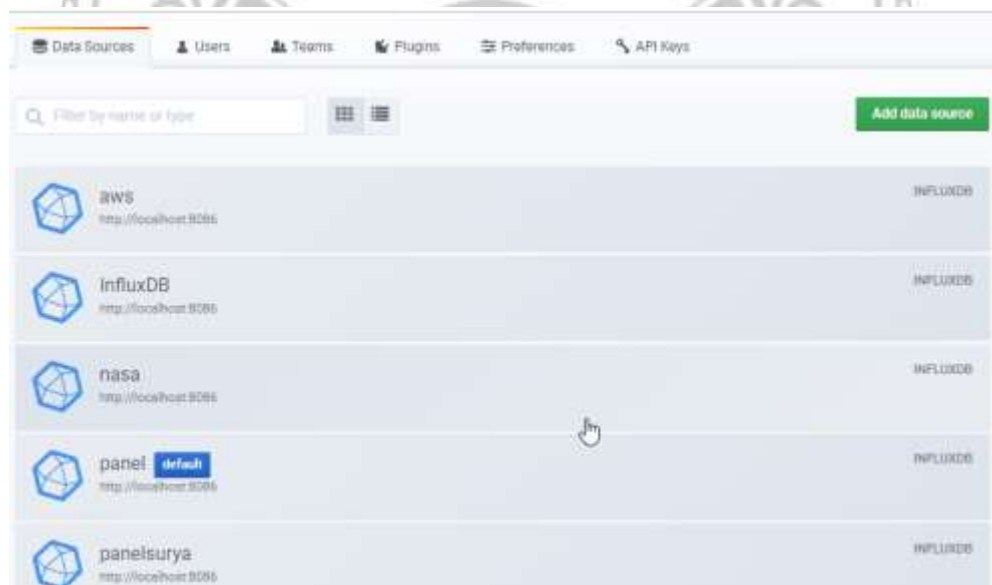
Gambar 4. 8 Query Data pada Panel Graph

Pada gambar 4.8 menunjukkan proses kueri data untuk menampilkan data sensor *humidity* dan *temperature* dengan menggunakan panel *graph*. Panel *graph* ini memiliki kueri editor yang diatur sehingga menampilkan data temperatur dan kelembaban sesuai dengan yang diinginkan. Pada a.1 menunjukkan pengaturan kueri panel, dengan memilih sumber data yang digunakan yaitu **panel**. Selanjutnya mengatur tampilan visualisasi pada a.2 seperti pengaturan *draw modes*, *mode options*, *hover tooltip*, *stacking and null value*. Lalu pada gambar a.3 untuk mengatur judul panel maupun deskripsi panel. Ukuran dari panel dapat diatur (diperbesar/diperkecil), dan panel yang telah dirancang juga dapat di *copy paste*.

Proses kueri data pada masing – masing panel dilakukan seperti pada gambar 4.8. Setiap Panel yang digunakan yang memiliki kueri editor yang berbeda yang dapat dirancang sesuai dengan kebutuhan sistem yang dibuat.

2) Mengatur *Data Source*

Data sensor yang tersimpan pada influxDB selanjutnya ditampilkan pada *dashboard* Grafana. Agar *database* InfluxDB terhubung dengan Grafana maka terlebih dahulu dilakukan pengaturan *data source* di Grafana dengan memilih InfluxDB sebagai sumber data yang menyimpan data sensor yang akan dimonitoring.



Gambar 4. 9 *Data Source*

3) Merancang *Dashboard*

Selanjutnya merancang *dashboard* visualisasi. Pada *dashboard* dapat diatur rentang waktu pengambilan data, dan juga *real time dashboard*. Hasil rancangan *dashboard* untuk visualisasi data sensor pada grafana terdapat 2 yaitu *dashboard* miniatur stasiun cuaca dan *dashboard* panel surya. *Dashboard*

visualisasi ini ditampilkan secara *real time* (berjalan setiap 5s). *Dashboard* visualisasi dapat diakses dengan mengakses alamat IP dari *controller server* <http://10.0.0.205:3000>.

a) *Dashboard* Miniatur Stasiun Cuaca



Gambar 4. 10 Tampilan *Dashboard* Miniatur Stasiun Cuaca

Dashboard miniatur stasiun cuaca menampilkan parameter cuaca seperti data suhu, kelembaban, tekanan udara dari sensor BME280, intensitas cahaya

matahari dari sensor BH1750, curah hujan dari sensor *Rain Drop*, data arah angin dari sensor *Wind Vane Direction*, data kecepatan angin dari sensor *Anemometer*.

b) *Dashboard* Panel surya



Gambar 4. 11 Tampilan *Dashboard* Panel Surya

Dashboard panel surya untuk menampilkan data sensor yaitu data arus, tegangan, daya dari sensor INA219 dihubungkan ke AKI dan Solar Panel, data relai, suhu permukaan panel dan data baterai.

4.2 Pengujian

Pengujian dilakukan sesuai dengan skema pengujian yang telah dirancang yaitu pengujian penerimaan dan verifikasi data, serta pengujian *query*.

4.2.1. Pengujian Penerimaan dan Verifikasi Data

Data sensor yang dikirim ke server terbagi menjadi 2 yaitu dari miniatur stasiun cuaca dan panel surya. Data sensor ini dikirim ke server secara *continue* terus menerus menggunakan protokol MQTT. Apabila miniatur stasiun cuaca dan panel surya tidak mengirim data ke server maka server akan menunggu hingga data dikirim ke server. Adapun proses penerimaan data sensor pada server dapat dilihat pada Gambar 4.12 dan Gambar 4.13.

```
Received a message on topic: INA219/shuntvoltage_B
2019-07-16 11:49:20: INA219/shuntvoltage_B -19.82
Finished writing to InfluxDB

Received a message on topic: INA219/loadvoltage_B
2019-07-16 11:49:20: INA219/loadvoltage_B 12.89
Finished writing to InfluxDB

Received a message on topic: INA219/current_mA_B
2019-07-16 11:49:21: INA219/current_mA_B -199.3
Finished writing to InfluxDB

Received a message on topic: INA219/power_mW_B
2019-07-16 11:49:21: INA219/power_mW_B 2572.0
Finished writing to InfluxDB

Received a message on topic: INA219/busvoltage_A
2019-07-16 11:49:22: INA219/busvoltage_A 12.84
Finished writing to InfluxDB

Received a message on topic: INA219/shuntvoltage_A
2019-07-16 11:49:22: INA219/shuntvoltage_A -0.19
Finished writing to InfluxDB

Received a message on topic: INA219/loadvoltage_A
2019-07-16 11:49:23: INA219/loadvoltage_A 12.84
Finished writing to InfluxDB

Received a message on topic: INA219/current_mA_A
2019-07-16 11:49:23: INA219/current_mA_A -1.9
Finished writing to InfluxDB

Received a message on topic: INA219/power_mW_A
2019-07-16 11:49:24: INA219/power_mW_A 16.0
Finished writing to InfluxDB
```

Gambar 4. 12 Proses Penerimaan Data Miniatur Stasiun Cuaca


```
percobaan2@Percobaan2: ~
-----
Received a message on topic: rain/Rain
2019-07-28 13:23:44: rain/Rain 1.0
Finished writing to InfluxDB
-----
Received a message on topic: Wine/Value
2019-07-28 13:24:01: Wine/Value 50.0
Finished writing to InfluxDB
-----
Received a message on topic: anemometer/Counter
2019-07-28 13:24:16: anemometer/Counter 456.0
Finished writing to InfluxDB
-----
Received a message on topic: anemometer/RPM
2019-07-28 13:24:16: anemometer/RPM 120.0
Finished writing to InfluxDB
-----
Received a message on topic: anemometer/Speedwind
2019-07-28 13:24:16: anemometer/Speedwind 1.01
Finished writing to InfluxDB
-----
Received a message on topic: bme/Pressure
2019-07-28 13:24:27: bme/Pressure 777.69
Finished writing to InfluxDB
-----
Received a message on topic: bme/Humidity
2019-07-28 13:24:27: bme/Humidity 80.55
Finished writing to InfluxDB
-----
Received a message on topic: bme/Temperature_C
2019-07-28 13:24:27: bme/Temperature_C 32.4
Finished writing to InfluxDB
-----
Received a message on topic: rain/Rain
2019-07-28 13:24:27: rain/Rain 1.0
Finished writing to InfluxDB
```

Gambar 4. 13 Proses Penerimaan Data Panel Surya

Berdasarkan gambar 4.12 dan gambar 4.13 menunjukkan bahwa, data sensor yang dikirim pada miniatur stasiun cuaca dan panel surya menggunakan protokol MQTT berhasil diterima di server. Data sensor yang telah diterima oleh server kemudian di *impor* ke *database* yang telah disediakan yaitu *mydb*. Data sensor yang tersimpan pada *database* ini, sesuai dengan data sensor yang dikirim dari miniatur stasiun cuaca dan panel surya.

Adapun data sensor yang dikirim dan diterima di server seperti pada tabel 4.1 dan tabel 4.2.

Tabel 4. 1 Data Sensor Miniatur Stasiun Cuaca Yang Dikirim Dan Diterima Server

Data ke-	Data Yang Dikirim						Data Yang Diterima						Keterangan
	Temp °C	Hum %H	Pressure Pa	Rain	Wind direction	Wind speed	Temp °C	Hum %H	Pressure Pa	Rain	Wind direction	Wind speed	
1	-140.04	80.55	777.64	0	135	3.42	-	80.55	777.64	0	135	3.42	Ok
2	-140.04	80.55	777.64	0	45	40.31	-	80.55	777.64	0	45	40.31	Ok
3	22.40	80.55	777.64	1	90	46.04	22.40	80.55	777.64	1	90	46.04	Ok
4	22.40	80.55	777.64	1	45	38.34	22.40	80.55	777.64	1	45	38.34	Ok
5	22.40	80.55	777.64	0	135	23.27	22.40	80.55	777.64	0	135	23.27	Ok
6	22.40	80.55	777.64	0	90	41.18	22.40	80.55	777.64	0	90	41.18	Ok
7	22.40	100	1255	1	315	46.14	22.40	100	1255	1	315	46.14	Ok
8	22.40	60.61	588.24	1	90	51.39	22.40	60.61	588.24	1	90	51.39	Ok
9	22.40	60.61	1255	1	270	42.59	22.40	60.61	1255	1	270	42.59	Ok
10	22.40	80.55	777.64	1	270	15.78	22.40	80.55	777.64	1	270	15.78	Ok
11	22.40	80.55	777.64	1	180	50.35	22.40	80.55	777.64	1	180	50.35	Ok
12	25.05	80.55	1008.68	1	180	0	25.05	80.55	1008.68	1	180	0	Ok
13	22.40	80.55	777.64	1	180	32.56	22.40	80.55	777.64	1	180	32.56	Ok
14	22.40	80.55	777.64	1	225	31.23	22.40	80.55	777.64	1	225	31.23	Ok
15	22.40	80.55	777.64	1	225	43.08	22.40	80.55	777.64	1	225	43.08	Ok

Tabel 4. 2 Data Sensor Panel Surya Yang Dikirim Dan Diterima Di Server

Data ke-	Data Yang Dikirim						Data Yang Diterima						Keterangan
	Temp °C	Power B	Shunt voltage	Bus voltage	Load voltage	light	Temp °C	Power	Shunt voltage	Bus voltage	Load voltage	Light	
1	24.5	3532	-21.13	12.62	12.6	20.83	24.5	3532	-21.13	12.62	12.6	20.83	Ok
2	24.25	2688	-21.36	12.62	12.6	20.83	24.25	2688	-21.36	12.62	12.6	20.83	Ok
3	24.12	2674	-21.05	12.61	12.59	22.5	24.12	2674	-21.05	12.61	12.59	22.5	Ok
4	23.94	2690	-21.04	11.62	12.59	25	23.94	2690	-21.04	11.62	12.59	25	Ok
5	23.87	2684	-21.37	12.61	12.61	21.67	23.87	2684	-21.37	12.61	12.61	21.67	Ok
6	23.75	2674	-21.22	12.63	12.6	22.5	23.75	2674	-21.22	12.63	12.6	22.5	Ok
7	23.81	2690	-21.15	12.62	12.61	30	23.81	2690	-21.15	12.62	12.61	30	Ok
8	23.94	2684	-21.38	12.64	12.6	27.5	23.94	2684	-21.38	12.64	12.6	27.5	Ok
9	24	2670	-21.17	12.62	12.63	26.67	24	2670	-21.17	12.62	12.63	26.67	Ok
10	24.06	2660	-21.3	12.65	12.61	25	24.06	2660	-21.3	12.65	12.61	25	Ok
11	24.12	2804	-21.19	12.63	12.61	24.17	24.12	2804	-21.19	12.63	12.61	24.17	Ok
12	24.25	2696	-21.17	12.64	12.6	30.83	24.25	2696	-21.17	12.64	12.6	30.83	Ok
13	24.31	2694	-21.92	12.62	12.6	30.83	24.31	2694	-21.92	12.62	12.6	30.83	Ok
14	24.5	2670	-21.1	12.62	12.59	30.83	24.5	2670	-21.1	12.62	12.59	30.83	Ok
15	24.5	2688	-21.37	12.62	12.59	20.83	24.5	2688	-21.37	12.62	12.59	20.83	Ok

Pada tabel 4.1 terdapat data yang tidak diterima di server yaitu pada data *temperature*. Pada miniatur stasiun cuaca, sebelum data sensor dikirim ke server terlebih dahulu difilter untuk validasi data yang salah. Data -140.00 merupakan data tidak valid. Oleh karena itu data tersebut tidak terbaca pada server. Sebelum data dikirim ke server, data terlebih dahulu disimpan ke *USB storage* sebagai data *logger* atau penyimpanan sementara. Jadi waktu pengambilan data dengan waktu penerimaan data pada server memiliki rentang waktu yang berbeda.

Berdasarkan hasil pengujian pada tabel 4.1 dan tabel 4.3 dapat disimpulkan bahwa data sensor yang dikirim oleh miniatur stasiun cuaca baik dari miniatur stasiun cuaca maupun dari panel surya sesuai dengan yang diterima di server.

Untuk tampilan data yang tersimpan pada server InfluxDB serta tampilan visualisasi pada grafana dapat dilihat pada lampiran 5.

4.2.2. Pengujian *Query* Data

Setiap panel yang disediakan oleh grafana memiliki *query* editor yang berbeda yang dapat diatur sesuai dengan kebutuhan sistem yang dibuat.

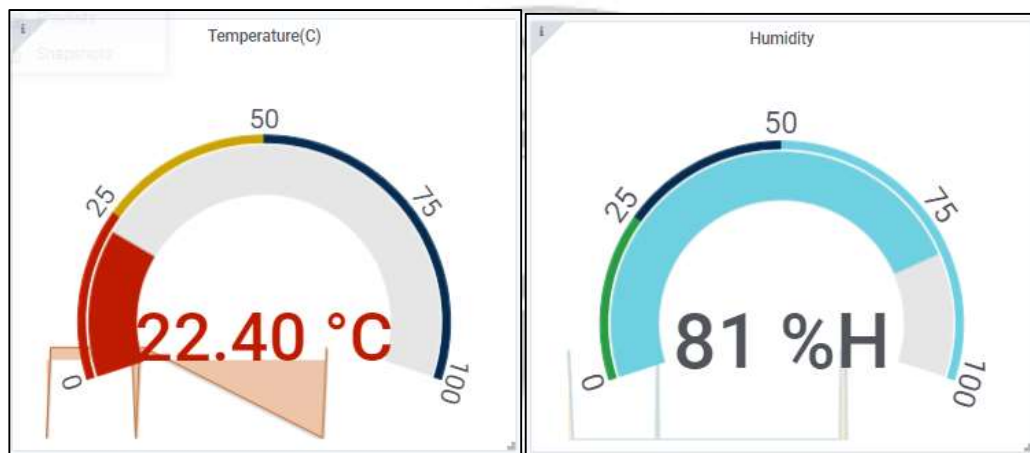
1) Pengujian *query* Data Sensor BME280

Sensor BME280 digunakan untuk mengukur parameter cuaca *temperature*, *humidity* dan *pressure*. Data *temperature* dan *humidity* ditampilkan dengan menggunakan panel *singlestat* pada grafana. Panel *singlestat* ini memiliki kueri editor yang dirancang agar dapat menampilkan data sesuai dengan data pada influxDB. Berikut merupakan pengujian kueri untuk data sensor BME280:

Tabel 4. 3 Data Sensor BME280 Pada InfluxDB

Sensor BME280	Current	MIN	MAX	MEAN
<i>Temperature</i>	22.4	22.4	22.4	7.91
<i>Humidity</i>	80.55	80.55	100	82.28
<i>Pressure</i>	777.64	588.24	1255	772.77

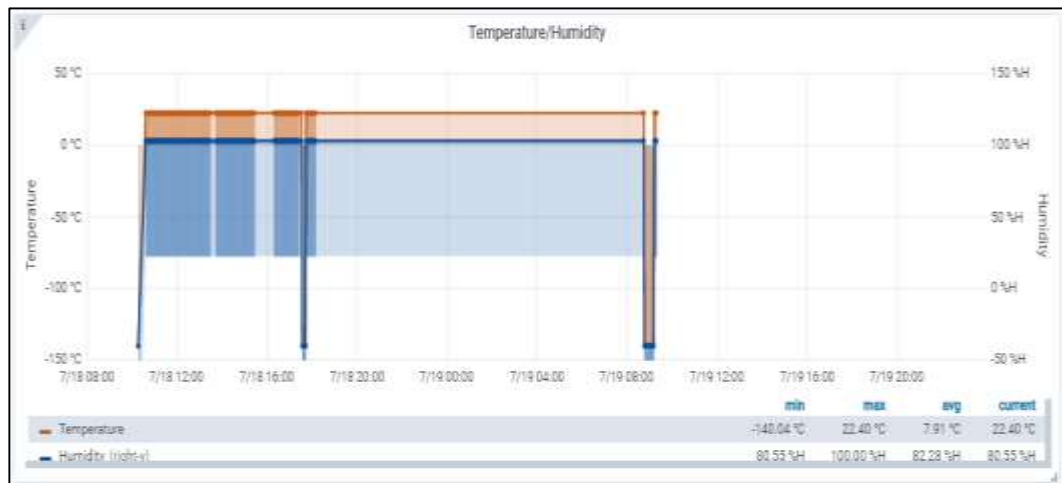
Berikut merupakan visualisasi data sensor BME280 pada *dashboard* grafana:



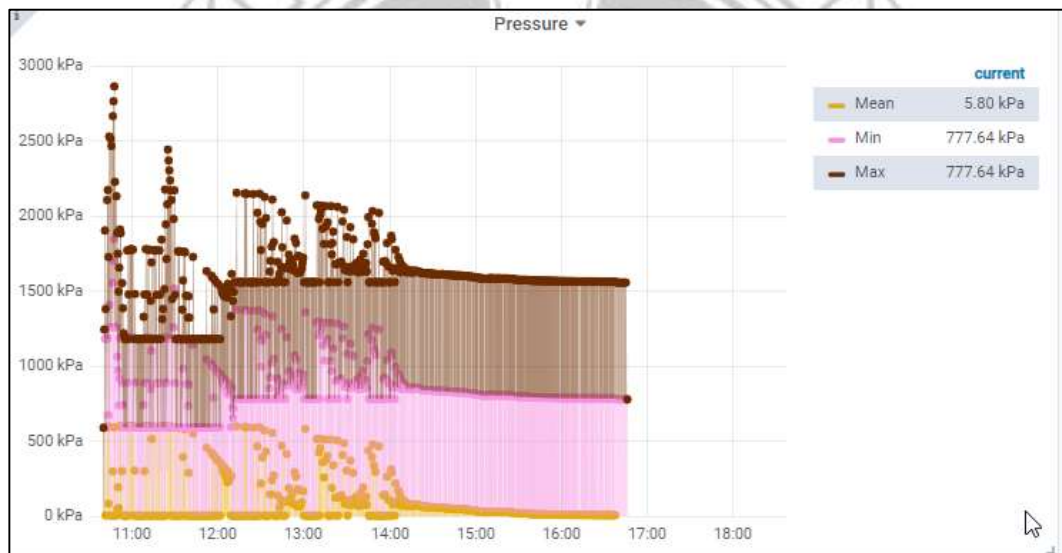
Gambar 4. 14 Visualisasi Data Suhu dan Kelembaban

Pada Gambar 4.14 menunjukkan visualisasi data suhu dan kelembaban. Data suhu menampilkan nilai 22.40°C yang merupakan data terbaru yang diterima dan tersimpan pada server influxDB. Begitupun data kelembaban. Perancangan kueri untuk data suhu dan kelembaban dengan panel *singlestat* menampilkan data sesuai dengan tabel 4.4 yang merupakan data dari influxDB.

Berikut merupakan grafik perubahan suhu dan kelembaban secara *time series*. Pada panel *temperature/humidity* menampilkan nilai *min*, *max*, *average*, dan *current*.



Gambar 4. 15 Grafik Data Suhu dan Kelembaban



Gambar 4. 16 Visualisasi Data Tekanan Udara

Pada gambar 4.16 merupakan grafik tekanan udara. Pada grafik menunjukkan perubahan tekanan udara yang terjadi, serta data mengenai nilai *min*, *max*, *mean*, dan *current*. Data yang ditampilkan pada panel *graph* ini sesuai dengan tabel 4.4 yang merupakan data pada *database* influxDB.

Berdasarkan hasil visualisasi data sensor BME280 diatas, menunjukkan bahwa perancangan kueri untuk data sensor BME280 pada grafana dengan

menggunakan panel *Singlestat* dan *graph*, dapat menampilkan data yang sesuai dengan data pada *database* InfluxDB.

2) Data Sensor *Wind Vane Direction*

Sensor *Wind vane direction* digunakan untuk mengukur parameter arah angin.

Panel yang digunakan untuk menampilkan data arah angin pada grafana yaitu *panel D3 Gauge*. Adapun keterangan data arah angin pada tabel 4.4.

Tabel 4. 4 Keterangan Data Arah Angin

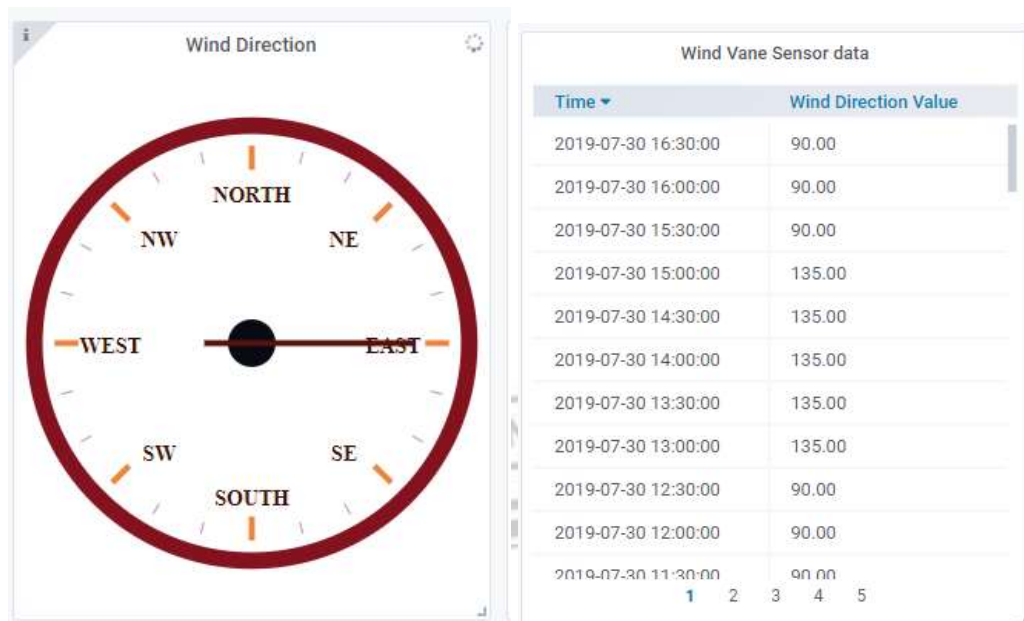
Value	Keterangan
0 ⁰	<i>North</i>
45 ⁰	<i>NE(North East)</i>
90 ⁰	<i>East</i>
135 ⁰	<i>SE(South East)</i>
180 ⁰	<i>South</i>
225 ⁰	<i>SW(South West)</i>
270 ⁰	<i>West</i>
315 ⁰	<i>NW(North West)</i>

Data arah angin diukur menggunakan sensor *Wind Vane Direction*. Adapun data arah angin yang tersimpan pada *database* influxDB seperti pada tabel 4.5 berikut:

Tabel 4. 5 Data *Wind Vane Direction*

Time	Current
2019-07-30T15:55:41Z	90 ⁰
2019-07-30T16:00:00Z	90 ⁰
2019-07-30T16:20:08Z	90 ⁰
2019-07-30T16:30:13Z	90 ⁰

Adapun tampilan visualisasi data arah angin pada grafana dengan menggunakan panel *D3 Gauge* seperti pada Gambar 4.17.



Gambar 4. 17 Visualisasi *Wind Direction*

Pada Gambar 4.17 menunjukkan tampilan visualisasi dari data arah angin. Data yang ditampilkan pada panel *wind direction* yaitu 90° menunjukkan arah timur/*EAST*. Data ini sesuai dengan data pada tabel 4.5.

Berdasarkan hasil pengujian pada tabel 4.5 dan Gambar 4.16, dapat disimpulkan bahwa data arah angin yaitu "*Wine/value*" yang ditampilkan pada panel *D3 Gauge* sesuai dengan data yang tersimpan pada *database influxDB*.

3) Data *Power*

Data *power* diukur menggunakan sensor INA219 yang dihubungkan ke AKI maupun PANEL. Data *power* ini ditampilkan dengan menggunakan panel *graph*. Adapun nilai *min*, *max*, *current* dan *mean* dari data *power* yang tersimpan pada *database influxDB* dapat dilihat pada Tabel 4.7.

Tabel 4. 6 Data Sensor *Power* Pada InfluxDB (mW)

Data Power	Min	Max	Mean	Current
AKI	4	46210	2138.42	1712
PANEL	0	47950	1599.03	66

Berikut merupakan tampilan visualisasi data *Power* pada *dashboard* grafana:



Gambar 4. 18 Visualisasi Data *Power*

Pada Gambar 4.18 dapat dilihat nilai *min*, *max* dan *mean* dari data *power*. Nilai *min*, *max*, *mean* dan *current* yang ditampilkan pada panel *graph* grafana sesuai dengan data pada tabel 4.6.

Berikut ini hasil pengujian kueri berdasarkan pengujian kueri pada data sensor BME280, data *wind vane* dan data *Power* yaitu

Tabel 4. 7 Hasil Pengujian *Query*

Data Sensor	Data pada InfluxDB	Jenis Panel	Data yang Ditampilkan pada <i>Dashboard</i>
<i>Temperature</i>	22.4	<i>Singlestat</i>	22.40°C
<i>Humidity</i>	80.55	<i>Singlestat</i>	81%H
<i>Pressure</i>	777.64	<i>Graph</i>	777.64
<i>Wind direction</i>	90	<i>D3 Gauge</i>	<i>East</i>
<i>Power AKI</i>	66 mW	<i>Graph</i>	66 mW

Berdasarkan hasil pengujian pada Tabel 4.7 menunjukkan bahwa perancangan kueri pada masing – masing panel yaitu panel *singlestat*, *graph*, dan *D3 gauge* mampu menampilkan informasi cuaca sesuai dengan data yang tersimpan di influxDB yaitu pada *database mydb*. Nilai yang ditampilkan dari masing – masing panel dapat berubah apabila miniatur stasiun cuaca dan panel surya mengirim data ke server. Hal ini disebabkan *dashboard* berjalan secara *real time*.

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari proses perancangan, implementasi, serta pengujian dapat ditarik kesimpulan yaitu:

Data sensor yang dikirim pada miniatur stasiun cuaca dan panel surya menggunakan protokol MQTT berhasil diterima di server dan disimpan ke *database* influxDB. Data *measurements* yang tersimpan pada *database* influxDB sesuai dengan topik pesan yang dikirim oleh miniatur stasiun cuaca dan panel surya.

Untuk menampilkan data sensor dari *database* ke *dashboard*, dilakukan kueri data pada setiap panel yang digunakan. Data sensor divisualisasikan dalam bentuk panel *graph*, *d3 gauge*, *singlestat*, *table* maupun *text*. Setiap bentuk panel ini memiliki *query editor* yang diatur sehingga menampilkan informasi yang menarik dan mudah dipahami. Hasil rancangan *dashboard* dan *query* miniatur stasiun cuaca dan panel surya ini, mampu menampilkan informasi mengenai parameter cuaca dalam bentuk visualisasi grafis secara *real time*.

5.2 Saran

Adapun saran untuk pengembangan selanjutnya yaitu:

1. Memanfaatkan fitur – fitur yang disediakan oleh Grafana.
2. Memanfaatkan *platform* yang disediakan InfluxData yaitu TICK STACK untuk monitoring data sensor.

DAFTAR PUSTAKA

- Dix, P. (2016) "Why Time Series Matters for Metrics Real Time and Sensor Data," *InfluxData Whitepaper*, hal. 1–13.
- Eclipse Fondation (2019) *mqtt — Message Queue Telemetry Transport*. Tersedia pada: <https://mosquitto.org/man/mqtt-7.html> (Diakses: 3 September 2019).
- Few, S. (2006) *Information Dashboard Design: The Effective Visual Communication of Data, The effective visual communication of data Sebastopol*. doi: 10.1017/S0021849904040334.
- Grafana (2019) *Welcome to the Grafana Documentation*. Tersedia pada: <https://grafana.com/docs/v4.3/> (Diakses: 22 Februari 2019).
- Influxdata (2018a) "Benchmarking InfluxDB vs. Cassandra for Time Series Data & Metrics Management," *Influxdata Technical Paper*, hal. 1–15. Tersedia pada: http://get.influxdata.com/rs/972-GDU-533/images/InfluxDB_vs_Cassandra.pdf.
- Influxdata (2018b) *InfluxDB is the Time Series Database in the TICK Stack*. Tersedia pada: <https://www.influxdata.com/time-series-platform/influxdb/> (Diakses: 12 November 2018).
- Krisnandi, D. (2011) "Perancangan dan Analisa Output Rangkaian Signal Conditioning Analog Melalui Mikrokontroler ATMega8535 Untuk Stasiun Cuaca," *Inkom: Jurnal Informatika, Sistem Kendali dan Komputer*, hal. 22–28.
- MQTT (2019) *FAQ - Frequently Asked Questions | Message Queuing Telemetry Transport (MQTT)*. Tersedia pada: <https://mqtt.org/faq> (Diakses: 20 Juli 2019).
- Naqvi, S. N. Z. dan YFantidou, S. (2017) "Time Series Databases and InfluxDB," *Universit'e Libre De Bruxelles*. Tersedia pada: https://cs.ulb.ac.be/public/_media/teaching/influxdb_2017.pdf.
- Nurbaitil, S. dan Cahyati, A. (2018) *Rancang Bangun Miniatur Stasiun Cuaca*. Universitas Hasanuddin.
- Sadiku, M. N. O. dkk. (2016) "Data visualization," *International Journal of Engineering Research And Advanced Technology(IJERAT)*, hal. 11–16. Tersedia pada: https://www.researchgate.net/publication/311597028_DATA_VISUALIZATION.
- Satria, G. O., Satrya, G. B. dan Herutomo, A. (2015) "Implementasi Protokol

MQTT Pada Smart Building Berbasis OpenMTC,” *e-Proceeding of Engineering*, hal. 6530–6537.

Sucipto, W., Hartawan, D. D. dan Setiawan, W. (2017) “Rancang Bangun Perangkat Pemantau Cuaca Otomatis berbasis Mikrokontroller pada Jaringan WLAN IEEE 802.11b,” *E-Journal SPEKTRUM*, hal. 48–55.

Summerfiel, M. (2010) *Programming in Python 3: A Complete Introduction to the Python Language*. United States of America: Pearson Education Inc.

Thalmann, D. (2013) “An interactive, discipline-independent data visualization system,” *Computers in Physics*, Vol.14(3), hal. 277–290. doi: 10.1002/spe.4380140307.

Wardani, I. K. (2011) “Manfaat Prediksi Cuaca Jangka Pendek Berdasarkan Data Radiosonde Dan Numerical Weather Prediction (NWP) Untuk Pertanian Daerah,” *Jurnal Elektronik Unipdu*, Vol(1). Tersedia pada: <http://www.journal.unipdu.ac.id/index.php/seminas/article/view/42>.

Yahwe, C. P., Isnawaty dan Aksara, L. . F. (2016) “Rancang Bangun Prototipe System Monitoring Kelembaban Tanah Melalui SMS berdasarkan Hasil Penyiraman Tanaman,” *semanTIK*, hal. 97–110.

Yong, G. K. (2019) *A Data Analytic Module To Extend Grafana Functionality*. Universiti Tunku Abdul Rahman.

**L
A
M
P
I
R
A
N**



Lampiran 1 Instalasi Dan Konfigurasi InfluxDB

1. Menambahkan *repository* influxdata:

```
root@CTSyam:/home/syam# curl -sL https://repos.influxdata.com/influxdb.key | apt
-key add -
OK
root@CTSyam:/home/syam# source /etc/os-release
root@CTSyam:/home/syam# test $VERSION_ID = "8" && echo "deb https://repos.influx
data.com/debian jessie stable" | tee /etc/apt/sources.list.d/influxdb.list
deb https://repos.influxdata.com/debian jessie stable
```

2. Install InfluxDB

```
root@CTSyam:/home/syam# apt-get update && apt-get install influxdb
Hit https://repos.influxdata.com jessie InRelease
Hit https://repos.influxdata.com jessie/stable amd64 Packages
Get:1 https://repos.influxdata.com jessie/stable Translation-en [151 B]
Get:2 https://repos.influxdata.com jessie/stable Translation-en [151 B]
Get:3 https://repos.influxdata.com jessie/stable Translation-en [151 B]
Get:4 https://repos.influxdata.com jessie/stable Translation-en [151 B]
Get:5 https://repos.influxdata.com jessie/stable Translation-en [151 B]
Ign https://repos.influxdata.com jessie/stable Translation-en
```

3. Konfigurasi influxDB

```
> root@Percobaan2:/home/percobaan2# influxd -config /etc/influxdb/influxdb.conf
88888888      .d888 888      88888888b. 8888888b.
888      d88P" 888      888  "Y88b 888  "88b
888      888 888 888      888 888 888 888 888 888 888 888 888 888 888 .88P
888 888888b. 8888888 888 888 888 888 888 888 888 888888888K.
888 888 "88b 888 888 888 888 Y8bd8P' 888 888 888 "Y88b
888 888 888 888 888 888 888 X88K 888 888 888 888
888 888 888 888 888 Y88b 888 .d8""8b. 888 .d88P 888 d88P
88888888 888 888 888 888 "Y88888 888 888 88888888P" 88888888P"

2019-07-20T04:22:10.118814Z      info      InfluxDB starting      {"log_id": "0GkPh5mW000", "versio
n": "1.7.7", "branch": "1.7", "commit": "f8fdf652f348fc9980997felc972e2b79ddd13b0"}
2019-07-20T04:22:10.118878Z      info      Go runtime      {"log_id": "0GkPh5mW000", "version": "gol
.11", "maxprocs": 1}
run: open server: listen: listen tcp 127.0.0.1:8088: bind: address already in use
```

- Menambahkan variabel INFLUXDB_CONFIG_PATH ke dalam environment

```
GNU nano 2.2.6      File: /etc/environment      Modified
INFLUXDB_CONFIG_PATH=/etc/influxdb/influxdb.conf
```

- Konfigurasi InfluxDB `nano /etc/influxdb/influxdb.conf`

```
[monitor]
# Whether to record statistics internally.
store-enabled = true

# The destination database for recorded statistics
store-database = "_internal"

# The interval at which to record statistics
store-interval = "10s"
```

```
[http]
# Determines whether HTTP endpoint is enabled.
enabled = true

# Determines whether the Flux query endpoint is enabled.
# flux-enabled = false

# Determines whether the Flux query logging is enabled.
# flux-log-enabled = false

# The bind address used by the HTTP service.
# bind-address = ":8086"

# Determines whether user authentication is enabled over HTTP/HTTPS.
auth-enabled = false

# The default realm sent back when issuing a basic auth challenge.
# realm = "InfluxDB"

# Determines whether HTTP request logging is enabled.
log-enabled = true
```

4. Menjalankan influxDB

```
root@CTSyam:/home/syam# service influxdb start
Starting influxdb...
influxdb process was started [ OK ]
```

5. Uji influxDB

```
root@Percobaan2:/home/percobaan2# influx
Connected to http://localhost:8086 version 1.7.7
InfluxDB shell version: 1.7.7
```

6. Membuat *database*

```
> create database mydb
```

```
> show databases
name: databases
name
----
_internal
influxdb
stasiun cuaca
iot
telegraf
aws
mydb
```

7. Membuat *users*

```
> CREATE USER "cuaca" WITH PASSWORD '123456' WITH ALL PRIVILEGES
```

```
> show users
user  admin
----  ----
cuaca true
titin false
```



Lampiran 2 Instalasi Dan Konfigurasi Grafana

1. Install *Stable*

```
root@CTSyam:/home/syam# wget https://dl.grafana.com/oss/release/grafana_5.4.2_amd64.deb
root@CTSyam:/home/syam# apt-get install -y adduser libfontconfig1
Reading package lists... Done
Building dependency tree
Reading state information... Done
adduser is already the newest version.
root@CTSyam:/home/syam# dpkg -i grafana_5.4.2_amd64.deb
Selecting previously unselected package grafana.
(Reading database ... 28317 files and directories currently installed.)
Preparing to unpack grafana_5.4.2_amd64.deb ...
Unpacking grafana (5.4.2) ...
Setting up grafana (5.4.2) ...
Adding system user `grafana' (UID 104) ...
Adding new user `grafana' (UID 104) with group `grafana' ...
```

2. Menambahkan *repository*

```
root@CTSyam:/home/syam# sudo apt-get install -y adduser libfontconfig1
root@CTSyam:/home/syam# add-apt-repository "deb https://packages.grafana.com/oss/deb stable main"
root@CTSyam:/home/syam# add-apt-repository "deb https://packages.grafana.com/oss/deb beta main"
root@CTSyam:/home/syam# wget -q -O - https://packages.grafana.com/gpg.key | apt-key add -
OK
```

3. Install grafana

```
root@CTSyam:/home/syam# apt-get install grafana
Reading package lists... Done
Building dependency tree
Reading state information... Done
grafana is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 73 not upgraded.
```

4. Menjalankan grafana

```
root@CTSyam:/home/syam# service grafana-server start
[ ok ] Starting Grafana Server:..
root@CTSyam:/home/syam# service grafana-server status
[ ok ] grafana is running.
```

5. Install paket *apt-transport-https* yang diperlukan untuk mengambil paket melalui HTTPS.

```
root@CTSyam:/home/syam# apt-get install -y apt-transport-https
Reading package lists... Done
Building dependency tree
Reading state information... Done
apt-transport-https is already the newest version.
```


Lampiran 3 Instalasi Dan Konfigurasi MQTT Server

1. Menambahkan *repostiry* mosquitto

```
root@CTSyam:/home/syam# wget http://repo.mosquitto.org/debian/mosquitto-repo.gpg
.key
root@CTSyam:/home/syam# apt-key add mosquitto-repo.gpg.key
OK
root@CTSyam:/home/syam# cd /etc/apt/sources.list.d/
root@CTSyam:/etc/apt/sources.list.d# wget http://repo.mosquitto.org/debian/mosqu
itto-jessie.list
```

2. Install Mosquitto Server dan Client

```
root@CTSyam:/home/syam# apt-get install mosquitto
Reading package lists... Done
Building dependency tree
Reading state information... Done
root@CTSyam:/home/syam# apt-get install mosquitto-clients
Reading package lists... Done
```

3. Cek mosquitto

```
root@CTSyam:/home/syam# service mosquitto status
[ ok ] mosquitto is running.
```

4. Menjalankan mosquitto

```
root@CTSyam:/home/syam# service mosquitto start
[ ok ] Starting network daemon:: mosquitto.
root@CTSyam:/home/syam# service mosquitto stop
[ ok ] Stopping network daemon:: mosquitto.
root@CTSyam:/home/syam# service mosquitto restart
[ ok ] Restarting network daemon:: mosquitto.
```

5. Testing komunikasi mosquitto

```
root@CTSyam:/home/syam# mosquitto_pub -d -t check_aja -m "hai ini dari publisher
"
Client mosqpub|5856-CTSyam sending CONNECT
Client mosqpub|5856-CTSyam received CONNACK (0)
Client mosqpub|5856-CTSyam sending PUBLISH (d0, q0, r0, m1, 'check_aja', ... (22
bytes))
Client mosqpub|5856-CTSyam sending DISCONNECT
root@CTSyam:/home/syam# mosquitto_sub -d -t check_aja
Client mosqsub|5854-CTSyam sending CONNECT
Client mosqsub|5854-CTSyam received CONNACK (0)
Client mosqsub|5854-CTSyam sending SUBSCRIBE (Mid: 1, Topic: check_aja, QoS: 0)
Client mosqsub|5854-CTSyam received SUBACK
Subscribed (mid: 1): 0
Client mosqsub|5854-CTSyam received PUBLISH (d0, q0, r0, m0, 'check_aja', ... (2
2 bytes))
hai ini dari publisher
Client mosqsub|5854-CTSyam sending PINGREQ
Client mosqsub|5854-CTSyam received PINGRESP
```

6. Membuat user dan password

```
root@CTSyam:/home/syam# mosquitto_passwd -c 123456 cuaca
Password:
Reenter password:
root@CTSyam:/home/syam# █
```

7. Install paho mqtt, core, python

```
root@CTSyam:/home/syam# pip install paho-mqtt
```

```
root@CTSyam:/home/syam# apt-get install core
Reading package lists... Done
```

```
root@CTSyam:/home/syam# git clone https://github.com/eclipse/paho.mqtt.python
```

```
root@CTSyam:/home/syam# cd paho.mqtt.python
root@CTSyam:/home/syam/paho.mqtt.python# python setup.py install
```



Lampiran 4: *Source Code* Program Penerimaan Data

```
#!/usr/bin/env python3
import paho.mqtt.client as mqtt
import requests
import StringIO
import time
from datetime import datetime
from influxDB import InfluxDBClient

# set influxDB configuration -----
dbhost = "10.0.0.205"
dbport = 8086
dbuser = "cuaca"
dbpassword = "123456"
dbname = "mydb"
# -----

# set mqtt configuration =====
mqtt_server = "10.0.0.205"
mqtt_port = 1883
mqtt_user = "cuaca"
mqtt_password = "123456"
# =====

def on_connect(client, userdata, flags, rc):
print("Connected with result code "+str(rc))
# set client subscriber -----
client.subscribe("bme/humidity")
client.subscribe("bme/temperature_F")
client.subscribe("bme/temperature_C")
client.subscribe("bme/pressure")
client.subscribe("rainsensor/rainfall")
client.subscribe("Wind/wind_direction")
client.subscribe("Wind/wind_speed")

# panel surya-----
client.subscribe("Cap/Bateray")
```

```

client.subscribe("RELAY/Relay")
client.subscribe("BH1750/Light")
client.subscribe("TEMPERATUR/celcius")
client.subscribe("INA219/busvoltage_C")
client.subscribe("INA219/shuntvoltage_C")
client.subscribe("INA219/current_mA_C")
client.subscribe("INA219/power_mW_C")
client.subscribe("INA219/loadvoltage_C")
client.subscribe("INA219/busvoltage_B")
client.subscribe("INA219/shuntvoltage_B")
client.subscribe("INA219/current_mA_B")
client.subscribe("INA219/power_mW_B")
client.subscribe("INA219/loadvoltage_B")

#-----
def on_message(client, userdata, msg):
    print("Received a message on topic: " + msg.topic)
# Use utc as timestamp
    now = datetime.now()
    receiveTime = now.strftime("%Y-%m-%d %H:%M:%S")
#receiveTime=datetime.datetime.utcnow()
    message=msg.payload.decode("utf-8")
    isfloatValue=False
    try:
        # Convert the string to a float so that it is stored as
        # a number and not a string in the database
        val = float(message)
        isfloatValue=True
    except:
        print("Could not convert " + message + " to a float
        value")
        isfloatValue=False
    if isfloatValue:
        print(str(receiveTime) + ": " + msg.topic + " " +
        str(val))

    json_body = [
        {

```

```

        "measurement": msg.topic,
        "time": str(receiveTime),
        "fields": {
            "value": val
        }
    ]
    dbclient.write_points(json_body)
    print("Finished writing to InfluxDB")
    print ("=====")
    client.publish("demo")
# Set up a client for InfluxDB
    dbclient = InfluxDBClient(dbhost, dbport, dbuser, dbpassword,
    dbname)
# Initialize the MQTT client that should connect to the
Mosquitto broker
    client = mqtt.Client()
    client.on_connect = on_connect
    client.on_message = on_message
    connOK=False

while(connOK == False):
    try:
        client.username_pw_set(mqtt_user, mqtt_password)
        client.connect(mqtt_server, mqtt_port, 60)
        connOK = True
    except:
        connOK = False
    time.sleep(2)

# Blocking loop to the Mosquitto broker
client.loop_forever()

```

Lampiran 5: Data Sensor Miniatur Stasiun Cuaca Pada Server

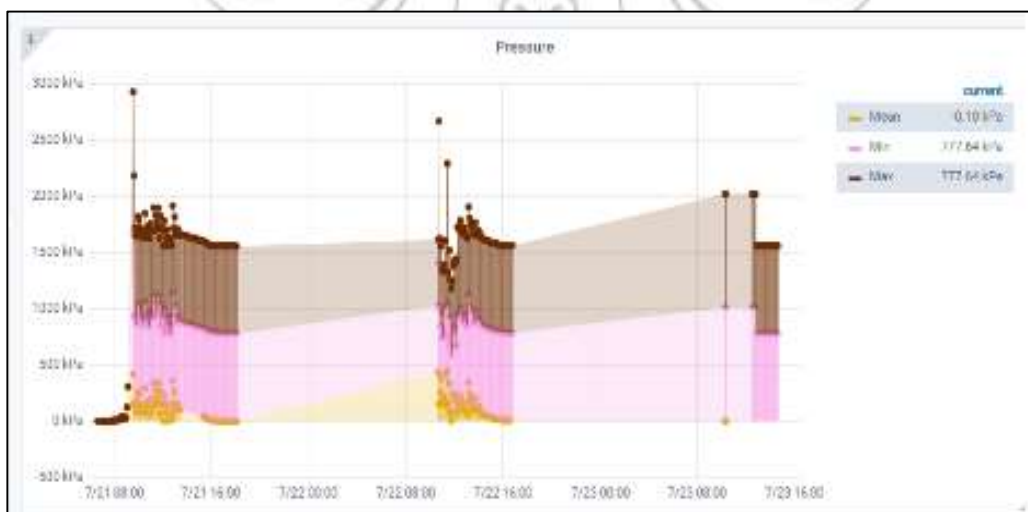
1. Data sensor BME280 (Tekanan Udara)

```
> precision rfc3339
> use mydb
Using database mydb
> select * from "bme/Pressure"
name: bme/Pressure
time                value
-----
2019-07-21T09:35:48Z 1255
2019-07-21T09:36:10Z 1255
2019-07-21T09:37:36Z 1255
2019-07-21T09:38:19Z 1255
2019-07-21T09:39:46Z 1255
2019-07-21T09:40:29Z 1255
2019-07-21T09:41:55Z 777.64
2019-07-21T09:42:39Z 777.64
2019-07-21T09:44:05Z 777.64
2019-07-21T09:44:49Z 777.64
2019-07-21T09:46:15Z 777.64
2019-07-21T09:46:59Z 777.64
2019-07-21T09:48:25Z 777.64
2019-07-21T09:49:09Z 777.64
2019-07-21T09:50:35Z 777.64
2019-07-21T09:51:19Z 777.64
2019-07-21T09:52:45Z 777.64
2019-07-21T09:53:29Z 777.64
2019-07-21T09:54:55Z 777.64
2019-07-21T09:55:39Z 777.64
2019-07-21T09:57:05Z 777.64
2019-07-21T09:57:49Z 777.64
2019-07-21T09:59:15Z 777.64
2019-07-21T09:59:59Z 777.64
2019-07-21T10:01:25Z 777.64
2019-07-21T10:02:08Z 777.64
2019-07-21T10:03:35Z 777.64
2019-07-21T10:04:18Z 777.64
2019-07-21T10:05:45Z 777.64
2019-07-21T10:06:28Z 777.64
```

Query untuk Menampilkan *min*, *max*, *mean*, dan *last* dari data Tekanan Udara

```
> select last(*) from "bme/Pressure"
name: bme/Pressure
time                last_value
-----
2019-07-23T14:39:59Z 777.64
> ^C
> select mean(*) from "bme/Pressure"
name: bme/Pressure
time                mean_value
-----
1970-01-01T00:00:00Z 772.7729784946318
> select min(*) from "bme/Pressure"
name: bme/Pressure
time                min_value
-----
2019-07-22T10:41:20Z 588.24
> select max(*) from "bme/Pressure"
name: bme/Pressure
time                max_value
-----
2019-07-21T09:35:48Z 1255
```

Tampilan visualisasi tekanan udara pada Grafana



2. Data Sensor Rainfall

Data Rainfall pada influxDB

```
> select * from "rain/Rain"
name: rain/Rain
time                value
-----
2019-07-21T09:35:48Z 1
2019-07-21T09:36:10Z 1
2019-07-21T09:37:37Z 1
2019-07-21T09:38:20Z 1
2019-07-21T09:39:46Z 1
2019-07-21T09:40:30Z 1
2019-07-21T09:41:56Z 1
2019-07-21T09:42:39Z 1
2019-07-21T09:44:06Z 1
2019-07-21T09:44:49Z 2
2019-07-21T09:46:16Z 2
2019-07-21T09:46:59Z 1
2019-07-21T09:48:26Z 1
2019-07-21T09:49:09Z 1
2019-07-21T09:50:36Z 2
2019-07-21T09:51:19Z 1
2019-07-21T09:52:46Z 1
2019-07-21T09:53:29Z 1
2019-07-21T09:54:56Z 1
2019-07-21T09:55:39Z 1
2019-07-21T09:57:06Z 1
2019-07-21T09:57:49Z 1
2019-07-21T09:59:16Z 1
2019-07-21T09:59:59Z 1
2019-07-21T10:01:26Z 1
2019-07-21T10:02:09Z 1
2019-07-21T10:03:36Z 1
2019-07-21T10:04:19Z 1
2019-07-21T10:05:46Z 1
2019-07-21T10:06:29Z 1
2019-07-21T10:07:56Z 1
2019-07-21T10:08:39Z 1
2019-07-21T10:10:06Z 1
2019-07-21T10:10:49Z 1
```

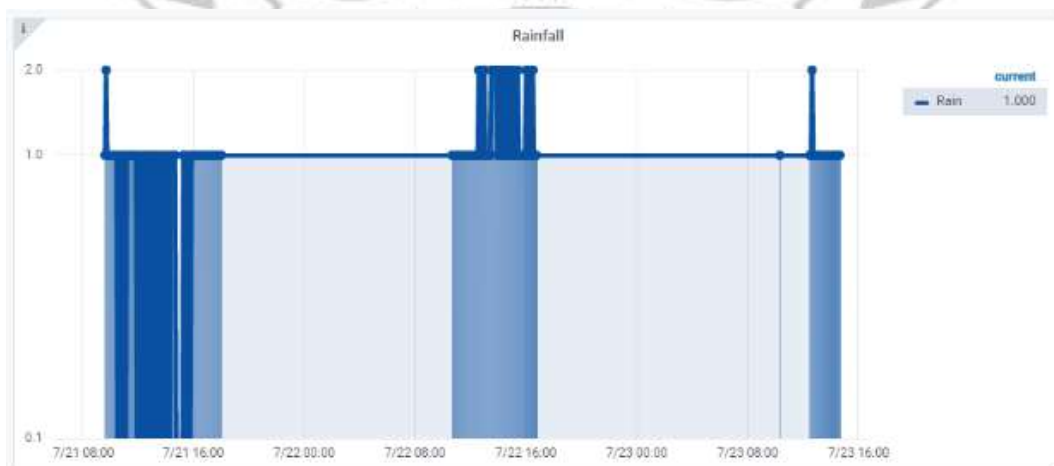
Keterangan

Nilai 1 menunjukkan peringatan hujan

Nilai 0 menunjukkan hujan

Nilai 2 menunjukkan tidak hujan

Visualisasi data rainfall



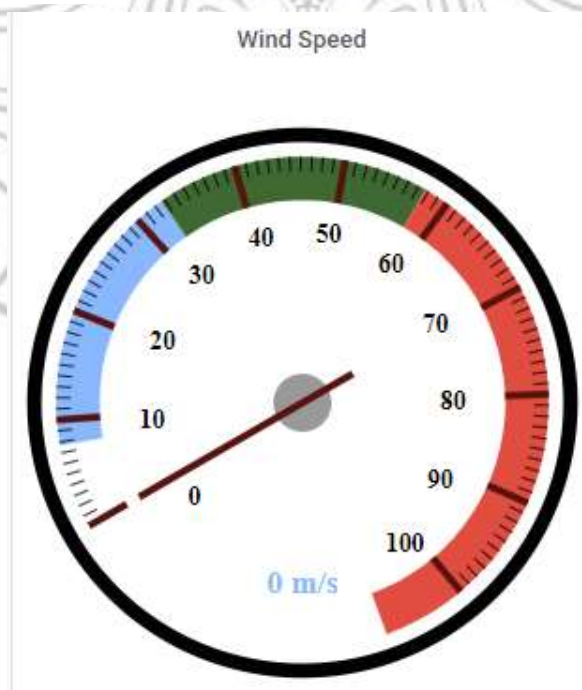
3. Data Senso Anemometer

Data pada influxDB

```
> select * from "anemometer/Speedwind"
name: anemometer/Speedwind
time          value
-----
2019-07-21T09:35:49Z 2.41
2019-07-21T09:36:00Z 2.92
2019-07-21T09:36:43Z 3.82
2019-07-21T09:38:09Z 3.47
2019-07-21T09:38:52Z 4.37
2019-07-21T09:40:19Z 3.37
2019-07-21T09:41:02Z 2.56
2019-07-21T09:42:29Z 3.22
2019-07-21T09:43:12Z 4.27
2019-07-21T09:44:39Z 3.97
2019-07-21T09:45:22Z 3.02
2019-07-21T09:46:49Z 3.97
2019-07-21T09:47:32Z 3.07
2019-07-21T09:48:59Z 2.81
2019-07-21T09:49:42Z 1.86
2019-07-21T09:51:08Z 2.66
2019-07-21T09:51:52Z 2.76
2019-07-21T09:53:18Z 2.76
2019-07-21T09:54:02Z 3.52
2019-07-21T09:55:28Z 5.13
2019-07-21T09:56:12Z 5.23
2019-07-21T09:57:38Z 3.62
2019-07-21T09:58:22Z 3.27
2019-07-21T09:59:48Z 3.92
2019-07-21T10:00:32Z 2.87
2019-07-21T10:01:58Z 4.52
2019-07-21T10:02:42Z 3.17
2019-07-21T10:04:08Z 3.77
```

```
> select last(*) from "anemometer/Speedwind"
name: anemometer/Speedwind
time          last_value
-----
2019-07-30T09:06:24Z 0
> select count(*) from "anemometer/Speedwind"
name: anemometer/Speedwind
time          count_value
-----
1970-01-01T00:00:00Z 2067
> select min(*) from "anemometer/Speedwind"
name: anemometer/Speedwind
time          min_value
-----
2019-07-21T17:22:57Z 0
> select max(*) from "anemometer/Speedwind"
name: anemometer/Speedwind
time          max_value
-----
2019-07-23T14:31:10Z 54.82
```

Visualisasi di Grafana



Data sensor *Wind Vane direction* / Arah Angin

```
> use mydb
Using database mydb
> select * from "Wine/Value"
name: Wine/Value
time                value
-----
2019-07-21T09:35:49Z 135
2019-07-21T09:36:28Z 135
2019-07-21T09:37:54Z 90
2019-07-21T09:38:37Z 90
2019-07-21T09:40:03Z 45
2019-07-21T09:40:47Z 45
2019-07-21T09:42:13Z 0
2019-07-21T09:42:57Z 225
2019-07-21T09:44:23Z 45
2019-07-21T09:45:07Z 45
2019-07-21T09:46:33Z 0
2019-07-21T09:47:17Z 45
2019-07-21T09:48:43Z 180
2019-07-21T09:49:26Z 180
2019-07-21T09:50:53Z 0
2019-07-21T09:51:36Z 90
2019-07-21T09:53:03Z 90
2019-07-21T09:53:46Z 180
2019-07-21T09:55:13Z 90
2019-07-21T09:55:56Z 45
2019-07-21T09:57:23Z 90
2019-07-21T09:58:06Z 135
2019-07-21T09:59:33Z 90
2019-07-21T10:00:16Z 90
2019-07-21T10:01:43Z 180
2019-07-21T10:02:26Z 90
2019-07-21T10:03:53Z 90
2019-07-21T10:04:36Z 90
2019-07-21T10:06:03Z 180
2019-07-21T10:06:46Z 180
2019-07-21T10:08:13Z 45
2019-07-21T10:08:56Z 45
```

```
> select min(*) from "Wine/Value"
name: Wine/Value
time                min_value
-----
2019-07-21T09:42:13Z 0
> select max(*) from "Wine/Value"
name: Wine/Value
time                max_value
-----
2019-07-21T13:03:44Z 315
> select mean(*) from "Wine/Value"
name: Wine/Value
time                mean_value
-----
1970-01-01T00:00:00Z 107.33369098712447
> select last(*) from "Wine/Value"
name: Wine/Value
time                last_value
-----
2019-07-23T14:40:17Z 180
> select (*) from "Wine/Value"
name: Wine/Value
time                value
-----
2019-07-21T09:35:49Z 135
2019-07-21T09:36:28Z 135
```

Visualisasi Pada Grafana

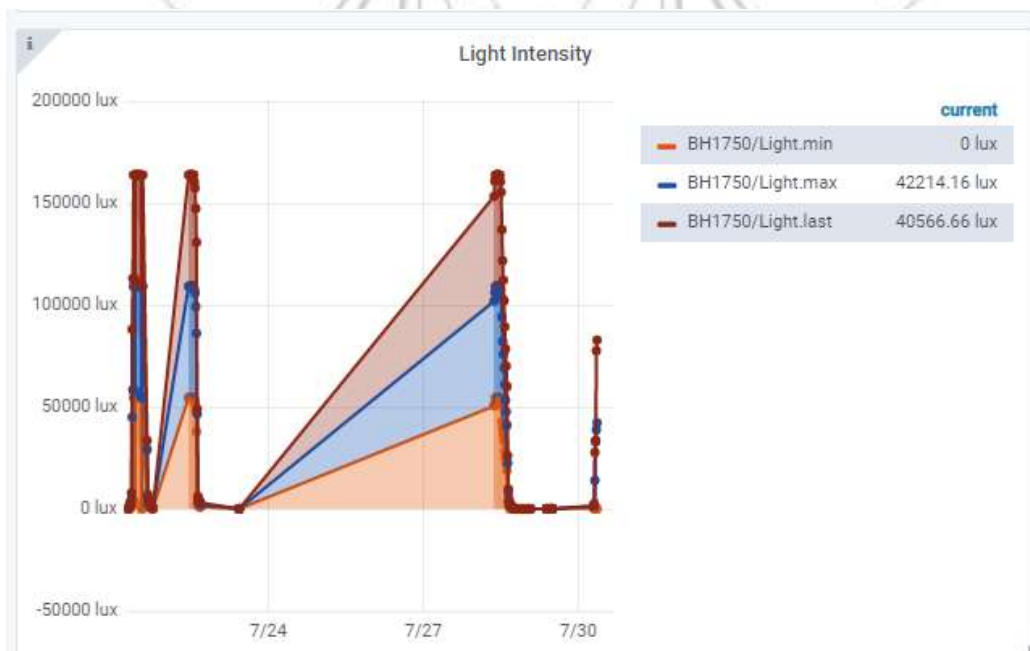


4. Data Sensor BH1750

```
> select * from "BH1750/Light"
name: BH1750/Light
time          value
-----
2019-07-21T06:40:06Z 3.33
2019-07-21T07:02:02Z 7.5
2019-07-21T07:02:29Z 7.5
2019-07-21T07:03:21Z 7.5
2019-07-21T07:03:47Z 7.5
2019-07-21T07:04:40Z 7.5
2019-07-21T07:05:06Z 7.5
2019-07-21T07:05:59Z 7.5
2019-07-21T07:06:25Z 7.5
2019-07-21T07:07:18Z 7.5
2019-07-21T07:07:45Z 6.67
2019-07-21T07:13:43Z 0
2019-07-21T07:14:09Z 375
2019-07-21T07:15:01Z 378.33
2019-07-21T07:15:27Z 377.5
2019-07-21T07:16:19Z 466.67
2019-07-21T07:16:44Z 766.67
2019-07-21T07:17:36Z 774.17
2019-07-21T07:18:02Z 777.5
2019-07-21T07:18:54Z 785.83
2019-07-21T07:19:20Z 786.67
2019-07-21T07:20:12Z 797.5
2019-07-21T07:20:37Z 693.33
2019-07-21T07:21:29Z 792.5
2019-07-21T07:21:55Z 783.33
2019-07-21T07:22:47Z 810
2019-07-21T07:23:13Z 813.33
2019-07-21T07:24:05Z 770
2019-07-21T07:24:30Z 843.33
2019-07-21T07:25:22Z 836.67
```

```
> select last(*) from "BH1750/Light"
name: BH1750/Light
time          last_value
-----
2019-07-30T09:11:52Z 40566.66
> select count(*) from "BH1750/Light"
name: BH1750/Light
time          count_value
-----
1970-01-01T00:00:00Z 3605
> select max(*) from "BH1750/Light"
name: BH1750/Light
time          max_value
-----
2019-07-21T08:59:46Z 54612.5
> select min(*) from "BH1750/Light"
name: BH1750/Light
time          min_value
-----
2019-07-21T12:46:39Z -1
```

Tampilan visualisasi pada grafana



5. Data Sensor Shunt Voltage

```
> select * from "INA219/shuntvoltage_A"
name: INA219/shuntvoltage_A
time          value
-----
2019-07-21T06:40:08Z -0.22
2019-07-21T07:02:08Z -0.21
2019-07-21T07:02:34Z -0.18
2019-07-21T07:03:27Z -0.18
2019-07-21T07:03:53Z -0.2
2019-07-21T07:04:46Z -0.19
2019-07-21T07:05:12Z -0.2
2019-07-21T07:06:05Z -0.18
2019-07-21T07:06:31Z -0.18
2019-07-21T07:07:24Z -0.2
2019-07-21T07:13:49Z -0.2
2019-07-21T07:14:15Z -0.2
2019-07-21T07:15:07Z -0.2
2019-07-21T07:15:33Z -0.2
2019-07-21T07:16:25Z -0.2
2019-07-21T07:16:50Z -0.18
2019-07-21T07:17:42Z -0.2
2019-07-21T07:18:08Z -0.17
2019-07-21T07:19:00Z -0.16
2019-07-21T07:19:26Z -0.2
2019-07-21T07:20:18Z -0.17
2019-07-21T07:20:43Z -0.2
2019-07-21T07:21:35Z -0.19
2019-07-21T07:22:01Z -0.2
2019-07-21T07:22:53Z -0.19
2019-07-21T07:23:19Z -0.17
2019-07-21T07:24:10Z -0.17
2019-07-21T07:24:36Z -0.19
2019-07-21T07:25:28Z -0.17
2019-07-21T07:25:54Z -0.19
2019-07-21T07:26:45Z 163.61
2019-07-21T07:27:11Z -0.04
2019-07-21T07:28:03Z -0.15
2019-07-21T07:28:29Z -0.16
2019-07-21T07:29:21Z -0.17
2019-07-21T07:29:47Z -0.19
2019-07-21T07:30:39Z 0.48
```

```
> select first(*) from "INA219/shuntvoltage_B"
name: INA219/shuntvoltage_B
time          first_value
-----
2019-07-21T06:40:07Z -3.53
> select min(*) from "INA219/shuntvoltage_B"
name: INA219/shuntvoltage_B
time          min_value
-----
2019-07-30T08:09:08Z -14.37
> select max(*) from "INA219/shuntvoltage_B"
name: INA219/shuntvoltage_B
time          max_value
-----
2019-07-28T20:06:22Z 287.58
> select mean(*) from "INA219/shuntvoltage_B"
name: INA219/shuntvoltage_B
time          mean_value
-----
1970-01-01T00:00:00Z 1.4285433397950784
> select last(*) from "INA219/shuntvoltage_B"
name: INA219/shuntvoltage_B
time          last_value
-----
2019-07-30T09:17:22Z 35.36
```

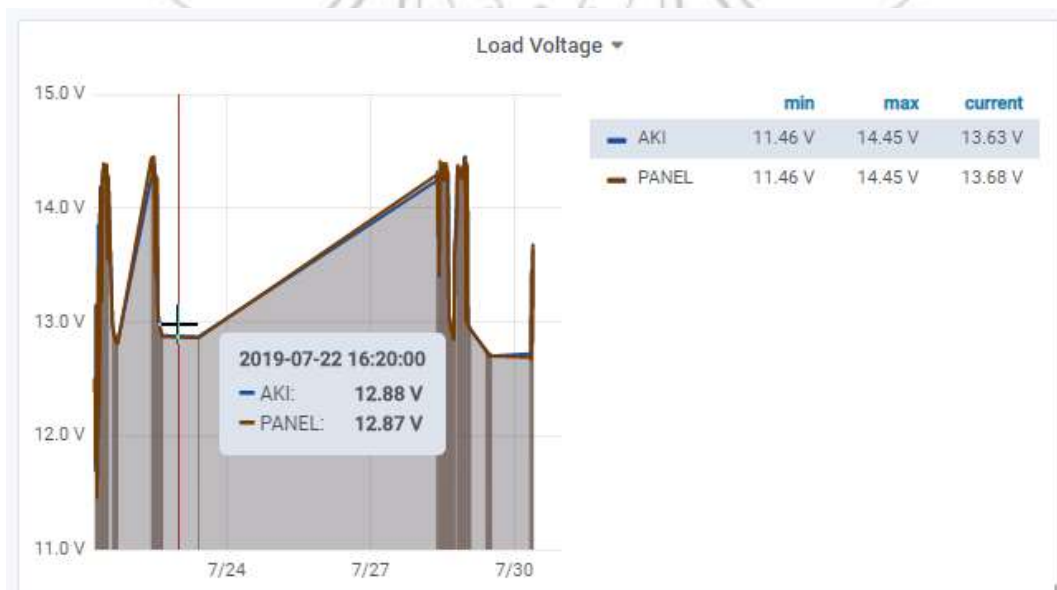
Tampilan visualisasi pada grafana



6. Data Sensor Load Voltage

```
> select * from "INA219/loadvoltage_A"
name: INA219/loadvoltage_A
time          value
-----
2019-07-21T06:40:08Z 12.51
2019-07-21T07:02:08Z 12.35
2019-07-21T07:02:35Z 12.35
2019-07-21T07:03:27Z 12.34
2019-07-21T07:03:54Z 12.34
2019-07-21T07:04:46Z 12.34
2019-07-21T07:05:13Z 12.34
2019-07-21T07:06:06Z 12.34
2019-07-21T07:06:32Z 12.34
2019-07-21T07:07:25Z 12.32
2019-07-21T07:13:49Z 12.42
2019-07-21T07:14:15Z 12.26
2019-07-21T07:15:07Z 11.86
2019-07-21T07:15:33Z 11.8
2019-07-21T07:16:25Z 11.71
2019-07-21T07:16:51Z 11.68
2019-07-21T07:17:43Z 12.04
2019-07-21T07:18:09Z 12.08
2019-07-21T07:19:00Z 12.05
2019-07-21T07:19:26Z 11.7
2019-07-21T07:20:18Z 11.63
2019-07-21T07:20:44Z 11.6
2019-07-21T07:21:36Z 11.55
2019-07-21T07:22:02Z 11.53
2019-07-21T07:22:53Z 11.46
2019-07-21T07:23:19Z 11.39
2019-07-21T07:24:11Z 11.36
2019-07-21T07:24:37Z 11.34
2019-07-21T07:25:29Z 11.21
2019-07-21T07:25:54Z 13.04
2019-07-21T07:26:46Z 14.22
2019-07-21T07:27:12Z 13.38
2019-07-21T07:28:04Z 13.16
2019-07-21T07:28:29Z 13.15
2019-07-21T07:29:22Z 13.14
```

Tampilan visualisasi pada grafana



7. Data Sensor Bus Voltage

```
> select * from "INA219/busvoltage_B"
name: INA219/busvoltage_B
time          value
-----
2019-07-21T06:40:07Z 12.51
2019-07-21T07:02:05Z 12.36
2019-07-21T07:02:31Z 12.36
2019-07-21T07:03:24Z 12.35
2019-07-21T07:03:50Z 12.35
2019-07-21T07:04:43Z 12.35
2019-07-21T07:05:09Z 12.35
2019-07-21T07:06:02Z 12.35
2019-07-21T07:06:28Z 12.35
2019-07-21T07:07:21Z 12.34
2019-07-21T07:13:46Z 12.51
2019-07-21T07:14:12Z 12.28
2019-07-21T07:15:04Z 11.88
2019-07-21T07:15:30Z 11.81
2019-07-21T07:16:22Z 11.72
2019-07-21T07:16:47Z 11.69
2019-07-21T07:17:39Z 11.96
2019-07-21T07:18:05Z 12.07
2019-07-21T07:18:57Z 12.1
2019-07-21T07:19:23Z 11.7
2019-07-21T07:20:15Z 11.63
2019-07-21T07:20:40Z 11.61
2019-07-21T07:21:32Z 11.55
2019-07-21T07:21:58Z 11.54
2019-07-21T07:22:50Z 11.44
2019-07-21T07:23:16Z 11.45
2019-07-21T07:24:07Z 11.38
```

```
using database mydb
> select last(*) from "INA219/busvoltage_B"
name: INA219/busvoltage_B
time          last_value
-----
2019-07-30T09:33:21Z 13.7
> select min(*) from "INA219/busvoltage_B"
name: INA219/busvoltage_B
time          min_value
-----
2019-07-21T12:46:43Z 0
> select max(*) from "INA219/busvoltage_B"
name: INA219/busvoltage_B
time          max_value
-----
2019-07-28T23:14:34Z 14.54
> select count(*) from "INA219/busvoltage_B"
name: INA219/busvoltage_B
time          count_value
-----
1970-01-01T00:00:00Z 3642
```

Tampilan visualisasi pada grafana



8. Data Sensor Current MA/Arus

Data sensor pada InfluxDB

```
> select * from "INA219/current_mA_A"
name: INA219/current_mA_A
time                value
-----
2019-07-21T06:40:08Z -1.9
2019-07-21T07:02:09Z -1.9
2019-07-21T07:02:35Z -1.9
2019-07-21T07:03:28Z -1.9
2019-07-21T07:03:54Z -2.1
2019-07-21T07:04:47Z -1.9
2019-07-21T07:05:13Z -1.9
2019-07-21T07:06:06Z -2.1
2019-07-21T07:06:32Z -1.9
2019-07-21T07:07:25Z -1.9
2019-07-21T07:13:50Z -1.9
2019-07-21T07:14:16Z -1.8
2019-07-21T07:15:08Z -2.2
2019-07-21T07:15:34Z -1.9
2019-07-21T07:16:26Z -1.9
2019-07-21T07:16:51Z -2
2019-07-21T07:17:43Z -1.6
2019-07-21T07:18:09Z -1.8
2019-07-21T07:19:01Z -2.1
2019-07-21T07:19:27Z -2
2019-07-21T07:20:19Z -1.9
2019-07-21T07:20:44Z -1.9
2019-07-21T07:21:36Z -1.9
2019-07-21T07:22:02Z -2
2019-07-21T07:22:54Z -2
2019-07-21T07:23:20Z -2.1
2019-07-21T07:24:11Z -1.9
2019-07-21T07:24:37Z -2
2019-07-21T07:25:29Z -1.9
2019-07-21T07:25:55Z -2
```

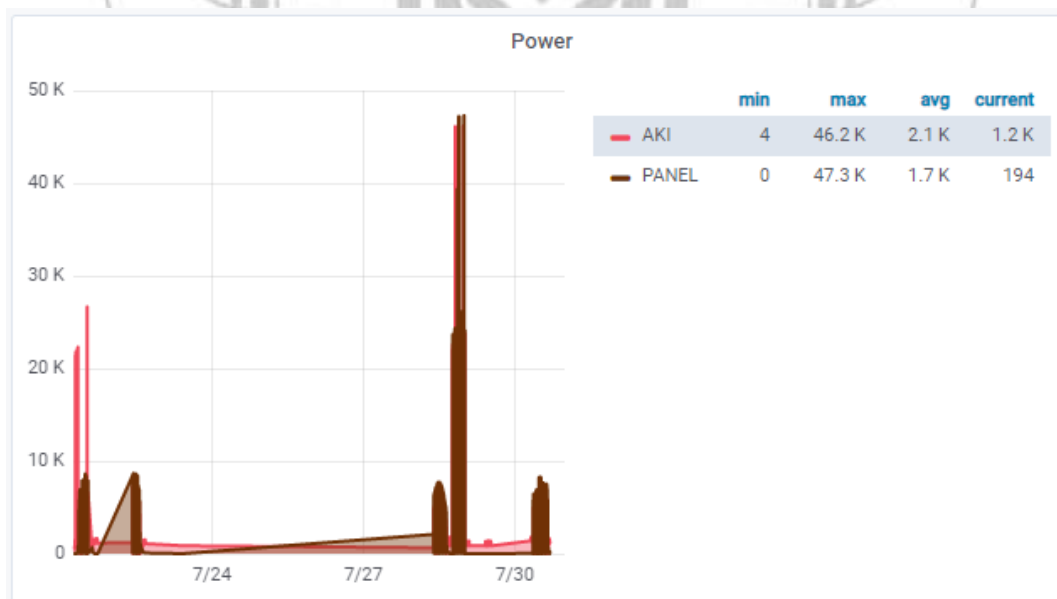
Tampilan visualisasi pada grafana



9. Data Sensor Power

```
> select min(*) from "INA219/power_mW_B"
name: INA219/power_mW_B
time          min_value
-----
2019-07-22T12:53:24Z 4
> select max(*) from "INA219/power_mW_B"
name: INA219/power_mW_B
time          max_value
-----
2019-07-28T23:22:35Z 46198
> select mean(*) from "INA219/power_mW_B"
name: INA219/power_mW_B
time          mean_value
-----
1970-01-01T00:00:00Z 2108.8960293780124
> select current(*) from "INA219/power_mW_B"
ERR: undefined function current()
> select last(*) from "INA219/power_mW_B"
name: INA219/power_mW_B
time          last_value
-----
2019-07-30T16:24:36Z 1212
```

Visualisasi pada grafana



11) Data Relay

```
2019-07-30T16:14:26Z 2
2019-07-30T16:14:48Z 2
2019-07-30T16:15:32Z 2
2019-07-30T16:15:53Z 2
2019-07-30T16:16:37Z 2
2019-07-30T16:16:59Z 2
2019-07-30T16:17:42Z 2
2019-07-30T16:18:04Z 2
2019-07-30T16:18:48Z 2
2019-07-30T16:19:09Z 2
2019-07-30T16:19:53Z 2
2019-07-30T16:20:15Z 2
2019-07-30T16:20:58Z 2
2019-07-30T16:21:20Z 2
2019-07-30T16:22:04Z 2
2019-07-30T16:22:26Z 2
2019-07-30T16:23:09Z 2
2019-07-30T16:23:31Z 2
2019-07-30T16:24:15Z 2
2019-07-30T16:24:36Z 2
2019-07-30T16:25:20Z 2
2019-07-30T16:25:42Z 2
2019-07-30T16:26:26Z 2
2019-07-30T16:26:47Z 2
2019-07-30T16:27:31Z 2
2019-07-30T16:27:53Z 2
2019-07-30T16:28:36Z 2
2019-07-30T16:28:58Z 2
2019-07-30T16:29:42Z 2
2019-07-30T16:30:04Z 2
```

Keterangan: 0 = Tidak mengisi
1: mengisi (Jalur dari PLN)
2: mengisi (Jalur dari Panel)
3: baterai Penuh

Visualisasi pada grafana

