

**INTEGRASI DAN VISUALISASI DATA SENSOR KELEMBABAN,  
SENSOR SUHU, DAN SENSOR PERGESERAN TANAH PADA EMBUNG  
UNIVERSITAS LAMPUNG**

**(Skripsi)**

Oleh

**VINI MEIRISKA**



**JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS LAMPUNG  
2022**

## ABSTRAK

### INTEGRASI DAN VISUALISASI DATA SENSOR KELEMBABAN, SENSOR PERGESERAN, DAN SENSOR SUHU PADA EMBUNG UNIVERSITAS LAMPUNG

Oleh

VINI MEIRISKA

Integrasi dan visualisasi data bertujuan untuk mengolah data dan menampilkan data yang didapat dari informasi data Sensor Kelembaban, Sensor Pergeseran, serta Sensor Suhu yang berada di sekitar embung Universitas Lampung. Embung di Universitas Lampung masih belum memiliki sistem monitoring untuk mengetahui keadaan kondisi tanah di sekitar embung untuk mewaspadai akan terjadinya longsor pada lereng embung agar pengelola embung dapat mengetahui dugaan rembesan sedari awal agar tidak terjadi banjir maupun amblesan di lingkungan sekitar Embung Universitas Lampung. Pengiriman data pada embung menggunakan teknologi komunikasi yaitu perangkat modul Lora *Shield* Arduino sehingga pengiriman data dapat terikirim secara *wireless* dengan jarak pengiriman data yang cukup jauh juga dengan konsumsi daya yang kecil. Data yang sudah terkirim dari *node sensor* ke *end node* akan melalui proses pengolahan data terlebih dahulu. Pengolahan data bertujuan agar data yang memiliki *volume*, *variety*, dan juga *value* yang besar dapat diolah dengan bantuan aplikasi pengolah data yaitu Pentaho sehingga didapatkan informasi baru hasil dari olah data secara akurat. Data sensor yang didapat merupakan data yang berasal dari Sensor *Strain Gauge*, Sensor *Moisture Soil*, dan Sensor BMP280 dengan keluaran informasi baru berupa data rata-rata nilai masing-masing sensor serta kondisi tanah pada tiap pengukuran oleh masing-masing sensor. Data kemudian di visualisasikan dengan dukungan *webhosting* 000webhost, sehingga hasil akhir dari penelitian menunjukkan tampilan grafik data tiap sensor beserta list data.

Kata Kunci: LoRa, *monitoring system*, Pentaho, *Strain Gauge*, *Moisture Soil*, BMP, MYSQL.

## **ABSTRACT**

### **INTEGRATION AND DATA VISUALIZATION OF HUMIDITY SENSORS, SHIFT SENSORS, AND TEMPERATURE SENSORS AT THE EMBODIMENT OF LAMPUNG UNIVERSITY**

**By**

**VINI MEIRISKA**

Data integration and visualization aims to process data and display data obtained from data information on Humidity Sensors, Shift Sensors, and Temperature Sensors around the Lampung University reservoir. The reservoir at the University of Lampung still does not have a monitoring system to determine the state of the soil conditions around the reservoir to be aware of the occurrence of landslides on the slopes of the reservoir so that the reservoir manager can find out the alleged seepage from the beginning so that there are no floods or subsidence in the environment around the reservoir of University of Lampung. Data transmission in the reservoir uses communication technology, namely the Lora Shield Arduino module device so that data transmission can be sent wirelessly with a long distance of data transmission with low power consumption. The data that has been sent from the sensor node to the end node will go through a data processing process first. Data processing aims that data that has a large volume, variety, and value can be processed with the help of a data processing application, namely Pentaho so that new information is obtained from data processing accurately. The sensor data obtained is data from the Strain Gauge Sensor, Moisture Soil Sensor, and the BMP280 Sensor with new information output in the form of data on the average value of each sensor and the soil conditions for each measurement by each sensor. The data is then visualized with the support of 000webhost webhosting, so that the final result of the study shows a graphical display of the data for each sensor along with a list of data.

**Keywords:** LoRa, monitoring system, Pentaho, Strain Gauge, Moisture Soil, BMP, MYSQL

**INTEGRASI DAN VISUALISASI DATA SENSOR KELEMBABAN,  
SENSOR SUHU, DAN SENSOR PERGESERAN TANAH PADA EMBUNG  
UNIVERSITAS LAMPUNG**

**Oleh**

**VINI MEIRISKA**

**Skripsi**

**Sebagai salah satu syarat untuk mencapai gelar  
SARJANA TEKNIK**

**Pada**

**Program Studi Teknik Elektro  
Jurusan Teknik Elektro  
Fakultas Teknik Universitas Lampung**



**FAKULTAS TEKNIK  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2022**

Judul Skripsi

**: INTEGRASI DAN VISUALISASI DATA  
SENSOR KELEMBABAN, SENSOR SUHU,  
DAN SENSOR PERGESERAN TANAH  
PADA EMBUNG UNIVERSITAS LAMPUNG**

Nama Mahasiswa

**: *Oini Meiriska***

Nomor Pokok Mahasiswa : 1515031101

Jurusan

**: Teknik Elektro**

Fakultas

**: Teknik**



**2. Mengetahui**

Ketua Jurusan  
Teknik Elektro

*[Handwritten Signature]*

**Herlinawati, S.T., M.T.**  
NIP 19710314 199903 2 001

Ketua Program Studi  
Teknik Elektro

*[Handwritten Signature]*

**Dr. Eng. Nining Purwasih, S.T., M.T.**  
NIP 19740422 200012 2 001



## MENGESAHKAN

### 1. Tim Penguji

Ketua

: **Dr. Ing. Ardian Ulvan, S.T., M.Sc.** .....



Sekretaris

: **Dr. Ing. Melvi, S.T., M.T.** .....



Penguji

Bukan Pembimbing : **Misfa Susanto, S.T., M.Sc., Ph.D.** .....



### 2. Dekan Fakultas Teknik



**Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc.**

NIP 19750928 200112 1 002

Tanggal Lulus Ujian Skripsi : **17 Juni 2022**

## SURAT PERNYATAAN

Saya yang bertanda tangan dibawah ini, menyatakan bahwa skripsi saya yang berjudul **“Integrasi dan Visualisasi Data Sensor Kelembaban, Sensor Suhu, dan Sensor Pergeseran Tanah pada Embung Universitas Lampung”** merupakan hasil karya sendiri dan bukan hasil karya orang lain. Semua hasil yang tertuang dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila dikemudian hari terbukti skripsi ini merupakan salinan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan ketentuan akademik yang berlaku.

Bandar Lampung, 17 Juni 2022



**Vini Meiriska**  
NPM. 1515031101

## RIWAYAT HIDUP



Penulis dilahirkan di Metro, pada tanggal 11 Mei 1997.

Penulis merupakan anak keempat dari tiga bersaudara dari pasangan Bapak Joni Efdi dan Ibu Ratna Dewi yang diberi nama Vini Meiriska. Mengenai riwayat pendidikan penulis, penulis lulus Sekolah Dasar (SD) di SD Negeri 06 Metro Barat tahun 2009, lulus Sekolah Menengah Pertama (SMP)

di SMPN 1 Metro pada tahun 2012, lulus Sekolah Menengah Atas (SMA) di SMAN 1 Metro pada tahun 2015, dan diterima di Jurusan Teknik Elektro Universitas Lampung (Unila) pada tahun 2015.

Selama menjadi mahasiswa, penulis aktif di Organisasi Himpunan Mahasiswa Teknik Elektro (HIMATRO) Fakultas Teknik sebagai Anggota Departemen Kominfo pada periode 2016-2018. Selain mengikuti organisasi, penulis juga berkesempatan menjadi asisten praktikum pada Laboratorium Telekomunikasi selama 2 periode 2016-2018, selain menjadi asisten penulis juga merupakan anggota kegiatan dalam bidang *Robotic* pada divisi *Soccer Robotic* pada tahun 2017-2018. Selain itu, penulis pernah melakukan Kerja Praktek (KP) selama 40 hari di Badan Penelitian dan Pengembangan Teknoogi (BPPT), Tangerang pada tahun 2017.



## PERSEMBAHAN

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Skripsi ini kupersembahkan untuk

” Ibu dan Ayah”

Yang selalu mendoakan penulis di waktu terbaik sepertiga malam terakhir dan senantiasa memberikan dukungan moril maupun materil dalam menyelesaikan Skripsi.

-TERIMAKASIH-

# Motto

*“Hai orang-orang mukmin, jika kamu menolong (agama) Allah, niscaya Dia akan menolongmu dan meneguhkan kedudukanmu”*

(QS. Muhammad: 7)

*“Kemudian apabila kamu telah membulatkan tekad, maka bertawakallah kepada Allah. Sesungguhnya Allah menyukai orang-orang yang bertawakal pada-Nya.”*

(QS. Ali-Imran: 159)

*“Allah akan meninggikan orang-orang yang beriman diantaramu dan orang-orang yang diberi ilmu pengetahuan beberapa derajat.”*

(QS. Al-Mujadalah: 11)

## SANWACANA

Alhamdulillahirobbil'alamiin, penulis haturkan puji syukur kehadiran Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan penelitian Tugas Akhir ini.

Tugas akhir dengan judul “Integrasi dan Visualisasi Data Sensor Kelembaban, Sensor Suhu, dan Sensor Pergeseran Tanah pada Embung Universitas Lampung” ini merupakan salah satu syarat untuk memperoleh gelar Sarjana Teknik pada Jurusan Teknik Elektro Fakultas Teknik Universitas Lampung.

Dalam masa perkuliahan dan penelitian, penulis mendapat banyak hal baik berupa dukungan, semangat, motivasi dan banyak hal yang lainnya. Untuk itu penulis mengucapkan terimakasih kepada :

1. Bapak Prof. Karomani, M.Si. selaku Rektor Universitas Lampung.
2. Bapak Dr. Eng. Helmy Fitriawan, S.T., M.sc. selaku Dekan Fakultas Teknik Universitas Lampung.
3. Bapak Khairudin, S.T., M.Sc., Ph.D., Eng. selaku Ketua Jurusan Teknik Elektro Fakultas Teknik Universitas Lampung.
4. Bapak Dr. Ing. Ardian Ulvan, S.T., M.Sc. selaku pembimbing utama skripsi yang telah dengan sabar dan memberikan begitu banyak waktu untuk membimbing, memberikan ilmu, semangat, motivasi dan juga arahan dari beliau.
5. Bapak Misfa Susanto, S.T., M.Sc., Ph.D. selaku penguji skripsi yang telah memberikan saran, memberi ilmu motivasi selama masa pendidikan sampai pengujianskripsi.
6. Ibu Dr. Ing. Melvi, S.T., M.T. juga selaku dosen penguji skripsi yang telah memberikan saran dan kritik yang membangun dalam penyusunan skripsi.

7. Ibu Dr. Ing. Melvi, S.T., M.T. selaku salah satu dosen TELTI dan juga kepala Laboratorium Telekomunikasi yang telah memberikan fasilitas dan mengadakan kegiatan-kegiatan yang baik serta bermanfaat untuk lab.
8. Ibu Dr. Eng. Nining Purwasih, S.T., M.T. selaku dosen pembimbing akademik (PA) yang telah memberikan nasihat dan arahan yang membangun.
9. Segenap dosen dan pegawai di Jurusan Teknik Elektro yang telah memberikan ilmu dan wawasan yang tak terlupakan oleh penulis.
10. Kedua orang tua penulis, untuk Ayah dan Ibu yang pastinya selalu mendoakan yang terbaik untuk penulis. Terimakasih untuk kesabaran dan supportnya, sehingga penulis dapat menyelesaikan tugas terakhir di jenjang perkuliahan ini.
11. Abang Viky, Uni Nanda, dan juga Uni Mela, selaku kakak kandung dari penulis, penulis mengucapkan terimakasih untuk dorongan, doa, dan juga motivasinya.
12. Teman-teman EIE 2015 yang sudah seperti keluarga sendiri. Terima kasih untuk segala kebaikan yang sudah diberikan.
13. Teman-teman SEKAR 2015, yang sudah saling support dan saling mengingatkan penulis sangat berterima kasih.
14. SORAIDA SABELLA sebagai kawan dan juga motivator penulis untuk menyelesaikan skripsi ini, terimakasih sudah mau memberikan ocehan yang sangat bermanfaat untuk penulis.

Semoga dengan bantuan yang telah diberikan secara tulus oleh semua pihak yang terkait, diberikan keberkahan oleh Allah SWT. Penulis meminta maaf atas segala kesalahan dan ketidaksempurnaan dalam penyusunan tugas akhir ini. Saran dan kritik membangun sangat diharapkan penulis demi kebaikan di masa yang akan datang. Terimakasih.

Bandar Lampung, 15 Juni 2022

Penulis,

**Vini Meiriska**

## DAFTAR ISI

	Halaman
<b>ABSTRAK .....</b>	<b>i</b>
<b>HALAMAN JUDUL .....</b>	<b>iii</b>
<b>LEMBAR PERSETUJUAN .....</b>	<b>iv</b>
<b>LEMBAR PENGESAHAN .....</b>	<b>v</b>
<b>RIWAYAT HIDUP .....</b>	<b>vii</b>
<b>PERSEMBAHAN.....</b>	<b>viii</b>
<b>SANWACANA .....</b>	<b>x</b>
<b>DAFTAR ISI.....</b>	<b>xii</b>
<b>DAFTAR GAMBAR.....</b>	<b>xvi</b>
<b>DAFTAR TABEL .....</b>	<b>xviii</b>
 <b>BAB I PENDAHULUAN</b>	
1.1 Latar Belakang .....	1
1.2 Tujuan Skripsi .....	2
1.3 Manfaat Skripsi .....	2
1.4 Rumusan Masalah .....	3
1.5 Batasan Masalah .....	3
1.6 Sistematika Penulisan.....	4



## BAB II TINJAUAN PUSTAKA

2.1 Penelitian Terkait .....	5
2.2 <i>Monitoring System</i> .....	6
2.3 <i>Big Data</i> .....	7
2.4 LoRa ( <i>Long Range</i> ) .....	9
2.5 Arduino .....	10
2.5.1 Arduino IDE .....	11
2.6 Sensor .....	12
2.6.1 Sensor <i>Strain Gauge</i> .....	12
2.6.2 Sensor <i>Soil Moisture YL-69</i> .....	14
2.6.3 Sensor BMP280 .....	15
2.7 <i>Pentaho Data Integration</i> .....	16
2.8 <i>Database</i> .....	17
2.8.1 MySQL .....	18
2.9 <i>Website</i> .....	18
2.9.1 <i>Webserver</i> .....	19

## BAB III METODE PENELITIAN

3.1 Waktu dan Tempat Penelitian .....	20
3.2 Alat dan Bahan .....	20
3.3 Metodologi Penelitian .....	21
3.3.1 Studi Literatur .....	21
3.3.2 Pembuatan Perangkat .....	21
3.3.2.1 Perangkat <i>End Node</i> .....	21
3.3.2.2 Perangkat <i>Node Sensor</i> .....	22
a. Perancangan Sensor <i>Strain Gauge</i> .....	23

b. Perancangan Sensor BMP280 .....	25
c. Perancangan Sensor <i>Moisture Soil</i> YL-69 .....	25
3.3.3 Perancangan Basis Data Sementara .....	26
3.3.4 Perancangan Pengolah Data.....	27
3.3.4.1 Proses Pengolahan Data Pentaho.....	27
a. Integrasi Data Sensor BMP.....	27
b. Integrasi Data Sensor <i>Strain Gauge</i> .....	31
c. Integrasi Data Sensor <i>Moisture Soil</i> YL-69.....	34
d. Tahap <i>Job Data</i> .....	38
e. Tahap Koneksi Pentaho ke <i>Database</i> .....	39
3.3.5 Perancangan <i>Website</i> .....	40
3.3.5.1 Perancangan <i>Database</i> .....	41
3.3.5.2 <i>Upload File Hosting</i> .....	42
3.3.5.3 Perancangan Halaman Utama <i>Website</i> .....	42
3.4 Perancangan Skenario Penelitian.....	43
3.4.1 Skenario Penelitian Sensor <i>Moisture Soil</i> .....	45
3.4.2 Skenario Sensor <i>Strain Gauge</i> .....	46
3.5 Diagram Alir.....	46

## **BAB IV HASIL DAN PEMBAHASAN**

4.1 Hasil Data <i>Node Sensor</i> .....	48
4.2 Hasil Pengolahan Data pada Pentaho .....	51
4.2.1 Hasil Integrasi Data Sensor BMP.....	51
4.2.2 Hasil Integrasi Data Sensor <i>Strain Gauge</i> .....	52
4.2.3 Hasil Integrasi Data Sensor <i>Moisture Soil</i> .....	54
4.3 Tampilan Hasil Monitoring di <i>Website</i> .....	54

**BAB V KESIMPULAN DAN SARAN**

5.1 Kesimpulan .....	62
5.2 Saran .....	63

**DAFTAR PUSTAKA****LAMPIRAN**

## DAFTAR GAMBAR

	Halaman
Gambar 2.1 <i>Big Data</i> 4v .....	7
Gambar 2.2 Arsitektur LoRa.....	10
Gambar 2.3 Arduino Uno.....	11
Gambar 2.4 <i>Strain Gauge Sensor</i> .....	13
Gambar 2.5 Sensor <i>Soil Moisture</i> YL-69.....	14
Gambar 2.6 Sensor BMP280 .....	16
Gambar 3.1 Arsitektur <i>End Node</i> .....	21
Gambar 3.2 Arsitektur <i>Node Sensor</i> .....	23
Gambar 3.3 Sensor <i>Strain Gauge</i> yang Terpasang pada Beam .....	24
Gambar 3.4 Rangkaian Instalasi Sensor <i>Strain Gauge</i> Terhadap hx711 .....	25
Gambar 3.5 Tampilan Blok-Blok <i>Transformation</i> terhadap Data Sensor BMP...28	
Gambar 3.6 Tampilan Perintah <i>String Cut</i> .....	28
Gambar 3.7 Tampilan Perintah <i>Group By</i> Data Sensor BMP.....	29
Gambar 3.8 Tampilan Perintah <i>Sort Row</i> Data Sensor BMP.....	29
Gambar 3.9 Tampilan <i>Insert/Update</i> pada Output Data Sensor BMP.....	30
Gambar 3.10 <i>Script</i> Tabel untuk Data Sensor BMP ke dalam <i>Database</i> .....	31
Gambar 3.11 Tampilan Blok-Blok <i>Transformation</i> Terhadap Data Sensor <i>Strain Gauge</i> .....	31
Gambar 3.12 Tampilan Tahap Perintah <i>Strings Cut</i> Data Sensor <i>Strain Gauge</i> ..	32

Gambar 3.13 Proses <i>Number Range</i> Data Sensor <i>Strain Gauge</i> .....	32
Gambar 3.14 Proses <i>Insert/Update</i> pada <i>Output</i> Data Sensor <i>Strain Gauge</i> .....	33
Gambar 3.15 <i>Script</i> Tabel untuk Data Sensor <i>Strain Gauge</i> ke dalam <i>Database</i>	34
Gambar 3.16 Tampilan Blok Transformation Terhadap Sensor Moisture Soil ....	35
Gambar 3.17 Tampilan Perintah Tahap <i>Strings Cut</i> Data Sensor <i>Moisture Soil</i> ..	35
Gambar 3.18 Proses <i>Group by</i> Data Sensor <i>Moisture Soil</i> .....	36
Gambar 3.19 Tahap <i>Number Range</i> pada Data Sensor <i>Moisture Soil</i> .....	36
Gambar 3.20 Proses <i>Insert/Update</i> pada <i>Output</i> Data Sensor <i>Moisture Soil</i> .....	37
Gambar 3.21 <i>Script</i> Tabel untuk Data Sensor <i>Moisture Soil</i> ke dalam <i>Database</i>	38
Gambar 3.22 Aliran <i>Job Data</i> yang Sukses pada Transformasi .....	39
Gambar 3.23 Tahap Membangun Koneksi ke <i>Database</i> .....	39
Gambar 3.24 Koneksi ke <i>Database</i> Sukses .....	40
Gambar 3.25 Tampilan <i>Database</i> PHPMyAdmin .....	41
Gambar 3.26 Tampilan Halaman Utama <i>Layout Website</i> .....	43
Gambar 3.27 Arsitektur Sistem Keseluruhan .....	44
Gambar 3.28 Diagram Alir Penelitian .....	47
Gambar 4.1 Hasil Pengolahan Data Sensor BMP .....	51
Gambar 4.2 Hasil Pengolahan Data Sensor <i>Strain Gauge</i> .....	52
Gambar 4.3 Hasil Pengolahan Data Sensor <i>Moisture Soil</i> .....	54
Gambar 4.4 Tampilan Halaman Menu Grafik Nilai <i>Temperature</i> Sensor BMP ..	55
Gambar 4.5 Tampilan Halaman Menu Grafik Nilai <i>Air Pressure</i> Sensor BMP...	55
Gambar 4.6 Tampilan Halaman Menu Grafik Sensor <i>Moisture Soil</i> Titik A dan Titik B .....	56
Gambar 4.7 Tampilan Halaman Menu Grafik Sensor <i>Strain Gauge</i> .....	58



## DAFTAR TABEL

	Halaman
Tabel 3.1 Alat dan Bahan.....	20
Tabel 3.2 Skenario <i>Output</i> pada Sensor <i>Moisture</i> .....	45
Tabel 3.3 Skenario <i>Output</i> pada Sensor <i>Strain Gauge</i> .....	46
Tabel 4.1 Hasil Pengukuran Sensor <i>Moisture Soil</i> .....	48
Tabel 4.2 Hasil Pengukuran Sensor BMP .....	49
Tabel 4.3 Hasil Pengukuran Sensor <i>Strain Gauge</i> .....	50
Tabel 4.4 Hasil Perhitungan <i>Strain</i> Berdasarkan Perhitungan.....	53
Tabel 4.5 Tampilan <i>List Data</i> AVG Sensor BMP .....	59
Tabel 4.6 Tampilan <i>List Data</i> AVG Sensor <i>Moisture Soil</i> .....	60
Tabel 4.7 Tampilan <i>List Data</i> AVG Sensor <i>Strain Gauge</i> .....	61

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Secara pengertian, embung merupakan wadah aliran surplus air pada sungai ataupun air irigasi saat hujan. Perihal berarti yang wajib ditelusuri merupakan bagaimana air hujan menyerap masuk ke dalam tanah agar ditahan sebanyak-banyaknya di daerah cekungan ataupun lembah terkhusus embung. Bendung, embung maupun bendungan ialah bangunan air yang banyak dibentuk selaku salah satu pemecahan dalam bermacam permasalahan yang berhubungan dengan sumber energi air, baik pemanfaatan, pengelolaan, pelestarian ataupun penindakan energi rusak dari sumber energi yang terbarukan tersebut. Namun, embung di Universitas Lampung masih belum memiliki sistem pemantauan untuk hal seperti mengetahui tingginya nilai rembesan tanah di sekitar embung untuk mewaspadaikan terjadinya longsor pada lereng embung. Resapan air ke dalam struktur waduk menyebabkan kerusakan struktural pada tanggul waduk jika terjadi berulang kali atau dalam jangka waktu yang lama. Mencegah banjir dan penurunan muka tanah di area tersebut sehingga operator bendungan dapat menemukan infiltrasi yang diharapkan sekitar Embung Universitas Lampung. Berdasarkan hal tersebut, maka kiranya perlu adanya sistem yang dapat mendeteksi penyebab terjadinya longsor

pada lereng embung sehingga kejadian tersebut dapat di pantau agar bisa diminimalisir.

Sistem pemantauan menggunakan teknologi modul *Long Range* (LoRa) untuk mendukung proses *monitoring* yang dilakukan secara terus menerus dan *realtime* yang akan menghasilkan data yang memiliki nilai. Data lantas akan diolah agar data dapat divisualisasikan sebagai informasi data yang mudah diakses. Data yang telah diekstraksi akan dikirimkan ke *Local Center* (LC) Universitas Lampung. Berdasarkan atas latar belakang ini, penulis bermaksud untuk mengumpulkan, menganalisa data, dan menampilkan data, lalu mengumpulkan data ke dalam basis data atau *database*. Hasil akhirnya akan dilakukan visualisasi atau penampilan data informasi yang akan ditampilkan ke *website* agar dapat dikonsumsi publik.

## **1.2 Tujuan Skripsi**

Adapun tujuan dari skripsi ini adalah sebagai berikut:

1. Mendapatkan informasi data dari sensor-sensor yang berada di sekitar embung.
2. Mengintegrasikan dan mengolah data ketiga sensor pada sistem *monitoring* yang digunakan untuk analisa data sensor secara akurat.
3. Menampilkan data melalui *website*.

## **1.3 Manfaat Skripsi**

Adapun manfaat yang diharapkan dari skripsi ini adalah sebagai berikut:

1. Sistem ini akan memberikan manfaat akan pengimplementasian dari teknologi LoRa dan *Big data* dalam membangun *platform* visualisasi, penyimpanan dan

penempatan data, serta analisa kondisi pada embung.

2. Menjadi landasan untuk penelitian selanjutnya dan hasil penelitian ini dapat menjadi referensi dan acuan dalam penelitian berikutnya.

#### **1.4 Rumusan Masalah**

Adapun rumusan masalah dari skripsi ini adalah sebagai berikut:

1. Bagaimana hasil informasi yang didapatkan saat diukur menggunakan sensor-sensor, Arduino serta LoRa?
2. Bagaimana penerapan sistem terintegrasi *big data* sehingga mampu melakukan pengolahan data?
3. Bagaimana cara menampilkan data hasil menggunakan *website*?

#### **1.5 Batasan Masalah**

Adapun batasan masalah dari skripsi ini adalah sebagai berikut:

1. Data yang digunakan dalam skripsi ini adalah data dari sensor-sensor yang diletakkan di sekitar lereng embung.
2. Tidak membahas secara rinci tentang sistem komunikasi LoRa.
3. Semua sensor tidak dikalibrasi.
4. Penelitian menggunakan pemrograman Arduino IDE, HTML, Pentaho *Data Integration*, php, *mysql* serta 000webhost.
5. Perancangan sistem diasumsikan sampai ke visualisasi informasi di mana informasi berupa data yang kemudian akan ditampilkan ke *web*.

## **1.6 Sistematika Penulisan**

Adapun sistematika penulisan dari skripsi ini adalah sebagai berikut:

### **BAB I. PENDAHULUAN**

Bab ini berisi latar belakang atau dasar pengkajian, tujuan dari suatu penulisan, manfaat dari penulisan, rumusan dari masalah, batasan dari masalah, serta sistematika dalam penulisan.

### **BAB II. TINJAUAN PUSTAKA**

Bab ini berisi tinjauan berupa kajian dan literatur penelitian sebelumnya yang berkaitan dengan topik skripsi.

### **BAB III. METODOLOGI PENELITIAN**

Bab ini berisi tentang waktu dan tempat penelitian, alat dan bahan yang digunakan dan metode penelitian yang digunakan.

### **BAB IV. HASIL DAN PEMBAHASAN**

Bab ini berisi hasil penelitian yang dikerjakan dan pengkajian dari hasil penelitian tersebut.

### **BAB V. SIMPULAN DAN SARAN**

Bab ini berisi tentang simpulan dari seluruh penelitian yang dilakukan dan saran untuk penelitian selanjutnya.



## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Penelitian Terkait**

Penelitian dalam riset skripsi berfokus terhadap pengkajian penelitian ataupun konsep-konsep sebelumnya tentang *monitoring system* dan pengolahan data. Penulis [1] melakukan penelitian dengan judul penelitian tentang Pengembangan Sistem *Real Time Monitoring* dan Peringatan Dini Longsor Berbasis Risiko oleh Luqmanul Hakim, dkk. Sistem *real-time monitoring* longsor mencakupi tolak ukur dari curah hujan, kelembapan tanah, kemiringan tanah, serta perpindahan tanah. Parameter tersebut diukur menggunakan sensor *tipping bucket*, *soil moisture*, *tiltmeter* serta *extensometer*. Pengujian sistem menciptakan informasi parameter yang terukur bisa ditampilkan serta membagikan data peringatan dini longsor buat segala status tingkatan resiko. Sistem pengolah informasi memakai Raspberry Pi 3B serta Mikro SD *Card* selaku penyimpanan informasi. Penulis [2] mengerjakan penelitian dengan suatu judul penelitian tentang Perancangan Sistem Buoy dan Sensor Sebagai Perangkat Pemantauan Lingkungan oleh Wahyu Purwanta. Penelitian mempersoalkan perencanaan *buoy* serta sensor pada waduk sungai dan laut yang terdiri dari peralatan *buoy* dan pusat pengolahan data yang saling tersambung menggunakan radio telemetri. Perencanaan dan peningkatan

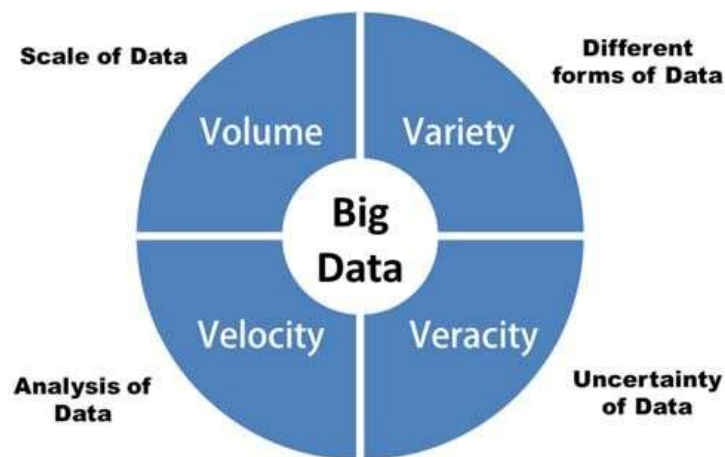
ditunjukkan pada aspek penggunaan dan kinerja sistem *buoy* besar. Penulis [3] menyinggung penelitian mengenai perancangan bangun peringatan bahaya dari longsor dan pemantauan pergeseran pada tanah yang memakai komunikasi berbasis GSM oleh Joko Priyanto, dkk. Penelitian ini nantinya hendak bertujuan merancang dan membuat sistem telemetri perpindahan tanah pada daerah rawan longsor secara digital yang dapat dijadikan petunjuk untuk mengidentifikasi jarak geseran tanah yang terhubung berdasarkan waktu. Data perpindahan tanah yang dihasilkan hendak ditampilkan pada fitur komputer sehingga *user* dapat memonitor dan mengidentifikasi bahaya longsor pada daerah yang sedang dipantau.

## **2.2 *Monitoring System***

*Monitoring system* merupakan proses pengumpulan serta analisis data yang sistematis yang berkesinambungan menimpa apa yang sesungguhnya terjadi sepanjang proses implementasi ataupun pelaksanaan program. *Monitoring system* ialah proses teratur mengoleksi informasi juga pengukuran atas objek program ataupun memantau pergantian. Dengan terdapatnya sistem *monitoring*, data bisa dikenal seluruh orang yang butuh mengenali tentang data sesungguhnya serta supaya keputusan bisa diambil pada waktu yang pas. Pemantauan program mempunyai banyak tujuan, bergantung pada tipe pemantauan serta konteksnya. Tetapi, tujuan utamanya mengumpulkan data serta setelah itu digunakan untuk membuat penyesuaian menunjang pengelolaan program [4]. *Monitoring system* digunakan agar mendapatkan pengukuran informasi, terdiri dari monitor yang dioperasikan untuk menangkap sesuatu peristiwa. *Monitoring system* dikonfigurasi pada bagian pendeteksi untuk mengetahui objek [5].

### 2.3 *Big Data*

*Big data* ialah koleksi informasi yang besar di mana di dalam *big data* dimensi informasi yang dihasilkan besar, mempunyai sebagian tipe informasi, serta informasi mengalir dengan kilat. Salah satu metode buat menganalisa big informasi ialah informasi *mining*. Teknologi *big data* manajemen penyimpanan informasi yang berasal dari *node sensor* [6]. Menurut Min Cen, dkk, 2014. Secara universal *big data* bisa dimaksud selaku sekumpulan data yang ditinjau dari ukurannya yang sangat besar (*volume*), *rapid* pertumbuhannya (*velocity*), informasi yang bermacam-macam dalam bermacam wujud/format (*variety*), dan mempunyai nilai tertentu(*value*).



Gambar 2.1. *Big data* 4V [6].

Big data ialah tata cara menganalisis kumpulan informasi besar semacam *machine learning* yang mengganti aplikasi ilmiah dibanyak bidang. Melalui adanay sistem *big data*, kumpulan informasi yang besar sanggup diolah secara mudah. *Big data* ialah koleksi informasi yang *volume* informasinya luar biasa besar, mempunyai keragaman sumber informasi yang besar, sehingga butuh dikelola dengan tata cara serta fitur bantu yang kinerjanya cocok [7].

*Big data* mengolah data bervolume besar, berkecepatan besar, serta berbagai macam yang menuntut wujud pemrosesan data yang hemat serta inovatif pengetahuan, pengambilan keputusan, serta otomatisasi proses. Teknologi *big data* punya kemampuan dalam menangani bermacam-macam bentuk data yang berbeda-beda. Umumnya terdapat dua kelompok data yang perlu diolah, yaitu :

1. Data terstruktur

Grup data dengan tipe, format, dan struktur data yang ditentukan. Sumber data termasuk data transaksi, data OLAP, RDBMS tradisional, *file* CSV, dan *spreadsheet*.

2. Data tidak terstruktur

Sekelompok data tekstual yang tidak memiliki format yang tidak beraturan atau struktur yang unik. Oleh karena itu, mengubahnya menjadi data terstruktur membutuhkan lebih banyak upaya, alat, dan waktu. Data ini dihasilkan oleh aplikasi internet seperti data log URL, media sosial, email, blog, video, dan audio [8].

Dalam pengelolaan *Big Data*, terdapat perangkat bantu (*tools*) yang dimanfaatkan untuk membantu proses pada tiap tahap pengolahan yaitu :

- a. *Acquired* yang terikat dan berhubungan dengan sumber dan cara mendapatkan data.
- b. *Accessed* terikat kinerja akses data. Data memerlukan susunan kelola, pengabungan, penyimpanan, dan komputasi agar dikelola di tingkat berikutnya dan diproses dengan Hadoop, Nvidia CUDA, Twitter Storm, dan GraphLab. *Data storage* memakai *storage* Neo4J, Titan, dan juga HDFS.

- c. *Analytic* terkait dengan informasi yang diambilnya, tetapi hasil pengelolaan data telah diproses. Analisis bersifat deskriptif (penyajian data), diagnostik (mencari sebab dan akibat berdasarkan data), prediktif (memprediksi kejadian yang akan datang), dan analisis preskriptif (pemilihan dan dampak dari setiap opsi yang direkomendasikan).
- d. *Application* berpautan dengan visualisasi, *reporting* hasil dari analitik [9].

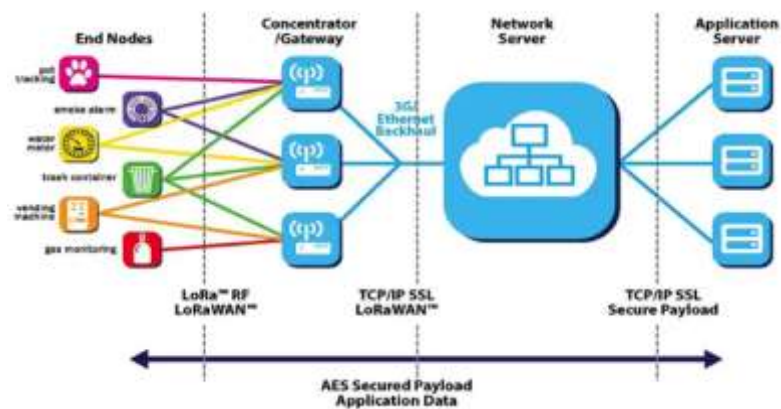
#### **2.4 LoRa (Long Range)**

LoRa (*Long Range*) adalah bentuk modulasi unik yang dikembangkan oleh Semtech. Modulasi yang dipakai menggunakan modulasi *Frequency Modulation*. Berfokus menghasilkan skala frekuensi cukup stabil. *Phase Shift Keying*, *Frequency Shift Keying*, dll juga dapat digunakan sebagai pola transmisi. LoRa (*long range*) adalah proses merubah gelombang periodik tertentu menjadi sinyal yang dapat mengirimkan informasi. Teknologi LoRa menyediakan *runtime* jangka panjang yang sangat menarik. Seperti konsumsi *power* yang cukup rendah dan transmisi data yang aman. Jaringan publik yang menggunakan jaringan LoRa dapat memberikan jangkauan sinyal yang lebih luas. LoRa adalah jaringan dengan cakupan yang lebih luas daripada jaringan seluler biasa. Selain itu, dapat dengan mudah menyesuaikan dengan kebutuhan. Kebutuhan ini mirip dengan perangkat elektronik lain yang menggunakan pengaturan nirkabel [10].

Teknologi menyajikan dua bagian susunan yaitu susunan fisik LoRa dan protokol komunikasi yang dibangun di atas susunan lapisan fisik LoRa yang mendasarinya. Protokol komunikasi sumber terbuka yang ditentukan oleh lapisan komunikasi



LoRaWAN (*Long Range Wide Area Network*). LoRa dan LoRaWAN memungkinkan konektivitas jarak jauh yang hemat biaya untuk perangkat IoT di industri pedesaan, terpencil, dan lepas pantai. Jaringan LoRaWAN yang digunakan dalam topologi bintang-bintang memiliki stasiun pangkalan yang menyampaikan data *antara node sensor* dan server jaringan. Komunikasi antara *node sensor* dan *base station* melalui saluran nirkabel menggunakan lapisan fisik LoRa, sedangkan koneksi antara gateway dan server pusat ditangani melalui jaringan berbasis backbone IP [11].



Gambar 2.2. Arsitektur Lora [11].

## 2.5 Arduino Uno

Arduino Uno yakni komputer mini atau disebut juga sebagai *microcontroller* yang punya 14 pin masukan dan keluaran di mana *input* 6 pin tadi bisa dipakai menjadi hasil PWM & masukan *analog* 6 pin, 16 MHz osilator kristal, sambungan USB, *jack power*, *ICSP header*, dan *reset button*. Agar papan mikrokontroler bisa dipakai, maka diperlukan untuk menghubungkan papan Arduino ke personal komputer memakai kabel USB [12].



Gambar 2.3. Arduino Uno [14].

### 2.5.1 Arduino IDE

Arduino IDE adalah akronim atas *Integrated Development Environment*, atau secara umum adalah lingkungan terprogram yang dipakai untuk melakukan pemrograman. Disebut seperti *environment* lantaran melalui aplikasi Arduino menjalankan tata olah program untuk melaksanakan perintah serta fungsi-fungsi di *running* dengan sintaks pemrograman. Arduino memakai bahasa program sendiri menyerupai bahasa C. Bahasa pemrograman Arduino (*Sketch*) telah dilakukan perubahan untuk memudahkan pemula melakukan pemrograman menurut bahasa aslinya. IC mikrokontroler Arduino ditanamkan pada suatu perangkat dan peralatan yang disebut atau bernama *bootlader* yang bertujuan untuk menjadi penengah antara *compiler* Arduino menggunakan *mikrokontroler*. Program diketik menggunakan aplikasi sketsa Arduino Software (IDE). *Sketch* ditulis pada suatu editor teks & disimpan pada arsip menggunakan ekstensi *.ino*. Penyunting teks pada *software* mempunyai fitur misalnya memotong atau menempel (*copy paste*) dan mencari atau mengganti yang memudahkan menyunting kode di *sketch*. Pada *Software* Arduino IDE terdapat kotak pesan hitam berperan dalam menampilkan status, misalnya

pesan yang *error*, *compile*, dan *upload* program ke Arduino. Pada elemen bawah kanan *Software* Arduino IDE, menerangkan papan yang tersusun bersama bagian *COM ports* yg dipakai [13].

## **2.6 Sensor**

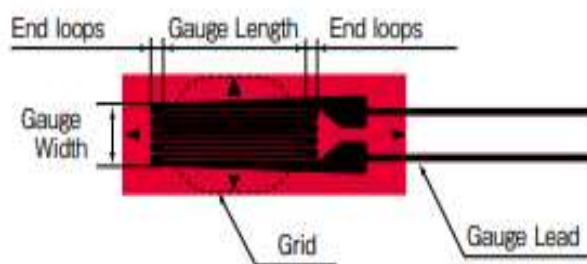
Sensor adalah komponen yang digunakan untuk mendeteksi suatu besaran fisik menjadi besaran listrik, sehingga dapat dianalisa dengan rangkaian listrik tertentu. Sensor yaitu bagian dari *transducer* yang berguna dalam melakukan *sensing* atau pengindraan. Adapun perubahan energi *external* yang masuk kebagian *input transducer*, sehingga perubahan kapasitas energi yang ditangkap segera dikirim ke bagian konvertor dari *transducer* untuk diubah menjadi energi listrik. Sensor memberi deskripsi yang cukup jelas tentang status kinerja suatu alat. Sensor memberikan info rinci dari setiap proses dalam suatu operasi dan merupakan bagian integral dari sistem kontrol. Sistem kontrol bertanggung jawab untuk mengoleksi dan menyiapkan data status proses dan kemudian meneruskannya ke prosesor. Setiap jenis sensor mempunyai rentang operasi tertentu. Sebuah sensor mengubah fenomena fisik seperti suhu, tekanan, level, panjang, posisi, atau ada atau tidaknya, menjadi tegangan, arus, frekuensi, pulsa, dan sebagainya [15].

### **2.6.1 Sensor *Strain Gauge***

*Strain Gauge* dibangun dengan membentuk kisi-kisi yang terbentuk dari listrik halus kawat resistansi atau *foil* resistansi logam yang terukir secara fotografis pada dasar isolasi listrik (*backing*) dan *lead gauge*. *Strain Gauge* sangat ideal untuk mengukur nilai regangan di tempat yang tidak mudah dijangkau. Hambatan listrik suatu logam

berubah secara proporsional dengan deformasi mekanis yang disebabkan oleh gaya eksternal yang diterapkan pada logam. Dengan mengikat logam tipis ke objek pengukuran melalui sebuah isolator listrik tipis, logam berubah bentuk tergantung pada deformasi benda yang diukur dan hambatan listriknya berubah.

*Strain gauge* ialah sensor untuk mengukur perubahan nilai resistansi dan mengukur nilai regangan. Pengukur regangan juga dapat mengukur tekanan. Hambatan listrik dari konduktor seperti kawat sebagian bergantung pada luas penampangnya. Semakin kecil penampang, semakin besar atau tinggi resistansi kawat. Pengukur regangan adalah kabel konduktif yang meregang di bawah pengaruh mekanis (ketegangan, tekanan, atau torsi) dan mengubah resistansinya dapat diverifikasi. Kawat dipasang pada penyangga dan penyangga dipasang pada objek yang akan diukur. Di sisi lain, kompresi linier yang meningkatkan penampang pengukur regangan mengurangi resistensi. Jika memasang pengukur regangan di suatu area, itu akan mengikuti pergerakan area tersebut. Pengukur regangan mengembang dan menyusut berdasarkan pada fleksibilitas area [15].

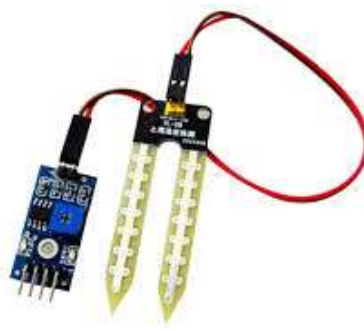


Gambar 2.4. *Strain Gauge Sensor* [15].

Ketika regangan dihasilkan dalam objek pengukuran, regangannya ditransfer ke kawat resistansi atau *foil* pengukur nilai regangan melalui dasar pengukur (*backing*). Akibatnya, kawat atau *foil* mengalami perubahan resistensi. Untuk pengukuran pengukur regangan, rangkaian *Wheatstone Bridge* dipakai untuk merubah perubahan nilai resistansi pengukur regangan menjadi nilai tegangan keluaran.

### 2.6.2 Sensor *Soil Moisture* YL-69

Sensor *soil* berupa alat atau sensor yang dapat mendeteksi atau menentukan suatu nilai kelembaban di dalam tanah. Sensor ini sangat sederhana, tetapi ideal untuk memantau kelembaban tanah. Sensor memiliki dua kaki *probe* yang berfungsi untuk mengalirkan listrik melalui tanah dan membaca resistansinya untuk membaca kadar air. Saat air berlebih, tanah menjadi lebih mudah untuk mengalirkan arus listrik (resistansi rendah), dan di tanah kering menjadi sulit untuk menghantarkan listrik (resistansi tinggi). Sensor ini sangat amat berfungsi untuk memperingatkan tentang tingkat kelembaban tanaman dan memantau kelembaban tanah. Sensor biasanya digunakan pendeteksi kelembaban tanah, jadi sangat cocok untuk membangun *system* kelembaban tanah. Sensor diatur oleh dua bagian seperti Gambar 2.5 berupa *electronic board* dan *probe* dengan dua bantalan yang mendeteksi kadar air.



Gambar 2.5. Sensor *Soil Moisture* YL-69 [16].

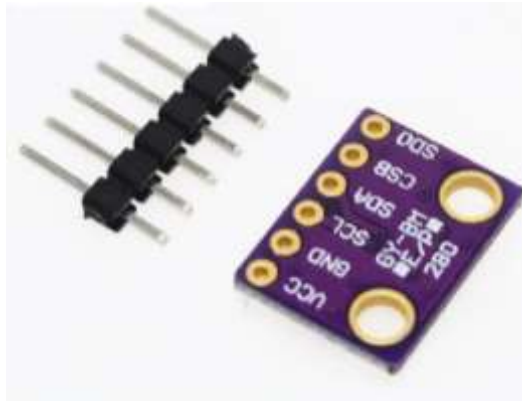
Nilai tegangan yang didapat dari sensor berubah sesuai dengan kandungan air pada tanah. Ketika tanah basah maka tegangan *output* atau keluaran berkurang, sedangkan saat tanah kering maka tegangan *output* atau keluaran meningkat. *Outputnya* bisa berupa sinyal digital (D0) LOW atau HIGH, tergantung kandungan airnya. Jika kelembaban tanah melebihi nilai ambang batas tertentu yang telah ditentukan, modul *output* LOW, dan jika tidak maka *output* HIGH. Ambang batas nilai untuk sinyal digital dapat disesuaikan menggunakan potensiometer. *Outputnya* bisa berupa sinyal analog sehingga akan mendapatkan nilai antara 0 dan 1023.

Sensor kelembaban tanah memberi tahu pengguna saat tanah terlalu kering atau terlalu lembab. Dikembangkan pada platform Arduino, perangkat ini mempunyai kegunaan untuk berkomunikasi secara *wireless* dengan perangkat lain dan menggunakan modul GPS *built-in* untuk mengukur kelembaban tanah dan garis lintang dan bujur lokasi dengannya [16].

### **2.6.3 Sensor BMP280**

BMP280 merupakan sensor tekanan barometrik mutlak yang dibuat terkhusus untuk mengukur nilai suhu atau *temperature*, tekanan udara atau *air pressure*, dan juga *altitude*. Modul sensor ditempatkan di LGA 8-pin yang memiliki luas sebesar  $2,0 \times 2,5$  mm<sup>2</sup> dan tinggi 0,95 mm. BMP280 memiliki ukuran yang sangat kecil dan konsumsi dayanya yang rendah sebesar 2,7 A @1Hz memungkinkan penerapan dalam baterai perangkat yang digerakkan seperti ponsel, modul GPS, atau jam tangan. BMP280 terdiri dari elemen tekanan resistif Piezo dan ASIC. ASIC

melakukan konversi atau perubah nilai analago to digital atau A/D dan memberikan hasil konversi serta spesifik data sensor melalui antarmuka digital. BMP280 memberikan fleksibilitas tertinggi bagi perancang dan dapat disesuaikan dengan kebutuhan mengenai akurasi, waktu pengukuran, dan konsumsi daya.



Gambar 2.6. Sensor BMP280 [17].

## 2.7 Pentaho Data Integration (PDI)

Pentaho Data Integrasi (PDI) berfungsi dalam proses ETL (Ekstraksi, Transformasi, serta *Loading*). PDI berperan ketika untuk hal memigrasi informasi, mensterilkan informasi, *loading* dari *file* ke *database* ataupun kebalikannya dalam *volume* besar. PDI sediakan *Graphical User Interface* (GUI) serta *drag-drop* komponen yang memudahkan *user*. Elemen yang sangat utama pada PDI merupakan *Transformation* serta *Job*. *Transformation* sekumpulan perintah merubah masukan jadi keluaran yang dimau (*input-proses-output*). Sebaliknya *Job* merupakan kumpulan instruksi melaksanakan transformasi. Terdapat 3 komponen dalam PDI: *Spoon*, *Pan* serta *Kitchen*. *Spoon* merupakan *user interface* membuat *Job* serta *Transformation*. *Pan* merupakan *tools* yang berperan membaca, merubah serta menulis informasi. Sebaliknya *Kitchen* mengeksekusi *Job* [18].

## 2.8 Database

*Database* adalah program suatu perangkat komputer yang memberikan pelayanan dengan manajemen *database* dan mengoperasikan komputer atau program aplikasi *database* yang menggunakan model *client/server*. Istilah ini juga menuju kepada komputer (biasanya *server*) yang mengoperasikan program yang berkaitan. Sistem manajemen basis data (SMBD) umumnya menyediakan fungsionalitas atau kinerja berdasarkan *server* berbasis data, dan beberapa SMBD sangat bergantung pada model *server-klien* untuk mengakses basis data. Contoh *database server* adalah MySQL, Microsoft SQL Server, dan Oracle.

*Database* saat ini sangat penting untuk setiap proyek. Mereka digunakan untuk memelihara catatan internal, untuk menyajikan data kepada pelanggan dan klien di *web world wide*, dan untuk mendukung banyak proses komersial lainnya. *Database* pula demikian ditemui selaku inti dari banyak penyelidikan ilmiah. Kekuatan *database* berasal dari kumpulan pengetahuan serta teknologi yang sudah tumbuh sepanjang sebagian dekade serta diwujudkan dalam fitur lunak *Database Management System* ataupun DBMS. DBMS merupakan perlengkapan yang jitu membuat serta mengelola besar jumlah informasi secara efektif serta bertahan dalam jangka waktu yang lama. Sistem ini merupakan salah satu tipe fitur lunak yang sangat familiar di lingkungan yang ada. Kemampuan yang disediakan DBMS kepada pengguna yaitu *persistent storage*, *programming interface* dan *transaction management* [19].



### 2.8.1 MySQL

MySQL adalah sebuah sistem basis data yang disebarluaskan secara bebas menurut lisensi BSD. *Software* ini adalah salah satu *database* yang paling banyak digunakan saat ini selain MySQL dan *Oracle* mysql menyediakan fitur yang berguna untuk replikasi *database*. MySQL menyediakan fitur yang berguna untuk replikasi basis data. Tambahan yang dilayani oleh MySQL yaitu DB Mirror, PGPool, Slony, PGCluster, dan lain sebagainya. MySQL sama dengan struktur *base* data yang umum dan juga kuat untuk urusan pertalian serta terbuka untuk dipakai oleh umum. MySQL menggambarkan sebuah perangkat lunak atau *software* suatu DBMS SQL yang *multithread* dan *multi-user* [20].

### 2.9 Website

*Website* atau situs web yakni halaman web (*webpage*) yang berkoneksi yang umumnya berisi kelompok informasi atau data berupa data teks, gambar, animasi, audio, *video* ataupun gabungan dari semuanya yang umumnya diperuntukkan untuk personal, kelompok, organisasi, dan industri. *Website* dipecah menjadi dua bagian berupa *web* statis dan dinamis. Berkarakter statis atau tetap apabila nilai atau informasi berisi tetap atau *fix* datanya dan berisi hanya berasal dari pemilik *website*, sedangkan *web* yang disebut karakter dinamis apabila informasinya berisi data yang selalu berganti dan dapat diubah oleh pemilik maupun *user* atau pengguna *website*. Penyusun dari *website* diantaranya terdapat *domain name/URL* (*Uniform Resource Locator*), *webhosting*, *script program* dan *maintenance web* [21].

### 2.9.1 *Web Server*

*Web Server* ialah sebuah sistem komputer yang menangani permintaan HTTP, protokol jaringan yang membantu menyampaikan informasi di *World Wide Web*. *Web server* melayani klien *web browser* seperti Google Chrome, Mozilla Firefox, Safari, dan Internet Explorer untuk menampilkan *web page* yang diminta oleh *user*. *Server web* pada dasarnya bertanggung jawab untuk menerima informasi yang berasal dari *browser* (klien) dan mengirimkannya kembali atau *feedback*. *Server web* berada di komputer *server (hosting)* dan *browser web* berada di komputer *user* atau pengguna. *File* situs *web* yang disimpan di komputer *server* yang diakses pengguna. Oleh karena itu, ketika pengguna mengunjungi halaman situs *web* di komputer *server*, *browser web* mengirimkan permintaan HTTP ke *server web*. *Server web* kemudian mencari halaman yang diminta [21].

## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Waktu dan Tempat Penelitian

Adapun waktu dan tempat penelitian untuk skripsi ini adalah sebagai berikut:

Waktu : Agustus 2021 — Desember 2021

Tempat : Laboratorium Teknik Telekomunikasi Jurusan Teknik Elektro  
Fakultas Teknik Universitas Lampung.

#### 3.2 Alat dan Bahan

Adapun perangkat serta bahan-bahan yang dibutuhkan untuk penelitian skripsi terdiri dari perangkat keras dan lunak pada Tabel 3.1 seperti berikut:

Tabel 3.1 Alat dan bahan

No.	Alat dan Bahan	Kegunaan
1	Komputer	Sebagai Pengontrol Arduino Uno dan LoRa
2.	Sensor BMP280	Pembaca Nilai <i>Temperature</i> dan <i>Air Pressure</i>
3.	Sensor <i>Moisture Soil</i> YL-69	Pembaca Nilai Kelembaban Tanah atau Kadar Air pada Tanah
4.	Sensor <i>Strain Gauge</i>	Pembaca Nilai <i>Strain</i> /Regangan
5.	Arduino Uno	Perangkat Mikrokontroler
6.	LoRa 915 Mhz	Perangkat Komunikasi/Pengirim Data
7.	Arduino IDE	<i>Software</i> Pemograman Arduino
8.	Pentaho	Sebagai Aplikasi Pengolah Data
9.	PHPmyAdmin	Penghubung antara PHP dengan MySQL
10.	<i>Webhosting</i>	Menyimpan Data-data pada <i>Website</i>

### 3.3 Metodologi Penelitian

Metode penelitian yang digunakan dalam mencapai tujuan penelitian skripsi yaitu menggunakan beberapa tahapan yang dilakukan sebagai berikut:

#### 3.3.1 Studi Literatur

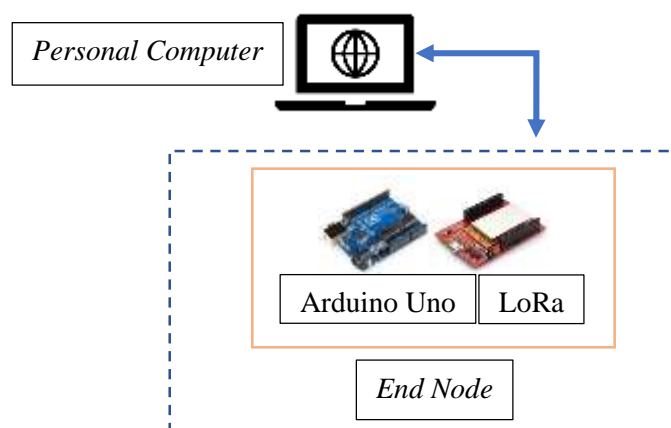
Pada studi literatur dilakukan pencarian informasi terkait topik skripsi, baik dari buku, jurnal, bahan dari internet maupun sumber-sumber lain yang berkaitan dan mendukung dengan skripsi ini. Pengkajian dikhususkan pada integrasi data, pengolahan data, *system* program dan pengiriman dari data pada sampai ke *web*.

#### 3.3.2 Pembuatan Perangkat

Perangkat yang dibangun dan dirancang terbagi menjadi dua jenis, yaitu perangkat pada *end node*, dan perangkat pada *sensor node*.

##### 3.3.2.1 Perangkat *End Node*

Perangkat *end node* bertujuan sebagai titik kumpul data dan sebagai penyimpanan data sementara. Perancangan perangkat *end node* ditunjukkan pada Gambar 3.1.



Gambar 3.1 Arsitektur *End Node*

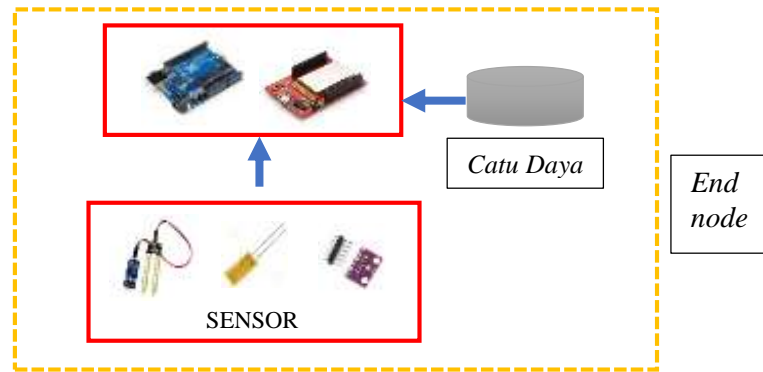
Implementasi *end node* pada penelitian ini menggunakan modul *LoRa* RFM95 beserta Arduino sebagai modul komunikasi LoRa. Modul *LoRa* di inisialisasi menggunakan Arduino IDE. Kedua perangkat itu dihubungkan menyesuaikan dengan *pinout* yang sudah ditentukan. *End node* berperan sebagai *receiver* atau titik kumpul hasil data dari *multisensor*.

Komponen-komponen yang digunakan beserta fungsinya pada *end node* adalah sebagai berikut:

1. Arduino Uno yang berfungsi sebagai mikrokontroler dan jembatan antara LoRa terhadap Komputer.
2. LoRa Shield Dragino berfungsi sebagai perangkat komunikasi *wireless* antara *node sensor* ke *end node*.
3. Komputer berfungsi untuk mengetahui *output* yang diterima oleh *end node* dan juga memeriksa apakah perangkat berjalan dengan baik atau tidak.

### **3.3.2.2 Perangkat *Node Sensor***

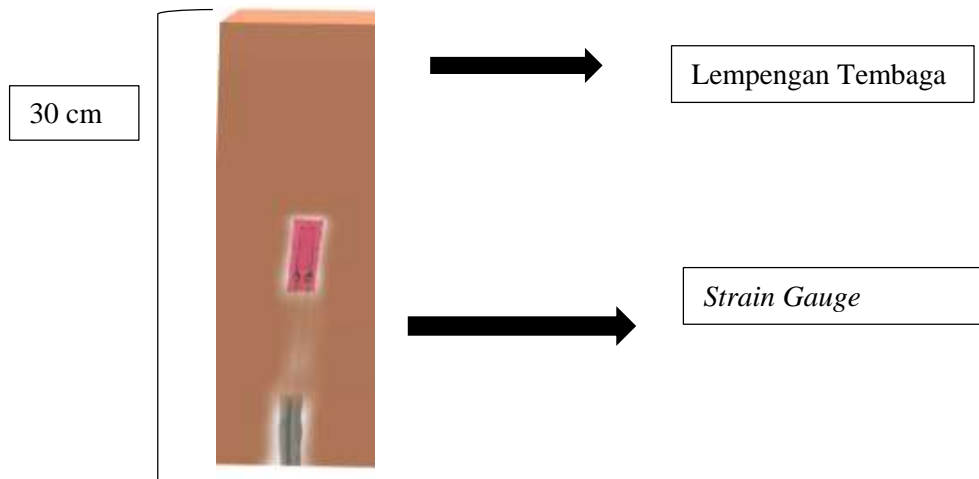
Perangkat *node sensor* bertujuan sebagai pengirim data, juga bertugas untuk mendeteksi kondisi sekitar, seperti kondisi kelembaban tanah, suhu ataupun pergeseran tanah. Perangkat *node sensor* terdiri dari sensor yang dilengkapi dengan Arduino Uno beserta Modul LoRa tipe SX1278 dan juga terhubung dengan *powerbank* sebagai daya terima. Arsitektur perangkat *node sensor* ditampilkan pada Gambar 3.2, di mana sensor data yang digunakan diantaranya adalah Sensor BMP280 yang bertugas untuk membaca nilai *temperature* dan juga *air pressure*, Sensor *Soil Moisture* YL-69 berfungsi untuk membaca nilai kelembaban pada tanah, serta Sensor *Strain Gauge* yang merupakan sensor pembaca nilai regangan.



Gambar 3.2 Arsitektur *Node Sensor*

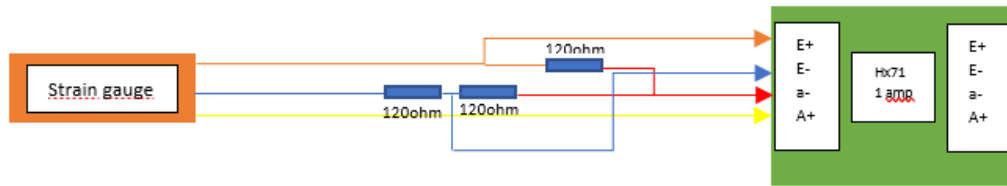
#### a. Perancang Sensor *Strain Gauge*

Awal pengerjaan penelitian melakukan perancangan terhadap regangan pada tanah pada saat pergeseran tanah terjadi. Regangan tersebut diukur menggunakan Sensor *Strain Gauge*, outputnya berupa tegangan, Pengerjaan pertama adalah merakit komponen-komponen yang terdiri dari Sensor *Strain Gauges*, Modul HX711, *Jumper wiring*, Arduino, kabel USB, beserta *LoRa* menjadi *inputan* ke Arduino. Instalasi dilakukan pada sensor tegangan regangan yang diletakkan pada batang tembaga atau *beam* dengan panjang tiap lempengan adalah 30 cm. Sensor yang diletakkan pada *beam* atau lempengan tembaga merupakan Sensor *Strain Gauges* yang dapat digunakan untuk mengukur regangan. Tipe Sensor *Strain Gauges* yang digunakan yaitu model FLA-5-11-5LT yang berukuran 5 mm, dengan besaran hambatan sebesar 120  $\Omega$  dan memiliki 3 buah kabel bawaan yang terpasang sepanjang 5 m.



Gambar 3.3 Sensor *Strain Gauge* yang Terpasang pada *Beam*.

Diketahui pada Gambar 4.5 bahwa *beam* yang digunakan berukuran 30 cm untuk masing-masing *Strain Gauge*. Lempengan yang digunakan merupakan lempengan tembaga dengan ketebalan 0.7 mm, panjang 5 cm. *Strain Gauge* diletakkan ditengah-tengah *beam* (titik ke 15 cm) sehingga saat beban atau *load* ditempatkan pada sisi kanan dan kiri *beam* maka *Strain Gauge* akan meregang dan *output* akan terlihat sementara pada Serial Monitor Arduino IDE. Pada *Strain Gauge* yang dipakai menggunakan lempeng tembaga sebagai bidang pengukuran dengan nilai Modulus Elastis sebesar  $1.1 \times 10^5$  mpa, *poisson rasio* untuk tembaga sebesar 0.355. Pada pengukuran *Strain Gauge* pada lempengan tembaga menggunakan penguat instrumentasi yaitu Modul HX711 yang berfungsi untuk mengkonversi perubahan yang terukur dalam perubahan resistansi dan mengkonversinya ke dalam besaran tegangan yang nantinya besaran ini diteruskan ke Arduino Uno.



Gambar 3.4 Rangkaian Instalasi Sensor *Strain Gauge* Terhadap hx711

Gambar 3.4 menunjukkan rangkaian yang dirancang untuk Sensor *Strain Gauge*. Terdapat rangkaian jembatan *Wheatstone* yang berfungsi untuk menemukan nilai resistensi atau hambatan terhadap Modul HX711. *Strain Gauge* diberi *amplifier* bertujuan untuk penguatan sinyal *output* dari *strain gauge*.

#### **b. Perancang Sensor BMP280**

Pada tahap awal pengerjaan penelitian dilakukan perancangan dengan merakit komponen-komponen yang terdiri dari Sensor BMP280, *Jumper Wiring*, *USB Cable*, *Arduino*, dan *LoRa*. Sensor BMP280 digunakan untuk mendeteksi nilai suhu yang memiliki satuan *Celcius* dan juga mengukur tekanan udara dengan nilai satuan hPa. Instalasi pertama yang dilakukan pada Sensor BMP280 adalah terhadap *Arduino Uno* yang *outputnya* diatur dengan *delay* waktu sebesar 30 detik sekali.

#### **c. Perancang Sensor *Moisture Soil* YL-69**

Instalasi dilakukan pada sensor *Moisture Soil* yang diletakkan atau ditanam ke dalam tanah. Sensor Kelembaban Tanah terdiri dari dua kaki, yang masing-masing kaki memiliki fungsi untuk mengukur kadar air di dalam tanah. Pin *header* pada sensor yang terhubung ke rangkaian *Amplifier* lalu terhubung ke *Arduino*. *Amplifier* memiliki Pin Data *Vin*, *Gnd*, *Analog* dan *Digital*. *Output* analog berupa tegangan



yang akan meningkat seiring dengan peningkatan tingkat kelembaban tanah. Sensor kelembaban tanah terdiri dari dua *probe* yang digunakan untuk mengukur kandungan volumetrik air. Kedua *probe* memungkinkan arus melewati tanah dan kemudian mendapat nilai resistansi untuk mengukur nilai kelembaban. Ketika ada air, tanah akan lebih banyak menghantarkan listrik yang berarti hambatannya akan berkurang. Karena itu, tingkat kelembapannya akan lebih tinggi. Tanah yang kering menghantarkan listrik dengan buruk, sehingga ketika air lebih sedikit, maka tanah akan lebih sedikit menghantarkan listrik yang berarti akan lebih banyak hambatan. Oleh karena itu, tingkat kelembaban akan lebih rendah. Sensor ini dapat dihubungkan dalam dua mode yaitu mode analog dan mode digital. Nilai persentase humiditas atau kelembaban pada Sensor *Moisture Soil* diambil dengan pengambilan data setiap 60 detik sekali. *Output* dari sensor berubah pada rentang nilai ADC dari 0 hingga 1023 sehingga direpresentasikan sebagai nilai kelembaban dalam persentase.

### **3.3.3 Perancangan Basis Data Sementara**

Perancangan basis data menggunakan *Database MySQL* yang memiliki fungsi sebagai tempat penyimpanan data yang berasal dari sensor-sensor yang sebelumnya sudah diinisialisasi. Data informasi tersebut akan ditampilkan ke dalam *website* yang dibuat setelah diintegrasikan. Sebelum pengguna membuka *database*, akan ada tampilan untuk *login* seperti memasukkan *username* dan *password database* sebagai *userdata*. Data yang berasal dari sensor-sensor akan disimpan dan masuk ke dalam *database* untuk sementara. Pada menu *database*, *user* dapat menghapus data jika diinginkan.

### **3.3.4 Perancangan Pengolah Data**

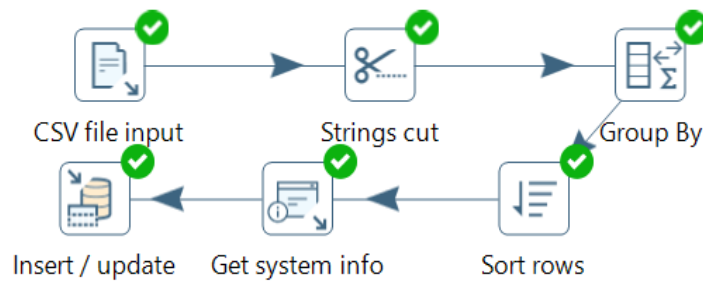
Data yang berasal dari *local database* dikirimkan ke dalam *software* Pentaho. Data lalu diolah dan diintegrasikan secara otomatis oleh Pentaho. Data yang acak dan besar ditransformasikan ke dalam Pentaho yang bertujuan untuk mendapatkan atribut yang diinginkan dengan format baru sehingga didapatkan informasi baru yang diperoleh dari pengolahan data. Pada tahap pengolahan data, semua data yang berasal dari *database* sementara kemudian dikirimkan ke dalam *table transformation* Pentaho. Data yang masuk berupa data Sensor BMP, Sensor *Strain Gauges*, dan Sensor *Moisture Soil*. Masing-masing sensor memiliki *transformation* agar data dapat diintegrasikan sesuai kebutuhan dan keperluan. Setelah seluruh data ditransformasikan, maka akan didapatkan nilai analisis data. Data yang telah diproses dan diolah pada tahap ini selanjutnya akan dikirimkan ketahap visualisasi menggunakan portal *website*. *Web* akan dibangun menggunakan pemograman PHP, AdminLte, dan html sehingga data dapat ditampilkan secara grafik dan tabel.

#### **3.3.4.1 Proses Pengolahan Data Pentaho**

Pada proses pengolahan data pada aplikasi Pentaho, proses dibagi menjadi beberapa tahap dan proses untuk tiap sensor yang akan diintegrasikan sebagai berikut.

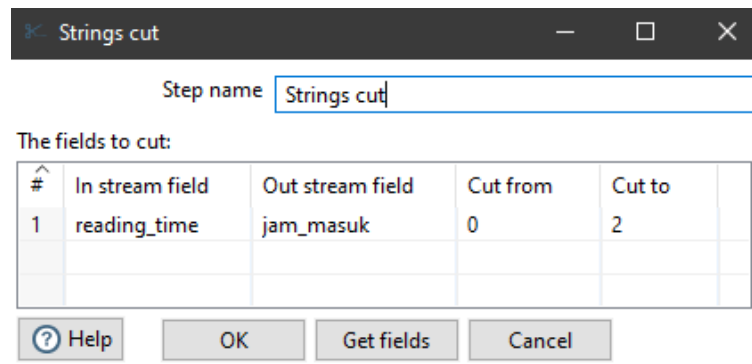
##### **a. Integrasi Data Sensor BMP**

Data pada Sensor BMP berasal dari *database* sementara yang selanjutnya dikirimkan ke dalam *software* Pentaho. Data ditransformasikan terlebih dahulu agar data dapat diproses dan diintegrasikan sebelum divisualisasikan. Tahap pertama yang dilakukan adalah membangun blok-blok yang ditunjukkan pada Gambar 3.5.



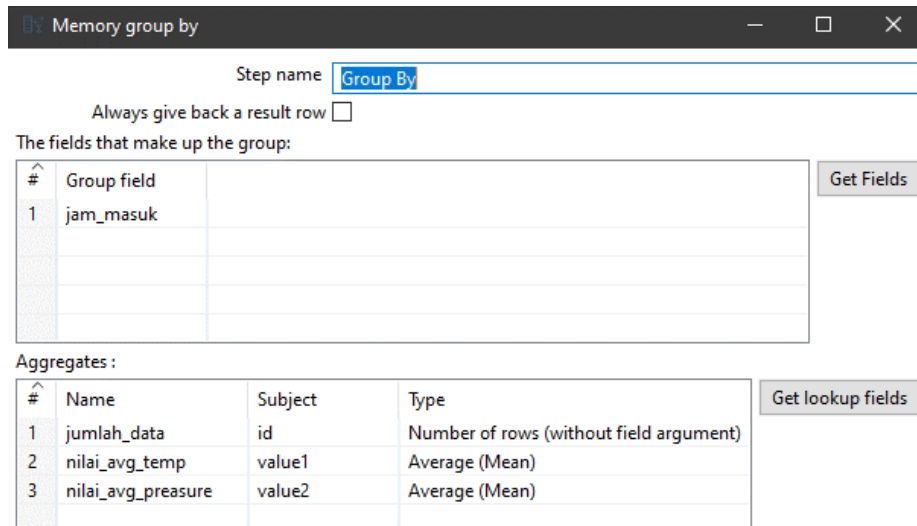
Gambar 3.5 Tampilan Blok-Blok *Transformation* terhadap Data Sensor BMP.

Pada Gambar 3.5 menunjukkan tahap *transformation* terhadap data Sensor BMP. Setiap blok-blok diagram memiliki fungsi dan seleksi data serta pengelompokan data. Blok *table input* berisi data *raw* yang berasal dari *database* sementara. Tahap pertama yang dilakukan adalah membuat koneksi terlebih dahulu dengan *database* agar data dapat ditarik.



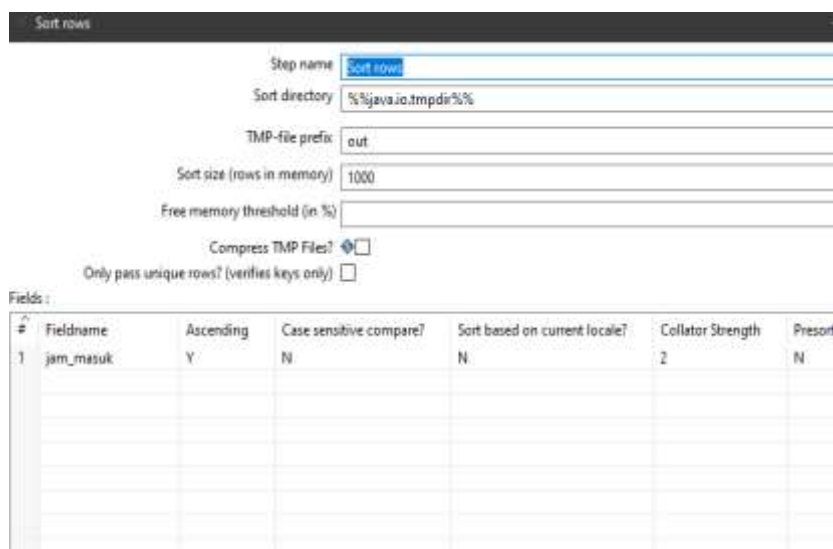
Gambar 3.6 Tampilan Perintah *String Cut*.

Gambar 3.6 menunjukkan proses pada *strings cut* pada data tabel Sensor *Moisture Soil*. *Strings cut* berfungsi untuk memotong data pada suatu kolom tabel data atau *substring* data sehingga didapatkan nilai keluaran baru. Pemotongan berdasarkan urutan angka pada karakter dimulai dari angka 0.



Gambar 3.7 Tampilan Perintah *Group By* Data Sensor BMP.

Gambar 3.7 menunjukkan proses *group by* terhadap data Sensor BMP. Proses *group by* bertujuan untuk mengelompokkan data. Pada tahap ketiga ini, proses transformasi yang dilakukan berfungsi untuk menghitung jumlah data yang dihasilkan dari memfilter subjek id dengan *type field* yang berupa jumlah baris pada tabel.



Gambar 3.8 Tampilan Perintah *Sort Row* Data Sensor BMP

Gambar 3.8 menunjukkan langkah berikutnya yaitu proses *sort row* yang berguna untuk mengurutkan baris berdasarkan *fields* yang ditentukan dan apakah baris harus diurutkan dalam urutan menaik atau menurun sehingga dari data yang tersusun dimulai dari nilai data yang terkecil yang tersusun ke data nilai data yang terbesar.

Step name: Insert / update  
 Connection: koneksi  
 Target schema: id18322286\_dbs  
 Target table: data\_sum\_bmp\_sensor\_2  
 Commit size: 100  
 Don't perform any updates:

The key(s) to look up the value(s):

#	Table field	Comparator	Stream field1	Stream field2
1	jam_masuk	=	jam_masuk	

Update fields:

#	Table field	Stream field	Update
1	jam_masuk	jam_masuk	Y
2	jumlah_data	jumlah_data	Y
3	nilai_avg_temp	nilai_avg_temp	Y
4	nilai_avg_preasure	nilai_avg_preasure	Y
5	tanggal_update	tanggal_update	Y

Gambar 3.9 Tampilan *Insert/Update* pada *Output Data Sensor BMP*.

Pada tahap terakhir, data final akan dikirimkan ke dalam *database*. Gambar 3.9 menunjukkan tampilan *table fields* yang siap dikirimkan ke dalam *database*. Sebelum data final masuk ke dalam *database*, *target schema* ditentukan berdasarkan nama *database* yang dibuat pada *hosting*. Agar data dapat ditarik ke dalam *database*, maka terlebih dahulu harus membuat tabel pada *database* dengan *script* yang ditunjukkan pada Gambar 3.10.

```

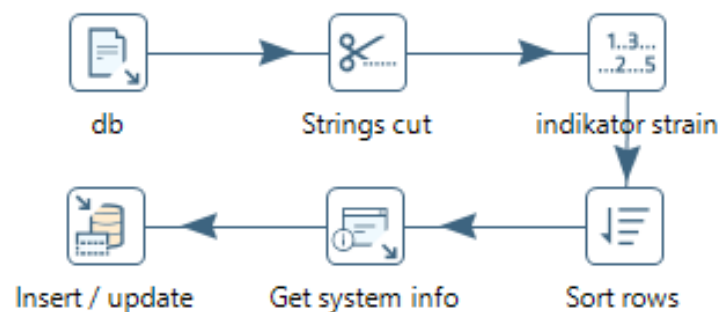
30 CREATE TABLE `data_sum_bmp_sensor_2` (
31   `tanggal_masuk` char(10) NOT NULL,
32   `jam_masuk` char(2) NOT NULL,
33   `jumlah_data` int(11) DEFAULT NULL,
34   `nilai_avg_temp` decimal(15,3) DEFAULT NULL,
35   `nilai_avg_preaure` decimal(15,3) DEFAULT NULL,
36   `tanggal_update` varchar(19) DEFAULT NULL
37 ) ENGINE=MyISAM DEFAULT CHARSET=utf8mb4 ROW_FORMAT=DYNAMIC;
38

```

Gambar 3.10 Script Tabel untuk Data Sensor BMP ke dalam Database.

### b. Integrasi Data Sensor *Strain Gauge*

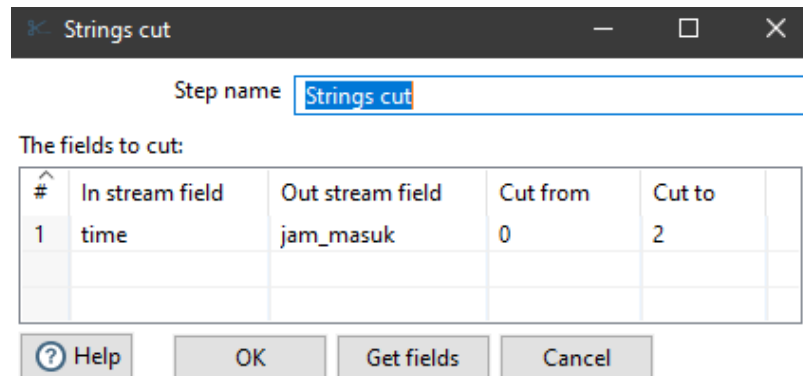
Data pada Sensor *Strain Gauge* juga berasal dari *database* sementara yang dikirimkan ke *software* Pentaho. Sebelum data divisualisasikan, data ditransformasikan terlebih dahulu agar data dapat diproses dan diintegrasikan. Tahap awal yang dilakukan adalah membangun blok-blok yang ditunjukkan pada Gambar 3.11.



Gambar 3.11 Tampilan Blok-Blok *Transformation* Terhadap Data Sensor *Strain Gauge*.

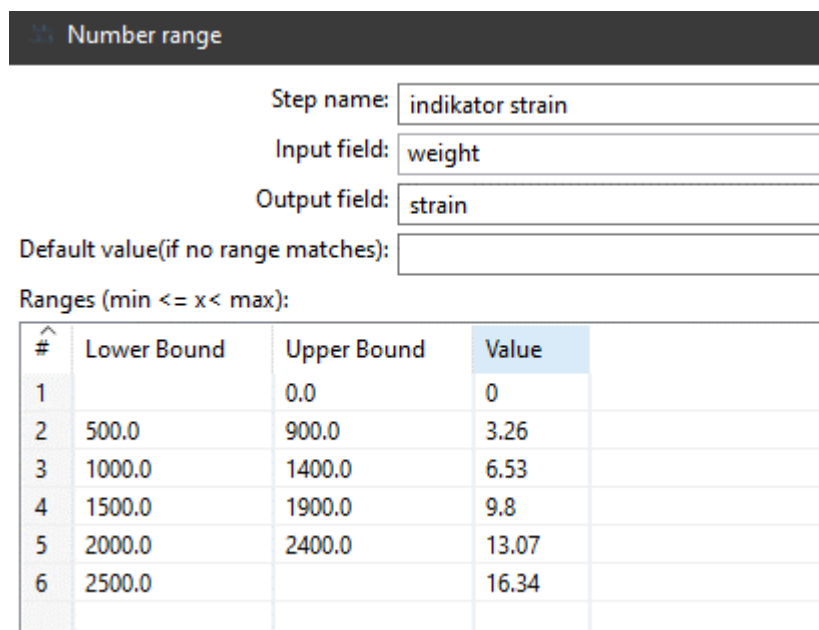
Pada Gambar 3.11 menunjukkan tahap *transformation* terhadap data Sensor *Strain Gauge*. Setiap blok-blok diagram memiliki fungsi dan seleksi data serta pengelompokan data. Blok *table input* atau db berisi data *raw* yang berasal dari

*database* sementara. Tahap pertama yang dilakukan adalah membuat koneksi terlebih dahulu dengan *database* agar data dapat ditarik.



Gambar 3.12 Tampilan Tahap Perintah *Strings Cut* Data Sensor *Strain Gauge*.

Gambar 3.12 menunjukkan proses pada *strings cut* pada data tabel Sensor *Moisture Soil*. *Strings cut* berfungsi untuk memotong data pada suatu kolom tabel data atau *substring* data sehingga didapatkan nilai keluaran baru. Pemotongan berdasarkan urutan angka pada karakter dimulai dari angka 0.



Gambar 3.13 Proses *Number Range* Data Sensor *Strain Gauge*.

Gambar 3.13 menunjukkan proses *number range* yang bertujuan untuk mendapatkan nilai *strain* berdasarkan beban yang diterima oleh *Strain Gauge*. Nilai *strain* berasal dari perhitungan antara nilai beban atau berat aksial yang sudah dikonversi ke dalam nilai tekanan dibagi dengan nilai luas area penampang tembaga. Nilai *strain* tersebut merupakan deformasi yang terjadi pada pada *beam* sesuai dengan perhitungan dan pengukuran.

Insert / update

Step name: Insert / update

Connection: koneksi

Target schema: id18322286\_dbs

Target table: data\_sum\_strain\_gauge\_2

Commit size: 100

Don't perform any updates:

The key(s) to look up the value(s):

#	Table field	Comparator	Stream field1	Stream field2
1	time	=	time	
2				

Update fields:

#	Table field	Stream field	Update
1	time	time	Y
2	weight	weight	Y
3	volt	volt	Y
4	jam_masuk	jam_masuk	Y
5	strain	strain	Y
6	tanggal_update	tanggal_update	Y

Gambar 3.14 Proses *Insert/Update* pada *Output* Data Sensor *Strain Gauge* ke dalam *Database*.



Pada tahap terakhir, data final akan dikirimkan ke dalam *database*. Gambar 3.14 menunjukkan tampilan *table fields* yang siap dikirimkan ke dalam *database*. Sebelum data final masuk ke dalam *database*, *target schema* ditentukan berdasarkan nama *database* yang dibuat pada *hosting*. Agar data dapat ditarik ke dalam *database*, maka terlebih dahulu harus membuat tabel pada *database* dengan *script* yang ditunjukkan pada Gambar 3.15.

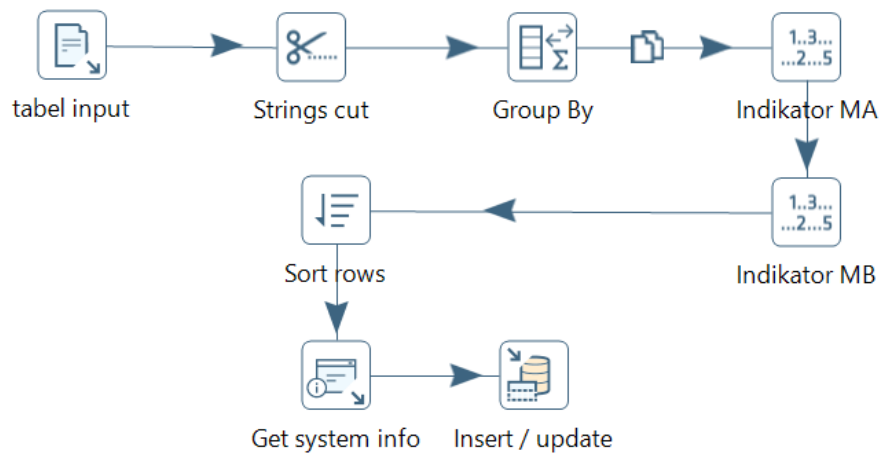
```
26 | --
27 | -- Table structure for table `data_sum_strain_gauge_2`
28 | --
29 |
30 | CREATE TABLE `data_sum_strain_gauge_2` (
31 |   `time` varchar(19) NOT NULL,
32 |   `weight` decimal(15,0) DEFAULT NULL,
33 |   `volt` decimal(15,0) DEFAULT NULL,
34 |   `jam_masuk` int(10) DEFAULT NULL,
35 |   `strain` float(15,2) DEFAULT NULL,
36 |   `tanggal_update` varchar(19) DEFAULT NULL
37 | ) ENGINE=MyISAM DEFAULT CHARSET=utf8mb4 ROW_FORMAT=DYNAMIC;
38 |
39 | --
40 | -- Indexes for table `data_sum_strain_gauge_2`
```

Clear   Format   Get auto-saved query

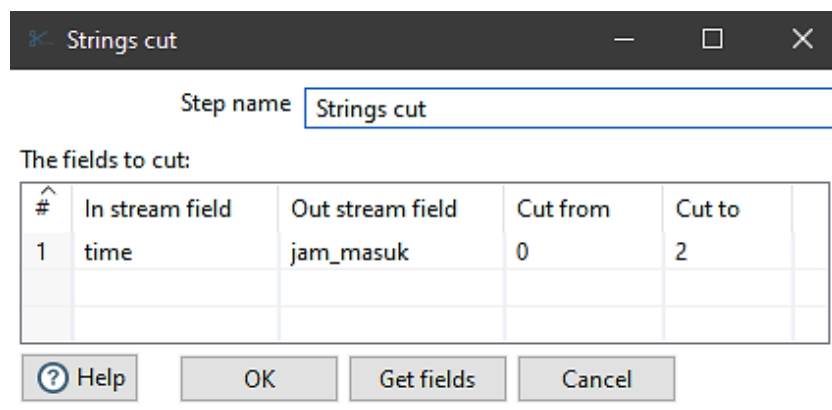
Gambar 3.15 *Script* Tabel untuk Data Sensor *Strain Gauge* ke dalam *Database*.

### c. Integrasi Data Sensor *Moisture Soil*

Data pada Sensor *Moisture* berasal dari *database* sementara yang dikirimkan ke *software* Pentaho. Sebelum data divisualisasikan, data ditransformasikan terlebih dahulu agar data dapat diproses dan diintegrasikan. Tahap awal yang dilakukan adalah membangun blok-blok yang ditunjukkan pada Gambar 3.16. Pada Gambar 3.16 menunjukkan tahap *transformation* terhadap data Sensor *Moisture Soil*. Setiap blok-blok memiliki fungsi dan seleksi data serta pengelompokan data. Blok *table input* berisi data *raw* yang berasal dari *database* sementara. Tahap pertama yang dilakukan adalah membuat koneksi terlebih dahulu dengan *database* agar data dapat ditarik.

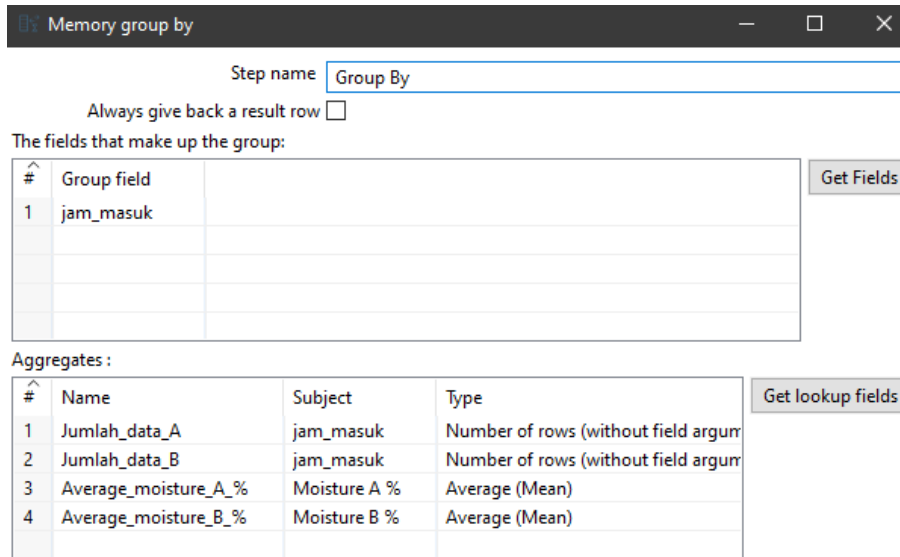


Gambar 3.16 Tampilan Blok *Transformation* Terhadap Sensor *Moisture Soil*.



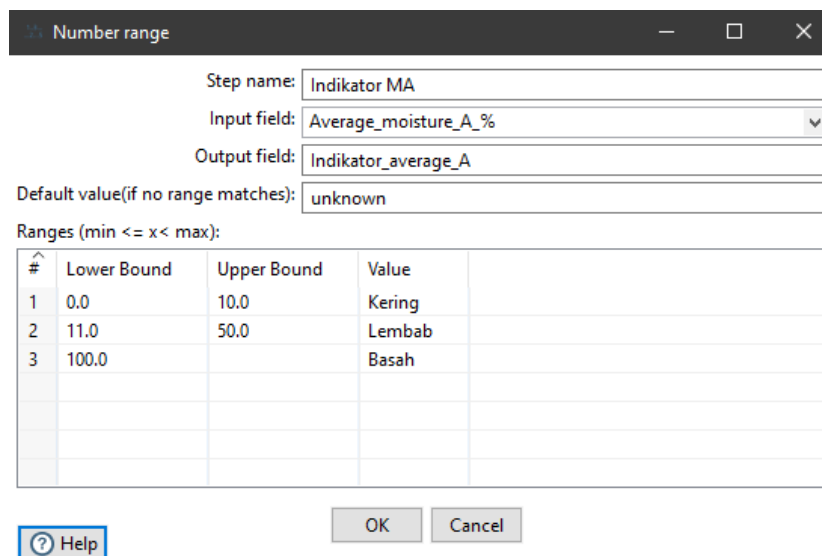
Gambar 3.17 Tampilan Perintah Tahap *Strings Cut* Data Sensor *Moisture Soil*.

Gambar 3.17 menunjukkan proses pada *strings cut* pada data tabel Sensor *Moisture Soil*. *Strings cut* berfungsi untuk memotong data pada suatu kolom tabel data atau *substring* data sehingga didapatkan nilai keluaran baru. Pemotongan berdasarkan urutan angka pada karakter dimulai dari angka 0.



Gambar 3.18 Proses *Group by* Data Sensor *Moisture Soil*.

Gambar 3.18 menunjukkan proses *group by*. Proses *group by* bertujuan untuk mengelompokkan data. Pada tahap ini, proses transformasi yang dilakukan berfungsi untuk menghitung jumlah data yang dihasilkan dari memfilter tabel dengan *type field* yang berupa jumlah baris.



Gambar 3.19 Tahap *Number Range* pada Data Sensor *Moisture Soil*

Tahap *number range* pada Gambar 3.19 bertujuan untuk menentukan nilai indikator tiap titik pada Sensor *Moisture Soil*. Sensor *Moisture Soil* membaca kelembaban tanah dengan nilai antara 0-100 dalam persen. Saat bertambah tinggi suatu nilai keluaran pada Sensor Kelembaban maka keadaan tanah akan semakin basah dan bilaa suatu nilai pada Sensor Kelembaban Tanah menuju ke rendah maka keadaan tanah akan bertambah kering. Sehingga transformasi yang dibutuhkan berupa penambahan nilai indikator pada tiap nilai Sensor *Moisture Soil*.

Insert / update

Step name: Insert / update

Connection: koneksi

Target schema: id18322286\_dbs

Target table: data\_sum\_moisture\_2

Commit size: 100

Don't perform any updates:

The key(s) to look up the value(s):

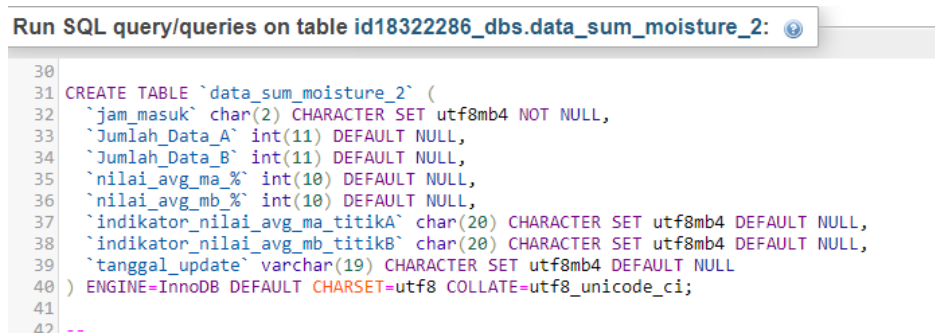
#	Table field	Comparator	Stream field1	Stream field2
1	jam_masuk	=	jam_masuk	
2				

Update fields:

#	Table field	Stream field	Update
1	jam_masuk	jam_masuk	Y
2	Jumlah_data_A	Jumlah_data_A	Y
3	Jumlah_data_B	Jumlah_data_B	Y
4	Average_moisture_A_%	Average_moisture_A_%	Y
5	Average_moisture_B_%	Average_moisture_B_%	Y
6	Indikator_average_A	Indikator_average_A	Y
7	Indikator_average_B	Indikator_average_B	Y
8	tanggal_update	tanggal_update	Y

Gambar 3.20 Proses *Insert/Update* pada *Output Data* Sensor *Moisture Soil*.

Pada tahap terakhir, data final akan dikirimkan ke dalam *database*. Gambar 3.20 menunjukkan tampilan tabel *fields data* yang siap dikirimkan ke dalam *database*. Sebelum data final masuk ke dalam *database*, *target schema* ditentukan berdasarkan nama *database* yang dibuat pada *hosting*. Tahap final dalam transformasi ini yaitu proses *insert/update* pada *output data Sensor Moisture Soil* ke dalam *database*. Target tabel yang dibuat berfungsi untuk menarik data dari Pentaho ke *database*. Sehingga data akan secara otomatis terkirim ke *database* setiap menjalankan Pentaho. Agar data dapat ditarik ke dalam *database*, maka terlebih dahulu harus membuat tabel pada *database* dengan *script* yang ditunjukkan pada Gambar 3.21.



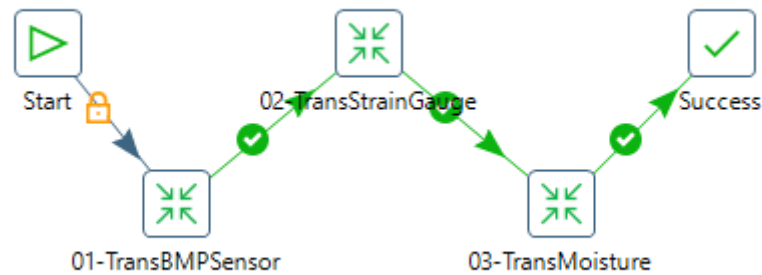
```
Run SQL query/queries on table id18322286_dbs.data_sum_moisture_2:
30
31 CREATE TABLE `data_sum_moisture_2` (
32   `jam_masuk` char(2) CHARACTER SET utf8mb4 NOT NULL,
33   `Jumlah_Data_A` int(11) DEFAULT NULL,
34   `Jumlah_Data_B` int(11) DEFAULT NULL,
35   `nilai_avg_ma_%` int(10) DEFAULT NULL,
36   `nilai_avg_mb_%` int(10) DEFAULT NULL,
37   `indikator_nilai_avg_ma_titikA` char(20) CHARACTER SET utf8mb4 DEFAULT NULL,
38   `indikator_nilai_avg_mb_titikB` char(20) CHARACTER SET utf8mb4 DEFAULT NULL,
39   `tanggal_update` varchar(19) CHARACTER SET utf8mb4 DEFAULT NULL
40 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
41
42 --
```

Gambar 3.21 *Script* Tabel untuk Data Sensor *Moisture Soil* ke dalam *Database*.

#### d. Tahap Job Data

Tugas transformasi dari Sensor BMP, Sensor *Strain Gauge*, hingga tugas Sensor *Moisture Soil* berhasil ditransformasikan dengan mengubah data ke *task* lain. Salah satu karakter hal terpenting tentang ETL adalah prosesnya harus diotomatisasi sebanyak mungkin. *Job* data digunakan untuk mengkoordinasikan ETL. Fungsi *job* data untuk mengatur urutan transformasi. Pada Gambar 3.22 *job* dilakukan dan

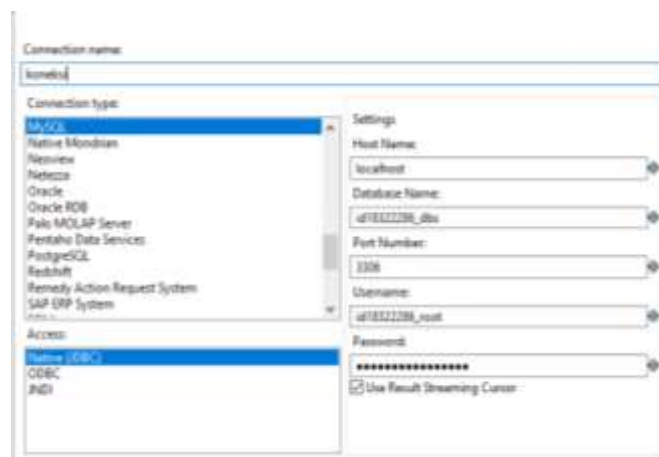
digunakan untuk mengatur aliran transformasi. *Job* yang berhasil ditandai dengan blok hijau *job entry* pada transformasi BMP, *Moisuture Soil*, dan *Strain Gauge* bernilai *true*, maka eksekusi *job* data sukses dan selesai.



Gambar 3.22 Aliran *Job* Data yang Sukses pada Transformasi.

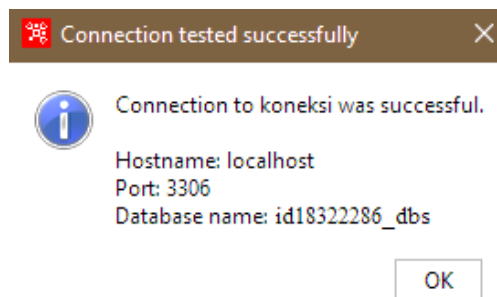
#### e. Tahap Koneksi Pentaho ke *Database*

Agar data yang telah di transformasi dari Pentaho dapat terkirim ke *database*, maka perlu membangun koneksi antara Pentaho dengan *database*. Tahap membangun koneksi ditunjukkan oleh Gambar 3.23.



Gambar 3.23 Tahap Membangun Koneksi ke *Database*.

*Connection type* yang digunakan adalah koneksi MySQL dengan akses JDBC. *Java Database Connectivity* atau disebut juga dengan singkatan JDBC berupa *Application Programming Interface* atau biasa disebut dengan API yang dibuat untuk mengakses suatu *database* berdasar SQL (*Structured Query Language*). Kemudian terdapat *host name* berupa *localhost*, *database name* berupa *id18322286\_dbs*, *port number* berupa 3306, *username* *id18322286\_root*, dan *password* yang harus diisi berdasarkan *database* yang telah dibuat pada *server* yang telah *terhosting*. Dengan adanya koneksi yang menjembatani antara Pentaho terhadap *database*, maka data secara otomatis akan terkirim ke basis data. Gambar 3.24 menunjukkan bahwa Pentaho berhasil terkoneksi ke *database*.



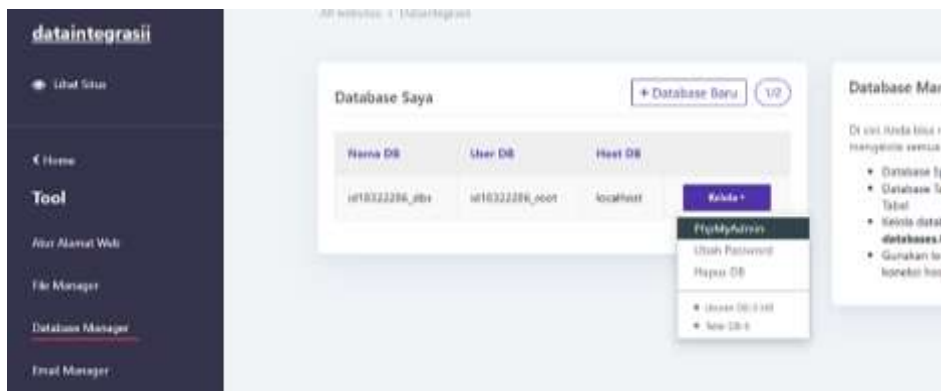
Gambar 3.24 Koneksi ke *Database* Sukses.

### 3.3.5 Perancangan *Website*

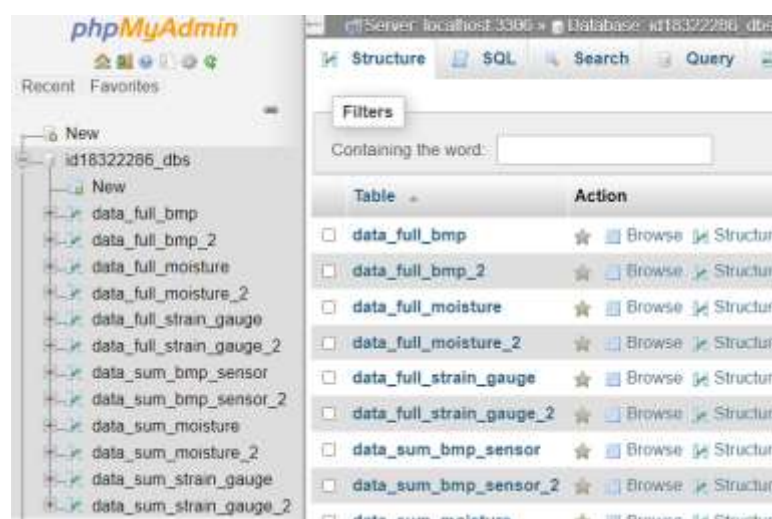
Pada tahap ini, dilakukan pembuatan *website* yang dapat diakses oleh pengguna dan mampu *memonitoring* dan menampilkan data perangkat. Selanjutnya data sensor ditampilkan dalam *website* supaya *user* bisa melihat data-data sensor dan nilai sensor yang dihasilkan. Tahap pembuatan *website* sendiri menggunakan *web hosting* gratis 000.webhost.com yang meliputi beberapa tahapan diantaranya yaitu pembuatan *database*, pembuatan *script* untuk *layout* keluaran data sensor-sensor, penguploadan *layout website* ke dalam *hosting* serta hasil *monitoring* di *website*.

### 3.3.5.1 Perancangan *Database*

Pada tahap perancangan *database*, *database* berguna dalam hal *management database*, sebaliknya dalam hal pengolahannya memakai aplikasi-aplikasi berbasis *website* yang dijalankan melalui *browser internet* dengan phpMyAdmin pada *hosting* 000webhost.com. Sebelum memulai menjalankan PhpMyAdmin, terlebih dahulu menjalankan *webserver apache* di dalam *hosting* dari 000webhost. 000webhosting sendiri memiliki manajemen *webhosting* dengan *Cpanel*. *Cpanel* adalah *webhosting control panel* berbasis GUI yang dirancang untuk memudahkan pengaturan *website* seperti ditunjukkan oleh Gambar 3.25.



Gambar 3.25 *Control Panel Database*.



Gambar 3.26 *Tampilan Database PHPMYAdmin*.



Gambar 3.26 menampilkan daftar tabel-tabel yang berasal dari Pentaho yang telah terintegrasi dan diolah. Daftar tabel terdiri dari data yang merupakan data *full sensor* di mana data tersebut merupakan data *raw* atau seluruh data awal yang belum diubah serta data yang sudah diolah di Pentaho. Tujuan pembuatan tabel *full* dan *sum* agar *user* dengan mudah dapat melihat perbandingan data yang sudah diintegrasikan dengan data awal.

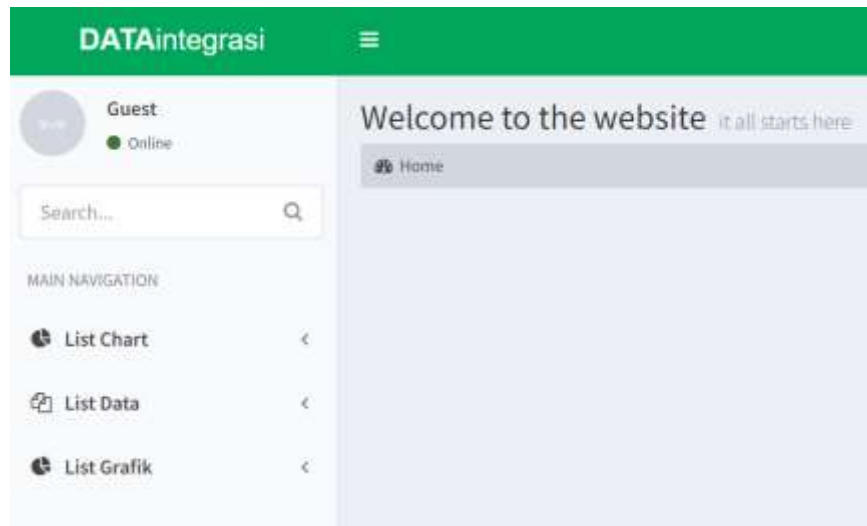
### **3.3.5.2 Upload File Hosting**

*Layout script web* yang telah selesai dibuat perlu diunggah ke *hosting* agar dapat diakses secara *online* oleh *user*. *Hosting server* yang dipakai adalah 000webhost. Sebelum mengunggah ke *hosting* dibutuhkan alamat *web*, *upload script* ke dalam *file manager* di kontrol panel pada 000webhost. Jenis *file* yang diunggah ialah *file HTML* serta *layout web AdminLTE CSS* yang merupakan *script* dalam pembuatan *web* yang akan ditampilkan pada *hosting* dan *file PHP* koneksi *web* dengan *database*. *Layout out* pada tampilan *website* berasal dari suatu *coding* yang sudah dikonfigurasi, sehingga data yang akan ditampilkan akan ditarik dari *database* ke halaman *web* dengan perantara konfigurasi yang sudah dilakukan di halaman *file manager*.

### **3.3.5.3 Perancangan Halaman Utama Website**

Pembuatan *layout web* pada penelitian ini menggunakan bahasa pemrograman PHP, HTML serta menggunakan CSS AdminLTE. HTML digunakan untuk membuat perintah dasar semacam pembuatan tabel, *header*, *title*, serta *text*. Pembuatan *style* tampilan untuk *web* diatur pada AdminLTE. Modul *web* didapat

dari *database* yang berisi informasi data-data sensor *monitoring* yang terkoneksi lewat bahasa pemrograman PHP.



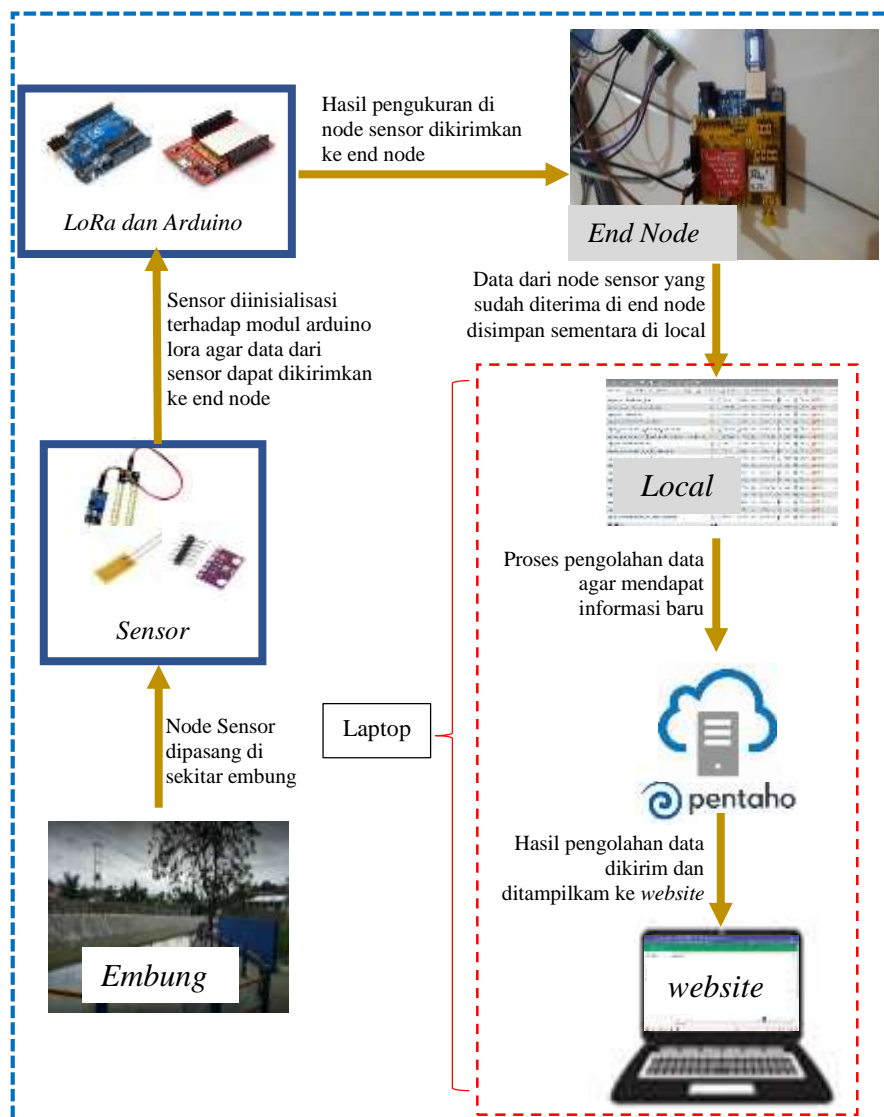
Gambar 3.27 Tampilan Halaman Utama *Layout Website*.

Gambar 3.27 menunjukkan *layout web* yang menampilkan visualisasi data yang merupakan tahap akhir tujuan dari penelitian. Halaman utama adalah halaman yang muncul saat *user* membuka *web* dan berisi menu-menu fungsi agar dapat diakses.

### 3.4 Perancangan Skenario Penelitian

Perancangan dilakukan setelah melakukan studi literatur. Proses awal dimulai dari pemograman sensor terhadap mikrokontroller yaitu Arduino yang telah tertanam dengan LoRa. Kemudian setelah berhasil dilakukan pembacaan data sensor dengan masukan berupa *temperature*, *pressure*, pergeseran tanah dan humiditi atau kelembaban tanah. Arduino beserta LoRa akan memproses inputan dari sensor. Kemudian data dari sensor akan dikirimkan ke *end node* dan disimpan sementara ke *local database*. Selanjutnya data akan diintegrasikan dan diolah sesuai dengan

*request* data yang dibuat. Data kemudian akan dikelompokkan sesuai dengan format yang diinginkan menggunakan Pentaho. Setelah data diolah, data dikirimkan ke *database* dengan *hosting* 000webhost. Data informasi yang sudah tersimpan ke dalam *database* akan diprogram kembali supaya informasi data dapat divisualisasikan secara grafik dan *list* data dengan bantuan PHP dan AdminLte pada halaman *website*. Visualisasi ini bertujuan agar *user* dapat dengan mudah melihat informasi yang *dimonitoring*. Pembuatan halaman *website* untuk visualisasi data *monitoring* dilakukan dengan mengimplementasikan PHP dan html.



Gambar 3.28 Arsitektur Sistem Keseluruhan.

### 3.4.1 Skenario Penelitian Sensor *Moisture*

Pada skenario pertama, Sensor *Moisture* diletakkan diberbagai titik yang berbeda pada tanah. Jarak tanah pada *end node* berkisar 8-10 meter. Data diambil dengan menggunakan Sensor *Moisture* ditambah dengan modul LoRa *Shield* dan Arduino. Pengambilan data berkisar 8 jam dengan tiap data masuk diambil selama 60 detik atau 1 menit. Tujuan pengambilan data dengan waktu 8 jam dan diletakkan dengan titik yang berbeda agar dapat diketahui kondisi tanah serta efisiensi LoRa terhadap data. Maka dapat diketahui apakah pengiriman data terjadi *delay* atau tidak. Kondisi pengiriman dianggap atau diasumsikan dalam keadaan tidak LOS sehingga data terkirim dengan adanya hambatan atau *obstacle*. Untuk perhitungan data yang akan masuk ke *end node* berjumlah 480 data selama per 8 jam, sehingga dapat diketahui pada titik mana data mengalami *delay* dan juga dapat diketahui kondisi tanah di sekitar titik yang ditanam Sensor *Moisture*. Contoh tabel skenario ditunjukkan pada tabel 3.2 sebagai berikut.

Tabel 3.2 Skenario *Output* pada Sensor *Moisture*

No.	Waktu	Persentase Kelembaban (%)		Kondisi Tanah
		Titik A	Titik B	
1	Hh/mm/ss	%	%	Basah/Lembab/Kering
2	Hh/mm/ss	%	%	Basah/Lembab/Kering
3	Hh/mm/ss	%	%	Basah/Lembab/Kering
4	Hh/mm/ss	%	%	Basah/Lembab/Kering
..480	Hh/mm/ss	%	%	Basah/Lembab/Kering

### 3.4.2 Skenario Sensor *Strain Gauge*

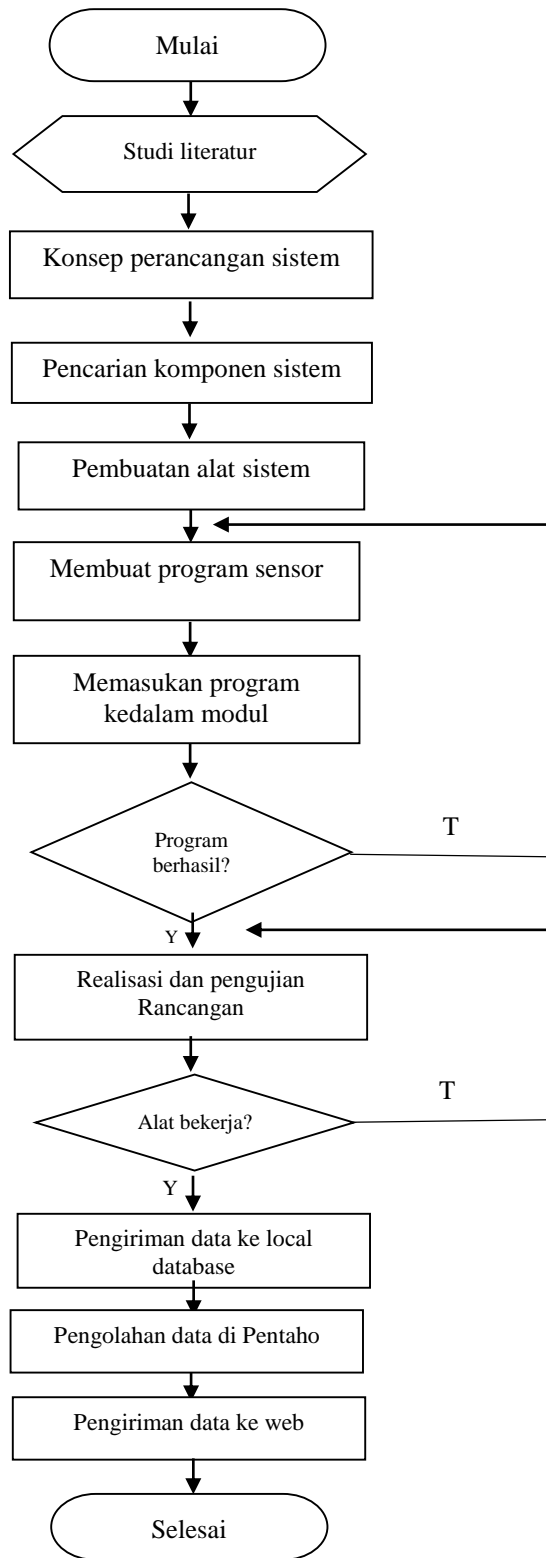
Pada skenario kedua untuk Sensor *Strain Gauge*, diletakkan di simulasi pentanahan pada Laboratorium Sipil Universitas Lampung. Pada simulasi tersebut, menggunakan bambu kecil pipih potong yang diasumsikan sebagai media tanah yang dianggap memiliki elastisitas menyerupai tanah. Untuk Sensor *Strain Gauge* sebagai *node* telah terhubung dengan modul *LoRa Shield* dan *Arduino Uno*. *Strain Gauge* ditempelkan pada *beam* tembaga untuk mengukur pergeseran yang terjadi jika diberi beban atau tekanan tertentu sehingga dapat dilihat *output* yang diinginkan, yaitu nilai Amplitudo/Tegangan, dan juga arah pergeseran apakah mengalami deformasi kestrain. Perhitungan nilai *strain* Contoh tabel skenario ditunjukkan pada tabel 3.3 sebagai berikut.

Tabel 3.3 Skenario *Output* pada Sensor *Strain Gauge*

Tanggal/waktu	Beban (gr)	Tegangan (V)	<i>Strain</i> (N/m <sup>2</sup> )
Hh/mm/ss	Beban Awal... gr	... v	... N/m <sup>2</sup>
Hh/mm/ss	Beban 2... gr	... v	... N/m <sup>2</sup>
Hh/mm/ss	Beban 3...gr	... v	... N/m <sup>2</sup>

### 3.5 Diagram Alir

Proses penelitian ini menggunakan diagram alir yaitu diagram alir proses penelitian. Proses awal penelitian dengan melakukan studi literatur, kemudian melakukan konsep perancangan sistem. Setelah konsep perancangan, dimulailah pembuatan alat serta pencarian komponen. Setelah alat selesai, dilakukan pemograman sensor terhadap modul lora dan arduino dan jika terdapat keluaran nilai, data tersebut akan diolah dan ditampilkan ke *website*. Untuk lebih jelasnya dapat dilihat Gambar 3.28.



Gambar 3.29 Diagram Alir Penelitian

## V. SIMPULAN DAN SARAN

### 5.1 Simpulan

Kesimpulan yang didapatkan dari hasil penelitian ini adalah sebagai berikut:

1. Data yang ditampilkan pada halaman utama *website* dapat menampilkan Grafik Sensor BMP yang membahas mengenai nilai temperatur dan tekanan udara berdasarkan waktu pembacaan, Grafik Sensor *Moisture Soil* yang membahas persentase kelembaban kondisi tanah berdasarkan waktu pembacaan, serta Grafik Sensor *Strain Gauge* yang membahas pengaruh beban terhadap tegangan yang dihasilkan.
2. Data dari titik A dan titik B Sensor *Moisture Soil* dapat mengirimkan data dengan jarak antar node +- 10 meter tanpa delay saat data terkirim ke *end node*.
3. Pada saat pengambilan data Sensor *Moisture Soil* pukul 12-13 siang terjadi gejala anomali akibat hujan selama 15 menit sehingga Sensor *Moisture Soil* menghasilkan data yang mengalami penurunan nilai secara drastis hingga - 89%.

## 5.2 Saran

Saran untuk penelitian selanjutnya adalah sebagai berikut:

1. Saat menginstalasi program dan mengukur nilai sensor, lebih baik dikalibrasi dahulu dengan teliti agar didapatkan nilai yang presisi dan akurat karena batasan masalah peneliti tidak terlalu membahas tentang perangkat sensor.
2. Untuk penelitian selanjutnya, diharapkan pengukuran dilakukan secara langsung disatu tempat, sehingga didapatkan nilai yang lebih akurat sesuai kondisi tempat pengukuran.