

**Pengembangan Database Gempabumi Stasiun Geofisika
Kelas I Padang Panjang Berbasis Web Menggunakan
Python MySQL**

*Web Based Earthquake Database Development At Stasiun
Geofisika Kelas I Padang Panjang Using Python MySQL*



**STASIUN GEOFISIKA KELAS I
PADANG PANJANG**

Disusun Oleh :

Shandy Yogaswara Surya Putra

NIP. 199412122013121001

Badan Meteorologi Klimatologi dan Geofisika

Stasiun Geofisika Kelas I Padang Panjang

Jl. Meteorologi, Silaing Bawah, Padang Panjang, Sumatera Barat - 27118

Telp/Fax. (0752)484236, 82236

Pernyataan Orisinalitas

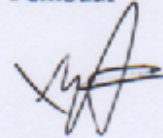
Saya, Shandy Yogaswara Surya Putra, NIP. 199412122013121001, menyatakan bahwa tulisan berjudul **Pengembangan Database Gempabumi Stasiun Geofisika Kelas I Padang Panjang Berbasis Web Menggunakan Python Mysql** beserta hasilnya merupakan karya asli dan belum pernah dilakukan di Stasiun Geofisika Kelas I Padang Panjang sebelumnya. Seluruh ide yang ada dalam tulisan ini kecuali yang saya gunakan sebagai kutipan, merupakan pendapat yang saya susun sendiri. Jika pernyataan di atas terbukti sebaliknya, maka saya bersedia menerima sanksi yang ditetapkan oleh pihak terkait.

Mengetahui,
Kepala Stasiun Geofisika Kelas I
Padang Panjang

Rivan Sianel, S.P., M.Si
NIP. 197509061998031001

Padang Panjang, 2 November 2021

Pembuat



Shandy Yogaswara Surya Putra
NIP. 199412122013121001

DAFTAR ISI

Pernyataan Orisinalitas	i
Daftar Isi	ii
BAB I Pendahuluan	1
BAB II Dasar Teori	3
BAB III Metode Kerja Sistem	5
BAB IV Implementasi	9
BAB V Kesimpulan	15
Daftar Pustaka	16
Lampiran	17

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sampai dengan pertengahan 2018, data geofisika di Stasiun Geofisika Kelas I Padang Panjang di arsipkan secara elektronik menggunakan office dan kemudian dipindahkan ke penyimpanan eksternal ketika sudah mulai banyak. Hal ini menimbulkan masalah ketika perlu mengakses data-data tersebut ketika penyimpanan eksternal tidak diketahui keberadaannya. Bahkan jika data telah diperoleh, petugas harus mengolah data tersebut sehingga dapat ditampilkan secara layak, seperti menjadi sebuah peta seismisitas. Hal ini membuat proses dari pengambilan data hingga data menjadi siap pakai memakan waktu yang tidak sebentar.

Atas dasar pemikiran di atas, maka tercetuslah ide untuk membuat sebuah sistem database mutakhir yang selain dapat menyimpan data, juga dapat memproses data mentah yang tersimpan menjadi data akhir yang siap digunakan sehingga dapat mengakomodir petugas yang tidak lancar dalam mengoperasikan software pemetaan terkini seperti Arcgis. Dengan demikian, terciptalah versi awal dari sistem database di Stasiun Geofisika Kelas I Padang Panjang.

Pada versi awal, sistem database menggunakan perpaduan antara Mysql sebagai pengolahan database, Python versi 2.7 sebagai bahasa programnya, dan GMT versi 5 sebagai aplikasi pemetaannya. Pada versi ini, petugas masih harus menginput kejadian gempabumi secara manual seperti saat mengetik di excel, dan hanya menghasilkan peta seismisitas saja.

Sampai dengan September 2021, tercatat sistem database ini telah mengalami perubahan sebanyak 11 kali, mulai dari penambahan grafik batang dan lingkaran analisa data sebagai keluaran, sampai dengan menghapus penggunaan GMT dan hanya bergantung pada modul Basemap dan Cartopy milik Python untuk pemetaan, dan terakhir mengubah sistem yang awalnya merupakan sebuah sistem *standalone* dan hanya dapat diakses di satu PC, menjadi sistem database berbasis

interaksi *server-client* yang dapat diakses oleh seluruh perangkat PC maupun mobile yang terhubung ke jaringan lokal.

1.2 Tujuan

Tujuan dari pengembangan ini adalah untuk mendesain sistem database gempabumi di Stasiun Geofisika Kelas I Padang Panjang yang berbasis pemrograman web sehingga dapat diakses seluruh pihak yang berkepentingan.

BAB II

DASAR TEORI

2.1 Database

Sistem pengelolaan database adalah sebuah sistem yang bertugas untuk mengatur pengelolaan data seperti memasukkan data, mengubah, dan membantu dalam pencarian data oleh pengguna (Andrasto, T. 2013).

2.2 MySQL

Dalam pengolahan database berbasis elektronik, salah satu yang sering digunakan adalah bahasa pemrograman MySQL. MySQL adalah sebuah program database server yang mampu menerima dan mengirimkan data dengan cepat oleh banyak pengguna menggunakan perintah dasar SQL atau *Structur Query Language*. MySQL sendiri memiliki lisensi yang berbayar dan gratis sehingga dapat digunakan untuk kepentingan personal tanpa harus membayar lisensi tersebut. Selain itu, MySQL juga dapat menyimpan data dengan kapasitas yang besar serta didukung password yang terenkripsi sehingga sistem database berbasis MySQL cukup aman dibanding dengan sistem penyimpanan data menggunakan metode lama seperti menggunakan dokumen (Saputro, H. 2012).

Terdapat tiga bentuk perintah dasar dalam SQL yang digunakan untuk mengelola database, yakni *Data Definition Language*, *Data Manipulation Language*, dan *Data Control Language*. *Data Definition Language* (DDL) adalah metode SQL yang digunakan untuk mendefinisikan sebuah database. Terdapat tiga perintah dalam DDL yakni CREATE untuk membuat database, DROP untuk menghapus database, dan ALTER untuk mengubah struktur database (Saputro, H. 2012).

Data Manipulation Language (DML) adalah metode SQL untuk memanipulasi database. Metode ini hanya dapat digunakan setelah DDL terjadi. Terdapat tiga perintah dalam DML yakni INSERT untuk memasukkan data ke dalam database, UPDATE untuk melakukan perubahan data pada database, dan DELETE untuk menghapus data dalam database (Saputro, H. 2012).

Data Control Language (DCL) adalah metode SQL yang digunakan untuk memberikan hak akses database. Terdapat empat perintah dalam DCL yakni GRANT untuk memberi izin pengguna mengakses database, REVOKE GRANT untuk membatalkan izin pengguna, COMMIT untuk menetapkan penyimpanan database, dan ROLLBACK untuk membatalkan penyimpanan database (Saputro, H. 2012).

Pengguna dapat menginstal MySQL pada perangkat yang digunakan dengan mengunduh installer dari situs MySQL. Cara instalasinya sendiri mirip dengan menginstal perangkat lunak kebanyakan pada masing-masing sistem operasi.

2.3 Python

Python merupakan bahasa pemrograman dengan orientasi objek (*Object Oriented Programming*) yang didesain sedemikian rupa sehingga mudah dibaca oleh pengguna (Kadarina dkk, 2019). Berdasar Tiobe Index per September 2021, bahasa pemrograman Python memiliki tingkat pencarian sebesar 11.67%, berada di posisi ke dua setelah C++ dengan tingkat pencarian sebesar 11.83% dari 100 bahasa pemrograman. Hal ini menunjukkan bahwa bahasa pemrograman python memiliki komunitas yang besar sehingga tidak akan sulit untuk mencari ilmu terkait.

Dalam pengolahan database, terdapat beberapa modul python yang dapat digunakan seperti SQLAlchemy yang berfungsi untuk menghubungkan Python dengan SQL, Matplotlib untuk membuat plot data, Numpy untuk pemrograman berbasis matematis, dan Pandas untuk mengolah dataframe.

Cara terbaik untuk menginstall Python adalah menggunakan perangkat lunak Anaconda atau Miniconda. Perangkat lunak ini akan mempermudah ketika harus memindahkan atau menduplikasi project terkait pemrograman python ke perangkat lainnya.

2.4 Web Programming

Pemrograman web atau *web Programming* merupakan istilah yang erat kaitannya dengan pengembangan sebuah website. Dalam mendesain sebuah website menggunakan bahasa pemrograman python, digunakan modul Flask.

Flask berfungsi sebagai kerangka kerja aplikasi dan tampilan dari sebuah web. Flask termasuk dalam jenis *microframework* karena tidak memerlukan alat atau modul tertentu dalam penggunaannya (Irsyad, R. 2018).

Flask bergantung pada *jinja engine template* dan WERKZEUG WSGI *toolkit*¹. Hal ini membuat penggunaan flask menjadi lebih mudah dibandingkan dengan framework lainnya serta sudah mencakup fungsi-fungsi dasar dalam pengembangan website seperti keperluan autentifikasi dan akses database.

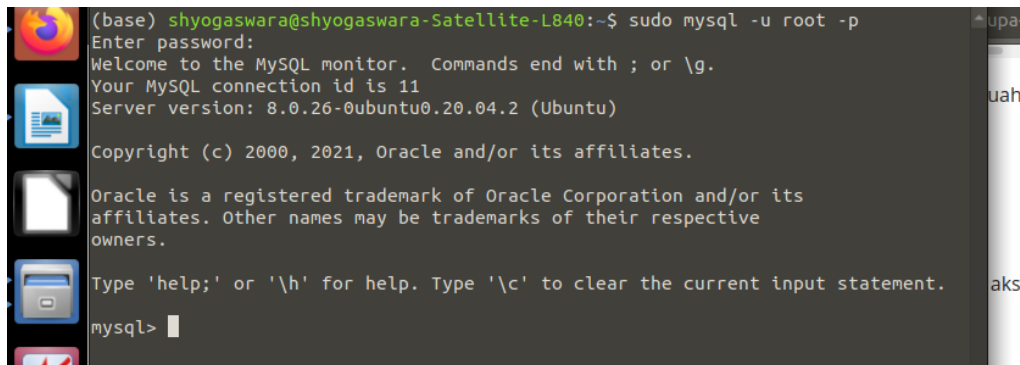
1 The Pallets Project, Flask, flask.palletsprojects.com/en/2.0.x/

BAB III

METODE KERJA SISTEM

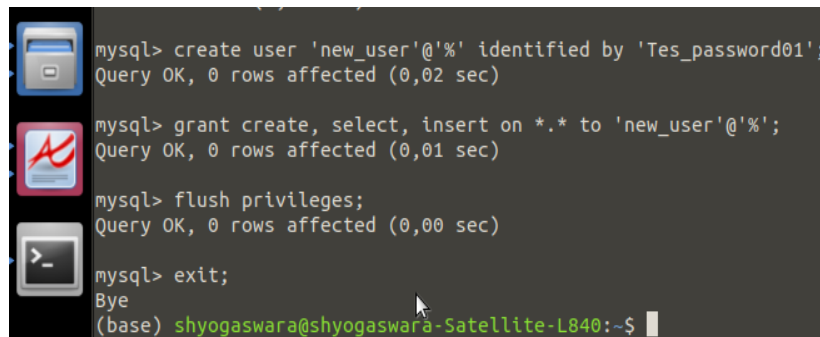
3.1 Persiapan MySQL

Setelah melakukan instalasi MySQL pada server, dibuatlah user baru yang diberikan akses pada database untuk digunakan sebagai client. Untuk melakukan hal ini, pengguna harus masuk ke MySQL sebagai root menggunakan password yang dibuat saat melakukan instalasi².



Gambar 3.1.1 Proses masuk ke MySQL sebagai root, sistem operasi yang digunakan adalah Ubuntu

Setelah masuk sebagai root, pengguna dapat membuat user baru yang diidentifikasi menggunakan password tertentu dan kemudian memberikan hak akses pada user tersebut. Adapun hak akses yang diberikan hanyalah akses untuk membuat database, mengambil, dan memodifikasi entri data sehingga user tersebut tidak dapat menghapus data pada database.



Gambar 3.1.2 membuat user baru pada MySQL sebagai client

² Oracle Corp, MySQL Documentation, dev.mysql.com/doc/refman/8.0/en/

Pada gambar 3.1.2, `new_user` merupakan nama user yang akan digunakan client untuk mengakses database. Tanda `%` pada baris `'new_user'@'%'` berfungsi agar `new_user` ini dapat digunakan oleh client dengan sembarang IP. Jika user ini hanya boleh digunakan pada IP tertentu, maka tanda `%` tadi diganti dengan IP perangkat tersebut. Untuk menggunakan `new_user`, client memasukkan password yang dibuat yakni `Tes_password01`.

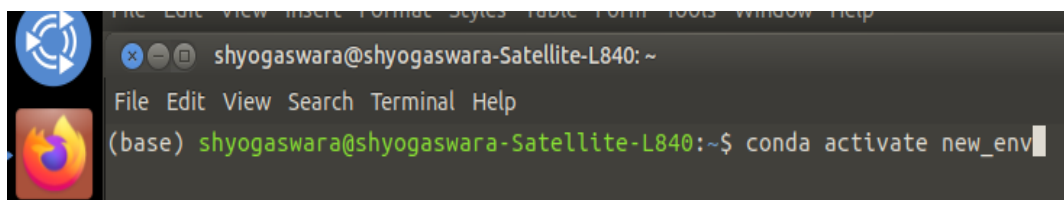
3.2 Persiapan Python

Setelah menginstall Anaconda maupun Miniconda, pengguna harus membuat *virtual environment* yang berisikan modul-modul python yang akan digunakan. Penggunaan *virtual environment* bertujuan agar ketika pengguna membuat proyek lainnya dan membutuhkan modul maupun versi python yang berbeda, pengguna tidak harus menghapus dan menginstall kembali python tersebut.



Gambar 3.2.1 membuat *virtual environment* baru pada Anaconda

Pada gambar 3.2.1 menunjukkan cara membuat *virtual environment* pada Anaconda³. *Virtual environment* ini bernama `env_name` dan menggunakan python versi 3.6. Setelah *virtual environment* selesai dibuat, pengguna dapat mengaktifkannya seperti pada gambar 3.2.2. Setelah diaktifkan, pengguna dapat menginstall modul-modul yang digunakan yakni Pandas, Flask, SQLAlchemy, dan Cartopy.



Gambar 3.2.2 cara mengaktifkan *virtual environment* pada Anaconda

³ Anaconda documentation, docs.anaconda.com/anaconda

Selain cara di atas, terdapat cara yang lebih mudah untuk membuat *virtual environment* pada Anaconda, yakni menggunakan file yml yang berisi konfigurasi *virtual environment* yang telah dibuat pada perangkat lain. Hal ini lebih mudah karena Anaconda sendirilah yang akan langsung mencari modul-modul yang diperlukan. Kekurangan metode ini adalah harus ada perangkat yang memiliki *virtual environment* yang telah komplit untuk dijadikan file yml.



```

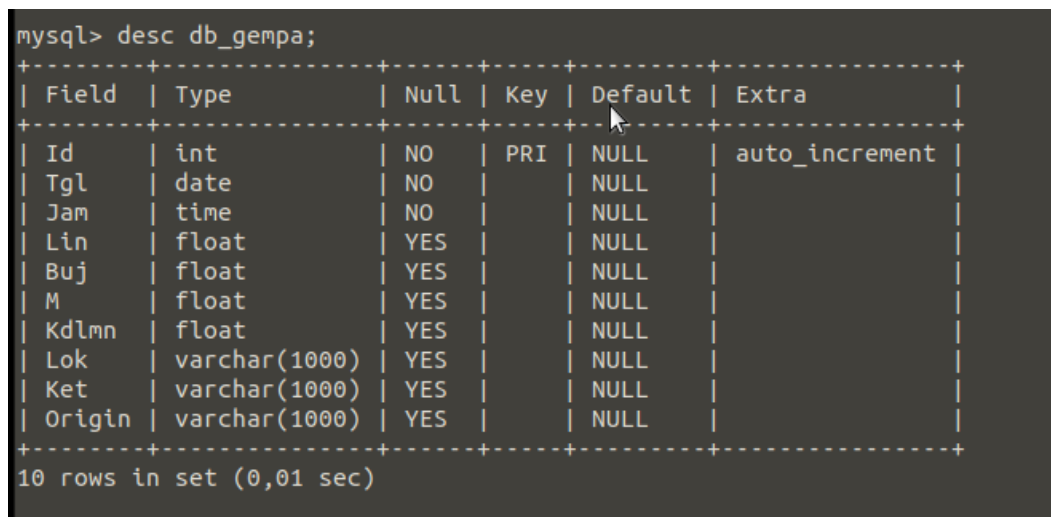
Setting environment variables: 1. Create the environment from the 'environment.yml' file*
shyogaswara@shyogaswara-Satellite-L840: ~
File Edit View Search Terminal Help
(base) shyogaswara@shyogaswara-Satellite-L840:~$ conda env create -f nama_file.yml

```

Gambar 3.2.3 membuat *virtual environment* Anaconda dari file bernama nama_file.yml

3.3 Rancangan skema database

Skema tabel database gempabumi memuat informasi parameter dari event gempabumi tersebut. Secara umum tabel memuat data berupa waktu, lokasi, dan kekuatan. Gambar 3.3.1 menunjukkan skema tabel database gempabumi dalam MySQL⁴.



```

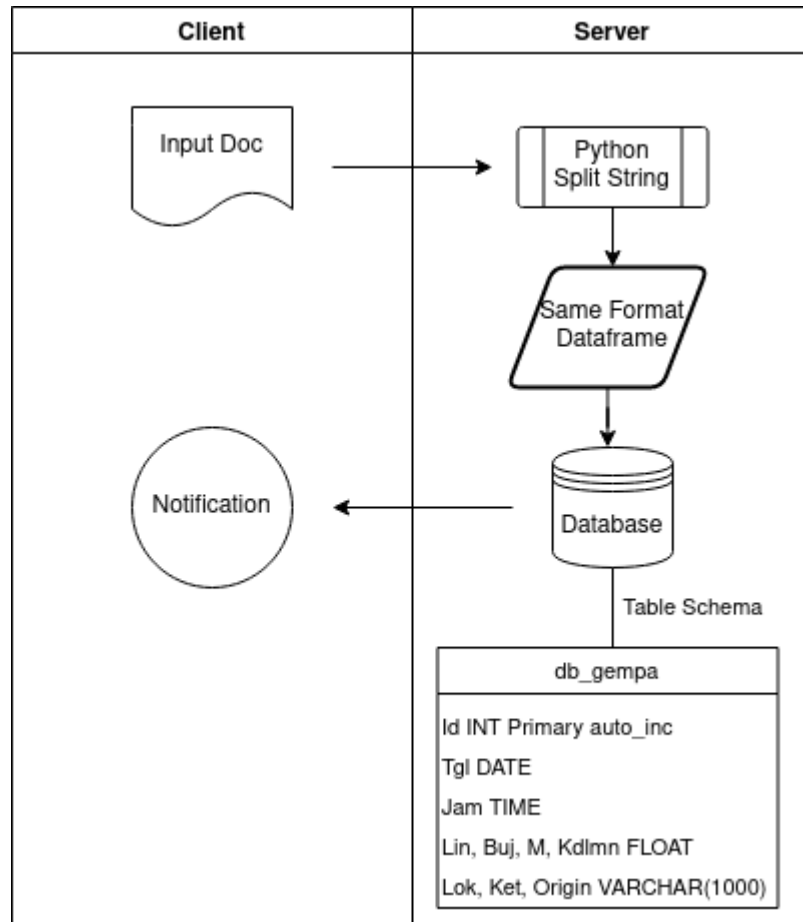
mysql> desc db_gempa;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| Id     | int           | NO   | PRI | NULL    | auto_increment |
| Tgl    | date          | NO   |     | NULL    |                |
| Jam    | time          | NO   |     | NULL    |                |
| Lin    | float         | YES  |     | NULL    |                |
| Buj    | float         | YES  |     | NULL    |                |
| M      | float         | YES  |     | NULL    |                |
| Kdlnm  | float         | YES  |     | NULL    |                |
| Lok    | varchar(1000) | YES  |     | NULL    |                |
| Ket    | varchar(1000) | YES  |     | NULL    |                |
| Origin | varchar(1000) | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
10 rows in set (0,01 sec)

```

Gambar 3.3.1 skema tabel database gempabumi

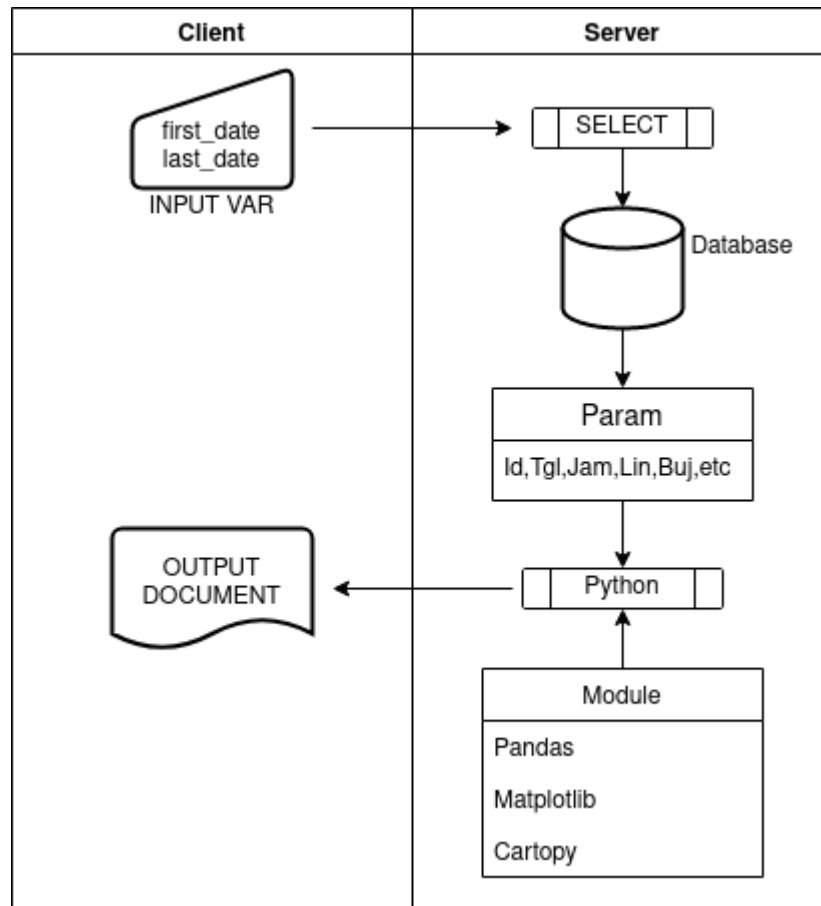
4 Oracle Corp, MySQL Documentation, dev.mysql.com/doc/refman/8.0/en/

Saat akan memasukkan data, pengguna mempersiapkan data berupa pesan pendek informasi gempabumi. Pesan pendek tersebut kemudian akan di olah sehingga memiliki format bacaan yang sama baik pesan pendek biasa maupun pesan pendek dengan data dirasakan.



Gambar 3.3.2 Diagram alir sederhana proses memasukkan data ke database.

Saat akan mengambil data dari tabel database, pengguna akan memasukkan variabel berupa periode data yang terdiri dari tanggal awal data dan tanggal akhir data. Variabel tersebut akan memfilter kolom Tgl pada tabel database yang tidak sesuai dan mengekstrak sisanya. Data yang telah teresktrak tadi kemudian diolah menggunakan modul Cartopy, Pandas, dan Matplotlib yang kemudian disatukan menjadi dokumen siap pakai.



Gambar 3.3.3 Diagram alir sederhana proses mengambil data dari database

BAB IV

IMPLEMENTASI

4.1 Pembuatan Database

Database dibuat menggunakan MySQL dengan Python sebagai backendnya. Setiap kali server di jalankan, sistem akan secara otomatis mengecek keberadaan database menggunakan fungsi `create_dbgempa` dari class `MainFunc` yang terdapat dalam file `classFunc.py` pada folder `bin` pada server. Class `MainFunc` dipanggil saat server dijalankan melalui `main.py`.

```
def create_dbgempa(self, table_name, engine):
    engine = engine
    if not engine.has_table(table_name):
        metadata = MetaData(engine)
        Table(table_name, metadata,
              Column('Id', Integer, primary_key=True, nullable=False),
              Column('Tgl', Date, nullable=False),
              Column('Jam', Time, nullable=False),
              Column('Lin', Float), Column('Buj', Float), Column('M', Float), Column('Kdlnn', Float),
              Column('Lok', String(1000)), Column('Ket', String(1000)), Column('Origin', String(1000))
              )
        metadata.create_all()
```

Gambar 4.1.1 Fungsi pada class `MainFunc` yang dijalankan untuk membuat database oleh server.

4.2 Deskripsi Halaman Web

Terdapat tiga halaman yang telah dipersiapkan pada website yakni Home, Database Gempabumi, dan About Us. Untuk halaman Home dan About Us dapat di isikan dengan informasi-informasi terkait Stasiun Geofisika Kelas I Padang Panjang. Sedangkan pada halaman Database Gempabumi terdapat fungsi-fungsi untuk menginput dan mengambil data.

4.2.1 Halaman Database Gempabumi

Halaman Database Gempabumi terbagi menjadi dua fungsi yakni untuk memasukkan data dan mengambil data. Kedua fungsi tersebut tersimpan kedalam file `auth.py` yang akan diakses ketika masuk ke halaman yang bersangkutan.

Gambar 4.2.1 Tampilan antar muka halaman Database Gempabumi untuk memasukkan data

Gambar 4.2.2 Tampilan antar muka halaman Database Gempabumi untuk mengambil data

4.2.2 Persiapan Untuk Memasukkan Data

Data parameter gempabumi yang dapat dimasukkan ke dalam database adalah informasi berupa pesan pendek yang di diseminasikan ke masyarakat. Data tersebut dapat disimpan ke dalam file teks maupun dimasukkan satu per satu pada halaman Database Gempabumi bagian input data.

Data ini kemudian akan dibaca oleh server menggunakan fungsi `rd1Event` jika data dimasukkan satu persatu, atau `rdMulti` jika data dimasukkan menggunakan file teks. Kedua fungsi ini bertujuan untuk melakukan pemisahan data masukan yang masih berformat *string* menjadi sebuah tabel dataframe yang akan dimasukkan ke dalam database.

```

def rd1Event(self,event_str):
    ev_str = re.sub(r',', '|', event_str.rstrip(), count=4)
    df = pd.DataFrame([x.split('|') for x in ev_str.split('\n')])
    df = self.raw2df(df)
    return df

def rdMulti(self,filename): # read raw data and convert it to usable
    for pd = io.StringIO()
    with open(filename) as f:
        for line in f:
            new_line = re.sub(r',', '|', line.rstrip(), count=4)
            print (new_line, file=for_pd)
    for pd.seek(0)
    df = pd.read_csv(for_pd, sep='|',engine='python', header=None)
    for pd.close()
    df = self.raw2df(df)
    return df

```

Gambar 4.2.3 Fungsi untuk mempersiapkan data mentah sebelum dimasukkan ke dalam database. Kedua fungsi ini terdapat pada class MainFunc

4.2.3 Pengambilan Data

Untuk mengambil data, variabel masukan berupa tanggal awal dan akhir dari periode data yang di inginkan akan diproses oleh server seperti diagram alir pada gambar 3.3.2. Variabel tersebut diperoleh menggunakan request form seperti pada gambar 4.2.2 atas. Setelah variabel diperoleh, server akan menjalankan fungsi sql2df yang terdapat pada class MainFunc untuk mengambil data dari database dan membuatnya berada dalam format dataframe untuk diproses lebih lanjut.

```

def sql2df(self,engine,table_name,start_date,end_date,path):
    query='Select * from {0} where Tgl between %s and %s'.format(table_name)
    df = pd.read_sql(query,con=engine, params=(start_date,end_date))
    df['Jam'] = (pd.Timestamp('now').normalize() + df['Jam']).dt.time
    df['Tgl'] = df['Tgl'].apply(lambda x: x.strftime('%Y-%m-%d'))
    df = df.sort_values(by=['Tgl','Jam'])
    #engine.close()
    path = os.path.join(path, 'output.csv')
    self.df2csv(df,path)
    return df

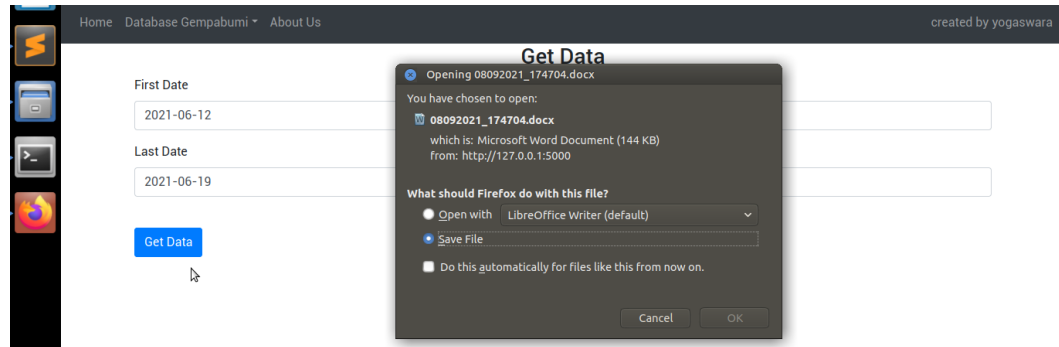
def sql_all2df(self,engine,table_name):
    query='Select * from {0}'.format(table_name)
    df = pd.read_sql(query,con=engine)
    engine.close()
    self.df2csv(df)
    return df

```

Gambar 4.2.4 Fungsi sql2df untuk mempersiapkan data yang diambil dari database.

Dataframe yang telah di peroleh menggunakan fungsi ini kemudian di olah menggunakan tiga fungsi lanjutan dari class MainFunc yakni df2Seismap untuk membuat peta seismisitas, df2MagHist untuk membuat histogram dan diagram lingkaran klasifikasi gempabumi berdasar magnitudo, serta df2DistHist untuk

membuat histogram dan diagram lingkaran klasifikasi gempabumi berdasar jarak terhadap Stasiun Geofisika Kelas I Padang Panjang. Setelah selesai diolah, data kemudian di download oleh client.



Gambar 4.2.6 Tampilan antar muka dokumen yang akan di download oleh client.

BAB V

KESIMPULAN

5.1 Kesimpulan

Penggunaan bahasa pemrograman python dengan mysql sebagai pengembangan sistem penyimpanan data sangat membantu petugas dalam hal pengolahan data yang lebih cepat, serta penyimpanan yang lebih hemat ruang dibanding penggunaan office seperti excel dan word untuk menyimpan setiap data untuk periode tertentu.

5.2 Pengembangan Ke Depan

Terdapat beberapa hal yang dapat dikembangkan ke depannya dari sistem ini, diantaranya :

1. Saat ini sistem hanya dapat diakses oleh perangkat yang terhubung ke jaringan lokal, sehingga tampilan yang dibuat sesederhana mungkin. Ke depannya sistem ini dapat dibuat menjadi publik, dengan menambahkan beberapa pengetatan seperti fungsi autentikasi untuk akses ke database.
2. Penambahan database gempabumi dirasakan yang berisi data narasi serta peta goncangan juga dapat dilakukan yang nantinya dapat dipadu dengan keluaran dari database gempabumi.
3. Pengembangan perangkat keras komputer server, yang mana saat ini menggunakan komputer spesifikasi lama dengan RAM 2 gb, HDD 250 gb, dan OS Lubuntu 20.04 64 bit.

Daftar Pustaka

- Anaconda Inc.**, Anaconda Documentation, “docs.anaconda.com/anaconda”, diakses pada 25 Februari 2021.
- Andrasto, T.**, 2013, Pengembangan Sistem Database Hasil Penelitian Dan Pengabdian Kepada Masyarakat Dosen UNNES, *Jurnal Teknik Elektro*, Vol.5 No.2
- Irsyad, R.**, 2018, Penggunaan Python Framework Flask Untuk Pemula, OSF Preprints, doi:10.31219/osf.io/t7u5r
- Kadarina, M., T.**, dkk, 2019, Pengenalan Bahasa Pemrograman Python Menggunakan Aplikasi Games Untuk Siswa/i Di Wilayah Kembangan Utara, *Jurnal Abdi Masyarakat*, Vo.5 No.1, hal. 11-16
- Oracle Corp.**, MySQL 8.0 Reference Manual, “dev.mysql.com/doc/refman/8.0”, diakses pada 25 Februari 2021.
- The Pallets Project.**, FLASK, “flask.palletsprojects.com/en/2.0.x/”, diakses pada 10 Agustus 2021

LAMPIRAN

```
auth = Blueprint('auth', __name__)

@auth.route('/db-gempa/input-data', methods=['GET','POST'])
def input_data():
    #UPLOAD TO SERVER#
    if request.method == 'POST':
        event = request.form.get('event')
        file = request.files['file']
        if file:
            filename = secure_filename(file.filename)
            file.save(os.path.join(UPLOAD_FOLDER,filename))
            fname = os.path.join(UPLOAD_FOLDER,filename)
            dt = mf.rdMulti(fname)
            #print(dt)
            mf.df2sql(engine,dt,db_table[0])
            flash('Multi Events Data File Uploaded!', category='success')
            return redirect(url_for('views.home'))
        elif len(event) > 70:
            dt = mf.rd1Event(event)
            mf.df2sql(engine,dt,db_table[0])
            #print(dt)
            flash_msg = f'{event} Uploaded'
            flash(flash_msg, category='success')
            return redirect(url_for('views.home'))
        else:
            flash('Please Input File for Multi Event Data or Type Real Event Message for Single Event!', category='error')

    return render_template('input_data.html')
```

Gambar L.1 Fungsi pada auth.py yang berkaitan dengan proses input data melalui tampilan web, terkait dengan gambar 4.2.1.

```
@auth.route('/db-gempa/get-data', methods=['GET','POST'])
def get_data():
    if request.method == 'POST':
        fDate = request.form.get('fDate')
        lDate = request.form.get('lDate')
        df = mf.sql2df(engine,db_table[0],fDate,lDate, UPLOAD_FOLDER)
        #print(df)
        mf.df2Seismap(df, UPLOAD_FOLDER)
        df_mag, max_val, max_valId = mf.df2MagHist(df, UPLOAD_FOLDER)
        df_dist = mf.df2DistHist(df, UPLOAD_FOLDER)
        fname = ['seismap.png','magnitude.png','distance.png','pie_magnitude.png','pie_distance.png']

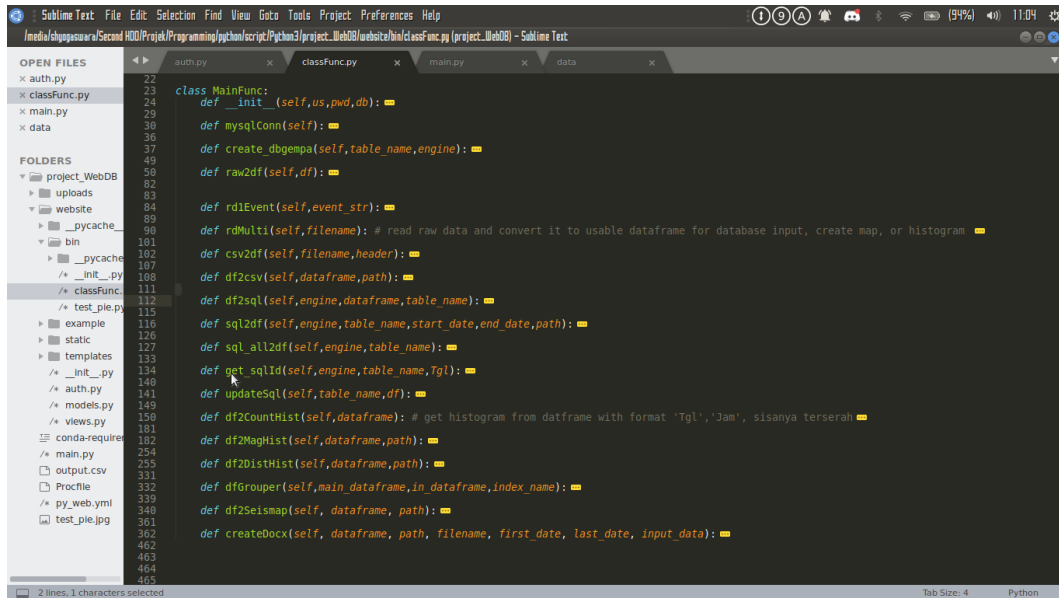
        # Script to make the documents from those picture, and then make it so client can download the doc
        # Input Data Document

        text_data = [len(df), df['M'].max(), df['M'].min(), max_val, max_valId]
        dataframe = [df, df_mag, df_dist]
        doc = mf.createDocx(dataframe,UPLOAD_FOLDER,fname,fDate,lDate,text_data)

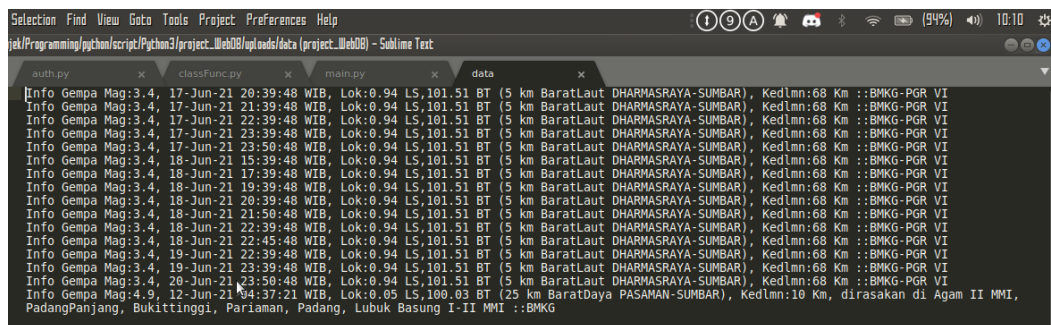
        return send_file(doc, as_attachment=True)

    return render_template('get_data.html')
```

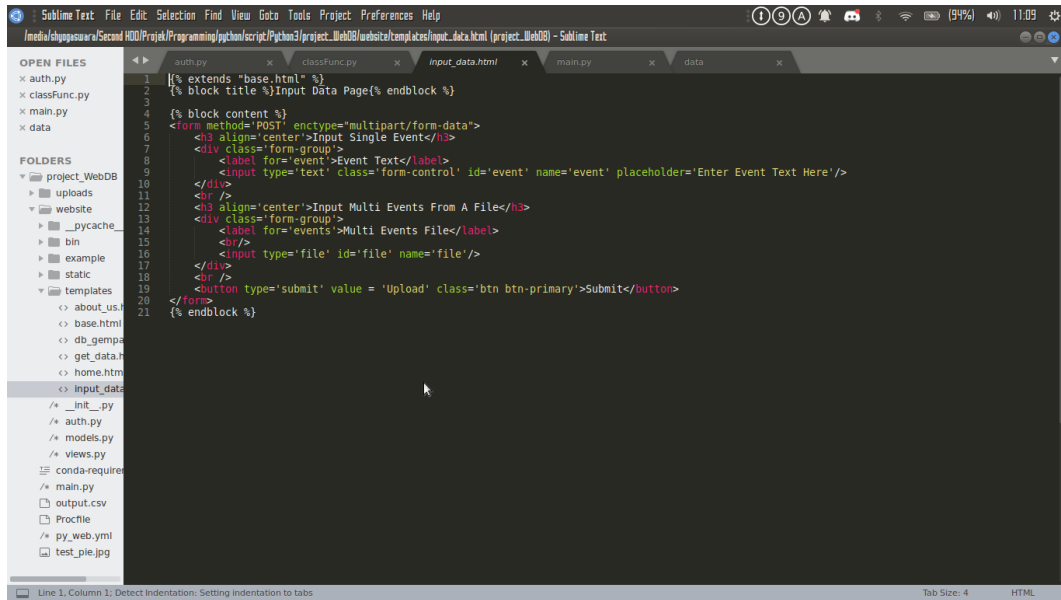
Gambar L.2 Fungsi pada auth.py yang berkaitan dengan proses ambil data melalui tampilan web, terkait dengan gambar 4.2.2



Gambar L.3 Class MainFunc yang berfungsi sebagai backend dari website. Class ini meliputi berbagai fungsi mulai dari koneksi ke database, membuat peta dan grafik, serta menyiapkan dokumen untuk diakses oleh client.

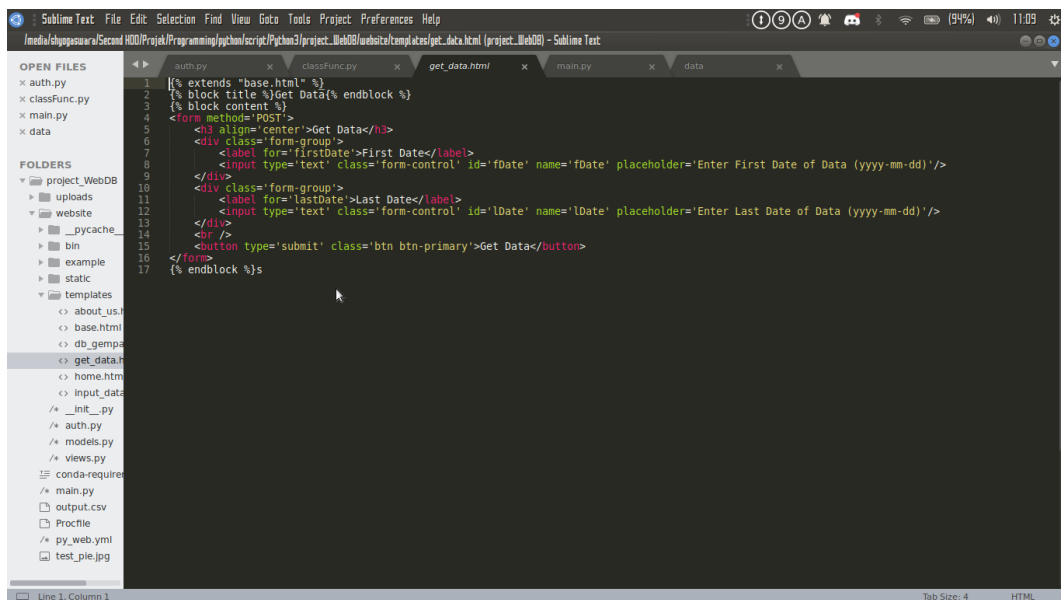


Gambar L.4 Contoh data inputan, terkait gambar 3.3.2 bagian input doc dan bab 4 subbab 4.2.2. Sistem dapat membaca format pesan pendek baik yang biasa maupun yang memiliki data dirasakan secara sekaligus dalam 1 file.



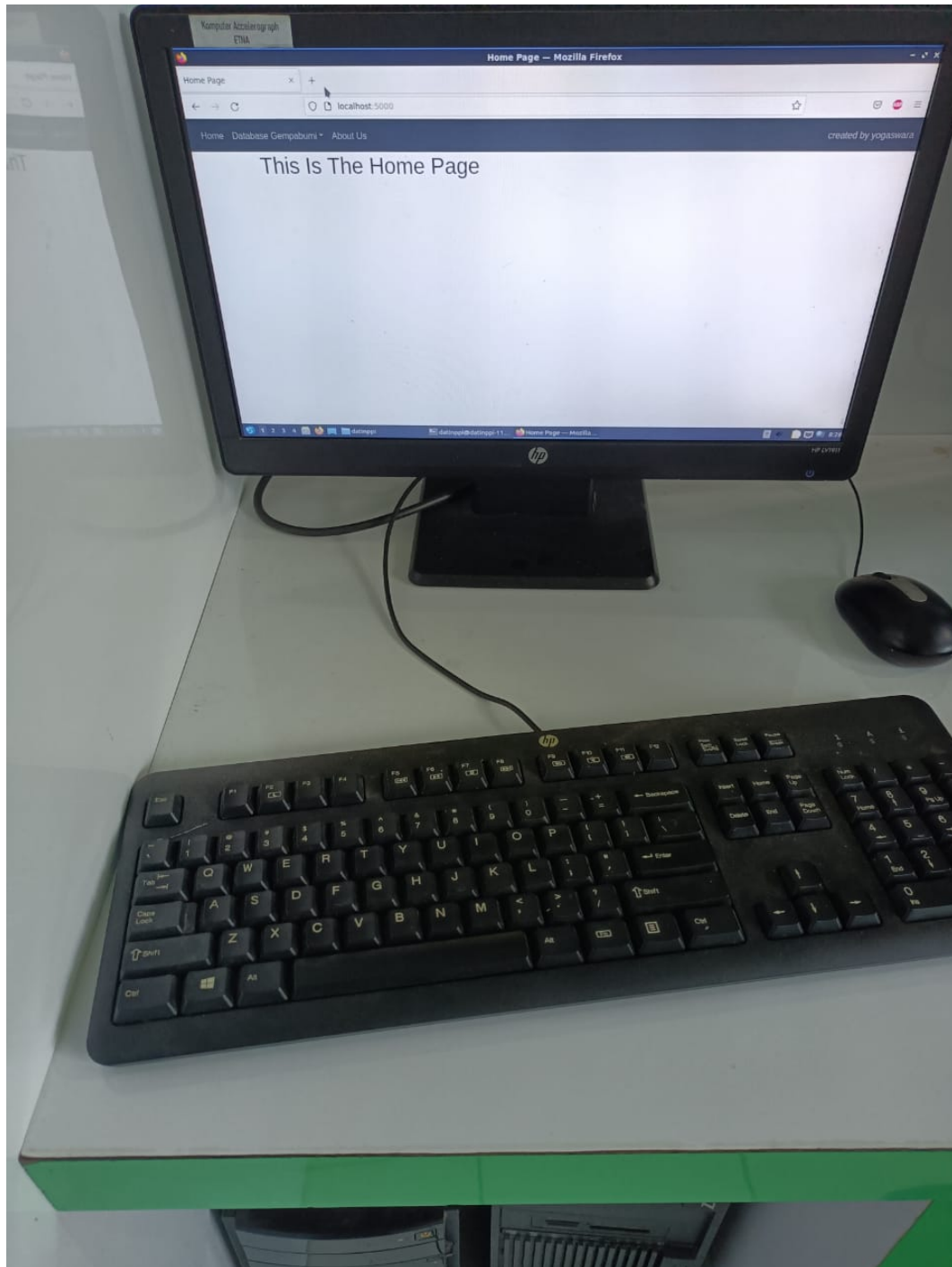
```
1 |{% extends "base.html" %}
2 |{% block title %}Input Data Page{% endblock %}
3 |
4 |{% block content %}
5 |<form method='POST' enctype='multipart/form-data">
6 |<h3 align='center'>Input Single Event</h3>
7 |<div class='form-group'>
8 |<label for='event'>Event Text</label>
9 |<input type='text' class='form-control' id='event' name='event' placeholder='Enter Event Text Here' />
10 |</div>
11 |<br />
12 |<h3 align='center'>Input Multi Events From A File</h3>
13 |<div class='form-group'>
14 |<label for='events'>Multi Events File</label>
15 |<input type='file' id='file' name='file' />
16 |</div>
17 |<button type='submit' value = 'Upload' class='btn btn-primary'>Submit</button>
18 |</form>
19 |
20 |{% endblock %}
```

Gambar L.5 html block tampilan halaman input data, terkait dengan gambar 4.2.1.



```
1 |{% extends "base.html" %}
2 |{% block title %}Get Data{% endblock %}
3 |{% block content %}
4 |<form method='POST'>
5 |<h3 align='center'>Get Data</h3>
6 |<div class='form-group'>
7 |<label form='firstDate'>First Date</label>
8 |<input type='text' class='form-control' id='fDate' name='fDate' placeholder='Enter First Date of Data (yyyy-mm-dd)' />
9 |</div>
10 |<div class='form-group'>
11 |<label form='lastDate'>Last Date</label>
12 |<input type='text' class='form-control' id='lDate' name='lDate' placeholder='Enter Last Date of Data (yyyy-mm-dd)' />
13 |</div>
14 |<br />
15 |<button type='submit' class='btn btn-primary'>Get Data</button>
16 |</form>
17 |{% endblock %}
```

Gambar L.6 html block tampilan halaman ambil data, terkait dengan gambar 4.2.2.



Gambar L.7 Komputer yang digunakan sebagai server ujicoba. Memiliki spesifikasi berupa OS Lubuntu 20.04 64 bit, RAM 2gb DDR2, HDD 250gb. Dapat diakses oleh 2 client sekaligus.