

# WORKSHOP

Data Analytics With Python & SQL

Bakti Siregar, M.Sc

2023-07-05



# Contents

<b>Kata Pengantar</b>	<b>5</b>
Maksud & Tujuan . . . . .	5
Jadwal Kegiatan . . . . .	5
Pemateri & Mediator . . . . .	7
Para Peserta . . . . .	8
Manfaat dan Fasilitas . . . . .	8
Syarat Peserta . . . . .	9
Masukan & Saran . . . . .	9
Ucapan Terima Kasih . . . . .	9
 <b>1 Pendahuluan</b>	 <b>11</b>
1.1 Apa itu SBD? . . . . .	11
1.2 Mengapa Python & SQL? . . . . .	13
1.3 MySQL vs PostgreSQL . . . . .	13
1.4 Instalasi Python (Anaconda) . . . . .	14
1.5 Instalasi MySQL (XAMPP) . . . . .	15
1.6 Instalasi PostgreSQL . . . . .	16
1.7 Praktikal (Hands On) . . . . .	18
1.8 Latihan . . . . .	19
 <b>2 Eksplorasi Data</b>	 <b>21</b>
2.1 Koneksi MySQL dan Python . . . . .	21
2.2 CRUDA Tabel . . . . .	23

2.3	Filter Tabel . . . . .	26
2.4	Tugas Kelompok . . . . .	27
2.5	Cara mengatasi XAMPP tidak berfungsi dengan baik . . . . .	28
<b>3</b>	<b>Perhitungan &amp; Statistik Dasar</b>	<b>29</b>
3.1	Fungsi Skalar vs Agregat . . . . .	29
<b>4</b>	<b>Join &amp; Subkueri Tabel</b>	<b>33</b>
4.1	Pengaturan Struktur Tabel . . . . .	33
<b>5</b>	<b>5W1H &amp; Visualisasi</b>	<b>35</b>
<b>6</b>	<b>Persentasi Tugas Akhir</b>	<b>37</b>
<b>7</b>	<b>Penutup</b>	<b>39</b>
		<b>41</b>

# Kata Pengantar

Di zaman serba digital seperti sekarang ini, data menjadi sumber rujukan utama dalam setiap perencanaan, pengendalian, dan pengembangan diberbagai aspek kehidupan kita. Dampak dari transformasi digital ini adalah semakin bertambahnya kuantitas data yang terus meningkat seperti air mengalir. Sehingga dibutuhkan pengendalian data dengan cepat, tepat dan akurat. Kemampuan dalam menggunakan bahasa pemrograman Python & SQL adalah jawaban bagi setiap individu (professional) yang bekerja dengan data tersebut. Perlu diketahui bahwa Python & SQL sudah terintegrasi dengan sangat baik, mudah dipelajari, fleksibel, dapat digunakan untuk mengakses, memperbarui, memodifikasi, menganalisis data, dan umumnya digunakan dalam basis data relasional yang sangat besar.

## Maksud & Tujuan

Maksud dari kegiatan ini adalah melanjutkan pelatihan data analitik sebelumnya yang berjudul Workshop Data Analytics with Python. Pada workshop kali ini akan digunakan Python dan SQL dengan melibatkan karyawan atau pimpinan instansi masing-masing di JNE. Tujuan yang ingin dicapai dari dilaksanakan pelatihan komputer aplikasi ini adalah :

- a. Mengenalkan kepada para karyawan pentingnya penguasaan Python dan SQL dalam mengolah data.
- b. Melatih untuk menggunakan Python dan SQL dalam praktek pengolahan data sesuai lingkup pekerjaan.
- c. Menerapkan penggunaan Python dan SQL dalam pekerjaan.

## Jadwal Kegiatan

Adapun rencana jadwal kegiatan dan topik pembahasan adalah sebagai berikut

**Pertemuan 1**

Waktu	Nama Kegiatan
09:00 ~ 09:15 WIB	Pembukaan
09:15 ~ 10:15 WIB	Materi I: Pengenalan Python+SQL
10:15 ~ 10:30 WIB	Coffee Break I
10:30 ~ 11:30 WIB	Materi II: Instalasi Python+SQL
11:30 ~ 13:00 WIB	ISOMA
13:00 ~ 14:00 WIB	Materi III : Koneksi Python+SQL
14:00 ~ 15:15 WIB	Materi IV : Pengantar Python+SQL
15:15 ~ 15:30 WIB	ISO
15:30 ~ 16:30 WIB	Kuis atau Praktikum oleh Peserta

**Pertemuan 2**

Waktu	Nama Kegiatan
09:00 ~ 10:15 WIB	Materi I: Pemilahan Data
10:15 ~ 10:30 WIB	Coffee Break I
10:30 ~ 11:30 WIB	Materi II: Pemilahan Data
11:30 ~ 13:00 WIB	ISOMA
13:00 ~ 14:00 WIB	Materi III : Meringkas Data
14:00 ~ 15:15 WIB	Materi IV : Meringkas Data
15:15 ~ 15:30 WIB	ISO
15:30 ~ 16:30 WIB	Kuis atau Praktikum oleh Peserta

**Pertemuan 3**

Waktu	Nama Kegiatan
09:00 ~ 10:15 WIB	Materi I: Mengelompokkan Data
10:15 ~ 10:30 WIB	Coffee Break I
10:30 ~ 11:30 WIB	Materi II: Mengelompokkan Data
11:30 ~ 13:00 WIB	ISOMA
13:00 ~ 14:00 WIB	Materi III : Menggabungkan Data
14:00 ~ 15:15 WIB	Materi IV : Menggabungkan Data
15:15 ~ 15:30 WIB	ISO
15:30 ~ 16:30 WIB	Kuis atau Praktikum oleh Peserta

**Pertemuan 4**

Waktu	Nama Kegiatan
09:00 ~ 10:15 WIB	Materi I: Konversi Waktu
10:15 ~ 10:30 WIB	Coffee Break I
10:30 ~ 11:30 WIB	Materi II: Menghitung Waktu
11:30 ~ 13:00 WIB	ISOMA
13:00 ~ 14:00 WIB	Materi III : Subkueri (Subqueries)
14:00 ~ 15:15 WIB	Materi IV : Subkueri (Subqueries)
15:15 ~ 15:30 WIB	ISO
15:30 ~ 16:30 WIB	Kuis atau Praktikum oleh Peserta

## Pertemuan 5

Waktu	Nama Kegiatan
09:00 ~ 10:15 WIB	Materi I: Pemahaman 5W1H
10:15 ~ 10:30 WIB	Coffee Break I
10:30 ~ 11:30 WIB	Studi Kasus: 5W1H
11:30 ~ 13:00 WIB	ISOMA
13:00 ~ 14:00 WIB	Studi Kasus: 5W1H + Visualisasi
14:00 ~ 15:15 WIB	Studi Kasus: 5W1H + Visualisasi
15:15 ~ 15:30 WIB	ISO
15:30 ~ 16:30 WIB	Kuis atau Praktikum oleh Peserta

## Pertemuan 6

Waktu	Nama Kegiatan
09:00-10:00 WIB	Persentasi Kelompok A
10:00 -10:30 WIB	Coffee Break I
10:30-11:30 WIB	Persentasi Kelompok B
11:30 -13:00 WIB	ISOMA
13:00-14:00 WIB	Persentasi Kelompok C
15:30-15:30 WIB	Pemabahasan Persentasi
15:30-Selesai	Penutupan Acara Pelatihan 2

## Pemateri & Mediator

Berikut ini adalah nama dan biografi singkat para penulis:

- **Bakti Siregar, M.Sc** Bakti Siregar seorang lulusan Magister Sains dari departemen matematika terapan di National Sun Yat-Sen University (Taiwan). Beliau adalah seorang Data Scientist profesional yang memiliki Kompetensi tidak dapat diragukan lagi dalam hal pengolahan data dengan menggunakan SQL dan bahasa pemrograman Python/Rstudio. Lebih dari pada itu, Dia juga akrab dalam memproses Big Data seperti pengenalan distributed computing, NoSQL Database, Hadoop, Spark dan lainnya. Dalam hal visualisasi data, Bakti Siregar mampu menginterpretasikan data dengan sangat baik menggunakan Tableau maupun Power BI. Beberapa project beliau dapat dilihat di link berikut: Rpubs, Github, Website, dan Kaggle.
- **Yosia** adalah salah satu mahasiswa terbaik di jurusan Statistika Universitas Matana. Dia juga memiliki minat dalam pembelajaran sains data dan akuturia khususnya melakukan komputasi dengan menggunakan R dan Python. Yosia bercita-cita suatu saat nanti akan menjadi seseorang yang ahli dibidang aktuaria maupun sains data. Dia memiliki minat teoretis yang luas serta minat dalam komputasi, Yosia pernah terlibat dalam menerbitkan di jurnal Pengabdian Kepada Masyarakat Salah satunya menjadi Assisten (**Workshop Data Analytics With Python & SQL**) yang juga aktif dalam berbagai aktifitas organisasi kampus.
- **Clara Della** adalah mahasiswi jurusan Statistik di Universitas Matana. Dia memiliki kemampuan dalam menggunakan beberapa perangkat lunak yang dibutuhkan dalam proses analisa data, seperti; R, Python, SQL, Tableau dan SPSS. Dia juga sudah pernah terlibat menjadi salah satu Assisten (**Workshop Data Analytics With Python & SQL**). Selain itu, Della juga aktif dalam berbagai aktifitas organisasi kampus.

## Para Peserta

Adun para peserta pelatihan adalah para karyawan JNE dengan jumlah maksimum peserta 15 orang, sebagai berikut:

## Manfaat dan Fasilitas

- Instruktur Berpengalaman di Bidang Analis Data (Akademisi & Praktisi)
- Pelatihan 6 hari (8 Jam Per-pertemuan)
- Modul Pelatihan (E-Book yang dapat diakses hanya untuk Peserta dari JNE)
- Sertifikat Kehadiran (Untuk peserta yang tidak lulus sertifikat kompetensi)



- Sertifikat Kompetensi dari Dsciencelabs (Analisis Data Lanjutan)
- Laporan Kegiatan Pelatihan

## Syarat Peserta

- Laptop yang sudah ter-install Anaconda dan XAMPP atau PostgreSQL
- Sudah terbiasa dengan data, minimal menggunakan Ms. Excel
- Berkemauan tinggi untuk belajar Koding

## Masukan & Saran

Akses untuk Ebook ini hanya tersedia secara gratis untuk para peserta Workshop di JNE. Meskipun banyak upaya telah dilakukan untuk pengembangan, tentu kami sangat mengharapkan masukan dari anda sekalian.

## Ucapan Terima Kasih

Kami juga ingin mengucapkan terima kasih yang sebesar-sebesar pada semua pihak yang terlibat dalam pengembangan buku ini, yakni; mahasiswa-i, dosen, dan Universitas Matana atas dukungan dalam upaya bersama kami untuk menyediakan materi untuk **Pelatihan Analisis Data Dengan Menggunakan Python dan SQL** ini.



# Chapter 1

## Pendahuluan

Sejak tahun 1970, **Structured Query Language (SQL)** telah digunakan oleh para programmer untuk membangun dan mengakses **Sistem Basis Data (SBD)**. Banyak sekali perdebatan mengenai cara penyebutan SQL ini, namun pada kenyataannya, kita dapat melafalkannya sebagai “sequel” ataupun “S.Q.L”. Mempelajari bahasa pemrograman umum seperti Python dan R adalah penting, namun tidak memiliki kemampuan SQL akan membuat kita amat sulit untuk mendapatkan pekerjaan dalam bidang pengolahan data.

Hampir semua nama besar di bidang teknologi menggunakan SQL seperti Uber, Netflix, dan Airbnb. Bahkan dalam perusahaan seperti Facebook, Google dan Amazon, yang telah membuat sendiri **SBD** berkemampuan tinggi, tetap menggunakan SQL untuk melakukan query dan analisis data.

### 1.1 Apa itu SBD?

Secara umum **SBD** dapat didefinisikan sebagai berikut:

#### 1.1.1 Komponen SBD

Adapun beberapa komponen dasar yang diperlukan dalam SBD adalah:

---

#### 1.1.2 Manfaat SBD

Manfaat atau kegunaan penerapan SBD cukup banyak dan cakupannya pun luas dalam mendukung keberadaan lembaga atau organisasi maupun perusahaan, diantaranya:

### 1.1.3 Definisi SQL vs NoSQL

Sebenarnya perbedaan antara SQL dan NoSQL secara mendasar sudah dapat dijelaskan dari akronimnya. *SQL* basis data relasional yang menggunakan ‘relasi’ (yang biasanya disebut tabel) untuk menyimpan data dan mencocokkan data tersebut dengan memakai karakteristik umum di setiap dataset.

Sedangkan, *NoSQL* adalah database yang menggunakan format JSON untuk setiap dokumennya sehingga mudah dibaca dan dimengerti. NoSQL banyak diminati karena memiliki performa yang tinggi dan bersifat non-relasional sehingga dapat memakai berbagai model data.

---

### 1.1.4 Perbedaan SQL vs NoSQL

Sebenarnya banyak perbedaan yang dimiliki di antara dua database tersebut tapi inilah perbedaan yang paling mencolok antara SQL dan NoSQL:

---

### 1.1.5 Top 7 SQL

Tercatat sampai bulan Februari 2020 ada 334 jenis database menurut db-engines.com. Berikut ini saya merangkum daftar 7 database terpopuler yang menggunakan SQL (Relasional):

---

### 1.1.6 Top 8 NoSQL

Kebanyakan basis data NoSQL digunakan dalam dunia aplikasi web waktu nyata (real-time web app). Berikut ini adalah ulasan 8 jenis basis data NoSQL yang paling populer digunakan diseluruh dunia:

---

## 1.2 Mengapa Python & SQL?

**Python** adalah sebuah bahasa pemrograman yang digunakan untuk membuat aplikasi, perintah komputer, dan melakukan analisis data. **SQL** adalah sintaks (bahasa) yang digunakan untuk mengatur dan mengambil informasi dari sistem basis data relasional dan yang juga dapat digunakan untuk membuat basis data sendiri. Jadi, kedua perangkat lunak ini mempunyai peran masing-masing. SQL ditujukan untuk mengakses dan mengambil data dari database. Sedangkan Python digunakan untuk menganalisis dan memanipulasi data dengan menggunakan regresi, time series maupun perhitungan lainnya.

Jika kita ingin bekerja di bidang pengolahan data, kita harus fokus kepada kemampuan yang diinginkan oleh para perusahaan pemberi kerja tersebut. Untuk posisi analis data, SQL termasuk kemampuan yang diutamakan, sekitar 57.4% dari keseluruhan lowongan pekerjaan. Berdasarkan survei yang dilakukan Dataquest dan StackOverflow pada gambar dibawah ini, dapat dilihat bahwa SQL bahkan lebih populer di kalangan data scientist dan data engineer dibandingkan Python dan R.

## 1.3 MySQL vs PostgreSQL

MySQL adalah sistem manajemen basis data relasional yang memungkinkan Anda untuk menyimpan data sebagai tabel dengan baris dan kolom. Sistem ini populer sehingga digunakan di banyak aplikasi web, situs web dinamis, dan sistem tertanam. PostgreSQL adalah sistem manajemen basis data relasional-objek yang menawarkan lebih banyak fitur daripada MySQL. Sistem ini memberi Anda lebih banyak fleksibilitas dalam tipe data, skalabilitas, konkurensi, dan integritas data.

MySQL dan PostgreSQL, Keduanya menyimpan data di dalam tabel yang terkait satu sama lain melalui nilai kolom umum. Namun keduanya sering dibandingkan karena terdapat beberapa perbedaan. Ingin mengenal lebih dalam? Simak penjelasan di bawah.

### 1.3.1 Kelebihan

MySQL	PostgreSQL
Integrasi bahasa pemrograman sangat luas;	Support framework website modern seperti Node.js dan Django; Support framework website modern seperti Node.js dan Django;

MySQL	PostgreSQL
Aplikasi ringan, tidak membutuhkan spesifikasi hardware yang tinggi;	Dirilis dengan lisensi PostgreSQL sendiri;
Struktur tabel dengan fleksibilitas tinggi;	Bersifat open source dan gratis;
Dibekali banyak administrative tools;	Skala besar, mampu memuat hingga ribuan transaksi data;
Bersifat open source dan gratis (versi basic);	Memiliki banyak fitur yang mumpuni;
Meski open source, MySQL menjamin keamanan tingkat tinggi;	Memiliki banyak fitur yang mumpuni;
Mendukung berbagai variasi Data Type;	Performa sangat baik meski menuntut query yang lebih kompleks;
Dapat digunakan banyak pengguna karena mendukung multi user.	Kecepatan analisis (read-write) sangat cepat; Keamanan yang lebih ketat.

### 1.3.2 Kekurangan

MySQL	PostgreSQL
Sistem manajemen database kurang cocok untuk aplikasi mobile dan game;	PostgreSQL tidak mendukung semua stack development;
Technical support MySQL dinilai kurang baik;	Meski memiliki integrasi dan skalabilitas tinggi, kecepatan PostgreSQL kalah unggul dibandingkan RDBMS lain;
Sulit diaplikasikan untuk manajemen database berskala besar.	Sistem kompatibilitas PostgreSQL menuntut pengguna untuk bekerja lebih keras dalam perbaikan dan perawatan.

## 1.4 Instalasi Python (Anaconda)

Berikut ini adalah proses langkah demi langkah tentang Cara Menginstal Anaconda di Windows:

### 1.4.1 Open site

Buka Situs Web Anaconda Lalu pilih Download pada logo Windows

### 1.4.2 Buka file exe yang telah didownload

Ketika layar di bawah ini muncul, klik Next.

### 1.4.3 Baca perjanjian lisensi dan klik I Agree.

### 1.4.4 Click on Next

### 1.4.5 Catat lokasi penginstalan Anda, lalu klik Next.

### 1.4.6 Customiza Anaconda

Ini adalah bagian penting dari proses instalasi. approach yang disarankan adalah tidak mencentang kotak untuk menambahkan Anaconda ke path Anda. Ini berarti Anda harus menggunakan Anaconda Navigator atau Anaconda Command Prompt (terletak di Start Menu di bawah “Anaconda”) ketika Anda ingin menggunakan Anaconda (Anda selalu dapat menambahkan Anaconda ke PATH Anda nanti jika Anda tidak mencentang kotak). Jika Anda ingin menggunakan Anaconda pada command prompt Anda (atau git bash, cmd, powershell, dll), gunakan alternative approach dan centang kotaknya.

### 1.4.7 Setup Complete

###Install Microsoft VSCode

Anda dapat menginstal Microsoft VSCode jika Anda mau, tetapi ini opsional.

### 1.4.8 Selesai

Click on Finish.

### 1.4.9 Video Instalasi

- Download Anaconda

## 1.5 Instalasi MySQL (XAMPP)

### 1.5.1 Download Aplikasi XAMPP

Silakan klik [disini](#) untuk mengunduh aplikasi XAMPP, pilih salah satu saja sesuai Operating System pada Komputer anda.

### 1.5.2 Install Aplikasi

Temukan file XAMPP.exe yang telah anda download, secara default biasanya disimpan di;

Selanjutnya, akan muncul Warning di klik **OK**

selanjutnya klik next

Klik next lagi, karena sudah dipilih secara default oleh XAMPP

### 1.5.3 Pilih Folder

Secara default akan membuat folder baru **C:\xampp**, lalu pilih next.

**note:** jika anda sudah pernah mendownload aplikasi xampp, perlu di hapus terlebih dahulu file xampp yang lama di file **C:\xampp**

### 1.5.4 Jalankan proses Instalasi

Tunggu proses instalasi selesai **Biasanya 5-10 menit, tergantung kecepatan komputer anda.**

### 1.5.5 XAMPP sudah terinstall

Setelah aplikasi terinstall, sudah bisa digunakan.

### 1.5.6 Video Instalasi XAMPP

## 1.6 Instalasi PostgreSQL

Berikut ini adalah proses langkah demi langkah tentang Cara Menginstal PostgreSQL di Windows:

### 1.6.1 Buka Browser

Klik <https://www.postgresql.org/download> and pilih Windows

### 1.6.2 Cek Option

Klik Download the installer Interactive Installer by EnterpriseDB



### 1.6.3 Pilih PostgreSQL version

Anda akan diminta untuk memilih versi PostgreSQL dan sistem operasi yang diinginkan. Pilih versi PostgreSQL terbaru dan OS sesuai dengan environment Anda, **klik tombol download**.

### 1.6.4 Open exe file

Setelah Anda mengunduh PostgreSQL, buka exe yang telah diunduh dan Klik berikutnya pada layar install welcome screen.

### 1.6.5 Pilih folder

Ubah direktori Instalasi jika diperlukan, jika tidak, biarkan default, **klik Next**.

### 1.6.6 Select components

Anda dapat memilih komponen yang ingin Anda instal di sistem Anda. Anda dapat menghapus centang pada Stack Builder (*disarankan ikuti secara default*), **klik Next**.

### 1.6.7 Check data location

Anda dapat mengubah lokasi data, **Klik Next**.

### 1.6.8 Masukkan Password

Masukkan kata sandi superuser. Catat kata sandi tersebut, **Klik Next**.

### 1.6.9 Cek opsi port

Biarkan nomor port menjadi default, **Klik Next**.

### 1.6.10 Cek Summary

Periksa pra-penginstalan summary, **Klik Next**

### 1.6.11 Ready to Install

Klik tombol Next

### 1.6.12 Check stack builder prompt

Setelah instalasi selesai, Anda akan melihat prompt Stack Builder. Hapus centang pada opsi tersebut. Kita akan menggunakan Stack Builder dalam tutorial selanjutnya, **Klik Finish**.

### 1.6.13 Launch PostgreSQL

Untuk launch PostgreSQL, buka Start Menu dan cari pgAdmin 4

### 1.6.14 Check pgAdmin

Anda akan melihat beranda pgAdmin

### 1.6.15 Cari PostgreSQL 15

Klik pada Servers > PostgreSQL 15 di sub sebelah kiri

### 1.6.16 Enter password

Masukkan kata sandi superuser yang ditetapkan selama instalasi, **Klik OK**

### 1.6.17 Cek Dashboard

Anda akan melihat Dashboard

### 1.6.18 Video Instalasi PostgreSQL

- Download PostgreSQL

## 1.7 Praktikal (Hands On)

Untuk memulai bagian ini, pastikan bahwa anda sudah meng-install Anaconda dan XAMPP di komputer yang akan digunakan. Kemudian, bukalah aplikasi Jupyterlab yang ada pada layar Anaconda Navigator, seperti diperlihatkan sebagai berikut:

Selanjutnya, akan muncul halaman baru pada browser seperti gambar berikut;

Selanjutnya, klik 'File > Open from URL'. Muncul tampilan sebagai berikut:

Terakhir, Copy-Paste dibawah ini pada kolom `Open URL`. Selanjutnya, ikuti tutorialnya.

- **Akses Databases MySQL dengan Python**
- **Akses Databases PostgreSQL dengan Python**
- **Contoh Aplikasi Sederhana Py\_SQL**

## 1.8 Latihan

Berikut ini adalah Soal Latihan yang harus dikerjakan bersama Kelompok masing-masing.

- **Kelompok A** Klik disini
  1. KOKO ARIF BUDIMAN
  2. ALDINO PRASETIAWAN
  3. SURISTIN ADNAN
  4. HAERUL
  5. AGUNG DARMANTO
- **Kelompok B** Klik disini
  1. M. SYAHRIZAL
  2. FAJAR DEMOKRASI PUTRA
  3. HARRY KHAIRULLAH
  4. IBNU FALAH
  5. RESTU KUSUMAWATI
- **Kelompok C** Klik disini
  1. ADE DARMAWAN
  2. RETNO WULANDARI
  3. FADILLA ANGGIE PRANANDA
  4. WISMOYO BRIHAYADI
  5. YUDHA PRATAMA



## Chapter 2

# Eksplorasi Data

Pada pertemuan sebelumnya telah dipelajari pengenalan dasar bagaimana mengakses basis data dengan menggunakan Python ke MySQL dan PostgreSQL. Pada pertemuan kedua ini, akan fokus untuk berlatih untuk melakukan CRUDA dan Filter tabel. Secara garis besar materi yang akan dibahas adalah;

- Mengulang kembali bagaimana meng-akses basis data dengan Python ke MySQL dan PostgreSQL
- Membuat dan menghapus basis data
- Membuat database lengkap dengan melakukan import data dari Excel (**database rekayasa untuk pelatihan Data Analytics di JNE**)
- Operasi tabel pada database
- Menambahkan pengaturan batasan tabel

### 2.1 Koneksi MySQL dan Python

Untuk mengakses basis data yang ada pada MySQL dengan Python dapat menggunakan koding berikut di Jupiterlab anda.

```
# import module
import mysql.connector

# koneksi server
db1 = mysql.connector.connect(
    host='localhost',
    user='bakti',
    passwd='123'
)
```

Jika tidak muncul pesan error pada saat menjalankan koding diatas pada Chunck Jupiterlab anda. Artinya, koneksi Python ke MySQL sudah terhubung dengan baik.

### 2.1.1 Memeriksa Daftar Database

Setelah kita terhubung dengan local server menggunakan Python dan MySQL. Sekarang saatnya kita mulai untuk memeriksa database apa saja yang ada didalamnya.

```
# membuat objek 'cursor' sebagai kata kunci eksekusi SQL
cursor1 = db1.cursor()

# Perintah SQL
sql = 'SHOW DATABASES'

# Eksekusi perintah SQL
cursor1.execute(sql)

# menampilkan daftar database
for x in cursor1:
    print(x)
```

### 2.1.2 Membuat Database

Berikut ini adalah koding yang dapat digunakan untuk membuat database baru dengan menggunakan koneksi `db.cursor()` yang ada pada objek `cursor`.

```
sql= 'CREATE DATABASE jne2'
cursor1.execute(sql)
```

### 2.1.3 Menghapus Database

Jika anda ingin menghapus database tertentu dari lokal server, maka dapat melakukannya dengan koding berikut:

```
sql= 'DROP DATABASE jne2'
cursor1.execute(sql)
```

### 2.1.4 Memeriksa Daftar Tabel

Untuk memeriksa daftar tabel yang ada pada local server. Perhatikan Koding berikut:

```

# import module
import mysql.connector

# koneksi salah satu database MySQL ke Python
db2 = mysql.connector.connect(
    host='localhost',
    user='bakti',
    passwd='123',
    database='jne2'
)

# membuat objek 'cursor' sebagai kata kunci eksekusi SQL
cursor2 = db2.cursor()

# Perintah SQL
sql = "SHOW TABLES"

# Eksekusi perintah SQL
cursor2.execute(sql)

# menampilkan daftar tabel
for x in cursor2:
    print(x)

```

## 2.2 CRUDA Tabel

Pada materi sebelumnya telah dipelajari bagaimana untuk melakukan CRUDA Tabel yang ada pada database. CRUDA merupakan singkatan dari **C**reate, **R**ead, **U**ppdate, **D**elele, dan **A**lter. Berikut ini akan diperlihatkan cara untuk menambahkan daftar database dari CSV. Langkah pertama, import data dari Excel menggunakan Python.

```

import pandas as pd

link_data = 'https://raw.githubusercontent.com/dsciencelabs/JNE_Data_Analytics_PySQL/main/data/data.csv'

df1 = pd.read_csv(link_data, sep=';' )
df1.head(5)

```

### 2.2.1 CREATE

Jika data tabel yang telah di import dari file CSV sudah ada dalam memori Jupiterlab (**Python**) anda, maka selanjutnya untuk menambahkan tabel tersebut

ke database adalah sebagai berikut:

```
import pymysql
from sqlalchemy import create_engine

# membuat engine
engine = create_engine("mysql+pymysql://{user}:{pw}@localhost/{db}"
                      .format(user="bakti",
                              pw="123",
                              db="jne2"))

# df1.to_sql('users', engine, if_exists = 'append', index = True)
df1.to_sql('user', engine, if_exists='replace', index = False)
```

### Tugas Kelompok:

Tambahkan tabel berikut ke database `jne2` dengan mengikuti materi yang sudah anda pelajari sebelumnya.

- [https://raw.githubusercontent.com/dsciencelabs/JNE\\_Data\\_Analytics\\_PySQL/main/data/data\\_latihan/products.csv](https://raw.githubusercontent.com/dsciencelabs/JNE_Data_Analytics_PySQL/main/data/data_latihan/products.csv)
- [https://raw.githubusercontent.com/dsciencelabs/JNE\\_Data\\_Analytics\\_PySQL/main/data/data\\_latihan/orders.csv](https://raw.githubusercontent.com/dsciencelabs/JNE_Data_Analytics_PySQL/main/data/data_latihan/orders.csv)
- [https://raw.githubusercontent.com/dsciencelabs/JNE\\_Data\\_Analytics\\_PySQL/main/data/data\\_latihan/order\\_items.csv](https://raw.githubusercontent.com/dsciencelabs/JNE_Data_Analytics_PySQL/main/data/data_latihan/order_items.csv)
- [https://raw.githubusercontent.com/dsciencelabs/JNE\\_Data\\_Analytics\\_PySQL/main/data/data\\_latihan/inventory\\_items.csv](https://raw.githubusercontent.com/dsciencelabs/JNE_Data_Analytics_PySQL/main/data/data_latihan/inventory_items.csv)
- [https://raw.githubusercontent.com/dsciencelabs/JNE\\_Data\\_Analytics\\_PySQL/main/data/data\\_latihan/events.csv](https://raw.githubusercontent.com/dsciencelabs/JNE_Data_Analytics_PySQL/main/data/data_latihan/events.csv)
- [https://raw.githubusercontent.com/dsciencelabs/JNE\\_Data\\_Analytics\\_PySQL/main/data/data\\_latihan/distribution\\_centers.csv](https://raw.githubusercontent.com/dsciencelabs/JNE_Data_Analytics_PySQL/main/data/data_latihan/distribution_centers.csv)

### 2.2.2 READ

Berbeda dengan materi materi dasar yang kita pelajari sebelumnya, untuk membaca data tabel dari database dalam bentuk dataframe dapat dilakukan dengan cara berikut;

```
import pandas as pd

en_co = engine.connect()

read1 = pd.read_sql_table('distribution_centers', en_co)
read1
```



### 2.2.3 UPDATE

Sama seperti yang telah dipelajari sebelumnya, melakukan update data pada tabel disini dapat dilakukan dengan cara berikut:

```
import pandas as pd

en_co = engine.connect()

sql = 'UPDATE distribution_centers SET name=%s, latitude=%s, longitude=%s WHERE id=%s'
val = ("JNE", 106.7977, -6.1774, 10)
cursor2.execute(sql, val)

db2.commit()

pd.read_sql_table('distribution_centers', en_co)
```

#### Tugas Kelompok:

- Tambahkan satu lokasi baru pada tabel `distribution_centers`
- Tambahkan tiga lokasi baru pada tabel `distribution_centers`

### 2.2.4 DELETE

Pada bagian hapus data sama seperti update data. Kita harus menggunakan query DELETE untuk menghapus dan WHERE untuk menentukan data yang akan dihapus.

```
sql = "DELETE FROM distribution_centers WHERE id=%s"
val = (11, )
cursor2.execute(sql, val)

db2.commit()

print("{} data dihapus".format(cursor2.rowcount))
```

#### Tugas Kelompok:

- Hapuskan tiga lokasi baru sebelumnya, yang anda tambahkan pada tabel `distribution_centers` secara bersamaan.

**Catatan:** Untuk menghapus kolom dalam tabel, gunakan sintaks berikut (perhatikan bahwa beberapa sistem database tidak mengizinkan penghapusan kolom):

```
sql = "ALTER TABLE distribution_centers \  
      DROP COLUMN email"  
  
cursor2.execute(sql)  
  
db2.commit()
```

### 2.2.5 ALTER

Ada kalanya kita ingin menambah, memodifikasi, menghapus yang ada pada tabel. Pernyataan yang digunakan adalah `ALTER TABLE`, seperti diperlihatkan sebagai berikut:

```
sql = 'ALTER TABLE distribution_centers \  
      ADD email VARCHAR(100) \  
      FIRST'  
cursor2.execute(sql)  
  
db2.commit()
```

## 2.3 Filter Tabel

Berikut ini adalah beberapa cara yang dapat digunakan untuk memilah-milah tabel dari database.

### 2.3.1 SELECT

```
pilih_tab = pd.read_sql('SELECT * FROM distribution_centers', db2)  
pilih_tab
```

Untuk memilih beberapa kolom saja dari tabel adalah sebagai berikut:

```
pilih_kol = pd.read_sql('SELECT id, name FROM distribution_centers', db2)  
pilih_kol
```

### 2.3.2 WHERE

Kondisi yang dinyatakan pada klausa `WHERE` bisa simpel ataupun kompleks. Kita dapat menggabungkan beberapa kondisi menggunakan logical connectives

(AND, OR dan NOT) menjadi sebuah kondisi tunggal. Berikut adalah contoh penggunaan klausa WHERE:

```
sql = 'SELECT * FROM distribution_centers WHERE id<=3'
kondisi = pd.read_sql(sql, db2)
kondisi
```

```
sql = 'SELECT * FROM distribution_centers WHERE id=3 AND name= "Houston TX"'
kondisi = pd.read_sql(sql, db2)
kondisi
```

### 2.3.3 ORDER BY

Klausa ORDER BY digunakan untuk menampilkan output dalam bentuk urutan alfabet ASCending atau DESCending.

```
sql = 'SELECT * FROM distribution_centers ORDER BY name ASC;'
or_by = pd.read_sql(sql, db2)
or_by
```

### 2.3.4 LIMIT

Klausa LIMIT digunakan untuk membatasi hasil query kita sesuai dengan angka yang kita tetapkan.

```
sql = 'SELECT * FROM distribution_centers LIMIT 3;'
or_by = pd.read_sql(sql, db2)
or_by
```

Coba perhatikan koding berikut:

```
sql = 'SELECT * FROM distribution_centers LIMIT 3,5'
or_by = pd.read_sql(sql, db2)
or_by
```

## 2.4 Tugas Kelompok

Diskusikan dan Kerjakan soal berikut bersama Kelompok Masing2.

- Tambahkan lah kolom `Product_Price` pada table `inventory_items`

- Hapuslah kolom `Product_Price` pada table `inventory_items`
- Tambahkan kolom `Id` pada table `orders`
- Hapuslah kolom `Id` pada table `orders`
- Tambahkan kolom `Total_Sales` pada table `order_items`
- Hapuslah kolom `Total_Sales` pada table `order_items`
- Tampilkanlah data yang `order_id >= 50` dari table `orders`
- Tampilkanlah data berdasarkan negara london dari table `events`
- Tampilkanlah data yang memiliki `product_id = 9482` dari table `inventory_items`
- Tampilkanlah data yang memiliki `id = 72884` dan dengan status cancelled dari table `order_items`
- Urutkanlah Data dari table `distribution_centers` dengan nama dari huruf z ke a
- Urutkanlah Data dari table `order_items` dengan `sale_price` dari terkecil ke besar
- Tambahkan Tabel dari local (tidak menggunakan link) ke database `jne2`
- Upload Tugas 2 Kelompok A Disini
- Upload Tugas 2 Kelompok B Disini
- Upload Tugas 2 Kelompok C Disini

## 2.5 Cara mengatasi XAMPP tidak berfungsi dengan baik

<https://www.amalankomputer.com/2021/01/cara-mengatasi-apache-shutdown.html>

## Chapter 3

# Perhitungan & Statistik Dasar

Pada dasarnya SQL bukan saja hanya dapat dipergunakan untuk melakukan manajemen database. Tetapi, dapat juga melakukan perhitungan matematika dan statistika dasar yang pada akhirnya dapat disimpan pada database untuk keperluan analisa. Secara spesifik, pelatihan ini akan membahas beberapa diantaranya yakni mengenai penggunaan fungsi sklar dan fungsi agregat.

### 3.1 Fungsi Skalar vs Agregat

**Fungsi skalar** atau fungsi math dalam SQL digunakan untuk mengembalikan nilai tunggal (*single value*) dari suatu nilai input yang diberikan pada perintah sebelumnya. Sedangkan, **Fungsi agregat** dalam SQL digunakan untuk melakukan perhitungan matematika pada sekelompok nilai dan kemudian mengembalikan nilai tunggal.

#### 3.1.1 Perhitungan Skalar

Sebelum kita memulai melakukan beberapa perhitungan skalar pada data yang kita pilih dari database. Pertama-tama, mari kita lihat dulu isi data pada tabel berikut:

```
from sqlalchemy import create_engine
import pymysql
import pandas as pd
```

```

# membuat engine
engine = create_engine("mysql+pymysql://{user}:{pw}@localhost/{db}"
                        .format(user="bakti",
                                pw="123",
                                db="jne2"))

# engine koneksi ke database
db_enco = engine.connect()

# Perintah SQL
sql = "inventory_items"

# Eksekusi perintah SQL dengan Python (panda)
pd.read_sql(sql,db_enco)

```

Berikut ini adalah beberapa contoh dan deskripsi Fungsi Skalar yang terdapat dalam SQL:

**Syntax fungsi ABS()** adalah sebagai untuk menampilkan nilai absolut dari suatu bilangan bulat. Sebelum memulai menggunakan fungsi ABS(), perlu untuk membuat objek koneksi dan kursor eksekusinya terlebih dahulu.

```

# import module
import mysql.connector

# koneksi salah satu database MySQL ke Python
db1 = mysql.connector.connect(
    host='localhost',
    user='bakti',
    passwd='123',
    database='jne2'
)

# membuat objek 'cursor' sebagai kata kunci eksekusi SQL
cursor1 = db1.cursor()

```

Selanjutnya, berikut ini diperlihatkan contoh penerapan fungsi ABS().

```

# Perintah SQL
sql = "SELECT ABS(cost) \
      FROM inventory_items \
      WHERE product_id=16898"

# Eksekusi perintah SQL
cursor1.execute(sql)

```

```
# menampilkan daftar tabel
for data in cursor1:
    print(data)
```

Dari hasil diatas diperlihatkan beberapa hasil yang sama. Untuk menangani hal tersebut, perhatikan koding berikut:

```
# Perintah SQL
sql = "SELECT DISTINCT ABS(cost) \
      FROM inventory_items \
      WHERE product_id=16898"

# Eksekusi perintah SQL
cursor1.execute(sql)

# menampilkan daftar tabel
for data in cursor1:
    print(data)
```

### 3.1.2 Perhitungan Agregat

Berikut ini adalah beberapa contoh dan deskripsi Fungsi Agregat yang terdapat dalam SQL:

**\*\*Fungsi COUNT(\*)\*\*** mengembalikan jumlah baris yang dikembalikan oleh pernyataan SELECT, termasuk NULL dan duplikat, sedangkan COUNT(column) mengembalikan jumlah baris yang dikembalikan oleh klausa SELECT.

```
# Perintah SQL
sql = "SELECT COUNT(product_category) \
      FROM inventory_items \
      WHERE cost<5"

# Eksekusi perintah SQL
cursor1.execute(sql)

# menampilkan daftar tabel
for data in cursor1:
    print(data)
```





## Chapter 4

# Join & Subkueri Tabel

### 4.1 Pengaturan Struktur Tabel

Pengaturan Struktur Tabel SQL digunakan untuk menentukan aturan struktur data dalam tabel. Batasan digunakan untuk membatasi jenis data yang dapat masuk ke dalam tabel, memastikan keakuratan dan keandalan data dalam tabel. Batasan dapat berupa tingkat kolom atau tingkat tabel. Batasan tingkat kolom berlaku untuk kolom, dan batasan tingkat tabel berlaku untuk keseluruhan tabel. Berikut Batasan yang biasanya digunakan dalam SQL:

- *NOT NULL*: Memastikan kolom tidak boleh memiliki nilai NULL
- *UNIQUE*: Memastikan bahwa semua nilai dalam kolom berbeda
- *PRIMARY KEY*: Kombinasi NOT NULL dan UNIQUE. Secara unik mengidentifikasi setiap baris dalam tabel
- *FOREIGN KEY*: Secara unik mengidentifikasi baris/record di tabel lain
- *CHECK*: Memastikan bahwa semua nilai dalam kolom memenuhi kondisi tertentu
- *DEFAULT*: Menetapkan nilai default untuk kolom saat tidak ada nilai yang ditentukan
- *INDEX*: Digunakan untuk membuat dan mengambil data dari database dengan sangat cepat
- *AUTO\_INCREMENT*: memungkinkan inset data unik pada kolom tertentu yang dihasilkan secara otomatis saat data baru dimasukkan ke dalam tabel. Seringkali kolom ini adalah adalah juga *PRIMARY KEY* dan ingin dibuat secara otomatis setiap kali catatan baru dimasukkan.



## Chapter 5

### 5W1H & Visualisasi



## Chapter 6

# Persentasi Tugas Akhir



## Chapter 7

# Penutup





