

LAPORAN TUGAS BESAR BASIS DATA
PYTHON COURSES IN COURSERA
KELOMPOK 7 - RA



Dosen Pengampu:

Riksa Meidy Karim, S.Kom., M.Si., M.Sc
Amalya Citra Pradana, S.Kom., M.Si., M.Sc.
Tirta Setiawan, S.Pd., M.Si..

Anggota Kelompok :

Dimas Wahyu Saputro (120450081)
Armanda Tantra Kumara (120450037)
Taj Shavira H. (120450059)
Fadia Dilla Sabine (120450011)
Khoirunnisa (120450029)

INSTITUT TEKNOLOGI SUMATERA
LAMPUNG SELATAN
2021/2022

1

Latar Belakang

Saat ini banyak sekali bahasa pemrograman yang digunakan oleh para developer, salah satunya adalah Python. Karena maraknya dan peningkatan penggunaan bahasa pemrograman Python, tidak sedikit orang yang penasaran dan mempelajari bahasa pemrograman tersebut. Menurut survei yang dilakukan TIOBE Index, Python berada di peringkat ke-5 pada tahun 2016 sebagai bahasa pemrograman terpopuler. Python merupakan bahasa pemrograman tingkat tinggi karena dapat digunakan untuk membuat aplikasi baik berupa desktop, web serta mobile.

Oleh karena itu, pada tugas kali ini kami menggunakan dataset yang berisikan beberapa orang atau kelompok yang ikut dalam kursus pelatihan bahasa pemrograman Python. Data tersebut kami ambil dari [Python Courses on Coursera | Kaggle](#) dengan tujuan untuk memudahkan pengelolaan dari dataset tersebut menggunakan basis data. Coursera sendiri merupakan platform belajar online yang didirikan pada tahun 2012 oleh dua profesor asal Universitas Stanford.

Dalam dataset tersebut terdiri dari nama Universitas, course atau nama program yang diambil, tipe pemrograman, review atau penilaian atau rating terhadap course tersebut, jumlah peserta, pemilihan voting dan tingkat kesulitan dari course tersebut. Selanjutnya dataset tersebut diolah menggunakan sistem basis data atau DBMS (*Database Management System*). Sistem basis data sendiri adalah sistem penyusunan dan pengelolaan rekaman-rekaman dengan menggunakan komputer, dengan tujuan untuk menyimpan atau merekam serta memelihara sata operasional lengkap dengan sebuah organisasi sehingga mampu menyajikan informasi yang diperlukan untuk pengambilan keputusan.

Identifikasi Masalah

Dari latar belakang diatas kita dapat merumuskan identifikasi masalah sebagai berikut :

1. Berapa course yang memiliki rating tertinggi dan terendah?
2. Berapa Universitas yang mengikuti course terbanyak?
3. Bagaimana tampilan baru setelah dikelola menggunakan sistem basis data?

Tujuan

Adapun tujuan akhir dari tugas akhir ini adalah :

1. Memahami penggunaan sistem basis data dalam pengelolaan dataset.
2. Mengetahui jenis course dengan rating tertinggi dan terendah.
3. Mengetahui nama Universitas dengan mengikuti course terbanyak.

2

Normalisasi pada Dataset

	university	course	type	review	votes	students	difficulty
0	University of Michigan	Python for Everybody	SPECIALIZATION	4.8	225236	2400000	Beginner
1	Google	Google IT Automation with Python	PROFESSIONAL CERTIFICATE	4.7	21768	410000	Beginner
2	University of Michigan	Python 3 Programming	SPECIALIZATION	4.7	15904	290000	Beginner
3	University of Michigan	Applied Data Science with Python	SPECIALIZATION	4.5	29358	730000	Intermediate
4	Università di Napoli Federico II	Data Science con Python e R	SPECIALIZATION				Intermediate
5	Google	Crash Course on Python	COURSE	4.8	17595	350000	Beginner
6	IBM	IBM Data Science	PROFESSIONAL CERTIFICATE	4.6	74747	830000	Beginner
7	University of Michigan	Programming for Everybody (Getting Started with Python)	COURSE	4.8	189604	2200000	Mixed
8	IBM	Data Science Fundamentals with Python and SQL	SPECIALIZATION	4.6	37454	450000	Beginner
9	IBM	IBM Data Analyst	PROFESSIONAL CERTIFICATE	4.6	36275	510000	Beginner
10	IBM	Python for Data Science, AI & Development	COURSE	4.6	21662	300000	Beginner
11	University of Pennsylvania	Introduction to Programming with Python and Java	SPECIALIZATION	4	179	11000	Beginner
12	Moscow Institute of Physics and Technology	Python	SPECIALIZATION	4.5	1695	71000	Intermediate
13	University of Michigan	Statistics with Python	SPECIALIZATION	4.6	2359	92000	Beginner
14	University of California, Irvine	IELTS Preparation	SPECIALIZATION	4.5	50	4000	Beginner
15	Rice University	Introduction to Scripting in Python	SPECIALIZATION	4.7	3460	100000	Beginner
16	University of California San Diego	Data Structures and Algorithms	SPECIALIZATION	4.6	13335	490000	Intermediate
17	University of Colorado Boulder	Data Structures and Algorithms	SPECIALIZATION	4.4	8	1600	Advanced
18	Pontificia Universidad Católica de Chile	Introducción a la programación en Python I: Aprendiendo	COURSE	4.4	2768	170000	Beginner
19	Codio	Programming in Python: A Hands-on Introduction	SPECIALIZATION	4.7	23	1600	Beginner
20	University of Michigan	Python Basics	COURSE	4.8	12882	250000	Beginner

Gambar 2.1 Data Awal

Relasi dengan data ganda menimbulkan banyak masalah. Selain boros ruang penyimpanan, ada tiga masalah utama yang muncul dan biasa disebut sebagai anomali update. Ketiga anomali update tersebut yaitu (1) anomali penambahan, (2) anomali penghapusan, dan (3) anomali modifikasi. Permasalahan anomali update ini dapat diatasi dengan normalisasi.

Anomali yang bisa terjadi:

Anomali merupakan proses pada basis data yang memberikan efek samping tidak diharapkan, anomali ini terjadi karena struktur tabel yang telah dibuat memiliki masalah. Masalah ini akan meningkatkan kompleksitas data dan menyulitkan dalam pemodifikasian data itu sendiri. Sehingga harus dilakukan cara-cara yang disebut normalisasi. Terdapat 3 macam anomali. Yaitu :

1. Anomali penambahan

Anomali penambahan dapat terjadi ketika ada data baru yang ditambahkan, tetapi ternyata terdapat elemen yang kosong dan elemen tersebut justru menjadi key.

2. Anomali penghapusan

Anomali penghapusan terjadi ketika suatu baris (tuple) yang tidak terpakai dihapus, kemudian sebagai akibatnya terdapat data lain yang hilang.

3. Anomali Modifikasi

Anomali modifikasi terjadi ketika ada sejumlah data yang diubah, tetapi tidak seluruhnya mengalami perubahan. Anomali modifikasi dapat menyebabkan ketidakkonsistenan pada data.

Proses Normalisasi

Normalisasi memiliki tujuan utama untuk mengelompokkan atribut ke dalam relasi sehingga data ganda menjadi minimal. Proses ini akan membagi tabel besar menjadi tabel yang lebih kecil dengan redundansi yang lebih sedikit. Tabel yang lebih kecil nantinya akan saling terkait satu sama lain melalui hubungan yang didefinisikan dengan baik.

1. 1NF (First Normal Form 'Bentuk Normal Pertama')

1NF merupakan bentuk normal pertama yang menyediakan sekumpulan persyaratan minimum untuk menormalkan database relasional. Suatu tabel dapat dikatakan 1NF jika dan hanya jika setiap atribut dari data tersebut hanya memiliki nilai tunggal dalam satu baris. Dalam 1NF, tidak diperbolehkan adanya *grouping* data atau duplikasi data.

Data yang digunakan telah memenuhi syarat 1NF karena data tersebut hanya memiliki nilai tunggal dalam satu baris.

2. 2NF (Second Normal Form 'Bentuk Normal Kedua')

2NF adalah bentuk normal kedua yang digunakan dalam database relasional. Agar tabel dapat memenuhi 2NF maka tabel tersebut harus dipatuhi dengan 1NF dan atribut apapun yang bukan merupakan bagian dari atribut e-prime harus sepenuhnya bergantung pada salah satu kandidat dalam tabel. Syarat 2NF adalah tidak diperkenankan adanya *partial functional dependency* kepada primary key dalam sebuah tabel.

DetailCourse							
id	course	university_id	type_id	difficulty_id	review	votes	students
1	Python for Everybody	U1	T1	D1	4.8	225236	2400000
2	Google IT Automation with Python	U2	T2	D1	4.7	21768	410000
3	Python 3 Programming	U1	T1	D1	4.7	15904	290000
4	Applied Data Science with Python	U1	T1	D2	4.5	29358	730000
5	Data Science con Python e R	U3	T1	D2			
6	Crash Course on Python	U2	T3	D1	4.8	17595	350000

University		DifficultyCourse	
university_id	University	difficulty_id	Difficulty Name
U1	University of Michigan	D1	Beginner
U2	Google	D2	Intermediate
U3	Università di Napoli Federico II	D3	Advanced
U4	IBM	D4	Mixed

Type Course	
type_id	Type Name
T1	SPECIALIZATION
T2	PROFESSIONAL CERTIFICATE
T3	COURSE
T4	GUIDED PROJECT
T5	DEGREE
T6	MASTER TRACK CERTIFICATE

Gambar 2. Sudah di normalisasi 2NF

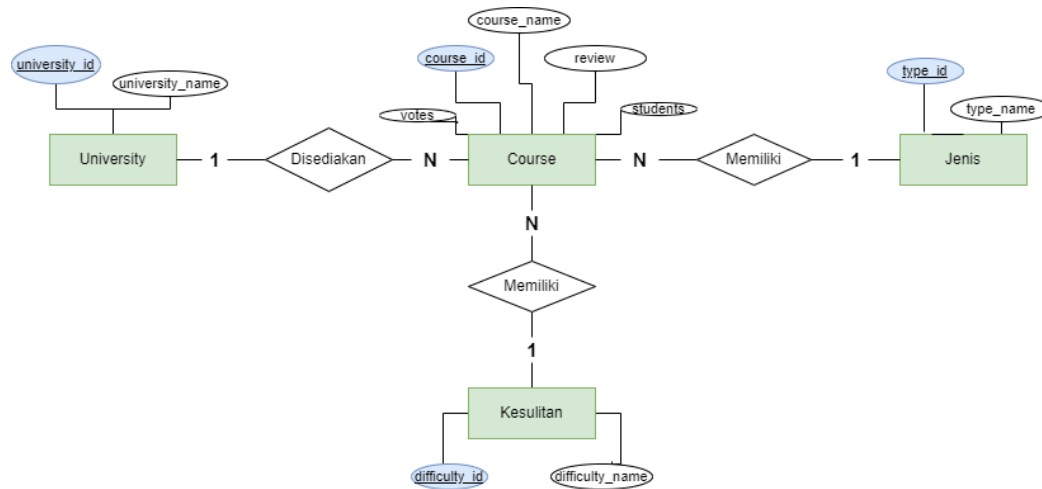
3

Buat Relational Schema nya

Relation schema (skema relasi) adalah nama relasi yang diikuti dengan kumpulan atribut-atributnya. Skema relasi merupakan sebuah cara yang dapat digunakan untuk merepresentasikan hubungan antara satu tabel dengan tabel lain melalui sebuah kolom kunci. Pada skema relasi, sebuah *primary key* suatu tabel merupakan *foreign key* pada tabel lainnya. Nantinya, kunci tersebut dapat digunakan untuk membantu kita menggabungkan informasi dari tabel-tabel yang terpisah.

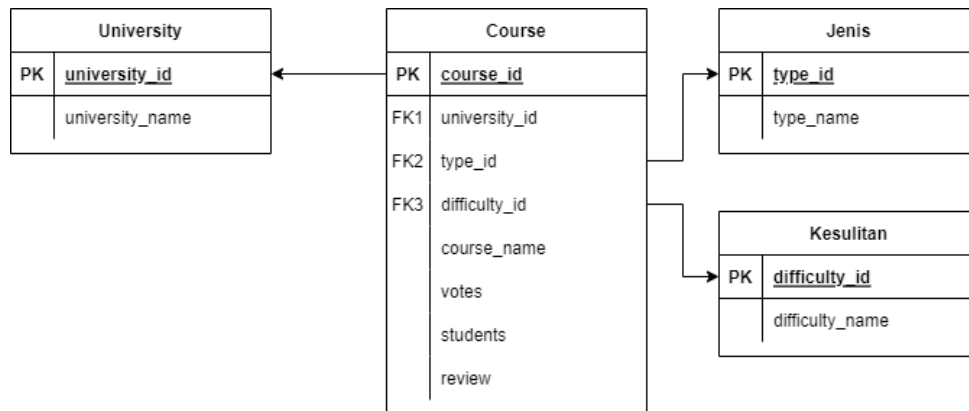
Kolom kunci merupakan suatu kolom yang berisi nilai-nilai yang menunjukkan sebuah observasi. Pada database relasional, terdapat dua jenis kolom kunci. Antara lain:

1. **Primary Key:** Primary key membentuk keunikan, yaitu hanya satu primary key yang diperbolehkan untuk setiap tabel. Primary key merupakan not null, atau dapat dikatakan juga sebagai constraint unique yang memiliki nilai not null.
2. **Foreign Key:** Digunakan sebagai hubungan antar primary atau unique ke yang lain. Foreign key bertujuan untuk menetapkan suatu kolom atau kombinasi dari beberapa kolom menjadi foreign key dari sebuah tabel. Foreign key disebut juga referential integrity constraint.



Gambar 3.1 ERD

- University mempunyai atribut : university_id dan university_name
- Course memiliki primary key course_id dengan atribut votes, review, course_name, dan student
- Jenis mempunyai primary key type_id dengan atribut type_name
- Kesulitan mempunyai primary key difficulty_id dengan atribut difficulty_name



Gambar 3.2 Relational Schema

Deskripsi permasalahan pada Gambar 3.2:

1. Course disediakan oleh university dan memiliki atribut: course_id (primary key), university_id (Foreign key 1), type_id (Foreign Key 2), difficulty_id (Foreign Key 3), course_name, votes, students, dan review.
2. Course memiliki jenis : course type_id berhubungan dengan jenis type_id (primary key) dan memiliki atribut yaitu type_name
3. Course memiliki tingkat kesulitan : difficulty_id (primary key), difficulty_name.

4

Implementasi syntax - DDL (*Data Definition Language*)

DDL merupakan singkatan dari *Data Definition Language*, yaitu kumpulan perintah pada *Structured Query Language* (SQL) untuk menggambarkan desain dari database secara keseluruhan. Selain itu *Data Definition Language* (DDL) digunakan untuk membuat, menghapus, maupun merubah struktur definisi tipe data dari objek pada database. Dalam DDL terdapat perintah, yaitu CREATE, ALTER, DROP, RENAME, dan SHOW yang akan dieksekusi ketika diakhiri dengan ‘;’ (titik koma). Penulis disini menggunakan perintah CREATE.

Perintah ini dapat digunakan ketika membuat, termasuk dalam membuat database baru atau tabel baru. Penulis disini menggunakan sintaks ini untuk membuat database baru dan tabel baru.

1. CREATE DATABASE

Create database disini berguna untuk membuat database baru yaitu *tubes_kel7_final*.

```
CREATE DATABASE IF NOT EXISTS tubes_kel7_final;
```

Setelah create database, database tersebut tidak dapat langsung digunakan. Untuk mengakses database yang telah dibuat menggunakan perintah

```
USE tubes_kel7_final;
```

2. CREATE TABLE

Create table disini berguna untuk membuat tabel baru pada database yang dibuat yaitu table, *university*, *jenis*, *kesulitan* dan *course*.

a. CREATE TABLE university

```
4 # =====
5 # TABEL UNIVERSITY
6 • CREATE TABLE IF NOT EXISTS university(
7     university_id VARCHAR(5) NOT NULL PRIMARY KEY,
8     university_name VARCHAR(255)
9 );
10 • DESCRIBE university;
11
12 # =====
```

Gambar 4.1 Sintaks untuk membuat tabel university

Dalam tabel terdapat CONSTRAINT untuk mendefinisikan, mengizinkan atau membatasi nilai apa yang dapat disimpan dalam kolom. Pada table university ini terdapat university_id dengan CONSTRAINT yang digunakan yaitu PRIMARY KEY.

```

12      # =====
13      # TABEL JENIS
14      ● ○ CREATE TABLE IF NOT EXISTS jenis(
15          |     type_id VARCHAR(5) NOT NULL PRIMARY KEY,
16          |     type_name VARCHAR(255)
17          | );
18      ● DESCRIBE jenis;
19
20      # =====

```

b. CREATE TABLE jenis

Gambar 4.2 Sintaks untuk membuat tabel jenis

Pada tabel jenis ini terdapat type_id dengan CONSTRAINT yang digunakan yaitu PRIMARY KEY.

c. CREATE TABLE kesulitan

```

20      # =====
21      # TABEL KESULITAN
22      ● ○ CREATE TABLE IF NOT EXISTS kesulitan(
23          |     difficulty_id VARCHAR(5) NOT NULL PRIMARY KEY,
24          |     difficulty_name VARCHAR(255)
25          | );
26      ● DESCRIBE kesulitan;
27
28      # =====

```

Gambar 4.3 Sintaks untuk membuat tabel kesulitan

Pada table kesulitan ini terdapat difficulty_id dengan CONSTRAINT yang digunakan yaitu PRIMARY KEY.

d. CREATE TABLE course

```
28      # =====
29      # TABLE COURSE
30      CREATE TABLE IF NOT EXISTS course(
31          course_id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
32          course_name VARCHAR(255),
33          university_id VARCHAR(5),
34          type_id VARCHAR(5),
35          difficulty_id VARCHAR(5),
36          votes INT,
37          students INT,
38          review FLOAT,
39          FOREIGN KEY (university_id) REFERENCES university (university_id),
40          FOREIGN KEY (type_id) REFERENCES jenis (type_id),
41          FOREIGN KEY (difficulty_id) REFERENCES kesulitan (difficulty_id)
42      );
43      DESCRIBE course;
44
45      # =====
```

Gambar 4.4 Sintaks untuk membuat tabel course

Pada table course ini terdapat difficulty_id dengan CONSTRAINT yang digunakan yaitu PRIMARY KEY. FOREIGN KEY terdapat pada university_id, type_id, dan difficulty_id.

5

Implementasi Syntax - Data Manipulation Language

DML atau Data Manipulation Language merupakan perintah SQL yang berhubungan dengan manipulasi atau pengolahan data dalam tabel. Ketika struktur sebuah database sudah terbentuk menggunakan perintah DDL atau Data Definition Language, langkah selanjutnya adalah mengolah data dalam tabel tersebut.

1. INSERT

Perintah ini untuk memasukkan sebuah record baru di dalam sebuah tabel database. Penulis menggunakan sintaks ini untuk memasukkan data ke dalam tabel *university*, tabel *jenis*, tabel *kesulitan*, dan tabel *course*.

```
45 | =====
46 | # DML ke table university
47 | • INSERT INTO university (university_id, university_name)
48 |   VALUES ("U1", "University of Michigan"), ("U2", "Google"), ("U3", "IBM"),
49 |     ("U4", "University of Pennsylvania"), ("U5", "Moscow Institute of Physics and Technology"), ("U6", "University of California, Irvine"),
50 |     ("U7", "Rice University"), ("U8", "University of California San Diego"), ("U9", "University of Colorado Boulder"),
51 |     ("U10", "Codio"), ("U11", "The Hong Kong University of Science and Technology"), ("U12", "DeepLearning.AI"),
52 |     ("U13", "Coursera Project Network"), ("U14", "Infosec"), ("U15", "HSE University"),
53 |     ("U16", "Wesleyan University"), ("U17", "Universidad Austral"), ("U18", "EDHEC Business School"), ("U19", "Universidad de los Andes"),
54 |     ("U20", "University of Toronto"), ("U21", "Google Cloud"), ("U22", "Amazon Web Services"),
55 |     ("U23", "University of London"), ("U24", "University of Alberta"), ("U25", "LearnQuest"),
56 |     ("U26", "National Taiwan University"), ("U27", "Johns Hopkins University"), ("U28", "Nanjing University"),
57 |     ("U29", "The University of Chicago"), ("U30", "University of Illinois at Urbana-Champaign")
58 | ;
59 | • SELECT * FROM university;
```

Gambar 5.1 Sintaks untuk memasukkan data ke tabel university

Sintaks Insert penulis gunakan untuk tabel yang lainnya, seperti tabel *jenis*, tabel *kesulitan*, dan tabel *course*.

2. SELECT

Select adalah sintaks yang digunakan untuk mengambil data dari objek database seperti tabel. Untuk mengambil semua baris data dapat menggunakan `*`. Jika data yang diambil spesifik maka dapat menggunakan nama kolom. Nama kolom yang dipilih dapat berjumlah lebih dari satu. Untuk mendapatkan data yang lebih spesifik, dapat menggunakan `where` untuk mendefinisikan kondisi.

Pada tugas akhir ini, penulis menggunakan sintaks *Select* untuk memeriksa apakah data yang dimasukkan saat melakukan *Insert* sudah benar atau belum. Ketika sintaks pada Gambar 5.1 dijalankan, akan mengisi kolom *university_id*, dan *university_name*. Hasilnya terlihat seperti pada Gambar 5.2.

university_id	university_name
U1	University of Michigan
U10	Codio
U11	The Hong Kong University of Science and Techn...
U12	DeepLearning.AI
U13	Coursera Project Network
U14	Infosec
U15	HSE University
U16	Wesleyan University
U17	Universidad Austral
U18	EDHEC Business School
U19	Universidad de los Andes
U2	Google
U20	University of Toronto
U21	Google Cloud
U22	Amazon Web Services
U23	University of London
U24	University of Alberta

Gambar 5.2 Syntax Select

3. UPDATE

Update adalah sintaks yang memungkinkan untuk memperbarui data tabel yang ada. Untuk mengubah baris data yang lebih spesifik dapat menggunakan *where* untuk mendefinisikan kondisi. Nama kolom yang akan diubah dapat berjumlah lebih dari satu. Misalnya, penulis ingin mengubah *university_name* menggunakan *university_id = U1*. Maka, yang dilakukan terlihat seperti pada Gambar 5.3. Telihat bahwa *university_name* pada *university_id = U1* sudah berubah menjadi Institut Teknologi Sumatera.

```

145 # implementasi DML - Update
146 • UPDATE university
147 SET university_name = "Institut Teknologi Sumatera"
148 WHERE university_id = "U1";

```

university_id	university_name
U1	Institut Teknologi Sumatera
U10	Codio
U11	The Hong Kong University of Science and Techn...
U12	DeepLearning.AI
U13	Coursera Project Network
U14	Infosec
U15	HSE University
U16	Wesleyan University
U17	Universidad Austral
U18	EDHEC Business School
U19	Universidad de los Andes
U2	Google
U20	University of Toronto
U21	Google Cloud
U22	Amazon Web Services

Gambar 5.3 Sintaks Update

Terlihat pada Gambar 5.3 bahwa *university_name* pada *university_id = U1* sudah berubah menjadi Institut Teknologi Sumatera.

4. DELETE

Delete adalah sintaks yang dapat menghapus baris data. Sintaks delete fungsinya mirip dengan sintaks truncate. Perbedaan sintaks delete adalah dapat menentukan dengan tepat apa yang ingin dihapus, sedangkan truncate memungkinkan untuk menghapus semua record yang terdapat dalam sebuah tabel. Untuk menghapus baris data yang lebih spesifik dapat menggunakan where untuk mendefinisikan kondisi.

Karena data pada tabel terlihat baik-baik saja, maka penulis tidak menggunakan sintaks ini. Namun, jika ingin menggunakan sintaks ini, caranya sebagai berikut:

```
mysql> TRUNCATE TABLE nama_tabel ;  
atau  
mysql> DELETE FROM nama_tabel ;  
atau  
mysql> DELETE FROM nama_tabel  
WHERE nama_kolom = kondisi_nilai_dicari  
;
```

6

Advanced Query - Insight

DML atau Data Manipulation Language merupakan perintah SQL yang berhubungan dengan manipulasi atau pengolahan data dalam tabel. Ketika struktur sebuah database sudah terbentuk menggunakan perintah DDL atau Da

1. Mendapatkan 5 course dengan rating terendah.

Untuk mendapatkan course dengan rating terendah, diperlukan kolom *review* pada tabel *course*. Jika hanya mengambil kolom *review*, maka akan kurang berguna. Oleh karena itu, penulis menambahkan kolom *course_name* dari tabel *course*, dan *university_name* dari tabel *university*. Selanjutnya, karena berasal dari tabel yang berbeda, maka diperlukan *join*. Setelah dilakukan *join*, dilakukan pengurutan menggunakan *order by*. Hasil yang didapatkan terlihat pada Gambar 6.1 yaitu Course *Introduction to Python Programming* oleh *University of Pennsylvlavana* dengan rating 3.9 menempati peringkat pertama.

```
3 # Mendapatkan 5 Course dengan Rating terendah
4 • SELECT course.course_name as NamaCourses,
5     university.university_name as Universitas,
6     course.review AS Rating
7 FROM course
8 INNER JOIN university ON course.university_id=university.university_id
9 ORDER BY course.review ASC
10 LIMIT 5;
```

NamaCourses	Universitas	Rating
Introduction to Python Programming	University of Pennsylvania	3.9
Introduction to Programming with Python and J...	University of Pennsylvania	4
Advanced Data Science with IBM	Moscow Institute of Physics and Technology	4.3
Python Project for Data Science	IBM	4.3
Data Structures and Algorithms	University of Colorado Boulder	4.4

Gambar 6.1 Lima Course dengan Rating Terendah

2. Mendapatkan 5 course dengan rating terbaik.

Cara yang dilakukan sama seperti sebelumnya. Hanya saja terdapat perbedaan pada pengurutan menggunakan *Order By*. Kali ini digunakan *DESC*, sehingga hasil yang didapatkan menurun. Hasilnya terlihat pada gambar 6.2.

NamaCourses	Universitas	Rating
Python Data Structures	Institut Teknologi Sumatera	4.9
Introdução à Ciência da Computação com Pytho...	Institut Teknologi Sumatera	4.9
Programación en Python	Universidad de los Andes	4.9
Python for Everybody	Institut Teknologi Sumatera	4.8
Deep Learning	DeepLearning.AI	4.8

Gambar 6.2 Lima Course dengan Rating Terbaik

- Mendapatkan rata-rata rating dari setiap universitas pembuat
Rata-rata rating dari setiap universitas yang mengeluarkan course sangat menarik untuk dilihat. Namun, jika yang dilihat hanya rating saja, data yang didapatkan akan kurang bagus. Penulis disini mengikutsertakan jumlah course dari setiap universitas. Caranya menggunakan sintaks di bawah ini:

```
# Mendapatkan rata-rata rating dari setiap universitas
SELECT university.university_name as Universitas,
       COUNT(course.review) AS JumlahCourse,
       AVG(course.review) AS RatingAVG
FROM course
INNER JOIN university ON course.university_id=university.university_id
GROUP BY course.university_id
ORDER BY course.review DESC;
```

Ketika sintaks di atas dijalankan, hasilnya akan terlihat seperti pada Gambar 6.3. Hal yang menarik disini, adalah Institut Teknologi Sumatera meskipun memiliki banyak course yang banyak, namun rata-rata ratingnya tetap tinggi.

	Universitas	JumlahCourse	RatingAVG
►	Universidad de los Andes	1	4.900000095367432
	Institut Teknologi Sumatera	13	4.730769267449012
	DeepLearning.AI	1	4.800000190734863
	Google	3	4.733333269755046
	Rice University	2	4.75
	Codio	1	4.699999809265137
	Wesleyan University	1	4.699999809265137
	University of Toronto	1	4.699999809265137
	University of London	1	4.699999809265137
	IBM	5	4.579999923706055
	University of Pennsylvania	5	4.339999961853027
	Moscow Institute of Physic...	4	4.524999976158142
	University of California Sa...	1	4.599999904632568
	Infosec	1	4.599999904632568
	EDHEC Business School	1	4.599999904632568
	University of California, Ir...	1	4.5
	Coursera Project Network	4	4.549999952316284
	Universidad Austral	1	4.5
	Google Cloud	1	4.5
	Amazon Web Services	1	4.5
	University of Colorado Bou...	1	4.400000095367432
	The Hong Kong University ...	1	4.400000095367432

Gambar 6.3 Rata-rata rating berdasarkan Universitas Pembuat

- Mendapatkan Universitas dengan Jumlah Course Lebih dari Tiga
Dari insight nomor tiga, sangat menarik untuk mengetahui universitas apa yang memiliki jumlah course lebih dari tiga. Hal itu dapat dilakukan dengan query di bawah.

```
# Mendapatkan universitas dengan jumlah course lebih dari tiga
SELECT university.university_name as Universitas,
       COUNT(course.course_id) AS JumlahCourse
FROM course
INNER JOIN university ON course.university_id=university.university_id
GROUP BY course.university_id
HAVING COUNT(course.course_id) > 3;
```

Setelah query di atas dijalankan, didapatkan hasil seperti pada Gambar 6.4. Hal yang menarik, dari 30 universitas, ternyata hanya 5 universitas yang memiliki course lebih dari tiga.

	Universitas	JumlahCourse
►	Institut Teknologi Sumatera	13
	Coursera Project Network	4
	IBM	5
	University of Pennsylvania	5
	Moscow Institute of Physics and Technology	4

Gambar 6.4 Universitas yang Memiliki Course Lebih dari Tiga

5. Menggabungkan seluruh tabel yang ada, dan hanya menampilkan kolom yang penting
 Karena penulis memiliki *super computer*, penulis tertarik untuk menggabungkan seluruh tabel yang ada menggunakan *join*. Query yang digunakan sebagai berikut:

```
# Menggabungkan seluruh tabel, tapi hanya menampilkan kolom yang penting
SELECT course_id, course_name, university_name, type_name, difficulty_name, votes, students, review
FROM course
INNER JOIN university ON course.university_id=university.university_id
INNER JOIN jenis ON course.type_id=jenis.type_id
INNER JOIN kesulitan ON course.difficulty_id=kesulitan.difficulty_id;
```

Ketika Query di atas berhasil dijalankan, akan menghasilkan output seperti pada gambar 6.5. Terlihat perubahan yang menarik dari tabel. Pada awalnya, tabel hanya terlihat foreign key saja. Namun, ketika dilakukan join, akan terlihat bagaimana data sebenarnya.

	course_id	course_name	university_name	type_name	difficulty_name	votes	students	review
►	1	Python for Everybody	Institut Teknologi Sumatera	SPECIALIZATION	Beginner	225236	2400000	4.8
	2	Google IT Automation with Python	Google	PROFESSIONAL CERTIFICATE	Beginner	21768	410000	4.7
	3	Python 3 Programming	Institut Teknologi Sumatera	SPECIALIZATION	Beginner	15904	290000	4.7
	5	Crash Course on Python	Google	COURSE	Beginner	17595	350000	4.8
	6	IBM Data Science	IBM	PROFESSIONAL CERTIFICATE	Beginner	74747	830000	4.6
	8	Data Science Fundamentals with Python and SQL	IBM	SPECIALIZATION	Beginner	37454	450000	4.6
	9	IBM Data Analyst	University of Pennsylvania	PROFESSIONAL CERTIFICATE	Beginner	36275	510000	4.6
	10	Python for Data Science, AI & Development	Moscow Institute of Physics and Technology	COURSE	Beginner	21662	300000	4.6
	11	Introduction to Programming with Python and J...	University of Pennsylvania	SPECIALIZATION	Beginner	179	11000	4
	12	Statistics with Python	Institut Teknologi Sumatera	SPECIALIZATION	Beginner	2359	92000	4.6
	13	IELTS Preparation	University of California, Irvine	SPECIALIZATION	Beginner	50	4000	4.5
	14	Introduction to Scripting in Python	Rice University	SPECIALIZATION	Beginner	3460	100000	4.7
	17	Programming in Python: A Hands-on Introduction	Codio	SPECIALIZATION	Beginner	23	1600	4.7
	18	Python Basics	Institut Teknologi Sumatera	COURSE	Beginner	12882	250000	4.8
	21	Introduction to Python	Coursera Project Network	GUIDED PROJECT	Beginner	1634	26000	4.5
	25	IBM Full Stack Cloud Developer	Moscow Institute of Physics and Technology	PROFESSIONAL CERTIFICATE	Beginner	22680	340000	4.6
	26	Python Programming: A Concise Introduction	Wesleyan University	COURSE	Beginner	4265	160000	4.7
	27	Aprende a programar con Python	Universidad Austral	SPECIALIZATION	Beginner	1714	43000	4.5
	29	Introduction to Python Programming	University of Pennsylvania	COURSE	Beginner	129	8100	3.9
	30	Fundamentals of Computer Science	Rice University	SPECIALIZATION	Beginner	3660	100000	4.8

Gambar 6.5 Join Seluruh Tabel