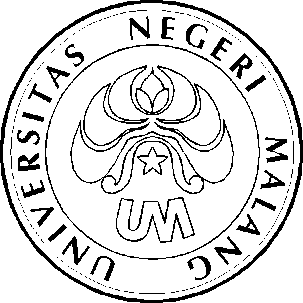
**LAPORAN MODUL V**

**PRAKTIKUM BASIS DATA**

**“RELASI DAN JOIN”**

Disusun untuk Memenuhi Matakuliah Praktikum Basis Data

Dibimbing oleh Triyanna Widyaningtyas, S.T., M.T.

****

Oleh:

Nama NIM

Nama NIM

S1 TI ‘17 OFF A

**UNIVERSITAS NEGERI MALANG**

**FAKULTAS TEKNIK**

**JURUSAN TEKNIK ELEKTRO**

**PRODI S1 TEKNIK INFORMATIKA**

**Maret 2018**

1. **Tujuan Praktikum**

* Memahami keterhubungan entitas di dalam basis data.
* Memahami jenis – jenis operasi pengambilan data di beberapa entitas
* Mampu menyelesaikan kasus retrieval yang melibatkan lebih dari satu entitas
* Memahami fungsi union pada mysql

1. **Dasar Teori**
2. Relationship

Relationship adalah suatu hubungan antara beberapa entitas. Konsep ini sangat penting sekali di dalam basis data, di mana memungkinkan entitas-entitas untuk saling berhubungan satu sama lain.

Didalam sebuah relationship, primary key memiliki peran penting untuk mengaitkan entitas. Selain itu, primary key juga digunakan untuk mendefinisikan Batasan keterhubungan.

1. Join

Join merupakan salah satu kontruksi dasar dari SQL dan basis data. Join dapat didefinisikan sebagai kombinasi record dari dua atau lebih table di dalam basis data relasional dan menghasilkan sebuah tabel (temporary) baru yang disebut sebagai joined tabel.

Join dapat diklasifikasikan ke dalam dua jenis, yaitu inner join dan outer join.

1. Inner Join

Inner join pada dasarnya adalah menemukan persimpangan (intersection) antara dua buah tabel.

Sintaks inner join diperlihatkan sebagai berikut:



Inner join juga dapat direpresentasikan dalam bentuk implisit sebagai berikut:



Misalkan terdapat tabel A dan B, maka hasil inner join dapat diperlihatkan sebagai bidang terarsir dalam diaram venn seperti gambar 1.



*Gambar 1. Inner Join*

1. Outer Join

Outer join dibagi ke dalam tiga jenis, yaitu left outer join, right outer join dan full outer join.

**Left Outer Join**

Left outer join atau left join mengembalikan semua nilai dari table kiri ditambah dengan nilai dari tabel kanan yang sesuai atau NULL jika tidak ada nilai yang sesuai.

Sintaks left outer join diperlihatkan sebagai berikut:



Left outer join antar tabel A dan B dapat diilustrasikan dalam diagram venn seperti gambar 2.



*Gambar 2. Left outer join*

**Right Outer Join**

Right outer join atau right join pada dasarnya sama seperti left join, namun dalam bentuk terbalik, kanan dan kiri.

Sintaks right outer join diperlihatkan sebagai berikut :



Right outer join antara table A dan B dapat diilustrasikan dalam diagram Venn seperti Gambar 3.



*Gambar 3. Right outer join*

**Full Outer Join**

Full outer join atau full join pada hakikatnya merupakan kombinasi dari left dan right join.

Sintaks full outer join diperlihatkan sebagai berikut :



Bentuk visual dari full outer join dapat diperlihatkan menggunakan diagram Venn seperti Gambar 4.



*Gambar 4. Full outer join*

Selain empat jenis join yang utama di atas, masih ada beberapa variasi join lainnya, seperti CROSS JOIN (cartesian product), NATURAL JOIN dan sebagainya.

Perlu juga diperhatikan, join bisa diimplementasikan dalam bentuk bersarang (nested join). Jadi, di dalam sebuah operasi join bisa terdapat operasi join lainnya.

1. Union

MySQL Union adalah statemaen yang mengkombinasikan dua buah atau lebih resulset dari beberapa table dengan statemen SELECT sehingga menjadi satu buah resulset.

Union Statemen memiliki beberapa ketentuan sebagai berikut :

1. Jumlah kolom/field dari setiap statement SELECT harus sama
2. Tipe data kolom/field dari setiap statement SELECT harus kompatibel

Secara default statemen UNION akan menghapus semua record duplikat dari resulset. Apabila Anda ingin record duplikat tetap di tampilkan maka pada resulset tuliskan secara explisit UNION ALL. Perbedaan Union dan Union All dapat dijelaskan pada gambar diagram Venn 5 dan 6.



Gambar 5. Union



Gambar 6. Union all

Fungsi Union sendiri dapat dijalankan dengan sintaks sebagai berikut :



1. **Tugas Latihan** (*Sertakan hasil screenshoot langkah-langkah*)
2. *…*
3. *…*
4. *…*
5. **Tugas Praktikum**
6. *Source code*
7. *Screenshoot program*
8. *Penjelasan / Analisis*
9. **Tugas Rumah**
10. Buatlah database baru dengan nama Universitas. Lalu didalamnya terdapat tabel-tabel berikut :

Tabel instruktur



Tabel matakuliah

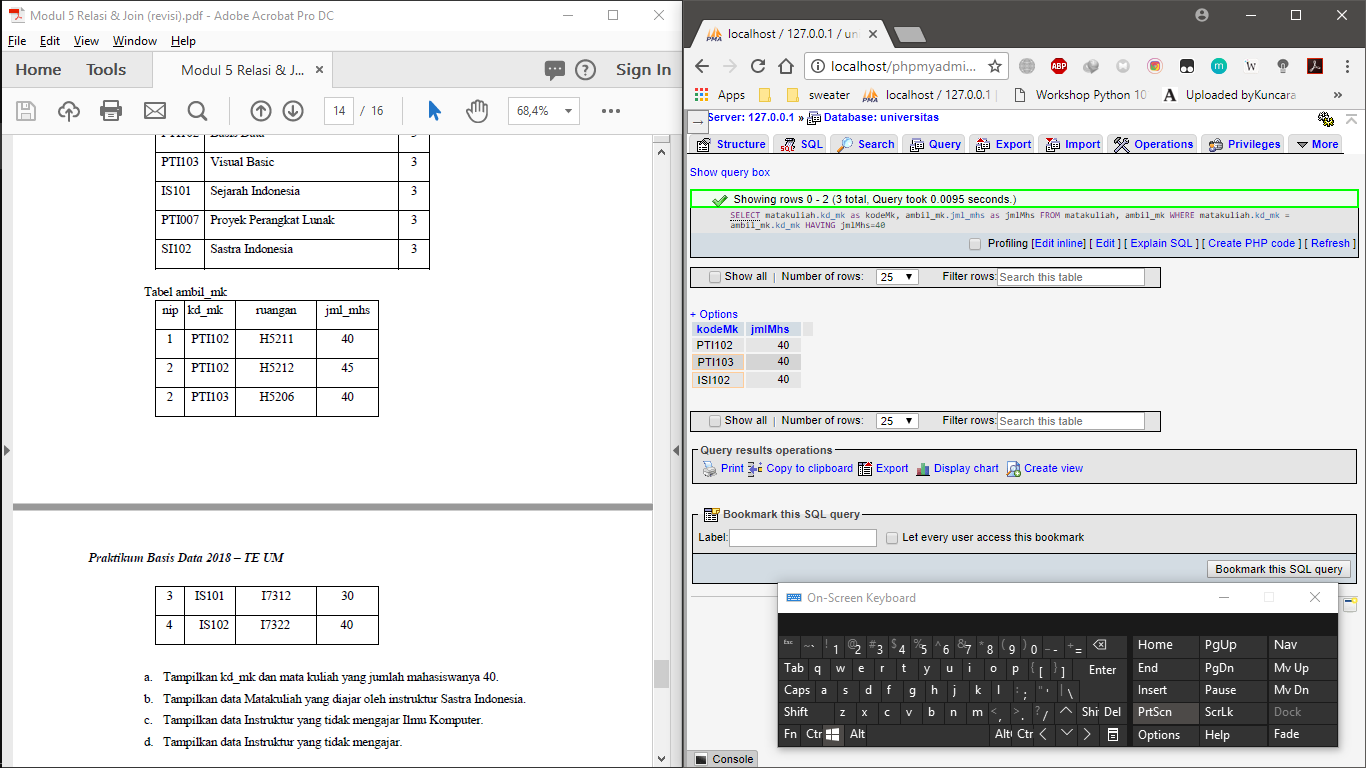


Tabel ambil\_mk

|  |  |  |  |
| --- | --- | --- | --- |
| nip | kd\_mk | ruangan | jml\_mhs |
| 1 | PTI102 | H5211 | 40 |
| 2 | PTI102 | H5212 | 45 |
| 2 | PTI103 | H5206 | 40 |
| 3 | ISI101 | I7312 | 30 |
| 4 | ISI102 | I7322 | 40 |

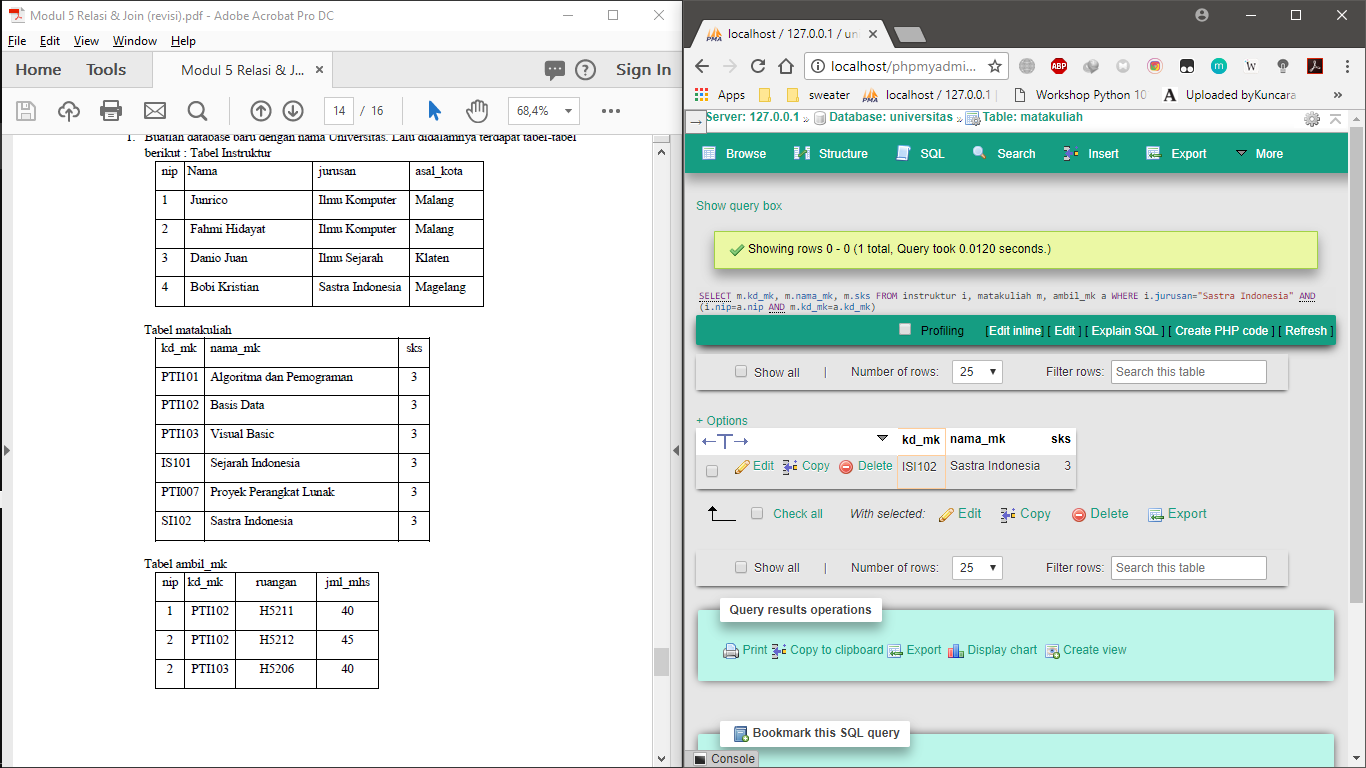
1. Tampilkan kd\_mk dan mata kuliah yang jumlah mahasiswanya 40.





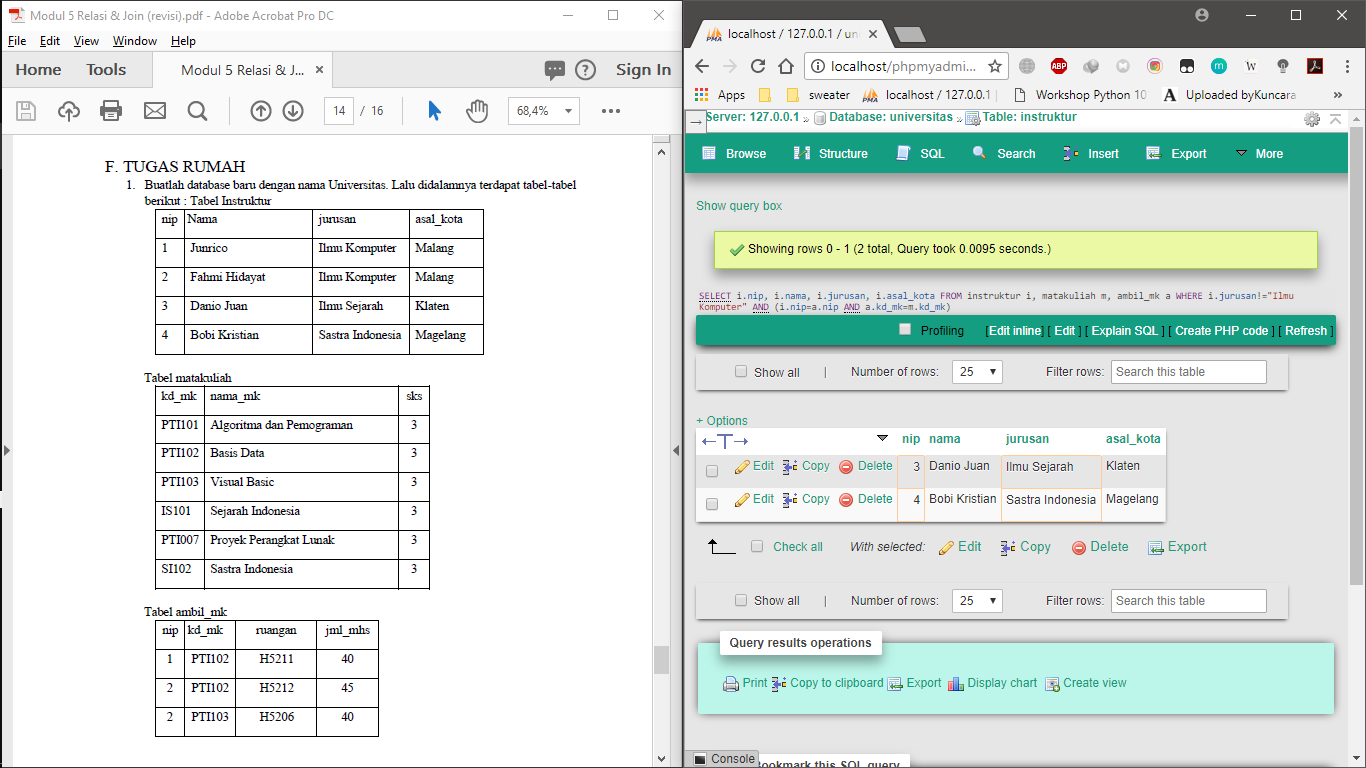
1. Tampilkan data Matakuliah yang diajar oleh instruktur Sastra Indonesia.





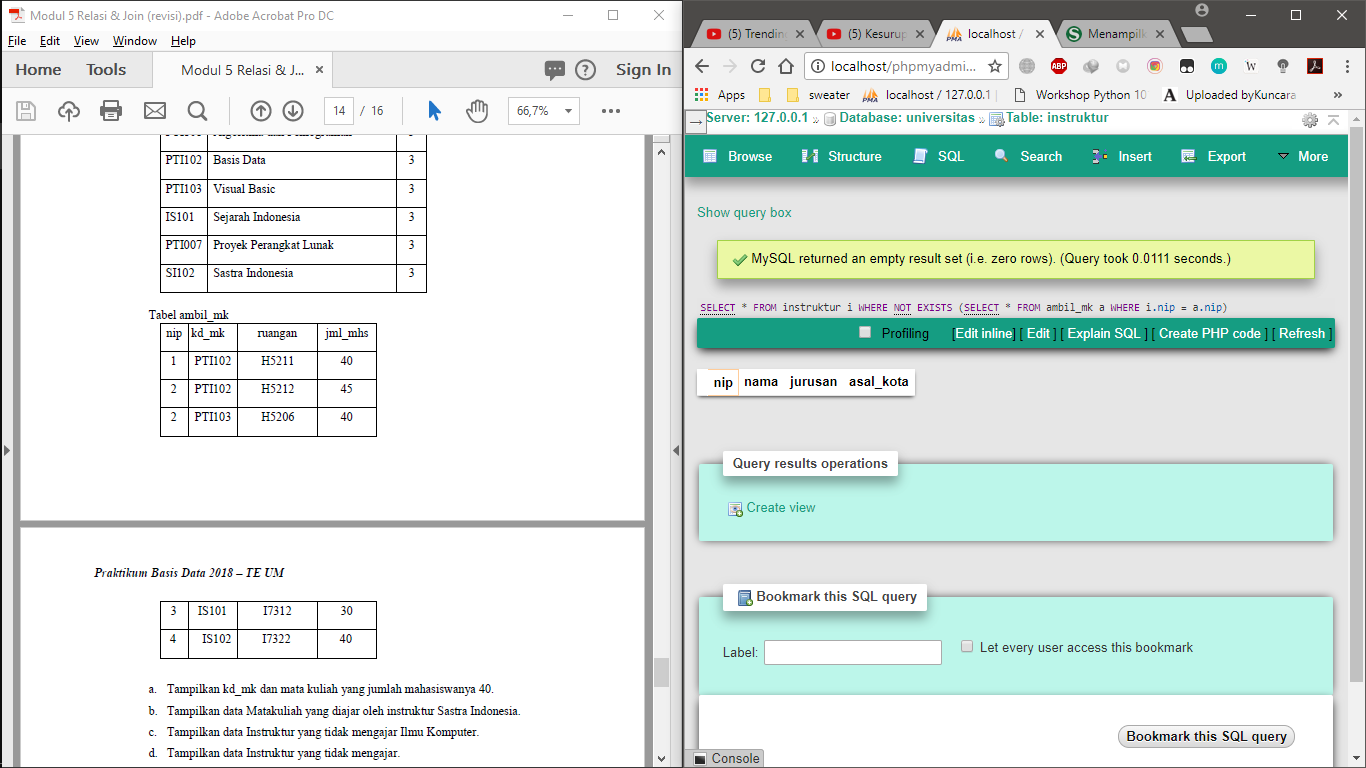
1. Tampilkan data Instruktur yang tidak mengajar Ilmu Komputer.





1. Tampilkan data instruktur yang tidak mengajar.





Syntax akan menampilkan data pada tabel instruktur dengan menggunakan alias i “SELECT \* FROM instruktur i”, dengan kondisi pada tabel instruktur yang tidak ada pada tabel ambil\_mk “WHERE NOT EXISTS” dengan kunci berupa nip “WHERE i.nip=a.nip”, tidak ada yang ditampilkan karena seluruh kunci nip pada tabel instruktur ada pada tabel ambil\_mk, yang berarti tidak ada instruktur yang tidak mengajar.

1. Buatlah tabel seperti di bawah ini.

Tabel customer

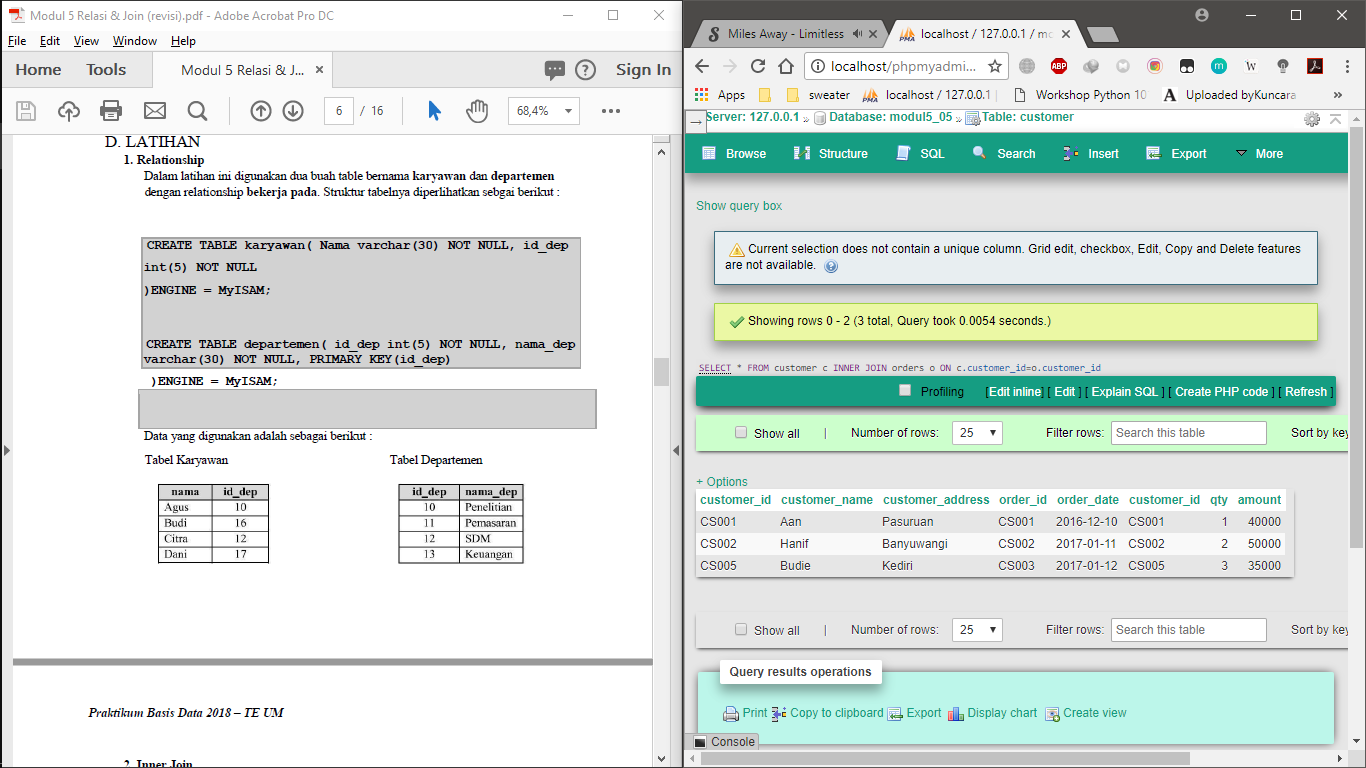
|  |  |  |
| --- | --- | --- |
| customer\_id | customer\_name | customer\_addres |
| CS001 | Aan | Pasuruan |
| CS002 | Hanif | Banyuwangi |
| CS003 | Mirza | Malang |
| CS004 | Tanti | Tegal |
| CS005 | Budie | Kediri |

Tabel orders

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| order\_id | order\_date | customer\_id | qty | amount |
| CS001 | 10-12-2016 | CS001 | 1 | 40000 |
| CS002 | 11-01-2017 | CS002 | 2 | 50000 |
| CS003 | 12-01-2017 | CS005 | 3 | 35000 |

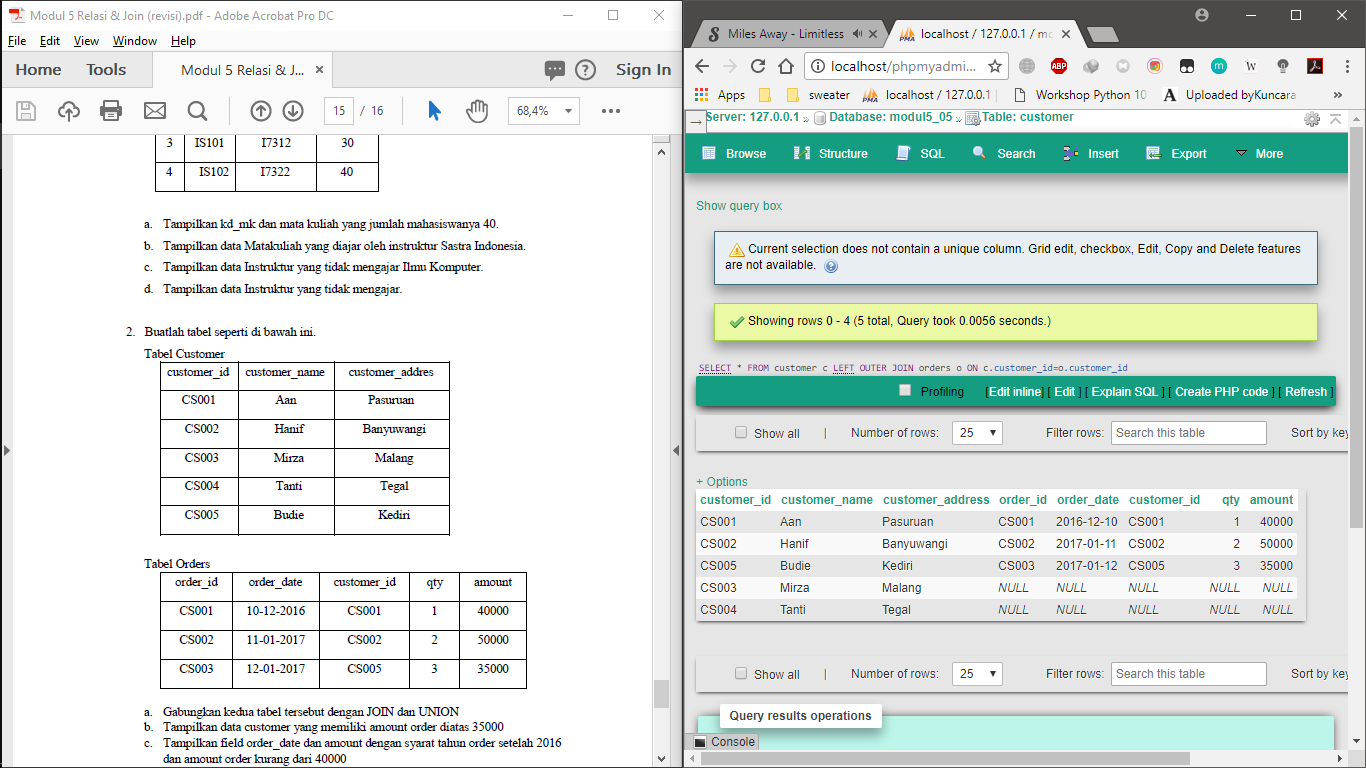
1. Gabungkan kedua tabel tersebut dengan JOIN dan UNION.





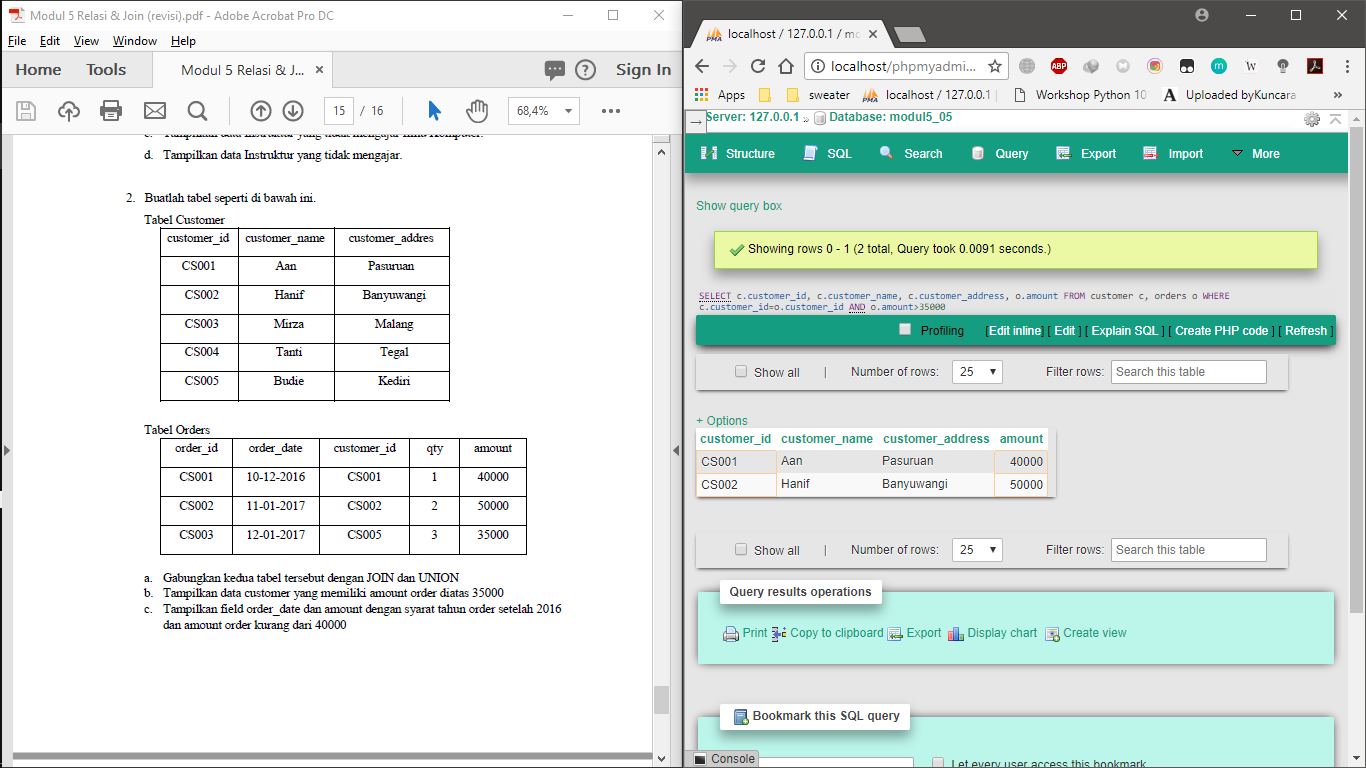
Syntax ini akan menampilkan semua field pada tabel customer yang dilakukan inner join dengan tabel orders “SELECT \* FROM customer c INNER JOIN orders o”. Pada inner join, nilai yang sama pada field customer\_id dari tabel customer dan tabel orders akan ditampilkan pada sintaks “ON c.customer\_id=o.customer\_id”.





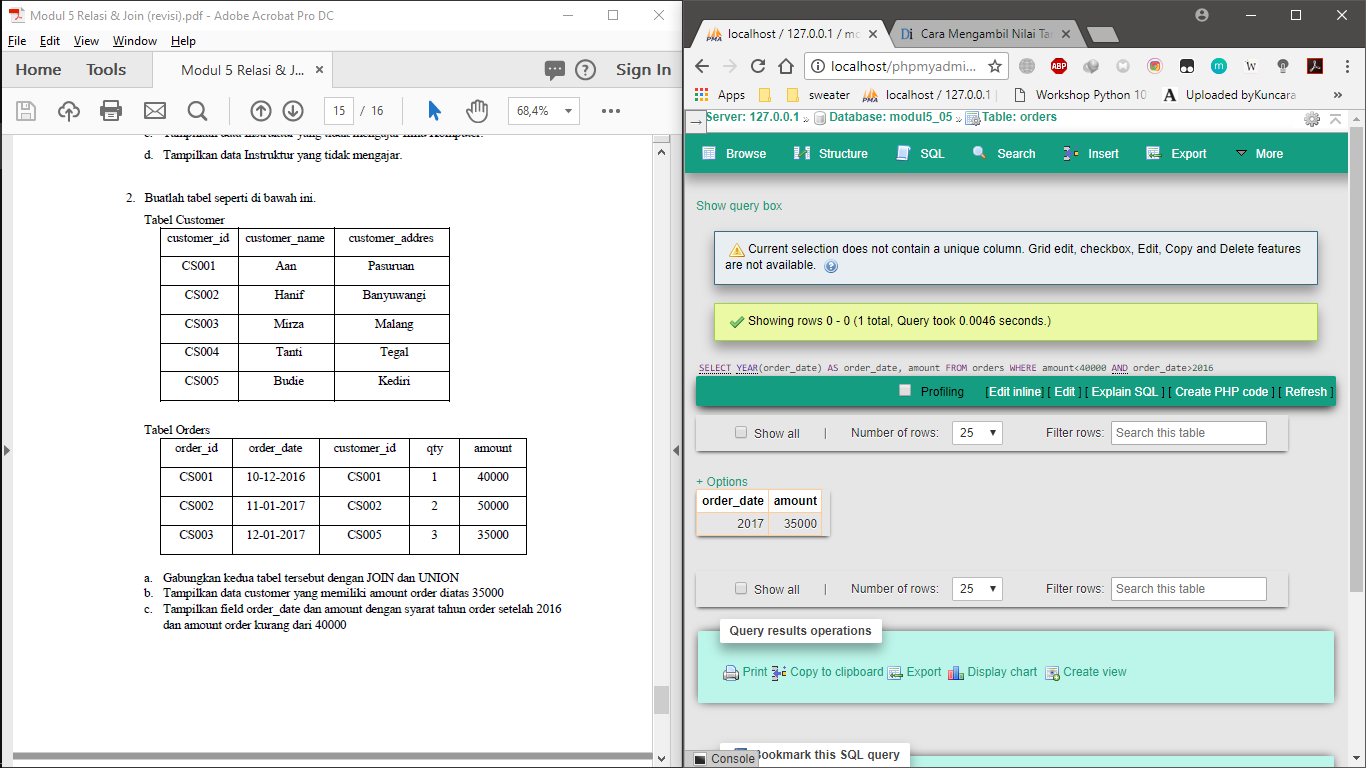
1. Tampilkan data customer yang memiliki amount order diatas 35000.





1. Tampilkan field order\_date dan amount dengan syarat tahun order setelah tahun 2016 dan amount order kurang dari 40000.





Pada syntax ini, akan ditampilkan tahun order dan jumlah order yang kurang dari 40000 “SELECT ...”. Syntax “YEAR(order\_date) AS order\_date” akan menampilkan tahun dari field order\_date dan akan ditampilkan pada field order\_date. Pada syntax “WHERE amount<40000 AND order\_date>2016” akan ditampilkan record amount dengan nilai kurang dari 40000 dan pada order\_date akan ditampilkan record setelah tahun 2016.

1. **Kesimpulan**

*Berisi kesimpulan yang diperoleh dalam modul yang dikerjakan, kesimpulan harus mencangkup semua elemen yang ada pada modul tersebut. Menggunakan bahasa indonesia yang baik dan benar. Elemen tambahan kesimpulan dapat berupa keluhan, problem solving yang dialami, dan sebagainya. Berupa paragraf, tidak numbered list atau bullet list.*

1. **Daftar Rujukan**

*Kadir, Abdul. 2013. Pemrograman Database MySQL untuk Pemula. Yogyakarta: MediaKom.*

*Penulisan daftar pustaka harus sesuai dengan PPKI*