

# **LAPORAN PRAKTIKUM**

## **MATA KULIAH PRAKTIKUM BASIS DATA**

Dosen Pengampu : Farid Angga Pribadi, S.Kom, M.Kom.

### **PERTEMUAN - 12**



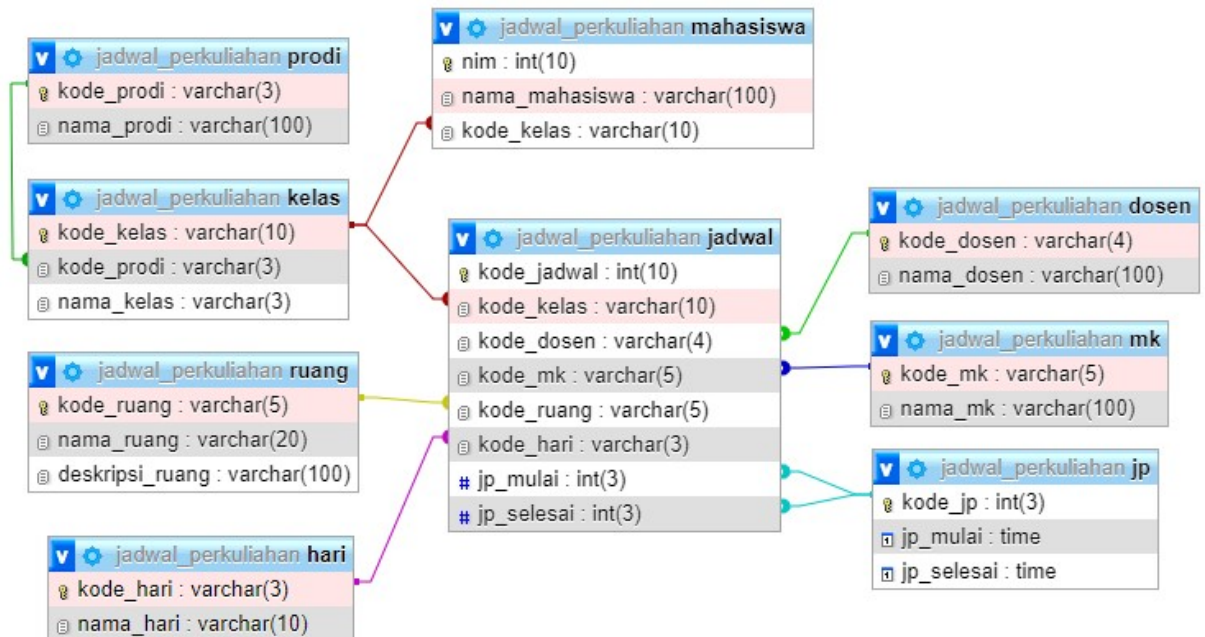
**Nama : M. Zidna Billah Faza**  
**NIM : 2341760030**  
**Prodi : D-IV Sistem Informasi Bisnis**

**JURUSAN TEKNOLOGI INFORMASI**  
**POLITEKNIK NEGERI MALANG**

**2024**

## Praktikum 1

- 1) Studi kasus yang digunakan sama dengan jobsheet DML dengan skema/model relasional/EER diagram dari database berikut.



- 2) Skema tersebut adalah skema database pada sebuah sistem informasi penjadwalan di jurusan Teknologi Informasi. Sesuai dengan percobaan pada jobsheet DML. Hapuslah tabel mahasiswa pada database jadwal\_perkuliahan dengan menggunakan perintah DROP TABLE, sehingga menghasilkan record keseluruhan tabel sebagai berikut (record tidak harus sama persis dengan tampilan ini):

```
MariaDB [jadwal_perkuliahan2]> select * from jadwal;
```

	kode_jadwal	kode_kelas	kode_dosen	kode_mk	kode_ruang	kode_hari	jp_mulai	jp_selesai
1	2021020101	D001	02001	0504	001	7	9	
2	2021010103	D001	02001	0506	002	9	11	
3	2021010105	D001	02001	0806	003	10	12	
4	2021010102	D001	02001	0506	004	1	3	
5	2021010106	D001	02001	0806	004	4	6	
6	2021010101	D001	02001	0506	004	7	9	
7	2021010104	D001	02001	0506	005	10	12	
8	2021010206	D002	02037	0702	001	7	12	
9	2021020202	D002	02036	0708	003	2	4	
10	2021010205	D002	02037	0713	004	1	6	
11	2021020209	D002	02025	0719	004	7	12	
12	2021020301	D003	02012	0508	002	1	4	
13	2021020302	D003	02012	0508	002	1	4	
14	2021010201	D003	02017	0719	003	2	5	
15	2021010202	D003	02017	0719	003	2	5	
16	2021010203	D003	02017	0507	005	2	5	
17	2021010106	D004	02028	0704	001	1	3	
18	2021020203	D004	02032	0507	001	10	12	
19	2021010201	D004	02034	0617	002	7	12	
20	2021020203	D004	02032	0708	004	1	6	
21	2021020104	D005	02016	0717	001	1	3	

```
MariaDB [jadwal_perkuliahan2]> select * from prodi;
```

kode_prodi	nama_prodi
001	D3 Manajemen Informatika
002	D4 Teknik Informatika

```
2 rows in set (0.001 sec)
```

```
MariaDB [jadwal_perkuliahan2]> select * from kelas;
```

kode_kelas	kode_prodi	nama_kelas
2021010101	001	MI-1A
2021010102	001	MI-1B
2021010103	001	MI-1C
2021010104	001	MI-1D
2021010105	001	MI-1E
2021010106	001	MI-1F
2021010107	001	MI-1H
2021010201	001	MI-2A
2021010202	001	MI-2B
2021010203	001	MI-2C
2021010204	001	MI-2D
2021010205	001	MI-2E

```
MariaDB [jadwal_perkuliahan2]> select * from ruang;
```

kode_ruang	nama_ruang	deskripsi_ruang
0501	RT01	Ruang Teori 1
0502	RT02	Ruang Teori 2
0503	RT03	Ruang Teori 3
0504	RT04	Ruang Teori 4
0505	RT05	Ruang Teori 5
0506	RT06	Ruang Teori 6
0507	RT07	Ruang Teori 7
0508	LPY1	Laboratorium Proyek 1
0615	LSI1	Laboratorium Sistem Informasi 1
0617	LSI2	Laboratorium Sistem Informasi 2
0618	LSI3	Laboratorium Sistem Informasi 3
0619	LPY2	Laboratorium Proyek 2
0620	LPY3	Laboratorium Proyek 3
0701	LPR1	Laboratorium Pemrograman 1
0702	LPR2	Laboratorium Pemrograman 2
0703	LPR3	Laboratorium Pemrograman 3
0704	LPR4	Laboratorium Pemrograman 4
0705	LPR5	Laboratorium Pemrograman 5
0706	LPR6	Laboratorium Pemrograman 6

```
MariaDB [jadwal_perkuliahan2]> select * from hari;
```

kode_hari	nama_hari
001	Senin
002	Selasa
003	Rabu
004	Kamis
005	Jumat
006	Sabtu
007	Minggu

```
7 rows in set (0.001 sec)
```

```
MariaDB [jadwal_perkuliahan2]> select * from dosen;
```

kode_dosen	nama_dosen
D001	Abdul Chalim, SAg., MPd.I
D002	Ade Ismail
D003	Agung Nugroho Pramudhita ST., MT.
D004	Ahmadi Yuli Ananta ST., MM.
D005	Ane Fany Novitasari, SH.MKn.
D006	Annisa Puspa Kirana MKom.
D007	Annisa Taufika Firdausi ST., MT.
D008	Anugrah Nur Rahmanto SSn., MDs.
D009	Ariadi Retno Ririd SKom., MKom.
D010	Arie Rachmad Syulistyo SKom., MKom.
D011	Arief Prasetyo SKom., MKom.
D012	Arwin Sumari ST., MT., DR.
D013	Atiqah Nurul Asri SPd., MPd.
D014	Bagas Satya Dian Nugraha, ST., MT.
D015	Banni Satria Andoko, S. Kom., M.MSI
D016	Budi Harijanto ST., MMKom.
D017	Cahya Rahmad ST., MKom. DR.Eng
D018	Candra Bella Vista SKom., MT.
D019	Candrasena Setiadi ST., MMT.
D020	Deasy Sandhya Elya Ikawati SSi., MSi.
D021	Deddy Kusbianto PA Ir. MMKom.

```
MariaDB [jadwal_perkuliah2]> select * from mk;
```

kode_mk	nama_mk
02001	Agama
02002	Aljabar Linier
02003	Algoritma dan Struktur Data
02004	Aljabar Linier
02005	Analisis Dan Desain Berorientasi Objek
02006	Bahasa Indonesia
02007	Bahasa Inggris
02008	Bahasa Inggris 2
02009	Bahasa Inggris Persiapan Kerja
02010	Basis Data
02011	Desain Pemrograman Web
02012	Digital Entrepreneurship
02013	E-Business
02014	Etika Profesi Bidang TI
02015	Internet Of Things
02016	Kewarganegaraan
02017	Komputasi Multimedia
02018	Machine Learning

```
MariaDB [jadwal_perkuliah2]> select * from jp;
```

kode_jp	jp_mulai	jp_selesai
1	07:00:00	07:50:00
2	07:50:00	08:40:00
3	08:40:00	09:30:00
4	09:40:00	10:30:00
5	10:30:00	11:20:00
6	11:20:00	12:10:00
7	12:50:00	13:40:00
8	13:40:00	14:30:00
9	14:30:00	15:20:00
10	15:30:00	15:30:00
11	16:20:00	17:10:00
12	17:10:00	18:00:00

12 rows in set (0.001 sec)

- 3) Setelah selesai membuat database diatas, lanjutkan ke Praktikum – Bagian 2.

## Praktikum 2

- 1) Untuk menampilkan baris dengan suatu kondisi tertentu, digunakan statement SELECT dengan WHERE. Eksekusi SQL berikut untuk menampilkan (record) pada tabel Ruang yang memiliki kolom nama\_ruang = LKJ1.

```
Run | New Tab | JSON
SELECT deskripsi_ruang
FROM ruang
WHERE nama_ruang = 'LKJ1';
```

deskripsi_ruang varchar(100)
Laboratorium Keamanan Jaringan 1

- 2) Untuk menampilkan data dengan eliminasi data yang sama (duplicate), digunakan statement SELECT dengan DISTINCT. Eksekusi SQL berikut untuk menampilkan kolom kode\_hari dari tabel jadwal yang bernilai tidak sama.

```
Run | New Tab | JSON
SELECT DISTINCT kode_hari
FROM jadwal; 4ms
```

kode_hari varchar(3)
001
002
003
004
005

- 3) Untuk menampilkan data yang spesifik digunakan statement SELECT dengan IN. Eksekusi SQL berikut untuk menampilkan kolom kode\_ruang, nama\_ruang dan deskripsi\_ruang dari tabel ruang yang memiliki nama\_ruang RT01 atau RT10.

```
Run | New Tab | JSON
SELECT *
FROM ruang
WHERE nama_ruang IN ('RT01', 'RT10');
```

kode_ruang varchar(5)	nama_ruang varchar(20)	deskripsi_ruang varchar(100)
0501	RT01	Ruang Teori 1
0806	RT10	Ruang Teori 10

- 4) Untuk menampilkan data pada jarak (range) tertentu digunakan statement SELECT dengan BETWEEN. Eksekusi SQL berikut untuk menampilkan kolom kode\_ruang dan nama\_ruang dan deskripsi\_ruang dari tabel ruang yang memiliki kode\_ruang antara 0501 dan 0508.

```
Run | New Tab | JSON
SELECT *
FROM ruang
WHERE kode_ruang
BETWEEN '0501' AND '0508';
```

kode_ruang varchar(5)	nama_ruang varchar(20)	deskripsi_ruang varchar(100)
0501	RT01	Ruang Teori 1
0502	RT02	Ruang Teori 2
0503	RT03	Ruang Teori 3
0504	RT04	Ruang Teori 4
0505	RT05	Ruang Teori 5
0506	RT06	Ruang Teori 6
0507	RT07	Ruang Teori 7
0508	LPY1	Laboratorium Proyek 1

- 5) Untuk menampilkan data yang memiliki kemiripan dengan keyword yang diinginkan digunakan SELECT dengan LIKE. Eksekusi SQL berikut untuk menampilkan kolom kode\_dosen, dan nama\_dosen, pada tabel dosen yang memiliki nama dengan huruf awal 'E'.

```
Run | New Tab | JSON
SELECT *
FROM dosen
WHERE nama_dosen like 'E%';
```



<b>kode_dosen</b> varchar(4)	<b>nama_dosen</b> varchar(100)
D028	Eka Larasati Amalia, SST., M
D029	Ekojono, ST., M.Kom.
D030	Elok Nur Hamdana, ST., MT
D031	Erfan Rohadi, ST., MEng., Ph

- 6) Untuk menampilkan susunan data dalam bentuk grup, digunakan SELECT dengan GROUP BY. Eksekusi SQL berikut untuk menampilkan kolom kode\_dosen, kode\_mk, kode\_ruang dan kode\_hari pada tabel jadwal yang dikelompokkan berdasarkan kode\_hari.

```
Run | New Tab | JSON
SELECT kode_dosen, kode_mk, kode_ruang, kode_hari
FROM jadwal
GROUP BY kode_hari; 4ms
```

<b>kode_dosen</b> varchar(4)	<b>kode_mk</b> varchar(5)	<b>kode_ruang</b> varchar(5)	<b>kode_hari</b> varchar(3)
D001	02001	0504	001
D001	02001	0506	002
D001	02001	0806	003
D001	02001	0506	004
D001	02001	0506	005

- 7) Untuk menampilkan baris secara spesifik dan terurut maju atau mundur, digunakan SELECT dengan ORDER BY. Eksekusi SQL berikut untuk menampilkan kolom kode\_jp dan jp\_mulai pada tabel jp dari jam yang paling mulai hingga berakhir.

```
Run | New Tab | JSON
SELECT kode_jp, jp_mulai
FROM jp
ORDER BY jp_mulai;
```



kode_jp int(3)	jp_mulai time
1	07:00:00
2	07:50:00
3	08:40:00
4	09:40:00
5	10:30:00
6	11:20:00
7	12:50:00
8	13:40:00
9	14:30:00
10	15:30:00
11	16:20:00
12	17:10:00

- 8) Untuk menampilkan data dengan kondisi dan atau atau tidak, digunakan SELECT dengan AND, OR and NOT. Eksekusi SQL berikut untuk menampilkan semua kolom pada tabel jadwal dengan kode\_hari = '001' dan jp\_mulai = 1.

```
Run | New Tab | JSON
SELECT *
FROM jadwal
WHERE kode_hari = '001' AND jp_mulai = 1;
```

kode_jadw int(10)	kode_kelas varchar(10)	kode_dose varchar(4)	kode_mk varchar(5)	kode_ruam varchar(5)	kode_hari varchar(3)	jp_mulai int(3)	jp_selesai int(3)
17	2021010106	D004	02028	0704	001	1	3
21	2021020104	D005	02016	0717	001	1	3
57	2021020204	D012	02018	0702	001	1	5
75	2021020103	D016	02038	0506	001	1	3
84	2021020302	D018	02039	0701	001	1	6
91	2021020207	D021	02005	0615	001	1	6
98	2021010205	D023	02023	0508	001	1	4
103	2021010201	D024	02037	0713	001	1	6
107	2021020305	D025	02024	0615	001	1	6
147	2021020108	D034	02008	0502	001	1	3
170	2021010101	D039	02040	0503	001	1	3
171	2021010102	D039	02040	0503	001	1	3
197	2021020208	D045	02037	0718	001	1	6
206	2021020308	D047	02022	0620	001	1	6
227	2021020206	D051	02037	0715	001	1	6
242	2021020106	D055	02003	0716	001	1	3
256	2021010203	D059	02023	0719	001	1	4
279	2021010301	D064	02006	0805	001	1	3

- 9) Untuk menampilkan data dari kolom yang terlibat dalam dua tabel dapat digunakan SELECT dengan UNION. UNION secara otomatis akan menghilangkan duplikasi. Eksekusi SQL berikut untuk menampilkan kolom kode\_hari yang ada di tabel hari atau jadwal.

```

Run | New Tab | JSON
SELECT kode_hari
FROM hari
UNION
SELECT kode_hari FROM jadwal; 2ms

```

kode_hari varchar
001
002
003
004
005
006
007

- 10) Untuk menampilkan data dari kolom yang terlibat dalam dua tabel dapat digunakan SELECT dengan UNION ALL. UNION ALL akan menampilkan duplikasi data. Eksekusi SQL berikut untuk menampilkan kolom nik yang ada di tabel penugasan atau departemen.

```
Run | New Tab | JSON
SELECT kode_hari
FROM hari
UNION ALL
SELECT kode_hari FROM jadwal; 2ms
```

[illegible]

### Praktikum 3

- 1) Untuk menampilkan data (test keanggotaan sub-query) yang berasal dari pemilihan tampilan data lain digunakan tambahan statement IN. Eksekusi SQL berikut untuk menampilkan kolom kode\_dosen, kode\_mk, kode\_hari, jp\_mulai dan jp\_selesai pada tabel jadwal dimana jp\_selesai adalah jp\_selesai yang paling lama pada tabel jadwal.

```
Run | New Tab | JSON
SELECT kode_dosen, kode_mk, kode_hari, jp_mulai, jp_selesai
FROM jadwal
WHERE jp_selesai
IN (SELECT MAX(jp_selesai) FROM jadwal);
```

kode_dose varchar	kode_mk varchar	kode_hari varchar	jp_mulai int	jp_selesai int
D001	02001	003	10	12
D001	02001	005	10	12
D002	02037	001	7	12
D002	02025	004	7	12
D004	02032	001	10	12
D004	02034	002	7	12
D005	02016	001	10	12
D006	02037	004	7	12
D006	02037	005	7	12
D007	02011	004	8	12
D008	02012	005	9	12
D008	02012	005	9	12
D009	02005	003	7	12
D011	02019	004	8	12
D015	02005	002	7	12

- 2) Untuk menampilkan data (perbandingan himpunan sub-query) yang berasal dari pemilihan tampilan data lain digunakan tambahan operator. Eksekusi SQL berikut untuk menampilkan kode\_dosen, kode\_mk, kode\_hari, jp\_mulai dan jp\_selesai pada tabel jadwal berdasarkan semua record kolom jp\_selesai harus bernilai kurang dari jp\_selesai yang dimiliki oleh jp\_selesai = 6 dari tabel jadwal.

```
Run | New Tab | JSON
SELECT kode_dosen, kode_mk, kode_hari, jp_mulai, jp_selesai
FROM jadwal
WHERE jp_selesai < ALL
(SELECT jp_selesai FROM jadwal WHERE jp_selesai = 6);
```

kode_dose varchar	kode_mk varchar	kode_hari varchar	jp_mulai int	jp_selesai int
D001	02001	004	1	3
D002	02036	003	2	4
D003	02012	002	1	4
D003	02012	002	1	4
D003	02017	003	2	5
D003	02017	003	2	5
D003	02017	005	2	5
D004	02028	001	1	3
D005	02016	001	1	3
D005	02016	002	1	3
D005	02016	004	1	3
D005	02016	005	1	3
D006	02010	002	1	3
D007	02038	003	3	5
D007	02038	003	3	5

- 3) Setelah berhasil mengeksekusi SQL tersebut, lanjutkan ke Praktikum - Bagian 4.

## Praktikum 4

- 1) Untuk menghitung rata-rata, digunakan tambahan statement AVG. Eksekusi SQL berikut untuk menampilkan rata-rata durasi perkuliahan dalam satu jadwal perkuliahan sesuai kolom jp\_mulai, jp\_selesai dari tabel jadwal.

```
Run | New Tab | JSON
SELECT AVG(jp_selesai - jp_mulai)
FROM jadwal;
```

AVG(jp\_selesai - jp\_mulai)  
newdecimal

3.2216

- 2) Untuk menghitung nilai tertinggi, digunakan tambahan statement MAX. Eksekusi SQL berikut untuk menampilkan dimulainya jam perkuliahan yang paling siang sesuai kolom jp\_mulai dari tabel jadwal.

```
Run | New Tab | JSON
SELECT MAX(jp_mulai)
FROM jadwal;
```

MAX(jp\_mulai)  
int

10

- 3) Untuk menghitung nilai terendah, digunakan tambahan statement MIN. Eksekusi SQL berikut untuk menampilkan berakhirnya jam perkuliahan yang paling cepat sesuai kolom jp\_selesai dari tabel jadwal.

```
Run | New Tab | JSON
SELECT MIN(jp_selesai)
FROM jadwal;
```

MIN(jp\_selesai)  
int

3

- 4) Untuk menghitung total jam ajar digunakan tambahan statement SUM. Eksekusi SQL berikut untuk menampilkan total jam ajar dosen dengan kode 'D001' sesuai jp\_mulai, jp\_selesai, dan kode\_dosen dari tabel jadwal.

```
Run | New Tab | JSON
SELECT SUM(jp_selesai - jp_mulai)
FROM jadwal
WHERE kode_dosen = 'D001';
```

SUM(jp\_selesai - jp\_mu  
newdecimal

14

Coba tampilkan jadwal dosen dengan kode 'D001' dan pastikan apakah benar total jam ajar dosen tersebut sesuai dengan perintah sum dengan nilai 14 pada gambar di atas.

```
Run | New Tab | JSON
SELECT * FROM jadwal WHERE kode_dosen = 'D001';
```

kode_jadw int(10)	kode_kelas varchar(10)	kode_dose varchar(4)	kode_mk varchar(5)	kode_ruang varchar(5)	kode_hari varchar(3)	jp_mulai int(3)	jp_selesai int(3)
1	2021020101	D001	02001	0504	001	7	9
2	2021010103	D001	02001	0506	002	9	11
3	2021010105	D001	02001	0806	003	10	12
4	2021010102	D001	02001	0506	004	1	3
5	2021010106	D001	02001	0806	004	4	6
6	2021010101	D001	02001	0506	004	7	9
7	2021010104	D001	02001	0506	005	10	12

- 5) Untuk menunjukkan jumlah baris pada kolom yang diinginkan digunakan COUNT(). Eksekusi SQL berikut untuk menampilkan jumlah baris pada kolom kode\_ruang dari tabel ruang.

```
Run | New Tab | JSON
SELECT COUNT(kode_ruang)
FROM ruang;
```

COUNT(kode\_ruang)  
bigint

36

- 6) Lanjutkan ke bagian Tugas!



## TUGAS

- 1) Himpunlah screenshot sintaksis SQL berikut hasil SELECT yang Anda dapat di komputer Anda pada bagian Praktikum dalam sebuah laporan dengan format PDF!
- 2) Bagaimana sintaksis untuk menampilkan semua kolom jadwal pada hari kamis yang dimulai pada pukul 08:40 ? Tunjukkan outputnya!

```
Run | New Tab | JSON
SELECT *
FROM jadwal
WHERE kode_hari = (SELECT kode_hari FROM hari WHERE nama_hari = 'Kamis')
AND jp_mulai = (SELECT kode_jp FROM jp WHERE jp_mulai = '08:40:00'); 7ms
```

kode_jadwal int	kode_kelas varchar	kode_dose varchar	kode_mk varchar	kode_ruang varchar	kode_hari varchar	jp_mulai int	jp_selesai int
80	2021020106	D017	02004	0805	004	3	6
194	2021020206	D044	02020	0707	004	3	5
195	2021020207	D044	02020	0707	004	3	5

- 3) Berapakah jumlah dosen yang mengajar pada hari selasa! Tunjukkan sintaksis dan outputnya! Kemudian tampilkan list semua kode dosen yang sesuai dengan kondisi tersebut! Tunjukkan sintaksis dan outputnya!

```
Run | New Tab | JSON | Copy
SELECT
  ROW_NUMBER() OVER () AS urutan,
  kode_dosen
FROM
  (SELECT DISTINCT jadwal.kode_dosen
   FROM jadwal
   WHERE jadwal.kode_hari = (SELECT kode_hari FROM hari WHERE nama_hari = 'Selasa'))
  AS kode_dosen_terpilih; 2ms
```

Cost: 2ms < 1 Total 51		
Q	urutan bigint	kode_dose varchar
> 27	27	D004
> 28	28	D013
> 29	29	D032
> 30	30	D043
> 31	31	D061
> 32	32	D074
> 33	33	D085
> 34	34	D005
> 35	35	D015
> 36	36	D034
> 37	37	D046
> 38	38	D062
> 39	39	D075
> 40	40	D006
> 41	41	D021
> 42	42	D035
> 43	43	D047
> 44	44	D063
> 45	45	D078
> 46	46	D007
> 47	47	D025
> 48	48	D036
> 49	49	D048
> 50	50	D064
> 51	51	D079

- 4) Tampilkan kode dosen yang mengajar pada hari Kamis dari tabel jadwal yang nama depannya diawali dengan huruf A. gunakan operasi intersect! Tunjukkan sintaksis dan output dari pencarian tersebut.

```
Run | New Tab | JSON | Copy
SELECT DISTINCT hari.nama_hari, dosen.kode_dosen, dosen.nama_dosen
FROM dosen, hari, jadwal j
WHERE j.kode_hari = hari.kode_hari
      AND hari.nama_hari = 'Kamis'
      AND j.kode_dosen = dosen.kode_dosen
      AND dosen.nama_dosen LIKE 'A%'; 2ms
```

nama_hari varchar	kode_dose varchar	nama_dosen varchar
Kamis	D001	Abdul Chalim, SAg., MPd.I
Kamis	D002	Ade Ismail
Kamis	D004	Ahmadi Yuli Ananta ST., MN
Kamis	D005	Ane Fany Novitasari, SH.MK
Kamis	D006	Annisa Puspa Kirana MKom.
Kamis	D007	Annisa Taufika Firdausi ST., I
Kamis	D008	Anugrah Nur Rahmanto SSr
Kamis	D011	Arief Prasetyo SKom., MKon
Kamis	D013	Atiqah Nurul Asri SPd., MPd

- 5) Tampilkan kode\_ruang tertentu pada hari tertentu dengan diurutkan berdasarkan dengan kode jam yang ter awal.

```
Run | New Tab | JSON | Copy
SELECT hari.nama_hari, ruang.kode_ruang, ruang.nama_ruang, ruang.deskripsi_ruang, jp.jp_mulai
FROM ruang, jadwal j, hari, jp
WHERE j.kode_hari = hari.kode_hari
      AND hari.nama_hari = 'Jumat'
      AND j.kode_ruang = ruang.kode_ruang
      AND j.jp_mulai = jp.kode_jp
ORDER BY jp.jp_mulai ASC; 3ms
```

nama_hari varchar	kode_ruang varchar	nama_ruang varchar	deskripsi_ruang varchar	jp_mulai time
Jumat	0707	LKJ1	Laboratorium Keamanan Jai	07:00:00
Jumat	0719	LPY4	Laboratorium Proyek 4	07:00:00
Jumat	0620	LPY3	Laboratorium Proyek 3	07:00:00
Jumat	0704	LPR4	Laboratorium Pemrogramar	07:00:00
Jumat	0615	LSI1	Laboratorium Sistem Inform	07:00:00
Jumat	0705	LPR5	Laboratorium Pemrogramar	07:00:00
Jumat	0501	RT01	Ruang Teori 1	07:00:00
Jumat	0706	LPR6	Laboratorium Pemrogramar	07:00:00
Jumat	0503	RT03	Ruang Teori 3	07:00:00
Jumat	0618	LSI3	Laboratorium Sistem Inform	07:00:00
Jumat	0715	LKJ3	Laboratorium Keamanan Jai	07:00:00
Jumat	0508	LPY1	Laboratorium Proyek 1	07:00:00
Jumat	0708	LPR7	Laboratorium Pemrogramar	07:00:00
Jumat	0720	LAI1	Laboratorium Kecerdasan B	07:00:00
Jumat	0620	LPY3	Laboratorium Proyek 3	07:00:00

GitHub :

[zidnafaz/Praktikum-Basis-Data \(github.com\)](https://github.com/zidnafaz/Praktikum-Basis-Data)

Google Drive :

<https://drive.google.com/drive/folders/1EWtAVnTPV3hhRXrjUouZ2evK5JkfrCkG?usp=sharing>