



LIBRARY

DATABASE

---

NANDO DYAS ARYA - KAMPUS MERDEKA 1 - ANGGA

SEPT 05

# CONTENT

---

## Define Use Case

Define the story of the library database with some case.

## Create ER Diagram

Minimum 3 Entities with 3 Attributes for each Entities.

## Create Syntax Table

Constraint (Primary Key, Not Null, References) .

## Create Syntax Insert Data

Each table at least 3 data

## Co reate scenarianalysis

At least 2 table

# DEFINE

---

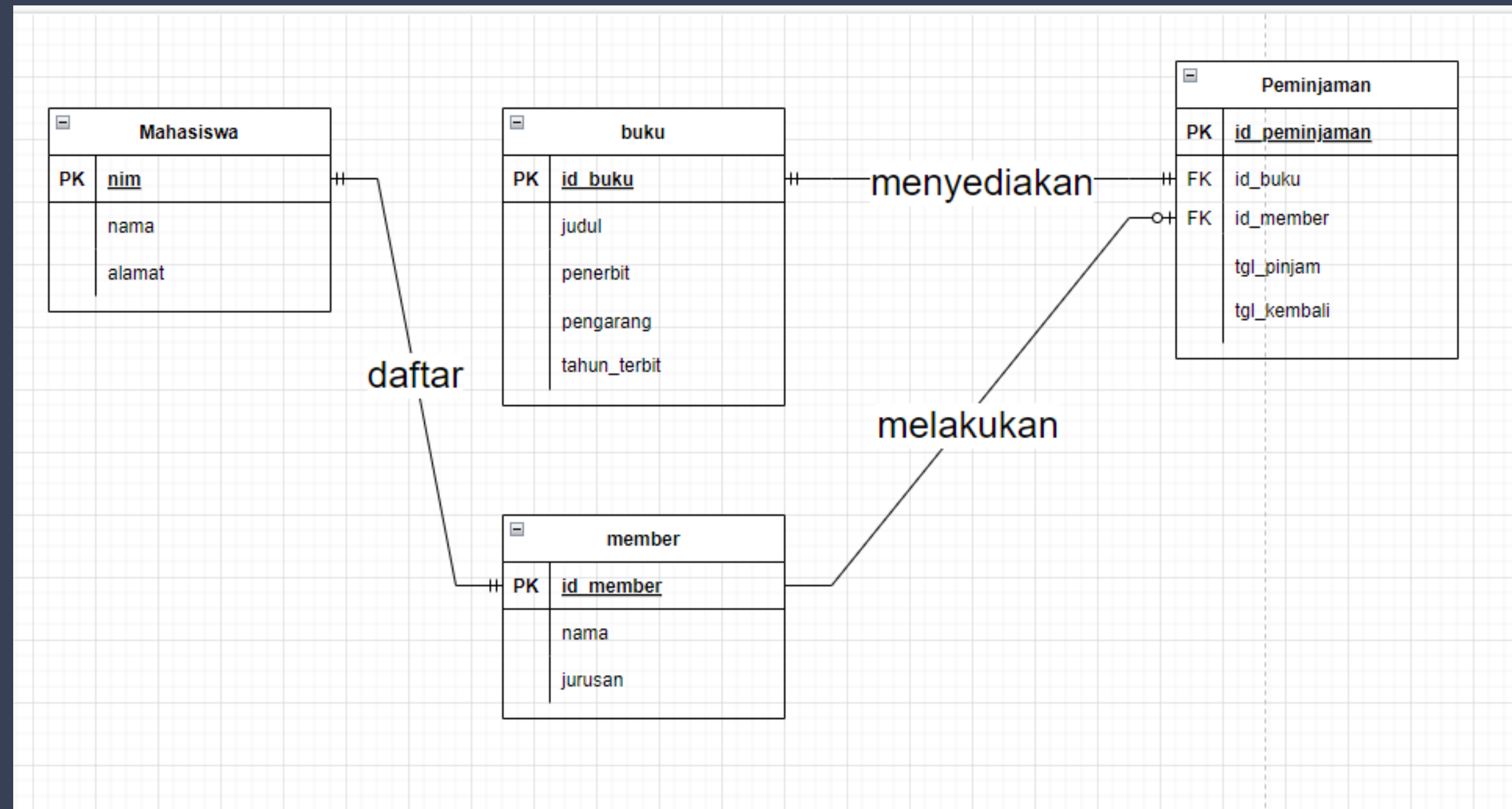
When registering to become a member of the MIPA Faculty library, the names, student numbers and addresses of students are recorded. After that, they will get a membership card consisting of ID, name and major, and they can borrow books from the library. Each book has data on book number, title, author, publisher, year of publication. One author can write several books.

A member may borrow several books. Each borrower will note the date of borrowing, date of return, loan id, book id and member id. Libraries apply a rule that borrowed books must be returned exactly one week after borrowing.

However, students are sometimes late in returning books, so they get punishment. The amount of the punishment is IDR 500,- per day of delay. Students are considered late if they return books later than 1 week. The punishment will be established by the student as of the 3rd of the month.

At the beginning of March 2020 the head of the library wants to check the punishment that must be pay by each student in February 2020.

# ER DIAGRAM



3 Entities with  
3 Attributes for each  
Entities.



# CREATE TABLE

---

## Table Mahasiswa & Buku

```
<postgres> Script-2  <postgres> Tugas Trainer SQL X
--create table mahasiswa
create table mahasiswa(
nim varchar(50) not null,
nama varchar(50) not null,
alamat varchar (50)not null,
primary key("nim")
);

--create table buku
create table buku(
id_buku varchar(50) not null,
judul varchar (50) not null,
penerbit varchar(50) not null,
pengarang varchar(50) not null,
tahun_terbit varchar(50) not null,
primary key ("id_buku")
);
```

## Table Member & Peminjaman

```
<postgres> Script-2  <postgres> Tugas Trainer SQL X
--create table member
create table member(
id_member varchar(50) not null,
nama varchar(50) not null,
jurusan varchar(50) not null,
primary key ("id_member")
);

-- create table peminjaman
create table peminjaman(
id_peminjaman varchar(50) not null,
id_buku varchar(50) not null,
id_member varchar(50) not null,
tgl_pinjam date not null,
tgl_kembali date not null,
primary key ("id_peminjaman"),
constraint fk_buku
foreign key("id_buku")
references "buku"("id_buku"),
constraint fk_member
foreign key("id_member")
references "member"("id_member")
);
```

# Data Mahasiswa

```
--insert data table mahasiswa
insert into mahasiswa
values
('1901', 'Ann', 'Batu'),
('1902', 'Widya', 'Malang'),
('1903', 'Tanta', 'Batu'),
('1904', 'Romi', 'Probolinggo'),
('1905', 'Alfian', 'Lamongan'),
('1906', 'Faris', 'Blitar'),
('1907', 'Naufal', 'Bojonegoro'),
('1908', 'Ihsan', 'Riau'),
('1909', 'Nijar', 'Blitar'),
('1910', 'Aisyah', 'Malang');
```

## Table & Data Mahasiswa

<postgres> Script-2      <postgres> Tugas Trainer SQL

Properties    Data    ER Diagram

**mahasiswa** | Enter a SQL expression to filter results (use Ctrl+)

	nim	nama	alamat	
1	1901	Ann	Batu	
2	1902	Widya	Malang	
3	1903	Tanta	Batu	
4	1904	Romi	Probolinggo	
5	1905	Alfian	Lamongan	
6	1906	Faris	Blitar	
7	1907	Naufal	Bojonegoro	
8	1908	Ihsan	Riau	
9	1909	Nijar	Blitar	
10	1910	Aisyah	Malang	

# INSERT DATA

## Data Buku

```
<postgres> Script-2  <postgres> Tugas Trainer SQL x
-- insert data table buku
insert into buku
values
('1011','Matahari','Rlangga','Dona','2005'),
('1012','Bulan','Gramedia','Doni','2003'),
('1013','Bintang','Amazon','Dona','2017'),
('1014','Bumi','Pena','Donu','2018'),
('1015','Mars','Amazon','Dono','2012'),
('1016','Asteroid','Grasindo','Dona','2017'),
('1017','Jupiter','Amazon','Doni','2018'),
('1018','Saturnus','Pena','Dono','2017'),
('1019','Uranus','Grasindo','Dona','2011'),
('1020','Neptunus','Pena','Dono','2016');
```

## Table & Data Buku

postgres Databases ka

buku					
Enter a SQL expression to filter results (use Ctrl+Space)					
	id_buku	judul	penerbit	pengarang	tahun_terbit
1	1011	Matahari	Rlangga	Dona	2005
2	1012	Bulan	Gramedia	Doni	2003
3	1013	Bintang	Amazon	Dona	2017
4	1014	Bumi	Pena	Donu	2018
5	1015	Mars	Amazon	Dono	2012
6	1016	Asteroid	Grasindo	Dona	2017
7	1017	Jupiter	Amazon	Doni	2018
8	1018	Saturnus	Pena	Dono	2017
9	1019	Uranus	Grasindo	Dona	2011
10	1020	Neptunus	Pena	Dono	2016

# INSERT DATA

## Data Member

```
<postgres> Script-2  <postgres> Tugas Trainer SQL X
--insert data table member
insert into "member"
values
('3001','Ann','Fisika'),
('3002','Alfian','Kimia'),
('3003','Tanta','Biologi'),
('3004','Romi','Fisika'),
('3005','Widya','Matematika'),
('3006','Faris','Biologi'),
('3007','Naufal','Matematika'),
('3008','Ihsan','Matematika'),
('3009','Nijar','Matematika'),
('3010','Aisyah','Fisika');
```

## Table & Data Member

member | Enter a SQL expression to filter results (use Ctrl+Space)

	id_member	nama	jurusan	
1	3001	Ann	Fisika	
2	3002	Alfian	Kimia	
3	3003	Tanta	Biologi	
4	3004	Romi	Fisika	
5	3005	Widya	Matematika	
6	3006	Faris	Biologi	
7	3007	Naufal	Matematika	
8	3008	Ihsan	Matematika	
9	3009	Nijar	Matematika	
10	3010	Aisyah	Fisika	



# INSERT DATA

## Data Peminjaman

```
<postgres> Script-2  <postgres> Tugas Trainer SQL X
--insert data table peminjaman
insert into peminjaman
values
('1','1011','3001','02/02/2020','09/02/2020'),
('2','1012','3002','03/02/2020','19/02/2020'),
('3','1013','3003','04/02/2020','11/02/2020'),
('4','1014','3004','05/02/2020','12/02/2020'),
('5','1015','3005','06/02/2020','17/02/2020'),
('6','1016','3006','07/02/2020','17/02/2020'),
('7','1017','3007','08/02/2020','15/02/2020'),
('8','1018','3008','09/02/2020','23/02/2020'),
('9','1019','3009','10/02/2020','24/02/2020'),
('10','1020','3010','11/02/2020','20/02/2020');
```

## Table & Data Peminjaman

postgres Databases kampus\_merdeka

peminjaman Enter a SQL expression to filter results (use Ctrl+Space)

	id_peminjaman	id_buku	id_member	tgl_pinjam	tgl_kembali
1	1	1011	3001	2020-02-02	2020-02-09
2	2	1012	3002	2020-02-03	2020-02-19
3	3	1013	3003	2020-02-04	2020-02-11
4	4	1014	3004	2020-02-05	2020-02-12
5	5	1015	3005	2020-02-06	2020-02-17
6	6	1016	3006	2020-02-07	2020-02-17
7	7	1017	3007	2020-02-08	2020-02-15
8	8	1018	3008	2020-02-09	2020-02-23
9	9	1019	3009	2020-02-10	2020-02-24
10	10	1020	3010	2020-02-11	2020-02-20

# SCENARIANALYSIS

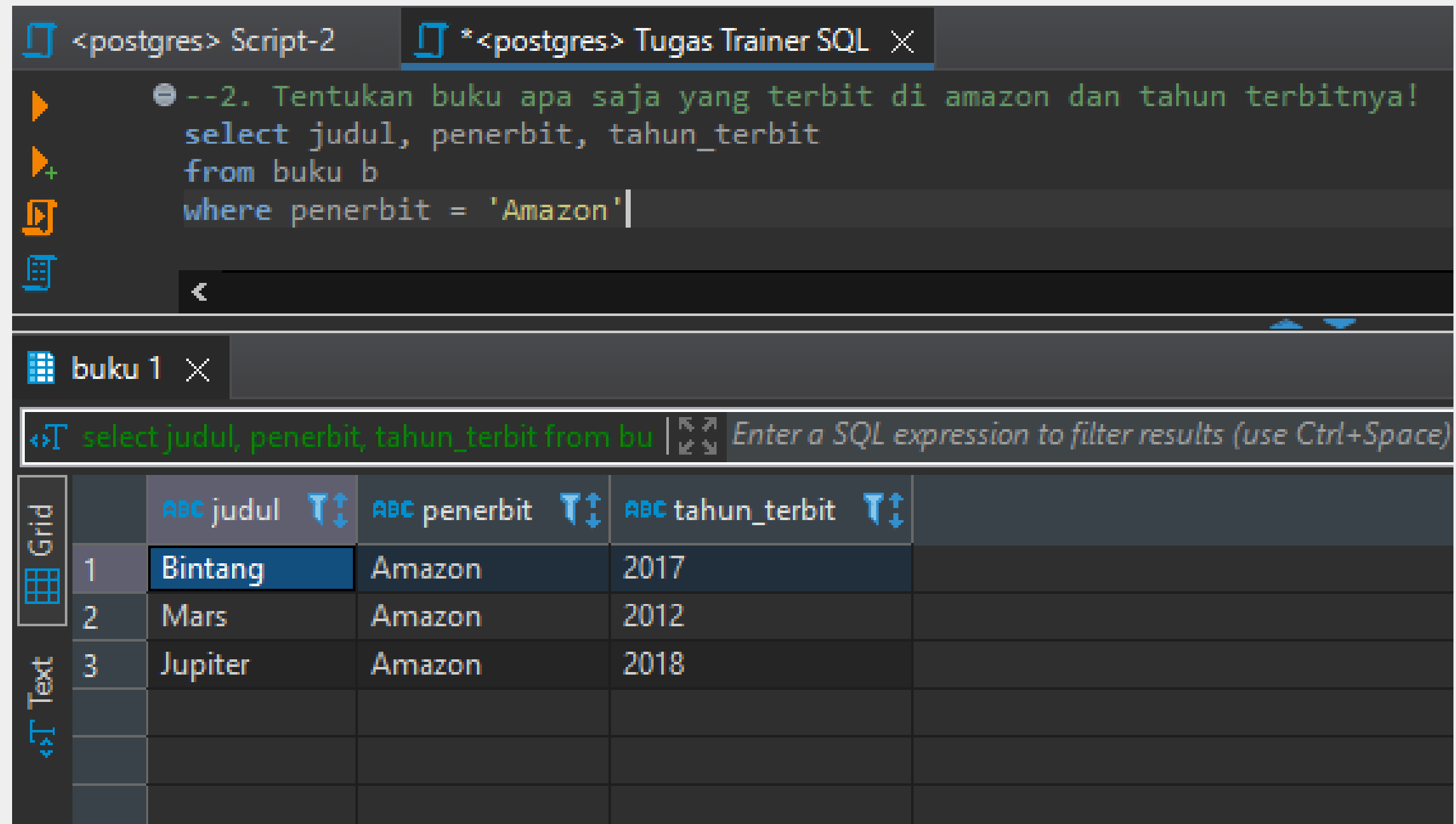
- Determine who is a member of the library who is in the department of physics, chemistry and mathematics

The screenshot shows a PostgreSQL SQL client interface. The top pane displays a SQL query: `--1. Tentukan siapa member perpustakaan yang berada di jurusan fisika, kimia dan mamtematika`, `select nama, jurusan`, `from "member" m`, `where jurusan not in ('Biologi')`. The bottom pane shows the results of the query in a table format. The table has two columns: 'nama' and 'jurusan'. The results are as follows:

	nama	jurusan
1	Ann	Fisika
2	Alfian	Kimia
3	Romi	Fisika
4	Widya	Matematika
5	Naufal	Matematika
6	Ihsan	Matematika
7	Nijar	Matematika
8	Aisyah	Fisika

# SCENARIANALYSIS

- Determine what books are published on amazon and the year they were published



The screenshot shows a PostgreSQL IDE interface. The top pane displays a SQL query in a script editor:

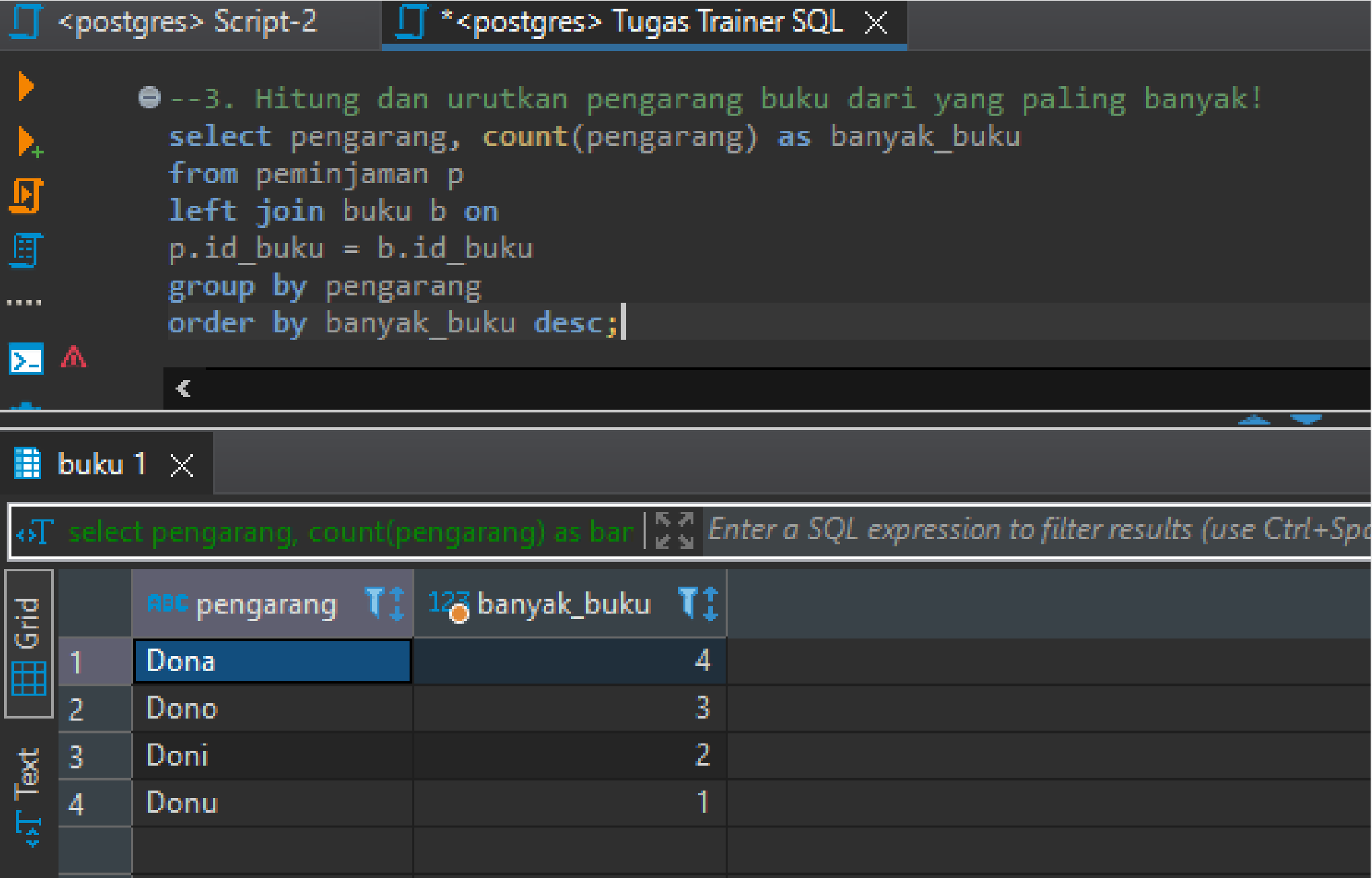
```
--2. Tentukan buku apa saja yang terbit di amazon dan tahun terbitnya!  
select judul, penerbit, tahun_terbit  
from buku b  
where penerbit = 'Amazon'
```

The bottom pane shows the results of the query in a table view. The table has four columns: 'judul', 'penerbit', 'tahun\_terbit', and an empty column. The results are as follows:

	judul	penerbit	tahun_terbit	
1	Bintang	Amazon	2017	
2	Mars	Amazon	2012	
3	Jupiter	Amazon	2018	

# SCENARIANALYSIS

- Count and order the authors of the books from the most!



The screenshot shows a PostgreSQL SQL client interface. The top panel displays a SQL query in a script editor. The query is designed to count the number of books written by each author and order them from highest to lowest count. The query text is as follows:

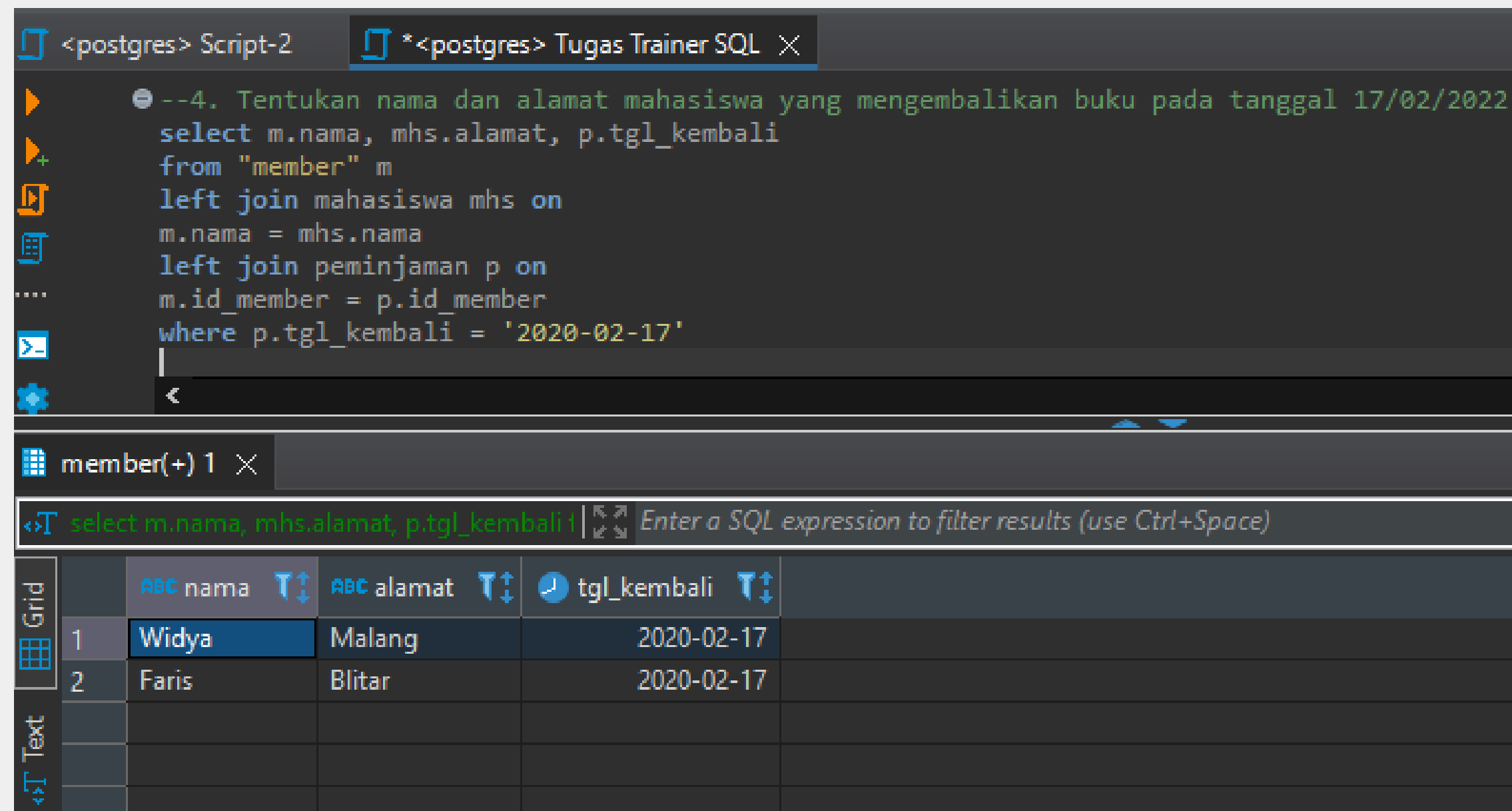
```
--3. Hitung dan urutkan pengarang buku dari yang paling banyak!  
select pengarang, count(pengarang) as banyak_buku  
from peminjaman p  
left join buku b on  
p.id_buku = b.id_buku  
group by pengarang  
order by banyak_buku desc;
```

Below the script editor, a results window titled 'buku 1' is open, showing the query results in a table. The table has two columns: 'pengarang' (author) and 'banyak\_buku' (number of books). The results are ordered by the number of books in descending order.

	ABC pengarang	123 banyak_buku
1	Dona	4
2	Dono	3
3	Doni	2
4	Donu	1

# SCENARIANALYSIS

- Determine the name and address of the student who returns the book on 17/02/2022!

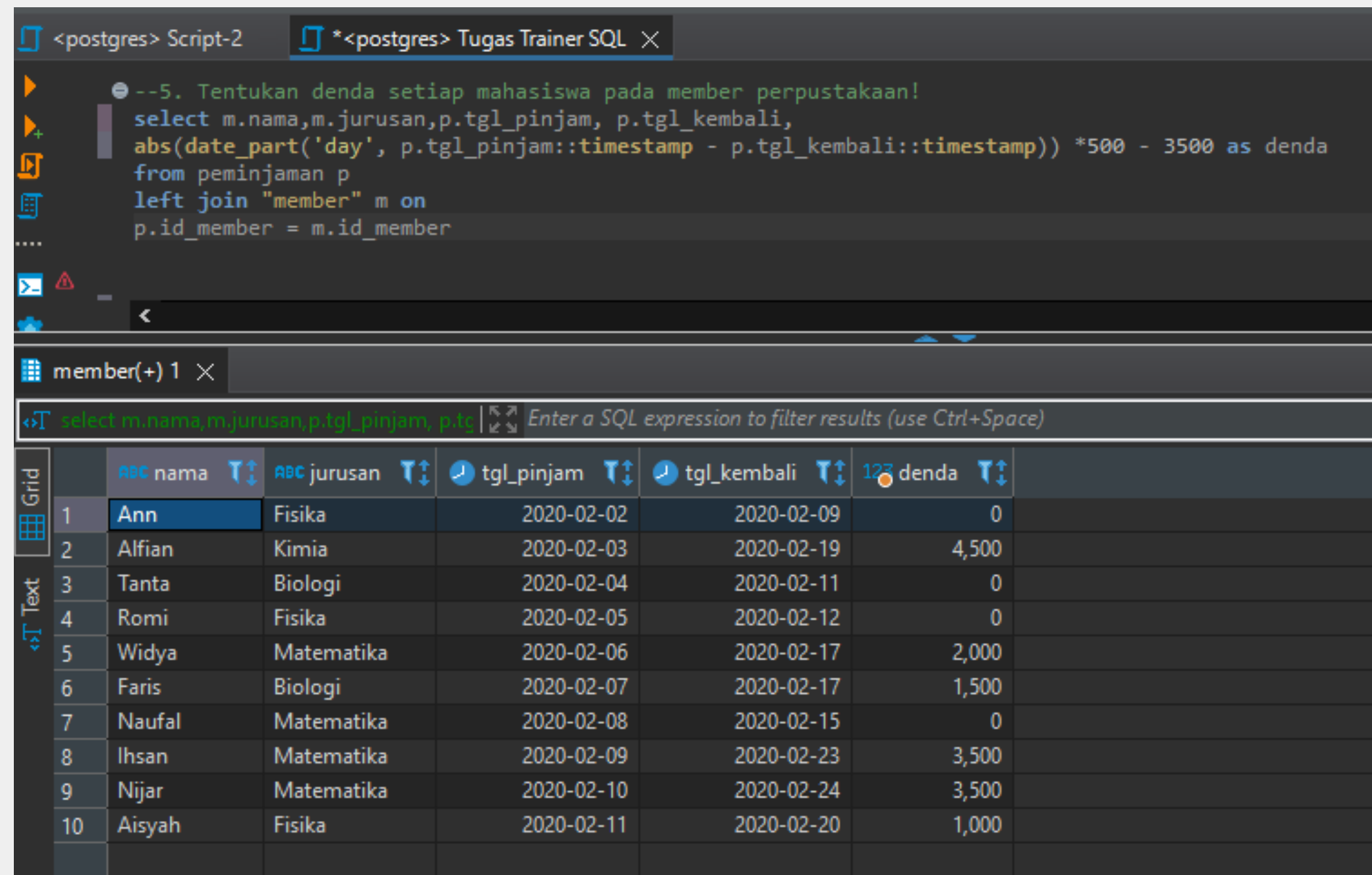


```
--4. Tentukan nama dan alamat mahasiswa yang mengembalikan buku pada tanggal 17/02/2022
select m.nama, mhs.alamat, p.tgl_kembali
from "member" m
left join mahasiswa mhs on
m.nama = mhs.nama
left join peminjaman p on
m.id_member = p.id_member
where p.tgl_kembali = '2020-02-17'
```

	nama	alamat	tgl_kembali
1	Widya	Malang	2020-02-17
2	Faris	Blitar	2020-02-17

# MAIN SCENARIO ANALYSIS

- Determine the punishment for each student on the library members!



The screenshot shows a PostgreSQL SQL client interface. The top pane displays a SQL query to calculate fines for library members. The bottom pane shows the results of the query in a table format.

**SQL Query:**

```
--5. Tentukan denda setiap mahasiswa pada member perpustakaan!  
select m.nama,m.jurusan,p.tgl_pinjam, p.tgl_kembali,  
abs(date_part('day', p.tgl_pinjam::timestamp - p.tgl_kembali::timestamp)) *500 - 3500 as denda  
from peminjaman p  
left join "member" m on  
p.id_member = m.id_member
```

**Query Results:**

	nama	jurusan	tgl_pinjam	tgl_kembali	denda
1	Ann	Fisika	2020-02-02	2020-02-09	0
2	Alfian	Kimia	2020-02-03	2020-02-19	4,500
3	Tanta	Biologi	2020-02-04	2020-02-11	0
4	Romi	Fisika	2020-02-05	2020-02-12	0
5	Widya	Matematika	2020-02-06	2020-02-17	2,000
6	Faris	Biologi	2020-02-07	2020-02-17	1,500
7	Naufal	Matematika	2020-02-08	2020-02-15	0
8	Ihsan	Matematika	2020-02-09	2020-02-23	3,500
9	Nijar	Matematika	2020-02-10	2020-02-24	3,500
10	Aisyah	Fisika	2020-02-11	2020-02-20	1,000



THANK YOU

