

Praktikum MBD 1

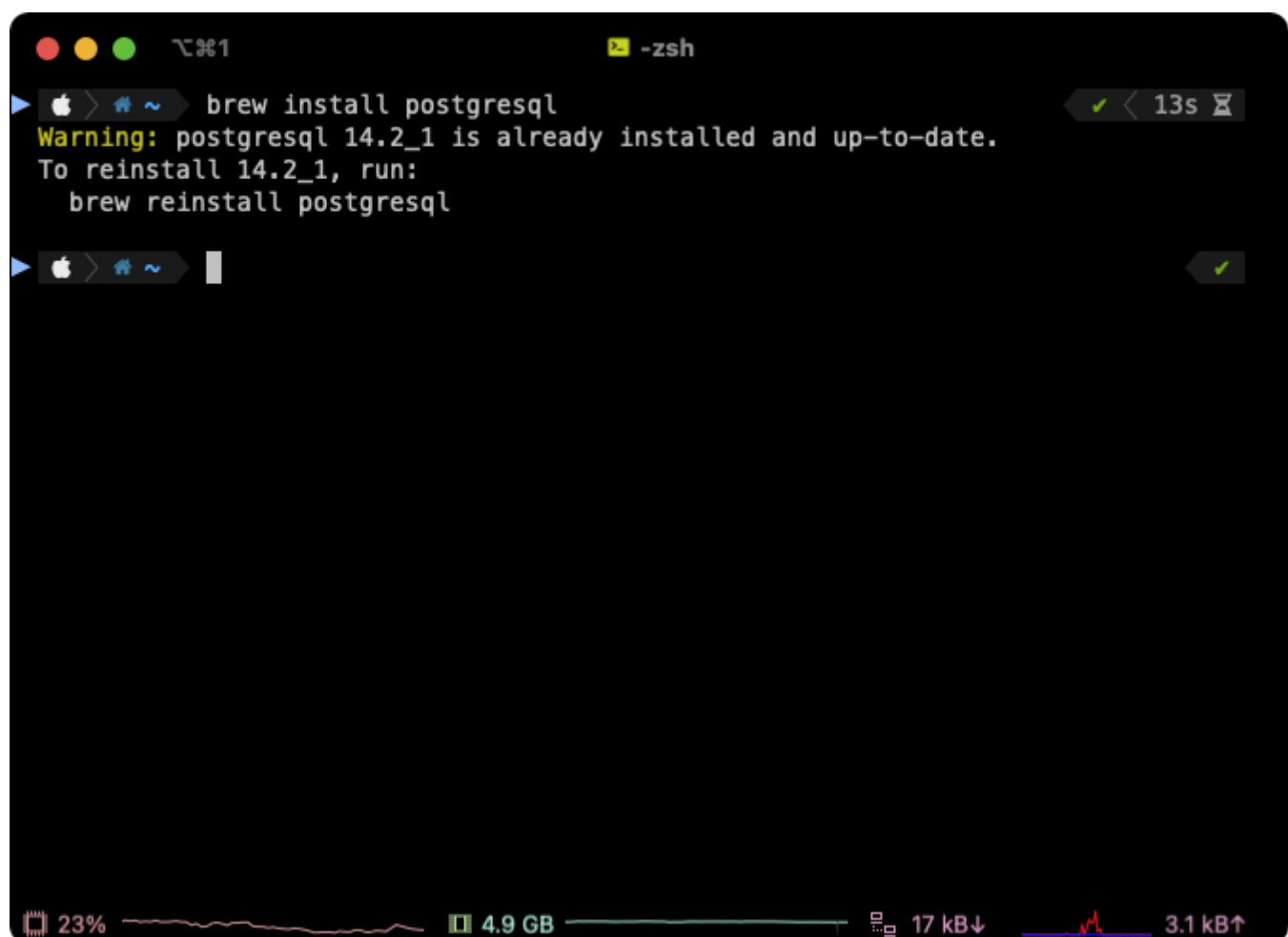
Install SQL dan IDE

Penulis akan menginstall PostgreSQL sebagai SQL pilihan penulis dan Datagrip sebaga IDE-nya.

Install PostgreSQL

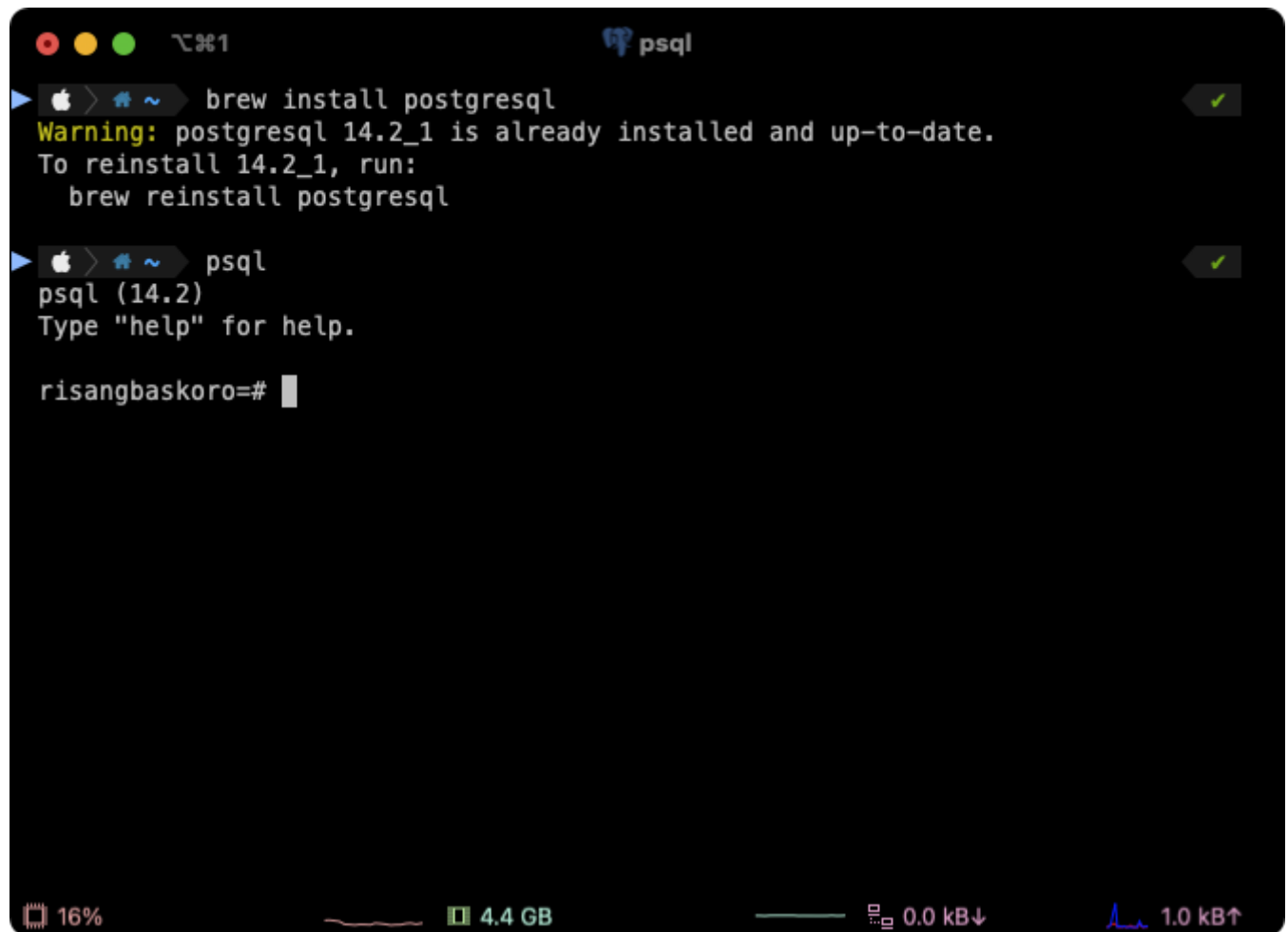
Penulis menggunakan macOS versi 12. Ada beberapa cara untuk menginstall PostgreSQL (Postgres atau pgSQL) di macOS. Penulis menggunakan *package manager* untuk Mac yaitu [Homebrew](#).

Penulis telah memiliki Homebrew, jadi penulis akan menggunakan terminal untuk menginstal Postgres. Instalasi menggunakan command `brew install postgresql`.



```
~ % brew install postgresql
Warning: postgresql 14.2_1 is already installed and up-to-date.
To reinstall 14.2_1, run:
  brew reinstall postgresql
~ %
```

Di gambar tersebut, penulis sudah menginstall Postgres sebelumnya. Untuk membuktikan bahwa instalasi sudah benar, jalankan Postgres dengan perintah `psql` di terminal.

A screenshot of a macOS terminal window with a dark background. The window title bar shows standard macOS window controls (red, yellow, green buttons) and a title "psql". The terminal content shows two commands being executed. The first command is "brew install postgresql", which results in a warning that postgresql 14.2_1 is already installed and up-to-date, and suggests running "brew reinstall postgresql". The second command is "psql", which enters the psql (14.2) prompt and shows "Type 'help' for help." followed by the user's prompt "risangbaskoro=#". On the right side of the terminal, there are two green checkmark icons indicating successful execution. At the bottom of the terminal window, there is a status bar showing "16%" battery, "4.4 GB" memory usage, "0.0 kB↓" network download, and "1.0 kB↑" network upload.

```
➤ [Apple] > [Home] ~ brew install postgresql ✓
Warning: postgresql 14.2_1 is already installed and up-to-date.
To reinstall 14.2_1, run:
  brew reinstall postgresql

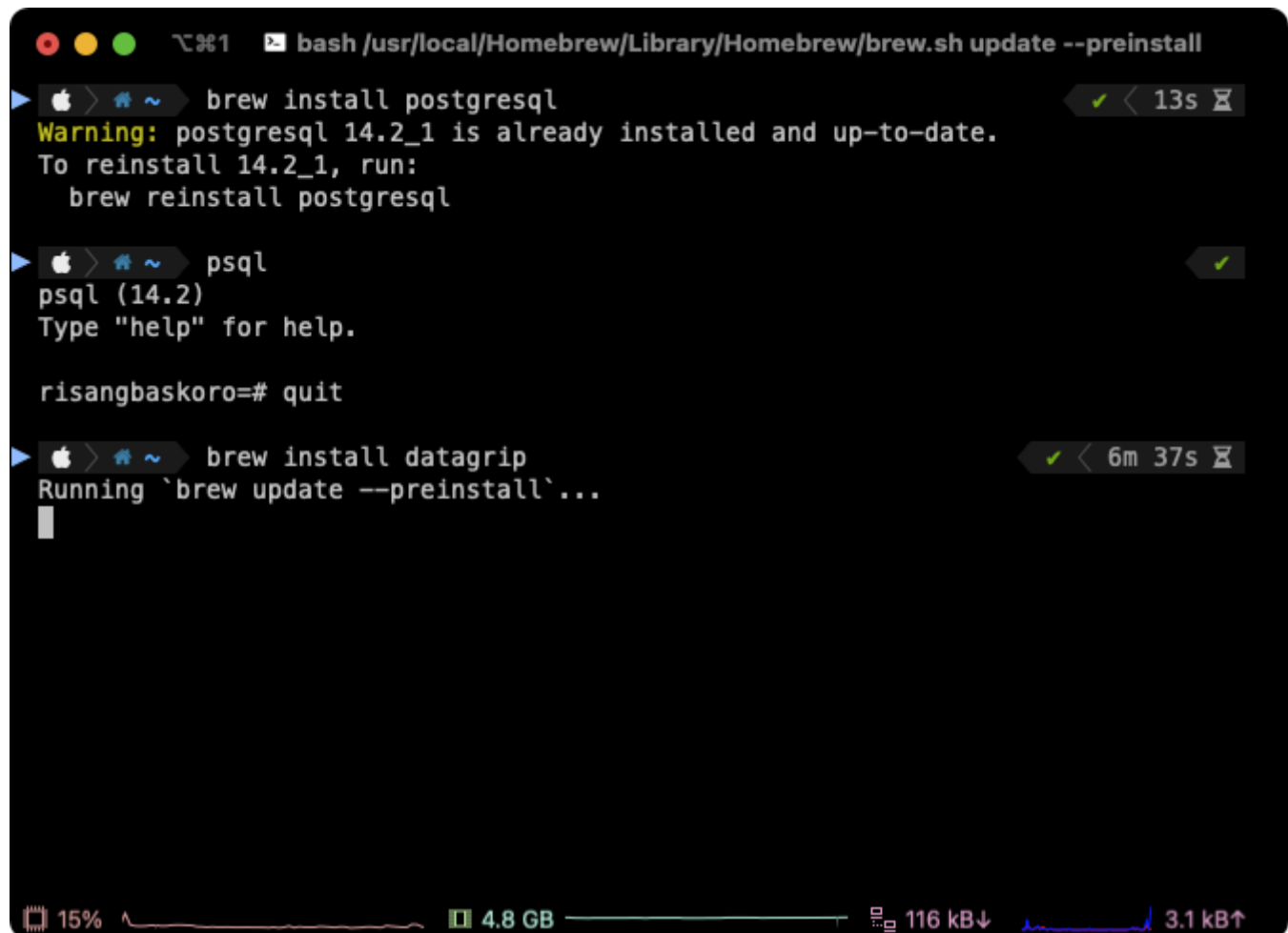
➤ [Apple] > [Home] ~ psql ✓
psql (14.2)
Type "help" for help.

risangbaskoro=#
```

Perintah `quit` atau `\q` untuk keluar dari PostgreSQL.

Install Datagrip

Penulis juga akan menggunakan Homebrew untuk menginstall Datagrip menggunakan perintah `brew install datagrip`.



```
bash /usr/local/Homebrew/Library/Homebrew/brew.sh update --preinstall  
➤ brew install postgresql 13s  
Warning: postgresql 14.2_1 is already installed and up-to-date.  
To reinstall 14.2_1, run:  
    brew reinstall postgresql  
➤ psql  
psql (14.2)  
Type "help" for help.  
risangbaskoro=# quit  
➤ brew install datagrip 6m 37s  
Running `brew update --preinstall`...
```

The terminal window shows the installation of PostgreSQL and Datagrip. The first command is `brew install postgresql`, which results in a warning that PostgreSQL 14.2_1 is already installed and up-to-date. The second command is `psql`, which opens the PostgreSQL command-line interface. The third command is `brew install datagrip`, which starts the installation of Datagrip. The terminal also shows the system status at the bottom: 15% battery, 4.8 GB memory, 116 kB download, and 3.1 kB upload.

Datagrip sudah bisa dibuka dari Launchpad maupun Spotlight.

Membuat Database dan Table

Membuat Database

Saat ini, penulis pertama akan menggunakan terminal untuk menambahkan database.

Untuk melihat semua database yang ada, penulis menggunakan perintah `\l`.

```

risangbaskoro=# \l
               List of databases
   Name          |  Owner   | Encoding | Collate | Ctype |           Access privileges
-----+-----+-----+-----+-----+-----
codegreen        | risangbaskoro | UTF8     | C       | C      |
kuliah_mbd       | risangbaskoro | UTF8     | C       | C      |
learn_prisma     | risangbaskoro | UTF8     | C       | C      |
postgres         | risangbaskoro | UTF8     | C       | C      |
risangbaskoro    | risangbaskoro | UTF8     | C       | C      |
template0       | risangbaskoro | UTF8     | C       | C      | =c/risangbaskoro
+
risangbaskoro    |             |          |         |        | risangbaskoro=CTc/
template1       | risangbaskoro | UTF8     | C       | C      | =c/risangbaskoro
+
risangbaskoro    |             |          |         |        | risangbaskoro=CTc/
(7 rows)

(END)

```

Penulis membuat membuat sebuah database dengan nama "risang" menggunakan perintah berikut:

```
create database risang;
```

Kemudian penulis melihat kembali database yang ada.

```

risangbaskoro=# create database risang;
CREATE DATABASE
risangbaskoro=# \l

```

Name	Owner	Encoding	Collate	Ctype	Access privileges
codegreen	risangbaskoro	UTF8	C	C	
kuliah_mbd	risangbaskoro	UTF8	C	C	
learn_prisma	risangbaskoro	UTF8	C	C	
postgres	risangbaskoro	UTF8	C	C	
risang	risangbaskoro	UTF8	C	C	
risangbaskoro	risangbaskoro	UTF8	C	C	
template0	risangbaskoro	UTF8	C	C	=c/risangbaskoro
+					
risangbaskoro					risangbaskoro=CTc/
template1	risangbaskoro	UTF8	C	C	=c/risangbaskoro
+					
risangbaskoro					risangbaskoro=CTc/

```

(8 rows)

(END)

```

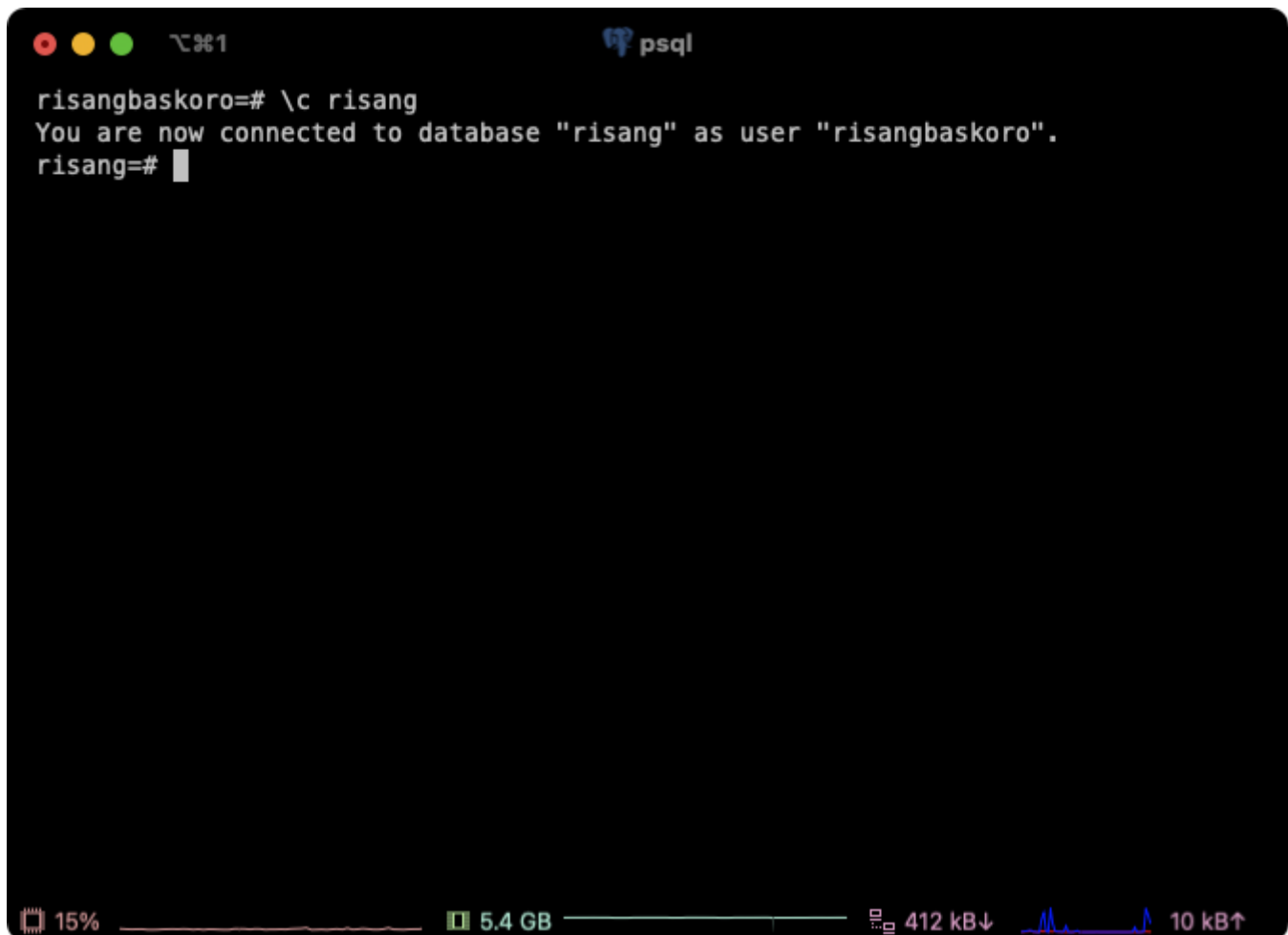
12% 4.9 GB 13 kB↓ 6.2 kB↑

Koneksi Database

Sebelum membuat table, pengguna memastikan bahwa koneksi yang tersambung adalah koneksi dengan database yang tepat.

Koneksi dari Terminal

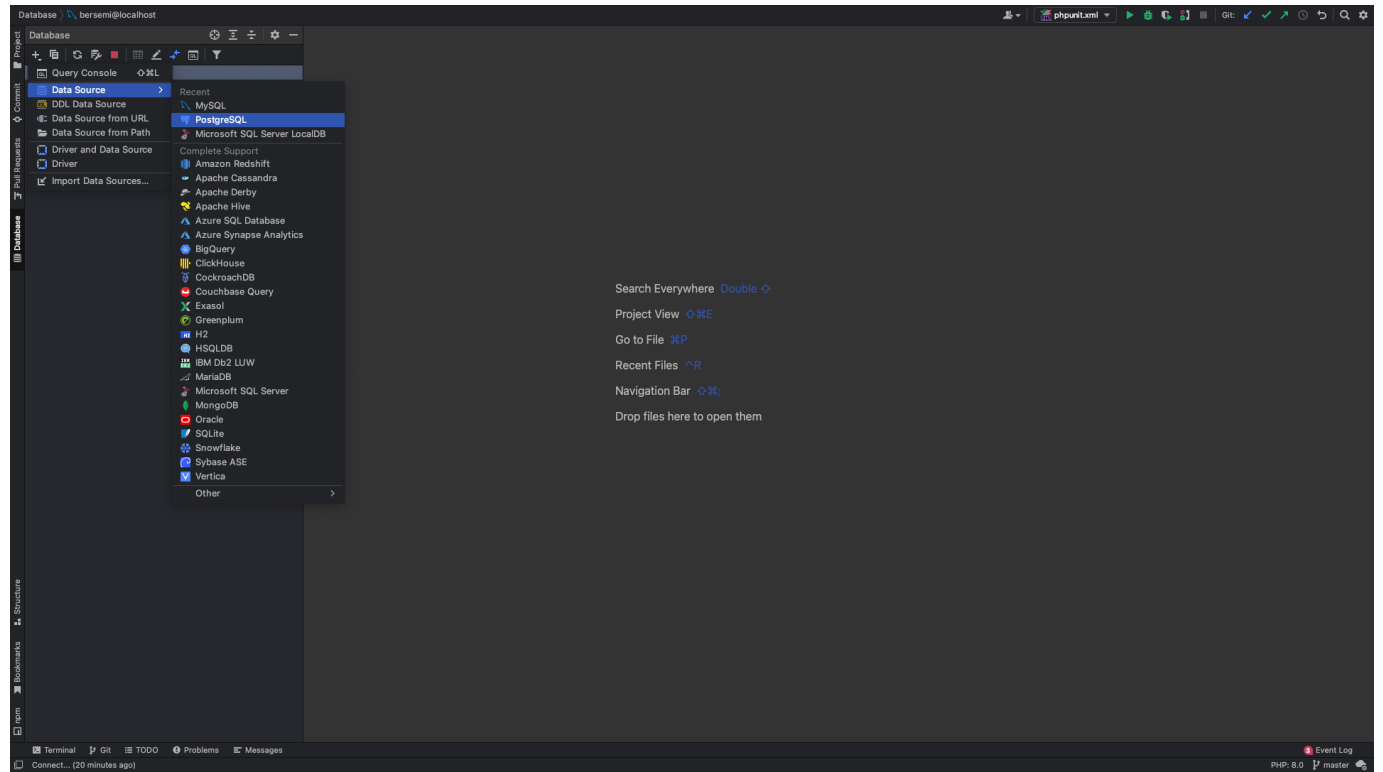
Di terminal, untuk menyambungkan koneksi dengan database bernama "risang" penulis menggunakan perintah `\c risang`.

A screenshot of a terminal window with a dark background. The window title bar shows three colored circles (red, yellow, green) and a keyboard icon. The terminal text shows a user prompt 'risangbaskoro=#' followed by the command '\c risang'. A message 'You are now connected to database "risang" as user "risangbaskoro".' is displayed. The prompt 'risang=#' is followed by a cursor. The bottom status bar shows '15%' on the left, '5.4 GB' in the center, and '412 kB↓' and '10 kB↑' on the right.

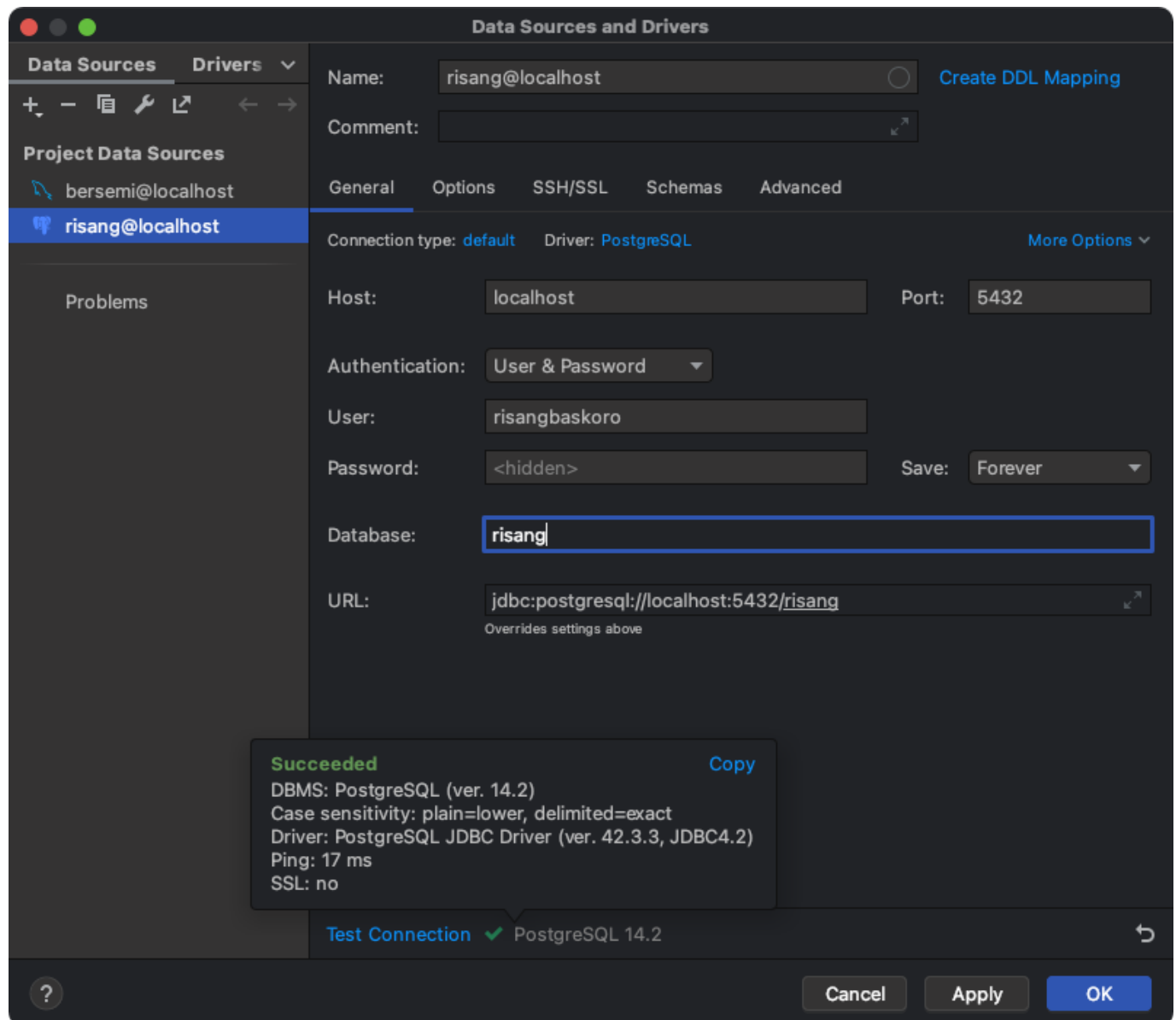
```
risangbaskoro=# \c risang
You are now connected to database "risang" as user "risangbaskoro".
risang=#
```

Koneksi dari Datagrip

Di Datagrip (atau produk JetBrains lainnya), penulis menggunakan tombol *new data source* di panel database.



Lalu memasukkan *credentials* table yang dituju:



Membuat Table

Penulis akan membuat 3 buah table dengan atribut sebagai berikut:

- users
 - id
 - name
 - email
 - password

- `created_at`
- `updated_at`
- `categories`
 - `id`
 - `title`
 - `slug`
 - `description`
- `posts`
 - `id`
 - `title`
 - `slug`
 - `body`
 - `user_id`
 - `category_id`
 - `created_at`
 - `updated_at`

Table `categories` dengan `posts` memiliki hubungan *one to many*, sebuah `category` memiliki banyak `post`, sementara tiap `post` memiliki satu `category`. Tiap *record* dari `user` juga dapat memiliki banyak *record* `post`, sementara `post` adalah milik `user`.

Query

Untuk mencapai hal tersebut, kode yang ditulis adalah sebagai berikut:

```
create table if not exists "users"
(
    "id"          serial primary key,
    "name"        varchar  not null,
    "email"       varchar  not null,
    "password"    varchar  not null,
    "created_at"  timestamp not null,
    "updated_at"  timestamp not null
);

create table if not exists "categories"
(
    "id"          serial primary key,
    "title"       varchar  not null,
    "slug"        varchar  not null,
    "description" text     not null
);

create table if not exists "posts"
(
    "id"          serial primary key,
    "title"       varchar  not null,
    "slug"        varchar  not null,
    "body"        text     not null,
    "user_id"     int references "users" ("id"),
    "category_id" int references "categories" ("id"),
    "created_at"  timestamp not null,
    "updated_at"  timestamp not null
);
```

Table Users

Untuk membuat table `users` tersebut, digunakan kode berikut:

```
create table if not exists "users"
(
    "id"          serial primary key,
    "name"        varchar  not null,
    "email"       varchar  not null,
    "password"    varchar  not null,
    "created_at"  timestamp not null,
    "updated_at"  timestamp not null
);
```

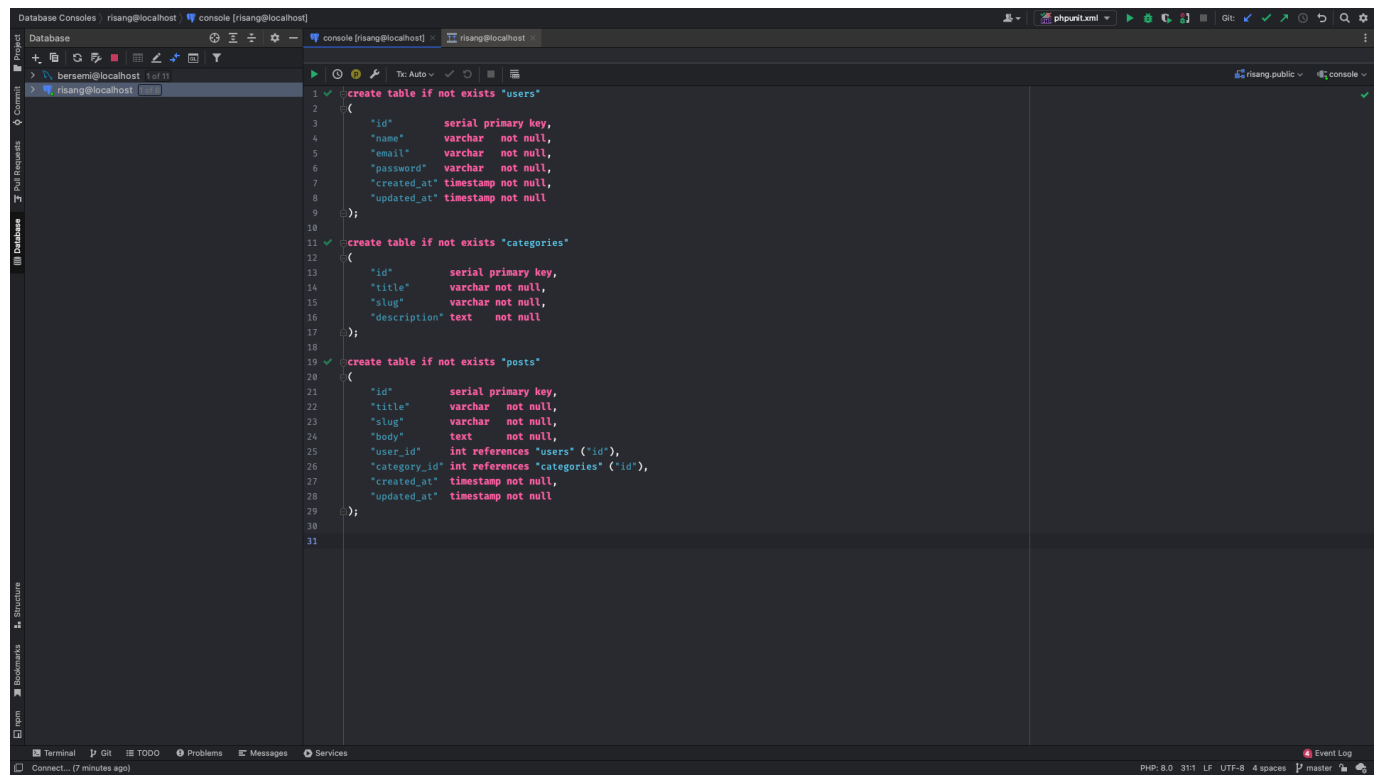
Table Categories

```
create table if not exists "categories"
(
    "id"          serial primary key,
    "title"       varchar  not null,
    "slug"        varchar  not null,
    "description" text      not null
);
```

Table Posts

```
create table if not exists "posts"
(
    "id"          serial primary key,
    "title"       varchar  not null,
    "slug"        varchar  not null,
    "body"        text      not null,
    "user_id"     int references "users" ("id"),
    "category_id" int references "categories" ("id"),
    "created_at"  timestamp not null,
    "updated_at"  timestamp not null
);
```

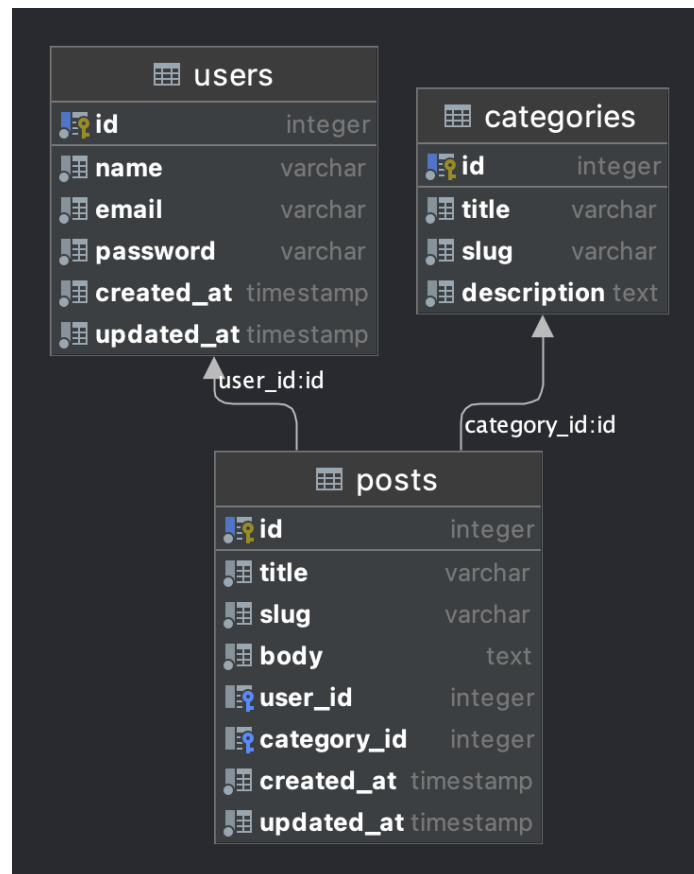
Screenshot



```
1 create table if not exists "users"
2 (
3     "id"          serial primary key,
4     "name"        varchar  not null,
5     "email"        varchar  not null,
6     "password"     varchar  not null,
7     "created_at"   timestamp not null,
8     "updated_at"   timestamp not null
9 );
10
11 create table if not exists "categories"
12 (
13     "id"          serial primary key,
14     "title"        varchar  not null,
15     "slug"         varchar  not null,
16     "description"   text     not null
17 );
18
19 create table if not exists "posts"
20 (
21     "id"          serial primary key,
22     "title"        varchar  not null,
23     "slug"         varchar  not null,
24     "body"         text     not null,
25     "user_id"      int references "users" ("id"),
26     "category_id"  int references "categories" ("id"),
27     "created_at"   timestamp not null,
28     "updated_at"   timestamp not null
29 );
30
31
```

Hasil

Dari praktik di atas, maka akan terbentuk database dengan struktur seperti berikut:



Kesimpulan

Dalam praktikum ini, penulis belajar menginstal PostgreSQL dan Datagrip, serta belajar membuat database dan table-table serta relasi di dalamnya.