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.

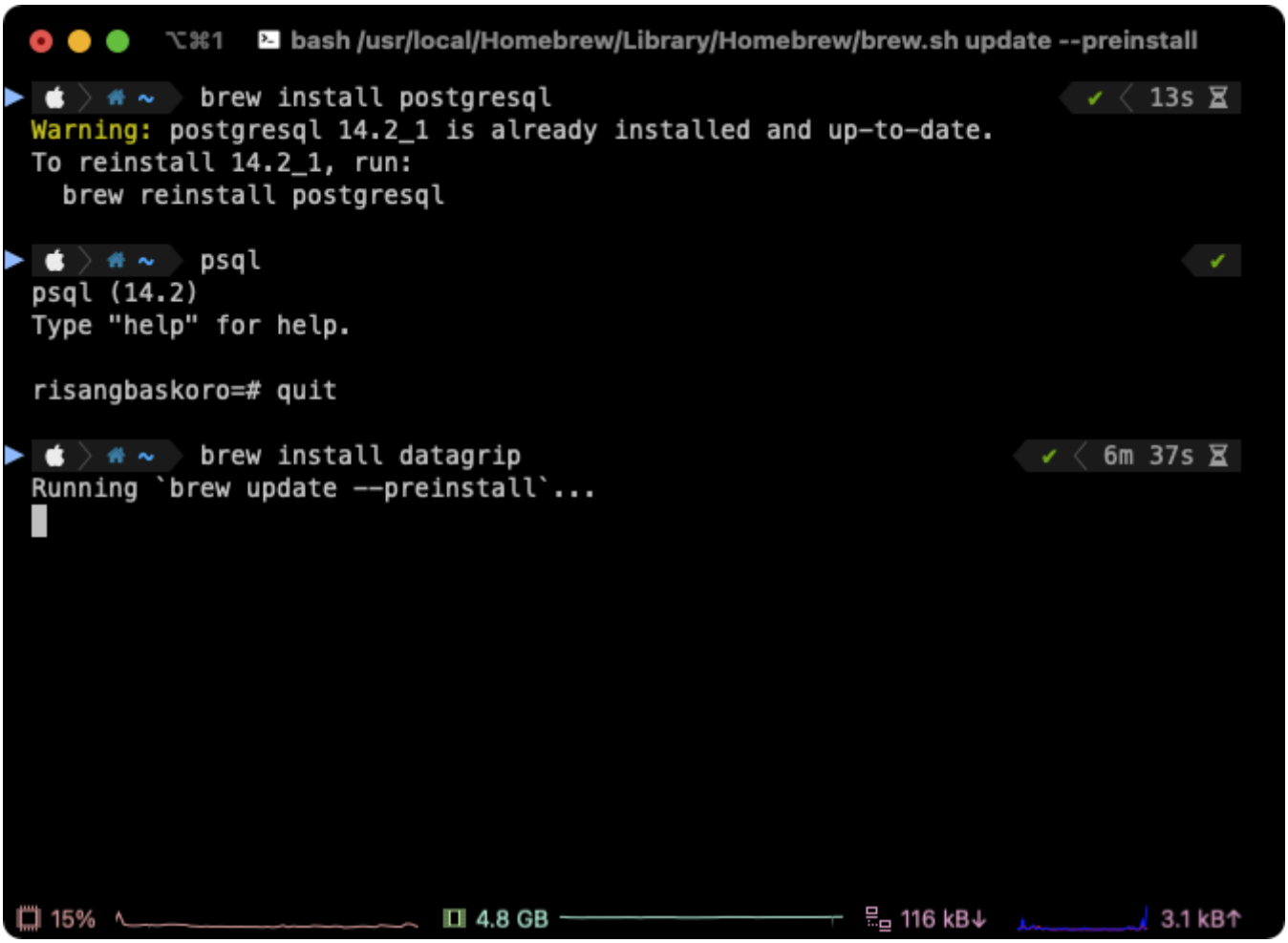
Penulis telah memiliki Homebrew, jadi penulis akan menggunakan terminal untuk menginstal Postgres. Instalasi menggunakan command `brew install postgresql`.

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

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

Install Datagrip

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



Datagrip sudah bisa dibuka dari Launchpad maupun Spotlight.

Membuat Database dan Table

Membuat Database

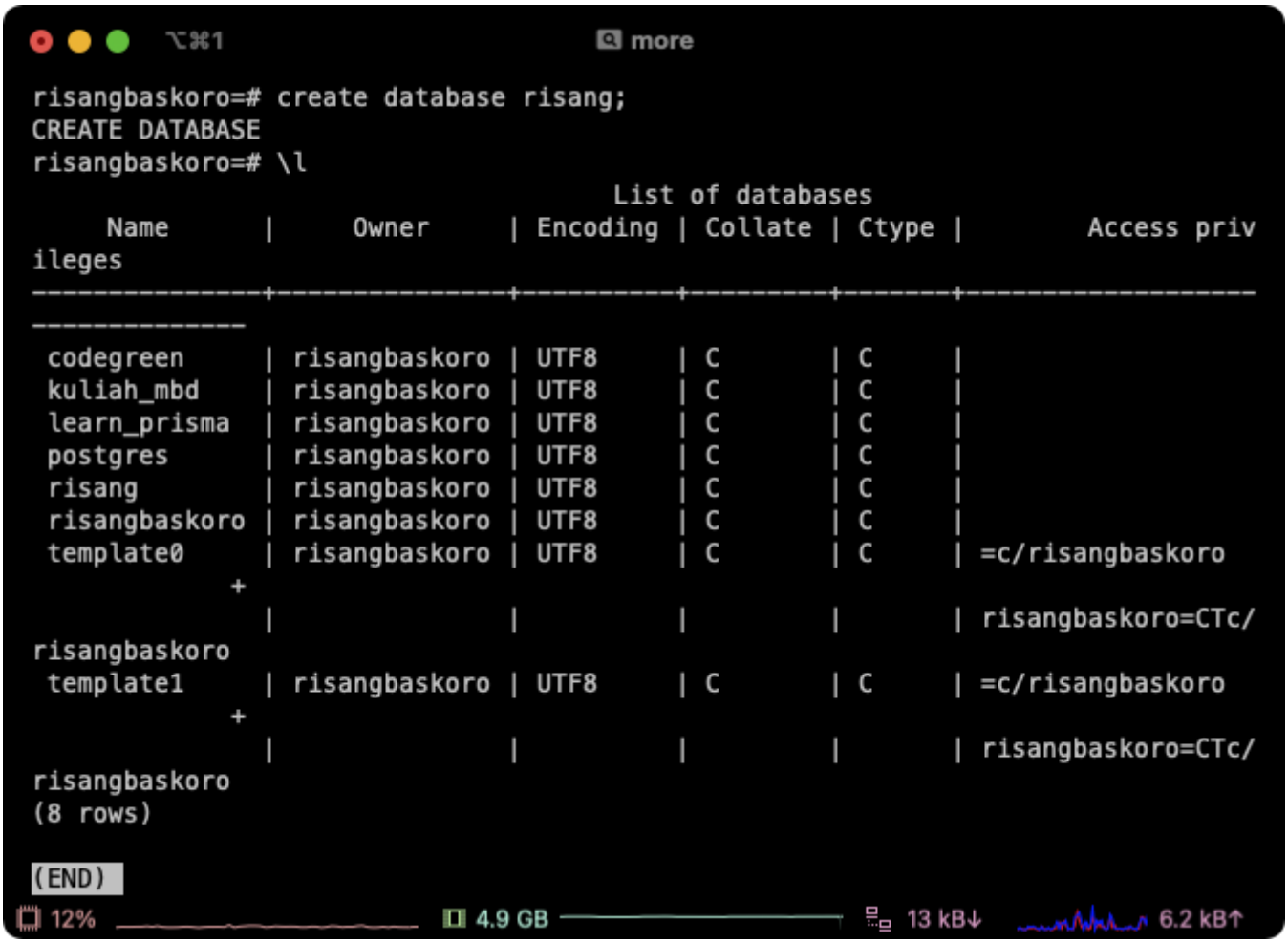
Saat ini, penulis pertama akan menggunakan terminal untuk menambahkan database.

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

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

```
create database risang;
```

Kemudian penulis melihat kembali database yang ada.

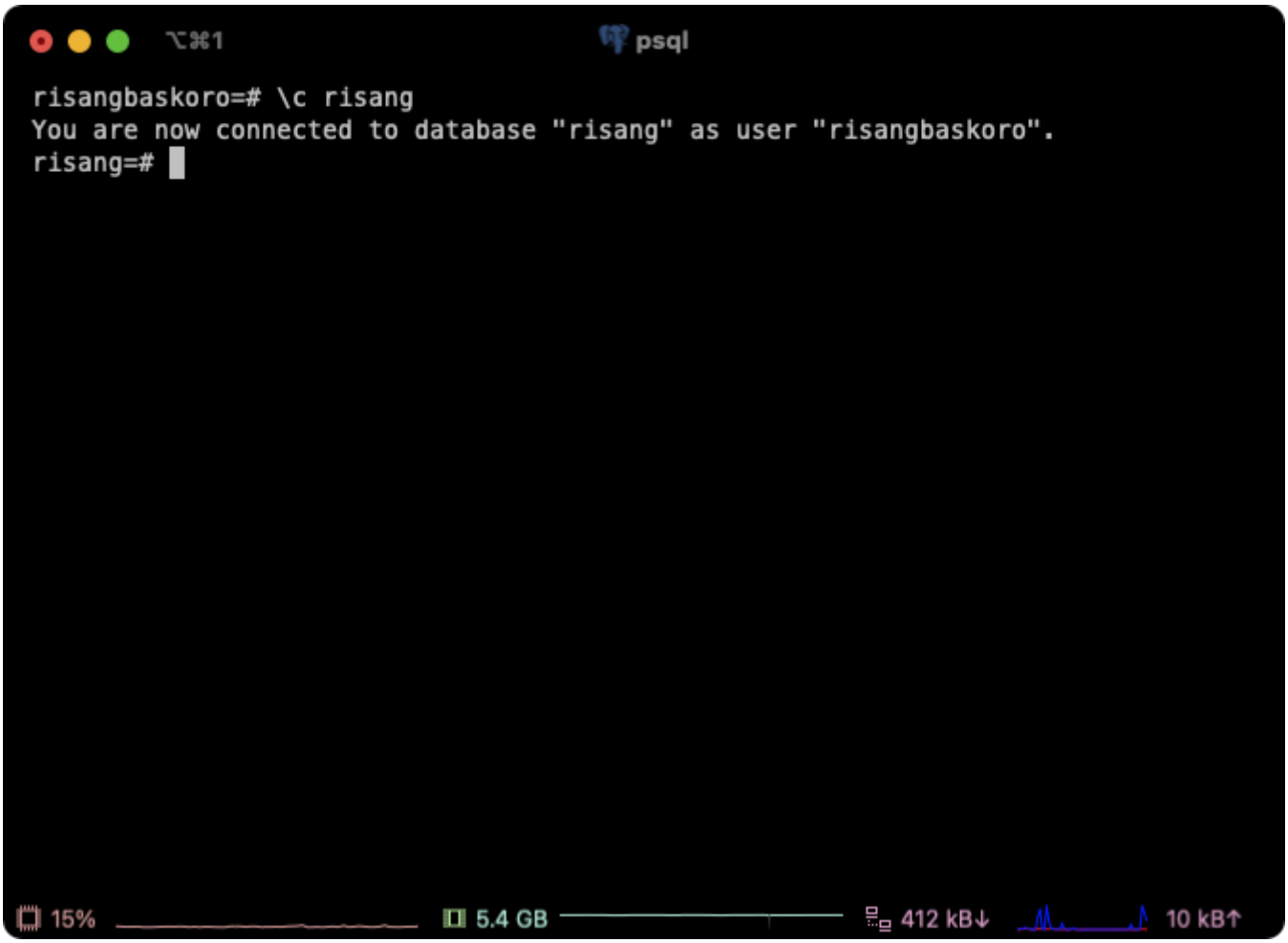


Koneksi Database

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

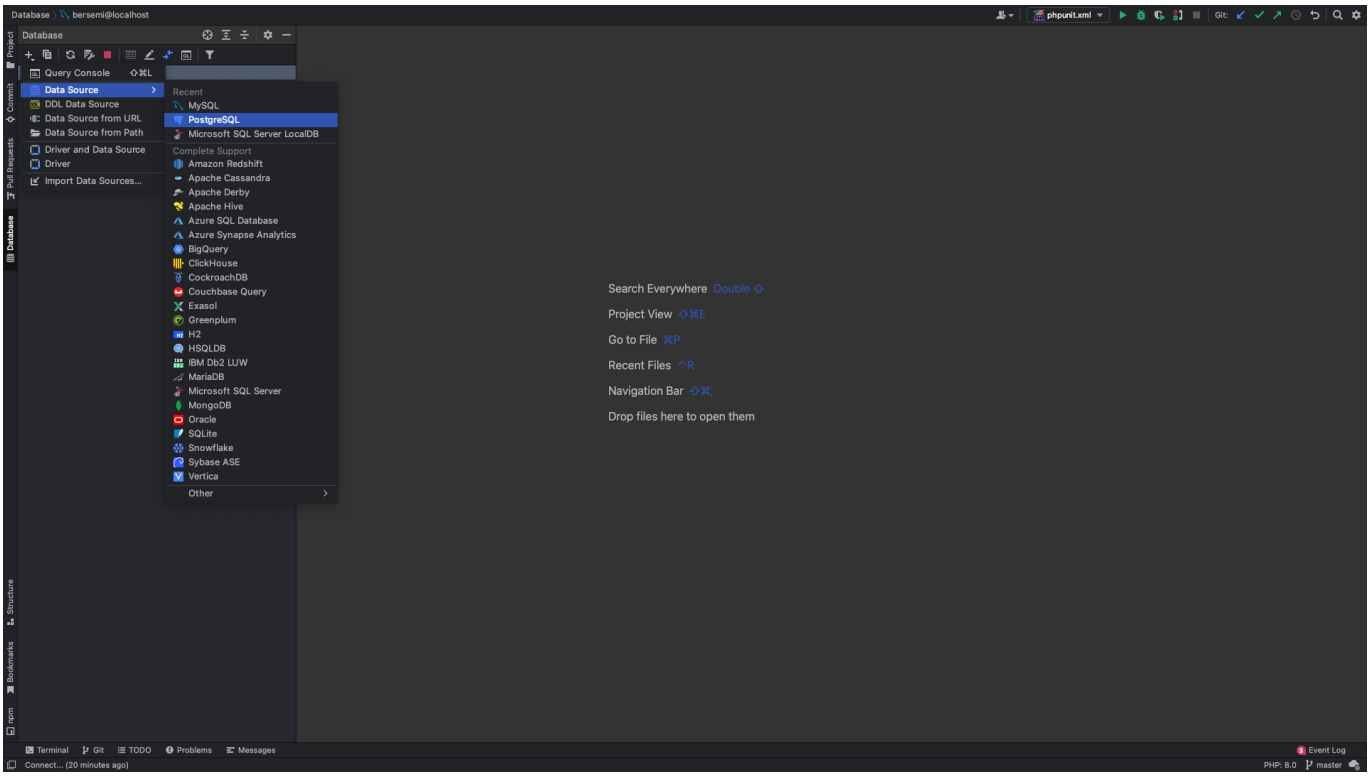
Koneksi dari Terminal

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

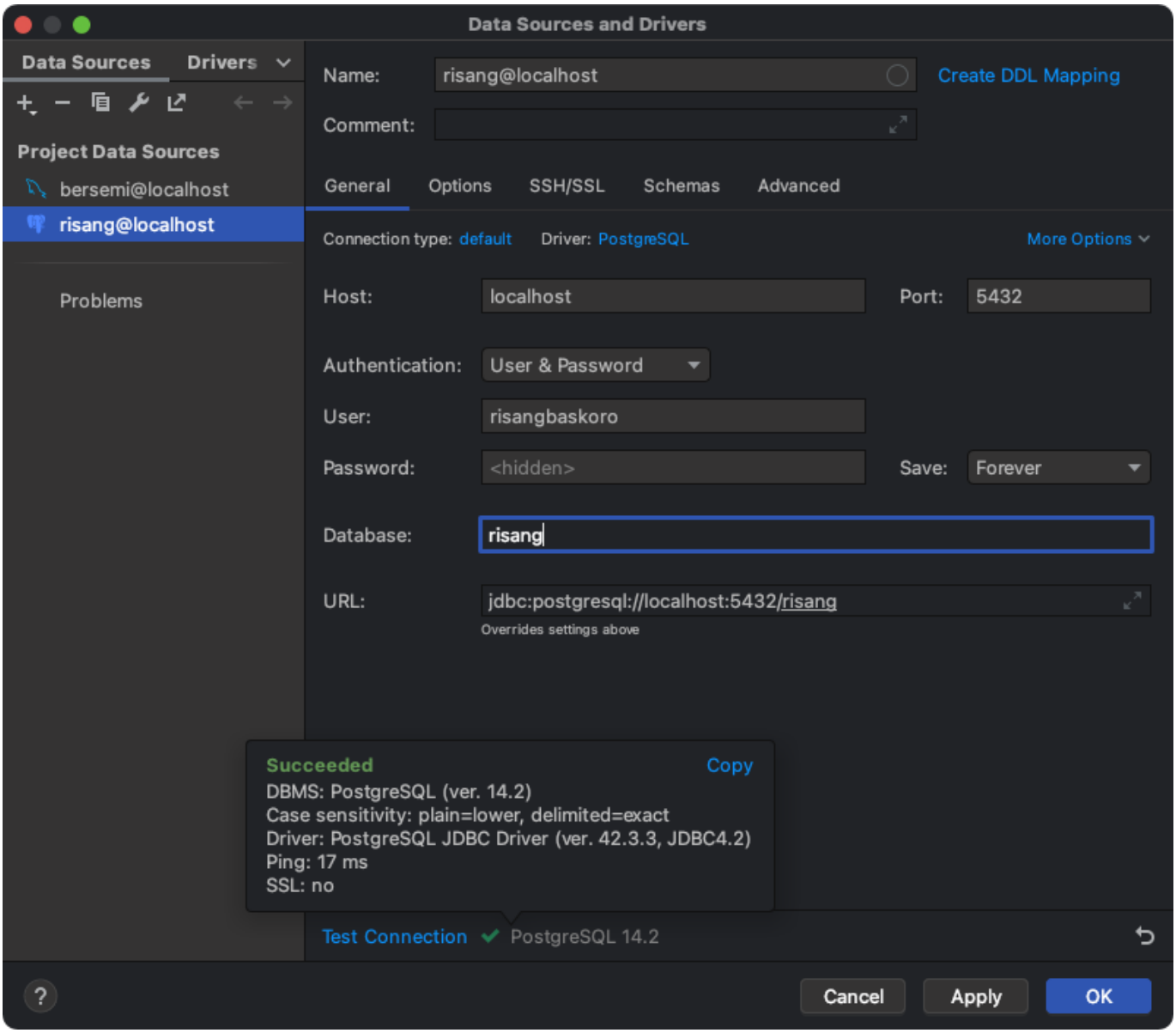


Koneksi dari Datagrip

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
);
```

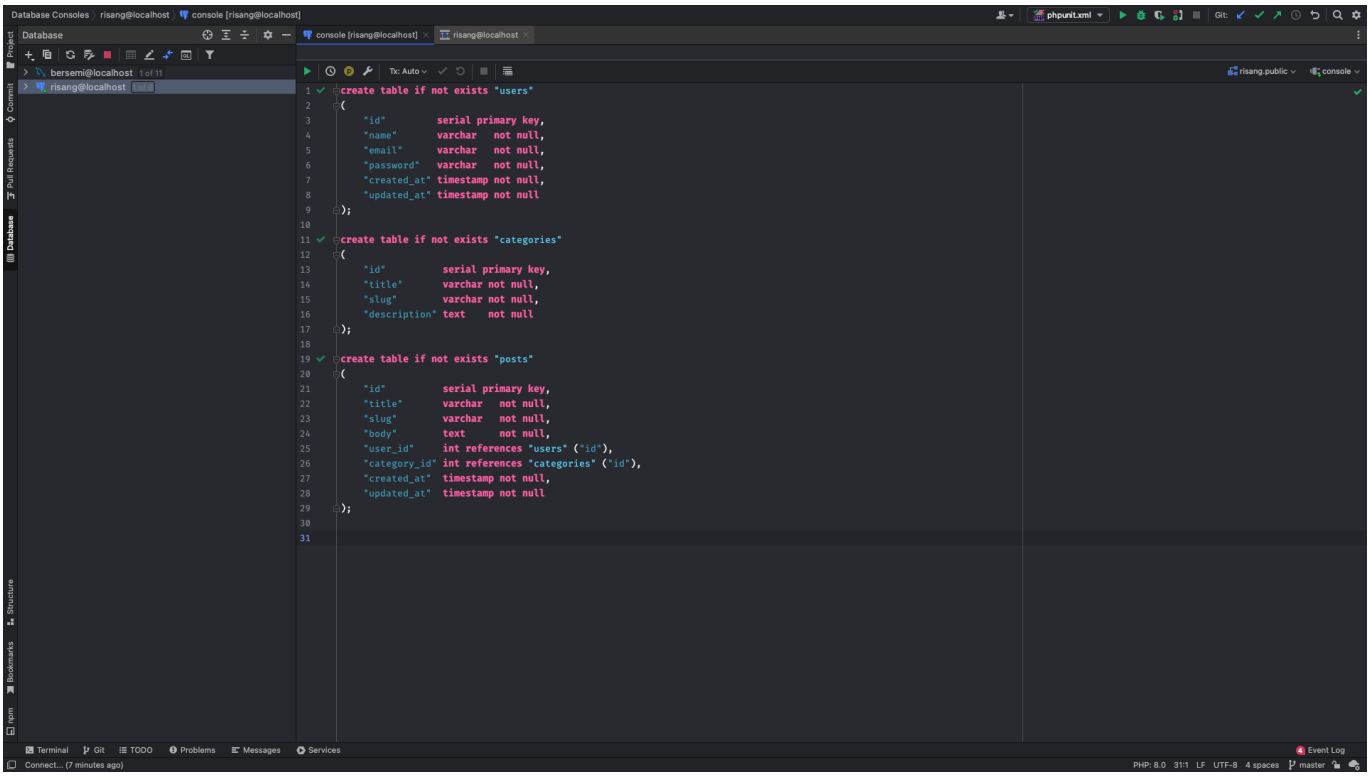
Table `Categories`

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

Table `Posts`

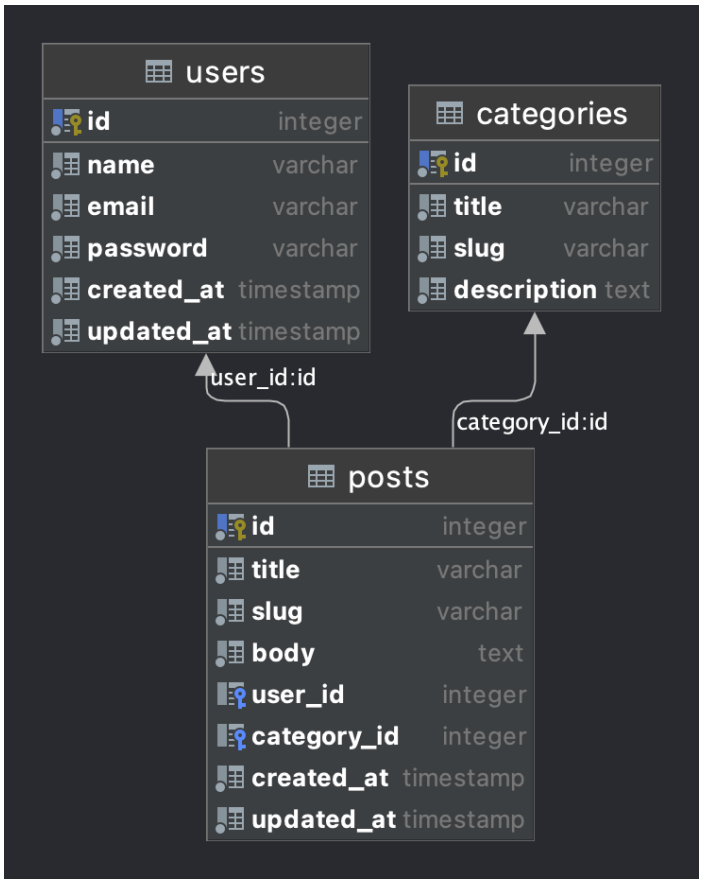
```
create table if not exists "posts"
(
    "id"          serial primary key,
    "title"       varchar   not null,
    "slug"        varchar   not null,
    "body"        text      not null,
    "user_id"     int references "users" ("id"),
    "category_id" int references "categories" ("id"),
    "created_at"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

Screenshot



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.