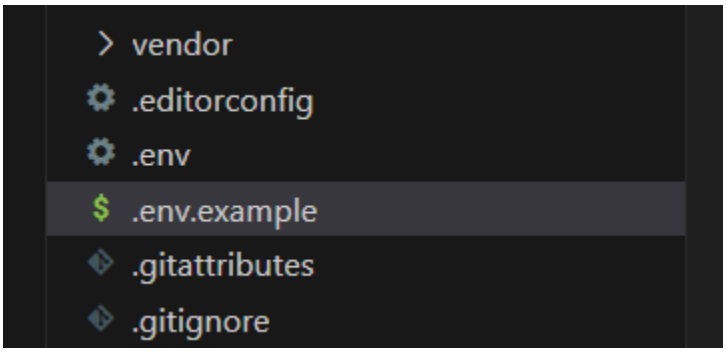
ENV, MIGRATION DAN OTENTIKASI PADA LARAVEL 11

File ENV pada laravel

File .env di Laravel adalah file yang digunakan untuk menyimpan konfigurasi lingkungan aplikasi. Konfigurasi ini biasanya berbeda antara lingkungan pengembangan, staging, dan produksi. File ini juga berfungsi untuk menyimpan informasi sensitif seperti kredensial basis data, kunci API, dan pengaturan lainnya, file .env berada pada folder root aplikasi laravel, jika saat pertama kali menginstall laravel, file .env ini belum tersedia, maka anda dapat mengcopy dari file .env.example dan mengubah namanya menjadi .env perhatikan gambar berikut.



Berikut adalah beberapa konfigurasi penting yang biasanya ada di dalam file .env Laravel:

Penjelasan Konfigurasi .env

1. APP Configuration
    - APP_NAME - Nama aplikasi Laravel anda.
    - APP_ENV - Lingkungan aplikasi, bisa diisi dengan local, staging, atau production.
    - APP_KEY - Kunci enkripsi aplikasi Laravel. Ini penting untuk enkripsi session, cookies, dan data lainnya. Kunci ini dihasilkan secara otomatis saat kamu menjalankan php artisan key:generate.
    - APP_DEBUG - Jika disetel ke true, aplikasi akan menampilkan pesan debug dan stack trace saat terjadi error. Pada lingkungan produksi, nilai ini sebaiknya disetel ke false.
    - APP_URL - URL dasar aplikasi anda, misalnya http://localhost untuk pengembangan lokal.
2. Database Configuration

    - DB_CONNECTION - Jenis koneksi basis data yang digunakan, seperti mysql, pgsql, sqlite, atau sqlsrv.
    - DB_HOST - Alamat host server basis data, biasanya 127.0.0.1 atau localhost untuk pengembangan lokal.
    - DB_PORT - Port yang digunakan oleh server basis data, default untuk MySQL adalah 3306.
    - DB_DATABASE - Nama basis data yang digunakan oleh aplikasi.
    - DB_USERNAME - Nama pengguna untuk koneksi ke basis data.
    - DB_PASSWORD - Kata sandi pengguna basis data.
3. Mail Configuration
    - MAIL_MAILER - Driver yang digunakan untuk pengiriman email. Defaultnya smtp. Bisa juga diisi dengan sendmail, mailgun, ses, dll.
    - MAIL_HOST - Host server email. Jika menggunakan Mailtrap untuk pengujian, maka biasanya smtp.mailtrap.io.
    - MAIL_PORT - Port untuk pengiriman email, biasanya 2525 untuk Mailtrap atau 587 untuk Gmail.
    - MAIL_USERNAME dan MAIL_PASSWORD - Kredensial untuk otentikasi ke server email.
    - MAIL_ENCRYPTION - Metode enkripsi yang digunakan saat mengirim email, misalnya tls.
4. Cache and Session
    - CACHE_DRIVER - Driver untuk menyimpan cache. Pilihan umum termasuk file, redis, memcached, dll.
    - SESSION_DRIVER - Driver yang digunakan untuk menyimpan sesi pengguna. Pilihan umum termasuk file, cookie, database, redis, dll.
    - SESSION_LIFETIME - Waktu kadaluarsa sesi dalam menit.
5. Redis Configuration
    - REDIS_HOST - Alamat host server Redis, biasanya 127.0.0.1 untuk lokal.
    - REDIS_PASSWORD - Kata sandi untuk otentikasi ke Redis, jika diperlukan.
    - REDIS_PORT - Port yang digunakan oleh Redis, defaultnya adalah 6379.
6. Broadcasting, Queue, and Cache
    - BROADCAST_DRIVER - Driver untuk broadcasting, misalnya log, redis, pusher, dll.
    - QUEUE_CONNECTION - Driver untuk antrian pekerjaan (queue), misalnya sync (sinkron) atau database (asinkron).
    - CACHE_DRIVER - Driver yang digunakan untuk cache, misalnya file, redis, memcached.
7. AWS (Optional) Jika kamu menggunakan layanan AWS untuk penyimpanan, seperti S3, tambahkan kredensial AWS di sini:

    - AWS_ACCESS_KEY_ID - Kunci akses AWS kamu.
    - AWS_SECRET_ACCESS_KEY - Kunci rahasia AWS.
    - AWS_DEFAULT_REGION - Wilayah AWS tempat bucket berada.
    - AWS_BUCKET - Nama bucket AWS S3.
8. Pusher Configuration (Optional) Pusher digunakan untuk notifikasi real-time:
    - PUSHER_APP_ID, PUSHER_APP_KEY, PUSHER_APP_SECRET, dan PUSHER_APP_CLUSTER digunakan untuk mengkonfigurasi layanan Pusher.

Mengubah Nilai .env

Untuk mengubah nilai di .env, cukup buka file tersebut dan ubah sesuai dengan konfigurasi yang diinginkan. Setelah melakukan perubahan, jalankan perintah berikut agar aplikasi Laravel bisa memuat ulang perubahan:

```
php artisan config:cache
```

Ini akan membersihkan cache konfigurasi yang mungkin masih memuat nilai lama dan memastikan Laravel memuat konfigurasi baru dari .env.

Menjaga Keamanan .env

File .env sebaiknya tidak pernah di-commit ke sistem kontrol versi seperti Git, terutama jika berisi informasi sensitif seperti kata sandi dan kunci API. Laravel secara otomatis menambahkan .env ke file .gitignore untuk mencegah hal ini, tapi anda tetap harus berhati-hati.

Percobaan satu mengkonfigurasi koneksi database

Silahkan ubah konfigurasi file .env anda pada bagian database sesuai dengan konfigurasi database anda, untuk penamaan database kita samakan menjadi db-sistem-toko silahkan perhatikan gambar berikut

.env sebelum diubah	.env sesudah diubah
22 DB_CONNECTION=sqlite 23 # DB_HOST=127.0.0.1 24 # DB_PORT=3306 25 # DB_DATABASE=laravel 26 # DB_USERNAME=root 27 # DB_PASSWORD=	22 DB_CONNECTION=mysql 23 DB_HOST=127.0.0.1 24 DB_PORT=3306 25 DB_DATABASE=db-sistem-toko 26 DB_USERNAME=root 27 DB_PASSWORD=

Migration pada Laravel

Migration di Laravel memudahkan pengelolaan skema database dengan cara yang terstruktur dan memungkinkan versioning. Ini sangat berguna dalam pengembangan aplikasi yang melibatkan banyak developer dan lingkungan yang berbeda. Dengan migration, perubahan pada skema database dapat dilakukan dengan aman dan mudah diatur. Migration di Laravel memiliki sebuah fitur yang memungkinkan kita untuk mengelola struktur basis data secara terstruktur dan versioning. Migration menyediakan cara yang mudah untuk membuat, memodifikasi, dan menghapus tabel serta kolom di database tanpa harus menulis query SQL secara langsung. Dengan migration, kita dapat mengontrol perubahan skema database dalam file kode dan melacak setiap perubahan yang terjadi dari waktu ke waktu. Migration sangat berguna dalam pengembangan tim karena memungkinkan setiap anggota tim untuk memperbarui struktur basis data dengan menjalankan perintah migration yang sama.

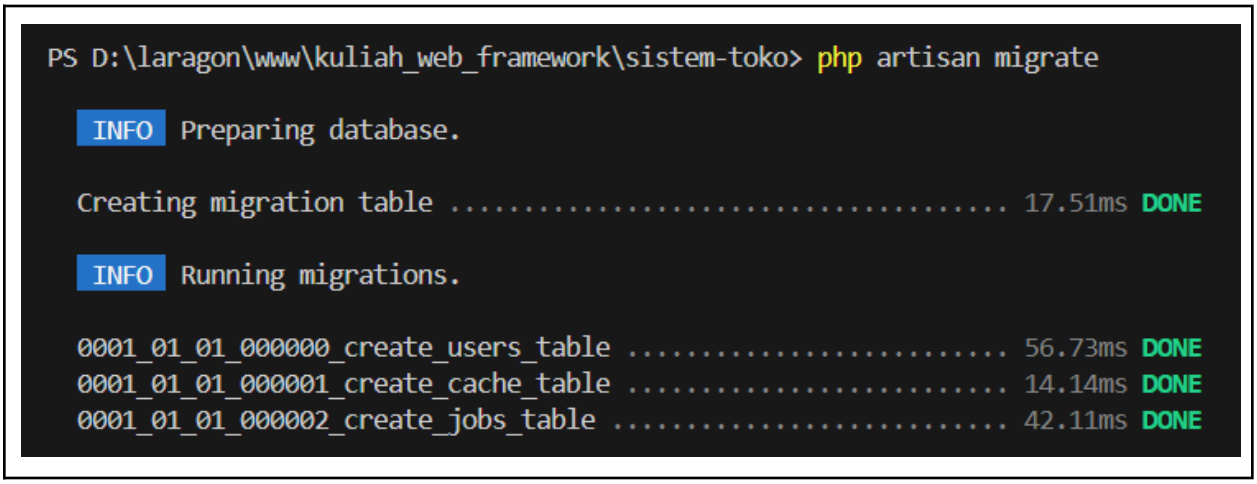
Kelebihan Menggunakan Migration:

1. Sebagai version kontrol dapat melacak perubahan pada struktur database seiring waktu.
2. Mempermudah kolaborasi antar developer dalam satu proyek karena setiap perubahan dapat dicatat dan di-share.
3. Sangat berguna untuk mengembangkan skema database secara otomatis di lingkungan yang berbeda, seperti local development, staging, dan production.

File migration ada di dalam direktori database/migrations/ saat pertama kali menginstall laravel 11 akan ada beberapa file migration yang telah tersedia, jika anda cek database yang sebelumnya telah dibuat, maka belum terdapat tabel apapun, untuk mengeksekusi file migration yang tersedia sebelumnya maka anda bisa menjalankan perintah

```
php artisan migrate
```

Hasil eksekusi perintah diatas seperti berikut :



Maka di dalam database kita akan terdapat tabel baru sesuai dengan definisi yang ada di dalam direktori migrations.

Untuk melihat jenis-jenis tipe data yang digunakan, anda dapat melihat di https://laravel.com/docs/11.x/migrations#available-column-types

Membuat File Migration di Laravel

Jika pada sebelumnya anda telah belajar pemrograman web native (tanpa framework) anda akan membuat tabel database maka biasanya akan langsung ke phpmyadmin atau CLI, jika saat ini kita menggunakan laravel maka kita akan mendefinisikan SQL DDL pada file migration yang kita bahas saat ini.

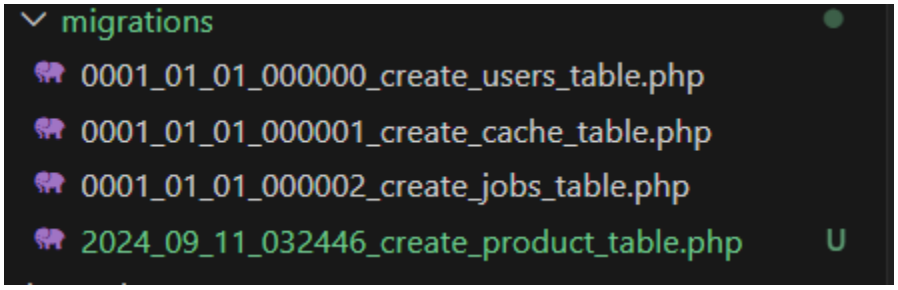
Untuk membuat migration baru di Laravel, anda bisa menggunakan perintah Artisan berikut :

```
php artisan make:migration create_product_table
```

Perintah di atas akan membuat file migration baru di direktori database/migrations. Setelah migration dibuat, anda akan melihat file yang berisi dua metode utama:

1. up(): Metode ini digunakan untuk mendefinisikan perubahan yang ingin kamu lakukan pada database (misalnya membuat tabel atau menambahkan kolom).

2. down(): Metode ini digunakan untuk membatalkan perubahan yang dilakukan oleh metode up() (misalnya menghapus tabel atau kolom yang telah ditambahkan).



Percobaan kedua membuat tabel product

Untuk membuat sebuah tabel database kita akan mengisi fungsi up( ) pada file migration yang create_product_table yang sebelumnya telah dibuat, silahkan perhatikan gambar berikut

```
15          Schema::create('product', function (Blueprint $table) {
16              $table->id();
17
18              $table->string("product_name");
19              $table->string("unit");
20              $table->string("type");
21              $table->string("information");
22              $table->integer("qty");
23
24              $table->timestamps();
25          });
```

Anda perlu menambahkan kode dari baris 18 sampai dengan 22 pada gambar diatas untuk membuat struktur tabel product, kemudian eksekusi file tersebut dengan perintah php artisan migrate Jika anda ingin membatalkan migration yang telah dijalankan (membatalkan tabel database), kamu dapat menggunakan perintah

```
php artisan migrate:rollback
```

Ini akan membatalkan (rollback) perubahan yang dibuat oleh migration terakhir, secara algoritma sebetulnya perintah rollback mengeksekusi sintak yang berada di fungsi down(). Seperti berikut

```
28      /**
29       * Reverse the migrations.
30       */
31      public function down(): void
32      {
33          Schema::dropIfExists('product');
34      }
```

Otentikasi dengan Laravel Breeze

Untuk membuat fitur otentikasi secara cepat, Laravel menyediakan package bernama Laravel Breeze. Laravel Breeze menyederhanakan proses pembuatan login, register, dan fitur otentikasi lainnya. Eksekusi perintah dibawah ini untuk menginstal Laravel Breeze

```
composer require laravel/breeze --dev
```

Perintah berikutnya untuk menginstall package

```
php artisan breeze:install
```

Jika diminta untuk memilih antara blade, vue dll maka anda dapat memilih blade, dengan catatan sebelumnya anda telah menginstall node.js dan npm atau mendownload langsung yang sudah disertakan npm nya pada link https://nodejs.org/en/download/prebuilt-installer kemudian setelah menginstal anda diharuskan melakukan restart pada visual studio code atau pada terminal yang digunakan. Kemudian Jika anda diminta untuk memilih testing framework, bisa dipilih opsi 0 PhpUnit.

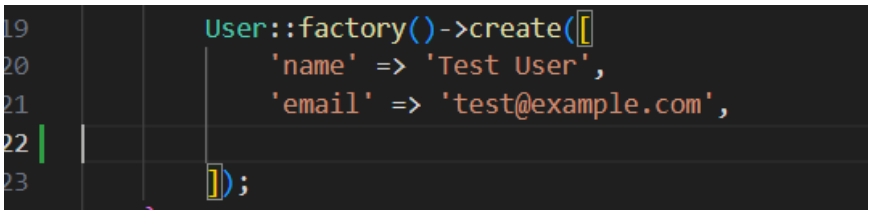
Selanjutnya kita perlu menginstall frontend dan migrasi dari laravel breeze

```
npm install
```

```
npm run dev
```

```
php artisan migrate
```

Selanjutnya untuk menambahkan data pengguna, anda perlu mengeksekusi file seeder yang berada di direktori database/seeders/DatabaseSeeder.php yang berisi perintah untuk membuat user demo, tetapi pastikan perintahnya tidak dikomentar, seperti gambar berikut

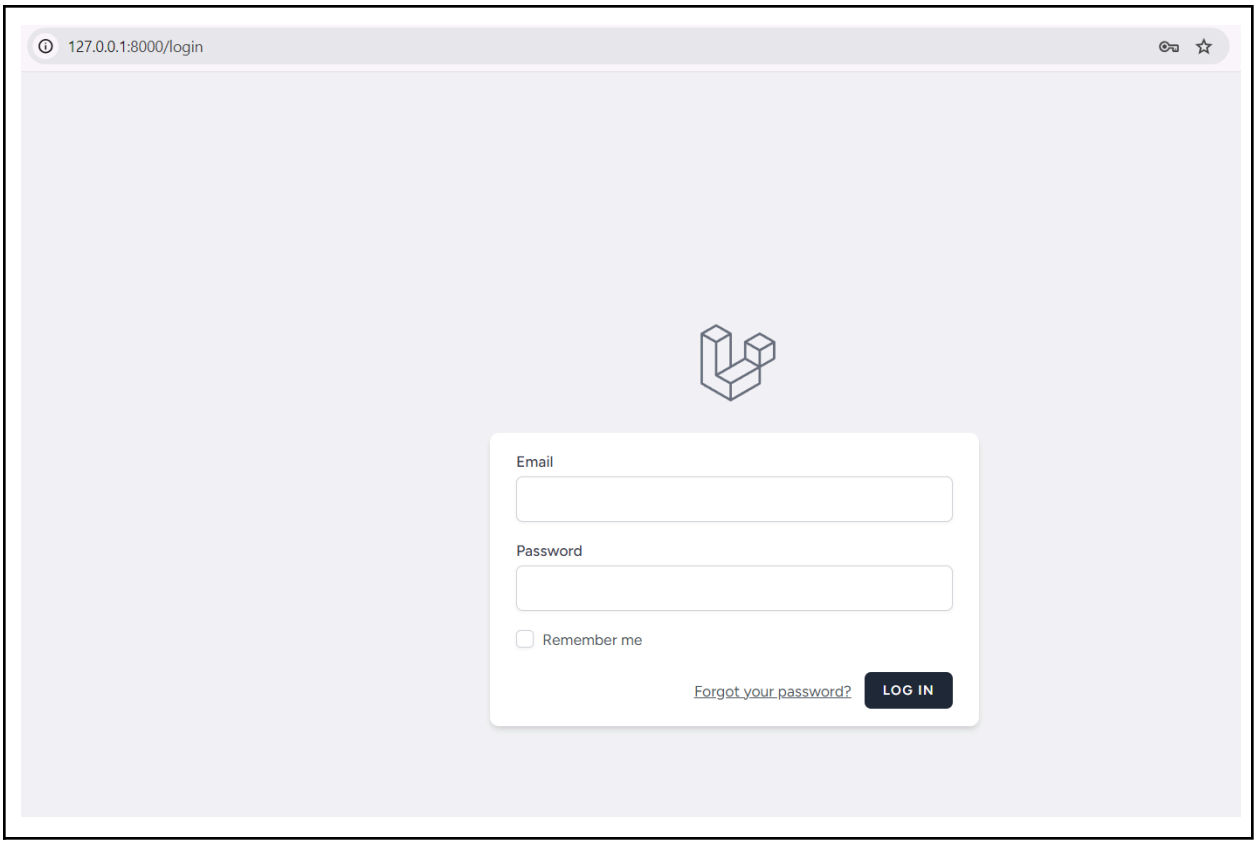


dengan perintah untuk membuat user demo seperti berikut

```
php artisan db:seed
```

Package breeze ini akan menambahkan route auth.php, controller Auth, Request Auth dan View auth, Anda dapat menyesuaikan tampilan halaman login dan register yang disediakan oleh Breeze dengan mengedit file blade di resources/views/auth

Dengan menjalankan perintah diatas akan menambahkan data pada tabel users dengan password default "password". Kemudian anda dapat mencoba untuk login.


127.0.0.1:8000/login
Email
Password
Remember me
Forgot your password?
LOG IN