

MIGRATION

1.1 Kompetensi Utama

- Diharapkan mahasiswa dapat memahami migration pada laravel.

1.2 Kompetensi Penunjang

- Mengenalkan kepada mahasiswa tentang migration
- Membuat migration pada laravel

1.3 Dasar Teori

1. Konfigurasi Database

Pengaksesan database menjadi salah satu fitur terpenting dari setiap *web application*. Dan konsep MVC sendiri sudah mengakomodasi komponen bernama **Model** untuk memprosesnya. Laravel menyediakan cukup banyak fitur pengelolaan database seperti *migration*, *DB facade*, *query builder*, serta *eloquent* Untuk aplikasi web berbasis PHP, database yang paling sering dipakai adalah **MySQL/MariaDB**, **SQLite**, **PostgreSQL** serta **SQL Server**.

File Konfigurasi .env dan config/database.php

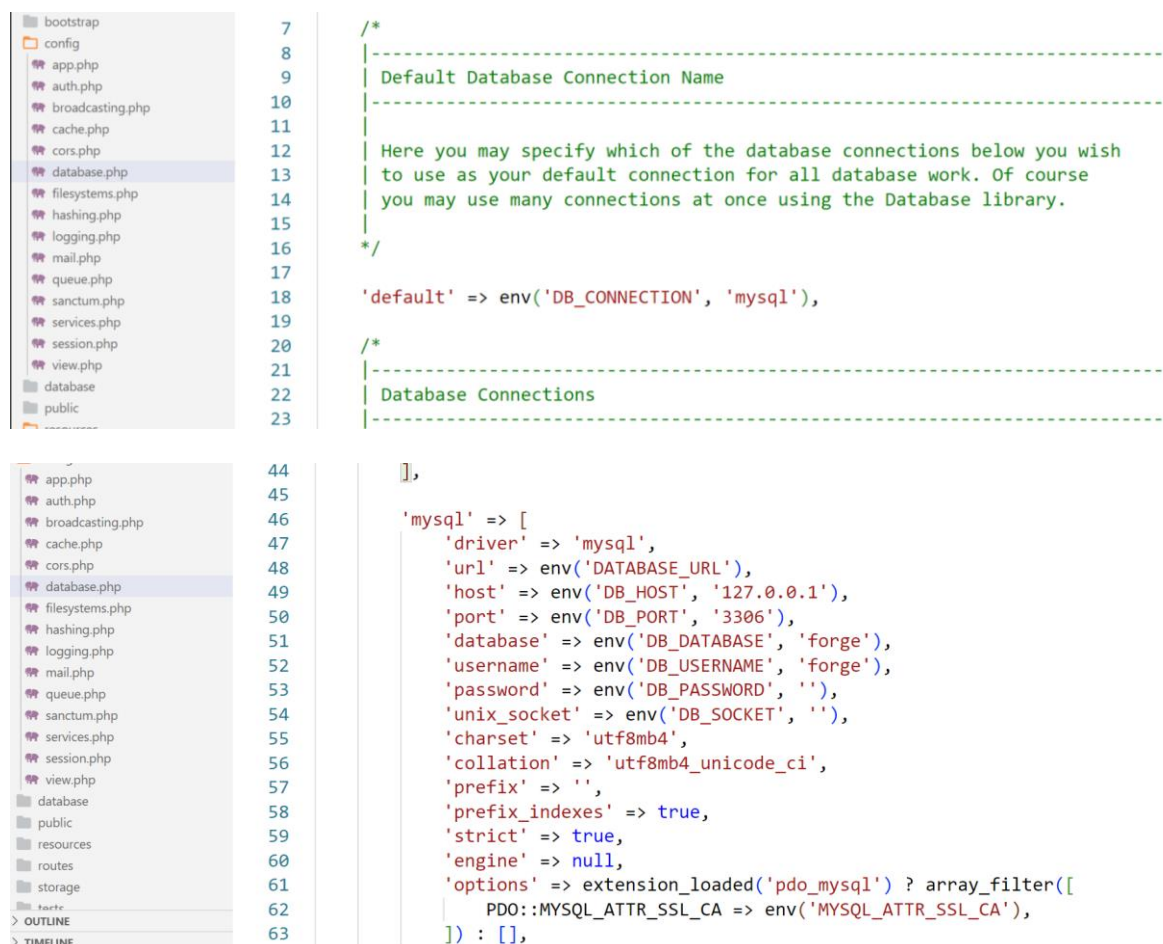
Laravel memiliki 2 lokasi pengaturan: konfigurasi lokal di file **.env** dan pengaturan global di folder **config**. Kita akan lihat file **.env** terlebih dahulu. Silahkan buka file **.env** dan cari baris berikut:

```
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:lQ4eXWXXKXvZ6dA/WEGjTG+qQFkXK615raON+hegJeM=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATIONS_CHANNEL=null
9 LOG_LEVEL=debug
10
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=app_lara10
15 DB_USERNAME=root
16 DB_PASSWORD=
17
18 BROADCAST_DRIVER=log
19 CACHE_DRIVER=file
```

Terdapat 6 baris pengaturan terkait database:

- **DB_CONNECTION:** Meskipun secara bawaan berisi 'mysql', ini bukanlah pengaturan tentang jenis database, tapi lebih ke *profile* yang berisi kumpulan konfigurasi ke database MySQL.
- **DB_HOST:** Dipakai untuk mengatur alamat server database. Secara bawaan sudah berisi 127.0.0.1, yakni alamat localhost.
- **DB_PORT:** Dipakai untuk mengatur port dari aplikasi server database. Secara bawaan berisi angka 3306, yakni port default bawaan MySQL.
- **DB_DATABASE:** Nama database yang akan dipakai. Pada konfigurasi di atas berisi **app_lara10**, yang artinya Laravel akan langsung mencari sebuah database bernama 'app_lara10'.
- **DB_USERNAME** dan **DB_PASSWORD:** Dipakai untuk membuat nama user dan password login ke database server. Secara bawaan nama user adalah root, dan password tidak diisi.

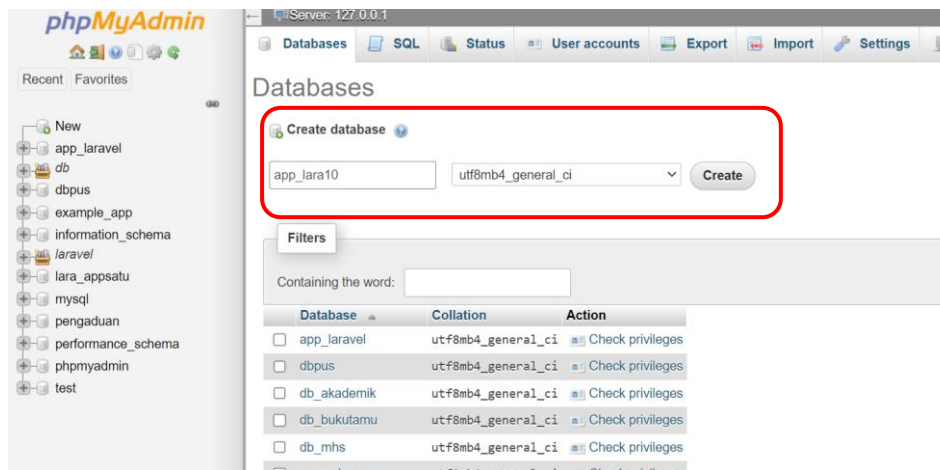
Sekarang mari kita lihat file konfigurasi kedua, yakni di config/database.php:



```
7  /*
8  |-----
9  | Default Database Connection Name
10 |-----
11 |
12 | Here you may specify which of the database connections below you wish
13 | to use as your default connection for all database work. Of course
14 | you may use many connections at once using the Database library.
15 |
16 */
17
18 'default' => env('DB_CONNECTION', 'mysql'),
19
20
21 /*
22 |-----
23 | Database Connections
24 |-----
25 |
26 | Here you may configure all of the database connections in your application.
27 | Each item must have a unique name and a configuration array.
28 |
29 | You may use the Database library to manage the connections in your application
30 | using the Database library.
31 |
32 | For more information on the Database library, visit the documentation:
33 | https://laravel.com/docs/database
34 |
35 | For more information on the Database library, visit the documentation:
36 | https://laravel.com/docs/database
37 |
38 | For more information on the Database library, visit the documentation:
39 | https://laravel.com/docs/database
40 |
41 | For more information on the Database library, visit the documentation:
42 | https://laravel.com/docs/database
43 |
44 |
45 |
46 |
47 |
48 |
49 |
50 |
51 |
52 |
53 |
54 |
55 |
56 |
57 |
58 |
59 |
60 |
61 |
62 |
63 |
64 |
65 |
66 |
67 |
68 |
69 |
70 |
71 |
72 |
73 |
74 |
75 |
76 |
77 |
78 |
79 |
80 |
81 |
82 |
83 |
84 |
85 |
86 |
87 |
88 |
89 |
90 |
91 |
92 |
93 |
94 |
95 |
96 |
97 |
98 |
99 |
100|
101|
102|
103|
104|
105|
106|
107|
108|
109|
110|
111|
112|
113|
114|
115|
116|
117|
118|
119|
120|
121|
122|
123|
124|
125|
126|
127|
128|
129|
130|
131|
132|
133|
134|
135|
136|
137|
138|
139|
140|
141|
142|
143|
144|
145|
146|
147|
148|
149|
150|
151|
152|
153|
154|
155|
156|
157|
158|
159|
160|
161|
162|
163|
164|
165|
166|
167|
168|
169|
170|
171|
172|
173|
174|
175|
176|
177|
178|
179|
180|
181|
182|
183|
184|
185|
186|
187|
188|
189|
190|
191|
192|
193|
194|
195|
196|
197|
198|
199|
200|
201|
202|
203|
204|
205|
206|
207|
208|
209|
210|
211|
212|
213|
214|
215|
216|
217|
218|
219|
220|
221|
222|
223|
224|
225|
226|
227|
228|
229|
230|
231|
232|
233|
234|
235|
236|
237|
238|
239|
240|
241|
242|
243|
244|
245|
246|
247|
248|
249|
250|
251|
252|
253|
254|
255|
256|
257|
258|
259|
260|
261|
262|
263|
264|
265|
266|
267|
268|
269|
270|
271|
272|
273|
274|
275|
276|
277|
278|
279|
280|
281|
282|
283|
284|
285|
286|
287|
288|
289|
290|
291|
292|
293|
294|
295|
296|
297|
298|
299|
300|
301|
302|
303|
304|
305|
306|
307|
308|
309|
310|
311|
312|
313|
314|
315|
316|
317|
318|
319|
320|
321|
322|
323|
324|
325|
326|
327|
328|
329|
330|
331|
332|
333|
334|
335|
336|
337|
338|
339|
340|
341|
342|
343|
344|
345|
346|
347|
348|
349|
350|
351|
352|
353|
354|
355|
356|
357|
358|
359|
360|
361|
362|
363|
364|
365|
366|
367|
368|
369|
370|
371|
372|
373|
374|
375|
376|
377|
378|
379|
380|
381|
382|
383|
384|
385|
386|
387|
388|
389|
390|
391|
392|
393|
394|
395|
396|
397|
398|
399|
400|
401|
402|
403|
404|
405|
406|
407|
408|
409|
410|
411|
412|
413|
414|
415|
416|
417|
418|
419|
420|
421|
422|
423|
424|
425|
426|
427|
428|
429|
430|
431|
432|
433|
434|
435|
436|
437|
438|
439|
440|
441|
442|
443|
444|
445|
446|
447|
448|
449|
450|
451|
452|
453|
454|
455|
456|
457|
458|
459|
460|
461|
462|
463|
464|
465|
466|
467|
468|
469|
470|
471|
472|
473|
474|
475|
476|
477|
478|
479|
480|
481|
482|
483|
484|
485|
486|
487|
488|
489|
490|
491|
492|
493|
494|
495|
496|
497|
498|
499|
500|
501|
502|
503|
504|
505|
506|
507|
508|
509|
510|
511|
512|
513|
514|
515|
516|
517|
518|
519|
520|
521|
522|
523|
524|
525|
526|
527|
528|
529|
530|
531|
532|
533|
534|
535|
536|
537|
538|
539|
540|
541|
542|
543|
544|
545|
546|
547|
548|
549|
550|
551|
552|
553|
554|
555|
556|
557|
558|
559|
560|
561|
562|
563|
564|
565|
566|
567|
568|
569|
570|
571|
572|
573|
574|
575|
576|
577|
578|
579|
580|
581|
582|
583|
584|
585|
586|
587|
588|
589|
590|
591|
592|
593|
594|
595|
596|
597|
598|
599|
600|
601|
602|
603|
604|
605|
606|
607|
608|
609|
610|
611|
612|
613|
614|
615|
616|
617|
618|
619|
620|
621|
622|
623|
624|
625|
626|
627|
628|
629|
630|
631|
632|
633|
634|
635|
636|
637|
638|
639|
640|
641|
642|
643|
644|
645|
646|
647|
648|
649|
650|
651|
652|
653|
654|
655|
656|
657|
658|
659|
660|
661|
662|
663|
664|
665|
666|
667|
668|
669|
670|
671|
672|
673|
674|
675|
676|
677|
678|
679|
680|
681|
682|
683|
684|
685|
686|
687|
688|
689|
690|
691|
692|
693|
694|
695|
696|
697|
698|
699|
700|
701|
702|
703|
704|
705|
706|
707|
708|
709|
710|
711|
712|
713|
714|
715|
716|
717|
718|
719|
720|
721|
722|
723|
724|
725|
726|
727|
728|
729|
730|
731|
732|
733|
734|
735|
736|
737|
738|
739|
740|
741|
742|
743|
744|
745|
746|
747|
748|
749|
750|
751|
752|
753|
754|
755|
756|
757|
758|
759|
760|
761|
762|
763|
764|
765|
766|
767|
768|
769|
770|
771|
772|
773|
774|
775|
776|
777|
778|
779|
780|
781|
782|
783|
784|
785|
786|
787|
788|
789|
790|
791|
792|
793|
794|
795|
796|
797|
798|
799|
800|
801|
802|
803|
804|
805|
806|
807|
808|
809|
810|
811|
812|
813|
814|
815|
816|
817|
818|
819|
820|
821|
822|
823|
824|
825|
826|
827|
828|
829|
830|
831|
832|
833|
834|
835|
836|
837|
838|
839|
840|
841|
842|
843|
844|
845|
846|
847|
848|
849|
850|
851|
852|
853|
854|
855|
856|
857|
858|
859|
860|
861|
862|
863|
864|
865|
866|
867|
868|
869|
870|
871|
872|
873|
874|
875|
876|
877|
878|
879|
880|
881|
882|
883|
884|
885|
886|
887|
888|
889|
890|
891|
892|
893|
894|
895|
896|
897|
898|
899|
900|
901|
902|
903|
904|
905|
906|
907|
908|
909|
910|
911|
912|
913|
914|
915|
916|
917|
918|
919|
920|
921|
922|
923|
924|
925|
926|
927|
928|
929|
930|
931|
932|
933|
934|
935|
936|
937|
938|
939|
940|
941|
942|
943|
944|
945|
946|
947|
948|
949|
950|
951|
952|
953|
954|
955|
956|
957|
958|
959|
960|
961|
962|
963|
964|
965|
966|
967|
968|
969|
970|
971|
972|
973|
974|
975|
976|
977|
978|
979|
980|
981|
982|
983|
984|
985|
986|
987|
988|
989|
990|
991|
992|
993|
994|
995|
996|
997|
998|
999|
1000|
```

Profile 'mysql' ini merujuk ke pengaturan lanjutan yang ada di bagian 'connection': Maksudnya, ketika nilai 'mysql' di isi ke dalam pengaturan DB_CONNECTION di file .env, profile konfigurasi 'mysql' inilah yang dipakai. Terdapat cukup banyak pengaturan yang bisa ubah, termasuk pilihan *charset*, *collation*, *engine*, dll. File config/database.php ini lebih ditujukan untuk pengaturan yang lebih detail.

Selanjutnya buat database sesuai dengan konfigurasi yang ada.



2. Pengertian Migration

Secara sederhana, **migration** adalah fitur Laravel untuk men-*generate* tabel dengan cepat. Hanya dengan 1 perintah php artisan, satu hingga puluhan tabel bisa langsung tersedia. Yang perlu kita lakukan hanyalah menyiapkan struktur tabel tersebut. Migration disebut sebagai **version control untuk database**. Selain dipakai membuat tabel, kita juga bisa membuat step-by-step perubahan struktur tabel.

3. Migration Bawaan Laravel

Pada Laravel, sudah terdapat beberapa file migration bawaan. File ini berada di folder **database/migrations**.



Setiap file migration dipakai untuk membuat 1 tabel. Artinya 4 file migration ini akan membuat 4 buah tabel ke dalam database. Nama file migration diawali dengan *timestamp*, yakni waktu file tersebut dibuat. Dalam contoh di atas, file migration `2014_10_12_000000_create_users_table.php` dibuat pada tanggal 12 Oktober 2014. Setelah timestamp, diikuti dengan nama migration. Nama ini boleh bebas, misalnya `2023_03_20_000000_halo_sedang_belajar.php`. Namun format penulisan yang disarankan adalah sebagai berikut:

`<timestamp>_<proses>_<nama_tabel(s)>_table.php`

Jika misalkan kita ingin membuat file migration untuk tabel mahasiswa, lebih baik ditulis sebagai:

`2023_03_20_000000_create_mahasiswas_table.php`

Perhatikan bahwa ada tambahan huruf 's' di akhir. Ini adalah bentuk plural atau jamak dari sebuah kata dalam bahasa Inggris. Dalam Laravel kita **di sarankan membuat nama tabel dalam bentuk jamak**, caranya tambahkan huruf s di akhir kata, seperti mahasiswa, dosen, barang, dst.

Mari lihat isi salah satu file, yakni `2014_10_12_000000_create_users_table.php`:

```

1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12     public function up(): void
13     {
14         Schema::create('users', function (Blueprint $table) {
15             $table->id();
16             $table->string('name');
17             $table->string('email')->unique();
18             $table->timestamp('email_verified_at')->nullable();
19             $table->string('password');
20             $table->rememberToken();
21             $table->timestamps();
22         });
23     }
24
25     /**
26      * Reverse the migrations.
27      */
28     public function down(): void
29     {
30         Schema::dropIfExists('users');
31     }
32 };
33

```

Di awal terdapat kode program untuk proses import beberapa class internal Laravel, yakni Migration, Blueprint dan Schema (baris 3 – 5). Kemudian di baris 7 masuk ke kode pembuatan class `CreateUsersTable` yang di extends dari class Migration. Class ini memiliki 2 buah method: **up()** dan **down()**. Di dalam method `up()` inilah kode program untuk membuat tabel kita tulis, yakni menggunakan *static method* `Schema::create()`. Argument pertama dari method ini berupa nama tabel, yakni 'users'. Sedangkan argument kedua berupa sebuah *closure* tempat kita menulis struktur kolom dari tabel 'users'.

Mulai dari baris 16 – 22 adalah tempat untuk menulis nama dan tipe kolom. Mengenai apa saja tipe kolom untuk migration akan kita bahas sesaat lagi, tapi bisa dilihat sekilas bahwa nantinya untuk tabel users akan terdapat kolom 'id', 'name', 'email', 'email_verified_at' serta 'password'.

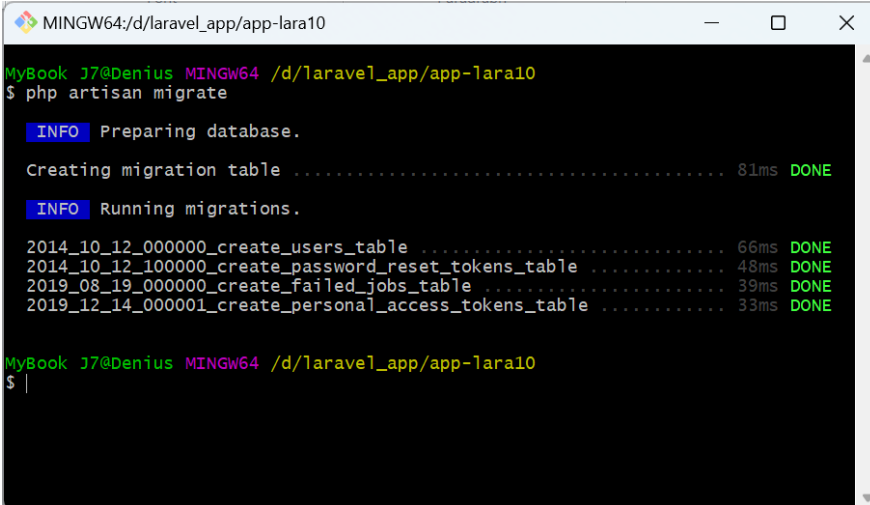
Method selanjutnya adalah `down()` di baris 28. Method ini berisi perintah kebalikan dari method `up()`. Jika di method `up()` kita membuat tabel 'users', maka di method `down()` ini ditulis perintah untuk menghapus tabel 'users' menggunakan perintah `Schema::dropIfExists('users')`.

Inilah teknik yang dipakai agar migration bisa berfungsi sebagai *version control*. Silahkan lihat juga kode program untuk kedua file migration lainnya, anda bisa perhatikan bahwa semua file identik satu sama lain, yang berubah hanyalah isi dari method `up()` dan `down()`.

4. Menjalankan Migration

Kita akan coba jalankan file migration bawaan Laravel. Ketiga file migration ini akan membuat tabel `users`, `password_resets` dan `failed_jobs` ke dalam database **laravel** yang sudah kita siapkan. Untuk menjalankan migration, silahkan buka **cmd**, masuk ke folder `laravel` dan jalankan perintah berikut:

```
php artisan migrate
```



```
MINGW64:/d/laravel_app/app-lara10
MyBook J7@Denius MINGW64 /d/laravel_app/app-lara10
$ php artisan migrate

  INFO  Preparing database.

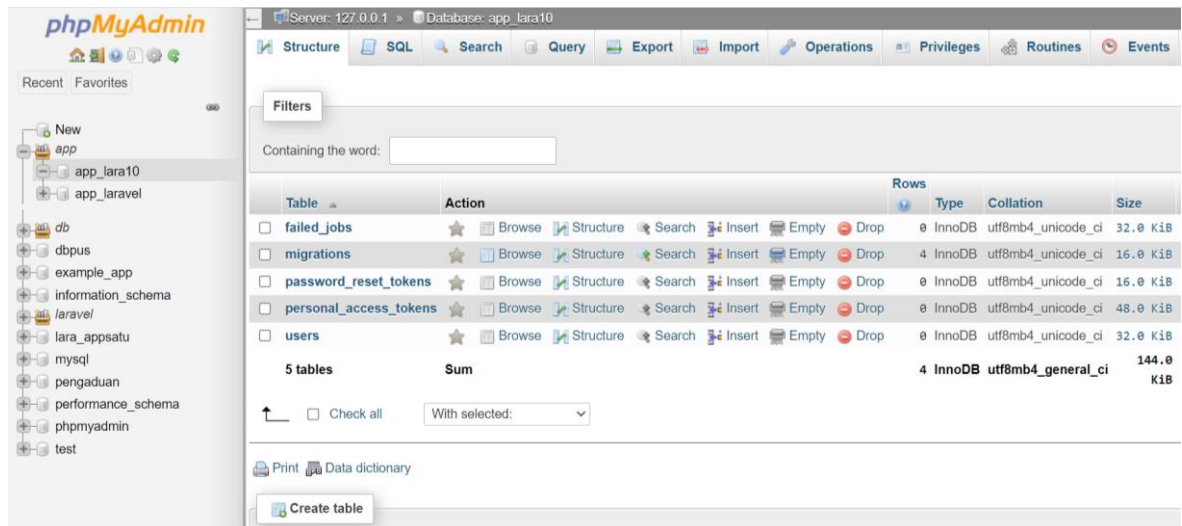
Creating migration table ..... 81ms DONE

  INFO  Running migrations.

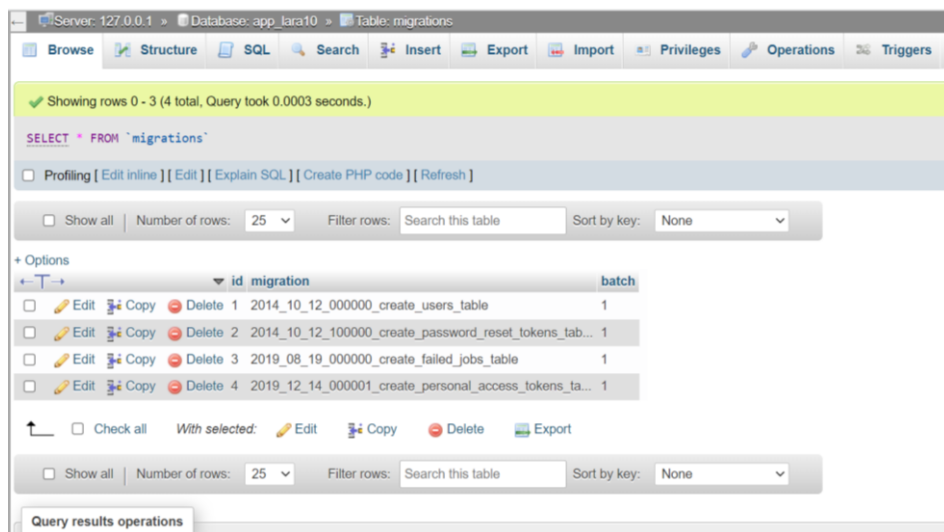
2014_10_12_000000_create_users_table ..... 66ms DONE
2014_10_12_100000_create_password_reset_tokens_table ..... 48ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 39ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 33ms DONE

MyBook J7@Denius MINGW64 /d/laravel_app/app-lara10
$ |
```

Jika tampil hasil seperti di atas, maka proses migration sudah sukses berjalan. Buka `phpMyAdmin` dan periksa isi database **app_lara10**.



Terdapat 5 tabel, yakni tabel users, password_resets, failed_jobs, personal_access_tokens serta tabel migrations. Tabel migrations secara khusus dipakai untuk mencatat proses migration yang terjadi

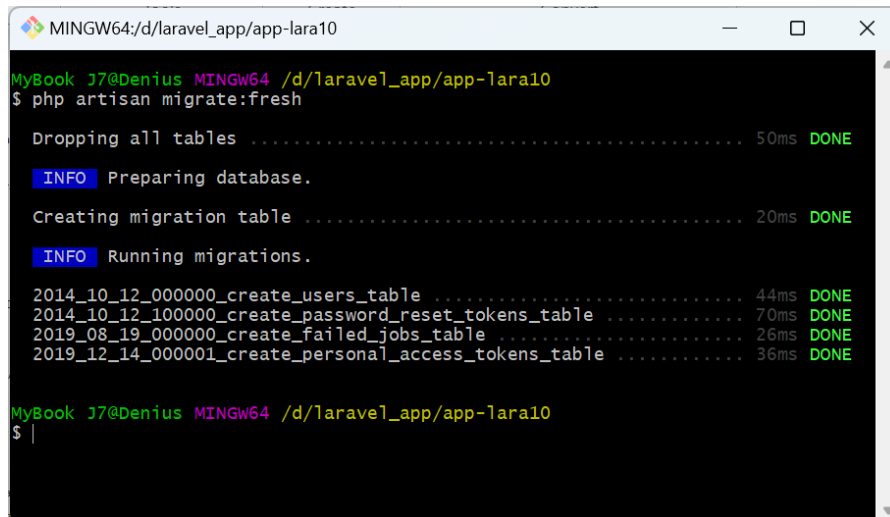


Tabel migration terdiri dari 3 kolom: id, migration dan batch. Kolom migration berisi semua file migration yang sudah dijalankan, sedangkan kolom batch dipakai untuk mencatat urutan proses migration. Semua file migration memiliki batch 1, karena kita memang baru menjalankan 1 kali perintah php artisan migrate.

Untuk menghapus isi tabel migration dan juga menghapus semua tabel yang ada bisa dilakukan dengan perintah
php artisan migrate:reset

Laravel juga menyediakan perintah php artisan migrate:fresh sebagai cara singkat untuk menghapus semua tabel dan langsung membuatnya kembali. Perintah ini ibarat perpaduan dari :

php artisan migrate:reset dan php artisan migrate dalam sekali jalan



```
MINGW64:/d/laravel_app/app-lara10
MyBook J7@Denius MINGW64 /d/laravel_app/app-lara10
$ php artisan migrate:fresh

Dropping all tables ..... 50ms DONE

INFO Preparing database.

Creating migration table ..... 20ms DONE

INFO Running migrations.

2014_10_12_000000_create_users_table ..... 44ms DONE
2014_10_12_100000_create_password_reset_tokens_table ..... 70ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 26ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 36ms DONE

MyBook J7@Denius MINGW64 /d/laravel_app/app-lara10
$ |
```

5. Migration Rollback

Pada saat menjalankan perintah php artisan migrate, file migration akan di eksekusi secara berurutan sesuai tanggal **timestamp**. File dengan timestamp terbaru akan dijalankan paling akhir. Pada prakteknya, bisa saja file migration ini ditambah secara bertahap. Misalkan programmer A sedang membuat fitur baru dan butuh tambahan sebuah tabel, maka dia bisa membuat dan menjalankan file migration untuk tabel_a. Esoknya programmer B juga bisa membuat file migration lain untuk tabel_b, dst.

Jika tiba-tiba terjadi masalah akibat penambahan tabel ini, kita bisa melakukan proses *rollback* sebagian terhadap file migration. Jadi seolah-olah kembali ke struktur database di hari sebelumnya. Untuk melihat daftar urutan migration, bisa dari perintah

php artisan migrate:status.



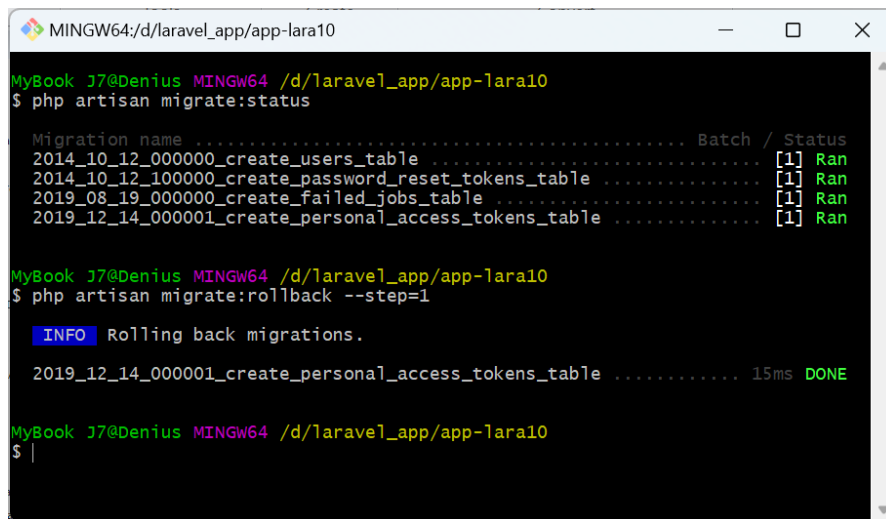
```
MINGW64:/d/laravel_app/app-lara10
MyBook J7@Denius MINGW64 /d/laravel_app/app-lara10
$ php artisan migrate:status

Migration name ..... Batch / Status
2014_10_12_000000_create_users_table ..... [1] Ran
2014_10_12_100000_create_password_reset_tokens_table ..... [1] Ran
2019_08_19_000000_create_failed_jobs_table ..... [1] Ran
2019_12_14_000001_create_personal_access_tokens_table ..... [1] Ran

MyBook J7@Denius MINGW64 /d/laravel_app/app-lara10
$
```


Di sini terlihat 4 file migration yang di urutkan berdasarkan timestamp, mulai dari tabel users, tabel password_resets, dan tabel failed_jobs, personal_access_tokens. Jika kita ingin mengembalikan posisi sebelum file migration terakhir, bisa menggunakan perintah berikut:

```
php artisan migrate:rollback --step=1
```



```
MINGW64:/d/laravel_app/app-lara10
MyBook J7@Denius MINGW64 /d/laravel_app/app-lara10
$ php artisan migrate:status

Migration name ..... Batch / Status
2014_10_12_000000_create_users_table ..... [1] Ran
2014_10_12_100000_create_password_reset_tokens_table ..... [1] Ran
2019_08_19_000000_create_failed_jobs_table ..... [1] Ran
2019_12_14_000001_create_personal_access_tokens_table ..... [1] Ran

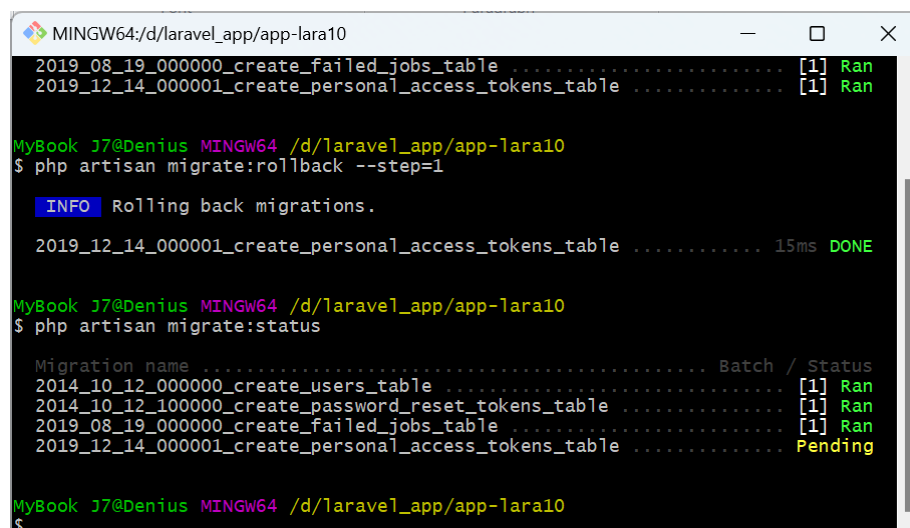
MyBook J7@Denius MINGW64 /d/laravel_app/app-lara10
$ php artisan migrate:rollback --step=1

INFO Rolling back migrations.

2019_12_14_000001_create_personal_access_tokens_table ..... 15ms DONE

MyBook J7@Denius MINGW64 /d/laravel_app/app-lara10
$ |
```

Perintah diatas akan me-*rollback* 1 file migration terakhir dan juga menghapus tabelnya. Akibat perintah ini, tabel personal_access_tokens sudah tidak ada lagi di database. Dalam project besar, bisa saja perlu mundur 6 atau 10 step jika diperlukan. Caranya, cukup mengatur nilai --step. Misalnya untuk mundur sebanyak 7 file migration, perintahnya adalah `php artisan migrate:rollback --step=7`. Jika perintah `php artisan migrate:status` dijalankan kembali, akan terlihat sedikit perubahan.



```
MINGW64:/d/laravel_app/app-lara10
2019_08_19_000000_create_failed_jobs_table ..... [1] Ran
2019_12_14_000001_create_personal_access_tokens_table ..... [1] Ran

MyBook J7@Denius MINGW64 /d/laravel_app/app-lara10
$ php artisan migrate:rollback --step=1

INFO Rolling back migrations.

2019_12_14_000001_create_personal_access_tokens_table ..... 15ms DONE

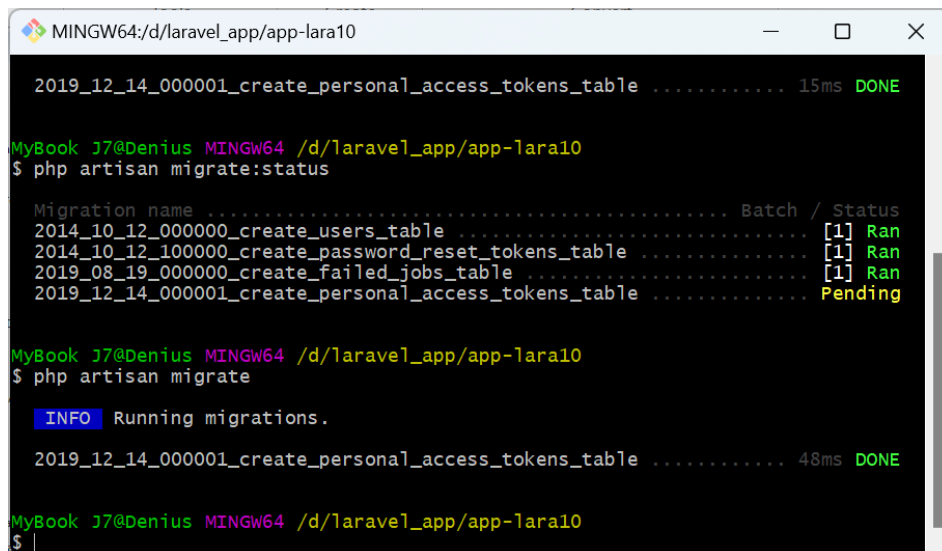
MyBook J7@Denius MINGW64 /d/laravel_app/app-lara10
$ php artisan migrate:status

Migration name ..... Batch / Status
2014_10_12_000000_create_users_table ..... [1] Ran
2014_10_12_100000_create_password_reset_tokens_table ..... [1] Ran
2019_08_19_000000_create_failed_jobs_table ..... [1] Ran
2019_12_14_000001_create_personal_access_tokens_table ..... Pending

MyBook J7@Denius MINGW64 /d/laravel_app/app-lara10
$
```

Sekarang isi kolom "Ran?" untuk file migration `personal_access_tokens` berisi Pending, yang menandakan file migration untuk tabel tersebut belum dijalankan atau sudah di rollback. Jika kita berubah pikiran dan ingin menjalankan kembali file migration untuk tabel `personal_access_tokens`, cukup dengan perintah

`php artisan migrate`.



```
MINGW64:/d/laravel_app/app-lara10

2019_12_14_000001_create_personal_access_tokens_table ..... 15ms DONE

MyBook J7@Denius MINGW64 /d/laravel_app/app-lara10
$ php artisan migrate:status

Migration name ..... Batch / Status
2014_10_12_000000_create_users_table ..... [1] Ran
2014_10_12_100000_create_password_reset_tokens_table ..... [1] Ran
2019_08_19_000000_create_failed_jobs_table ..... [1] Ran
2019_12_14_000001_create_personal_access_tokens_table ..... Pending

MyBook J7@Denius MINGW64 /d/laravel_app/app-lara10
$ php artisan migrate

[INFO] Running migrations.

2019_12_14_000001_create_personal_access_tokens_table ..... 48ms DONE

MyBook J7@Denius MINGW64 /d/laravel_app/app-lara10
$
```

Hasilnya, tabel `personal_access_tokens` sudah bisa digunakan kembali. Inilah fitur migration sebagai *version control*, dimana kita bisa 'maju-mundur' untuk semua file migration. Jika ternyata tabel baru yang ditambahkan menjadi masalah, tinggal di-rollback saja. Atau jika ada tabel yang tidak sengaja terhapus, tinggal jalankan ulang file migration.

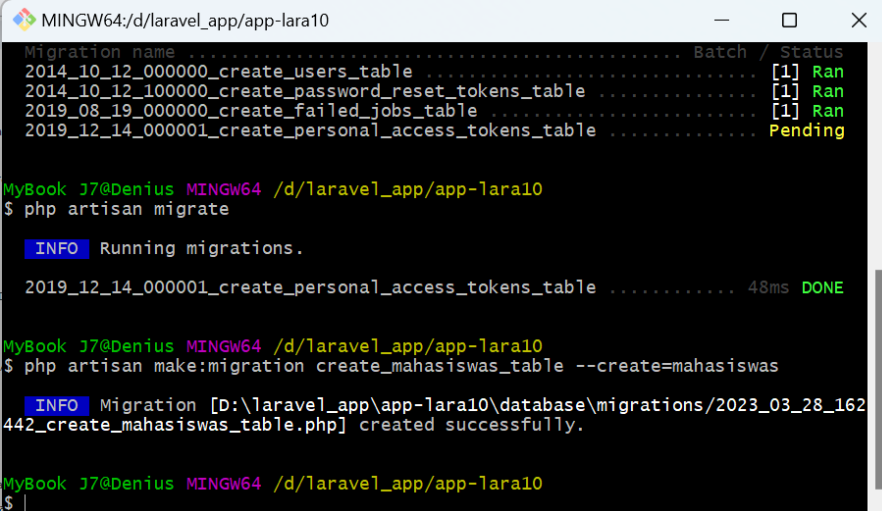
6. Membuat Migration

Sama seperti controller, file migration bisa dibuat dari teks editor. Caranya, copy kode dari file migration yang sudah ada, lalu edit di bagian yang diperlukan. Atau cara yang lebih praktis adalah dari perintah **php artisan** dengan format berikut:

```
php artisan make:migration <nama_migration> --create=<nama_tabel>
```

Untuk `nama_migration` bisa di isi suka-suka, namun penulisan terbaik adalah dengan format: `<nama_proses>_<nama_tabel(s)>_table`. Sebagai contoh, untuk membuat file migration tabel **mahasiswas**, bisa dengan menjalankan perintah berikut:

```
php artisan make:migration create_mahasiswas_table --create=mahasiswas
```



```
MINGW64;d:\laravel_app\app-lara10
Migration name ..... Batch / Status
2014_10_12_000000_create_users_table ..... [1] Ran
2014_10_12_100000_create_password_reset_tokens_table ..... [1] Ran
2019_08_19_000000_create_failed_jobs_table ..... [1] Ran
2019_12_14_000001_create_personal_access_tokens_table ..... Pending

MyBook J7@Denius MINGW64 /d/\laravel_app\app-lara10
$ php artisan migrate

  INFO  Running migrations.

 2019_12_14_000001_create_personal_access_tokens_table ..... 48ms DONE

MyBook J7@Denius MINGW64 /d/\laravel_app\app-lara10
$ php artisan make:migration create_mahasiswas_table --create=mahasiswas

  INFO  Migration [D:\laravel_app\app-lara10\database\Migrations\2023_03_28_162442_create_mahasiswas_table.php] created successfully.

MyBook J7@Denius MINGW64 /d/\laravel_app\app-lara10
$
```

Hasilnya, di dalam folder database\Migrations akan tampil file 2023_03_28_162442_create_mahasiswas_table.php. Nama file ini akan berbeda karena tanggal timestamp di generate secara otomatis yang diambil dari sistem komputer.

```
database > migrations > 2023_03_28_162442_create_mahasiswas_table.php > ...
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12     public function up(): void
13     {
14         Schema::create('mahasiswas', function (Blueprint $table) {
15             $table->id();
16             $table->timestamps();
17         });
18     }
19
20     /**
21      * Reverse the migrations.
22      */
23     public function down(): void
24     {
25         Schema::dropIfExists('mahasiswas');
26     }
27 };
28
```

Kode ini sangat mirip seperti yang ada di file migration bawaan Laravel. Kita akan langsung fokus ke method up() saja, karena di sinilah tempat untuk menulis struktur tabel. Format penulisan nama kolom tabel adalah sebagai berikut:

```
$table-><tipe_data_kolom>(<nama_kolom>)
```

Untuk tipe_data_kolom, tidak selalu sesuai dengan bawaan MySQL. Laravel menyediakan berbagai tipe data baru yang kadang merupakan gabungan dari tipe data kolom MySQL. Sebagai contoh, di baris 15 terdapat pemanggilan method `$table->id()`, ini dipakai untuk membuat kolom bertipe **big integer** yang langsung di set dengan **auto increment**. Untuk daftar Jenis tipe data kolom yang bisa dipakai lengkapnya bisa ke <https://laravel.com/docs/10.x/migrations>

# Available Column Types		
The schema builder blueprint offers a variety of methods that correspond to the different types of columns you can add to your database tables. Each of the available methods are listed in the table below:		
bigIncrements	jsonb	string
bigInteger	lineString	text
binary	longText	timeTz
boolean	macAddress	time
char	mediumIncrements	timestampTz
dateTimeTz	mediumInteger	timestamp
dateTime	mediumText	timestampTz
date	morphs	timestamps
decimal	multiLineString	tinyIncrements
double	multiPoint	tinyInteger
enum	multiPolygon	tinyText
float	nullableMorphs	unsignedBigInteger
foreignId	nullableTimestamps	unsignedDecimal

Dari daftar ini ada beberapa tipe data yang bisa men-generate kolom lain. Sebagai contoh `$table->timestamps()` akan menambah 2 buah kolom, yakni `created_at` and `updated_at` ke dalam tabel. Kedua kolom bertipe timestamp yang akan terisi otomatis pada saat data tabel ditambah / dimodifikasi (tidak perlu diinput manual).

Kolom `created_at` and `updated_at` ini sangat bermanfaat proses mengurutkan data berdasarkan tanggal dibuat dan tanggal modifikasi. Disarankan untuk selalu menambah `$table->timestamps()` untuk setiap tabel yang dibuat. Selain itu beberapa method punya pengaturan tambahan yang di input sebagai argument kedua atau ketiga. Sebagai contoh, `$table->char('name', 100)` dipakai untuk membuat kolom 'name' dengan tipe data CHAR(100). Begitu juga dengan method `$table->double('amount', 8,2)` yang dipakai untuk membuat kolom 'amount' dengan tipe data DOUBLE(8,2)

Kita akan tambah beberapa kolom lain ke dalam file migrate mahasiswa:

```

database > migrations > 2023_03_28_162442_create_mahasiswas_table.php > class > up > Closure
9      /**
10     * Run the migrations.
11     */
12     public function up(): void
13     {
14         Schema::create('mahasiswas', function (Blueprint $table) {
15             $table->id();
16             $table->char('nim',10);
17             $table->string('nama');
18             $table->string('tempat_lahir');
19             $table->date('tgl_lahir');
20             $table->string('prodi');
21             $table->decimal('ipk',3,2);
22             $table->string('alamat');
23             $table->timestamps();
24         });
25     }
26
27     /**
28     * Reverse the migrations.
29     */
30     public function down(): void

```

Save file migrate, lalu jalankan kembali php artisan migrate:

```

MINGW64/d/laravel_app/app-lara10
$ php artisan migrate
INFO Running migrations.
2019_12_14_000001_create_personal_access_tokens_table ..... 48ms DONE

MyBook J7@Denius MINGW64 /d/laravel_app/app-lara10
$ php artisan make:migration create_mahasiswas_table --create=mahasiswas
INFO Migration [D:\laravel_app\app-lara10\database\migrations\2023_03_28_162442_create_mahasiswas_table.php] created successfully.

MyBook J7@Denius MINGW64 /d/laravel_app/app-lara10
$ php artisan migrate
INFO Running migrations.
2023_03_28_162442_create_mahasiswas_table ..... 18ms DONE

MyBook J7@Denius MINGW64 /d/laravel_app/app-lara10
$

```

Server: 127.0.0.1 > Database: app_lara10 > Table: mahasiswas

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	
<input type="checkbox"/> 2	nim	char(10)	utf8mb4_unicode_ci		No	None			
<input type="checkbox"/> 3	nama	varchar(255)	utf8mb4_unicode_ci		No	None			
<input type="checkbox"/> 4	tempat_lahir	varchar(255)	utf8mb4_unicode_ci		No	None			
<input type="checkbox"/> 5	tgl_lahir	date			No	None			
<input type="checkbox"/> 6	prodi	varchar(255)	utf8mb4_unicode_ci		No	None			
<input type="checkbox"/> 7	ipk	decimal(3,2)			No	None			
<input type="checkbox"/> 8	alamat	varchar(255)	utf8mb4_unicode_ci		No	None			
<input type="checkbox"/> 9	created_at	timestamp			Yes	NULL			
<input type="checkbox"/> 10	updated_at	timestamp			Yes	NULL			

☐ Check all
 With selected: Browse Change Drop Primary Unique Index Spatial Fulltext

Inilah struktur tabel yang dihasilkan dari migration. Tergantung kebutuhan, kita tinggal menambah kolom yang sesuai berdasarkan daftar method yang ada di tabel sebelumnya.

Laravel juga menyediakan method lanjutan untuk menambah atribut khusus atau *modifier* ke dalam kolom. Misalnya untuk membuat nilai default, men-set kolom agar bisa menerima nilai null, mengatur charset, dll. Penulisan method ini di-*chaining* dari method yang dipakai untuk membuat kolom.

Sebagai contoh, untuk men-set kolom 'nim' agar tidak bisa diinput dengan nilai yang sama, bisa ditulis sebagai:

```
$table->char('nim',8)->unique();
```

Atau agar kolom 'ipk' memiliki nilai default 1.00 bisa ditulis dengan perintah:

```
$table->decimal('ipk',3,2)->default(1.00);
```

7. Alter Table Migration

Migration tidak hanya dipakai untuk membuat dan menghapus tabel, tapi juga untuk modifikasi struktur tabel (menjalankan query ALTER). Ini cukup sering kita lakukan sepanjang pembuatan project. Misalnya tiba-tiba client memberitahu ada perubahan data yang butuh modifikasi struktur tabel. Sebagai antisipasi, kita bisa membuat file migration baru yang berisi perintah modifikasi tersebut. Jika ternyata ada masalah, dengan 1 perintah rollback, tabel bisa kembali ke bentuk semula.

Agar bisa melakukan modifikasi tabel ke dalam migration, Laravel butuh sebuah library tambahan bernama [Doctrine DBAL](#). Proses instalasi library ini sangat mudah, cukup jalankan perintah berikut di folder laravel:

```
composer require doctrine/dbal
```

Berikutnya, kita modifikasi struktur tabel mahasiswa. Silahkan buat file migration baru dengan nama `alter_mahasiswa_table`:

```
php artisan make:migration alter_mahasiswa_table --table=mahasiswa
```

```
MINGW64:/d/laravel_app/app-lara10
MyBook J7@Denius MINGW64 /d/laravel_app/app-lara10
$ php artisan make:migration alter_mahasiswa_table --table=mahasiswa

[INFO] Migration [D:\laravel_app\app-lara10\database\migrations\2023_03_28_165745_alter_mahasiswa_table.php] created successfully.

MyBook J7@Denius MINGW64 /d/laravel_app/app-lara10
$
```

Kemudian buka file migration ini di teks editor. Terlihat method up() dan down() tidak berisi kode apapun karena Laravel bisa mendeteksi kalau tabel mahasiswa sudah ada, sehingga tidak mungkin file migration juga dipakai pembuatan tabel lagi. Modifikasi file migration sebagai berikut:

```
database > migrations > 2023_03_28_165745_alter_mahasiswa_table.php > class > down > Closure
12 public function up(): void
13 {
14     Schema::table('mahasiswa', function (Blueprint $table) {
15         $table->renameColumn('nama', 'nama_lengkap');
16         $table->string('email')->after('tgl_lahir');
17         $table->dropColumn('ipk');
18     });
19 };
20
21
22 /**
23  * Reverse the migrations.
24  */
25 public function down(): void
26 {
27     Schema::table('mahasiswa', function (Blueprint $table) {
28         $table->renameColumn('nama_lengkap', 'nama');
29         $table->dropColumn('email');
30         $table->decimal('ipk', 3, 2)->default(1.00);
31     });
32 };
33 }
```

Kode program di dalam method up() dan down() harus berpasangan agar proses *rollback* bisa berlangsung dengan baik. Sekarang mari kita jalankan perubahan ini dengan perintah `php artisan migrate`:

```
MyBook J7@Denius MINGW64 /d/laravel_app/app-lara10
$ php artisan migrate

[INFO] Running migrations.

2023_03_28_165745_alter_mahasiswa_table ..... 60ms DONE

MyBook J7@Denius MINGW64 /d/laravel_app/app-lara10
$
```

Server: 127.0.0.1 > Database: app_lara10 > Table: mahasiswa

[Browse](#)
[Structure](#)
[SQL](#)
[Search](#)
[Insert](#)
[Export](#)
[Import](#)
[Privileges](#)
[Operations](#)
[Triggers](#)

[Table structure](#)
[Relation view](#)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/> 2	nim	char(10)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/> 3	nama_lengkap	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/> 4	tempat_lahir	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/> 5	tgl_lahir	date			No	None			Change Drop More
<input type="checkbox"/> 6	email	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/> 7	prodi	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/> 8	alamat	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/> 9	created_at	timestamp			Yes	NULL			Change Drop More
<input type="checkbox"/> 10	updated_at	timestamp			Yes	NULL			Change Drop More

☐ Check all
 With selected:
 [Browse](#)
[Change](#)
[Drop](#)
[Primary](#)
[Unique](#)
[Index](#)
[Spatial](#)
[Fulltext](#)

Sekarang tabel mahasiswa telah berhasil di modifikasi. Kolom 'nama' sudah berubah jadi 'nama_lengkap', kolom 'email' juga telah ditambahkan dan berada setelah kolom 'tanggal_lahir', serta kolom 'ipk' juga tidak lagi ditemukan.

Apabila setelah beberapa saat terjadi masalah atau kita berubah pikiran, perubahan ini bisa di rollback dengan `php artisan migrate:rollback --step=1`: