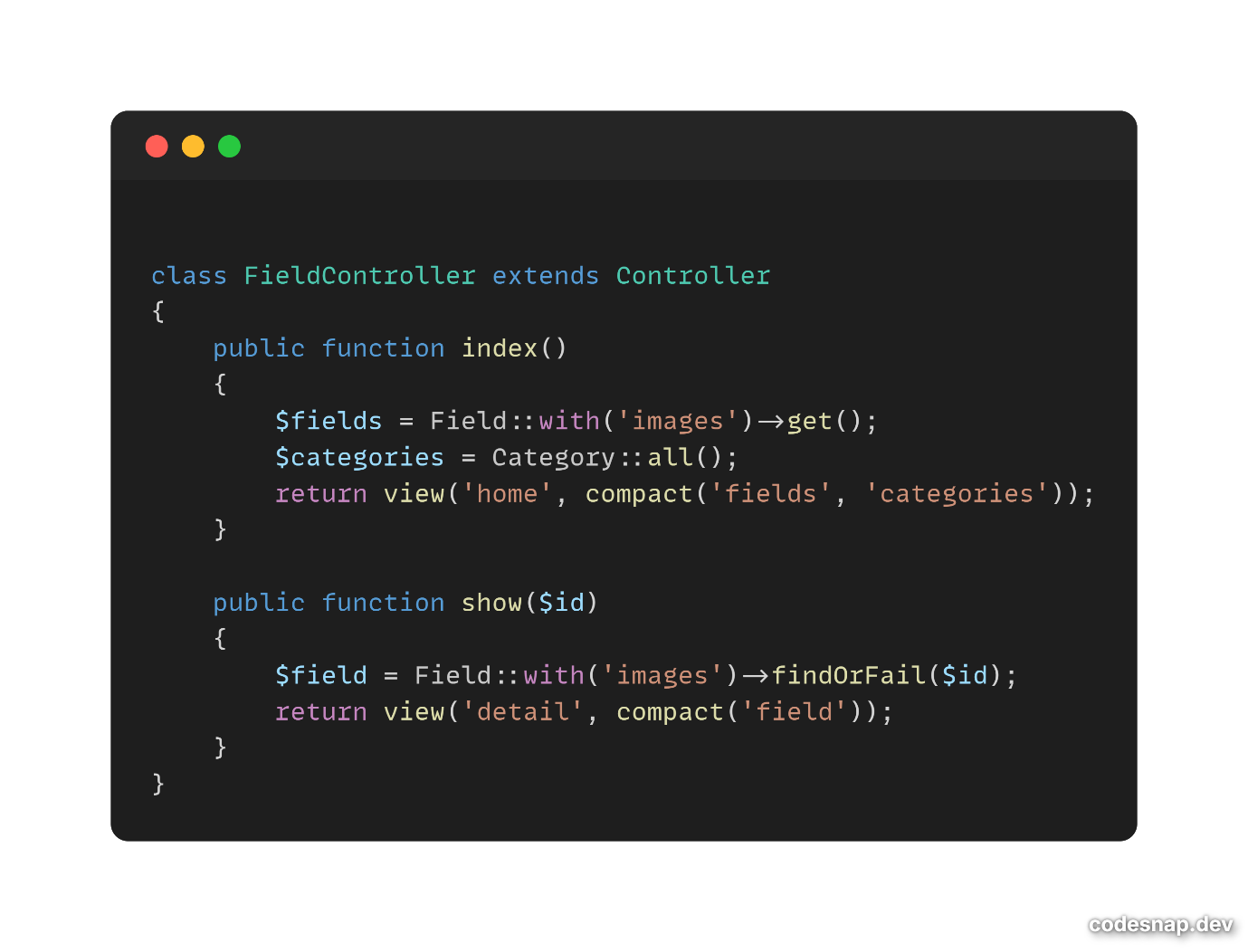
1. Controller
2. Field Controller



Berikut penjelasan detail dari kode **FieldController** (Laravel Controller):

**Class & Fungsi**

class FieldController extends Controller

* Ini adalah controller di Laravel yang mewarisi Controller bawaan Laravel.
* Biasanya dipakai untuk menangani permintaan HTTP (request) yang berkaitan dengan "Field".

**Method index()**

public function index()

{

$fields = Field::with('images')->get();

$categories = Category::all();

return view('home', compact('fields', 'categories'));

}

**Penjelasan:**

* **Field::with('images')->get();**  
  Mengambil semua data **Field**, sekaligus mengambil relasi images.  
  → with('images') berarti eager loading relasi "images" agar tidak terjadi N+1 query.
* **Category::all();**  
  Mengambil semua data kategori.
* **return view('home', compact('fields', 'categories'));**  
  Mengirim data $fields dan $categories ke view bernama home.blade.php.

**Tujuan:**  
Method ini digunakan untuk menampilkan halaman utama (home) yang menampilkan daftar semua field beserta gambarnya, dan juga semua kategori.

**Method show($id)**

public function show($id)

{

$field = Field::with('images')->findOrFail($id);

return view('detail', compact('field'));

}

**Penjelasan:**

* **Field::with('images')->findOrFail($id);**  
  Mencari satu data **Field** berdasarkan $id.  
  → Jika tidak ketemu, akan melempar error 404 (findOrFail).  
  → Sekaligus mengambil relasi images.
* **return view('detail', compact('field'));**  
  Mengirim data $field ke view detail.blade.php.

**Tujuan:**  
Method ini dipakai untuk menampilkan detail dari satu Field tertentu, lengkap dengan gambar-gambarnya.

**Kesimpulan**

* **index()** → Menampilkan daftar field dan kategori di halaman home.
* **show($id)** → Menampilkan detail satu field tertentu di halaman detail.

1. Booking Controller



**📄 Kode BookingController**

class BookingController extends Controller

{

public function store(Request $request)

➡️ Membuat controller bernama BookingController yang mewarisi Controller Laravel.  
➡️ Method store() dipakai untuk menyimpan data booking yang dikirim melalui form.

**✅ Validasi data request**

$validated = $request->validate([

'field\_id' => 'required|exists:fields,id',

'name' => 'required|string|max:255',

'address' => 'required|string',

'whatsapp\_number'=> 'required|string|regex:/^[0-9]{10,15}$/',

'booking\_time' => 'required|date|after:now',

]);

➡️ Memvalidasi input dari user:

* **field\_id** wajib diisi dan harus ada di tabel fields kolom id.
* **name** wajib, string, maksimal 255 karakter.
* **address** wajib, string.
* **whatsapp\_number** wajib, hanya angka, panjang 10–15 digit.
* **booking\_time** wajib, harus berupa tanggal (format date), dan harus di masa depan.

**✅ Mengambil data Field & harga**

$field = Field::findOrFail($validated['field\_id']);

$price = $field->price\_per\_hour;

➡️ Mengambil data lapangan berdasarkan ID yang sudah divalidasi.  
➡️ Jika tidak ketemu, otomatis error 404.  
➡️ Mengambil harga per jam dari lapangan (property price\_per\_hour).

**✅ Cek apakah sudah dibooking**

$existingBooking = Booking::where('field\_id', $validated['field\_id'])

->where('booking\_time', $validated['booking\_time'])

->whereIn('status', ['pending', 'confirmed'])

->exists();

➡️ Mengecek apakah sudah ada booking di lapangan dan waktu yang sama.  
➡️ Hanya cek booking dengan status **pending** atau **confirmed** (jadi slot yang sudah di-booking tapi belum selesai dianggap sudah diambil).  
➡️ exists() mengembalikan **true** kalau ada data.

**✅ Jika sudah dibooking, kembalikan error**

if ($existingBooking) {

return back()->withErrors(['booking\_time' => 'Waktu ini sudah dipesan.'])->withInput();

}

➡️ Kalau waktu sudah dibooking, kembalikan ke halaman sebelumnya.  
➡️ Kirim pesan error "Waktu ini sudah dipesan" dan withInput() supaya data form tetap terisi.

**✅ Menyimpan booking baru**

Booking::create(array\_merge($validated, [

'price' => $price,

'status' => 'pending',

]));

➡️ Membuat booking baru.  
➡️ Data yang divalidasi digabung (array\_merge) dengan harga (price) dan status default (pending).

**✅ Pesan sukses**

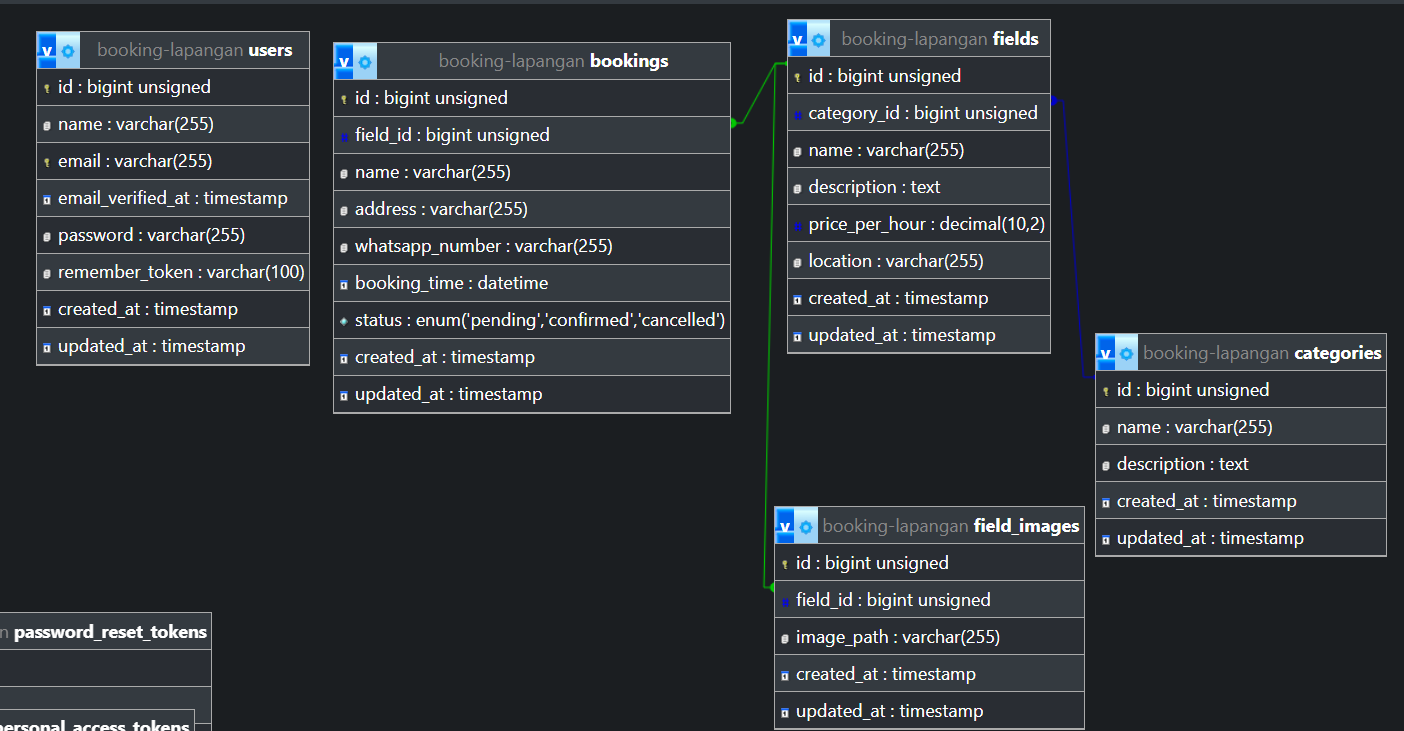
return back()->with('success', 'Booking telah dibuat, menunggu konfirmasi admin.');

➡️ Setelah berhasil, kembalikan ke halaman sebelumnya.  
➡️ Tampilkan pesan sukses: **"Booking telah dibuat, menunggu konfirmasi admin."**

**✅ Kesimpulan**

* Controller ini digunakan untuk menerima form booking.
* Validasi ketat (field wajib, format WhatsApp, tanggal masa depan).
* Cek apakah slot waktu sudah diambil.
* Jika masih tersedia, booking dibuat dengan status pending, harga diisi otomatis.
* User mendapat feedback (error atau sukses).

1. ERD



**📄 Penjelasan ERD Booking Lapangan**

ERD ini menggambarkan relasi antar tabel pada sistem booking lapangan. Terdapat lima tabel utama, yaitu users, bookings, fields, categories, dan field\_images.

**✅ Tabel users**

Tabel ini menyimpan data pengguna sistem yang dapat melakukan booking. Setiap user memiliki kolom id, name, email, email\_verified\_at, password, remember\_token, serta timestamp created\_at dan updated\_at untuk mencatat waktu pembuatan dan pembaruan data.

**✅ Tabel categories**

Tabel ini berfungsi menyimpan kategori dari lapangan, seperti misalnya kategori "futsal", "badminton", atau "basket". Kolom penting di tabel ini adalah id, name, description, serta timestamp created\_at dan updated\_at.

**✅ Tabel fields**

Tabel ini menyimpan data lapangan yang tersedia untuk booking. Terdapat kolom id, category\_id sebagai foreign key ke tabel categories, name, description, price\_per\_hour yang menunjukkan harga sewa per jam, location, serta created\_at dan updated\_at. Relasi antara fields dan categories adalah **many-to-one**, artinya setiap field hanya bisa memiliki satu kategori, tetapi satu kategori bisa digunakan oleh banyak field.

**✅ Tabel field\_images**

Tabel ini digunakan untuk menyimpan gambar lapangan. Memiliki kolom id, field\_id sebagai foreign key yang menghubungkan gambar ke lapangan, image\_path untuk menyimpan lokasi file gambar, serta created\_at dan updated\_at. Relasi antara field\_images dan fields adalah **many-to-one**, artinya satu lapangan bisa memiliki banyak gambar.

**✅ Tabel bookings**

Tabel ini menyimpan data pemesanan lapangan oleh user. Kolom yang terdapat di dalamnya meliputi id, field\_id sebagai foreign key ke tabel fields, name (nama pemesan), address, whatsapp\_number, booking\_time (tanggal dan jam booking), serta status yang berisi nilai enum: pending, confirmed, atau cancelled. Terdapat juga kolom created\_at dan updated\_at. Relasi antara bookings dan fields adalah **many-to-one**, artinya satu lapangan bisa memiliki banyak booking.

**Kesimpulan Relasi**

* **categories ➡️ fields**: one-to-many (satu kategori bisa digunakan banyak lapangan).
* **fields ➡️ field\_images**: one-to-many (satu lapangan memiliki banyak gambar).
* **fields ➡️ bookings**: one-to-many (satu lapangan bisa dibooking banyak kali).
* **users**: Berdiri sendiri di diagram ini, biasanya digunakan untuk login atau otorisasi (meskipun di booking tidak langsung ada relasi foreign key ke users).

**Manfaat desain ini**

Desain ERD ini memungkinkan sistem untuk mengelola data lapangan dengan kategori, menampilkan galeri gambar, serta menangani booking yang aman (dengan pengecekan status). Dengan demikian, sistem dapat mendukung proses booking lapangan yang terstruktur, fleksibel, dan terkontrol.