

## DOKUMENTASI UAS PEMROGRAMAN BASIS DATA

Nama: Marvello Adwitya Nyahu

NIM: 2455011013

Kelas: D4 TRPL 3A

### 1. INSERT

SQL Manual: `INSERT INTO kategori (nama_kategori) VALUES ('Cat & Perlengkapan');`

Laravel ORM: `Kategori::create(['nama_kategori' => 'Cat & Perlengkapan']);`

### 2. SELECT + WHERE

SQL Manual: `SELECT * FROM produk WHERE stok <= stok_minimum AND status = 'aktif';`

Laravel ORM: `$produk = Produk::whereColumn('stok', '<=', 'stok_minimum')->where('status', 'aktif')->get();`

### 3. UPDATE

SQL Manual: `UPDATE produk SET harga_jual = 70000 WHERE id_produk = 1;`

Laravel ORM: `Produk::find(1)->update(['harga_jual' => 70000]);`

### 4. DELETE

SQL Manual: `DELETE FROM supplier WHERE id_supplier = 10;`

Laravel ORM: `Supplier::destroy(10);`

## 5. JOIN

### INNER JOIN

SQL Manual: `SELECT p.nama_produk, k.nama_kategori FROM produk p INNER JOIN kategori k ON p.id_kategori = k.id_kategori;`

Laravel ORM: `$produk = Produk::with('kategori')->get();`

### LEFT JOIN

SQL Manual: `SELECT p.nama_pelanggan, n.id_nota FROM pelanggan p LEFT JOIN penjualan n ON p.id_pelanggan = n.id_pelanggan;`

Laravel ORM: `Pelanggan::leftJoin('penjualan', 'pelanggan.id_pelanggan', '=', 'penjualan.id_pelanggan')->get();`

### JOIN tanpa JOIN keyword

SQL Manual: `SELECT p.nama_produk, s.nama_satuan FROM produk p, satuan s WHERE p.id_satuan = s.id_satuan;`

Laravel ORM: `DB::select("SELECT p.nama_produk, s.nama_satuan FROM produk p, satuan s WHERE p.id_satuan = s.id_satuan");`

## 6. CROSS JOIN

SQL Manual: `SELECT p.nama_produk, s.nama_supplier FROM produk p CROSS JOIN supplier s;`

Laravel ORM: `DB::table('produk')->crossJoin('supplier')->get();`

## 7. UNION & UNION ALL

SQL Manual: `(SELECT nama_pelanggan FROM pelanggan) UNION ALL (SELECT nama_pegawai FROM pegawai);`

Laravel ORM: `$query = DB::table('pelanggan')->select('nama_pelanggan')->unionAll(DB::table('pegawai')->select('nama_pegawai'))->get();`

## 8. TCL (Transaction Control Language)

Implementasi untuk validasi bisnis agar stok tidak minus.

- START TRANSACTION / commit / rollback

SQL Manual: START TRANSACTION; UPDATE produk SET stok = stok - 1 WHERE id\_produk = 1;  
COMMIT; (atau ROLLBACK; jika gagal)

Laravel ORM:

```
DB::beginTransaction(); // Memulai transaksi  
  
try {  
  
    // Proses pengurangan stok dan simpan nota  
  
    DB::commit(); // Simpan permanen  
  
} catch (\Exception $e) {  
  
    DB::rollback(); // Batalkan semua jika error  
  
}
```

- SAVEPOINT

SQL Manual: SAVEPOINT sp\_nota\_dibuat;

Laravel ORM: DB::transaction(function () { DB::unprepared('SAVEPOINT sp1'); }); (Atau otomatis dikelola Laravel dalam blok transaksi)