**LAPORAN PERTEMUAN 13 TEORI**

**SISTEM BASIS DATA**



**Ditulis oleh :**

|  |  |
| --- | --- |
| NAMA | : Ilham Romadhani |
| NIM | : 23.230.0030 |
| KELAS | : 2P52 |

**Dosen Pengampu :**

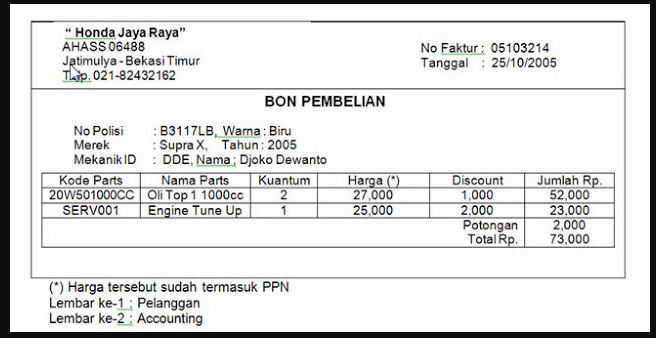
Mosses Aidjilli, M.Kom

PROGRAM STUDI SISTEM INFORMASI

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER

(STMIK) WIDYA PRATAMA PEKALONGAN

2024



**1NF**



### **Kriteria Normalisasi 1NF**

1. **Nilai Tunggal:** Setiap kolom dalam tabel harus berisi nilai-nilai yang tidak dapat dibagi lagi. Setiap kolom harus berisi satu nilai saja.
2. **Baris Unik:** Setiap baris dalam tabel harus unik. Tidak boleh ada duplikasi baris.
3. **Konsistensi Tipe Data:** Setiap kolom harus memiliki tipe data yang sama di seluruh baris tabel.
4. **Tidak Ada Grup Berulang:** Tidak boleh ada grup berulang dalam tabel. Setiap kolom harus independen dari kolom lainnya.

### **Penjelasan 1NF untuk Tabel Pertama:**

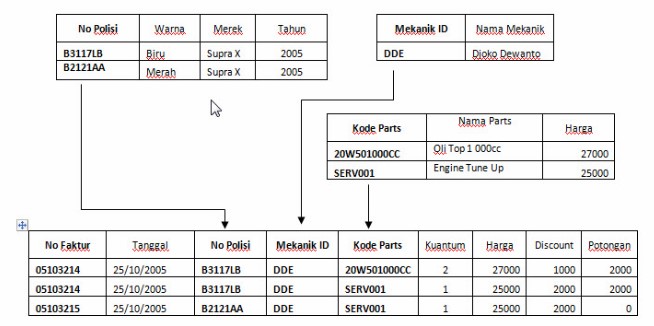
Pada tabel tersebut, setiap kolom berisi satu nilai tunggal, misalnya **No Faktur** hanya berisi satu nomor faktur, dan **No Polisi** hanya berisi satu nomor polisi. Setiap baris unik dan tidak ada duplikasi, karena kombinasi kolom seperti No Faktur, No Polisi, dan Kode Parts memastikan setiap entri berbeda. Tipe data konsisten di seluruh baris, dan tidak ada grup berulang.

### **Penjelasan 1NF untuk Tabel Kedua:**

Setiap kolom berisi satu nilai tunggal, misalnya **Nama Parts** hanya berisi satu nama part, dan Kuantum hanya berisi satu nilai kuantum. Meskipun ada baris yang tampak mirip, kombinasi kolom yang ada memastikan setiap baris unik. Tipe data konsisten di seluruh kolom, dan tidak ada grup berulang.

Kesimpulannya, kedua tabel tersebut telah memenuhi First Normal Form (1NF) dengan memastikan bahwa setiap kolom berisi nilai tunggal, setiap baris unik, tipe data konsisten, dan tidak ada grup berulang.

**2NF**



### **Kriteria Normalisasi 2NF**

1. **Memenuhi 1NF:** Tabel harus terlebih dahulu memenuhi semua kriteria 1NF.
2. **Menghilangkan Ketergantungan Parsial:** Semua atribut non-kunci harus bergantung sepenuhnya pada kunci utama. Tidak boleh ada atribut yang bergantung hanya pada bagian dari kunci utama jika kunci utama tersebut terdiri dari lebih dari satu kolom.

**PENJELASAN :**

* **Tabel Kendaraan**

Pada tabel Kendaraan, setiap kolom berisi informasi tentang kendaraan tertentu, seperti nomor polisi, warna, merek, dan tahun. Semua kolom ini sepenuhnya bergantung pada kunci utama No Polisi.

* **Tabel Mekanik**

Pada tabel Mekanik, setiap kolom berisi informasi tentang mekanik, seperti ID mekanik dan nama mekanik. Semua kolom ini sepenuhnya bergantung pada kunci utama Mekanik ID.

* **Tabel Parts**

Pada tabel Parts, setiap kolom berisi informasi tentang parts, seperti kode parts, nama parts, dan harga. Semua kolom ini sepenuhnya bergantung pada kunci utama Kode Parts.

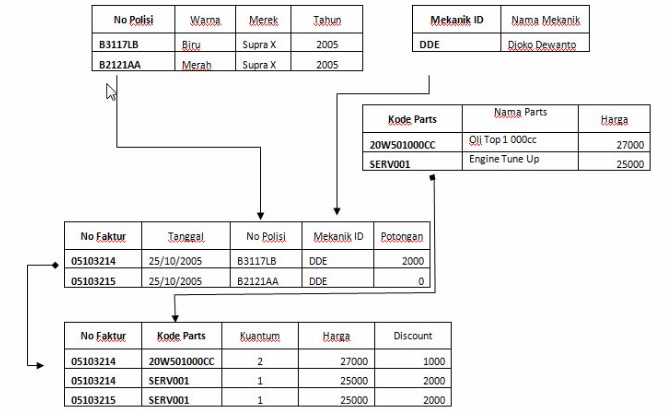
* **Tabel Transaksi**

Pada tabel Transaksi, kombinasi No Faktur, No Polisi, Mekanik ID, dan Kode Parts berfungsi sebagai composite primary key. Tabel ini berisi informasi tentang faktur dan pembelian parts, seperti nomor faktur, tanggal, nomor polisi, ID mekanik, kode parts, kuantum, harga, discount, jumlah, potongan, dan total. Semua kolom ini sepenuhnya bergantung pada composite primary key tersebut.

### **Kesimpulan**

Dengan memecah tabel utama menjadi beberapa tabel lebih kecil dan memastikan setiap tabel memiliki primary key yang tepat, kita menghilangkan ketergantungan parsial dan memastikan bahwa setiap atribut non-kunci sepenuhnya bergantung pada primary key.

**3NF**



**PENEJELASAN :**

**1. Tabel Kendaraan (atas kiri):**

- Primary Key: No Polisi

- Atribut: Warna, Merek, Tahun

- Sudah memenuhi 3NF karena semua atribut non-kunci bergantung langsung pada primary key.

**2. Tabel Mekanik (atas kanan):**

- Primary Key: Mekanik ID

- Atribut: Nama Mekanik

- Sudah memenuhi 3NF karena hanya ada satu atribut non-kunci yang bergantung langsung pada primary key.

**3. Tabel Parts (kanan):**

- Primary Key: Kode Parts

- Atribut: Nama Parts, Harga

- Sudah memenuhi 3NF karena semua atribut non-kunci bergantung langsung pada primary key.

**4. Tabel Faktur (tengah):**

- Primary Key: No Faktur

- Atribut: Tanggal, No Polisi, Mekanik ID, Potongan

- Sudah memenuhi 3NF karena semua atribut non-kunci bergantung langsung pada primary key, dan tidak ada ketergantungan transitif.

**5. Tabel Detail Faktur (bawah):**

- Primary Key: Kombinasi No Faktur dan Kode Parts

- Atribut: Kuantum, Harga, Discount

- Sudah memenuhi 3NF karena semua atribut non-kunci bergantung langsung pada primary key gabungan.

**Penjelasan 3NF:**

1. Tidak ada pengulangan kelompok data (sudah 1NF).

2. Semua atribut non-kunci bergantung sepenuhnya pada primary key (sudah 2NF).

3. Tidak ada ketergantungan transitif (atribut non-kunci tidak bergantung pada atribut non-kunci lainnya).

Keuntungan dari struktur 3NF ini:

1. Menghindari redundansi data: Informasi disimpan sekali saja di tabel yang sesuai.

2. Konsistensi data: Perubahan data hanya perlu dilakukan di satu tempat.

3. Fleksibilitas: Mudah untuk menambah atau mengubah data tanpa mempengaruhi struktur keseluruhan.

**Hubungan antar tabel:**

- Tabel Faktur terhubung dengan Tabel Kendaraan melalui No Polisi.

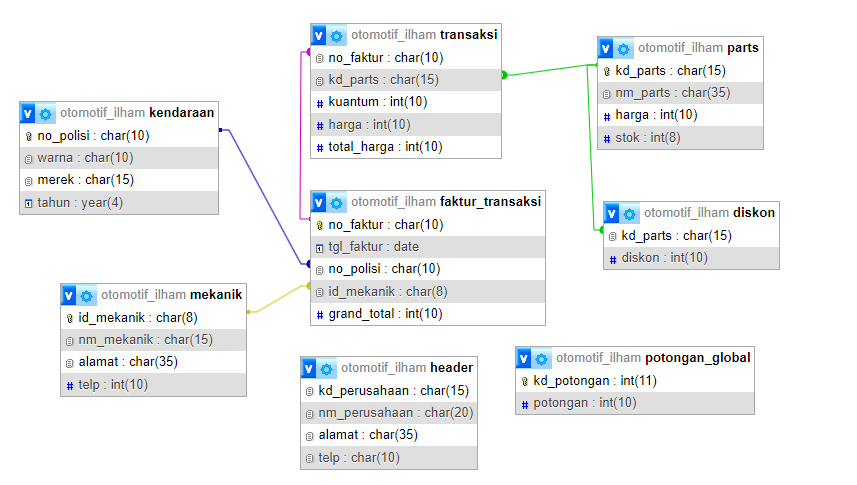
- Tabel Faktur terhubung dengan Tabel Mekanik melalui Mekanik ID.

- Tabel Detail Faktur terhubung dengan Tabel Faktur melalui No Faktur.

- Tabel Detail Faktur terhubung dengan Tabel Parts melalui Kode Parts.

Struktur ini memungkinkan untuk melacak layanan yang diberikan pada kendaraan tertentu, oleh mekanik tertentu, dengan parts yang digunakan, tanpa duplikasi data yang tidak perlu.

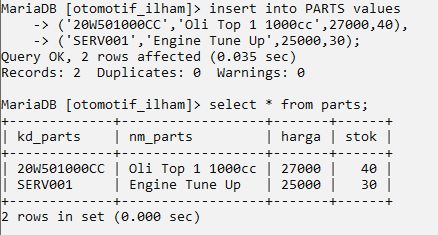
**TABEL-TABEL BESERTA RELASINYA 3NF**

****

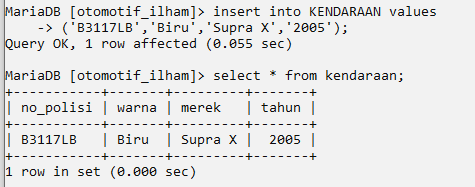
**Proses Penginputan Data :**

* **MASTER**

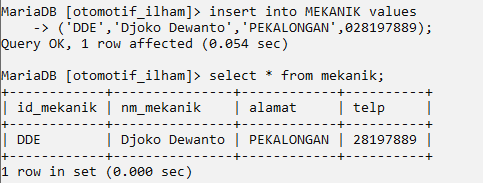
Tabel Parts di isi dua data/record



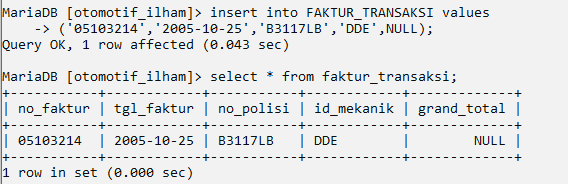
Tabel Kendaraan di isi satu record atau satu kendaraan



Tabel Mekanik contoh satu mekanik saja



Tabel Faktur Transaksi insert satu data karena akan melakukan 1 transaksi

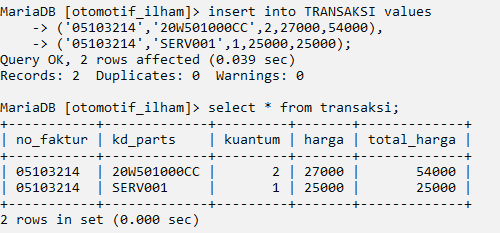


Grand\_total null karena belum input transaksi parts yang di beli.

* **TRANSAKSI**

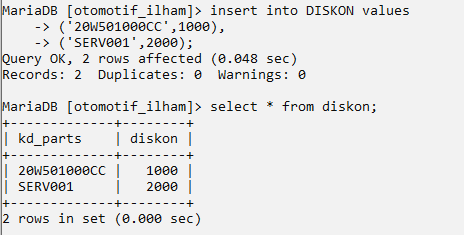
Kemudian kita melakukan transaksi pembelian 2 parts untuk service

di tabel transaksi.



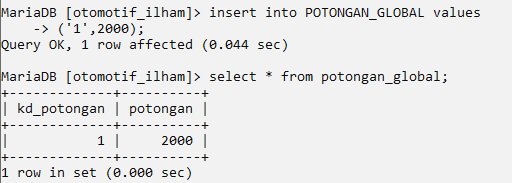
Nah dari transaksi di atas nanti akan kita hitung grand total nya.

Insert tabel diskon karena dari ke 2 parts tersebut memiliki diskon.

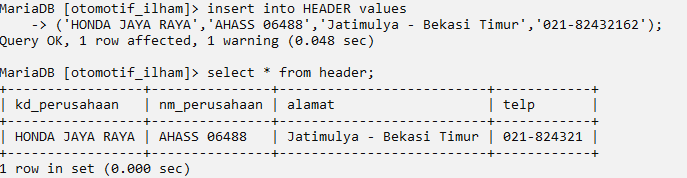


* **UTILITI**

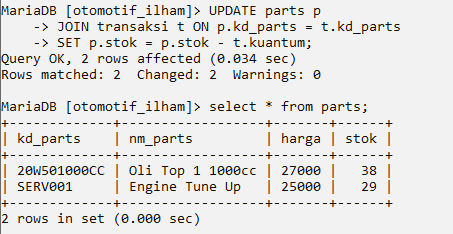
Insert tabel potongan\_global untuk memberikan potongan setiap transaksi



Insert tabel header untuk membuat template yang akan di gunakan di struk

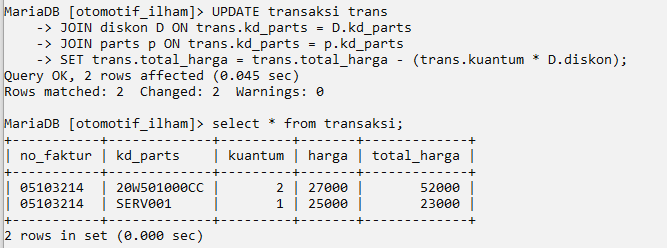


**#** Kemudian setelah melakukan transaksi harusnya stok parts berkurang kan sesuai dengan kuantum pembelian/transaksi. Maka dari itu perlu update record parts agar stok nya berkurang menggunakan rumus berikut :



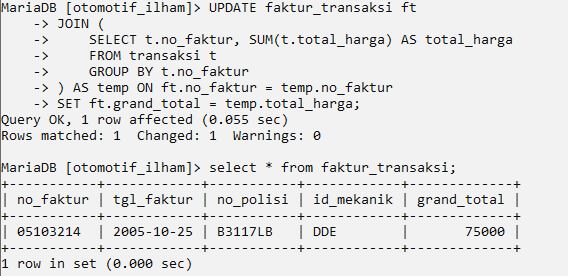
Stok sudah berkurang

**#** Selanjutnya di karenakan ada diskon dari ke 2 parts tersebut maka total harga dikurangi diskon dari masing-masing transaksi dengan menggunakan rumus berikut :



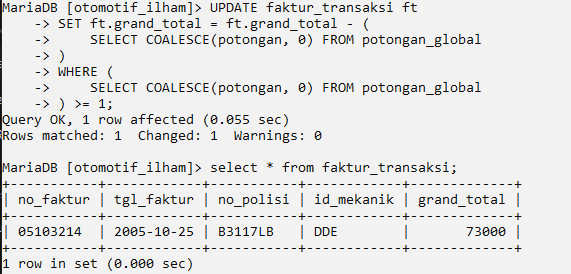
Total\_harga tanpa diskon dikurangi Kuantum \* diskon

**#** Setelah diskon kita akan update grand\_total dengan SUM.



Grand\_toal sudah terupdate dari sum(total\_harga).

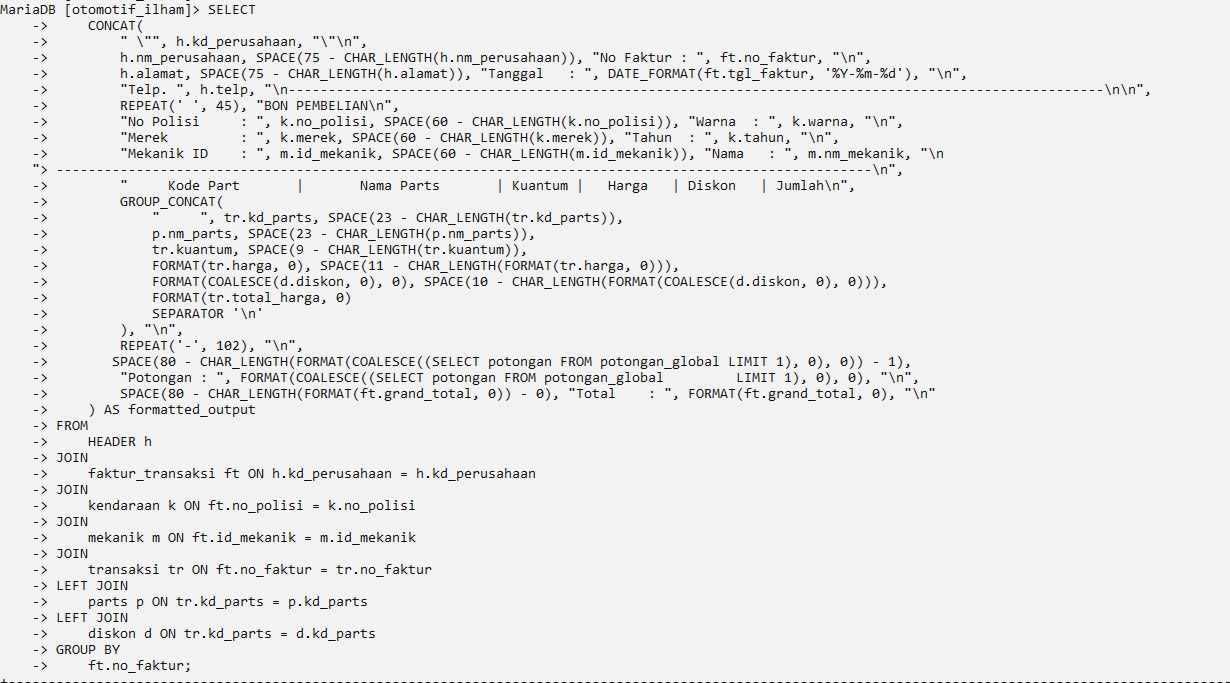
**#** Kemudian setelah grand\_total sudah di dapatkan, maka berikan update untuk mengurangi grand total dengan mengambil nilai potongan yang ada pada tabel potongan\_global dimana jika nilai potongan >= 1 maka grand\_total akan dikurangi.

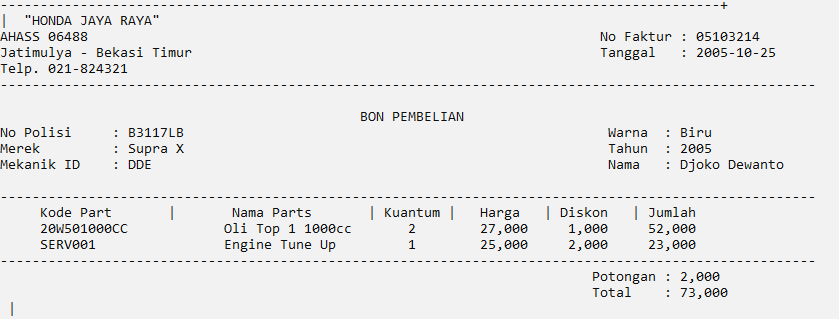


Terlihat grand\_total sudah terupdate nilainya yakni berkurang karena nilai potongan >= 1.

* Subquery (SELECT COALESCE(potongan, 0) FROM potongan\_global) digunakan untuk mengambil nilai potongan global dari tabel potongan\_global.
* Kondisi WHERE memastikan bahwa pengurangan grand\_total hanya dilakukan jika nilai potongan dari potongan\_global lebih besar atau sama dengan 1.

**OUTPUT CETAK STRUK**

****

****

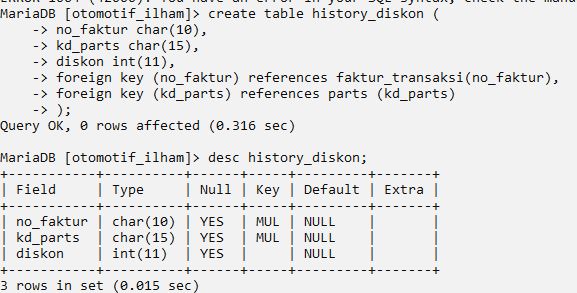
**Boyce–Codd normal form (BCNF or 3.5NF)**

Dari normalisasi tabel-tabel sampai ke 3NF, masih ada kekurangan dari sistem basis data tersebut. Yakni kita tidak tau history kapan dan berapa nilai diskon dan potongan.

Karena sewaktu-waktu nilai diskon dan potongan bisa berubah sesuai peraturan perusahaan, namun kita ingin tau history diskon dan potongan pada suatu transaksi di masa lalu.

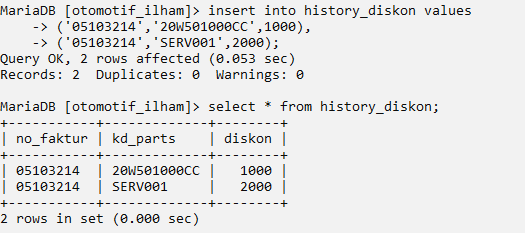
Maka dari itu untuk membuat sistem basis data tersebut menjadi BCNF atau tidak ada lagi data yang kosong dan tidak terjadi redudanci makan akan di buatkan 2 tabel lagi yakni history\_diskon dan history potongan yang bertujuan untuk meningkatakan integritas data.

**Membuat tabel history diskon** :

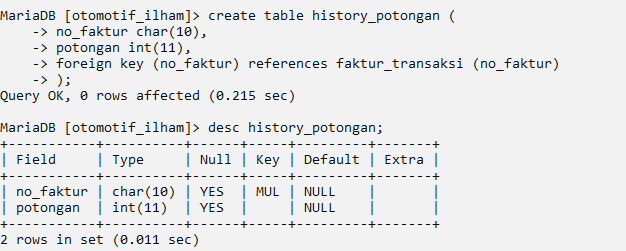


Berelasi dengan tabel faktur\_transaksi dimana terdapat tgl\_faktur (waktu kerjadian transaksi) dan tabel parts untuk menentukan part/produk yang di diskon.

**Kemudian insert 2 record untuk 2 parts yang di diskon dari transaksi di atas tadi.**

****

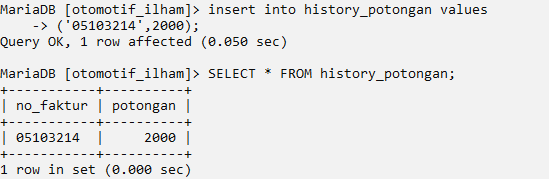
**Selain diskon kita juga perlu mengetahui history\_potongan.**

****

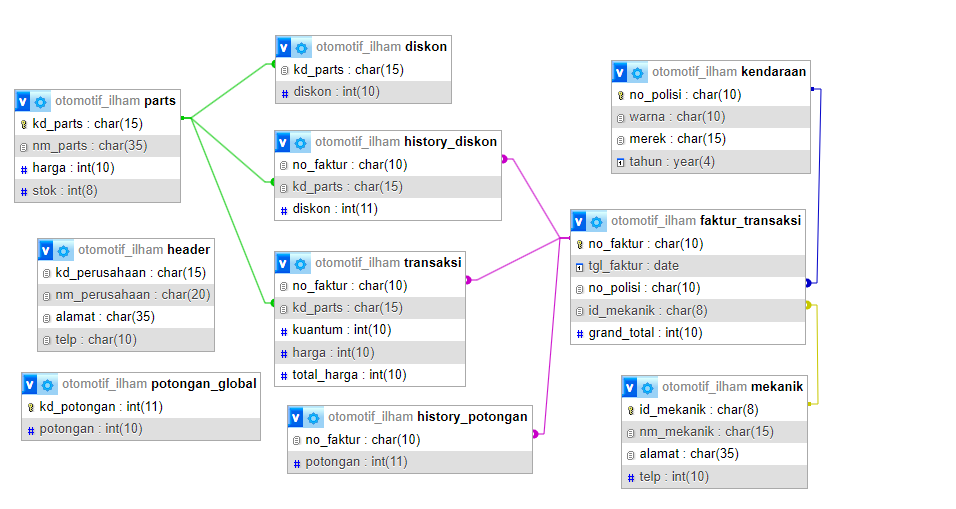
Berelasi dengan tabel faktur\_transaksi untuk mengetahui tgl\_faktur (waktu kejadian transaksi)

Potongan diambil dari nilai potongan global pada saat itu.

**Untuk itu kita insert data potongan sesuai transaksi diatas tadi.**

****

**Dengan demikian maka normalisasi sudah mencapai BCNF.**

****

**SELESAI**

1. Buatlah ERD dari Kasus di bawah ini :

**(TAMPILKAN KOMPONEN , KARDINALITAS DAN TAHAP  PEMBUATAN ERD)**

Sebuah usaha penitipan motor yang berada di dekat terminal bis antar  kota ingin membuat sistem penitipan motor dengan data :

a. Data pelanggan (penitip) terdiri dari data : no ktp, nama , alamat , telp  hp

b. Data petugas operator terdiri dari data : no pegawai, nama , alamat ,  no telp

c. Data harga titip perhari : kode kendaraan dan harga jasa titip per hari (catatan ada 3 jasa yaitu sepeda , sepeda motor, mobil) harganya  berbeda beda untuk tiap harinya menitip

d. Data transaksi terdiri dari : tgl nitip , tgl ambil kendaraan, jumlah  hari, kode kendaraan, no ktp, no kendaraan, biaya titip, no pegawai.

e. Data di atas bisa di tambahkan data pendukung yang lain jika anda  inginkan tetapi tidak boleh keluar dari jalur data penitipan motor.

**KOMPONEN :**

Tabel Pelanggan

|  |  |
| --- | --- |
| Field | Type |
| No\_ktp | Varchar(20) |
| nama | Varchar(100) |
| Alamat | text |
| Telp\_hp | Varchar(20) |

Tabel Petugas

|  |  |
| --- | --- |
| Field | Type |
| No\_pegawai | Varchar(20) |
| nama | Varchar(100) |
| Alamat | text |
| Telp\_hp | Varchar(20) |

Tabel Harga Tititp Perhari

|  |  |
| --- | --- |
| Field | Type |
| Kode\_kendaraan | Varchar(20) |
| Jenis\_kendaraan | Varchar(50) |
| Harga\_titip\_perhari | Int(11) |

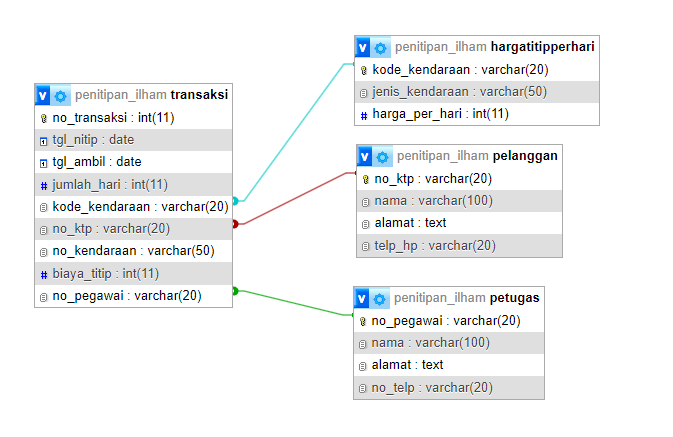
Tabel Transaksi

|  |  |
| --- | --- |
| Field | Type |
| No\_transaksi | Int(11) |
| Tgl\_nitip | date |
| Tgl\_ambil | date |
| Jumlah\_hari | Int(11) |
| Kode\_kendaraan | Varchar(20) |
| No\_ktp | Varchar(20) |
| No\_kendaraan | Varhchar(50) |
| Biaya\_titip | Int(11) |
| No\_pegawai | Varchar(20) |

**KARDINALITAS & ERD :**

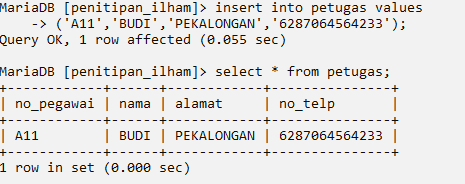
### 

1. **Pelanggan - Transaksi**
   * Kardinalitas: 1
   * Seorang pelanggan dapat melakukan banyak transaksi, tetapi satu transaksi hanya dilakukan oleh satu pelanggan.
2. **Petugas - Transaksi**
   * Kardinalitas: 1
   * Seorang petugas operator dapat menangani banyak transaksi, tetapi satu transaksi hanya ditangani oleh satu petugas.
3. **Harga Titip Perhari - Transaksi**
   * Kardinalitas: 1
   * Satu harga titip per hari berlaku untuk banyak transaksi, tetapi satu transaksi hanya berlaku untuk satu harga titip.

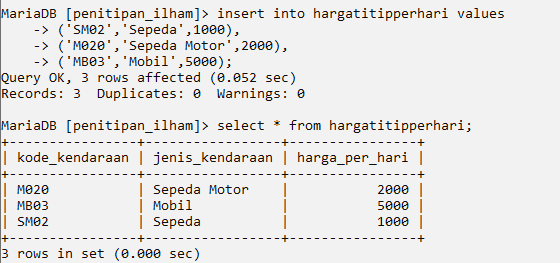
****

**INSERT DATA MELAKUKAN PENITIPAN :**

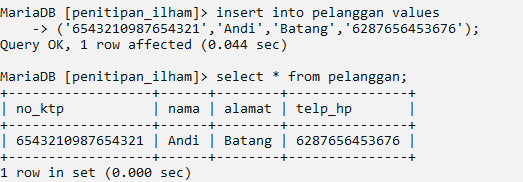
input data petugas dulu.

****

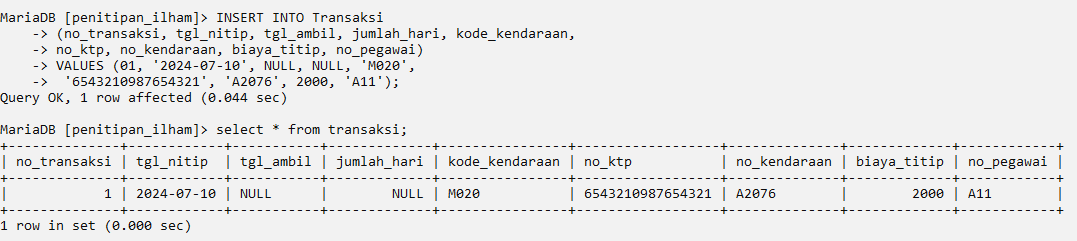
Input data harga penitipan perhari



Input data pelanggan **Andi** yang mau menitip kendaraan



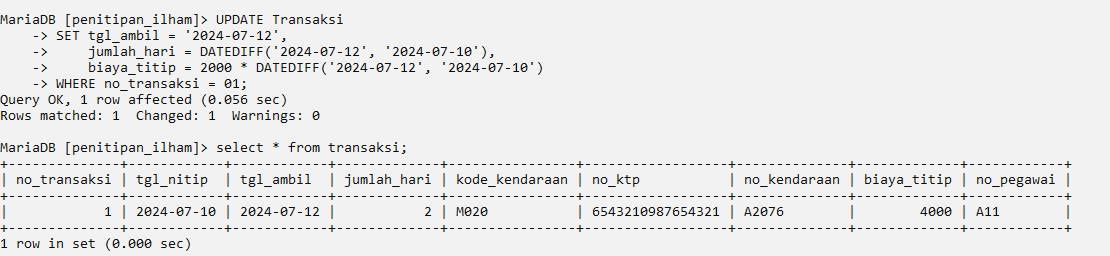
Setelah itu input transaksi penitipan



Kenapa tgl\_ambil dan jumlah\_hari null? Karena belum melakukan pengambilan jadi belum tau tgl\_ambil dan jumlah\_hari nya.

**UPDATE DATA MELAKUKAN PENGAMBILAN :**

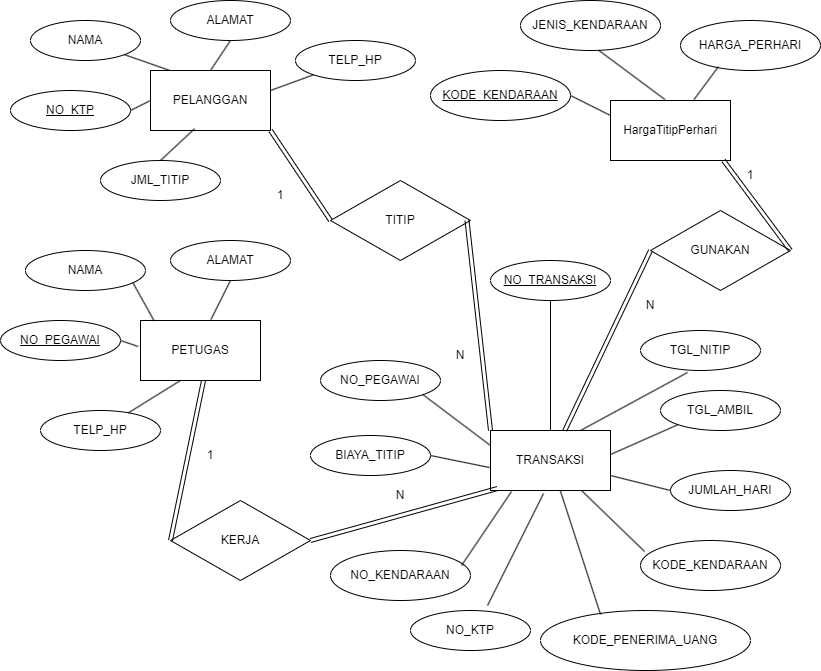
Kemudian setelah melakukan penitipan pelanggan akan mengambil sepeda motornya pada tanggal 12-07-2024, Maka menggunakan update data sebagai berikut :

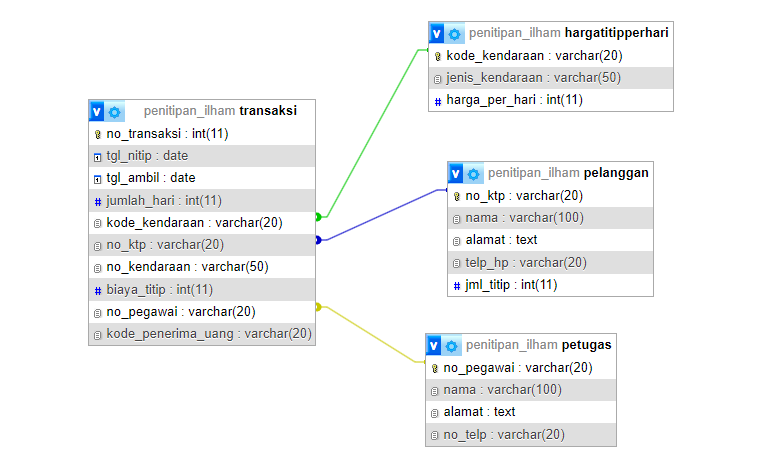


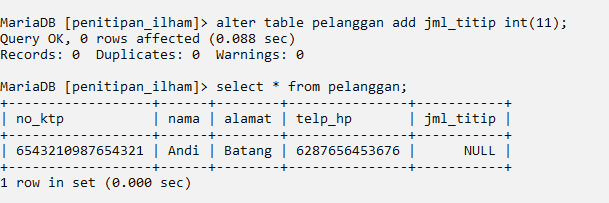
* **UPDATE Transaksi**: Menentukan tabel yang akan diupdate, yaitu Transaksi.
* **SET tgl\_ambil = '2024-07-12'**: Mengatur kolom tgl\_ambil untuk diupdate dengan nilai '12-07-2024'.
* **jumlah\_hari = DATEDIFF('2024-07-12', '2024-07-10')**: Menghitung selisih hari antara tanggal '12-07-2024' (tgl\_ambil) dan '10-07-2024' (tgl\_nitip) menggunakan fungsi DATEDIFF. Hasilnya akan disimpan ke kolom jumlah\_hari.
* **biaya\_titip = 2000 \* DATEDIFF('2024-07-12', '2024-07-10'):** Menghitung biaya titip berdasarkan *harga titip per hari* (2000 dalam contoh ini) dikali jumlah hari. Hasil perhitungan ini akan disimpan ke kolom biaya\_titip.
* **WHERE no\_transaksi = 01**: Membatasi operasi update hanya pada baris dengan no\_transaksi yang sama dengan 01.

**TAMBAHAN DATA PENDUKUNG :**

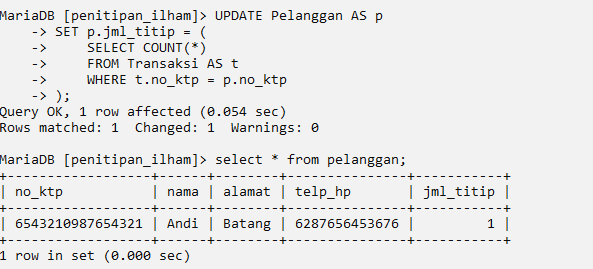
1. Untuk memberikan pelayanan yang berbeda kepada pelanggan dengan jumlah titip tertentu maka diperlukan atribut pendukung yakni jumlah titip yang akan di tambahkan ke tabel pelanggan. Jadi pelayanan tergauntng sebarapa banyak jumlah penitipan pelanggan.
2. Darti tabel-tabel di atas petugas/pegawai bekerja secara shit, Atau bergantian dengan waktu yang sudah di tentukan, misal si A dari pagi sampai sore si B dari sore sampe pagi. Nah dengan itu timbul sebuah masalah dimana perlu tau siapa pegawai yang menerima uang sedangkan 2 orang yang saling bergantian shift tidak terdata siapa yang menerima uang. Maka dari itu buat atribut baru untuk menyimpan siapa pegawai yang menerima uang dengan atribut beranama **kode\_penerima\_uang**  di tabel transaksi yang nilainya di ambil dari **no\_pegawai.**

****

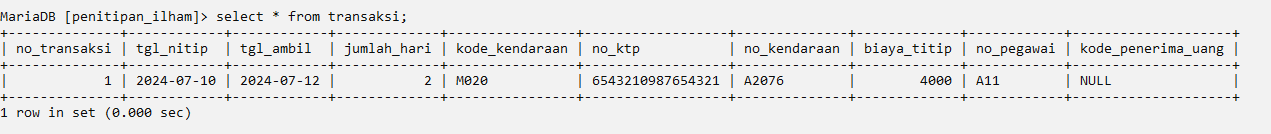




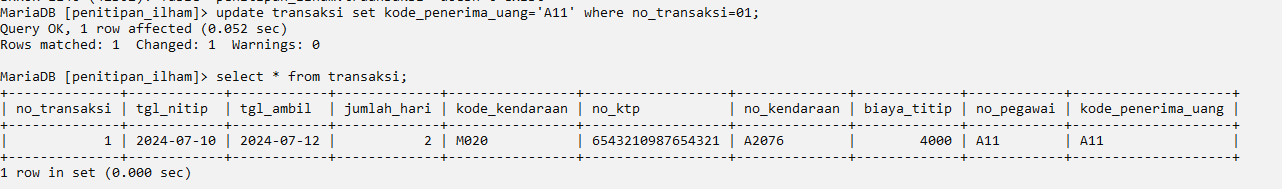
Setelah itu UPDATE jml\_titip berdasarkan jumlah record di tabel transaksi dengan no\_ktp yang sama:



Terlihat nilai jml\_titip sudah ada yakni benar 1 record sesuai dengan transaksi.

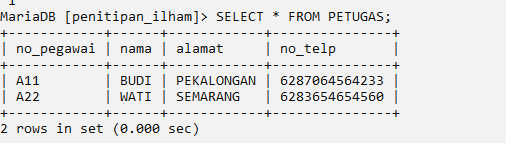


Nilai kode\_penerima\_uang null dulu, karena akan di update.

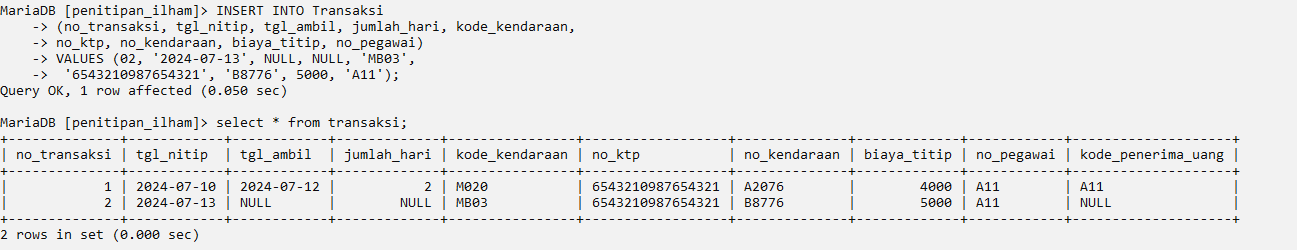


Nah bagaiamana jika pegawai yang menerima uang berbeda, apakah bisa? **Bisa,** input data petugas lagi.

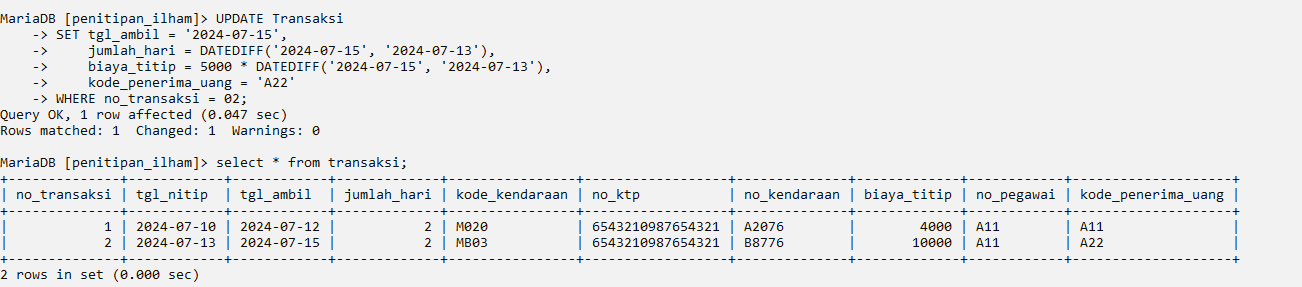




Melakukan Penitipan lagi dengan pegawai yang di titipi dan penerima uang berbeda pegawai.



Setelah itu melakukan pengambilan dengan UPDATE kolom tgl\_ambil, jumlah\_hari, biaya titip dan kode\_penerima\_pegawai.



**tgl\_ambil = '2024-07-15'**: Mengubah kolom tgl\_ambil menjadi tanggal '2024-07-15'. Ini mengindikasikan tanggal di mana kendaraan diambil kembali oleh pelanggan.

**jumlah\_hari = DATEDIFF('2024-07-15', '2024-07-13'):** Menghitung jumlah hari antara tanggal '2024-07-15' dan '2024-07-13', yaitu 2 hari. Hasil perhitungan ini disimpan di kolom jumlah\_hari. Hal ini menggambarkan berapa lama kendaraan tersebut dititipkan.

**biaya\_titip = 5000 \* DATEDIFF('2024-07-15', '2024-07-13')**: Menghitung biaya titip dengan mengalikan harga per hari (5000) dengan jumlah hari yang dihitung sebelumnya, yaitu 2 hari. Hasil perhitungan ini disimpan di kolom biaya\_titip. Biaya titip ini dikenakan sesuai dengan tarif yang berlaku untuk jenis kendaraan yang dititipkan.

**kode\_penerima\_uang = 'A22'**: Mengubah nilai kolom kode\_penerima\_uang menjadi 'A22'. Ini menunjukkan kode atau identifikasi untuk pihak yang menerima pembayaran biaya titip tersebut.

**WHERE no\_transaksi = 02**: Membatasi perubahan hanya untuk data dengan nomor transaksi 02. Ini memastikan bahwa perubahan hanya berlaku untuk transaksi spesifik dengan nomor 02 di dalam tabel Transaksi.

2. Terangkan Pemampatan data berikan contohnya

**Contoh Pemampatan Data:**

* Data Sebelum Pemampatan:

Tabel Pelanggan

|  |  |  |  |
| --- | --- | --- | --- |
| No\_ktp | Nama | Alamat | Telp\_hp |
| 123456 | John Doe | Jl. Mawar No. 12, Jakarta | 081234567890 |
| 789012 | Jane Smith | Jl. Melati No. 34, Bandung | 082345678901 |

Tabel Transaksi

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| id\_transaksi | tgl\_nitip | tgl\_ambil | jumlah\_hari | kode\_kendaraan | Kode\_kendaraaan | no\_kt | no\_kendaraan | biaya\_titip | no\_pegawai |
| 1 | 2024-07-0 | 2024-07-05 | 081234567890 | 4 | SEP | 123456 | B1234XYZ | 40000 | 1001 |
| 2 | 2024-07-02 | 2024-07-06 | 082345678901 | 4 | MOB | 789012 | B5678XYZ | 120000 | 1002 |

**Langkah Pemampatan Data:**

1. **Normalisasi Alamat:**
   * Pisahkan alamat menjadi beberapa bagian seperti jalan, kota, dan kode\_pos.
2. **Optimalisasi Data Transaksi:**
   * Gunakan indeks pada kolom yang sering dicari seperti no\_ktp dan kode\_kendaraan.
   * Harga titip per hari dapat disimpan dalam tabel terpisah untuk menghindari pengulangan harga pada setiap transaksi.

* Data Setelah Pemampatan:

Tabel Pelanggan: