

PROJECT DATABASE UNTUK SAINS DATA

Database Sistem Manajemen *Online Shop* Suku Cadang



**Disusun oleh:
Kelompok 3**

Abdul Muhaimin	2106703052
Adriaan Waleleng	2106725072
Diki Wahyudi	2106709131
Favian Sulthan Wafi	2106706205
Justin Martinus	2106724832

**PROGRAM STUDI STATISTIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS INDONESIA
2023**

DAFTAR ISI

DAFTAR ISI	1
1. Pendahuluan.....	2
2. Tujuan	3
3. Metode	3
3.1 Rancangan Database	3
3.2 Alur Kerja	5
4. Hasil dan Pembahasan	6
4.1 <i>Entity Relationship Diagram</i> (ERD).....	6
4.2 Tabel Relasional.....	10
4.3 Normalisasi	11
4.4 Simulasi.....	12
4.4.1 Simulasi CREATE dan READ.....	12
4.4.2 Simulasi INSERT, UPDATE, dan DELETE.....	22
4.5 Analisis.....	27
5. Kesimpulan.....	31
DAFTAR PUSTAKA.....	32

1. Pendahuluan

Online shop atau yang disebut dengan toko *online* adalah sistem belanja *online* yang mana pembeli dapat melihat harga dan jenis-jenis produk, menanyakan hal apapun terkait produk kepada penjual secara langsung, yang dapat dilakukan melalui LINE, Facebook, Instagram, Whatsapp, website, atau aplikasi khusus (*Online Shop, Market Place, E-Commerce. Apa Bedanya?*, n.d.). Di zaman modern sekarang ini, *online shop* telah banyak digunakan oleh berbagai orang dari kalangan remaja hingga dewasa. Saat ini, semakin banyak pebisnis yang berjualan melalui website, *e-commerce*, serta *marketplace* di Indonesia. Bukan hanya oleh pelaku UMKM saja, tapi bahkan berbagai macam *brand* besar pun turut ikut masuk serta berjualan di *marketplace* secara *online*. Dengan banyaknya bisnis serta customer dari bisnis tersebut yang terlibat dalam transaksi *online*, data dari kegiatan tersebut akan bertambah banyak setiap harinya. Data tersebut dapat membantu bisnis mengidentifikasi tren dan pola terkait pelanggan sehingga dapat memberikan gambaran strategi yang lebih tepat dalam meningkatkan loyalitas pelanggan. Oleh karena itu, dibutuhkan suatu sistem database untuk mengelola data tersebut agar data dapat digunakan. Dengan adanya sistem database, dapat memudahkan pemilik bisnis atau karyawan untuk mengakses dan mendapatkan data karena semua data terorganisasi dengan baik.

Dalam *mini project* ini, kelompok kami fokus pada *online shop* di bidang otomotif yaitu sebuah website yang dapat digunakan untuk transaksi suku cadang. Suku cadang adalah komponen dari mesin yang dicadangkan untuk perbaikan atau penggantian bagian kendaraan yang mengalami kerusakan (*Onderdil*, n.d.). Pengunjung website dapat melakukan transaksi ataupun tidak. Namun, kunjungan ke website tersebut akan tercatat ke dalam sistem database. Suku cadang yang disediakan pada website ini terdapat empat jenis yaitu bagian keelektrikan, bagian logam, bagian karet, dan bagian transmisi. Keempat jenis suku cadang tersebut memiliki suku cadangnya masing-masing.

Sistem database yang akan dibuat memiliki sistem *restock*. Setiap jenis suku cadang tersebut akan diurus oleh beberapa karyawan sesuai dengan jenis tugasnya masing-masing. Jika suku cadang habis, karyawan tersebut dapat memesan ke distributor yang berkaitan sehingga stok dari suku cadang tersebut dapat bertambah.

2. Tujuan

Tujuan dari pembuatan sistem database untuk *online shop* suku cadang ini yaitu sebagai berikut.

1. Mengetahui lebih dalam mengenai sistem transaksi dan *restocking* pada *online shop*.
2. Memudahkan dalam pengambilan informasi (*information retrieval*).
3. Meminimalkan redudansi data atau kesalahan dalam penambahan data yang rentan dilakukan dalam *restocking*.

3. Metode

3.1 Rancangan Database

Dalam pembuatan database ini, kelompok kami telah mengidentifikasi proses-proses yang terjadi dalam sistem manajemen *online shop* suku cadang. Hal ini dilakukan agar nantinya dapat ditentukan entitas-entitas dan hubungan antarentitas yang ada dalam sistem manajemen. Berikut beberapa proses utama dari sistem manajemen *online shop* suku cadang:

- Pelanggan perlu memiliki akun sebelum mengunjungi *website*
- Pelanggan dapat membeli berbagai jenis suku cadang dengan setiap pembelian akan tersimpan id pembelian, id pelanggan, tanggal pembelian, nomor produk, kuantitas, dan harga total pembelian.
- Pembelian yang dilakukan pelanggan akan diurus oleh karyawan.
- Suku cadang akan di-*restock* dengan membuat pesanan. Distributor akan mendapatkan riwayat pesanan yang dibuat dan akan tersimpan id pesanan, tanggal pesanan, nomor produk, kuantitas, dan harga total pesanan.

Dari proses di atas, dapat ditentukan entitas utama dari database sistem manajemen *online shop* suku cadang, yaitu sebagai berikut.

1. Pelanggan: merupakan rincian dari data pelanggan yang terdiri dari id, nama, e-mail, alamat, umur, tanggal lahir, dan nomor telepon pelanggan.
 - a. Akun: merupakan rincian dari data akun pelanggan yang terdiri dari *username*, dan *password* pelanggan.
2. Suku Cadang: merupakan rincian dari data produk suku cadang yang terdiri dari nomor, kode, nama, harga, kuantitas, dan harga jual produk.
 - a. Jenis: merupakan rincian dari data jenis produk suku cadang yang terdiri dari kode dan klasifikasi produk.
 - b. Pembelian: merupakan rincian dari data pembelian pelanggan yang terdiri dari id pembelian, id pelanggan, tanggal pembelian, nomor produk, kuantitas, dan harga total pembelian.

- c. Pesanan: merupakan daftar harga pembelian dari nomor produk yang terdiri dari nomor produk dan harga pembelian
3. Karyawan: merupakan rincian dari data karyawan yang terdiri dari id karyawan, nama karyawan, alamat karyawan, nomor telepon karyawan, kode produk dan tugas.
4. Distributor: merupakan rincian dari data distributor suku cadang yang terdiri dari id, nama, alamat, dan nomor telepon distributor.
 - a. Riwayat Belanja: merupakan rincian dari data pesanan untuk *me-restock* suku cadang yang terdiri dari id pesanan, tanggal pesanan, nomor produk, kuantitas pesanan, dan harga total pesanan.

Berdasarkan entitas tersebut, hubungan antarentitasnya yaitu sebagai berikut.

- Pelanggan-Akun (1:1): seorang pelanggan hanya memiliki satu akun dan satu akun hanya dimiliki oleh seorang pelanggan.
- Pelanggan-Suku Cadang (M:N): seorang pelanggan dapat membeli banyak suku cadang dan satu suku cadang dapat dibeli oleh banyak pelanggan.
- Suku Cadang-Jenis (N:1): satu suku cadang masuk ke salah satu jenis/klasifikasi dan satu jenis/klasifikasi dapat terdiri dari banyak suku cadang.
- Jenis-Karyawan (1:N): satu jenis suku cadang akan diurus oleh banyak karyawan dan seorang karyawan akan mengurus satu jenis suku cadang.
- Suku Cadang-Pesanan (1:1): satu suku cadang akan di-*restock* dengan membuat satu pesanan dan setiap pesanan yang dibuat hanya untuk merestock satu suku cadang.
- Pesanan-Distributor (1:1): setiap pesanan yang dibuat untuk *me-restock* suku cadang akan menghasilkan riwayat pesanan yang akan diterima oleh satu distributor dan satu distributor hanya menerima satu riwayat pesanan dari pesanan yang dibuat.

Batasan masalah untuk database yang dibuat yaitu sebagai berikut.

- Alamat pelanggan pada tabel Pelanggan diasumsikan sama dengan alamat pengiriman barang.
- Meskipun barang yang sama dipesan pada distributor yang berbeda, barang tersebut akan dianggap sama dan dimasukkan ke stok dari *no_prod* (nomor produk) yang sesuai. Misalkan akan dipesan suku cadang Piston pada Distributor A dan Distributor B. Walaupun distributornya berbeda, Piston dari kedua distributor tersebut diasumsikan sama sehingga kedua pesanan dari dua distributor tersebut sama-sama akan ditambahkan pada kuantitas Piston pada tabel Suku Cadang.
- Tempat penyimpanan suku cadang berada pada tempat yang sama, tidak pada daerah yang berbeda.

3.2 Alur Kerja

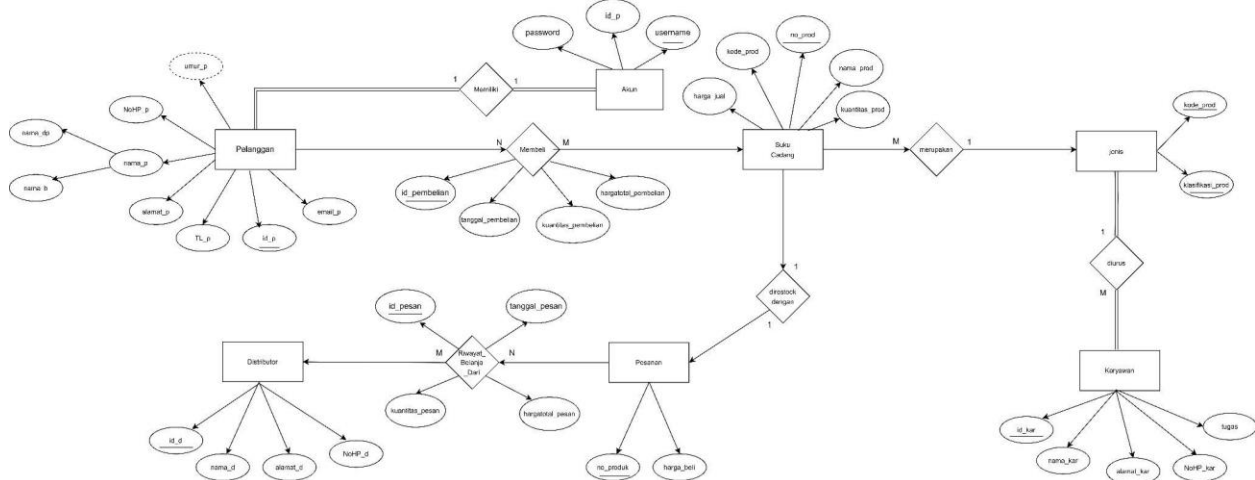
Berikut adalah alur kerja yang digunakan kelompok kami dalam membuat database sistem manajemen *online shop* suku cadang.

1. Mengidentifikasi proses-proses dalam sistem manajemen *online shop* suku cadang. Pelanggan mengunjungi website menggunakan akun yang terdaftar, setelah itu baru bisa membeli berbagai jenis produk suku cadang yang tersedia.
 - Karyawan akan mengurus pembelian yang dilakukan pelanggan serta melakukan restock produk dengan membuat pesanan ke distributor.
2. Menentukan entitas dan atribut.
 - Entitas terdiri dari pelanggan, suku cadang, karyawan, distributor akun, jenis, dan pesanan.
3. Menentukan *primary key* dan *foreign key* dari setiap entitas
 - Menentukan *primary key* dan *foreign key* dari entitas yang telah dibuat untuk menggambarkan hubungan antarentitas.
4. Pembuatan ER Diagram.
 - Pembuatan ERD menggunakan software untuk memberikan representasi visual dari desain database.
5. Pembuatan tabel relasional.
 - Mengubah ERD menjadi tabel relasional menggunakan software untuk melihat lebih jelas hubungan antarentitas.
6. Menyiapkan data.
 - Membuat data dummy secara manual dan dengan bantuan Python yang akan dimasukkan ke dalam database. Untuk harga pembelian suku cadang, kami mencari informasi harga tersebut dari *marketplace* seperti Shopee, Tokopedia, dan lain-lain. Data yang lain sebagian kami buat secara manual dan sebagiannya lagi kami *random* menggunakan Python. Berikut ini link Google Colaboratory yang kami gunakan untuk *me-random* data. https://colab.research.google.com/drive/1w1OxSpVT4ODKlq_H9VFc-E5CzZzzkp0z#scrollTo=Ynf7TaXESwD6
7. Menuliskan kode dan melakukan simulasi serta analisis.
 - Kode yang akan digunakan untuk simulasi adalah CREATE, INSERT, UPDATE, dan DELETE.
 - Kode yang akan digunakan untuk analisis adalah JOIN, ORDER BY, dan GROUP BY.
8. Membuat kesimpulan.
 - Membuat kesimpulan dari hasil simulasi dan analisis.

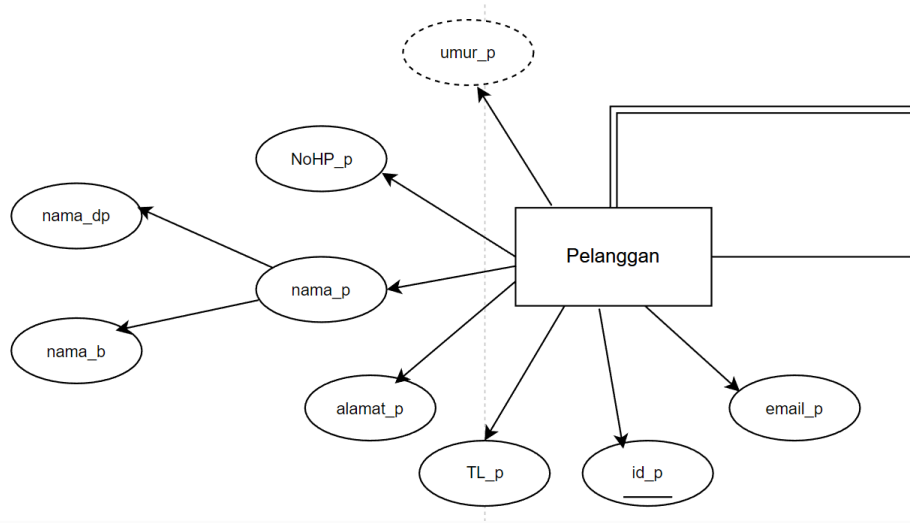
4. Hasil dan Pembahasan

4.1 Entity Relationship Diagram (ERD)

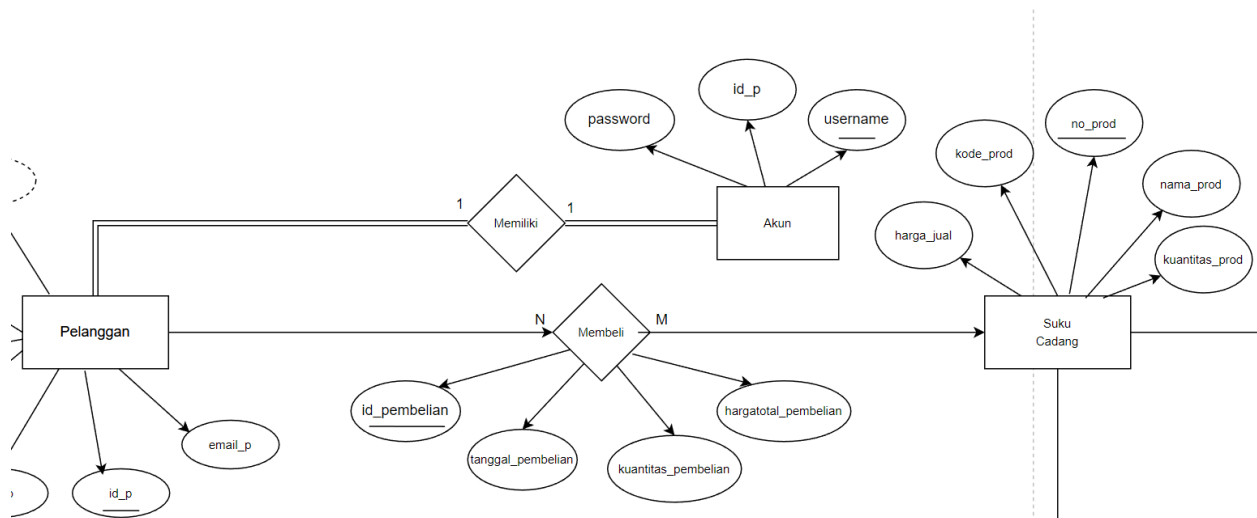
Dari rancangan database pada bagian 3.2, ERD untuk database ini, yaitu sebagai berikut.



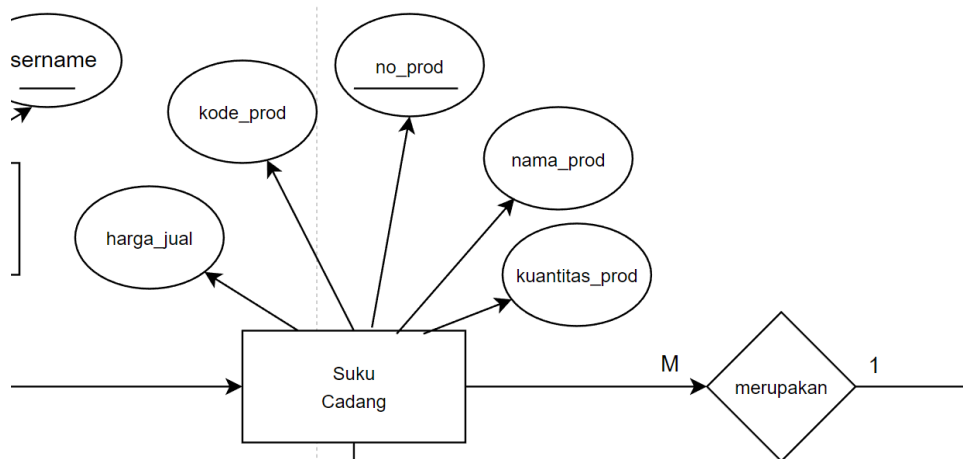
Penjelasan akan dibagi menjadi bagian-bagian tertentu pada ERD tersebut.



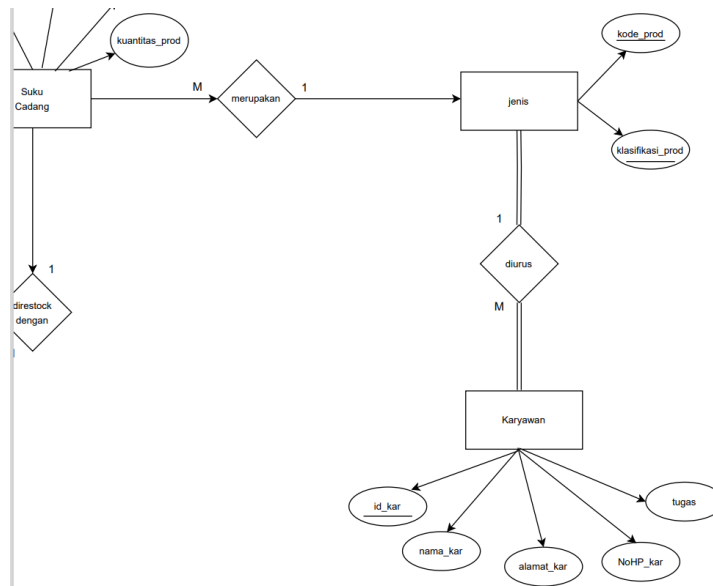
- Pada sistem manajemen *online shop* terdapat berbagai macam **Pelanggan** yang memiliki atribut **umur_p**, **NoHP_p**, **nama_p** (**nama_dp** dan **nama_b**), **alamat_p**, **TL_p**, **id_p**, dan **email_p**. Disini kami mengasumsikan bahwa setiap pelanggan hanya dapat memiliki satu email dan satu nomor telepon. **Catatan:** p = Pelanggan, no/No = Nomor, HP = *Handphone* (Telepon), dp = Depan, b = Belakang, dan TL = Tanggal Lahir.



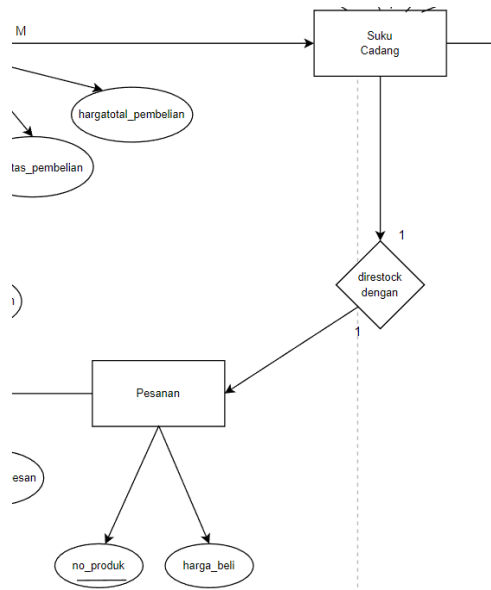
- Relasi **Pelanggan** dengan akun ialah *one to one*, yaitu setiap pelanggan hanya dapat memiliki satu akun di mana entitas akun memiliki atribut yang terdiri dari **password**, **id_p**, dan **username** (*primary key*).
- Relasi **Pelanggan** dengan **Suku Cadang** adalah **Membeli** dengan tipe hubungan *many to many*. Karena relasi ini *many to many*, butuh dibuat tabel untuk relasi Membeli. Tabel ini berisi **id_pembelian** sebagai *primary key* dan **tanggal_pembelian**. Pada relasi ini terdapat atribut **kuantitas_pembelian** dan **hargatotal_pembelian** yang diperhatikan lebih khusus karena kuantitas pembelian secara otomatis membuat kuantitas produk pada tabel **Suku Cadang** berkurang dan **hargatotal_pembelian** merupakan hasil perkalian dari **kuantitas_pembelian** pada tabel **Membeli** dan **harga_jual** dari tabel **Suku Cadang**. Selain itu, terdapat juga atribut tanggal pembelian yang terurut.



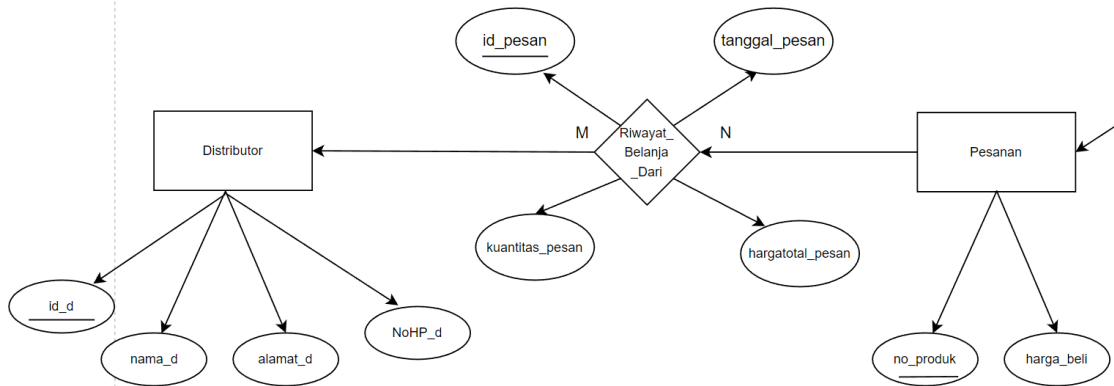
- Banyak **Pelanggan** dapat membeli berbagai macam **Suku Cadang** yang mana entitas **Suku Cadang** memiliki atribut **harga_jual**, **kode_prod**, **no_prod** (*primary key*), **nama_prod**, dan **kuantitas_prod**. Catatan: prod = Produk.



- Relasi tabel **Suku Cadang** dengan **jenis** adalah *many to one*, yaitu beberapa produk **Suku Cadang** dikelompokkan menjadi satu jenis (kelistrikan, transmisi, karet, dan logam) berdasarkan kode produknya. Tabel **jenis** memiliki atribut **kode_prod** (*primary key*) dan **klasifikasi_prod** (*primary key*).
- Tabel **Jenis** memiliki relasi *one to many* dengan tabel **Karyawan** karena cara pembagian tugas untuk karyawan dibagikan ke per klasifikasi produk. Misalnya, karyawan A memiliki tugas pengemasan, berarti dia hanya mengemas produk yang memiliki **klasifikasi_prod** berlabel 1 saja misalnya, tidak mengemas untuk **klasifikasi_prod** berlabel 2. Ini berarti setiap klasifikasi produk memiliki banyak karyawan untuk mengurus hanya klasifikasi tersebut saja, tetapi tiap karyawan hanya mengurus salah satu klasifikasi produk saja. Tabel **Karyawan** sendiri memiliki atribut **id_kar**, **nama_kar**, **alamat_kar**, **NoHP_kar**, dan **tugas**. Atribut tugas adalah rincian lebih lanjut tugas masing-masing karyawan dalam klasifikasi yang mereka urus. **Catatan:** kar = karyawan.



- Produk **Suku Cadang** dapat di-restock dengan **Pesanan** yang memiliki relasi *one to one* sehingga produk yang telah menipis, maka kita dapat me-restock-nya dengan melakukan pesanan. Tabel **Pesanan** memiliki atribut **no_produk** (*primary key*) dan **harga_beli**.

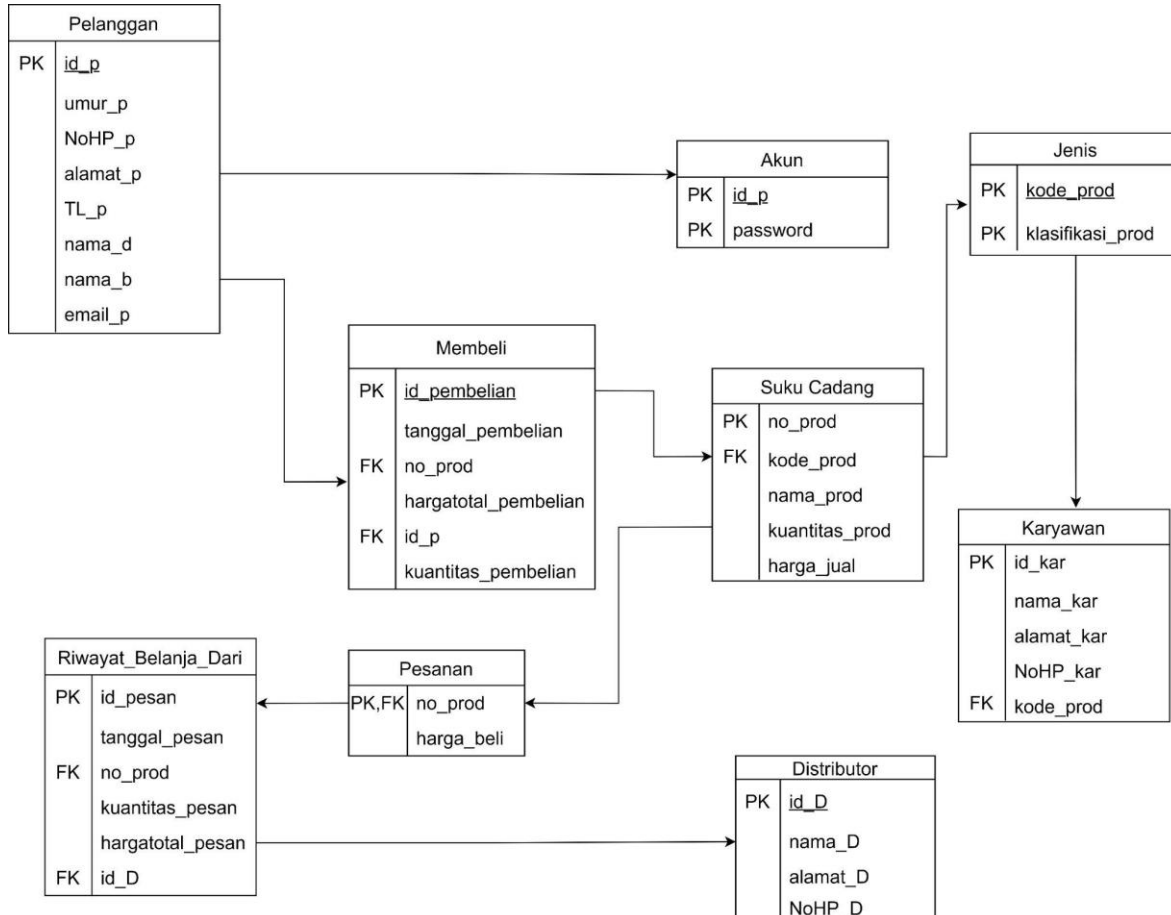


- Tabel **Pesanan** dengan **Distributor** memiliki relasi **Riwayat_Belanja_Dari** dengan hubungan *many to many* sehingga dibutuhkan tabel baru untuk relasi. Tabel **Riwayat_Belanja_Dari** memiliki atribut **id_pesan** sebagai *primary key* dan **tanggal_pesan** yang terurut. Selain itu, mirip dengan tabel **Membeli**, atribut **kuantitas_pesan** dan **hargatotal_pesan** pada tabel **Riwayat_Belanja_Dari** perlu lebih diperhatikan. Atribut **kuantitas_pesan** juga memengaruhi kuantitas pada tabel **Suku Cadang**, stok pada tabel **Suku Cadang** akan langsung menambah sesuai jumlah kuantitas pesanan yang dilakukan. Sementara itu, **hargatotal_pesan** merupakan hasil perkalian dari **kuantitas_pesan** pada tabel **Riwayat_Belanja_Dari** dengan **harga_beli** dari tabel **Pesanan**.

- Tabel **Distributor** memiliki atribut **id_d** sebagai *primary key*, **nama_d**, **alamat_d**, dan **NoHP_d** sebagai informasi pelengkap. **Catatan:** d = distributor.

4.2 Tabel Relasional

Berdasarkan ERD yang telah dibuat pada bagian 4.1, didapatkan tabel relasional sebagai berikut.



Foreign Key (FK) berfungsi untuk menghubungkan antarentitas. Misalnya, dengan menggunakan FK kami dapat melihat pelanggan mana membeli suku cadang apa saja. Penggunaan FK tergantung pada jenis relasi antar entitas (*one-to-one*, *one-to-many*, *many-to-many*)

Untuk hubungan *many to many* dibuat tabel terpisah untuk yang menjadi relasi mereka. Misalnya, dalam kasus ini, pelanggan-membeli-suku cadang (pelanggan bisa membeli banyak suku cadang) dapat dilihat membeli ada tabel tersendiri di mana *primary key* (PK) dari pelanggan dan suku cadang menjadi *foreign key* (FK) membeli. Contoh lain adalah pesanan ke riwayat belanja dari ke distributor.

Untuk hubungan *one-to-many*, PK dari yang menjadi tabel *one* harus menjadi FK pada tabel yang *many*. Dalam kasus ini misalnya jenis ke karyawan terlihat PK dari tabel jenis yaitu kode_prod menjadi FK di tabel karyawan

Untuk hubungan *one-to-one*, PK dari salah satu entitas menjadi FK disalah satunya. Dalam kasus ini misalnya pelanggan ke akun, PK pelanggan yaitu id_p menjadi FK di akun.

4.3 Normalisasi

Pada bagian ini, akan dicek apakah tabel Pembelian dan tabel Riwayat Belanja memenuhi sampai bentuk normal ke berapa dan apakah memerlukan normalisasi atau tidak. Kedua tabel tersebut penting untuk dicek normalisasinya karena melibatkan banyak suku cadang pada tabelnya. Tujuan dilakukan normalisasi yaitu untuk menghilangkan dan mengurangi redudansi data dan memastikan dependensi data secara tepat (data berada pada tabel yang tepat). Kedua tabel tersebut akan sering bertambah datanya karena pesanan dari pelanggan (tabel Pembelian) atau *restock* yang dilakukan karyawan (tabel Riwayat Belanja). Oleh karena itu, perlu dilakukan pengecekan normalisasi, agar tidak terdapat banyak redudansi data pada tabel tersebut.

- Tabel Pembelian
 - 1NF (*First Normal Form*): Tabel Pembelian tidak memiliki atribut yang bernilai jamak. Orang yang membeli banyak produk akan tercatat sesuai pada no_prod dalam baris yang berbeda. Setiap *record* unik karena terdapat id_pembelian sebagai primary key.
 - 2NF (*Second Normal Form*): Tabel Pembelian telah memenuhi 1NF. Tabel ini hanya memiliki satu *primary key*. Semua atribut *non-primary key* bergantung pada satu *primary key* tersebut sehingga tidak ada ketergantungan parsial.
- Tabel Riwayat Belanja
 - 1NF (*First Normal Form*): tabel Riwayat Belanja sudah dalam 1NF tidak terdapat *multivalued* pada semua atributnya. Tabel Riwayat Belanja sudah memiliki *primary key* yaitu id_pesanan.
 - 2NF (*Second Normal Form*): Tabel pembelian telah memenuhi 2NF. Tabel pelanggan tidak memiliki ketergantungan parsial karena semua atribut *non-primary key* bergantung pada id_pesanan (*primary key*).

Karena kedua tabel tersebut telah memenuhi sampai 2NF, maka tidak perlu dibuat pemisahan atribut atau membuat tabel baru pada masing-masing kedua tabel tersebut.

4.4 Simulasi

4.4.1 Simulasi CREATE dan READ

Pada tahap ini, akan dilakukan simulasi pembuatan seluruh tabel yang diperlukan dan masing-masing record dari setiap tabel tersebut. Perintah *query* yang digunakan dalam membuat tabel baru adalah CREATE TABLE dan untuk membuat record dari setiap tabelnya digunakan perintah INSERT TO. Selain itu, diperlukan perintah CREATE TRIGGER untuk mendapat umur pelanggan secara otomatis dari tanggal lahir, penambahan serta pengurangan kuantitas produk pada tabel Suku Cadang melalui pembelian dan pemesanan produk, dan menghitung total harga dari pembelian dan pemesanan produk. Berikut simulasi yang dilakukan untuk membuat tabel baru, *trigger* yang digunakan, dan *record* dari setiap tabel.

- Membuat Tabel Baru

Pertama, akan dibuat tabel pelanggan dan akun dengan atribut-atributnya menggunakan perintah CREATE TABLE.

```
1  /* Creating Table */
2
3  CREATE TABLE PELANGGAN (
4      id_p INTEGER PRIMARY KEY,
5      nama_dp TEXT,
6      nama_b TEXT,
7      NoHP_p INTEGER,
8      alamat_p TEXT,
9      email_p TEXT,
10     TL_p TEXT,
11     umur_p INTEGER
12 );
13
14 CREATE TABLE AKUN (
15     username TEXT PRIMARY KEY,
16     password TEXT,
17     id_p INTEGER,
18     FOREIGN KEY(id_p) REFERENCES PELANGGAN(id_p)
19 );
20
```

Selanjutnya, akan dibuat tabel karyawan, pesanan, dan distributor dengan atribut-atributnya menggunakan perintah CREATE TABLE.

```

CREATE TABLE KARYAWAN (
    id_kar INTEGER PRIMARY KEY,
    nama_kar TEXT,
    alamat_kar TEXT,
    NoHP_kar TEXT,
    kode_prod INTEGER,
    tugas TEXT,
    FOREIGN KEY(kode_prod) REFERENCES JENIS(kode_prod)
);

CREATE TABLE PESANAN (
    no_prod INTEGER PRIMARY KEY,
    harga_beli NUMERIC,
    FOREIGN KEY (no_prod) REFERENCES SUKU_CADANG (no_prod)
);

CREATE TABLE DISTRIBUTOR (
    id_d INTEGER PRIMARY KEY,
    nama_d TEXT,
    alamat_d TEXT,
    NoHP_d INTEGER
);

```

Terakhir, akan dibuat tabel jenis, suku cadang, membeli, dan riwayat_belanja_dari dengan atribut-atributnya menggunakan perintah CREATE TABLE.

```

21 CREATE TABLE JENIS (
22     kode_prod INTEGER PRIMARY KEY AUTOINCREMENT,
23     klasifikasi_prod TEXT
24 );
25 CREATE TABLE SUKU_CADANG (
26     no_prod INTEGER PRIMARY KEY AUTOINCREMENT,
27     kode_prod INTEGER,
28     nama_prod TEXT,
29     kuantitas_prod INTEGER,
30     harga_jual NUMERIC,
31     FOREIGN KEY (kode_prod) REFERENCES JENIS(kode_prod)
32 );
33 CREATE TABLE MEMBELI (
34     id_pembelian INTEGER PRIMARY KEY AUTOINCREMENT,
35     tanggal_pembelian TEXT,
36     no_prod INTEGER,
37     hargatotal_pembelian NUMERIC,
38     id_p INTEGER,
39     kuantitas_pembelian TEXT,
40     FOREIGN KEY (id_p) REFERENCES PELANGGAN(id_p),
41     FOREIGN KEY (no_prod) REFERENCES SUKU_CADANG(no_prod) ON DELETE CASCADE ON UPDATE CASCADE
42 );
43
44 CREATE TABLE RIWAYAT_BELANJA_DARI (
45     id_pesan INTEGER PRIMARY KEY AUTOINCREMENT,
46     tanggal_pesan TEXT,
47     no_prod INTEGER,
48     kuantitas_pesan INTEGER,
49     hargatotal_pesanan NUMERIC,
50     id_d INTEGER,
51     FOREIGN KEY (no_prod) REFERENCES PESANAN (no_prod) ON DELETE CASCADE ON UPDATE CASCADE,
52     FOREIGN KEY (id_d) REFERENCES DISTRIBUTOR (id_d)
53 );

```

- Membuat TRIGGER

Pertama, akan dibuat *trigger* untuk menghitung umur pelanggan berdasarkan tanggal lahir menggunakan perintah CREATE TRIGGER

```

79  /* Creating Trigger */
80  /* Age calculation in TABLE PELANGGAN */
81
82  CREATE TRIGGER calculate_age_trigger AFTER INSERT ON PELANGGAN
83  BEGIN
84      UPDATE PELANGGAN
85      SET umur_p = (strftime('%Y', 'now') - strftime('%Y', TL_p)) -
86      CASE WHEN strftime('%m-%d', 'now') < strftime('%m-%d', TL_p)
87      THEN 1 ELSE 0 END
88      WHERE id_p = NEW.id_p;
89  END;
90
91  CREATE TRIGGER update_age_trigger AFTER UPDATE OF TL_p ON PELANGGAN
92  BEGIN
93      UPDATE PELANGGAN
94      SET umur_p = (strftime('%Y', 'now') - strftime('%Y', NEW.TL_p)) -
95      CASE WHEN strftime('%m-%d', 'now') < strftime('%m-%d', NEW.TL_p)
96      THEN 1 ELSE 0 END
97      WHERE id_p = NEW.id_p;
98  END;

```

Kemudian, akan dibuat *trigger* untuk memperbarui kuantitas tiap pembelian pelanggan atau membeli dari distributor menggunakan perintah CREATE TRIGGER.

```

101  /* Updating quantity available after restocking and order in TABLE SUKU_CADANG */
102
103
104  CREATE TRIGGER update_quantity_available
105  AFTER INSERT ON MEMBELI
106  FOR EACH ROW
107  BEGIN
108      UPDATE SUKU_CADANG
109      SET kuantitas_prod = kuantitas_prod - NEW.kuantitas_pembelian
110      WHERE no_prod = NEW.no_prod;
111  END;
112
113  CREATE TRIGGER update_quantity_available2
114  AFTER INSERT ON RIWAYAT_BELANJA_DARI
115  FOR EACH ROW
116  BEGIN
117      UPDATE SUKU_CADANG
118      SET kuantitas_prod = kuantitas_prod + NEW.kuantitas_pesan
119      WHERE no_prod = NEW.no_prod;
120  END;
121
122  /* Adding the total price for restocking and order in TABLE MEMBELI and TABLE RIWAYAT_BELANJA_DARI */
123
124  CREATE TRIGGER update_total_price
125  AFTER INSERT ON MEMBELI
126  FOR EACH ROW
127  BEGIN
128      UPDATE MEMBELI
129      SET hargatotal_pembelian = (SELECT kuantitas_pembelian * SUKU_CADANG.harga_jual FROM SUKU_CADANG
130      WHERE SUKU_CADANG.no_prod = NEW.no_prod)
131      WHERE id_pembelian = NEW.id_pembelian;
132  END;
133
134  CREATE TRIGGER update_total_price2
135  AFTER UPDATE ON MEMBELI
136  FOR EACH ROW
137  BEGIN
138      UPDATE MEMBELI
139      SET hargatotal_pembelian = (SELECT kuantitas_pembelian * SUKU_CADANG.harga_jual FROM SUKU_CADANG
140      WHERE SUKU_CADANG.no_prod = NEW.no_prod)
141      WHERE id_pembelian = NEW.id_pembelian;
142  END;

```

Terakhir, akan dibuat *trigger* untuk menghitung total harga dari pembelian dan pemesanan produk menggunakan perintah CREATE TRIGGER.

```

CREATE TRIGGER update_total_price
AFTER INSERT ON MEMBELI
FOR EACH ROW
BEGIN
    UPDATE MEMBELI
    SET hargatotal_pembelian = (SELECT kuantitas_pembelian * SUKU_CADANG.harga_jual FROM SUKU_CADANG WHERE SUKU_CADANG.no_prod = NEW.no_prod)
    WHERE id_pembelian = NEW.id_pembelian;
END;

CREATE TRIGGER update_total_price2
AFTER UPDATE ON MEMBELI
FOR EACH ROW
BEGIN
    UPDATE MEMBELI
    SET hargatotal_pembelian = (SELECT kuantitas_pembelian * SUKU_CADANG.harga_jual FROM SUKU_CADANG WHERE SUKU_CADANG.no_prod = NEW.no_prod)
    WHERE id_pembelian = NEW.id_pembelian;
END;

```

```

CREATE TRIGGER update_total_price3
AFTER INSERT ON RIWAYAT_BELANJA_DARI
FOR EACH ROW
BEGIN
    UPDATE RIWAYAT_BELANJA_DARI
    SET hargatotal_pesanan = (SELECT kuantitas_pesanan * PESANAN.harga_beli FROM PESANAN WHERE PESANAN.no_prod = NEW.no_prod)
    WHERE id_pesanan = NEW.id_pesanan;
END;

CREATE TRIGGER update_total_price4
AFTER UPDATE ON RIWAYAT_BELANJA_DARI
FOR EACH ROW
BEGIN
    UPDATE RIWAYAT_BELANJA_DARI
    SET hargatotal_pesanan = (SELECT kuantitas_pesanan * PESANAN.harga_beli FROM PESANAN WHERE PESANAN.no_prod = NEW.no_prod)
    WHERE id_pesanan = NEW.id_pesanan;
END;

```

- Membuat *Record* dari Setiap Tabel

Selanjutnya, akan dimasukkan seluruh *record* dari data dummy yang telah dibuat ke dalam tabel yang telah dibentuk sebelumnya menggunakan perintah INSERT INTO.

1. Tabel Pelanggan

```

174 INSERT INTO PELANGGAN (id_p, nama_dp, nama_b, alamat_p, TL_p, email_p, NoHP_p) VALUES
175 (1000001, 'Stacy', 'Wagner', 'Jalan Aipda Karel Satsuit Tubun, Jakarta', '1958-01-18', 'Stacy9341@gmail.com', 6284219257513),
176 (1000002, 'Colleen', 'Smith', 'Jalan Asia Afrika, Bandung', '1975-03-25', 'Colleen9673@gmail.com', 6284550522882),
177 (1000003, 'Timothy', 'Wells', 'Jalan Pahlawan Bedul, Makassar', '1979-07-22', 'Timothy8773@gmail.com', 6282399105057),
178 (1000004, 'Frank', 'Smith', 'Jalan Mawar, Jakarta', '1952-04-30', 'Frank920@gmail.com', 6284491079635),
179 (1000005, 'Adriana', 'Galloway', 'Jalan Banteng, Semarang', '1961-11-30', 'Adriana718@gmail.com', 6283542420250),
180 (1000006, 'Jill', 'Wagner', 'Jalan Formulae, Jakarta', '1976-07-23', 'Jill11465@gmail.com', 6286535804163),
181 (1000007, 'Christy', 'Cox', 'Jalan Bintang, Jakarta', '1991-06-22', 'Christy4220@gmail.com', 6284682498686),
182 (1000008, 'Michelle', 'Fitzpatrick', 'Jalan Matahari, Jayapura', '1963-03-28', 'Michelle3992@gmail.com', 6285533851992),
183 (1000009, 'Jessica', 'Grant', 'Jalan Jupiter, Surabaya', '1990-04-06', 'Jessica7956@gmail.com', 6284594960174),
184 (1000010, 'Christina', 'Williams', 'Jalan Saturn, Banyuwangi', '1986-07-14', 'Christina3200@gmail.com', 6283327256322),
185 (1000011, 'Heather', 'Johnson', 'Jalan Mars, Blitar', '1958-11-21', 'Heather9697@gmail.com', 6280688352264),
186 (1000012, 'Sara', 'Brown', 'Jalan Neptunus, Merauke', '1967-10-31', 'Sara6130@gmail.com', 6286299442155),
187 (1000013, 'Andrew', 'Torres', 'Jalan Hatta, Cirebon', '1992-10-11', 'Andrew3146@gmail.com', 6284525262483),
188 (1000014, 'Brent', 'Robertson', 'Jalan Imam Bonjol, Padang', '1983-06-29', 'Brent4643@gmail.com', 6285133175448),
189 (1000015, 'Christopher', 'Hebert', 'Jalan Diponegoro, Jakarta', '1977-12-30', 'Christopher3917@gmail.com', 6286484887175),
190 (1000016, 'Ashley', 'Scott', 'Jalan Sam Ratulangi, Manado', '1977-12-23', 'Ashley4379@gmail.com', 6281265825535),
191 (1000017, 'Christina', 'Wright', 'Jalan Pattimura, Ambon', '1982-04-27', 'Christina3630@gmail.com', 6284986150812),
192 (1000018, 'Sarah', 'Jones', 'Jalan Dewi Sartika, Solo', '1975-09-27', 'Sarah1961@gmail.com', 628400976621),
193 (1000019, 'Elizabeth', 'Ford', 'Jalan Damar, Jakarta', '1972-07-02', 'Elizabeth8853@gmail.com', 6289017740418),
194 (1000020, 'Patrick', 'Wallace', 'Jalan Kawi, Yogyakarta', '1994-12-16', 'Patrick4232@gmail.com', 6283486621438);

```

	id_p	nama_dp	nama_b	NoHP_p	alamat_p	email_p	TL_p	umur_p
1	1000001	Stacy	Wagner	6284219257513	Jalan Aipda Karel Satsuit Tubu...	Stacy9341@gmail.com	1958-01-18	65
2	1000002	Colleen	Smith	6284550522882	Jalan Asia Afrika, Bandung	Colleen9673@gmail.com	1975-03-25	48
3	1000003	Timothy	Wells	6282399105057	Jalan Pahlawan Bedul, Makassar	Timothy8773@gmail.com	1979-07-22	43
4	1000004	Frank	Smith	6284491079635	Jalan Mawar, Jakarta	Frank920@gmail.com	1952-04-30	71
5	1000005	Adriana	Galloway	6283542420250	Jalan Banteng, Semarang	Adriana718@gmail.com	1961-11-30	61
6	1000006	Jill	Wagner	6286535804163	Jalan Formulae, Jakarta	Jill11465@gmail.com	1976-07-23	46
7	1000007	Christy	Cox	6284682498686	Jalan Bintang, Jakarta	Christy4220@gmail.com	1991-06-22	31
8	1000008	Michelle	Fitzpatrick	6285533851992	Jalan Matahari, Jayapura	Michelle3992@gmail.com	1963-03-28	60
9	1000009	Jessica	Grant	6284594960174	Jalan Jupiter, Surabaya	Jessica7956@gmail.com	1990-04-06	33
10	1000010	Christina	Williams	6283327256322	Jalan Saturn, Banyuwangi	Christina3200@gmail.com	1986-07-14	36
11	1000011	Heather	Johnson	6280688352264	Jalan Mars, Blitar	Heather9697@gmail.com	1958-11-21	64
12	1000012	Sara	Brown	6286299442155	Jalan Neptunus, Merauke	Sara6130@gmail.com	1967-10-31	55
13	1000013	Andrew	Torres	6284525262483	Jalan Hatta, Cirebon	Andrew3146@gmail.com	1992-10-11	30
14	1000014	Brent	Robertson	6285133175448	Jalan Imam Bonjol, Padang	Brent4643@gmail.com	1983-06-29	39
15	1000015	Christopher	Hebert	6286484887175	Jalan Diponegoro, Jakarta	Christopher3917@gmail.com	1977-12-30	45
16	1000016	Ashley	Scott	6281265825535	Jalan Sam Ratulangi, Manado	Ashley4379@gmail.com	1977-12-23	45
17	1000017	Christina	Wright	6284986150812	Jalan Pattimura, Ambon	Christina3630@gmail.com	1982-04-27	41
18	1000018	Sarah	Jones	628400976621	Jalan Dewi Sartika, Solo	Sarah1961@gmail.com	1975-09-27	47
19	1000019	Elizabeth	Ford	6289017740418	Jalan Damar, Jakarta	Elizabeth8853@gmail.com	1972-07-02	50
20	1000020	Patrick	Wallace	6283486621438	Jalan Kawi, Yogyakarta	Patrick4232@gmail.com	1994-12-16	28

2. Tabel Jenis dan Distributor

```

196
197 INSERT INTO JENIS (kode_prod, klasifikasi_prod) VALUES
198 (1, 'Kelistrikan'),
199 (2, 'Transmisi'),
200 (3, 'Karet'),
201 (4, 'Logam');
202
203
204 INSERT INTO DISTRIBUTOR (id_d, nama_d, alamat_d, NoHP_d) VALUES
205 (320001, 'Distributor A', 'Jalan Blitar,Jakarta', 6284206969101),
206 (320002, 'Distributor B', 'Jalan Kertanegara,Jakarta', 6284206962911),
207 (320003, 'Distributor C', 'Jalan Sumur Batu, Jakarta', 6284206967212),
208 (320004, 'Distributor D', 'Jalan Hang Tuah, Jakarta', 6284206968203);
209

```

	kode_prod	klasifikasi_prod
1	1	Kelistrikan
2	2	Transmisi
3	3	Karet
4	4	Logam

	id_d	nama_d	alamat_d	NoHP_d
1	320001	Distributor A	Jalan Blitar,Jakarta	6284206969101
2	320002	Distributor B	Jalan Kertanegara,Jakarta	6284206962911
3	320003	Distributor C	Jalan Sumur Batu, Jakarta	6284206967212
4	320004	Distributor D	Jalan Hang Tuah, Jakarta	6284206968203

3. Tabel Akun

```

212 INSERT INTO AKUN (id_p, username, password) VALUES
213 (1000001, 'Tyler9401@gmail.com', 'P8PSHqz4MWIC'),
214 (1000002, 'Nancy3456@gmail.com', 'rRcHyKuCEsN'),
215 (1000003, 'Melanie9053@gmail.com', 'f2HP3UcJil'),
216 (1000004, 'Matthew3257@gmail.com', '8B9PBLKx'),
217 (1000005, 'Robert3564@gmail.com', 'fee1WU7ogpkv'),
218 (1000006, 'Sarah2668@gmail.com', '45jr0WOeV'),
219 (1000007, 'Brian9932@gmail.com', 'qSuTGuqD72B'),
220 (1000008, 'Joseph2204@gmail.com', 'oQDdFgsGC4'),
221 (1000009, 'Ryan1127@gmail.com', 'nmkhW2oST'),
222 (1000010, 'Manuel6201@gmail.com', 'y5lrISuf'),
223 (1000011, 'Craig3559@gmail.com', 'SJqyAw2RMPJ'),
224 (1000012, 'Erin663@gmail.com', 'UeTmreCd'),
225 (1000013, 'Tammy3622@gmail.com', 'Sna34iJa7'),
226 (1000014, 'Michael7700@gmail.com', 'SOU3loXzw'),
227 (1000015, 'Elizabeth9970@gmail.com', 'qSOacFwE'),
228 (1000016, 'John8100@gmail.com', 'g6V8Yq5NL'),
229 (1000017, 'Tyler836@gmail.com', 'wEGsJjUxN'),
230 (1000018, 'Andrew5157@gmail.com', 'ffgNqRDLMPs'),
231 (1000019, 'Jason3936@gmail.com', 'wGGx7rMbZ0'),
232 (1000020, 'Alison9631@gmail.com', 'y52IgcFzCO');
233

```

	username	password	id_p
1	Tyler9401@gmail.com	P8PSHqz4MWIC	1000001
2	Nancy3456@gmail.com	rRcHyKuCEsN	1000002
3	Matthew3257@gmail.com	8B9PBLKx	1000004
4	Robert3564@gmail.com	fee1WU7ogpkv	1000005
5	Sarah2668@gmail.com	45jr0WOeV	1000006
6	Brian9932@gmail.com	qSuTGuqD72B	1000007
7	Joseph2204@gmail.com	oQDdFgsGC4	1000008
8	Ryan1127@gmail.com	nmkhW2oST	1000009
9	Manuel6201@gmail.com	y5lrISuf	1000010
10	Craig3559@gmail.com	Craig1234567	1000011
11	Erin663@gmail.com	UeTmreCd	1000012
12	Tammy3622@gmail.com	Sna34iJa7	1000013
13	Michael7700@gmail.com	SOU3loXzw	1000014
14	Elizabeth9970@gmail.com	qSOacFwE	1000015
15	John8100@gmail.com	g6V8Yq5NL	1000016
16	Tyler836@gmail.com	wEGsJjUxN	1000017
17	Andrew5157@gmail.com	ffgNqRDLMPs	1000018
18	Jason3936@gmail.com	wGGx7rMbZ0	1000019
19	Alison9631@gmail.com	y52IgcFzCO	1000020

4. Tabel Karyawan

```

236 INSERT INTO KARYAWAN (id_kar, nama_kar, alamat_kar, NoHP_kar, tugas, kode_prod) VALUES
237 (90001, 'Ahmad Utomo', 'Jalan Gadjah Mada, Jakarta', 6286726212303, 'Pembelian', 1),
238 (90002, 'Bambang Hartono', 'Jalan Joyoboyo, Jakarta', 6281662695309, 'Pemesanan', 1),
239 (90003, 'Dwi Prabowo', 'Jalan Kebon Sirih, Jakarta', 6283509659693, 'Penyimpanan', 1),
240 (90004, 'Edi Santoso', 'Jalan Arjuno, Jakarta', 6280482241048, 'Perawatan', 1),
241 (90005, 'Faisal Suryanto', 'Jalan Majapahit, Jakarta', 6282840996043, 'Pembelian', 2),
242 (90006, 'Gunawan Suryanto', 'Jalan Dukuh, Jakarta', 6288133061387, 'Pemesanan', 2),
243 (90007, 'Hendro Santoso', 'Jalan Sriwijaya, Jakarta', 6288916385030, 'Penyimpanan', 2),
244 (90008, 'Irfan Kusuma', 'Jalan Dharmawangsa, Jakarta', 6284738309123, 'Perawatan', 2),
245 (90009, 'Joko Utomo', 'Jalan Ampera Raya, Jakarta', 6285144997906, 'Pembelian', 3),
246 (90010, 'Kurniawan Hartono', 'Jalan Margonda, Jakarta', 6285862946762, 'Pemesanan', 3),
247 (90011, 'Lukman Prabowo', 'Jalan Bintaro Raya, Jakarta', 6283277419803, 'Penyimpanan', 3),
248 (90012, 'Maulana Suryanto', 'Jalan Raya Condet, Jakarta', 6280194979693, 'Perawatan', 3),
249 (90013, 'Nugroho Prabowo', 'Jalan Raya Cipayung, Jakarta', 6280014862792, 'Pembelian', 4),
250 (90014, 'Onno Hartono', 'Jalan Jaksa, Jakarta', 6286011952150, 'Pemesanan', 4),
251 (90015, 'Prabowo Utomo', 'Jalan Jambore, Jakarta', 6284750515493, 'Penyimpanan', 4),
252 (90016, 'Rizki Nugroho', 'Jalan Pluit, Jakarta', 6281293073983, 'Perawatan', 4);
253

```

	id_kar	nama_kar	alamat_kar	NoHP_kar	kode_prod	tugas
1	90001	Ahmad Utomo	Jalan Gadjah Mada, Jakarta	6286726212303	1	Pembelian
2	90002	Bambang Hartono	Jalan Joyoboyo, Jakarta	6281662695309	1	Pemesanan
3	90003	Dwi Prabowo	Jalan Kebon Sirih, Jakarta	6283509659693	1	Penyimpanan
4	90004	Edi Santoso	Jalan Arjuno, Jakarta	6280482241048	1	Perawatan
5	90005	Faisal Suryanto	Jalan Majapahit, Jakarta	6282840996043	2	Pembelian
6	90006	Gunawan Suryanto	Jalan Dukuh, Jakarta	6288133061387	2	Pemesanan
7	90007	Hendro Santoso	Jalan Sriwijaya, Jakarta	6288916385030	2	Penyimpanan
8	90008	Irfan Kusuma	Jalan Dharmawangsa, Jakarta	6284738309123	2	Perawatan
9	90009	Joko Utomo	Jalan Ampera Raya, Jakarta	6285144997906	3	Pembelian
10	90010	Kurniawan Hartono	Jalan Margonda, Jakarta	6285862946762	3	Pemesanan
11	90011	Lukman Prabowo	Jalan Bintaro Raya, Jakarta	6283277419803	3	Penyimpanan
12	90012	Maulana Suryanto	Jalan Raya Condet, Jakarta	6280194979693	3	Perawatan
13	90013	Nugroho Prabowo	Jalan Raya Cipayung, Jakarta	6280014862792	4	Pembelian
14	90014	Onno Hartono	Jalan Jaksa, Jakarta	6286011952150	4	Pemesanan
15	90015	Prabowo Utomo	Jalan Jambore, Jakarta	6284750515493	4	Penyimpanan
16	90016	Rizki Nugroho	Jalan Pluit, Jakarta	6281293073983	4	Perawatan

5. Tabel Suku Cadang

```

256 INSERT INTO SUKU_CADANG (no_prod, kode_prod, nama_prod, kuantitas_prod, harga_jual) VALUES
257 (1, 1, 'Alternator', 7, 2700000),
258 (2, 1, 'Starter Motor', 9, 510000),
259 (3, 1, 'Ignition Coil', 8, 175000),
260 (4, 1, 'Spark Plugs', 8, 46000),
261 (5, 1, 'Fuses', 15, 35000),
262 (6, 2, 'Fluid', 15, 138000),
263 (7, 2, 'Filter', 10, 50000),
264 (8, 2, 'Clutch Kit', 15, 45000),
265 (9, 2, 'Solenoid', 10, 145000),
266 (10, 2, 'Exhaust', 10, 158000),
267 (11, 3, 'Tires', 7, 440000),
268 (12, 3, 'Belt', 10, 190000),
269 (13, 3, 'Hoses', 15, 50000),
270 (14, 3, 'Bushings', 10, 300000),
271 (15, 3, 'Pedal', 8, 85000),
272 (16, 4, 'Piston', 7, 198000),
273 (17, 4, 'Rotors', 7, 3400000),
274 (18, 4, 'Muffler', 8, 450000),
275 (19, 4, 'Chassis', 2, 2000000),
276 (20, 4, 'Fan', 2, 11500000);
277

```

	no_prod	kode_prod	nama_prod	kuantitas_prod	harga_jual					
1	1	1	Alternator	9	2700000	11	11	3	Tires	4 440000
2	2	1	Starter Motor	2	510000	12	12	3	Belt	7 190000
3	3	1	Ignition Coil	8	175000	13	13	3	Hoses	15 50000
4	4	1	Spark Plugs	-2	46000	14	14	3	Bushings	10 300000
5	5	1	Fuses	13	35000	15	15	3	Pedal	17 85000
6	6	2	Fluid	11	138000	16	16	4	Piston	5 198000
7	7	2	Filter	0	50000	17	17	4	Rotors	7 3400000
8	8	2	Clutch Kit	9	45000	18	18	4	Muffler	8 450000
9	9	2	Solenoid	6	145000	19	19	4	Chassis	2 20000000
10	10	2	Exhaust	11	158000	20	20	4	Fan	2 11500000

6. Tabel Membeli

```

279 INSERT INTO MEMBELI (tanggal_pembelian, id_p, no_prod, kuantitas_pembelian) vi
280 ('2022-01-21', 1000013, 8, 6),
281 ('2022-02-14', 1000015, 20, 1),
282 ('2022-02-19', 1000001, 19, 1),
283 ('2022-03-08', 1000017, 12, 3),
284 ('2022-03-19', 1000008, 11, 1),
285 ('2022-04-14', 1000018, 11, 2),
286 ('2022-04-18', 1000020, 17, 3),
287 ('2022-04-21', 1000020, 1, 2),
288 ('2022-05-11', 1000011, 5, 2),
289 ('2022-05-28', 1000008, 9, 4),
290 ('2022-06-15', 1000008, 14, 3),
291 ('2022-07-09', 1000005, 3, 3),
292 ('2022-07-12', 1000009, 14, 2),
293 ('2022-07-19', 1000018, 16, 2),
294 ('2022-07-27', 1000006, 5, 5),
295 ('2022-09-24', 1000007, 4, 3),
296 ('2022-10-05', 1000017, 7, 5),
297 ('2022-10-08', 1000005, 17, 1),
298 ('2022-10-21', 1000020, 6, 3),
299 ('2022-11-15', 1000001, 7, 2),
300
301 ('2022-11-17', 1000002, 6, 5),
302 ('2022-11-20', 1000016, 2, 4),
303 ('2022-12-06', 1000020, 5, 2),
304 ('2022-12-18', 1000020, 11, 2),
305 ('2022-12-20', 1000001, 4, 2),
306 ('2023-03-04', 1000008, 2, 3),
307 ('2023-03-11', 1000004, 7, 3),
308 ('2023-03-25', 1000001, 4, 1),
309 ('2023-05-05', 1000017, 10, 5),
310 ('2023-05-18', 1000002, 3, 1);

```

	id_pembelian	tanggal_pembelian	no_prod	hargatotal_pembelian	id_p	kuantitas_pembelian
1	1	2022-01-21	8	270000	1000013	6
2	2	2022-02-14	20	11500000	1000015	1
3	3	2022-02-19	19	20000000	1000001	1
4	4	2022-03-08	12	570000	1000017	3
5	5	2022-03-19	11	440000	1000008	1
6	6	2022-04-14	11	880000	1000018	2
7	7	2022-04-18	17	10200000	1000020	3
8	8	2022-04-21	1	5400000	1000020	2
9	9	2022-05-11	5	70000	1000011	2
10	10	2022-05-28	9	580000	1000008	4
11	11	2022-06-15	14	900000	1000008	3
12	12	2022-07-09	3	525000	1000005	3
13	13	2022-07-12	14	600000	1000009	2
14	14	2022-07-19	16	396000	1000018	2
15	15	2022-07-27	5	175000	1000006	5

16	16	2022-09-24	4	138000	1000007	3
17	17	2022-10-05	7	250000	1000017	5
18	18	2022-10-08	17	3400000	1000005	1
19	19	2022-10-21	6	414000	1000020	3
20	20	2022-11-15	7	100000	1000001	2
21	21	2022-11-17	6	690000	1000002	5
22	22	2022-11-20	2	2040000	1000016	4
23	23	2022-12-06	5	70000	1000020	2
24	24	2022-12-18	11	880000	1000020	2
25	25	2022-12-20	4	92000	1000001	2
26	26	2023-03-04	2	1530000	1000008	3
27	27	2023-03-11	7	150000	1000004	3
28	28	2023-03-25	4	46000	1000001	1
29	29	2023-05-05	10	790000	1000017	5
30	30	2023-05-18	3	175000	1000002	1

7. Tabel Pesanan

```

315
316 INSERT INTO PESANAN (no_prod, harga_beli) VALUES
317 (1, 2300000),
318 (2, 480000),
319 (3, 140000),
320 (4, 30000),
321 (5, 25000),
322 (6, 115000),
323 (7, 38000),
324 (8, 29000),
325 (9, 125000),
326 (10, 132000),
327 (11, 383000),
328 (12, 150000),
329 (13, 35000),
330 (14, 250000),
331 (15, 60000),
332 (16, 170000),
333 (17, 3000000),
334 (18, 370000),
335 (19, 16500000),
336 (20, 9800000);

```

	no_prod	harga_beli
1	1	2300000
2	2	480000
3	3	140000
4	4	30000
5	5	25000
6	6	115000
7	7	38000
8	8	29000
9	9	125000
10	10	132000

11	11	383000
12	12	150000
13	13	35000
14	14	250000
15	15	60000
16	16	170000
17	17	3000000
18	18	370000
19	19	16500000
20	20	9800000

8. Tabel Riwayat_Belanja_Dari

```

339 INSERT INTO RIWAYAT_BELANJA_DARI (tanggal_pesan, no_prod, kuantitas_pesan, id_d) VALUES
340 ('2022-01-27', 15, 4, 320002),
341 ('2022-06-28', 11, 2, 320003),
342 ('2022-10-07', 19, 1, 320001),
343 ('2022-11-27', 20, 1, 320001),
344 ('2022-12-26', 17, 4, 320002),
345 ('2023-02-03', 3, 2, 320003),
346 ('2023-02-13', 15, 5, 320004),
347 ('2023-02-28', 14, 5, 320004),
348 ('2023-03-29', 3, 2, 320004),
349 ('2023-04-13', 1, 4, 320001),
350 ('2023-04-15', 6, 4, 320003),
351 ('2023-04-22', 5, 7, 320002);

```

	id_pesan	tanggal_pesan	no_prod	kuantitas_pesan	hargatotal_pesanan	id_d
1	1	2022-01-27	15	4	240000	320002
2	2	2022-06-28	11	2	766000	320003
3	3	2022-10-07	19	1	16500000	320001
4	4	2022-11-27	20	1	9800000	320001
5	5	2022-12-26	17	4	12000000	320002
6	6	2023-02-03	3	2	280000	320003
7	7	2023-02-13	15	5	300000	320004
8	8	2023-02-28	14	5	1250000	320004
9	9	2023-03-29	3	2	280000	320004
10	10	2023-04-13	1	4	9200000	320001
11	11	2023-04-15	6	4	460000	320003
12	12	2023-04-22	5	7	175000	320002
13	13	2023-05-29	10	3	396000	320004
14	14	2023-05-29	10	3	396000	320004

4.4.2 Simulasi INSERT, UPDATE, dan DELETE

Pada tahap ini, akan dilakukan simulasi untuk memasukkan data baru pada tabel (**INSERT**), memperbarui data pada tabel (**UPDATE**), dan menghapus data pada tabel (**DELETE**). Berikut akan dijelaskan untuk melakukan simulasi tersebut.

Pertama, misalkan ada pelanggan baru yang melakukan transaksi pembelian suku cadang. Selain itu, ada penambahan stok dari no_prod = 10 (Exhaust). Oleh karena itu, akan dilakukan pemasukan data baru pada tabel membeli dan riwayat belanja menggunakan perintah INSERT INTO.

```
/* Simulasi*/
INSERT INTO MEMBELI (tanggal_pembelian, id_p, no_prod, kuantitas_pembelian) VALUES
('2023-05-29', 1000020, 4, 2);

INSERT INTO RIWAYAT_BELANJA_DARI (tanggal_pesan, no_prod, kuantitas_pesan, id_d) VALUES
('2023-05-29', 10, 3, 320004);
```

Hasilnya adalah sebagai berikut:

- INSERT pada Tabel Membeli

Sebelum

	id_pembelian	tanggal_pembelian	no_prod	hargatotal_pembelian	id_p	kuantitas_pembelian
1	1	2022-01-21	8	270000	1000013	6
2	2	2022-02-14	20	11500000	1000015	1
3	3	2022-02-19	19	20000000	1000001	1
4	4	2022-03-08	12	570000	1000017	3
5	5	2022-03-19	11	440000	1000008	1
6	6	2022-04-14	11	880000	1000018	2
7	7	2022-04-18	17	10200000	1000020	3
8	8	2022-04-21	1	5400000	1000020	2
9	9	2022-05-11	5	70000	1000011	2
10	10	2022-05-28	9	580000	1000008	4
11	11	2022-06-15	14	900000	1000008	3
12	12	2022-07-09	3	525000	1000005	3
13	13	2022-07-12	14	600000	1000009	2
14	14	2022-07-19	16	396000	1000018	2
15	15	2022-07-27	5	175000	1000006	5
16	16	2022-09-24	4	138000	1000007	3

Sesudah

	id_pembelian	tanggal_pembelian	no_prod	hargatotal_pembelian	id_p	kuantitas_pembelian
15	15	2022-09-27	5	170000	1000000	5
16	16	2022-09-24	4	138000	1000007	3
17	17	2022-10-05	7	250000	1000017	5
18	18	2022-10-08	17	3400000	1000005	1
19	19	2022-10-21	6	414000	1000020	3
20	20	2022-11-15	7	100000	1000001	2
21	21	2022-11-17	6	690000	1000002	5
22	22	2022-11-20	2	2040000	1000016	4
23	23	2022-12-06	5	70000	1000020	2
24	24	2022-12-18	11	880000	1000020	2
25	25	2022-12-20	4	92000	1000001	2
26	26	2023-03-04	2	1530000	1000008	3
27	27	2023-03-11	7	150000	1000004	3
28	28	2023-03-25	4	46000	1000001	1
29	29	2023-05-05	10	790000	1000017	5
30	30	2023-05-18	3	175000	1000002	1
31	34	2023-05-29	4	92000	1000020	2

- INSERT pada Tabel Riwayat_Belanja_Dari

Sebelum

	id_pesanan	tanggal_pesanan	no_produk	kuantitas_pesanan	harga_total_pesanan	id_d
1	1	2022-01-27	15	4	240000	320002
2	2	2022-06-28	11	2	766000	320003
3	3	2022-10-07	19	1	16500000	320001
4	4	2022-11-27	20	1	9800000	320001
5	5	2022-12-26	17	4	12000000	320002
6	6	2023-02-03	3	2	280000	320003
7	7	2023-02-13	15	5	300000	320004
8	8	2023-02-28	14	5	1250000	320004
9	9	2023-03-29	3	2	280000	320004
10	10	2023-04-13	1	4	9200000	320001
11	11	2023-04-15	6	4	460000	320003
12	12	2023-04-22	5	7	175000	320002
13	13	2023-05-29	10	3	396000	320004
14	14	2023-05-29	10	3	396000	320004

Sesudah

	id_pesanan	tanggal_pesanan	no_produk	kuantitas_pesanan	harga_total_pesanan	id_d
1	1	2022-01-27	15	4	240000	320002
2	2	2022-06-28	11	2	766000	320003
3	3	2022-10-07	19	1	16500000	320001
4	4	2022-11-27	20	1	9800000	320001
5	5	2022-12-26	17	4	12000000	320002
6	6	2023-02-03	3	2	280000	320003
7	7	2023-02-13	15	5	300000	320004
8	8	2023-02-28	14	5	1250000	320004
9	9	2023-03-29	3	2	280000	320004
10	10	2023-04-13	1	4	9200000	320001
11	11	2023-04-15	6	4	460000	320003
12	12	2023-04-22	5	7	175000	320002
13	13	2023-05-29	10	3	396000	320004
14	14	2023-05-29	10	3	396000	320004
15	15	2023-05-29	10	3	396000	320004

Selanjutnya, misalkan ada pelanggan yang sudah lama tidak melakukan transaksi. Akun dari pelanggan tersebut akan dihapus karena tidak ada aktivitas transaksi lagi. Penghapusan data dari tabel akun menggunakan perintah DELETE.

```
DELETE FROM AKUN
WHERE id_p = 1000003;
```

Hasilnya sebagai berikut:

- DELETE pada Tabel Akun

Sebelum

	username	password	id_p
1	Tyler9401@gmail.com	P8PSHqz4MWIC	1000001
2	Nancy3456@gmail.com	rRcHyKuCEsN	1000002
3	Melanie9053@gmail.com	f2HP3UcJi1	<u>1000003</u>
4	Matthew3257@gmail.com	8B9PBLKx	1000004
5	Robert3564@gmail.com	fee1WU7ogpkv	1000005
6	Sarah2668@gmail.com	45jr0WOeV	1000006

Sesudah

	username	password	id_p
1	Tyler9401@gmail.com	P8PSHqz4MWIC	1000001
2	Nancy3456@gmail.com	rRcHyKuCEsN	1000002
3	Matthew3257@gmail.com	8B9PBLKx	1000004
4	Robert3564@gmail.com	fee1WU7ogpkv	1000005
5	Sarah2668@gmail.com	45jr0WOeV	1000006

Terakhir, misalnya terdapat pelanggan yang ingin mengganti *password*. Akan dilakukan pembaruan data pada tabel akun, yaitu memperbarui *password* pelanggan menggunakan perintah UPDATE.

```
UPDATE AKUN
SET password = 'Craig1234567'
WHERE id_p = 1000011;
```

Hasilnya sebagai berikut:

- UPDATE pada Tabel Akun

Sebelum

	username	password	id_p
1	Matthew3257@gmail.com	8B9PBLKx	1000004
5	Robert3564@gmail.com	fee1WU7ogpkv	1000005
6	Sarah2668@gmail.com	45jr0WOeV	1000006
7	Brian9932@gmail.com	qSuTGuqD72B	1000007
8	Joseph2204@gmail.com	oQDdFgsGC4	1000008
9	Ryan1127@gmail.com	nmkhW2oST	1000009
10	Manuel6201@gmail.com	y5lrISuf	1000010
11	Craig3559@gmail.com	SJqyAw2RMPJ	1000011
12	Erin663@gmail.com	UeTmreCd	1000012
13	Tammy3622@gmail.com	Sna34iJa7	1000013

Sesudah

	username	password	id_p
1	Tyler9401@gmail.com	P8PSHqz4MWIC	1000001
2	Nancy3456@gmail.com	rRcHyKuCEsN	1000002
3	Melanie9053@gmail.com	f2HP3UcJi1	1000003
4	Matthew3257@gmail.com	8B9PBLKx	1000004
5	Robert3564@gmail.com	fee1WU7ogpkv	1000005
6	Sarah2668@gmail.com	45jr0WOeV	1000006
7	Brian9932@gmail.com	qSuTGuqD72B	1000007
8	Joseph2204@gmail.com	oQDdFgsGC4	1000008
9	Ryan1127@gmail.com	nmkhW2oST	1000009
10	Manuel6201@gmail.com	y5lrISuf	1000010
11	Craig3559@gmail.com	<u>Craig1234567</u>	1000011
12	Erin663@gmail.com	UeTmreCd	1000012
13	Tammy3622@gmail.com	Sna34iJa7	1000013

4.5 Analisis

Akan dilakukan analisis tabel-tabel yang dibuat dengan menggunakan JOIN dan GROUP BY tabel. Pertama, ingin diketahui alamat dari pelanggan yang melakukan pesanan. Karena pelanggan yang memiliki akun mungkin saja tidak membeli suku cadang yang ditawarkan, akan digunakan INNER JOIN agar *output* yang dikeluarkan tidak ada yang kosong pada kolom-kolom tertentu. Akan digabungkan tabel Pelanggan dan tabel Membeli menggunakan *code* INNER JOIN. Didapatkan hasil berikut ini.

```

382  /* Analisis*/
383  SELECT PELANGGAN.nama_dp, PELANGGAN.nama_b, PELANGGAN.alamat_p, MEMBELI.tanggal_pembelian,
384  MEMBELI.no_prod, MEMBELI.hargatotal_pembelian, MEMBELI.kuantitas_pembelian, SUKU_CADANG.nama_prod
385  FROM PELANGGAN
386  INNER JOIN MEMBELI ON PELANGGAN.id_p = MEMBELI.id_p
387  INNER JOIN SUKU_CADANG ON MEMBELI.no_prod = SUKU_CADANG.no_prod;
388

```

	nama_dp	nama_b	alamat_p	tanggal_pembelian	no_prod	hargatotal_pembelian	kuantitas_pembelian	nama_prod
1	Andrew	Torres	Jalan Hatta, Cirebon	2022-01-21	8	270000	6	Clutch Kit
2	Christopher	Hebert	Jalan Diponegoro, Jakarta	2022-02-14	20	11500000	1	Fan
3	Stacy	Wagner	Jalan Aipda Karel Satsuit Tubu...	2022-02-19	19	20000000	1	Chassis
4	Christina	Wright	Jalan Pattimura, Ambon	2022-03-08	12	570000	3	Belt
5	Michelle	Fitzpatrick	Jalan Matahari, Jayapura	2022-03-19	11	440000	1	Tires
6	Sarah	Jones	Jalan Dewi Sartika, Solo	2022-04-14	11	880000	2	Tires
7	Patrick	Wallace	Jalan Kawi, Yogyakarta	2022-04-18	17	10200000	3	Rotors
8	Patrick	Wallace	Jalan Kawi, Yogyakarta	2022-04-21	1	5400000	2	Alternator
9	Heather	Johnson	Jalan Mars, Blitar	2022-05-11	5	70000	2	Fuses
10	Michelle	Fitzpatrick	Jalan Matahari, Jayapura	2022-05-28	9	580000	4	Solenoid
11	Michelle	Fitzpatrick	Jalan Matahari, Jayapura	2022-06-15	14	900000	3	Bushings
12	Adriana	Galloway	Jalan Banteng, Semarang	2022-07-09	3	525000	3	Ignition Coil
13	Jessica	Grant	Jalan Jupiter, Surabaya	2022-07-12	14	600000	2	Bushings
14	Sarah	Jones	Jalan Dewi Sartika, Solo	2022-07-19	16	396000	2	Piston
15	Jill	Wagner	Jalan Formulae, Jakarta	2022-07-27	5	175000	5	Fuses
16	Christy	Cox	Jalan Bintang, Jakarta	2022-09-24	4	138000	3	Spark Plugs
17	Christina	Wright	Jalan Pattimura, Ambon	2022-10-05	7	250000	5	Filter
18	Adriana	Galloway	Jalan Banteng, Semarang	2022-10-08	17	3400000	1	Rotors
19	Patrick	Wallace	Jalan Kawi, Yogyakarta	2022-10-21	6	414000	3	Fluid
20	Stacy	Wagner	Jalan Aipda Karel Satsuit Tubu...	2022-11-15	7	100000	2	Filter
21	Colleen	Smith	Jalan Asia Afrika, Bandung	2022-11-17	6	690000	5	Fluid
22	Ashley	Scott	Jalan Sam Ratulangi, Manado	2022-11-20	2	2040000	4	Starter Motor
23	Patrick	Wallace	Jalan Kawi, Yogyakarta	2022-12-06	5	70000	2	Fuses
24	Patrick	Wallace	Jalan Kawi, Yogyakarta	2022-12-18	11	880000	2	Tires
25	Stacy	Wagner	Jalan Aipda Karel Satsuit Tubu...	2022-12-20	4	92000	2	Spark Plugs
26	Michelle	Fitzpatrick	Jalan Matahari, Jayapura	2023-03-04	2	1530000	3	Starter Motor
27	Frank	Smith	Jalan Mawar, Jakarta	2023-03-11	7	150000	3	Filter
28	Stacy	Wagner	Jalan Aipda Karel Satsuit Tubu...	2023-03-25	4	46000	1	Spark Plugs
29	Christina	Wright	Jalan Pattimura, Ambon	2023-05-05	10	790000	5	Exhaust
30	Colleen	Smith	Jalan Asia Afrika, Bandung	2023-05-18	3	175000	1	Ignition Coil
31	Patrick	Wallace	Jalan Kawi, Yogyakarta	2023-05-29	4	92000	2	Spark Plugs
32	Patrick	Wallace	Jalan Kawi, Yogyakarta	2023-05-29	4	92000	2	Spark Plugs

Kedua, ingin diketahui perbandingan harga pembelian dan harga penjualan suku cadang. Hal tersebut dilakukan untuk mengecek apakah terjadi kesalahan ketik pada data di tabel Suku Cadang. Jika harga penjualan (pada tabel Suku Cadang) suatu suku cadang lebih rendah daripada harga pembelian (pada tabel Pesanan), mengindikasikan terjadinya kesalahan ketik pada data yang di-input-kan. Dengan menggunakan *code JOIN*, didapatkan hasil sebagai berikut.

```
367 SELECT SUKU_CADANG.nama_prod, SUKU_CADANG.kuantitas_prod, SUKU_CADANG.harga_jual, PESANAN.harga_beli
368 FROM SUKU_CADANG
369 INNER JOIN PESANAN ON PESANAN.no_prod = SUKU_CADANG.no_prod;
```

	nama_prod	kuantitas_prod	harga_jual	harga_beli
1	Alternator	9	2700000	2300000
2	Starter Motor	2	510000	480000
3	Ignition Coil	8	175000	140000
4	Spark Plugs	2	46000	30000
5	Fuses	13	35000	25000
6	Fluid	11	138000	115000
7	Filter	0	50000	38000
8	Clutch Kit	9	45000	29000
9	Solenoid	6	145000	125000
10	Exhaust	5	158000	132000
11	Tires	4	440000	383000
12	Belt	7	190000	150000
13	Hoses	15	50000	35000
14	Bushings	10	300000	250000
15	Pedal	17	85000	60000
16	Piston	5	198000	170000
17	Rotors	7	3400000	3000000
18	Muffler	8	450000	370000
19	Chassis	2	20000000	16500000
20	Fan	2	11500000	9800000

Terlihat bahwa harga jual dari suku cadang sudah lebih besar daripada harga belinya, untuk semua suku cadang, yang mana menandakan tidak ada kesalahan ketik pada harga tersebut. Selain itu, terlihat bahwa pada kuantitas filter tidak ada (0), yang berarti perlu dilakukan *restocking* pada suku cadang tersebut.

Ketiga, ingin diketahui pada bulan apakah terjadi banyak transaksi suku cadang. Oleh karena itu, dibutuhkan bantuan *code* GROUP BY dan ORDER BY untuk mengetahui hal tersebut. Didapatkan hasil seperti berikut ini.

```
397  
398 SELECT strftime('%Y-%m', tanggal_pembelian) AS month, COUNT(*) AS record_count  
399 FROM MEMBELI  
400 GROUP BY month  
401 ORDER BY record_count DESC;
```

	month	record_count
1	2023-05	4
2	2022-07	4
3	2023-03	3
4	2022-12	3
5	2022-11	3
6	2022-10	3
7	2022-04	3
8	2022-05	2
9	2022-03	2
10	2022-02	2
11	2022-09	1
12	2022-06	1
13	2022-01	1

Pada gambar diatas, terlihat bahwa transaksi terbanyak terjadi pada bulan Mei pada Tahun 2023 sebanyak 4 transaksi yang dilakukan.

Keempat, ingin diketahui lebih spesifik, penjualan suku cadang apa pada bulan tertentu yang jumlahnya paling banyak. Digunakan *code* berikut ini.

```
381 SELECT strftime('%Y-%m', MEMBELI.tanggal_pembelian) AS month, SUKU_CADANG.nama_prod, SUM(kuantitas_pembelian) AS jumlah_penjualan  
382 FROM MEMBELI  
383 INNER JOIN SUKU_CADANG ON SUKU_CADANG.no_prod = MEMBELI.no_prod  
384 GROUP BY month  
385 ORDER BY jumlah_penjualan DESC;
```

	month	nama_prod	jumlah_penjualan
1	2022-07	Ignition Coil	12
2	2022-11	Filter	11
3	2022-10	Filter	9
4	2023-03	Starter Motor	7
5	2022-04	Tires	7
6	2023-05	Exhaust	6
7	2022-12	Fuses	6
8	2022-05	Fuses	6
9	2022-01	Clutch Kit	6
10	2022-03	Belt	4
11	2022-09	Spark Plugs	3
12	2022-06	Bushings	3
13	2022-02	Fan	2

Terlihat bahwa suku cadang Ignition Coil terjual paling banyak yaitu sebanyak 12 produk selama bulan Juli tahun 2022. Berdasarkan informasi tersebut, perusahaan perlu mempersiapkan banyak stok Ignition Coil pada setiap bulan Juli agar ketika banyak pelanggan memesan Ignition Coil pada bulan tersebut, perusahaan mempunyai stok yang cukup.

5. Kesimpulan

Dengan menggunakan software SQLite, dapat dibuat sistem manajemen data *online shop* suku cadang dengan cukup mudah. Melalui proses CREATE and READ, kita dapat membuat tabel Akun, Distributor, Jenis, Karyawan, Membeli, Pelanggan, Pesanan, Riwayat_Belanja_Dari, dan Suku_Cadang serta menyimpan data-data dari seluruh entitas ke dalam tabel tersebut. Kemudian, dengan proses UPDATE and DELETE, kita dapat memperbarui dan menghapus data dalam tabel yang telah dibuat, misalnya memperbarui *password* atau menghapus akun pelanggan.

Dengan melakukan proses JOIN data, kita dapat menggabungkan beberapa tabel menjadi satu melalui sebuah kolom yang menghubungkan keduanya untuk melihat apakah ada kesamaan antar tabel walau ada entitas yang berbeda, tetapi isinya dapat kita hubungkan. Kemudian, dengan proses GROUP BY dan ORDER BY, kita dapat mengelompokkan data-data yang memiliki nilai yang sama dan mengurutkan datanya baik dari nilai yang terkecil maupun terbesar.

Dengan memanfaatkan proses JOIN, GROUP BY, dan ORDER BY, kita dapat menganalisis data *online shop* suku cadang, seperti:

1. Melihat produk apa saja yang dibeli pelanggan, baik dari harga total, tanggal pembelian, kuantitas, dan nama produk suku cadang yang dibeli.
2. Melihat keuntungan dari setiap penjualan produk suku cadang.
3. Melihat jumlah produk suku cadang yang terjual pada suatu bulan.
4. Melihat produk spesifik yang terjual paling banyak selama periode satu bulan.

Dengan adanya pembuatan database ini, diharapkan dapat mempermudah penjual dalam menyimpan dan mengakses informasi pelanggan, karyawan, pesanan, dan distributor. Dengan demikian, *online shop* dapat memperoleh lebih banyak keuntungan karena kinerjanya menjadi lebih efisien dan efektif.

DAFTAR PUSTAKA

14 Toko Online Terbaik di Indonesia untuk Berbagai Kebutuhan! (2023, Februari 5).

Gramedia.com. Diakses pada 8 Mei 2023, dari <https://www.gramedia.com/best-seller/toko-online-terbaik-di-indonesia/>

Inilah Manfaat Big Data Bagi Perusahaan dan Bisnis. (n.d.). Kuncie. Diakses pada 8 Mei 2023, dari <https://www.kuncie.com/posts/manfaat-big-data-bagi-perusahaan>

Nield, Thomas. (2016). *Getting Started With SQL*.

Online Shop, Market Place, E-Commerce. Apa bedanya? (n.d.). Kominfo Kab. Kediri.

Diakses pada 8 Mei 2023, dari <https://diskominfo.kedirikab.go.id/baca/online-shop-market-place-ecommerce-apa-bedanya>