**25 oktober 2024 pertemuan 1.**

**database penjualan dilengkapi dengan 3 tabel**

1. tabel pelanggan

tabel ini menyimpan informasi pelanggan yang melakukan pembelian.

| **bidang** | **tipe data** | **deskripsi** |
| --- | --- | --- |
| id\_pelanggan | ke dalam | primary key, id unik untuk setiap pelanggan |
| nama | varchar(100) | nama pelanggan |
| e-mail | varchar(100) | alamat email pelanggan |
| nomor telepon | varchar(15) | nomor telepon pelanggan |
| alamat | teks | alamat pelanggan |

penjelasan:

1. customer\_id: merupakan kolom kunci utama (primary key) yang memiliki nilai unik untuk setiap pelanggan. kolom ini otomatis bertambah (peningkatan otomatis) setiap kali ada pelanggan baru ditambahkan.
2. name: menyimpan nama pelanggan, menggunakan tipe data varchardengan batasan 100 karakter.
3. email: menyimpan alamat email pelanggan, juga dengan tipe data varchardan batas 100 karakter untuk menjaga konsistensi panjang email.
4. phone\_number: menyimpan nomor telepon pelanggan dengan panjang maksimal 15 karakter. cukup untuk format nomor telepon internasional.
5. address: menyimpan alamat pelanggan dalam tipe data text, karena alamat web bisa bervariasi panjangnya.
6. tabel produk

tabel ini menyimpan informasi produk yang dijual.

| **bidang** | **tipe data** | **deskripsi** |
| --- | --- | --- |
| id\_produk | ke dalam | primary key, id unik untuk setiap produk |
| nama produk | varchar(100) | nama produk |
| harga | desimal(10, 2) | harga produk |
| jumlah\_stok | ke dalam | jumlah stok yang tersedia |

penjelasan:

1. product\_id: kolom ini adalah kunci utama (primary key) yang berfungsi sebagai id unik untuk setiap produk.
2. product\_name: namavarchardengan batas maksimal 100 karakter. price:
3. harga produk menggunakan tipe data decimald engan format dua angka desimal (misalnya, 10,00 atau 1000,50) untuk mencatat harga dengan presisi.
4. stock\_quantity: menyimpan jumlah stok yang tersedia untuk produk tersebut.
5. tabel penjualan

tabel ini menyimpan informasi transaksi penjualan. relasinya adalah setiap penjualan terkait dengan satu pelanggan dan satu atau lebih produk.

| **bidang** | **tipe data** | **deskripsi** |
| --- | --- | --- |
| id\_penjualan | ke dalam | primary key, id unik untuk setiap transaksi |
| id\_pelanggan | ke dalam | foreign key dari tabel pelanggan (pelanggan) |
| id\_produk | ke dalam | foreign key dari tabel produk (produk) |
| kuantitas | ke dalam | jumlah produk yang dibeli |
| tanggal\_penjualan | tanggal | tanggal penjualan |
| jumlah\_total | desimal(10, 2) | total harga dari penjualan (kuantitas \* harga) |

penjelasan:

1. sale\_id: kolom ini adalah primary key untuk mengidentifikasi unik setiap transaksi penjualan.
2. customer\_id: foreign key yang merujuk ke kolom customer\_iddi tabel pelanggan untuk menentukan siapa yang membeli.
3. product\_id: foreign key yang merujuk ke kolom product\_iddi tabel produk untuk menentukan produk yang dijual.
4. quantity: jumlah unit produk yang dibeli dalam transaksi tersebut.
5. sale\_date: tanggal transaksi penjualan dicatat menggunakan tipe data date.
6. total\_amount: total harga dari transaksi, dihitung dari quantitydikalikan dengan price.

hubungan antar tabel:

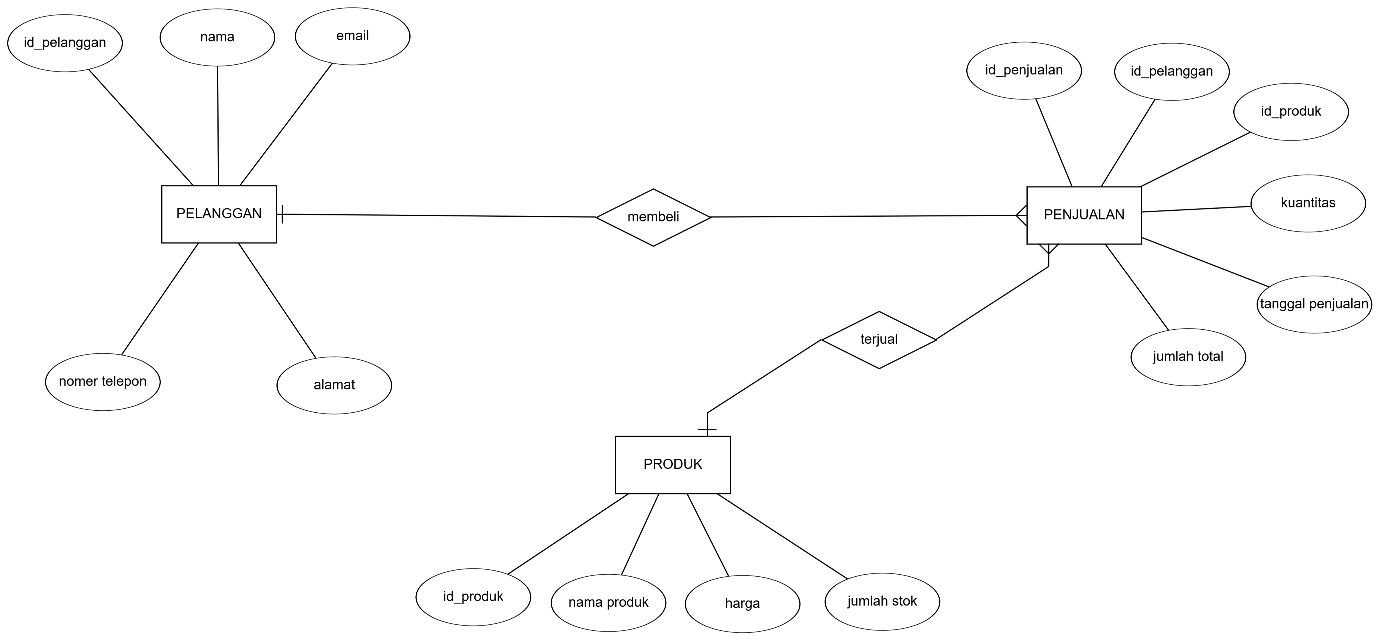
1. pelanggan dengan sales memiliki hubungan satu ke banyak (one-to-many), karena satu pelanggan dapat melakukan banyak transaksi.
2. produk dengan sales juga memiliki hubungan satu ke banyak (one-to-many), karena satu produk bisa terjual di banyak transaksi.

hubungan kardinalitas

1. pelanggan ke penjualan :
2. kardinalitas : satu ke banyak
3. deskripsi : satu pelanggan dapat melakukan banyak transaksi penjualan, tetapi setiap transaksi hanya dilakukan oleh satu pelanggan.
4. produk yang dijual :
5. kardinalitas : satu ke banyak
6. deskripsi : satu produk dapat terjual dalam banyak transaksi, tetapi setiap transaksi hanya mencakup satu produk tertentu. desain erd dalam erd, tabel-tabel dan relasi dapat digambarkan seperti berikut:

**buta desain erd dengan hubungan database penjualan di atas**

dalam erd, tabel-tabel dan relasi dapat digambarkan seperti berikut:



penjelasan erd:

1. pelanggan (pelanggan) memiliki customer\_id sebagai primary key (pk) yang berfungsi untuk mengidentifikasi pelanggan.
2. sales memiliki sale\_id sebagai pk dan menggunakan customer\_id serta product\_id sebagai foreign key (fk) untuk menghubungkan ke tabel customers dan products .
3. produk (produk) memiliki product\_id sebagai pk yang digunakan untuk mengidentifikasi produk.

tugas !!!

**membuat database perpajakan dilengkapi dengan erd**

**tugas tabel perpajakan**

untuk membuat database perpajakan, kita bisa membaginya menjadi 5 tabel utama, yang meliputi data wajib pajak, jenis pajak, transaksi pembayaran, riwayat pembayaran, dan tarif pajak. berikut ini adalah desain struktur tabel-tabel yang dapat digunakan:

**1. tabel wajib\_pajak**

tabel ini menyimpan data tentang pajak wajib, yang mencakup informasi dasar mereka.

| **nama kolom** | **tipe data** | **keterangan** |
| --- | --- | --- |
| id\_wajib\_pajak | int (kunci utama) | id unik untuk wajib pajak |
| nama | varchar(100) | nama lengkap wajib pajak |
| alamat | varchar(255) | alamat lengkap wajib pajak |
| npwp | varchar(20) | nomor npwp wajib pajak |
| telepon | varchar(15) | nomor telepon wajib pajak |
| e-mail | varchar(50) | email wajib pajak |

**2. tabel jenis\_pajak**

tabel ini menyimpan informasi jenis pajak yang ada.

| **nama kolom** | **tipe data** | **keterangan** |
| --- | --- | --- |
| id\_jenis\_pajak | int (kunci utama) | id unik untuk jenis pajak |
| nama\_pajak | varchar(50) | nama jenis pajak |
| deskripsi | teks | deskripsi mengenai jenis pajak |

**3. tabel tarif\_pajak**

tabel ini mencatat tarif pajak untuk berbagai jenis pajak, yang bisa bervariasi tergantung ketentuan yang berlaku.

| **nama kolom** | **tipe data** | **keterangan** |
| --- | --- | --- |
| id\_tarif | int (kunci utama) | id unik untuk tarif pajak |
| id\_jenis\_pajak | int (kunci asing) | id jenis pajak |
| tarif | desimal(5, 2) | tarif pajak dalam persen |
| tanggal\_berlaku | tanggal | tanggal mulai berlakunya tarif |

1. **tabel transaksi\_pembayaran**

tabel ini menyimpan informasi pembayaran yang dilakukan oleh wajib pajak.

| **nama kolom** | **tipe data** | **keterangan** |
| --- | --- | --- |
| id\_transaksi | int (kunci utama) | id unik untuk transaksi |
| id\_wajib\_pajak | int (kunci asing) | id wajib pajak yang melakukan pembayaran |
| id\_jenis\_pajak | int (kunci asing) | id jenis pajak yang diajukan |
| tanggal\_bayar | tanggal | tanggal pembayaran |
| jumlah\_bayar | desimal(15, 2) | jumlah yang berbahasa |
| status | varchar(20) | status pembayaran (misal: lunas, menunggu) |

**5. tabel riwayat\_pembayaran**

tabel ini mencatat riwayat pembayaran tiap wajib pajak, yang bisa berisi informasi status riwayat atau proses pembayaran.

| **nama kolom** | **tipe data** | **keterangan** |
| --- | --- | --- |
| id\_riwayat | int (kunci utama) | id unik untuk riwayat pembayaran |
| id\_transaksi | int (kunci asing) | id dari transaksi pembayaran |
| tanggal\_riwayat | tanggal | tanggal riwayat dicatat |
| keterangan | varchar(255) | keterangan terkait status atau proses pembayaran |

dengan desain ini, setiap tabel memiliki peran khusus dalam menyimpan informasi terkait perpajakan. struktur ini juga memungkinkan integrasi antar tabel melalui kunci asing, sehingga memudahkan pelacakan data transaksi, riwayat, dan wajib pajak secara keseluruhan.

jelaskan masing-masing dari table

**chatgpt mengatakan:**

berikut adalah penjelasan masing-masing tabel dalam database perpajakan:

**1. tabel wajib\_pajak**

tabel wajib\_pajakmenyimpan informasi dasar dari setiap individu atau entitas yang menjadi subjek pajak. setiap wajib pajak memiliki data seperti nama, alamat, nomor telepon, dan informasi penting lainnya. data ini sangat berguna untuk mengidentifikasi dan menghubungi wajib pajak.

* **id\_wajib\_pajak** : id unik yang mengidentifikasi wajib pajak. menjadi kunci utama pada tabel ini.
* **nama** : nama lengkap dari pajak wajib.
* **alamat** : alamat lengkap wajib pajak, yang bisa digunakan untuk mengirim pemberitahuan atau korespondensi.
* **npwp** : nomor pokok wajib pajak yang bersifat unik untuk setiap wajib pajak.
* **telepon** : nomor telepon yang dapat dihubungi.
* **email** : alamat email wajib pajak yang berguna untuk pemberitahuan elektronik atau komunikasi lain.

**2. tabel jenis\_pajak**

tabel jenis\_pajakberisi informasi tentang berbagai jenis pajak yang berlaku. jenisnya mencakup pajak penghasilan, pajak pertambahan nilai, pajak bumi dan bangunan, dan sebagainya. dengan tabel ini, kita dapat mengelompokkan transaksi pajak berdasarkan jenisnya.

* **id\_jenis\_pajak** : id unik untuk mengidentifikasi jenis pajak, yang menjadi kunci utama.
* **nama\_pajak** : nama dari jenis pajak, seperti pajak penghasilan, pajak pertambahan nilai, dll.
* **deskripsi** : penjelasan singkat mengenai jenis pajak untuk memahami tujuan atau mencakup pajak tersebut.

**3. tabel tarif\_pajak**

tabel tarif\_pajakmenyimpan data tarif atau persentase pajak untuk setiap jenis pajak. tarif pajak dapat berubah sesuai dengan kebijakan yang berlaku. dengan tabel ini, kita bisa melacak tarif yang berlaku pada periode tertentu.

* **id\_tarif** : id unik untuk setiap tarif, yang menjadi kunci utama.
* **id\_jenis\_pajak** : kunci asing yang merujuk ke tabel jenis\_pajakuntuk menunjukkan jenis pajak yang diberlakukan tarif tersebut.
* **tarif** : nilai persentase tarif pajak untuk jenis pajak tertentu. misalnya, ppn mungkin memiliki tarif 10%.
* **tanggal\_berlaku** : tanggal mulai berlakunya tarif ini. berguna untuk melacak perubahan tarif pajak dari waktu ke waktu.

**4. tabel transaksi\_pembayaran**

tabel transaksi\_pembayaranpencatatan setiap transaksi pembayaran pajak yang dilakukan oleh wajib pajak. tabel ini penting untuk mengelola data pembayaran pajak dan mengetahui jumlah yang telah dipesan oleh wajib pajak.

* **id\_transaksi** : id unik untuk setiap transaksi pembayaran, yang menjadi kunci utama.
* **id\_wajib\_pajak** : kunci asing yang merujuk ke tabel wajib\_pajak, yang menunjukkan siapa yang melakukan pembayaran.
* **id\_jenis\_pajak** : kunci asing ke tabel jenis\_pajak, menunjukkan jenis pajak yang dibayar.
* **tanggal\_bayar** : tanggal pembayaran dilakukan.
* **jumlah\_bayar** : jumlah uang yang dibayarkan oleh wajib pajak dalam transaksi ini.
* **status** : status pembayaran, seperti "lunas" atau "menunggu", untuk menunjukkan apakah pembayaran sudah selesai atau belum.

**5. tabel riwayat\_pembayaran**

tabel riwayat\_pembayaranmenyimpan riwayat atau catatan perubahan status dari setiap transaksi pembayaran. tabel ini memungkinkan kita melacak berbagai perubahan status atau tahapan dari transaksi pembayaran.

* **id\_riwayat** : id unik untuk setiap catatan riwayat, yang menjadi kunci utama.
* **id\_transaksi** : kunci asing yang merujuk ke transaksi\_pembayaran, untuk mengidentifikasi transaksi yang dicatat dalam riwayat ini.
* **tanggal\_riwayat** : tanggal pencatatan riwayat, berguna untuk mengetahui waktu perubahan status atau tahapan pembayaran.
* **keterangan** : informasi tambahan tentang status atau proses pembayaran, misalnya "pembayaran diterima", "proses verifikasi", atau "pembayaran ditolak".

dengan struktur tabel ini, data pajak dapat dikelola dan diproses secara sistematis, memungkinkan pencarian informasi yang lebih efisien dan akurat.

**anda berkata:**

buatkan hubungan kardinalitas antar tabel

**chatgpt mengatakan:**

chatgpt

berikut adalah hubungan kardinalitas antar tabel pada database perpajakan yang telah dijelaskan:

**1. hubungan wajib\_pajakdantransaksi\_pembayaran**

* **kardinalitas** : satu ke banyak (1)
* **penjelasan** : setiap wajib pajak dapat melakukan lebih dari satu transaksi pembayaran. oleh karena itu, terdapat hubungan satu ke banyak antara tabel wajib\_pajakdan transaksi\_pembayaran.
* **implementasi** : kolom id\_wajib\_pajakpada tabel transaksi\_pembayaranmerupakan foreign key yang merujuk ke kolom id\_wajib\_pajakpada tabel wajib\_pajak.

**2. hubungan jenis\_pajakdantransaksi\_pembayaran**

* **kardinalitas** : satu ke banyak (1)
* **penjelasan** : setiap jenis pajak dapat memiliki banyak transaksi pembayaran yang terkait dengannya. hal ini berarti satu jenis pajak dapat dibayar oleh banyak wajib pajak.
* **implementasi** : kolom id\_jenis\_pajakdi tabel transaksi\_pembayaranadalah foreign key yang merujuk ke kolom id\_jenis\_pajakdi tabel jenis\_pajak.

**3. hubungan jenis\_pajakdantarif\_pajak**

* **kardinalitas** : satu ke banyak (1)
* **penjelasan** : setiap jenis pajak dapat memiliki beberapa tarif pajak yang berlaku, misalnya karena adanya perubahan tarif pajak dari waktu ke waktu.
* **implementasi** : kolom id\_jenis\_pajakdi tabel tarif\_pajakadalah foreign key yang merujuk ke kolom id\_jenis\_pajakdi tabel jenis\_pajak.

**4. hubungan transaksi\_pembayarandanriwayat\_pembayaran**

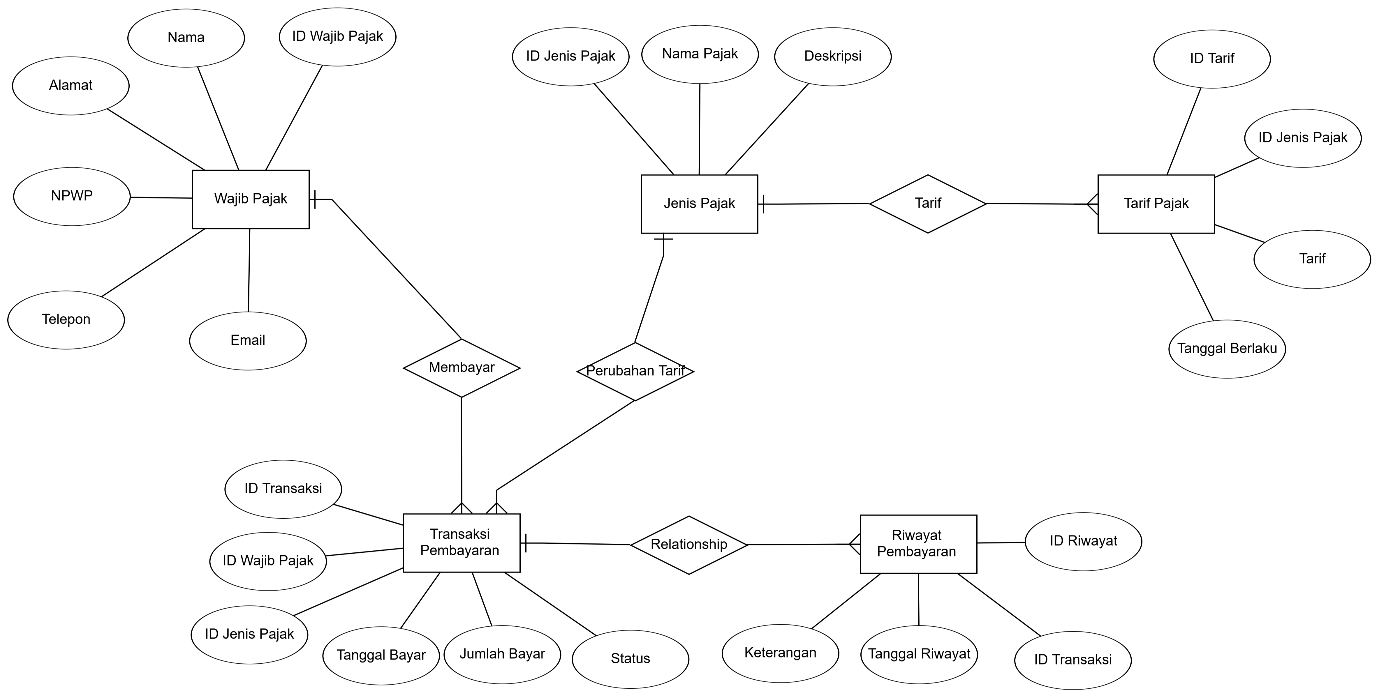
* **kardinalitas** : satu ke banyak (1)
* **penjelasan** : setiap transaksi pembayaran dapat memiliki beberapa riwayat pembayaran yang mencatat perubahan status atau tahapan pembayaran.
* **implementasi** : kolom id\_transaksipada tabel riwayat\_pembayaranadalah foreign key yang merujuk ke kolom id\_transaksipada tabel transaksi\_pembayaran.

**gambaran hubungan antar tabel**

* **wajib\_pajak (1) ---- (n) transaksi\_pembayaran**
* **jenis\_pajak (1) ---- (n) transaksi\_pembayaran**
* **jenis\_pajak (1) ---- (n) tarif\_pajak**
* **transaksi\_pembayaran (1) ---- (n) riwayat\_pembayaran**

dengan hubungan kardinalitas ini, data perpajakan dapat dihubungkan dan diakses secara efisien untuk memudahkan pengelolaan data transaksi, wajib pajak, dan tarif yang berlaku.

**desain erd dengan hubungan database perpajakan di atas**

****