**PROJECT BASIS DATA**

**PADA SISTEM INFORMASI (APLIKASI)**

**PENJUALAN ALAT TULIS SMKN 1 KARANG BARU**

****

**OLEH:**

**AHMAD ARJUN TRISULA**

**NISN. 0083825885**

**REKAYASA PERANGKAT LUNAK**

**SMK NEGERI 1 KARANG BARU**

**PEMERINTAH PROVINSI ACEH**

**2024**

**Langkah-langkah Membuat ERD**

1. Menentukan entitas
2. Menentukan atribut termasuk atribut kunci (Primary key)
3. Identifikasi relasi
4. Menentukan kardinalitas
5. **Menentukan Entitas**

Berdasarkan aturan-aturan yang di definisikan di atas dapat kita tentukan jumlah entitas ada sebanyak 4 yakni:



1. **Menentukan Relasi & Kardinalitasny**



1. **Menentukan Atribut**

Selanjutnya dari Keempat entitas tersebut kita jabarkan atribut-atribut yang melekat pada masing-masing entitas. Atribut yang bersifat unik akan di jadikan sebagai atribut kunci (*primary key*).

**1. Pelanggan (S1)**

**Pelanggan (S1)**

* id\_pelanggan int 11 not null primarykey auto increment
* nama\_pelanggan varchar 30
* desa\_pelanggan varchar 50
* kec\_pelanggan varchar 30
* hp\_pelanggan varchar 30

**2. Petugas (S2)**

* id\_petugas int 11 not null primarykey auto increment
* nama\_petugas varchar 30 not null
* desa\_petugas varchar 50 not null
* kec\_petugas varchar 30 not null
* hp\_petugas varchar 30 not null
* jabatan varchar 30 not null
* username varchar 6
* password varchar 5

**3. Layanan**

* id\_layanan int 11 not null primarykey auto increment
* nama\_layanan varchar 50 not null
* harga int 11 not null

**4. Transaksi**

* id\_transaksi int 11 not null primarykey auto increment
* id\_pelanggan int 11 not null foreignkey
* id\_petugas int 11 not null foreignkey
* id\_barang int 11 not null foreignkey
* tanggal date not null
* jumlah int 11 not null

*Atribut dengan kode (PK) akan menjadi atribut kunci (primary key) pada masing-masing entitas.*

**Hasil ERD Penjualan Layanan Showroom Honda**

Dari tahap-tahap di atas maka dapat di buat rancangan ERD petugasan layanan Bengkel komputer adalah sebagai berikut:

**ERD Showroom Honda Sebelum Normalisasi**



**ERD Showroom Honda Setelah Normalisasi**



**DESAIN LOGIKAL**

Desain logikal yaitu proses pembuatan model dari informasi yang digunakan perusahaan berdasarkan model dan data spesifik. Deskripsi implementasi *database* berdasarkan hasil desain logikal dengan *Entity Relationship Diagram* (ERD) pada *Database Management System* (DBMS) menghasilkan ERT sebagai berikut



**DESAIN FISIKAL & SOURCE SQL**

**Menggambarkan Rancangan Entitas Pada Basisdata Secara Fisikal (Physical Data Disaign) serta Membuat Source SQL pembuatan Masing-Masing Tabel/Entitas**

**Tabel pembeli**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Nama kolom** | **Tipe data** | **Lebar** | **Null** | **kunci** | **keterangan** |
| 1 | **Idpembeli** | Int | 11 | no | Pk | Ai |
| 2 | namapembeli | varchar | 30 |  |  |  |
| 3 | alamatpembeli | varchar | 100 |  |  |  |

create Table pembeli (

Idpembeli int(11)not null auto\_increment,

namapembeli varchar (30),

alamat varchar (100),

primary key (id\_pembeli)

);

**Tabel admin**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Nama kolom** | **Tipe data** | **Lebar** | **Null** | **kunci** | **keterangan** |
|  | **Idadmin** | int | 11 |  |  |  |
|  | namaadmin | vachar | 30 |  |  |  |
|  | username | vachar | 100 |  |  |  |
|  | Password | vachar | 30 |  |  |  |
|  | alamat | vachar | 100 |  |  |  |

create table admin (

idadmin int(11)primary key not null auto\_increment,

namaadmin varchar (30),

username varchar (100),

password varchar (30),

alamat varchar (100)

);

**Tabel barang**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Nama kolom** | **Tipe data** | **Lebar** | **Null** | **kunci** | **keterangan** |
| 1 | **idbarang** | int | 11 | no | pk | ai |
| 2 | idsuplier | int | 11 | no | fk |  |
| 3 | Namabarang | varchar | 100 |  |  |  |
| 4 | Harga | float | 15 |  |  |  |
| 5 | stok | int | 11 |  |  |  |

create table barang (

idbarang int(11)not null primary key auto\_increment,

idsuplier int (11),

namabarang varchar (100),

harga float (15),

stok int (11),

constraint idsuplier foreign key (idsuplier) references suplier (idsuplier)

);

**Tabel suplier**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Nama kolom** | **Tipe data** | **Lebar** | **Null** | **kunci** | **keterangan** |
| 1 | **idsuplier** | int | 11 | no | pk | ai |
| 2 | namasuplier | vachar | 30 | no |  |  |
| 3 | alamat | vachar | 100 | n |  |  |

create table suplier (

idsuplier int(11)not null primary key auto\_increment,

namasuplier varchar (30) ,

alamat int(20) not null

);

**Tabel transaksi**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Nama kolom** | **Tipe data** | **Lebar** | **Null** | **kunci** | **keterangan** |
| 1 | **Idtransaksi** | int | 11 | no | pk | ai |
| 2 | *Idbarang* | int | 11 | no | fk |  |
| 3 | *Idadmin* | int | 11 | no | fk |  |
| 4 | jumlah | int | 11 |  |  |  |

create table transaksi (

idtransaksi int(11)primary key not null auto\_increment,

idbarang int(11),

idadmin int(11),

jumlah int (11),

constraint idbarang foreign key (idbarang) references barang (idbarang),

constraint idadmin foreign key (idadmin) references admin (idadmin)

);

**Tabel detiltransaksi**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Nama kolom** | **Tipe data** | **Lebar** | **Null** | **kunci** | **keterangan** |
| **1** | **iddetiltransaksi** | **int** | **11** | nol | **pk** | **ai** |
| 2 | *Idtransaksi* | int | 11 | no | fk |  |
| 3 | *Idpembeli* | int | 11 | no | fk |  |
| 4 | total | float | 15 |  |  |  |
| 5 | tanggal | date |  |  |  |  |

create table detiltransaksi (

iddetiltransaksi int(11)primary key not null auto\_increment,

idtransaksi int(11),

idpembeli int(11),

total float (15),

taanggal date,

constraint idtransaksi foreign key (idtransaksi) references transaksi (idtransaksi),

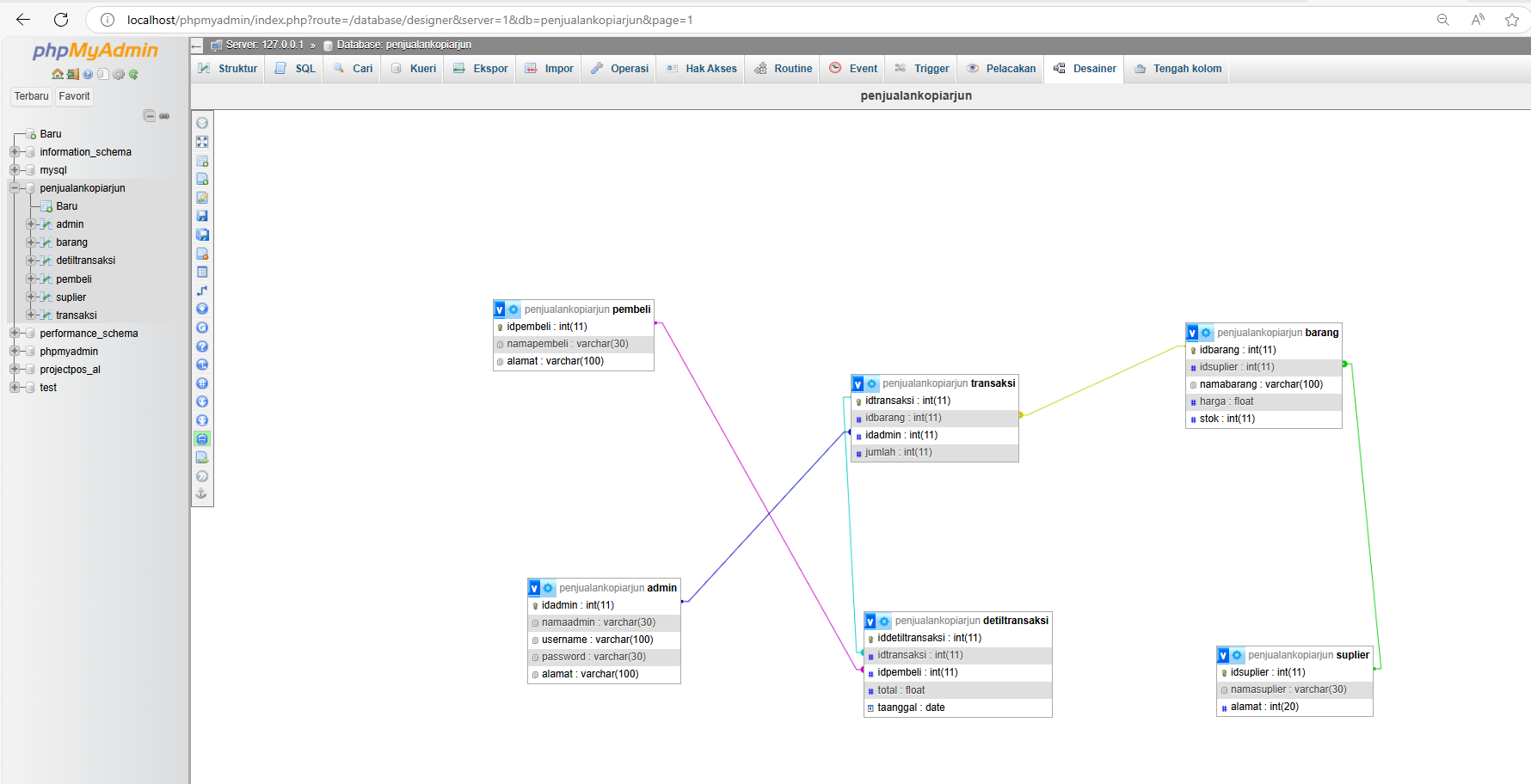
constraint idpembeli foreign key (idpembeli) references pembeli (idpembeli)

);

**Implementasi Syntax Sql Database Melalui Cmd**

1. **Membuat Basisdata**
2. **Membuat Tabel**
3. **Medeskripsikan tabel**

**Hasil Desain Konseptual Skema Relasi Database Di Phpmyadmin**

****

**MENGISI DATA RECORD**

**Cara Manual Dengan Source Code**

Memasukkan data dalam jumlah banyak

Insert into namatabel1

Values

(‘variable1’,’variable2’,’variabel3’,’dst’),

(‘variable1’,’variable2’,’variabel3’,’dst’),

(‘variable1’,’variable2’,’variabel3’,’dst’),

(‘variable1’,’variable2’,’variabel3’,’dst’);

Insert into layanan

Values

(' ','Instal Ulang (Paket Standar)','50000'),

(' ','Instal Ulang (Paket Komplit)','80000'),

(' ','Hang / Freeze','50000'),

(' ','Layar Biru / Bluescreen','50000'),

(' ','Lambat / Lemot','35000'),

(' ','Sering Mati Tiba Tiba','30000'),

(' ','Restart Tiba Tiba','25000'),

(' ','Blank / Tidak Ada Tampilan','40000'),

(' ','Ada Bunyi Tiiit','25000'),

(' ','Mati Total','200000'),

(' ','Kadang Nyala Kadang Tidak','80000'),

(' ','Ada Bunyi Berderik','25000'),

(' ','Laptop/Pc Kepanasan','25000'),

(' ','Tampilan Laptop Bergaris Horisontal/Vertical','100000'),

(' ','Gagal Instal Ulang','100000'),

(' ','Tidak Keluar Suara','100000'),

(' ','Tampilan Layar Besar Besar / Pecah Pecah','25000'),

(' ','Tidak Bisa Masuk Windows / Loading Terus','25000'),

(' ','Kena Virus (Pembersihan Virus)','25000'),

(' ','Tidak Bisa Koneksi Internet','25000'),

(' ','Hardware Tidak Terdeteksi','25000');



Insert into transaksi

Values

(' ','2','2','2021-01-22'),

(' ','3','3','2021-01-23'),

(' ','5','1','2021-01-24'),

(' ','4','4','2021-01-25'),

(' ','6','1','2021-01-26'),

(' ','4','2','2021-01-27'),

(' ','5','3','2021-01-28'),

(' ','5','1','2021-01-29'),

(' ','2','4','2021-01-30'),

(' ','5','1','2021-01-31'),

(' ','2','2','2021-02-01'),

(' ','5','3','2021-02-02'),

(' ','5','1','2021-02-03'),

(' ','2','4','2021-02-04'),

(' ','5','1','2021-02-05'),

(' ','2','2','2021-02-06'),

(' ','5','3','2021-02-07'),

(' ','2','1','2021-02-08'),

(' ','5','4','2021-02-09'),

(' ','2','1','2021-02-10');

Insert into detiltransaksi

Values

('2','2','2','100000'),

('3','3','3','100000'),

('5','5','5','200000'),

('2','4','4','100000'),

('3','7','7','100000'),

('5','4','4','100000'),

('5','5','5','200000'),

('5','5','5','100000'),

('2','2','2','100000'),

('5','5','5','200000'),

('2','2','2','100000'),

('2','2','2','100000'),

('2','3','3','200000'),

('3','5','5','100000'),

('5','4','4','100000'),

('7','7','7','200000'),

('4','4','4','100000'),

('5','5','5','100000'),

('5','5','5','100000'),

('2','2','2','200000'),

('5','5','5','100000'),

('2','2','2','100000');

('2','2','2','100000'),

('3','3','3','100000'),

('5','5','5','200000'),

('2','4','4','100000'),

('3','7','7','100000'),

('5','4','4','100000'),

('5','5','5','200000'),

('5','5','5','100000'),

('2','2','2','100000'),

('5','5','5','200000'),

('2','2','2','100000'),

('2','2','2','100000'),

('2','3','3','200000'),

('3','5','5','100000'),

('5','4','4','100000'),

('7','7','7','200000'),

('4','4','4','100000'),

('5','5','5','100000'),

('5','5','5','100000'),

('2','2','2','200000'),

('5','5','5','100000'),

('2','2','2','100000'),

Melihat isi data / record yang telah dimasukkan

Select \*from namatabel1;



**Cara IMPORT Dari Excel**

Buka file yang sudah dieksport tadi



Import record yang sudah diisikan ke file csv tadi dengan cara: Masuk ke database dan buka tabel yang akan diimport datanya



Lalu cari file yang sudah diisi datanya tadi dan ubah format filenya menjadi csv Using Load Data



Jika sudah tekan go/kirim



Jika berhasil, maka akan tampil semua data record dari file csv yang sudah diisi tadi

Dan lakukan ke semua tabel yang ada



**SELECT:**

Bentuk umum.

Select \*from namatabel1

**Output :**

Memilih kolom tertentu.

SELECT nama\_kolom1, nama\_kolom2 FROM nama\_tabel;

**Output :**

Memilih kolom dengan mengurutkan dari terkecil.

SELECT \* FROM nama\_tabel order by kolom\_dipilih ASC;

**Output :**

Memilih kolom dengan mengurutkan dari terbesar.

SELECT \* FROM nama\_tabel order by kolom\_dipilih DESC;

**Output :**

**UPDATE**

Update sebaris data record.

UPDATE nama\_tabel SET kolom1=data1, kolom2=data2,... WHERE kolom=data;

**Output :**

**DELETE**

Delete sebaris data record.

DELETE FROM nama\_tabel WHERE kolom=data;

**Output :**

Delete sebuah tabel.

DELETE FROM nama\_tabel;

**Output :**

**SELECT SUBQUERY:**

SELECT nama\_produk FROM ms\_produk LIMIT 3;

**Output :**

SELECT nama\_layanan FROM layanan LIMIT 3;

SELECT ms\_produk.nama\_produk AS nama from ms\_produk;

SELECT layanan.nama\_layanan AS layanan\_yang\_tersedia from layanan;

**Output :**

SELECT \* FROM ms\_produk AS t1;

## SELECT \* FROM petugas AS P1;

**Output :**

SELECT t1.kode\_produk, t1.nama\_produk FROM ms\_produk AS t1;

SELECT t1.kode\_layanan, t1.nama\_layanan FROM layanan AS t1;

**Output :**

SELECT \* FROM ms\_produk WHERE nama\_produk = 'Gantungan Kunci DQLab';

SELECT \* FROM layanan WHERE nama\_layanan = ‘ganti Hardisk';

**Output :**

SELECT \* FROM ms\_produk WHERE harga < 50000;

SELECT \* FROM layanan WHERE harga < 50000;

**Output :**

SELECT \* FROM ms\_produk WHERE nama\_produk = 'Gantungan Kunci DQLab' AND harga < 50000;

SELECT \* FROM layanan WHERE nama\_layanan = 'Lambat / Lemot ' AND harga\_layanan < 50000;

Output :

**SELECT nama\_layanan, COUNT(id\_layanan) as jumlah FROM layanan WHERE harga\_layanan=’50000’ GROUP BY harga\_layanan HAVING COUNT(id\_layanan)>2;**

**Output :**

**SELECT JOINT MULTITABLE:**

SELECT pelanggan.nama\_pelanggan, pembayaran.total\_bayar

FROM pelanggan INNER JOIN pembayaran ON pelanggan.id\_pelanggan=pembayaran.id\_pelanggan;  
**Output :**

SELECT petugas.nama\_petugas, layanan.nama\_layanan, transaksi.jumlah

FROM petugas

INNER JOIN transaksi ON petugas.id\_petugas = transaksi.id\_petugas

INNER JOIN layanan ON transaksi.id\_layanan = layanan.id\_layanan;

SELECT pelanggan.id\_pelanggan, pelanggan.nama\_pelanggan, pembayaran.id\_bayar, pembayaran.tanggal\_bayar FROM pelanggan LEFT JOIN pembayaran ON pelanggan.id\_pelanggan=pembayaran.id\_pelanggan;

**Output :**

SELECT pelanggan.id\_pelanggan, pelanggan.nama\_pelanggan, pembayaran.id\_bayar, pembayaran.tanggal\_bayar FROM pelanggan RIGHT JOIN pembayaran ON pelanggan.id\_pelanggan=pembayaran.id\_pelanggan;

**Output :**

select a.\*, b.\*

from pelanggan a

join pembayaran b using (id\_pelanggan);

**Output :**

select a.\*, b.\*

from pelanggan a

left join pembayaran b

using (id\_pelanggan);

**Output :**

select a.\*, b.\*

from pelanggan a

right join pembayaran b

using (id\_pelanggan);

**Output :**

select a.\*, b.\*

from pelanggan a

right join pembayaran b

using (id\_pelanggan)

union all

select a.\*, b.\*

from pelanggan a

left join pembayaran b

using (id\_pelanggan);

select a.\*, b.\*

from pelanggan a

right join pembayaran b

using (id\_pelanggan)

union all

select a.\*, b.\*

from pelanggan a

left join pembayaran b

using (id\_pelanggan);

**Output :**

select a.nama\_pelanggan, b.total\_bayar

from pelanggan a

right join pembayaran b

using (id\_pelanggan)

union all

select a.nama\_pelanggan, b.total\_bayar

from pelanggan a

left join pembayaran b

using (id\_pelanggan);

select a.nama\_pelanggan, b.tanggal

from pelanggan a

right join transaksi b

using (id\_pelanggan)

union all

select a.nama\_pelanggan, b.tanggal

from pelanggan a

left join transaksi b

using (id\_pelanggan);

**Output :**