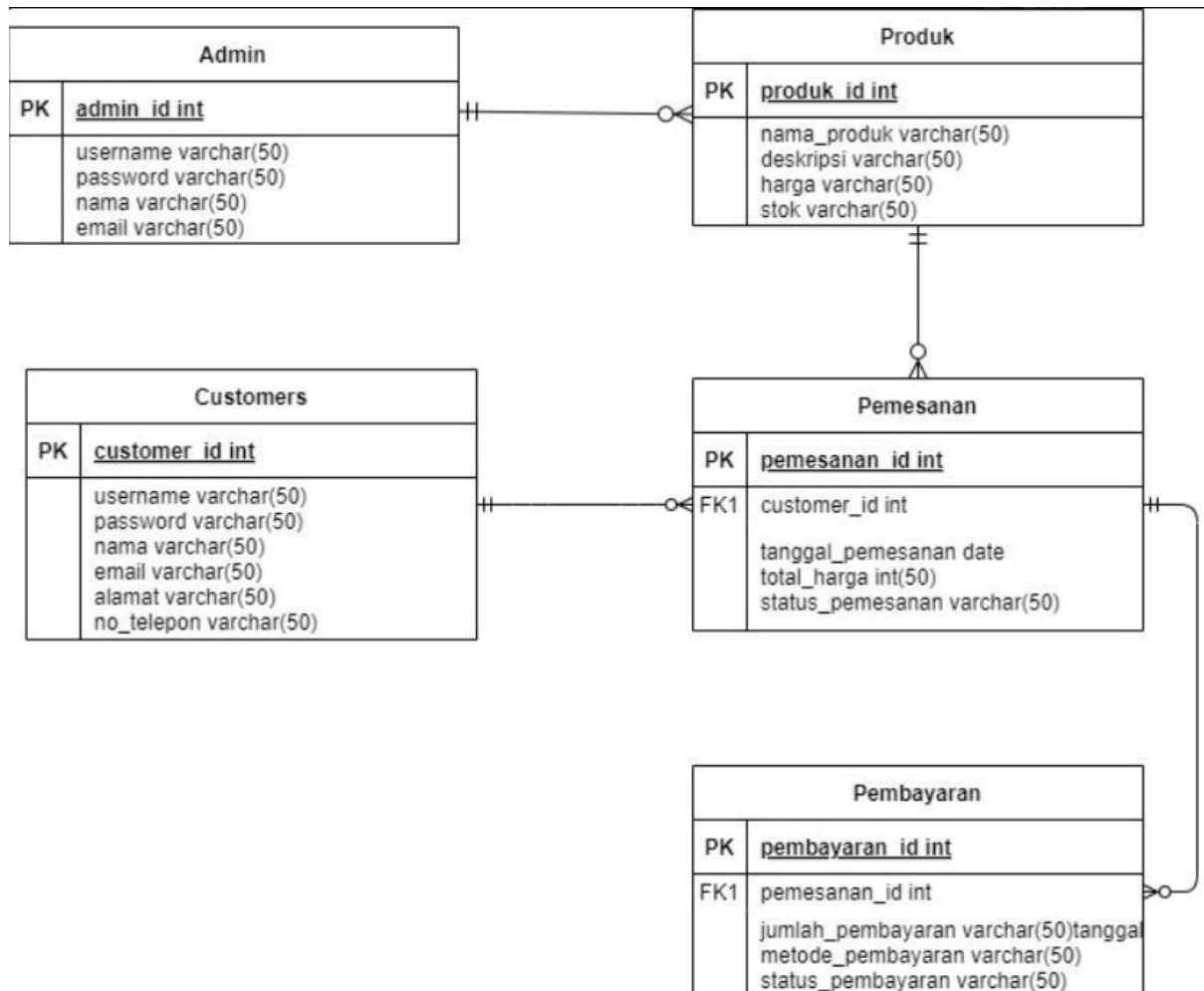


CONVERT CLASS DIAGRAM KE CODE



```

<?php
// Koneksi Database class Database {
private $host = "localhost"; private
$username = "root"; private $password
= ""; private $database =
"online_shop"; protected $connection;

    public function __construct() {
        try {
            $this->connection = new mysqli($this->host, $this->username,
            $this->password, $this->database); if ($this->connection->
            connect_error) { throw new
            Exception("Connection failed: " .
            $this->connection->connect_error);
        }
        } catch (Exception $e) { echo "Error:
        " . $e->getMessage(); }
    }
}

// Class Admin class Admin
extends Database { private
$admin_id; private $username;
private $password; private
$nama; private $email;

    public function __construct() {
        parent::__construct();
    }

    public function login($username, $password) {
        $query = "SELECT * FROM admin WHERE username = ? AND password =
        ?";
        $stmt = $this->connection->prepare($query);
        $stmt->bind_param("ss", $username, $password);
        $stmt->execute();
        $result = $stmt->get_result();

        if($result->num_rows == 1) {
            $row = $result->fetch_assoc();
            $this->admin_id = $row['admin_id'];
            $this->username = $row['username'];
            $this->nama = $row['nama']; $this-
            >email = $row['email']; return
            true;
        }
    }
}

```

```

    }
    return false;
}

public function register($username, $password, $nama, $email) {
    $query = "INSERT INTO admin (username, password, nama, email)
VALUES (?, ?, ?, ?)";
    $stmt = $this->connection->prepare($query);
    $stmt->bind_param("ssss", $username, $password, $nama, $email);
    return $stmt->execute();
}
}

// Class Produk class Produk
extends Database { private
$produk_id; private
$nama_produk; private
$deskripsi; private $harga;
private $stok;

    public function __construct() {
        parent::__construct();
    }

    public function tambahProduk($nama_produk, $deskripsi, $harga,
$stok) {
        $query = "INSERT INTO produk (nama_produk, deskripsi, harga,
stok)
VALUES (?, ?, ?, ?)";
        $stmt = $this->connection->prepare($query);
        $stmt->bind_param("ssss", $nama_produk, $deskripsi, $harga,
$stok); return $stmt->execute();
    }
}

```

```

    }

    public function updateProduk($produk_id, $nama_produk, $deskripsi,
    $harga, $stok) {
        $query = "UPDATE produk SET nama_produk=?, deskripsi=?, harga=?,
    stok=? WHERE produk_id=?";
        $stmt = $this->connection->prepare($query);
        $stmt->bind_param("ssssi", $nama_produk, $deskripsi, $harga,
        $stok,
    $produk_id); return $stmt-
        >execute();
    }

    public function getProduk($produk_id) {
        $query = "SELECT * FROM produk WHERE produk_id = ?";
        $stmt = $this->connection->prepare($query);
        $stmt->bind_param("i", $produk_id);
        $stmt->execute();
        return $stmt->get_result()->fetch_assoc();
    }

    public function getAllProduk() {
        $query = "SELECT * FROM produk";
        $result = $this->connection->query($query);
        return $result->fetch_all(MYSQLI_ASSOC);
    }

    public function deleteProduk($produk_id) {
        $query = "DELETE FROM produk WHERE produk_id = ?";
        $stmt = $this->connection-
        >prepare($query); $stmt->bind_param("i",
        $produk_id); return $stmt->execute();
    }
}

// Class Customer class
Customer extends Database {
    private $customer_id; private
    $username; private $password;
    private $nama; private $email;
    private $alamat; private
    $no_telepon;

    public function __construct() {
        parent::__construct();
    }

    return
} public

function

```

```

        $query =
        public function register($username, $password, $nama, $email,
        $alamat,
        $no_telepon) {
            $query = "INSERT INTO customers (username, password, nama,
        email, alamat, no_telepon)
            VALUES (?, ?, ?, ?, ?, ?)";
            $stmt = $this->connection->prepare($query);
            $stmt->bind_param("sssss", $username, $password, $nama, $email,
        $alamat, $no_telepon);
            $stmt->execute();

            login($username, $password) {
                "SELECT * FROM customers WHERE username = ? AND password =
        ?";

            $stmt = $this->connection->prepare($query);
            $stmt->bind_param("ss", $username, $password);
            $stmt->execute();
            $result = $stmt->get_result();

            if($result->num_rows == 1) {
                $row = $result->fetch_assoc();
                $this->customer_id = $row['customer_id'];
                $this->username = $row['username'];
                $this->nama = $row['nama'];
                $this->email = $row['email'];
                $this->alamat = $row['alamat']; $this->
                no_telepon = $row['no_telepon']; return
                true;
            }
            return false;
        }

        public function updateProfile($customer_id, $nama, $email, $alamat,
        $no_telepon) {
            $query = "UPDATE customers SET nama=?, email=?, alamat=?,
        no_telepon=? WHERE customer_id=?";
            $stmt = $this->connection->prepare($query);
            $stmt->bind_param("ssssi", $nama, $email, $alamat, $no_telepon,
        $customer_id); return $stmt->
            execute();
        }
    }

    // Class Pemesanan class
    Pemesanan extends Database {

        return
    } public

    function

```

```

        $query =
private $pemesanan_id; private
$customer_id; private
$tanggal_pemesanan; private
$total_harga; private
$status_pemesanan;

    public function __construct() {
        parent::__construct();
    }

    public function buatPemesanan($customer_id, $total_harga) {
        $tanggal = date('Y-m-d H:i:s');
        $status = "PENDING";

        $query = "INSERT INTO pemesanan (customer_id, tanggal_pemesanan,
total_harga, status_pemesanan)
                VALUES (?, ?, ?, ?)";
        $stmt = $this->connection->prepare($query);
        $stmt->bind_param("isds", $customer_id, $tanggal, $total_harga,
$status); if($stmt->execute()) { return $this-
>connection->insert_id;
        } false;

        updateStatus($pemesanan_id, $status) {
            "UPDATE pemesanan SET status_pemesanan = ? WHERE
pemesanan_id = ?";
            $stmt = $this->connection->prepare($query); $stmt-
>bind_param("si", $status, $pemesanan_id); return
            $stmt->execute();
        }

    public function getPemesanan($pemesanan_id) {
        $query = "SELECT * FROM pemesanan WHERE pemesanan_id = ?";
        $stmt = $this->connection->prepare($query);
        $stmt->bind_param("i", $pemesanan_id);
        $stmt->execute();
        return $stmt->get_result()->fetch_assoc();
    }

    public function getPemesananByCustomer($customer_id) {
        $query = "SELECT * FROM pemesanan WHERE customer_id = ?";
        $stmt = $this->connection->prepare($query);
        $stmt->bind_param("i", $customer_id);
        $stmt->execute();
        return $stmt->get_result()-
>fetch_all(MYSQLI_ASSOC); }

    return
} public

function

```

```

        $query =
    }

    // Class Pembayaran class
    Pembayaran extends Database {
    private $pembayaran_id; private
    $pemesanan_id; private
    $jumlah_pembayaran; private
    $metode_pembayaran; private
    $status_pembayaran; private
    $tanggal;

        public function __construct() {
            parent::__construct();
        }

        public function prosesPembayaran($pemesanan_id, $jumlah_pembayaran,
        $metode_pembayaran) {
            $tanggal = date('Y-m-d H:i:s');
            $status = "PENDING";

            $query = "INSERT INTO pembayaran (pemesanan_id,
            jumlah_pembayaran, metode_pembayaran, status_pembayaran, tanggal)
            VALUES (?, ?, ?, ?, ?)";
            $stmt = $this->connection->prepare($query);
            $stmt->bind_param("idsss", $pemesanan_id, $jumlah_pembayaran,
            $metode_pembayaran, $status, $tanggal);
            return $stmt->execute();
        }

        public function updateStatus($pembayaran_id, $status) {
            $query = "UPDATE pembayaran SET status_pembayaran = ? WHERE
            pembayaran_id = ?";
            $stmt = $this->connection->prepare($query);
            $stmt->bind_param("si", $status, $pembayaran_id);
            $stmt->execute();

            getPembayaran($pembayaran_id) {

                return
            } public

            function

```

```

        "SELECT * FROM pembayaran WHERE pembayaran_id = ?";
$stmt = $this->connection->prepare($query);
$stmt->bind_param("i", $pembayaran_id);
$stmt->execute();
return $stmt->get_result()->fetch_assoc();
}

public function getPembayaranByPemesanan($pemesanan_id) {
    $query = "SELECT * FROM pembayaran WHERE pemesanan_id =
    ?";
    $stmt = $this->connection->prepare($query);
    $stmt->bind_param("i", $pemesanan_id);
    $stmt->execute();
    return $stmt->get_result()-
    >fetch_assoc(); }
}

// SQL untuk membuat database dan tabel-tabel yang diperlukan
/*
CREATE DATABASE online_shop;
USE online_shop;

CREATE TABLE admin ( admin_id INT PRIMARY
    KEY AUTO_INCREMENT, username
    VARCHAR(50) NOT NULL, password
    VARCHAR(50) NOT NULL, nama VARCHAR(50)
    NOT NULL, email VARCHAR(50) NOT NULL
);

CREATE TABLE produk ( produk_id INT PRIMARY
    KEY AUTO_INCREMENT, nama_produk
    VARCHAR(50) NOT NULL, deskripsi
    VARCHAR(50), harga VARCHAR(50) NOT NULL,
    stok VARCHAR(50) NOT NULL
);

CREATE TABLE customers ( customer_id INT
    PRIMARY KEY AUTO_INCREMENT, username
    VARCHAR(50) NOT NULL, password
    VARCHAR(50) NOT NULL, nama VARCHAR(50)
    NOT NULL, email VARCHAR(50) NOT NULL,
    alamat VARCHAR(50) NOT NULL, no_telepon
    VARCHAR(50) NOT NULL
);

CREATE TABLE pemesanan ( pemesanan_id INT
    PRIMARY KEY AUTO_INCREMENT, customer_id
    INT, tanggal_pemesanan DATE, total_harga
    INT(50), status_pemesanan VARCHAR(50),
    FOREIGN KEY (customer_id) REFERENCES customers(customer_id)
);

```


\$query =

```
CREATE TABLE pembayaran ( pembayaran_id INT  
    PRIMARY KEY AUTO_INCREMENT, pemesanan_id  
    INT, jumlah_pembayaran VARCHAR(50),
```

```
    metode_pembayaran VARCHAR(50),  
    status_pembayaran VARCHAR(50),  
    tanggal DATE,  
    FOREIGN KEY (pemesanan_id) REFERENCES pemesanan(pemesanan_id)  
);  
*/  
  
// Contoh penggunaan:  
/*  
// Membuat objek customer  
$customer = new Customer();  
  
// Registrasi customer baru  
$customer->register(  
    "ariana_grande",  
    "password123",  
    "Ariana Grande",  
    "arianaGrn@email.com",  
    "Jl. Hasym Ashari No. 123",  
    "08123456789"  
);  
  
// Login customer  
if($customer->login("ariana_grande", "password123")) {  
    echo "Login berhasil!";  
}  
  
// Membuat objek produk  
$produk = new Produk();  
  
// Menambah produk baru  
$produk->tambahProduk(  
    "Laptop Gaming",  
    "Laptop gaming dengan spesifikasi tinggi",  
    "15000000",  
    "10"  
);  
  
// Membuat pemesanan  
$pemesanan = new Pemesanan();  
$pemesanan_id = $pemesanan->buatPemesanan(1, 15000000);  
  
// Proses pembayaran  
if($pemesanan_id) {  
    $pembayaran = new Pembayaran();  
    $pembayaran->prosesPembayaran($pemesanan_id, 15000000,  
    "TRANSFER_BANK");  
}  
*/
```

?>