

## JAVA

1. D. `i = 5` and `j = 6`
2. C. After the `start()` method completes
3. D. `new Alpha1().get("veggie").soundOff();`
4. B. Class B's constructor has no arguments  
D. Class B's constructor includes a call to `super()`
5. C. Compilation fails
6. Method adalah fungsi yang terdapat di dalam sebuah kelas yang menentukan aksi atau perilaku yang dapat dilakukan oleh objek dari kelas tersebut.

Atribut adalah variabel dalam sebuah kelas yang digunakan untuk menyimpan data atau properti yang dimiliki oleh objek dari kelas tersebut.

7. Enkapsulasi merupakan sebuah konsep yang digunakan untuk menyembunyikan detail implementasi suatu kelas dan membatasi akses ke data atau metode di dalamnya, enkapsulasi sendiri biasanya diterapkan dengan modifier akses seperti `private`, `protected`, dan `public`.

Inheritance merupakan sebuah konsep yang memungkinkan suatu subkelas (child class) mewarisi atribut dan metode dari kelas induknya (parent class), dengan menggunakan inheritance kode dapat lebih terorganisir karena mendukung prinsip reusability.

## MySQL

1. Right Join : Mengambil semua data dari tabel yang terletak di sebelah kanan dan hanya data yang cocok dari tabel yang terletak di sebelah kiri.

Left Join : Mengambil semua data dari tabel yang terletak di sebelah kiri dan hanya data yang cocok dari tabel yang terletak di sebelah kanan.

Inner Join : Mengambil data yang hanya memiliki kecocokan antara dua tabel berdasarkan kondisi.

Outer Join : Mengambil semua data dari kedua tabel, apabila tidak memiliki kecocokan maka tabel yang tidak cocok akan berisi null.

Contoh Query (Menampilkan daftar janji temu antara pasien dan dokter):

Right Join :

SELECT

```
doctors.doctor_id,  
doctors.name AS doctor_name,  
doctors.specialty,  
appointments.appointment_date,  
patients.name AS patient_name
```

FROM doctors

RIGHT JOIN appointments ON doctors.doctor\_id = appointments.doctor\_id

RIGHT JOIN patients ON appointments.patient\_id = patients.patient\_id;

Query di atas menghubungkan tabel appointments dengan patients dan doctors untuk menampilkan daftar dokter termasuk yang belum memiliki janji temu.

Left Join :

SELECT

patients.patient\_id,

patients.name AS patient\_name,

appointments.appointment\_date,

doctors.name AS doctor\_name

FROM patients

LEFT JOIN appointments ON patients.patient\_id = appointments.patient\_id

LEFT JOIN doctors ON appointments.doctor\_id = doctors.doctor\_id;

Query di atas menghubungkan tabel appointments dengan patients dan doctors untuk menampilkan daftar pasien termasuk yang belum memiliki janji temu.

Inner Join :

SELECT

appointments.appointment\_id,

patients.name AS patient\_name,

doctors.name AS doctor\_name,

doctors.specialty,

appointments.appointment\_date

FROM appointments

INNER JOIN patients ON appointments.patient\_id = patients.patient\_id

INNER JOIN doctors ON appointments.doctor\_id = doctors.doctor\_id;

Query di atas menghubungkan tabel appointments dengan patients dan doctors untuk menampilkan daftar dokter dan pasien yang sudah memiliki janji temu.

Outer Join :

SELECT

patients.name AS patient\_name,

```

doctors.name AS doctor_name,
appointments.appointment_date
FROM patients
LEFT JOIN appointments ON patients.patient_id = appointments.patient_id
LEFT JOIN doctors ON appointments.doctor_id = doctors.doctor_id
UNION
SELECT
patients.name AS patient_name,
doctors.name AS doctor_name,
appointments.appointment_date
FROM doctors
RIGHT JOIN appointments ON doctors.doctor_id = appointments.doctor_id
RIGHT JOIN patients ON appointments.patient_id = patients.patient_id;

```

Query di atas menghubungkan tabel appointments dengan patients dan doctors untuk menampilkan semua pasien dan dokter termasuk yang belum memiliki janji temu.

2. Query untuk menampilkan data janji temu pada tanggal 21 Februari 2025.

```
SELECT * FROM appointments WHERE appointment_date = '2025-02-21';
```

## PHP

1. Fungsi di PHP dapat dibuat dengan menggunakan keyword function, adapun cara untuk memanggilnya adalah dengan menggunakan echo dan menyertakan() pada function yang kita panggil.

```

<?php
function greetPatient($name) {
    return "Halo, $name! selamat datang di klinik Jambu";
}
echo greetPatient("Bronto")
?>

```

2. Koneksi PHP ke MySQL

```

<?php
$servername = "localhost";
$username = "root";

```

```

$password = "";
$database = "klinik";
$conn = new mysqli($servername, $username, $password, $database);
if ($conn->connect_error) {
    die("Koneksi gagal: " . $conn->connect_error);
}
?>

```

### 3. Framework PHP yang saya ketahui beserta kelebihanannya:

#### a. CodeIgniter

CodeIgniter adalah framework PHP memiliki keunggulan di ukurannya yang kecil, hanya sekitar 2MB. Framework ini menggunakan arsitektur Model-View-Controller (MVC) yang memudahkan pengembangan aplikasi dengan pemisahan logika bisnis dan tampilan. Salah satu keunggulan utama CodeIgniter adalah kemudahan konfigurasi dan dokumentasi yang jelas, sehingga cocok bagi pemula maupun developer yang ingin membuat aplikasi dengan cepat tanpa harus bergantung pada banyak dependensi. CodeIgniter memiliki performa yang sangat baik untuk aplikasi kecil hingga menengah yang membutuhkan kecepatan tinggi karena tidak terlalu banyak menggunakan resource.

#### b. Laravel

Laravel memiliki keunggulan utama pada Eloquent ORM, yang menyederhanakan interaksi dengan database menggunakan sintaks yang sederhana. Selain itu, Laravel dilengkapi dengan Blade Templating Engine yang membantu dalam penyusunan tampilan antarmuka. Laravel juga memiliki fitur seperti routing yang fleksibel, middleware, queue, serta sistem autentikasi bawaan, sehingga mendukung pengembangan aplikasi yang kompleks. Dengan komunitas yang luas dan dokumentasi yang detail, Laravel dapat menjadi pilihan untuk proyek dengan skala menengah hingga besar.

#### c. Symfony

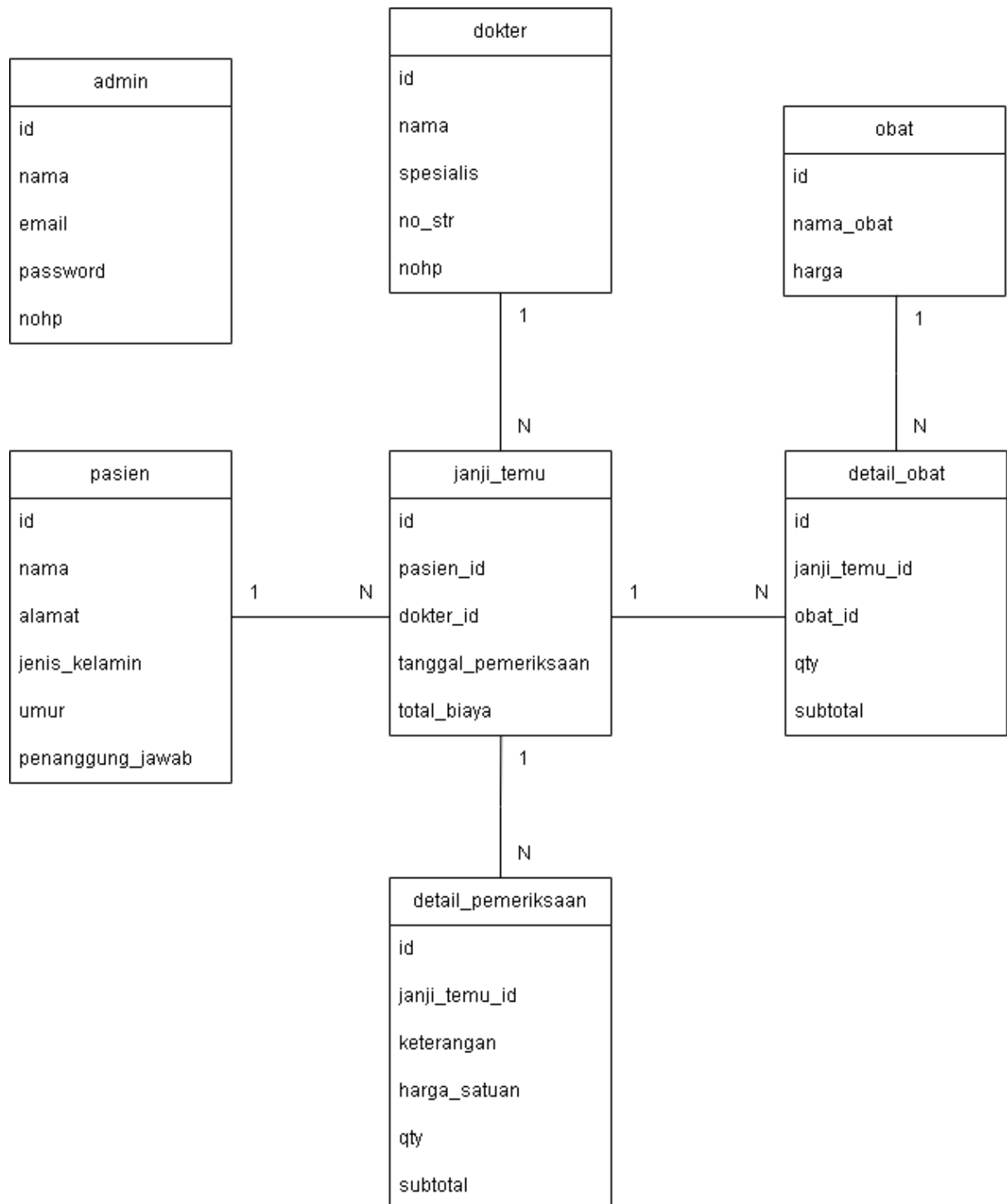
Keunggulan utama Symfony adalah reusability komponen, di mana banyak bagian dari framework ini dapat digunakan kembali dalam proyek lain atau bahkan diintegrasikan dengan framework lain seperti Laravel. Symfony juga memiliki dukungan penuh terhadap standar PSR (PHP Standard Recommendation), sehingga memudahkan developer dalam membangun aplikasi yang sesuai dengan best practices pada industri. Symfony juga memiliki sistem dependency injection yang kuat, sehingga dapat memberikan kontrol lebih terhadap pengelolaan layanan dalam aplikasi. Meski memiliki kurva pembelajaran yang cukup tinggi, Symfony sangat cocok untuk proyek besar yang memerlukan kestabilan dan skalabilitas tinggi.

4. Untuk mendukung tampilan web, selain PHP, saya menggunakan HTML sebagai dasar struktur halaman dan CSS untuk mengatur desain antarmuka. Selain itu, saya memanfaatkan CSS Framework seperti Bootstrap atau Tailwind CSS yang menyediakan berbagai komponen siap pakai, sehingga dapat mempercepat proses pengembangan. JavaScript juga saya gunakan

untuk menambahkan interaktivitas serta meningkatkan pengalaman pengguna dalam aplikasi web.

### Studi Kasus

1. Buatlah LRS (Logical Record Structure) berdasarkan nota tersebut.



2. Buatlah tampilan program tersebut berdasarkan LRS yang telah dibuat.

## Beranda



## Daftar Dokter

**Klinik Jambu** Daftar Dokter

Search for something

**Tambah Dokter**

Form fields for adding a doctor: Nama, No. Str, Spesialis, and Jadwal Tersedia.

**Daftar Dokter**

No.	Nama	No. Str	Spesialis	Jadwal	Aksi
1.	Thania	#12548796	Spesialis THT	Jumat - Minggu	Edit Hapus
2.	Raka	#12548796	Dokter Umum	Rabu - Jumat	Edit Hapus
3.	Shinta	#12548796	Spesialis Anak	Seni - Kamis	Edit Hapus
4.	Wilson	#12548796	Spesialisin Anak	Selasa - Kamis	Edit Hapus
5.	William	#12548796	Dokter Umum	Senin - Jumat	Edit Hapus

Navigation: < Previous 1 2 3 4 Next >

Daftar Pasien

Klinik Jambu

Beranda

Daftar Dokter

Daftar Pasien

Janji Temu

Stok Obat

Pemeriksaan

Pengaturan

Daftar Pasien

Tambah Pasien

Nama

Jenis Kelamin

Tanggal Lahir

No Rekam Medis

Alamat

No. Handphone

Riwayat Penyakit

Tambah

Daftar Pasien

No.	Nama	Jenis Kelamin	Tanggal Lahir	No. Rekam Medis	Alamat	No. Hp	Riwayat Penyakit	Aksi
1.	Thania	Wanita	22-05-2004	88241233	Jl. Jalan	081234567890	maag	Edit Hapus
2.	Raka	Pria	22-05-2004	938741412	Jl. Itu	088724232112	DBD	Edit Hapus
3.	Raka	Pria	22-05-2004	938741412	Jl. Itu	088724232112	DBD	Edit Hapus
4.	Raka	Pria	22-05-2004	938741412	Jl. Itu	088724232112	DBD	Edit Hapus
5.	Raka	Pria	22-05-2004	938741412	Jl. Itu	088724232112	DBD	Edit Hapus

< Previous

1

2

3

4

Next >

Janji Temu

Klinik Jambu

Beranda

Daftar Dokter

Daftar Pasien

Janji Temu

Stok Obat

Pemeriksaan

Pengaturan

Janji Temu

Buat Janji Temu

Nama Pasien

Nama Dokter

Tanggal Janji Temu

Cari Nama Pasien...

Dropdown Nama Dokter

Keluhan

Buat Janji

Daftar Janji Temu

No.	Nama Pasien	Dokter	Tanggal	Keluhan	Status	Aksi
1.	Jane	Thania	22-02-2025	Terasa sakit di bagian....	Pending	Edit Hapus
2.	John	Shinta	22-02-2025	Terasa sakit di bagian....	Pending	Edit Hapus
3.	John	William	22-02-2025	Terasa sakit di bagian....	Pending	Edit Hapus
4.	Jane	Wilson	22-02-2025	Terasa sakit di bagian....	Pending	Edit Hapus
5.	John	Shinta	22-02-2025	Terasa sakit di bagian....	Pending	Edit Hapus

< Previous

1

2

3

4

Next >

## Hasil Periksa

**Klinik Jambu** Hasil Pemeriksaan

Search for something

**Daftar Pemeriksaan**

No.	Nama Pasien	Dokter	Tanggal Pemeriksaan	Diagnosa	Status Pembayaran	Aksi
1.	Jane	Thania	22-02-2025	Terasa sakit di bagian....	Pending	Edit Cetak Nota Hapus
2.	John	Shinta	22-02-2025	Terasa sakit di bagian....	Pending	Edit Cetak Nota Hapus
3.	John	William	22-02-2025	Terasa sakit di bagian....	Pending	Edit Cetak Nota Hapus
4.	Jane	Wilson	22-02-2025	Terasa sakit di bagian....	Pending	Edit Cetak Nota Hapus
5.	John	Shinta	22-02-2025	Terasa sakit di bagian....	Pending	Edit Cetak Nota Hapus

< Previous 1 2 3 4 Next >

3. Tulis Query berdasarkan LRS yang telah dibuat
- a. Tampilkan data pelanggan yang mempunyai kelamin perempuan, umur antara 19 sampai 30, dan pemeriksaan pada bulan Agustus 2015.

SELECT

pasien.pasien\_id,  
pasien.nama AS pasien\_nama,  
pasien.umur AS pasien\_umur,  
janji\_temu.tanggal\_pemeriksaan

FROM janji\_temu

INNER JOIN pasien ON janji\_temu.pasien\_id = pasien.pasien\_id

WHERE pasien.jenis\_kelamin = 'Perempuan'

AND pasien.umur BETWEEN 19 AND 30

AND janji\_temu.tanggal\_pemeriksaan BETWEEN '2015-08-01' AND '2015-08-31';

- b. Tampilkan data semua dokter yang mempunyai transaksi dengan pasien ataupun tidak selama setahun di 2015.

SELECT

dokter.id AS dokter\_id,  
dokter.nama AS dokter\_nama,  
dokter.spesialis,  
janji\_temu.tanggal\_pemeriksaan,  
pasien.nama AS pasien\_nama

FROM dokter

LEFT JOIN janji\_temu ON dokter\_id = janji\_temu.dokter\_id



LEFT JOIN pasien ON janji\_temu.pasien\_id = pasien\_id  
WHERE janji\_temu.tanggal\_pemeriksaan BETWEEN '2015-01-01' AND '2015-12-31';  
c. Hitung jumlah obat dan total uang per-obat selama bulan Agustus sampai Desember 2015.

```
SELECT
    obat.nama_obat,
    SUM(detail_obat.qty) AS total_jumlah_terjual,
    SUM(detail_obat.subtotal) AS total_pendapatan
FROM detail_obat
INNER JOIN obat ON detail_obat.obat_id = obat.id
INNER JOIN janji_temu ON detail_obat.janji_temu_id = janji_temu.id
WHERE janji_temu.tanggal_pemeriksaan BETWEEN '2015-08-01' AND '2015-12-31'
GROUP BY obat.nama_obat;
```

d. Tampilkan 10 Jenis obat apa saja yang paling banyak digunakan selama tahun 2015.

```
SELECT
    obat.nama_obat,
    SUM(detail_obat.qty) AS total_jumlah_terjual
FROM detail_obat
INNER JOIN obat ON detail_obat.obat_id = obat_id
INNER JOIN janji_temu ON detail_obat.janji_temu_id = janji_temu.id
WHERE janji_temu.tanggal_pemeriksaan BETWEEN '2015-01-01' AND '2015-12-31'
GROUP BY obat.nama_obat
ORDER BY total_jumlah_terjual DESC
```

e. Tampilkan data pelanggan Jika berumur dibawah 18 tahun “Anak-anak”, jika berumur 18 tahun sampai 30 “Dewasa”, dan jika lebih dari 30 “Orang tua”.

```
SELECT
    id AS pasien_id,
    nama AS pasien_nama,
    umur,
    CASE
        WHEN umur < 18 THEN 'Anak-anak'
        WHEN umur BETWEEN 18 AND 30 THEN 'Dewasa'
        ELSE 'Orang tua'
    END AS kategori_umur
FROM pasien;
```