KUNCI JAWABAN & RUBRIK PENILAIAN
UJIAN EVALUASI KOMPREHENSIF BASIS DATA

Bagian A: Pilihan Ganda Analitis (Bobot: 30% / 5 Poin per Soal)

1. **Jawaban: C** (Total Participation dan Disjoint)
   Penjelasan: Total Participation (garis ganda) karena setiap Sivitas **wajib** terdaftar sebagai Dosen atau Mahasiswa. Disjoint (simbol 'd' atau garis pisah) karena aturan bisnis melarang seseorang menjadi Dosen dan Mahasiswa sekaligus.

2. **Jawaban: B** (Karena Keterlibatan Proyek adalah entitas mandiri...)
   Penjelasan: Agregasi (P3) digunakan ketika sebuah relasi (Bekerja_Pada) memiliki relasi lain dengan entitas berbeda (Peralatan). Relasi Bekerja_Pada diperlakukan sebagai entitas utuh agar dapat dihubungkan dengan Peralatan.

3. **Jawaban: C** (Membuat tabel baru yang berisi PK dari A dan B)
   Penjelasan: Relasi N:M tidak bisa diselesaikan hanya dengan menaruh FK di salah satu tabel. Mutlak harus membuat tabel *Bridge* (penghubung) yang menampung PK kedua entitas sebagai *Composite Primary Key* (P4/P5).

4. **Jawaban: C** (`NUMERIC(9,2)`)
   Penjelasan: Tipe data uang/desimal (P6) tidak boleh menggunakan `FLOAT` atau `DOUBLE` karena rentan terhadap *floating-point error* (pembulatan). `NUMERIC` atau `DECIMAL` menjamin presisi angka secara eksak.

5. **Jawaban: B** (7)
   Penjelasan: Fungsi agregat `COUNT(Nama_Kolom)` akan **mengabaikan** baris yang bernilai `NULL`. Jika ada 10 baris dan 3 baris bernilai `NULL`, maka hasilnya $10 - 3 = 7$. (Berbeda dengan `COUNT(*)` yang akan menghasilkan 10).

6. **Jawaban: (Tidak ada yang tepat di opsi, namun secara logika menguji pemahaman _)**
   Penjelasan: Wildcard _ berarti **tepat satu karakter apapun**. Query `LIKE '_a%'` berarti mencari string dengan 2 huruf pertama adalah "karakter apapun" diikuti huruf 'a'.
   - Al Pacino (A-l) → Salah
   - Sean Penn (S-e) → Salah
   - Leonardo (L-e) → Salah
   - Tom Hanks (T-o) → Salah

   Rubrik: Berikan poin penuh (5) jika mahasiswa menganalisis bahwa semua opsi salah namun menjelaskan logika _. Berikan poin 2 jika mereka memilih acak tanpa alasan yang jelas.

Bagian B: Esai & Studi Kasus (Bobot: 70%)

7. EERD to Relational Schema (20 Poin)

Strategi Single Table Inheritance (Menggabungkan Semua Entitas): Karena menggunakan strategi Single Table, seluruh atribut dari Supertype dan Subtype digabungkan menjadi satu tabel besar. Atribut yang tidak dimiliki oleh suatu baris akan diisi nilai `NULL`.

Struktur Tabel:

- **Table:** `Sivitas`
- **Primary Key:** `ID_Sivitas`
- **Atribut Supertype:** `Nama`, `Email`
- **Atribut Dosen:** `NIDN`, `Gelar`
- **Atribut Mahasiswa:** `NIM`, `Tahun_Masuk`
- **Atribut Mahasiswa S1:** `SKS`, `IPK`
- **Atribut Mahasiswa S2:** `Judul_Tesis`, `Pembimbing`
- **Discriminator / Pembeda (Opsional tapi nilai plus):** `Tipe_Sivitas` (untuk membedakan apakah dia Dosen, Mhs S1, atau Mhs S2).

Rubrik:

- 5 Poin: Menyebutkan Nama Tabel dan PK dengan benar.
- 5 Poin: Menggabungkan atribut Supertype dan Subtype level 1 (Dosen/Mhs).
- 10 Poin: Menggabungkan seluruh atribut Subtype (termasuk S1 dan S2) ke dalam satu tabel.

8. DDL SQL (20 Poin)

```sql
CREATE TABLE atmwithdrawal (
    transaction_id INT PRIMARY KEY,
    account_number CHAR(3),
    amount NUMERIC(9,2) NOT NULL,
    withdraw_date DATE,
    FOREIGN KEY (account_number)
        REFERENCES account(number)
        ON DELETE CASCADE,
    CHECK (amount >= 0)
);
```

Rubrik:

- 5 Poin: Deklarasi tabel, kolom, dan tipe data tepat.
- 5 Poin: `PRIMARY KEY` pada `transaction_id` dan `NOT NULL` pada amount.
- 5 Poin: `FOREIGN KEY` dan `ON DELETE CASCADE` ditulis dengan benar.
- 5 Poin: Constraint `CHECK (amount >= 0)` ditulis dengan benar.

9. DML SQL (15 Poin)

a. Menampilkan Nama dan Saldo Nasabah Checking di atas 1000 (7.5 Poin)

```sql
SELECT customer.name, account.balance
FROM customer
JOIN account ON customer.id = account.cust_id
WHERE account.type = 'checking'
  AND account.balance > 1000;
```

Rubrik: 3 Poin `JOIN ON` yang benar, 2.5 Poin `WHERE` klausa, 2 Poin `SELECT` kolom.

b. Menampilkan Total Penarikan Bulan April 2008 (7.5 Poin)

```sql
SELECT SUM(amount)
FROM atmwithdrawal
WHERE withdraw_date BETWEEN '2008-04-01' AND '2008-04-30';
-- ATAU menggunakan LIKE:
-- WHERE withdraw_date LIKE '2008-04-%';
```

Rubrik: 3 Poin fungsi `SUM()`, 4.5 Poin filter tanggal yang tepat.

10. Analisis Query (15 Poin)

a. Apa hasil logis dari query tersebut? (5 Poin)

Query tersebut akan menampilkan daftar nama aktor (`MovieStar.name`) yang pernah membintangi suatu film, dengan syarat nama aktor tersebut ada di tabel `StarsIn`, dan judul filmnya ada di tabel `Movie`. (Menghubungkan 3 tabel menggunakan `WHERE` sebagai jembatan relasi).

b. Apa yang terjadi jika klausa WHERE dihapus? (10 Poin)

Jika klausa `WHERE` dihapus, hasilnya adalah **Cartesian Product** (Perkalian Kartesian / Cross Join). Sistem akan mengalikan setiap baris di tabel `MovieStar` dengan setiap baris di tabel `StarsIn`, lalu dikalikan lagi dengan setiap baris di tabel `Movie`. Hasilnya adalah data sampah (tidak berelasi) dalam jumlah yang sangat besar (Misal: 10 Star $\times$ 20 StarsIn $\times$ 15 Movie = 3.000 baris data acak).