

Bundle Solusi UTS Basis Data (IF2240)

Semester II 2024/2025

16 April 2025

IF '23

Razi Rachman Widyadhana - @zirachw

Apabila terdapat kesalahan pada file, silakan kontak penulis :)

Dilarang membawa Hardcopy saat ujian berlangsung

This page intentionally left blank

Daftar Isi

| | | |
|----------|----------------------|----------|
| 1 | UTS 2019/2020 | 4 |
| | Nomor 1 | 5 |
| | Nomor 2 | 6 |
| | Nomor 3 | 7 |
| | Nomor 4 | 9 |
| | Nomor 5 | 12 |

UTS 2019/2020

| | | |
|---------|-------|----|
| Nomor 1 | | 5 |
| Nomor 2 | | 6 |
| Nomor 3 | | 7 |
| Nomor 4 | | 9 |
| Nomor 5 | | 12 |

Nomor 1

Berikan jawaban yang jelas dan singkat.

- (a) Apa yang dimaksud dengan Data, Basis Data, Sistem Basis Data, dan Sistem Manajemen Basis Data? Pastikan hubungan antara keempat terminologi ini tergambar di dalam jawaban Anda.
- (b) Sebutkan dan jelaskan 3 (tiga) permasalahan yang ditemukan pada aplikasi basis data yang langsung memanfaatkan file system. Jelaskan bagaimana pendekatan sistem basis data mengatasi ketiga permasalahan tersebut
- (c) Jelaskan apa yang dimaksud dengan physical data independence dan logical data independence dalam kaitannya dengan abstraksi data (physical, logical, dan view level).

Jawab:

- (a) Definisi masing-masing terminologi:

- **Data:**

Data adalah representasi tersimpan dari objek dan peristiwa yang bermakna yang dikumpulkan dalam basis data (*database*). Data dapat berupa:

- Structured: angka, teks, tanggal
- Unstructured: gambar, video, dokumen

- **Basis Data:**

Basis Data adalah kumpulan data yang terorganisir secara logis dan saling berhubungan. Basis data menyimpan data dalam format terstandarisasi dan nyaman dengan dikelola oleh Sistem Manajemen Basis Data (DBMS).

- **Sistem Manajemen Basis Data:**

DBMS adalah sistem perangkat lunak yang digunakan untuk membuat, memelihara, dan menyediakan akses terkontrol ke database pengguna. DBMS mengelola sumber daya data seperti sistem operasi mengelola sumber daya perangkat keras.

- **Sistem Basis Data:**

Sistem Basis Data adalah lingkungan yang mencakup DBMS, database, perangkat keras, perangkat lunak, orang, dan prosedur yang bekerja bersama untuk mengelola data.

- (b) Tiga permasalahan aplikasi basis data dengan file system:

- **Program-Data Dependence:**

Setiap programmer aplikasi harus memelihara data sendiri, setiap program memerlukan kode untuk metadata setiap file, dan setiap program harus memiliki rutinitas pemrosesan sendiri, menyebabkan format file tidak standar.

Solusi:

DBMS menyediakan manajemen terpusat dan akses data dalam cara yang sama dengan tersedianya Standard Query Language (SQL)

- **Data Redundancy:**

Pemborosan ruang karena data yang sama disimpan berulang kali, menyebabkan masalah pemeliharaan, dan inkonsistensi data saat perubahan terjadi di satu file tetapi tidak di file lainnya.

Solusi:

DBMS menciptakan repositori terpusat data bersama, mengurangi duplikasi data dan meningkatkan kualitas data melalui penerapan aturan validasi.

- **Limited Data Sharing:**

Tidak ada kontrol terpusat atas data, menyebabkan kesulitan berbagi data antar aplikasi dan pengguna.

- **Solusi:**

DBMS memungkinkan berbagi data yang lebih baik, dengan pengguna yang berbeda mendapatkan tampilan yang berbeda dari data yang sama.

(c) Physical dan Logical Data Independence:

- **Physical Data Independence:**

Kemampuan untuk memodifikasi skema fisik tanpa mengubah skema logis. Aplikasi bergantung pada skema logis, sehingga perubahan pada struktur penyimpanan fisik tidak akan memengaruhi aplikasi yang mengakses data. Ini terkait dengan pemisahan antara level fisik (bagaimana data disimpan) dan level logis dalam abstraksi data.

- **Logical Data Independence:**

Kemampuan untuk mengubah skema logis tanpa memengaruhi skema tampilan (view level). Ini memungkinkan perubahan pada struktur logis database tanpa memengaruhi bagaimana pengguna akhir melihat dan berinteraksi dengan data. Ini terkait dengan pemisahan antara level logis dan level tampilan dalam abstraksi data.

Interface antara berbagai level dan komponen harus didefinisikan dengan baik sehingga perubahan di beberapa bagian tidak secara serius memengaruhi yang lain, menciptakan independensi data.

Nomor 2

Berikan jawaban yang jelas dan singkat.

- (a) Sebutkan dan jelaskan 3 (tiga) tipe model data yang dihasilkan selama proses perancangan hingga implementasi basis data.
- (b) Jelaskan karakteristik utama dari masing-masing jenis pemodelan data berikut.
 - i. Relational model
 - ii. Object-oriented model
 - iii. Semi-structured data model

Jawab:

- (a) Tiga tipe model data:

1. **Conceptual data model:**

Sekumpulan spesifikasi data yang bersifat independen terhadap teknologi dan digunakan untuk mendiskusikan kebutuhan awal dengan para pemangku kepentingan bisnis.

2. **Logical data model:**

Struktur data yang dapat diimplementasikan dalam database.

3. **Physical data model:**

Mengorganisasi data ke dalam tabel, serta mempertimbangkan detail akses, performa, dan penyimpanan.

(b) Karakteristik utama tiap jenis pemodelan data berikut:

1. Relational model:

Memodelkan sebagai kumpulan tabel yang terdiri dari tuple (baris) dan attributes (kolom) untuk merepresentasikan data dan hubungan antar data tersebut.

2. Object-oriented model:

Memodelkan dengan pendekatan sebagai kumpulan objek yang terdiri atas variable, message, dan method. Direpresentasikan secara diagramatis oleh Class diagram.

3. Semi-structured data model:

Mengorganisasi data ke dalam tabel, serta mempertimbangkan detail akses, performa, dan penyimpanan.

Nomor 3

Dalam basis data sebuah toko buku, setiap data penulis, penerbit, pelanggan, dan gudang, dapat dikenali secara unik melalui ID. Buku dikenali secara unik melalui ISBN. Seorang penulis bisa menulis satu atau lebih buku dan sebuah buku bisa ditulis oleh satu atau lebih penulis. Sebuah buku diterbitkan oleh hanya satu penerbit. Ketika seorang pelanggan membeli sebuah buku, sebuah keranjang belanja disediakan untuknya (setiap keranjang belanja diidentifikasi secara unik melalui ID). Untuk setiap keranjang belanja, disimpan data item-item (buku) yang dibelinya. Toko ini juga melakukan pencatatan terkait jumlah stok buku yang disimpan di gudang yang mereka miliki.

Diberikan skema basis data relasional dari suatu toko buku sbb.

| | |
|------------------|--|
| Penulis | = (ID, Nama, Alamat, Telepon) |
| Menulis | = (Penulis, Buku) |
| Buku | = (ISBN, Judul, Penerbit, Genre, Tahun, Harga) |
| Penerbit | = (ID, Nama, Alamat, Telepon, Website) |
| Pelanggan | = (ID, Email, Nama, Telepon, Alamat) |
| KeranjangBelanja | = (ID, Tanggal, Pelanggan) |
| ItemKeranjang | = (Keranjang, Buku, JumlahBuku) |
| Penyimpanan | = (Gudang, Buku, JumlahBuku) |
| Gudang | = (ID, Alamat, Telepon) |

(a) Gambarkan diagram skema relasional untuk skema basis data di atas yang memperlihatkan juga *primary key*, *foreign key reference*, dan keterhubungan antar relasi pada basis data berdasarkan deskripsi di atas.

(b) Jelaskan apa yang Anda ketahui mengenai empat jenis *integrity constraints* berikut.

- i. *domain constraint*
- ii. *entity integrity*
- iii. *action assertion*
- iv. *referential integrity*

Jawab:

(a) Berdasarkan deskripsi dan skema, saya dapat menentukan:

- *Primary keys:*

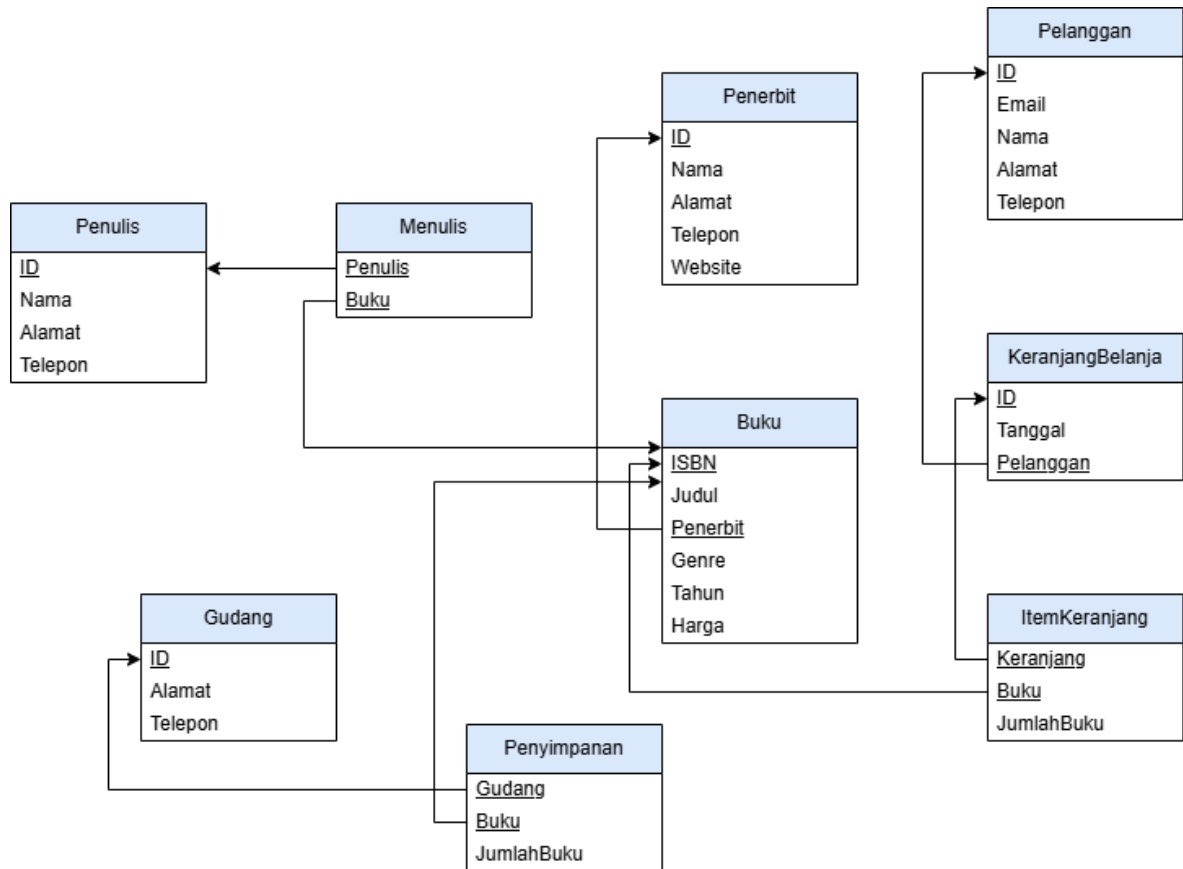
ID untuk Penulis, Penerbit, Pelanggan, Gudang, KeranjangBelanja; ISBN untuk Buku

- Menulis adalah relasi *many-to-many* antara Penulis dan Buku

- *Foreign keys:*

Penerbit dalam Buku; Pelanggan dalam KeranjangBelanja; Keranjang dalam ItemKeranjang; Buku dalam ItemKeranjang, Menulis, dan Penyimpanan; Gudang dalam Penyimpanan

Maka, diperoleh Diagram Skema Relasional:



(b) Empat jenis *integrity constraints*:

i. **Domain Constraint:**

Domain Constraint adalah aturan yang menyatakan bahwa semua nilai yang muncul dalam kolom/atribut sebuah relasi harus berasal dari domain yang sama. Definisi domain biasanya terdiri dari nama domain, arti/maksud, tipe data, ukuran (atau panjang), dan nilai-nilai yang diperbolehkan atau rentang yang diizinkan. Domain constraint memastikan bahwa nilai-nilai yang dimasukkan ke dalam database sesuai dengan karakteristik yang ditentukan untuk atribut tersebut.

ii. **Entity Integrity:**

Entity Integrity adalah aturan yang menyatakan bahwa tidak ada primary key attribute (atau komponen dari primary key attribute) yang boleh bernilai null. Aturan ini dirancang untuk memastikan bahwa setiap relasi memiliki primary key dan nilai data untuk primary key tersebut valid. Entity integrity menjamin bahwa setiap entitas dapat diidentifikasi secara unik.

iii. Action Assertion:

Action Assertion adalah jenis integrity constraint yang mendefinisikan kondisi atau aturan yang harus dipenuhi ketika terjadi perubahan data (seperti insert, update, atau delete). Action assertion memastikan bahwa perubahan data tidak menyebabkan pelanggaran terhadap aturan bisnis atau logika aplikasi yang telah ditetapkan. Constraint ini dapat diterapkan sebagai trigger atau prosedur yang dijalankan saat terjadi modifikasi data untuk memvalidasi bahwa integritas data tetap terjaga.

iv. Referential Integrity:

Referential Integrity adalah aturan yang menyatakan bahwa nilai foreign key HARUS cocok dengan nilai primary key di relasi yang direferensikan (atau foreign key dapat bernilai null). Aturan ini mempertahankan konsistensi antar baris dari dua relasi. Sebagai contoh, jika ada aturan penghapusan (delete rules), dapat mengikuti beberapa metode: Restrict (tidak mengizinkan penghapusan "parent" jika masih ada baris terkait di "dependent"), Cascade (secara otomatis menghapus baris "dependent" yang berhubungan dengan "parent" yang dihapus), atau Set-to-Null (mengatur foreign key di dependent menjadi null jika menghapus dari parent).

Nomor 4

Diberikan skema basis data relasional dari suatu toko buku sbb.

| | |
|------------|---|
| Karyawan | = (<u>IDKaryawan</u> , Nama, Gaji, TanggalBergabung, Departemen) |
| Departemen | = (<u>NamaDepartemen</u> , NoRuangan, Anggaran) |
| Bonus | = (<u>IDKaryawan</u> , <u>TanggalBonus</u> , NilaiBonus) |
| Jabatan | = (<u>IDKaryawan</u> , <u>PosisiJabatan</u> , MulaiMenjabat) |

Keterangan:

- Atribut yang digarisbawahi merupakan *primary key* dari relasi.
- Relasi Jabatan mencatat posisi jabatan setiap karyawan selama mereka bekerja di perusahaan tersebut dan kapan mulai menjabat.
- Bonus(IDKaryawan), Jabatan(IDKaryawan) dan Karyawan(Departemen) merupakan *foreign key* secara berturut turut *reference* ke Karyawan(IDKaryawan), Karyawan(IDKaryawan) dan Departemen>NamaDepartemen).

- Tuliskan ekspresi SQL untuk membuat relasi Karyawan.
- Tuliskan 3 (tiga) ekspresi SQL yang berbeda yang digunakan untuk menghasilkan tuple dari relasi Bonus yang nilai dari atribut IDKaryawan-nya tidak ada di atribut IDKaryawan dari relasi Karyawan.

Untuk soal 4(c)-4(e), tuliskan ekspresi SQL untuk melakukan query berikut.

- Menampilkan daftar nama departemen yang rata-rata gaji karyawannya lebih dari 5.000.000.
- Menampilkan daftar pasangan karyawan yang memiliki gaji yang sama. Setiap pasangan hanya boleh muncul 1 kali.

Contoh:

Karyawan

| IDKaryawan | Nama | Gaji | TanggalBergabung | Departemen |
|------------|-------|-----------|------------------|------------|
| 1 | Andi | 5.000.000 | 2018-01-08 | Finance |
| 2 | Anton | 6.000.000 | 2018-02-08 | Marketing |
| 3 | David | 5.000.000 | 2018-01-04 | Finance |
| 4 | Lia | 5.000.000 | 2018-01-09 | Sales |

Query akan menghasilkan:

| IDKaryawan1 | Nama1 | IDKaryawan2 | Nama2 | Gaji |
|-------------|-------|-------------|-------|-----------|
| 1 | Andi | 3 | David | 5.000.000 |
| 1 | Andi | 4 | Lia | 5.000.000 |
| 3 | David | 4 | Lia | 5.000.000 |

- (e) Menambahkan tuple ke dalam relasi Bonus dengan nilai NilaiBonus adalah 1,2 kali gaji untuk karyawan dengan gaji $\leq 7.000.000$ dan 1,3 kali gaji untuk karyawan dengan gaji $> 7.000.000$, dengan TanggalBonus adalah '2020-03-05'.

Jawab:

- (a) Query membuat relasi Karyawan

```
CREATE TABLE Karyawan = (
    IDKaryawan : CHAR(5),
    Nama : VARCHAR(50),
    Gaji : NUMERIC(10,2),
    TanggalBergabung : DATE,
    Departemen : VARCHAR(20),
    PRIMARY KEY (IDKaryawan),
    FOREIGN KEY (Departemen) REFERENCES Departemen(NamaDepartemen)
);
```

- (b) 3 Ekspresi SQL yang berbeda:

```
SELECT *
FROM Bonus
WHERE IDKaryawan
NOT IN (SELECT IDKaryawan
        FROM Karyawan);
```

```
SELECT *
FROM Bonus B
WHERE NOT EXISTS (SELECT *
                  FROM Karyawan K
                  WHERE B.IDKaryawan = K.IDKaryawan);
```

```
SELECT B.*
FROM Bonus B
LEFT OUTER JOIN Karyawan K
USING (IDKaryawan)
WHERE K.IDKaryawan IS NULL;
```

- (c) Daftar nama departemen yang rata-rata gaji karyawannya lebih dari 5.000.000.

```
SELECT K.Departemen
FROM Karyawan K
GROUP BY K.Departemen
HAVING AVG(Gaji) > 5000000;
```

- (d) Daftar pasangan karyawan yang memiliki gaji yang sama. Setiap pasangan hanya boleh muncul 1 kali

```
SELECT K1.Nama, K2.Nama
FROM Karyawan K1, Karyawan K2
WHERE K1.Gaji = K2.Gaji
AND K1.IDKaryawan < K2.IDKaryawan;
```

- (e) Menambahkan tuple ke dalam relasi Bonus

```
INSERT INTO Bonus
SELECT IDKaryawan, '2020-03-05', Gaji * 1.2
FROM Karyawan
WHERE Gaji <= 7000000;

INSERT INTO Bonus
SELECT IDKaryawan, '2020-03-05', Gaji * 1.3
FROM Karyawan
WHERE Gaji > 7000000;
```

atau bisa juga

```
INSERT INTO Bonus
SELECT IDKaryawan, '2020-03-05',
CASE
    WHEN Gaji <= 7000000 THEN Gaji * 1.2
    ELSE Gaji * 1.3
END
FROM Karyawan;
```

Nomor 5

Buatlah diagram entity-relationship (diagram ER) untuk memodelkan persoalan berikut. Gunakan nama-nama yang jelas sesuai yang tertera dalam deskripsi. Jika ada aspek pemodelan yang tidak dapat ditangkap dengan diagram ER, tambahkan sebagai keterangan. Buat asumsi sendiri untuk hal-hal yang tidak disebutkan secara jelas dalam deskripsi di atas dan tuliskan asumsi-asumsi tersebut.

Saat ini beberapa mahasiswa Informatika ITB tengah sibuk membuat Game Pokemon. Untuk membuat game ini dibutuhkan sebuah basis data yang menyimpan data permainan. Setiap pemain berperan sebagai trainer. Setiap trainer memiliki nama (unik), jenis kelamin, dan experience point (exp). Trainer memelihara minimal satu jenis Pokemon. Setiap jenis Pokemon dikenali melalui IDPokemon yang unik, dan memiliki atribut lain yaitu nama, tipe (contoh: tipe air, tipe api, satu Pokemon bisa memiliki satu tipe atau lebih), dan rarity dari Pokemon. Terdapat jenis Pokemon yang sangat langka sehingga mungkin tidak dimiliki trainer manapun. Beberapa jenis Pokemon memiliki satu kekuatan atau lebih, setiap kekuatan diberi nomor urut yang unik per jenis Pokemon. Setiap kekuatan memiliki nama, tipe, dan besar serangan. Trainer bersama dengan Pokemonnya dapat bertarung dengan trainer dan Pokemon lain. Pada setiap pertarungan dicatat waktu dan tempat pertarungan berlangsung.

Trainer dapat memiliki sejumlah barang. Barang dibedakan melalui IDBarang yang unik, nama barang, tipe, dan deskripsi kegunaan barang. Terdapat bermacam-macam tipe barang, antara lain batu kekuatan dan pokeball. Batu kekuatan adalah barang yang digunakan pada saat bertarung. Untuk setiap batu kekuatan perlu disimpan informasi terkait durasi waktu sebelum efeknya hilang. Pokeball merupakan barang yang digunakan untuk menangkap Pokemon. Terdapat beberapa level pokeball yang membedakan kemampuan pokeball untuk menangkap Pokemon.

Jawab:

Berdasarkan deskripsi soal, digunakan beberapa asumsi berikut untuk memperjelas spesifikasi:

Asumsi:

1. Setiap Pokemon hanya dapat dimiliki oleh satu **Trainer** pada satu waktu
2. Setiap kekuatan memiliki ID_Kekuatan unik, meskipun namanya mungkin sama
3. Dalam pertarungan, kedua **Trainer** menggunakan Pokemon mereka
4. Item dapat dimiliki oleh sejumlah **Trainer**
5. ID dari **Power** memiliki nomor urut yang unik per jenis **Pokemon**, maka ID dari Pokemon merupakan **diskriminan** dari *Weak Entity Power*
6. Terdapat jenis **Pokemon** yang sangat langka yang mungkin tidak dimiliki oleh **Trainer** manapun

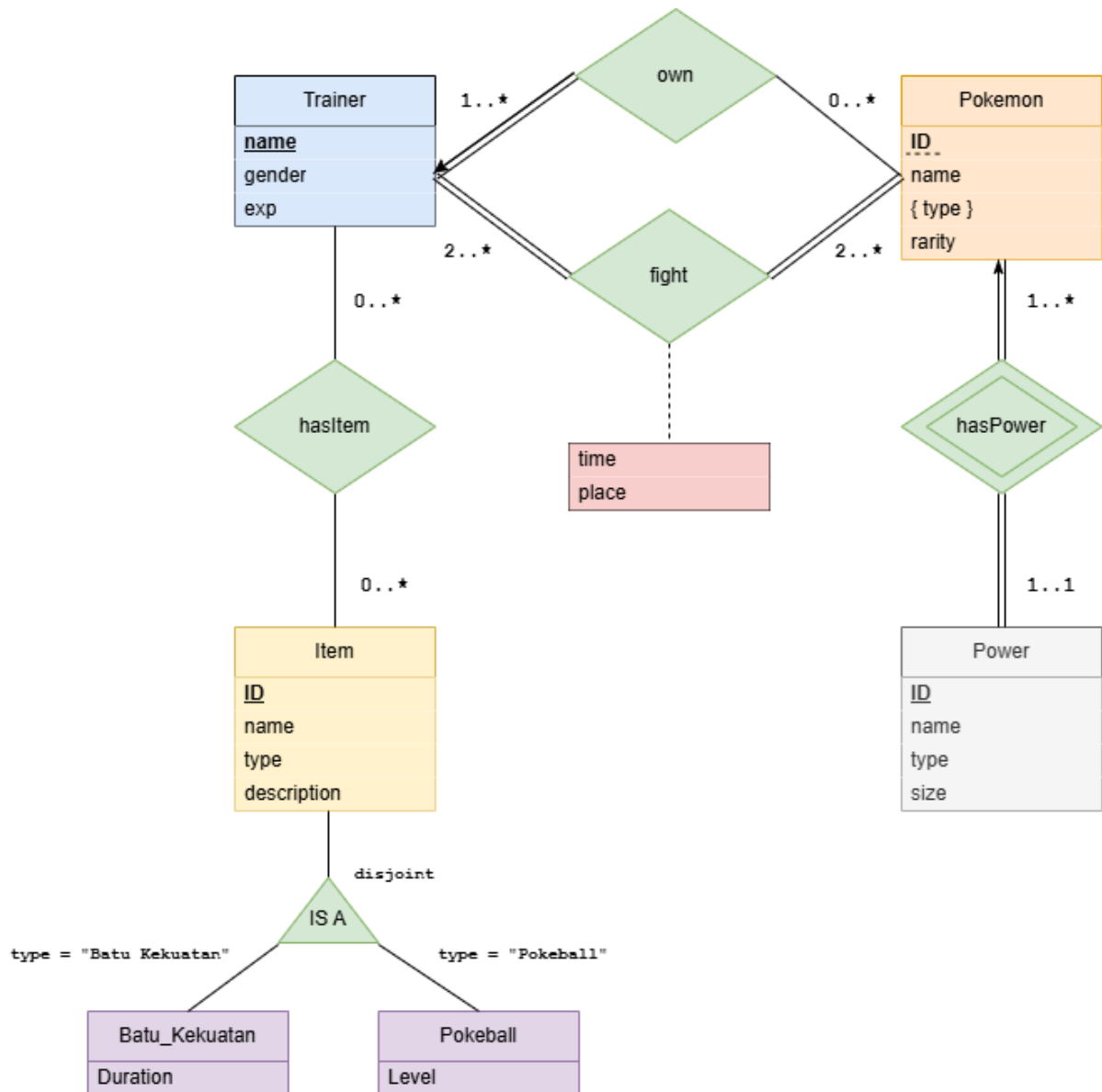
Maka, dapat diidentifikasi beberapa entitas, relasi, dan atribut berikut:

Entity:

Trainer = (name, gender, exp)
 Pokemon = (ID, name, {type}, rarity)
 Kekuatan = (ID, name, type besar_serangan)
 Barang = (ID, name, type, description)
 Pokeball = (level)
 Batu_Kekuatan = (durasi_efek)

Relationship:

own (relationship antara Trainer dengan Pokemon) - One-to-many
 Fight (relationship antara Trainer dengan Pokemon) - Many-to-Many
 hasPower (relationship antara Pokemon dengan Kekuatan) - One-To-Many
 hasItem (relationship antara Trainer dengan Barang) - Many-to-many

E-R Diagram:

To be Continued...