**RANCANGAN BASIS DATA APLIAKSI “HARMONY LIFE” UNTUK MENDUKUNG KESEIMBANGAN KEHIDUPAN KESEHARIAN**

Disusun oleh:

Kelompok 10

Adam Herlambang (2023071007)

Gardavi Nabhan Gussasy (2023071010)

**Abstrak**

*Aplikasi "Harmony Life" dirancang untuk memberikan solusi teknologi bagi individu dengan kesibukan tinggi seperti Ibu Maya, seorang ibu bekerja yang berusaha menyeimbangkan karier, kesehatan, dan keluarga. Penelitian ini bertujuan untuk merancang basis data yang mendukung fungsi aplikasi, termasuk pengelolaan data aktivitas, preferensi kesehatan, dan aktivitas keluarga. Proses perancangan mencakup analisis kebutuhan pengguna, pembuatan tabel dengan atribut yang relevan, pengaturan relasi antar tabel, serta implementasi query untuk Create, Read, Update, dan Delete (CRUD). Hasil penelitian menunjukkan bahwa basis data ini mampu menghasilkan data yang konsisten dan relevan serta mendukung berbagai fitur aplikasi secara optimal.*

# Pendahuluan

Teknologi informasi telah menjadi bagian yang tak terpisahkan dari kehidupan sehari-hari, khususnya bagi individu dengan jadwal yang padat seperti Ibu Maya, seorang direktur pemasaran di startup teknologi. Selain menjalankan perannya sebagai seorang profesional, Ibu Maya juga menghadapi tantangan dalam mengelola keseimbangan antara tanggung jawab karier, kesehatan pribadi, dan waktu berkualitas bersama keluarganya yang terdiri dari dua anak kecil.

Dalam upaya menjawab tantangan ini, dirancang sebuah aplikasi bernama "Harmony Life" yang bertujuan untuk membantu pengguna mengatur aktivitas harian, menetapkan tujuan kesehatan, dan merencanakan aktivitas keluarga. Basis data berperan penting dalam mendukung pengelolaan informasi dalam aplikasi ini. Penelitian ini difokuskan pada perancangan

# Bahan dan Metode

**1. Identifikasi Kebutuhan Pengguna**

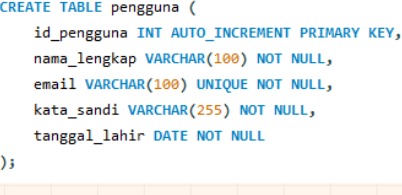
Dalam proses perancangan basis data, langkah pertama adalah mengidentifikasi kebutuhan pengguna berdasarkan tujuan dan fitur aplikasi. Berikut adalah kebutuhan utama yang diidentifikasi:

* **Data Pribadi Pengguna:** Informasi yang dibutuhkan meliputi nama lengkap, email, kata sandi, dan tanggal lahir. Data ini diperlukan untuk autentikasi dan personalisasi pengalaman pengguna.
* **Data Aktivitas Harian:** Informasi yang disimpan meliputi nama aktivitas, deskripsi, prioritas (Rendah, Sedang, Tinggi), dan tanggal pelaksanaan. Data ini membantu pengguna mengatur jadwal dan prioritas aktivitas.
* **Preferensi Kesehatan:** Informasi yang diperlukan mencakup tujuan olahraga dan kebutuhan nutrisi. Preferensi ini9 digunakan untuk memberikan rekomendasi kesehatan yang sesuai.
* **Aktivitas Keluarga:** Informasi meliputi jenis aktivitas keluarga dan frekuensi pelaksanaannya (Harian, Mingguan, Bulanan). Data ini membantu pengguna merencanakan waktu berkualitas bersama keluarga.

**2. Pembuatan Tabel**

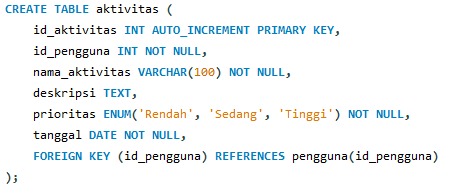
Tabel-tabel dirancang berdasarkan kebutuhan data yang telah diidentifikasi. Setiap tabel memiliki atribut yang relevan, dan relasi antar tabel ditentukan menggunakan Foreign Key untuk menjaga integritas data. Berikut adalah desain tabel utama:

**Tabel pengguna:** Menyimpan data pribadi pengguna.

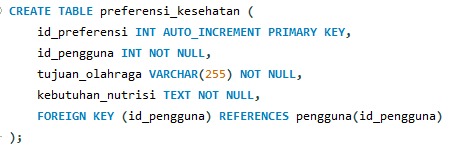




**Tabel aktivitas:** Menyimpan data aktivitas harian pengguna.

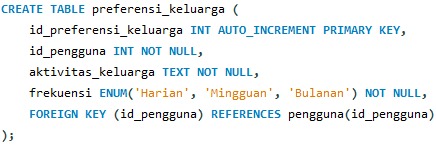


**Tabel preferensi\_kesehatan:** Menyimpan data preferensi kesehatan pengguna.

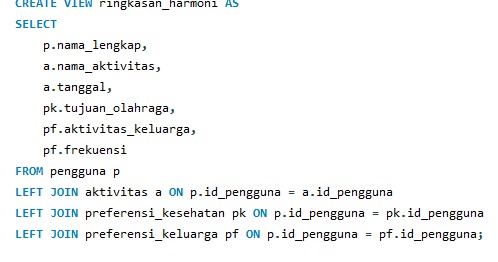




**Tabel preferensi\_keluarga:** Menyimpan data aktivitas keluarga pengguna

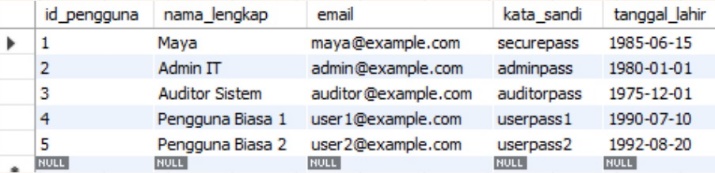


**Tabel view:** melihat semua informasi penting tanpa harus memahami struktur atau query dari tabel-tabel terkait.



**Insert Tabel Pengguna**





**Read data dari Tabel pengguna**

SELECT \* FROM pengguna;

**Update data di Tabel pengguna**

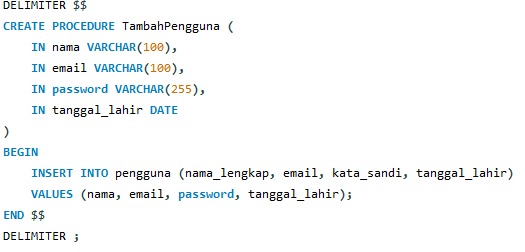
**UPDATE pengguna**

SET nama\_lengkap = 'Jane Doe'

WHERE id\_pengguna = 1;

**Store Procedure**

Store procedure membantu mengelola operasi basis data secara otomatis. Contoh untuk menambah pengguna:



**Normalisasi**

Normalisasi dilakukan hingga 3NF:

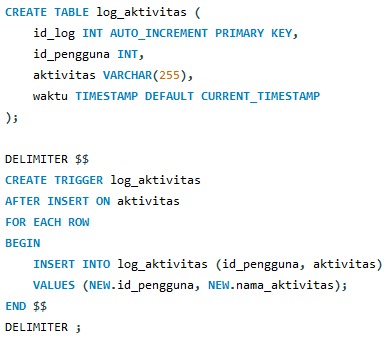
1. **1NF:** Data diorganisasikan ke dalam tabel dengan kolom yang atomik.
2. **2NF:** Semua atribut non-primer sepenuhnya bergantung pada primary key.
3. **3NF:** Tidak ada atribut yang bergantung transitif pada primary key.

Contoh:

* Tabel pengguna dan aktivitas terpisah memastikan setiap tabel hanya menyimpan informasi yang relevan.
* Atribut id\_pengguna menjadi penghubung antar tabel.

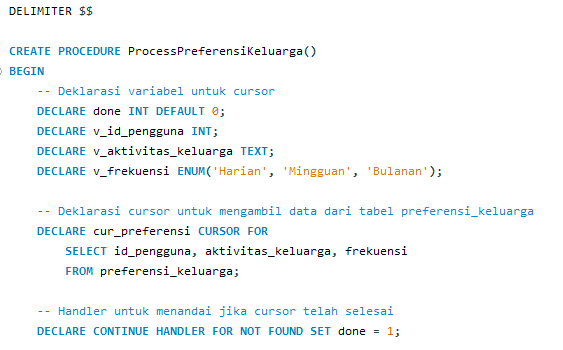
**Trigger**

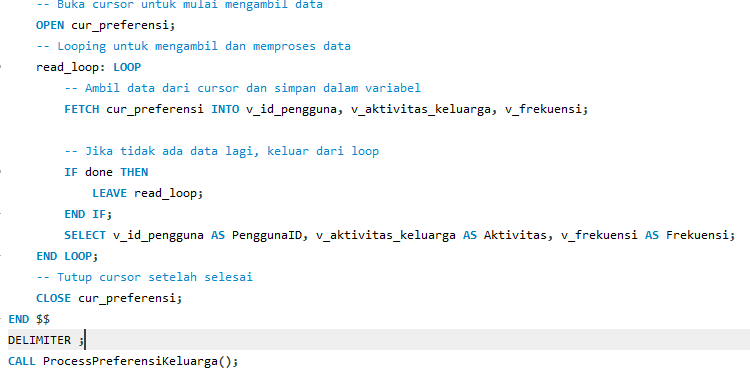
Trigger digunakan untuk tindakan otomatis. Contoh, untuk log setiap kali data pengguna diubah:



**Manage Cursor**

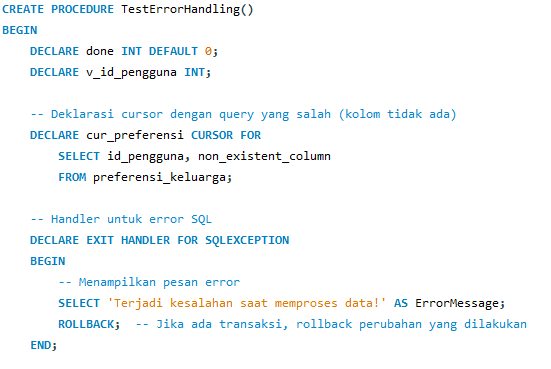
Digunakan untuk memproses data secara baris demi baris dalam sistem basis data. Proses ini dimulai dengan mendeklarasikan cursor untuk mengambil data dari kolom-kolom tertentu, seperti id\_pengguna,aktivitas\_keluarga dan frekuensi Setelah itu, cursor dibuka menggunakan perintah OPEN, yang memungkinkan pengambilan data secara bertahap dengan perintah FETCH. Data yang diambil disimpan dalam variabel untuk diproses lebih lanjut. Proses pengolahan berjalan dalam sebuah loop hingga semua data selesai diproses, yang ditandai dengan handler NOT FOUND untuk menghentikan loop saat tidak ada lagi data yang tersisa. Setelah semua data selesai diproses, cursor ditutup menggunakan perintah CLOSE untuk membebaskan sumber daya yang digunakan. Pendekatan ini sangat efektif untuk menangani pengolahan data yang membutuhkan tingkat fleksibilitas dan kontrol tinggi pada setiap baris data.

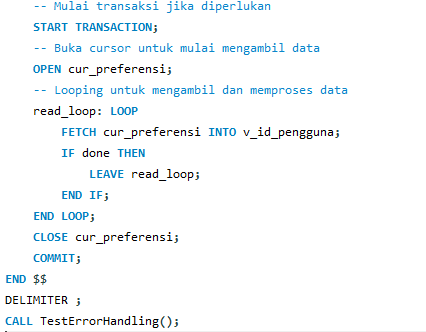




**Error Handling**

Berfungsi untuk memastikan konsistensi data dan menangani kesalahan dengan efisien dalam sistem basis data. Prosedur ini dirancang untuk mendeteksi dan merespons kesalahan yang terjadi selama eksekusi query. Jika sistem mencoba menjalankan query yang salah, seperti mengakses kolom yang tidak ada dalam tabel, error handler akan secara otomatis menangani situasi tersebut. Proses ini melibatkan pengaktifan handler khusus yang akan menampilkan pesan error dan menjalankan perintah ROLLBACK untuk membatalkan transaksi yang sedang berlangsung. Dengan demikian, perubahan data yang tidak valid tidak akan tersimpan dalam basis data. Selain itu, dalam skenario di mana error terjadi dalam proses loop menggunakan cursor, handler memastikan bahwa loop dihentikan untuk mencegah propagasi kesalahan lebih lanjut. Pendekatan ini memberikan perlindungan tambahan terhadap inkonsistensi data sekaligus menjaga stabilitas sistem saat terjadi kesalahan selama proses eksekusi.





**Security**

Langkah-langkah keamanan:

**1. Hak Akses Terbatas:** Hak akses sesuai peran:

GRANT SELECT, INSERT ON pengguna TO 'user\_app'@'localhost';

GRANT ALL ON pengguna TO 'admin\_app'@'localhost';

**2. Enkripsi Kata Sandi:** Kata sandi dienkripsi menggunakan algoritma hashing seperti bcrypt di aplikasi.

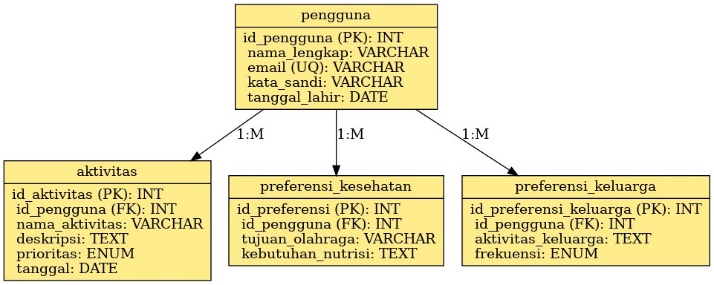
**3. Validasi Input:** Mencegah SQL injection dengan prepared statements.

**4. SSL Connection:** Pastikan koneksi basis data menggunakan SSL untuk keamanan data yang ditransfer.

**Ilustrasi Relasi Antar Tabel**

Berikut adalah diagram relasi entitas (ERD) yang menunjukkan hubungan antar tabel dalam basis data aplikasi "Harmony Life". Gambar ini memberikan gambaran visual mengenai struktur data dan relasi utama, yaitu antara tabel pengguna, aktivitas, preferensi kesehatan, dan preferensi keluarga. Desain ini dirancang untuk mendukung pengelolaan informasi yang efisien dan konsisten sesuai kebutuhan aplikasi.

*Gambar: Diagram Relasi Entitas (ERD)*



# 3. Query CRUD dan View

**Query CRUD:** Query CRUD (Create, Read, Update, Delete) digunakan untuk mengelola data dalam tabel. Contoh query CRUD adalah sebagai berikut:

-- Create data pengguna

INSERT INTO pengguna (nama\_lengkap, email, kata\_sandi, tanggal\_lahir)

VALUES ('Maya', 'maya@example.com', 'securepassword', '1986-07-12');

-- Read data pengguna

SELECT \* FROM pengguna;

-- Update data pengguna

UPDATE pengguna

SET nama\_lengkap = 'Maya Santoso'

WHERE id\_pengguna = 1;

-- Delete data pengguna

DELETE FROM pengguna WHERE id\_pengguna = 1;



**View:** View dibuat untuk menyajikan data secara ringkas kepada pengguna. Contoh view yang dirancang adalah sebagai berikut:

CREATE VIEW ringkasan\_harmoni AS

SELECT

p.nama\_lengkap,

a.nama\_aktivitas,

pk.tujuan\_olahraga,

pf.aktivitas\_keluarga

FROM pengguna p

LEFT JOIN aktivitas a ON p.id\_pengguna = a.id\_pengguna

LEFT JOIN preferensi\_kesehatan pk ON p.id\_pengguna = pk.id\_pengguna

LEFT JOIN preferensi\_keluarga pf ON p.id\_pengguna = pf.id\_pengguna;

# Hasil Pembahasan

**Hasil Pembuatan Tabel**

Tabel-tabel yang dirancang berhasil memenuhi kebutuhan data aplikasi "Harmony Life." Tabel pengguna menyimpan data pribadi pengguna, tabel aktivitas mencatat aktivitas harian dengan prioritas tertentu, tabel preferensi\_kesehatan menyimpan tujuan olahraga dan kebutuhan nutrisi, dan tabel preferensi\_keluarga mencatat aktivitas keluarga.

**Pembahasan Relasi Antar Tabel**

Setiap tabel dihubungkan melalui atribut id\_pengguna sebagai Foreign Key. Hal ini memastikan bahwa data yang disimpan di tabel aktivitas, preferensi\_kesehatan, dan preferensi\_keluarga terkait langsung dengan data pengguna yang bersangkutan.

**Efisiensi dan Keamanan Data**

* **Efisiensi:** Penggunaan indeks pada kolom email di tabel pengguna meningkatkan kecepatan pencarian data.
* **Keamanan:** Hak akses diberikan berdasarkan peran pengguna, seperti admin, manajer, dan auditor, untuk memastikan data sensitif tetap terlindungi.

**View Data**

View ringkasan\_harmoni menyediakan ringkasan data yang relevan, mencakup nama pengguna, aktivitas harian, tujuan kesehatan, dan aktivitas keluarga. Hal ini mempermudah pengguna dalam memantau keseimbangan hidup mereka.

**Kesimpulan**

Perancangan basis data aplikasi "Harmony Life" telah berhasil dilakukan dengan pendekatan relasional. Basis data ini mampu mendukung pengelolaan data pengguna secara efisien, konsisten, dan relevan. Dengan adanya view, pengguna dapat melihat ringkasan informasi secara ringkas dan mudah dipahami.

**REFERENSI**

[1] Connolly, T., & Begg, C. (2015). *Database Systems: A Practical Approach to Design, Implementation, and Management*. Pearson Education.

[2] Silberschatz, A., Korth, H. F., & Sudarshan, S. (2019). *Database System Concepts*. McGraw-Hill Education.

[3] Elmasri, R., & Navathe, S. B. (2016). *Fundamentals of Database Systems*. Pearson Education.