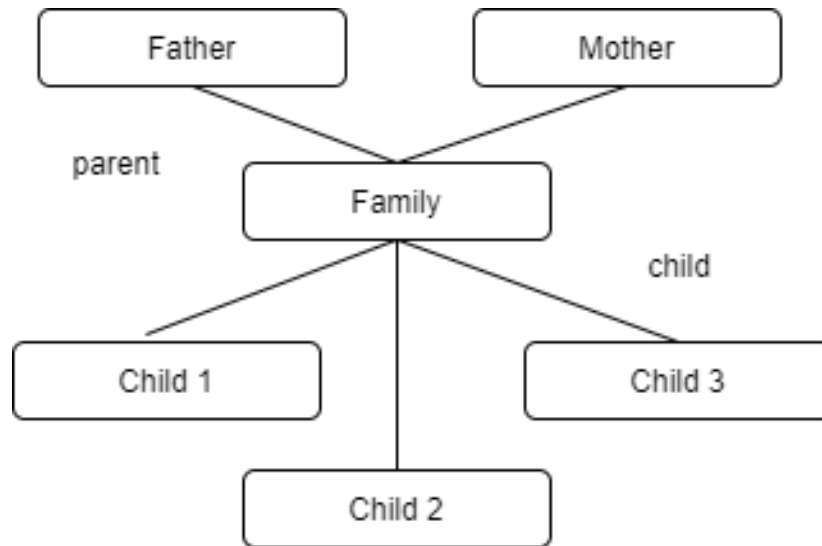


JAWABAN NOMOR 3 DAN 4

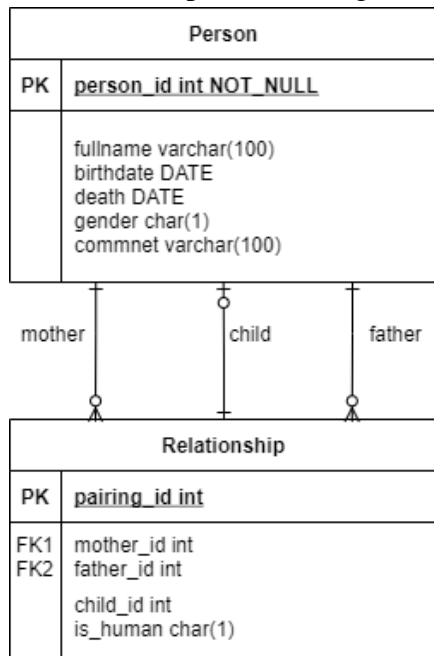
3. Family Tree Data Structure

3.1 gambaran struktur data dari pohon keluarga



Penjelasan: struktur dalam graph ini dapat memuat informasi tentang hubungan dan status dari pasangan orangtua dan hubungan mereka dengan anak-anak mereka, entah itu manusia atau hewan, bila semua anaknya manusia, ada yang diadopsi atau *blood-related*. Satu orang didalam keluarga itu selalu terkoneksi dengan setidaknya satu keluarga sebagai anak, berapa banyak keluarga sebagai orang tua, bisa jadi seorang itu bisa saja memiliki suami/istri yang diakui lebih dari satu, entah masih berhubungan atau statusnya cerai hidup atau mati. dengan graph ini, semua lintas keluarga bisa memungkinkan dan strukturnya disederhanakan, karena setiap orang hanya terkaitkan ke n-keluarga -- entah sebagai orangtua atau anak -- dan setiap keluarga hanya memegang 2 orangtua (dari setiap gender) dan daftar dari anak-anaknya. Keadana cerai hidup atau mati bisa diatasi dan pasangan LGBT juga bisa diakomodasikan

3.2 ERD dari pohon keluarga



Penjelasan: gambar ERD ini mungkin kurang jelas, tapi pada ERD ini memiliki 2 tabel, yang satu tabel person dan satu lagi hubungan antar person. Setip garis relasi pada graph ini menggambarkan:

- Setiap ibu bisa menjadi ibu dari 0-ke-banyak anak (person_id).
- Setiap bapak bisa menjadi bapak dari 0-ke-banyak anak (person_id).
- Setiap anak hanya bisa punya 0-ke-satu bapak dan ibu, kandung atau adopsi, jika adopsi, maka mother_id dan/atau father_id lama akan diganti menjadi mother_id dan/atau father_id yang menagdopsi.

3.3 Pseudo-code untuk mengambil data dari pohon keluarga ini

Kelas Person{

Variable nama, relasi

Construct(nama){

Nama ini adalah nama

Relasi ini adalah berbentuk array

}

Fungsi untuk menambah relasi(type_relasinya, si_orang){

Jika disini tidak ada array relasi yang memilik tipe relasi tertentu{

Masukan tipe relasi kedalam array relasi

}

Masukan seseorang kedalam array tipe relasi didalam relasi

}

Fungsi untuk mendapatkan beberapa relasi(type_relasinya){

Jika disini tidak ada array relasi yang memilik tipe relasi tertentu{

Kembalikan array kosong

```

    }
    Kembalikan array relasi yang memilik tipe relasi di fungsi ini
  }
  Fungsi untuk mendapatkan satu relasi(type_relasinya){
    Variable relasi berasal dari array relasi yang memilik tipe relasi di fungsi ini
    Kembalikan array relasi yang kosong jika isi datanya null, atau isi array relasi dalam urutan ke
0
  }
  Fungsi untuk menambah orangtua(nama ayah dan ibu){
    Si ayah adalah ayah dari si anak yang sudah dtambahkan ke fungsi tambhakan anak
    Si ibu adalah ibu dari si anak yang sudah dtambahkan ke fungsi tambhakan anak
  }
  Fungsi menambahkan anak(nama si anak){
    Si anak ini akan dimasukan ke tambah relasi dengan tipe relasis ebagai anak dengan nama anak
itu sendiri
    Dan si anak ini akan diinklud juga relasi ortunya.
  }
  Fungsi menambahkan pasutri(nama si anak){
    Si pasangan ini akan dimasukan ke tambah relasi dengan tipe relasis ebagai pasangan dengan
nama pasangan itu sendiri
    Dan si pasangan ini akan diinklud juga relasi pasangannya.
  }

fungsi dapatkan orangtua { kembalikan nilai fungsi metode dapatkan hubungan sebagai
orangtua }
fungsi dapatkan anak { kembalikan nilai fungsi metode dapatkan hubungan sebagai anak }
fungsi dapatkan pasutri { kembalikan nilai fungsi metode dapatkan hubungan sebagai pasutri }
}

```

4. Pengetahuan Dasar Flutter

4.1 Jawaban: Response

4.2 Jawaban: Future getName() async { }

4.3 Jawaban:

```

RaisedButton(
  Child: Text('Contact us'),
  onPressed: () => print('Contact us')
),

```