

Nama : Dzakwan Irfan Ramdhani
NIM : H1D022043
Shift Awal : C
Shift Baru : D

PENJELASAN SOURCE CODE

1. Import library

`import 'dart:io';` Mengimpor library `dart:io`, yang menyediakan fungsi untuk berinteraksi dengan input dan output melalui command prompt, seperti membaca input dari pengguna (`stdin`) dan menampilkan output (`stdout`).

```
import 'dart:io';
```

2. Kelas Kontak

- `class Kontak`: Mendefinisikan kelas `Kontak` yang digunakan untuk menyimpan informasi kontak.
- `String nama`; dan `String nomorTelepon`;: Dua atribut dalam kelas ini yang menyimpan nama dan nomor telepon dari kontak.
- `Kontak(this.nama, this.nomorTelepon);`: Konstruktor dari kelas `Kontak`, yang menerima nama dan nomor telepon sebagai parameter saat objek `Kontak` dibuat. Dengan menggunakan kata kunci `this`, parameter diinisialisasi langsung ke atribut kelas.

```
class Kontak {  
    String nama;  
    String nomorTelepon;  
  
    Kontak(this.nama, this.nomorTelepon);  
}
```

3. Fungsi main

- `void main()`: Fungsi utama tempat eksekusi program dimulai.
- `List<Kontak> daftarKontak = [];`: Membuat sebuah list kosong `daftarKontak` untuk menyimpan objek `Kontak`.
- `while (true)`: Memulai loop tanpa batas (infinite loop) yang akan terus berjalan sampai pengguna memilih untuk keluar dari program.
- `print()`: Digunakan untuk menampilkan teks ke layar. Di sini, teks yang menampilkan menu utama dan pilihan opsi kepada pengguna.
- `stdout.write('Pilih opsi (1-4): ');`: Menampilkan teks ke layar tanpa pindah ke baris baru, sehingga input pengguna akan berada di baris yang sama.

- `String? pilihan = stdin.readLineSync();`: Membaca input dari pengguna sebagai string. `String?` menunjukkan bahwa pilihan bisa bernilai null jika tidak ada input.

```
void main() {
    List<Kontak> daftarKontak = [];

    while (true) {
        print('\n==== Daftar Kontak ====');
        print('1. Tampilkan Semua Kontak');
        print('2. Tambah Kontak Baru');
        print('3. Hapus Kontak');
        print('4. Keluar');
        stdout.write('Pilih opsi (1-4): ');
        String? pilihan = stdin.readLineSync();
```

4. Kondisi Pilihan

- `if (pilihan == '1') {...}`: Mengecek apakah pengguna memilih opsi 1. Jika iya, maka fungsi `tampilkanKontak` dipanggil dengan `daftarKontak` sebagai argumen.
- `else if (pilihan == '2') {...}`: Mengecek apakah pengguna memilih opsi 2. Jika iya, maka fungsi `tambahKontak` dipanggil dengan `daftarKontak` sebagai argumen.
- `else if (pilihan == '3') {...}`: Mengecek apakah pengguna memilih opsi 3. Jika iya, maka fungsi `hapusKontak` dipanggil dengan `daftarKontak` sebagai argumen.
- `else if (pilihan == '4') {...}`: Mengecek apakah pengguna memilih opsi 4. Jika iya, maka teks "Keluar dari program..." ditampilkan, dan `break` digunakan untuk keluar dari loop.
- `else {...}`: Jika pilihan tidak valid (bukan 1, 2, 3, atau 4), maka teks "Pilihan tidak valid, coba lagi." ditampilkan.

```
if (pilihan == '1') {
    tampilkanKontak(daftarKontak);
} else if (pilihan == '2') {
    tambahKontak(daftarKontak);
} else if (pilihan == '3') {
    hapusKontak(daftarKontak);
} else if (pilihan == '4') {
    print('Keluar dari program...');
    break;
} else {
    print('Pilihan tidak valid, coba lagi.');
```

5. Fungsi tampilkanKontak

- void tampilkanKontak(List<Kontak> daftarKontak): Fungsi ini menerima sebuah list daftarKontak sebagai parameter dan menampilkan semua kontak yang tersimpan.
- if (daftarKontak.isEmpty): Mengecek apakah daftarKontak kosong. Jika iya, maka pesan "Belum ada kontak yang disimpan." ditampilkan.
- else {...}: Jika daftarKontak tidak kosong, maka program akan menampilkan semua kontak yang tersimpan.
- for (int i = 0; i < daftarKontak.length; i++): Loop yang iterasi sebanyak jumlah kontak yang ada di daftarKontak. Setiap kontak akan ditampilkan dengan format Nomorurut. Nama: ..., Nomor:

```
void tampilkanKontak(List<Kontak> daftarKontak) {  
    if (daftarKontak.isEmpty) {  
        print('Belum ada kontak yang disimpan.');    } else {  
        print('\n==== Kontak ====');  
        for (int i = 0; i < daftarKontak.length; i++) {  
            print('${i + 1}. Nama: ${daftarKontak[i].nama}, Nomor:  
${daftarKontak[i].nomorTelepon}');        }  
    }  
}
```

6. Fungsi tambahKontak

- void tambahKontak(List<Kontak> daftarKontak): Fungsi ini digunakan untuk menambahkan kontak baru ke daftarKontak.
- stdout.write('Masukkan Nama: '); dan stdout.write('Masukkan Nomor Telepon: ');: Meminta pengguna untuk memasukkan nama dan nomor telepon.
- String? nama = stdin.readLineSync(); dan String? nomorTelepon = stdin.readLineSync();: Membaca input pengguna untuk nama dan nomor telepon.
- if (nama != null && nomorTelepon != null): Mengecek apakah pengguna telah memasukkan nama dan nomor telepon yang valid (tidak null). Jika valid, maka kontak baru ditambahkan ke daftarKontak menggunakan daftarKontak.add(Kontak(nama, nomorTelepon));.
- else {...}: Jika salah satu dari nama atau nomorTelepon null, maka program menampilkan pesan bahwa input tidak boleh kosong.

```
void tambahKontak(List<Kontak> daftarKontak) {  
    stdout.write('Masukkan Nama: ');
```

```
String? nama = stdin.readLineSync();
stdout.write('Masukkan Nomor Telepon: ');
String? nomorTelepon = stdin.readLineSync();

if (nama != null && nomorTelepon != null) {
    daftarKontak.add(Kontak(nama, nomorTelepon));
    print('Kontak berhasil ditambahkan.');
```

```
} else {
    print('Nama atau nomor telepon tidak boleh kosong.');
```

```
}
}
```

7. Fungsi hapusKontak

- void hapusKontak(List<Kontak> daftarKontak): Fungsi ini menghapus kontak dari daftarKontak berdasarkan nomor urut yang diberikan oleh pengguna.
- if (daftarKontak.isEmpty): Mengecek apakah daftarKontak kosong. Jika iya, maka pesan "Belum ada kontak yang disimpan." ditampilkan.
- stdout.write('Masukkan nomor urut kontak yang ingin dihapus: '); Meminta pengguna untuk memasukkan nomor urut kontak yang ingin dihapus.
- String? input = stdin.readLineSync(); Membaca input dari pengguna sebagai string.
- int? index = int.tryParse(input ?? ""); Mencoba mengonversi input ke tipe int. Jika input tidak dapat dikonversi, index akan menjadi null.
- if (index != null && index > 0 && index <= daftarKontak.length): Mengecek apakah index valid (tidak null, lebih besar dari 0, dan tidak melebihi jumlah kontak yang ada). Jika valid, kontak pada urutan tersebut dihapus menggunakan daftarKontak.removeAt(index - 1);.
- else {...}: Jika index tidak valid, program menampilkan pesan bahwa nomor urut tidak valid.

```
void hapusKontak(List<Kontak> daftarKontak) {
    if (daftarKontak.isEmpty) {
        print('Belum ada kontak yang disimpan.');
```

```
} else {
    stdout.write('Masukkan nomor urut kontak yang ingin dihapus: ');
    String? input = stdin.readLineSync();
    int? index = int.tryParse(input ?? '');

    if (index != null && index > 0 && index <= daftarKontak.length) {
        daftarKontak.removeAt(index - 1);
        print('Kontak berhasil dihapus.');
```

```
    } else {  
        print('Nomor urut tidak valid.');
```