Nama : Putri Narila Tanaba

Npm : 0735221170

Kelas : 5IF3

Mata Kuliah : Pemrograman Mobile

Code program:

enum FaseProyek { Perencanaan, Pengembangan, Evaluasi }

class ProdukDigital {

  String namaProduk;

  double harga;

  String kategori;

  ProdukDigital(this.namaProduk, this.harga, this.kategori);

  double terapkanDiskon() {

    if (kategori == 'NetworkAutomation' && harga >= 200000) {

      return harga \* 0.85; // Diskon 15%

    }

    return harga;

  }

}

abstract class Karyawan {

  String nama;

  int umur;

  String peran;

  Karyawan(this.nama, {required this.umur, required this.peran});

  void bekerja();

}

mixin Kinerja {

  int produktivitas = 0;

  void updateProduktivitas(int nilai) {

    if (nilai < 0 || nilai > 100) {

      throw Exception('Nilai produktivitas harus antara 0 dan 100.');

    }

    produktivitas = nilai;

    print('Produktivitas diperbarui menjadi $produktivitas.');

  }

}

class KaryawanTetap extends Karyawan with Kinerja {

  KaryawanTetap(String nama, {required int umur, required String peran})

    : super(nama, umur: umur, peran: peran);

  @override

  void bekerja() {

    print('$nama bekerja sebagai Karyawan Tetap.');

  }

}

class KaryawanKontrak extends Karyawan with Kinerja {

  KaryawanKontrak(String nama, {required int umur, required String peran})

    : super(nama, umur: umur, peran: peran);

  @override

  void bekerja() {

    print('$nama bekerja sebagai Karyawan Kontrak.');

  }

  @override

  void updateProduktivitas(int nilai) {

    super.updateProduktivitas(nilai);

    if (peran == 'Manager' && produktivitas < 85) {

      throw Exception('Produktivitas Manager harus di atas atau sama dengan 85.');

    }

    print('Produktivitas $nama diperbarui menjadi $produktivitas.');

  }

}

class Perusahaan {

  List<Karyawan> karyawanAktif = [];

  List<Karyawan> karyawanNonAktif = [];

  void tambahKaryawan(Karyawan karyawan) {

    if (karyawanAktif.length < 20) {

      karyawanAktif.add(karyawan);

      print('${karyawan.nama} ditambahkan sebagai karyawan aktif.');

    } else {

      print('Batas maksimal karyawan aktif telah tercapai.');

    }

  }

  void resignKaryawan(Karyawan karyawan) {

    if (karyawanAktif.contains(karyawan)) {

      karyawanAktif.remove(karyawan);

      karyawanNonAktif.add(karyawan);

      print('${karyawan.nama} telah resign dan menjadi karyawan non-aktif.');

    }

  }

}

void main() {

  var produk1 = ProdukDigital('Router', 250000, 'NetworkAutomation');

  double hargaAkhir = produk1.terapkanDiskon();

  print('Harga akhir produk ${produk1.namaProduk}: $hargaAkhir');

  Perusahaan tongIT = Perusahaan();

  var k1 = KaryawanTetap('Alice', umur: 30, peran: 'Developer');

  var k2 = KaryawanKontrak('Bob', umur: 25, peran: 'Manager');

  tongIT.tambahKaryawan(k1);

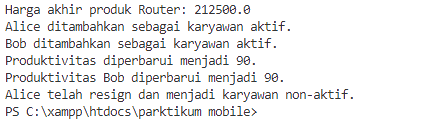
  tongIT.tambahKaryawan(k2);

  k2.updateProduktivitas(90);

  tongIT.resignKaryawan(k1);

}

Berikut ini adalah hasil running code program diatas:



Gambar 2.1 hasil *running* program

Referensi:

1. <https://dart-tutorial.com/object-oriented-programming/enum-in-dart/>
2. <https://dart-tutorial.com/object-oriented-programming/abstract-class-in-dart/>
3. <https://dart-tutorial.com/object-oriented-programming/mixins-in-dart/>
4. <https://dart-tutorial.com/asynchronous-programming/asynchronous-programming-in-dart/>