Nama : Siti Hajar Kasman

Npm : 0735221159

Kelas : 5IF3

Mata Kuliah : Pemrograman Mobile

1. Impelementasi Source Code

enum FaseProyek { Perencanaan, Pengembangan, Evaluasi }

class ProdukDigital {

  String namaProduk;

  double harga;

  String kategori;

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

  void terapkanDiskon(double diskon) {

    if (kategori == 'NetworkAutomation') {

      harga -= harga \* diskon / 100;

    }

  }

}

abstract class Karyawan {

  String nama;

  int umur;

  String peran;

  Karyawan(this.nama, this.umur, this.peran);

  void bekerja();

}

class KaryawanTetap extends Karyawan {

  KaryawanTetap(String nama, int umur, String peran) : super(nama, umur, peran);

  @override

  void bekerja() {

    print('$nama bekerja dengan jam kerja tetap.');

  }

}

class KaryawanKontrak extends Karyawan {

  KaryawanKontrak(String nama, int umur, String peran) : super(nama, umur, peran);

  @override

  void bekerja() {

    print('$nama bekerja dengan jam kerja kontrak.');

  }

}

mixin Kinerja {

  int produktivitas = 80;

  void naikkanProduktivitas() {

    produktivitas += 5;

  }

}

class Manager extends Karyawan with Kinerja {

  Manager(String nama, int umur) : super(nama, umur, 'Manager');

  @override

  void naikkanProduktivitas() {

    if (produktivitas <= 85) {

      produktivitas = 85;

    } else {

      produktivitas += 5;

    }

  }

  @override

  void bekerja() {

    print('$nama bekerja sebagai Manager dengan produktivitas $produktivitas.');

  }

}

class Perusahaan {

  List<Karyawan> karyawanAktif = [];

  List<Karyawan> karyawanNonAktif = [];

  static const int maxKaryawanAktif = 20;

  void tambahKaryawan(Karyawan karyawan) {

    if (karyawanAktif.length < maxKaryawanAktif) {

      karyawanAktif.add(karyawan);

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

    } else {

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

    }

  }

  void resignKaryawan(Karyawan karyawan) {

    karyawanAktif.remove(karyawan);

    karyawanNonAktif.add(karyawan);

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

  }

}

class Proyek {

  FaseProyek faseSaatIni;

  Proyek(this.faseSaatIni);

  void transisiFase(FaseProyek faseBaru) {

    if (faseSaatIni == FaseProyek.Perencanaan && faseBaru == FaseProyek.Pengembangan) {

      faseSaatIni = faseBaru;

      print('Transisi ke Pengembangan berhasil.');

    } else if (faseSaatIni == FaseProyek.Pengembangan && faseBaru == FaseProyek.Evaluasi) {

      faseSaatIni = faseBaru;

      print('Transisi ke Evaluasi berhasil.');

    } else {

      print('Transisi tidak valid.');

    }

  }

}

void main() {

  var produk = ProdukDigital('Router', 1500, 'NetworkAutomation');

  print('Harga sebelum diskon: ${produk.harga}');

  produk.terapkanDiskon(10); // Diskon 10%

  print('Harga setelah diskon: ${produk.harga}');

  var karyawanTetap = KaryawanTetap('Ali', 30, 'Programmer');

  var karyawanKontrak = KaryawanKontrak('Budi', 28, 'Desainer');

  karyawanTetap.bekerja();

  karyawanKontrak.bekerja();

  var manager = Manager('Joko', 40);

  print('Produktivitas awal: ${manager.produktivitas}');

  manager.naikkanProduktivitas();

  print('Produktivitas setelah 30 hari: ${manager.produktivitas}');

  manager.bekerja();

  var perusahaan = Perusahaan();

  perusahaan.tambahKaryawan(manager);

  perusahaan.tambahKaryawan(karyawanTetap);

  var proyek = Proyek(FaseProyek.Perencanaan);

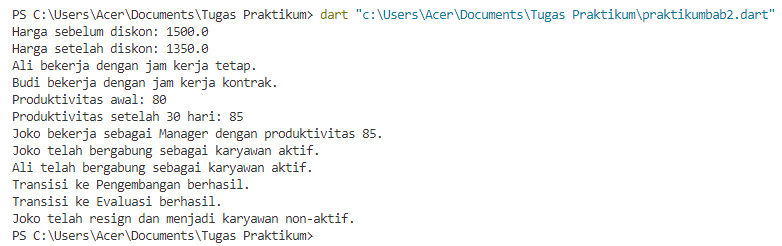
  proyek.transisiFase(FaseProyek.Pengembangan);

  proyek.transisiFase(FaseProyek.Evaluasi);

  perusahaan.resignKaryawan(manager);

}

1. Hasil Output dari program diatas:



1. Referensi yang dipakai oleh program diatas.

* <https://dart.dev/guides/language/language-tour#enums>
* <https://dart.dev/guides/language/language-tour#mixin>
* <https://dart.dev/guides/language/language-tour>