*#include* <stdio.h>

*#include* "garis.h"

*/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/*

*/\* DEFINISI PRIMITIF                                                 \*/*

*/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/*

*/\* \*\*\* Konstruktor membentuk GARIS \*\*\* \*/*

void MakeGARIS(POINT P1, POINT P2, GARIS \*L){

    PAwal(\*L) = P1;

    PAkhir(\*L) = P2;

}

*/\* I.S. P1 dan P2 terdefinisi \*/*

*/\* F.S. L terdefinisi dengan L.PAw = P1 dan L.PAkh = P2 \*/*

*/\* Membentuk sebuah L dari komponen-komponennya \*/*

*/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/*

*/\* KELOMPOK INTERAKSI DENGAN I/O DEVICE, BACA/TULIS                  \*/*

*/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/*

void BacaGARIS(GARIS \*L){

    POINT P1,P2;

    BacaPOINT(&P1);

    BacaPOINT(&P2);

*while* (EQ(P1,P2)){

        printf("Garis tidak valid\n");

        BacaPOINT(&P1);

        BacaPOINT(&P2);

    }

    MakeGARIS(P1,P2,L);

}

*/\* I.S. sembarang \*/*

*/\* F.S. L terdefinisi sebagai GARIS yang valid \*/*

*/\* Proses : mengulangi membaca dua buah nilai P1 dan P2 sehingga dapat*

*membentuk GARIS yang valid*

*GARIS disebut valid jika titik awal tidak sama dengan titik akhirnya. \*/*

*/\* Jika pembacaan garis tidak valid, diberikan pesan kesalahan:*

*"Garis tidak valid" \*/*

*/\* Contoh:*

*2 3*

*2 3*

*Garis tidak valid*

*2 3*

*3 3*

*--> terbentuk garis L dengan L.PAw = <2,3> dan L.PAkh = <3,3> \*/*

void TulisGARIS(GARIS L){

    printf("(");

    TulisPOINT(PAwal(L));

    printf(",");

    TulisPOINT(PAkhir(L));

    printf(")");

}

*/\* I.S. L terdefinisi \*/*

*/\* F.S. L tertulis di layar dengan format "((x1,y1),(x2,y2))"*

*tanpa tambahan karakter apa pun di depan, di tengah, atau di belakang,*

*termasuk spasi dan enter \*/*

*/\* Output X dan Y harus dituliskan dalam bilangan riil dengan 2 angka di belakang koma. \*/*

*/\* \*\*\* Kelompok operasi garis \*\*\* \*/*

float PanjangGARIS(GARIS L){

*return* (Panjang(PAwal(L),PAkhir(L)));

}

*/\* Menghitung panjang garis L : berikan rumusnya \*/*

float Gradien(GARIS L){

*return*((Ordinat(PAkhir(L))-Ordinat(PAwal(L)))/(Absis(PAkhir(L))-Absis(PAwal(L))));

}

*/\* Mengembalikan Gradien (m) dari L \*/*

*/\* Gradien garis yang melalui (x1,y1) dan (x2,y2) adalah: (y2-y1)/(x2-x1) \*/*

*/\* Prekondisi : x1 != x2 \*/*

*/\* \*\*\* Kelompok predikat \*\*\* \*/*

boolean IsTegakLurus(GARIS L1, GARIS L2){

*return* (Gradien(L1)\*Gradien(L2) == -1);

}

*/\* Menghasilkan true jika L1 tegak lurus terhadap L2 \*/*

*/\* Dua garis saling tegak lurus jika hasil perkalian kedua gradiennya = -1 \*/*

boolean IsSejajar(GARIS L1, GARIS L2){

*return* (Gradien(L1)==Gradien(L2));

}

*/\* Menghasilkan true jika L "sejajar" terhadap L1 \*/*

*/\* Dua garis saling sejajar jika memiliki gradien yang sama \*/*