3. Tipe bentukan garis (isSejajar, panjang garis)

TYPE GARIS

DEFINISI TYPE

type point : <x: real , y: real>
{<x,y> adalah sebuah point, dengan x adalah absis, y adalah ordinat }

type garis : <P1: point , P2: point>
{<P1,P2> adalah garis direpresentasikan oleh dua titik P1 dan P2 yang berada dalam bidang dua dimensi }

DEFINISI DAN SPESIFIKASI SELEKTOR

Absis : point → real
{ Absis(P) Memberikan Absis Point P }

Ordinat : point → real
{ Ordinat(P) Memberikan ordinat Point P }

GarisAwal : garis → point
{ GarisAwal(G) memberikan titik awal garis G }

GarisAkhir : garis → point
{ GarisAkhir(G) memberikan titik akhir garis G }

DEFINISI DAN SPESIFIKASI KONSTRUKTOR

MakePoint : 2 real → point
{ MakePoint(a,b) membentuk sebuah point dari a dan b dengan a sebagai absis dan b
sebagai ordinat }

MakeGaris : 2 point → garis
{ MakeGaris(P1, P2) membentuk sebuah garis dengan titik awal P1 dan titik akhir P2 }

DEFINISI DAN SPESIFIKASI OPERATOR TERHADAP GARIS

PanjangGaris : garis → boolean
{ PanjangGaris(garis) menghitung panjang garis antara dua titik Absis(garis) dan Ordinat(garis) menggunakan rumus jarak Euclidean }

DEFINISI DAN SPESIFIKASI PREDIKAT

IsSejajar? : 2 garis → boolean
{ IsSejajar(garisAwal, garisAkhir) mengecek apakah garis yang dibentuk oleh Absis(garisAwal)-Ordinat(garisAwal) dan Absis(GarisAkhir)-Ordinat(GarisAkhir) sejajar. Dua garis sejajar jika dan hanya jika gradien (kemiringan) kedua garis sama. }

DEFINISI OPERATOR/FUNGSI LAIN TERHADAP POINT

Jarak : 2 point → real
{Jarak(P1,P2) : menghitung jarak antara 2 point P1 dan P2 }

Gradien : 2 point → real
{Gradien(P1,P2) : menghitung gradien antara 2 point P1 dan P2}
{ Fungsi antara yang dipakai : FX2 adalah pangkat dua yang pernah didefinisikan pada
least square dan SQRT(X) adalah fungsi dasar untuk menghitung akar}

REALISASI

Jarak (P1,P2) :
SQRT (FX2 (Absis(P1) – Absis(P2)) +
FX2 (Ordinat(P1) – Ordinat (P2)))

Gradien (P1,P2) :
((Ordinat(P2) - Ordinat(P1)) / (Absis(P2) - Absis(P1)))

IsSejajar? (G1,G2) :
If Gradien(GarisAwal(G1), GarisAkhir(G1)) == Gradien(GarisAwal(G2), GarisAkhir(G2) then true
else false

PanjangGaris(G):
Jarak(GarisAwal(G), GarisAkhir(G))

4. Tipe bentukan segiempat (isBujurSangkar, isJajargenjang, AreaBujurSangkar)

TYPE GARIS

DEFINISI TYPE

type point : <x: real , y: real>
{<x,y> adalah sebuah point, dengan x adalah absis, y adalah ordinat }

type garis : <P1: point, y: point>
{<P1,P2> adalah garis direpresentasikan oleh dua titik P1 dan P2 yang berada dalam bidang dua dimensi }

type segiempat : <G1: garis, G2: garis, G3: garis, G4: garis>
{ Sebuah segiempat direpresentasikan oleh empat garis G1, G2, G3, dan G4 yang berada dalam bidang dua dimensi }

DEFINISI DAN SPESIFIKASI SELEKTOR

Absis : point → real
{ Absis(P) Memberikan Absis Point P }

Ordinat : point → real
{ Ordinat(P) Memberikan ordinat Point P }

GarisAwal : garis → point
{ GarisAwal(G) memberikan titik awal garis G }

GarisAkhir : garis → point
{ GarisAkhir(G) memberikan titik akhir garis G }

GarisSatuSegiempat : segiempat → garis
{ GarisSatuSegiEmpat(segiempat) memberikan garis ke satu pada bagian bawah segiempat}

GarisDuaSegiempat : segiempat → garis
{ GarisDuaSegiEmpat(segiempat) memberikan garis ke dua pada bagian kanan segiempat}

GarisTigaSegiempat : segiempat → garis
{ GarisTigaSegiEmpat(segiempat) memberikan garis ke tiga pada bagian atas segiempat}

GarisEmpatSegiempat : segiempat → garis

{ GarisEmpatSegiEmpat(segiempat) memberikan garis ke empat pada bagian kiri segiempat}

DEFINISI DAN SPESIFIKASI KONSTRUKTOR

MakePoint : 2 real → point
{ MakePoint(a,b) membentuk sebuah point dari a dan b dengan a sebagai absis dan b sebagai ordinat }

MakeGaris : 2 point → garis
{ MakeGaris(P1, P2) membentuk sebuah garis dengan titik awal P1 dan titik akhir P2 }

MakeSegiempat : 4 garis → segiempat
{ MakeSegiempat(*G1*, *G2*, *G3*, *G4*) membentuk sebuah segiempat dengan:
garis kesatu G1 pada bagian bawah,
garis kedua G2 pada bagian kanan,
garis ketiga G3 pada bagian atas,
garis keempat G4 pada bagian kiri }

DEFINISI OPERATOR/FUNGSI LAIN TERHADAP POINT

Jarak : 2 point → real
{Jarak(P1,P2) : menghitung jarak antara 2 point P1 dan P2 }

Gradien : 2 point → real
{Gradien(P1,P2) : menghitung gradien antara 2 point P1 dan P2}

{ Fungsi antara yang dipakai : FX2 adalah pangkat dua yang pernah didefinisikan pada least square dan SQRT(X) adalah fungsi dasar untuk menghitung akar}

DEFINISI DAN SPESIFIKASI OPERATOR TERHADAP GARIS

PanjangGaris : garis → real
{ PanjangGaris(garis) menghitung panjang garis antara dua titik Absis(garis) dan Ordinat(garis) menggunakan rumus jarak Euclidean }

DEFINISI DAN SPESIFIKASI OPERATOR TERHADAP SEGIEMPAT

AreaBujurSangkar : segiempat → real
{ AreaBujurSangkar(*segiempat*) menghitung luas area dari bujur sangkar yang diberikan menggunakan rumus kuadrat dari panjang garis}

DEFINISI DAN SPESIFIKASI PREDIKAT

IsBujurSangkar? : segiempat → boolean
{ IsBujurSangkar(segiempat) mengecek apakah segiempat yang diberikan adalah bujur sangkat. Segiempat dikatakan bujur sangkar jika keempat sisinya sama panjang . }

IsJajarGenjang? : segiempat → boolean
{ IsJajarGenjang(segiempat) mengecek apakah segiempat yang diberikan adalah jajar genjang. Segiempat dikatakan jajar genjang jika sisi-sisi yang berhadapan sama panjang dan sejajar. }

DEFINISI DAN SPESIFIKASI FUNGSI TAMBAHAN

FX2 : real -> real
{ FX2(x) adalah hasil kuadrat dari x }

REALISASI

```
Jarak (P1,P2) :
        SQRT (FX2 (Absis(P1) – Absis(P2)) +
                FX2 (Ordinat(P1) – Ordinat (P2)))
```

```
Gradien (P1,P2) :
        ((Ordinat(P2) - Ordinat(P1)) / (Absis(P2) - Absis(P1)))

IsBujurSangkar(segiempat):
        if Jarak (
                GarisAwal(GarisSatuSegiempat(segiempat)),
                GarisAkhir(GarisSatuSegiempat(segiempat)),
        ) * Jarak (
                GarisAwal(GarisDuaSegiempat(segiempat)),
                GarisAkhir(GarisDuaSegiempat(segiempat)),
        ) = Jarak (
                GarisAwal(GarisTigaSegiempat(segiempat)),
                GarisAkhir(GarisTigaSegiempat(segiempat)),
        ) then true
        else false

IsJajargenjang(segiempat):
        if Gradien(
                GarisAwal(GarisSatuSegiempat(segiempat)),
                GarisAkhir(GarisSatuSegiempat(segiempat)),
        ) = Gradien(
                GarisAwal(GarisTigaSegiempat(segiempat)),
                GarisAkhir(GarisTigaSegiempat(segiempat)),
        ) and Gradien(
                GarisAwal(GarisSatuSegiempat(segiempat)),
                GarisAkhir(GarisTigaSegiempat(segiempat)),
        ) = Gradien(
                GarisAwal(GarisDuaSegiempat(segiempat)),
                GarisAkhir(GarisDuaSegiempat(segiempat)),
        ) and Jarak(
                GarisAwal(GarisSatuSegiempat(segiempat)),
                GarisAkhir(GarisSatuSegiempat(segiempat)),
        ) = Jarak(
                GarisAwal(GarisTigaSegiempat(segiempat)),
                GarisAkhir(GarisTigaSegiempat(segiempat)),
        ) and  Jarak(
                GarisAwal(GarisSatuSegiempat(segiempat)),
                GarisAkhir(GarisTigaSegiempat(segiempat)),
        ) = Jarak(
                GarisAwal(GarisDuaSegiempat(segiempat)),
                GarisAkhir(GarisDuaSegiempat(segiempat)),
        ) then true
        else false
```

```
AreaBujurSangkar(segiempat):
    FX2(
        Jarak(
            GarisAwal(GarisSatuSegiempat(segiempat)),
            GarisAkhir(GarisSatuSegiempat(segiempat)),
        )
    )
```