

### 3. Tipe bentukan garis (isSejajar, panjang garis)

<b>TIPE BENTUKAN GARIS</b>
<b>DEFINISI TYPE</b> <u>type point</u> : <x: <u>real</u> , y: <u>real</u> > { <x,y> adalah sebuah point, dengan x adalah absis, y adalah ordinat } <u>type garis</u> : <P1: <u>point</u> , P2: <u>point</u> > { <P1,P2> adalah garis direpresentasikan oleh dua titik P1 dan P2 yang berada dalam bidang dua dimensi }
<b>DEFINISI DAN SPESIFIKASI SELEKTOR</b> <b>Absis</b> : <u>point</u> → <u>real</u> { Absis(P) Memberikan Absis Point P } <b>Ordinat</b> : <u>point</u> → <u>real</u> { Ordinat(P) Memberikan ordinat Point P } <b>GarisAwal</b> : <u>garis</u> → <u>point</u> { GarisAwal(G) memberikan titik awal garis G } <b>GarisAkhir</b> : <u>garis</u> → <u>point</u> { GarisAkhir(G) memberikan titik akhir garis G }
<b>DEFINISI DAN SPESIFIKASI KONSTRUKTOR</b> <b>MakePoint</b> : 2 <u>real</u> → <u>point</u> { MakePoint(a,b) membentuk sebuah point dari a dan b dengan a sebagai absis dan b sebagai ordinat } <b>MakeGaris</b> : 2 <u>point</u> → <u>garis</u> { MakeGaris(P1, P2) membentuk sebuah garis dengan titik awal P1 dan titik akhir P2 }
<b>DEFINISI DAN SPESIFIKASI OPERATOR TERHADAP GARIS</b> <b>PanjangGaris</b> : <u>garis</u> → <u>boolean</u> { PanjangGaris(garis) menghitung panjang garis antara dua titik Absis(garis) dan Ordinat(garis) menggunakan rumus jarak Euclidean }
<b>DEFINISI DAN SPESIFIKASI PREDIKAT</b> <b>IsSejajar?</b> : 2 <u>garis</u> → <u>boolean</u> { 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. }
<b>DEFINISI OPERATOR/FUNGSI LAIN TERHADAP POINT</b> <b>Jarak</b> : 2 <u>point</u> → <u>real</u> { Jarak(P1,P2) : menghitung jarak antara 2 point P1 dan P2 } <b>Gradien</b> : 2 <u>point</u> → <u>real</u> { 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}
<b>REALISASI</b> Jarak (P1,P2) : $\text{SQRT} (\text{FX2} (\text{Absis}(\text{P1}) - \text{Absis}(\text{P2})) + \text{FX2} (\text{Ordinat}(\text{P1}) - \text{Ordinat} (\text{P2})))$

<p>Gradien (P1,P2) :</p> $((\text{Ordinat}(P2) - \text{Ordinat}(P1)) / (\text{Absis}(P2) - \text{Absis}(P1)))$ <p>IsSejajar? (G1,G2) :</p> <p>    If Gradien(GarisAwal(G1), GarisAkhir(G1)) == Gradien(GarisAwal(G2), GarisAkhir(G2)) <u>then</u> true</p> <p>    <u>else</u> false</p> <p>PanjangGaris(G):</p> <p>    Jarak(GarisAwal(G), GarisAkhir(G))</p>

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

TIPE BENTUKAN SEGIEMPAT
<p><b>DEFINISI TYPE</b></p> <p><b>type point</b> : &lt;x: <u>real</u> , y: <u>real</u>&gt;</p> <p>    {&lt;x,y&gt; adalah sebuah point, dengan x adalah absis, y adalah ordinat }</p> <p><b>type garis</b> : &lt;P1: <u>point</u>, P2: <u>point</u>&gt;</p> <p>    {&lt;P1,P2&gt; adalah garis direpresentasikan oleh dua titik P1 dan P2 yang berada dalam bidang dua dimensi }</p> <p><b>type segiempat</b> : &lt;G1: <u>garis</u>, G2: <u>garis</u>, G3: <u>garis</u>, G4: <u>garis</u>&gt;</p> <p>    { Sebuah segiempat direpresentasikan oleh empat garis G1, G2, G3, dan G4 yang berada dalam bidang dua dimensi }</p>
<p><b>DEFINISI DAN SPESIFIKASI SELEKTOR</b></p> <p><b>Absis</b> : <u>point</u> → <u>real</u></p> <p>    { Absis(P) Memberikan Absis Point P }</p> <p><b>Ordinat</b> : <u>point</u> → <u>real</u></p> <p>    { Ordinat(P) Memberikan ordinat Point P }</p> <p><b>GarisAwal</b> : <u>garis</u> → <u>point</u></p> <p>    { GarisAwal(G) memberikan titik awal garis G }</p> <p><b>GarisAkhir</b> : <u>garis</u> → <u>point</u></p> <p>    { GarisAkhir(G) memberikan titik akhir garis G }</p> <p><b>GarisSatuSegiempat</b> : <u>segiempat</u> → <u>garis</u></p> <p>    { GarisSatuSegiEmpat(segiempat) memberikan garis ke satu pada bagian bawah segiempat }</p> <p><b>GarisDuaSegiempat</b> : <u>segiempat</u> → <u>garis</u></p> <p>    { GarisDuaSegiEmpat(segiempat) memberikan garis ke dua pada bagian kanan segiempat }</p> <p><b>GarisTigaSegiempat</b> : <u>segiempat</u> → <u>garis</u></p> <p>    { GarisTigaSegiEmpat(segiempat) memberikan garis ke tiga pada bagian atas segiempat }</p> <p><b>GarisEmpatSegiempat</b> : <u>segiempat</u> → <u>garis</u></p> <p>    { GarisEmpatSegiEmpat(segiempat) memberikan garis ke empat pada bagian kiri segiempat }</p>

### **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 sangkar. 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) :

$$\text{SQRT} (\text{FX2} (\text{Absis}(\text{P1}) - \text{Absis}(\text{P2})) + \text{FX2} (\text{Ordinat}(\text{P1}) - \text{Ordinat} (\text{P2})))$$

Gradien (P1,P2) :

$$((\text{Ordinat}(\text{P2}) - \text{Ordinat}(\text{P1})) / (\text{Absis}(\text{P2}) - \text{Absis}(\text{P1})))$$

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)),
        )
    )

```

## 2. Tipe bentukan tanggal (diktat)

### **REALISASI**

```

NextNDay(date, N):
    If dpm(Month(date)) + Day(date) - 1 + (if Month(date) > 2 and
    IsKabisat(Year(date)) then 1 else 0) + N > ( if IsKabisat(Year(date)) then 366 else 365):
        then HrKeDate(
            Year(date) + 1,
            (
                (dpm(Month(date)) + Day(date) - 1
                + (1 if Month(date) > 2 and IsKabisat(Year(date)) else 0))
                + N - (if IsKabisat(Year(date)) then 366 else 365)
            ),
        )
    else HrKeDate(
        Year(date),
        (dpm(Month(date))
        + Day(date) - 1
        + (1 if Month(date) > 2 and IsKabisat(Year(date)) else 0)
        ) + N,
    )

```

```

HrKeDate(Thn, TotalHr):
    depend on TotalHr
        TotalHr <= 31: MakeDate(TotalHr, 1, Thn)
        TotalHr <= (31 + (if IsKabisat(Thn) then 29 else 28)):
        MakeDate(TotalHr - 31, 2, Thn)
        TotalHr <= (62 + (if IsKabisat(Thn) then 29 else 28)):
        MakeDate(TotalHr - (31 + (if IsKabisat(Thn) then 29 else 28)), 3, Thn)
        TotalHr <= (92 + (if IsKabisat(Thn) then 29 else 28)):
        MakeDate(TotalHr - (62 + (if IsKabisat(Thn) then 29 else 28)), 4, Thn)
        TotalHr <= (123 + (if IsKabisat(Thn) then 29 else 28)):
        MakeDate(TotalHr - (92 + (if IsKabisat(Thn) then 29 else 28)), 5, Thn)
        TotalHr <= (153 + (if IsKabisat(Thn) then 29 else 28)):
        MakeDate(TotalHr - (123 + (if IsKabisat(Thn) then 29 else 28)), 6, Thn)
        TotalHr <= (184 + (if IsKabisat(Thn) then 29 else 28)):
        MakeDate(TotalHr - (153 + (if IsKabisat(Thn) then 29 else 28)), 7, Thn)

```

<p>TotalHr &lt;= (215 + (if IsKabisat(Thn) then 29 else 28)): MakeDate(TotalHr - (184 + (if IsKabisat(Thn) then 29 else 28)), 8, Thn)</p> <p>TotalHr &lt;= (245 + (if IsKabisat(Thn) then 29 else 28)): MakeDate(TotalHr - (215 + (if IsKabisat(Thn) then 29 else 28)), 9, Thn)</p> <p>TotalHr &lt;= (276 + (if IsKabisat(Thn) then 29 else 28)): MakeDate(TotalHr - (245 + (if IsKabisat(Thn) then 29 else 28)), 10, Thn)</p> <p>TotalHr &lt;= (306 + (if IsKabisat(Thn) then 29 else 28)): MakeDate(TotalHr - (276 + (if IsKabisat(Thn) then 29 else 28)), 11, Thn)</p> <p>TotalHr &lt;= (337 + (if IsKabisat(Thn) then 29 else 28)): MakeDate(TotalHr - (306 + (if IsKabisat(Thn) then 29 else 28)), 12, Thn)</p>