

Visualisasi dan Perhitungan Geometri dengan EMT

Euler menyediakan beberapa fungsi untuk melakukan visualisasi dan perhitungan geometri, baik secara numerik maupun analitik (seperti biasanya tentunya, menggunakan Maxima). Fungsi-fungsi untuk visualisasi dan perhitungan geometri tersebut disimpan di dalam file program "geometry.e", sehingga file tersebut harus dipanggil sebelum menggunakan fungsi-fungsi atau perintah-perintah untuk geometri.

```
>load geometry
```

Numerical and symbolic geometry.

Fungsi-fungsi Geometri

Fungsi-fungsi untuk Menggambar Objek Geometri:

defaultd:=textheight()*1.5: nilai asli untuk parameter d
setPlotrange(x1,x2,y1,y2): menentukan rentang x dan y pada bidang

koordinat

setPlotRange(r): pusat bidang koordinat (0,0) dan batas-batas

sumbu-x dan y adalah -r sd r

plotPoint (P, "P"): menggambar titik P dan diberi label "P"
plotSegment (A,B, "AB", d): menggambar ruas garis AB, diberi label

"AB" sejauh d

plotLine (g, "g", d): menggambar garis g diberi label "g" sejauh d
plotCircle (c,"c",v,d): Menggambar lingkaran c dan diberi label "c"
plotLabel (label, P, V, d): menuliskan label pada posisi P

Fungsi-fungsi Geometri Analitik (numerik maupun simbolik):

turn(v, phi): memutar vektor v sejauh phi
turnLeft(v): memutar vektor v ke kiri
turnRight(v): memutar vektor v ke kanan
normalize(v): normal vektor v
crossProduct(v, w): hasil kali silang vektor v dan w.
lineThrough(A, B): garis melalui A dan B, hasilnya [a,b,c] sdh.

$ax+by=c$.

`lineWithDirection(A,v)`: garis melalui A searah vektor v
`getLineDirection(g)`: vektor arah (gradien) garis g
`getNormal(g)`: vektor normal (tegak lurus) garis g
`getPointOnLine(g)`: titik pada garis g
`perpendicular(A, g)`: garis melalui A tegak lurus garis g
`parallel (A, g)`: garis melalui A sejajar garis g
`lineIntersection(g, h)`: titik potong garis g dan h
`projectToLine(A, g)`: proyeksi titik A pada garis g
`distance(A, B)`: jarak titik A dan B
`distanceSquared(A, B)`: kuadrat jarak A dan B
`quadrance(A, B)`: kuadrat jarak A dan B
`areaTriangle(A, B, C)`: luas segitiga ABC
`computeAngle(A, B, C)`: besar sudut $\angle ABC$
`angleBisector(A, B, C)`: garis bagi sudut $\angle ABC$
`circleWithCenter (A, r)`: lingkaran dengan pusat A dan jari-jari r
`getCircleCenter(c)`: pusat lingkaran c
`getCircleRadius(c)`: jari-jari lingkaran c
`circleThrough(A,B,C)`: lingkaran melalui A, B, C
`middlePerpendicular(A, B)`: titik tengah AB
`lineCircleIntersections(g, c)`: titik potong garis g dan lingkaran c
`circleCircleIntersections (c1, c2)`: titik potong lingkaran $c1$ dan

c2

`planeThrough(A, B, C)`: bidang melalui titik A, B, C

Fungsi-fungsi Khusus Untuk Geometri Simbolik:

`getLineEquation (g,x,y)`: persamaan garis g dinyatakan dalam x dan y
`getHesseForm (g,x,y,A)`: bentuk Hesse garis g dinyatakan dalam x dan

y dengan titik A pada

sisi positif (kanan/atas) garis
`quad(A,B)`: kuadrat jarak AB
`spread(a,b,c)`: Spread segitiga dengan panjang sisi-sisi a,b,c , yakni

$\sin(\alpha)^2$ dengan

α sudut yang menghadap sisi a .
`crosslaw(a,b,c,sa)`: persamaan 3 quads dan 1 spread pada segitiga

dengan panjang sisi a, b, c .

`triplespread(sa,sb,sc)`: persamaan 3 spread sa,sb,sc yang membentuk

suatu segitiga

`doublespread(sa)`: Spread sudut rangkap Spread 2ϕ , dengan

$sa = \sin(\phi)^2 \text{ spread } a$.

Contoh 1: Luas, Lingkaran Luar, Lingkaran Dalam Segitiga

Untuk menggambar objek-objek geometri, langkah pertama adalah menentukan rentang sumbu-sumbu koordinat. Semua objek geometri akan digambar pada satu bidang koordinat, sampai didefinisikan bidang koordinat yang baru.

```
>setPlotRange(-0.5,2.5,-0.5,2.5); // mendefinisikan bidang koordinat baru
```

Sekarang tentukan tiga titik dan gambarkan titik-titik tersebut.

```
>A=[1,0]; plotPoint(A,"A"); // definisi dan gambar tiga titik
>B=[0,1]; plotPoint(B,"B");
>C=[2,2]; plotPoint(C,"C");
```

Kemudian tiga segmen.

```
>plotSegment(A,B,"c"); // c=AB
>plotSegment(B,C,"a"); // a=BC
>plotSegment(A,C,"b"); // b=AC
```

Fungsi geometri mencakup fungsi untuk membuat garis dan lingkaran. Format untuk garis adalah $[a,b,c]$, yang mewakili garis dengan persamaan $ax+by=c$.

```
>lineThrough(B,C) // garis yang melalui B dan C
```

```
[-1, 2, 2]
```

Hitung garis tegak lurus melalui titik A pada garis BC.

```
>h=perpendicular(A,lineThrough(B,C)); // garis h tegak lurus BC melalui A
```

Dan persimpangannya dengan BC.

```
>D=lineIntersection(h,lineThrough(B,C)); // D adalah titik potong h dan BC
```

Buat plot itu.

```
>plotPoint(D,value=1); // koordinat D ditampilkan
>aspect(1); plotSegment(A,D); // tampilkan semua gambar hasil plot...()
```

Hitung luas ABC:

$$L_{\triangle ABC} = \frac{1}{2}AD.BC.$$

```
>norm(A-D)*norm(B-C)/2 // AD=norm(A-D), BC=norm(B-C)
```

1.5

Bandingkan dengan rumus determinan.

```
>areaTriangle(A,B,C) // hitung luas segitiga langsung dengan fungsi
```

1.5

Cara lain menghitung luas segitiga ABC:

```
>distance(A,D)*distance(B,C)/2
```

1.5

Sudut titik C.

```
>degprint(computeAngle(B,C,A))
```

36°52'11.63''

Sekarang lingkaran luar segitiga.

```
>c=circleThrough(A,B,C); // lingkaran luar segitiga ABC
>R=getCircleRadius(c); // jari2 lingkaran luar
>O=getCircleCenter(c); // titik pusat lingkaran c
>plotPoint(O,"O"); // gambar titik "O"
>plotCircle(c,"Lingkaran luar segitiga ABC");
```

Tampilkan koordinat titik pusat dan jari-jari lingkaran luar.

```
>O, R
```

```
[1.166666666667, 1.166666666667]
1.17851130198
```

Sekarang akan digambar lingkaran dalam segitiga ABC. Titik pusat lingkaran dalam adalah titik potong garis-garis bagi sudut.

```
>l=angleBisector(A,C,B); // garis bagi <ACB
>g=angleBisector(C,A,B); // garis bagi <CAB
>P=lineIntersection(l,g) // titik potong kedua garis bagi sudut
```

```
[0.860379610028, 0.860379610028]
```

Tambahkan semuanya ke plot.

```
>color(5); plotLine(l); plotLine(g); color(1); // gambar kedua garis bagi sudut  
>plotPoint(P,"P"); // gambar titik potongnya  
>r=norm(P-projectToLine(P,lineThrough(A,B))) // jari-jari lingkaran dalam
```

0.509653732104

```
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC"): // gambar lingkaran dal
```

Latihan

1. Tentukan ketiga titik singgung lingkaran dalam dengan sisi-sisi segitiga ABC.
2. Gambar segitiga dengan titik-titik sudut ketiga titik singgung tersebut. Merupakan segitiga apakah itu?
3. Hitung luas segitiga tersebut.
4. Tunjukkan bahwa garis bagi sudut yang ke tiga juga melalui titik pusat lingkaran dalam.
5. Gambar jari-jari lingkaran dalam.
6. Hitung luas lingkaran luar dan luas lingkaran dalam segitiga ABC. Adakah hubungan antara luas kedua lingkaran tersebut dengan luas segitiga ABC?

Contoh 2: Geometri Smbolik

Kita dapat menghitung geometri eksak dan simbolik menggunakan Maxima.

Berkas geometry.e menyediakan fungsi yang sama (dan lebih banyak lagi) dalam Maxima. Namun, kini kita dapat menggunakan perhitungan simbolik.

```
>A &= [1,0]; B &= [0,1]; C &= [2,2]; // menentukan tiga titik A, B, C
```

Fungsi untuk garis dan lingkaran bekerja sama seperti fungsi Euler, tetapi menyediakan perhitungan simbolik.

```
>c &= lineThrough(B,C) // c=BC
```

$[-1, 2, 2]$

Kita dapat dengan mudah mendapatkan persamaan garis.

```
>$getLineEquation(c,x,y), $solve(%,y) | expand // persamaan garis c
```

$$\left[y = \frac{x}{2} + 1 \right]$$

$$\left[y = \frac{x}{2} + 1 \right]$$

```
>$getLineEquation(lineThrough([x1,y1],[x2,y2]),x,y), $solve(%,y) // persamaan garis melalui
```

$$\left[y = \frac{-(x_1 - x) y_2 - (x - x_2) y_1}{x_2 - x_1} \right]$$

$$\left[y = \frac{-(x_1 - x) y_2 - (x - x_2) y_1}{x_2 - x_1} \right]$$

```
>$getLineEquation(lineThrough(A,[x1,y1]),x,y) // persamaan garis melalui A dan (x1, y1)
```

$$(x_1 - 1) y - x y_1 = -y_1$$

```
>h &= perpendicular(A,lineThrough(B,C)) // h melalui A tegak lurus BC
```

$$[2, 1, 2]$$

```
>Q &= lineIntersection(c,h) // Q titik potong garis c=BC dan h
```

$$\begin{bmatrix} 2 & 6 \\ 5 & 5 \end{bmatrix}$$

```
>$projectToLine(A,lineThrough(B,C)) // proyeksi A pada BC
```

$$\left[\frac{2}{5}, \frac{6}{5} \right]$$

```
>$distance(A,Q) // jarak AQ
```

$$\frac{3}{\sqrt{5}}$$

```
>cc &= circleThrough(A,B,C); $cc // (titik pusat dan jari-jari) lingkaran melalui A, B, C
```

$$\left[\frac{7}{6}, \frac{7}{6}, \frac{5}{3\sqrt{2}} \right]$$

```
>r:=getCircleRadius(cc); $r , $float(r) // tampilkan nilai jari-jari
```

1.178511301977579

```
>$computeAngle(A,C,B) // nilai <ACB
```

$$\arccos\left(\frac{4}{5}\right)$$

```
>$solve(getLineEquation(angleBisector(A,C,B),x,y),y)[1] // persamaan garis bagi <ACB
```

$$y = x$$

```
>P &= lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A)); $P // titik potong 2 ga
```

$$\left[\frac{\sqrt{2}\sqrt{5}+2}{6}, \frac{\sqrt{2}\sqrt{5}+2}{6} \right]$$

```
>P() // hasilnya sama dengan perhitungan sebelumnya
```

[0.86038, 0.86038]

Garis dan Lingkaran yang Berpotongan

Tentu saja, kita juga dapat memotong garis dengan lingkaran, dan lingkaran dengan lingkaran.

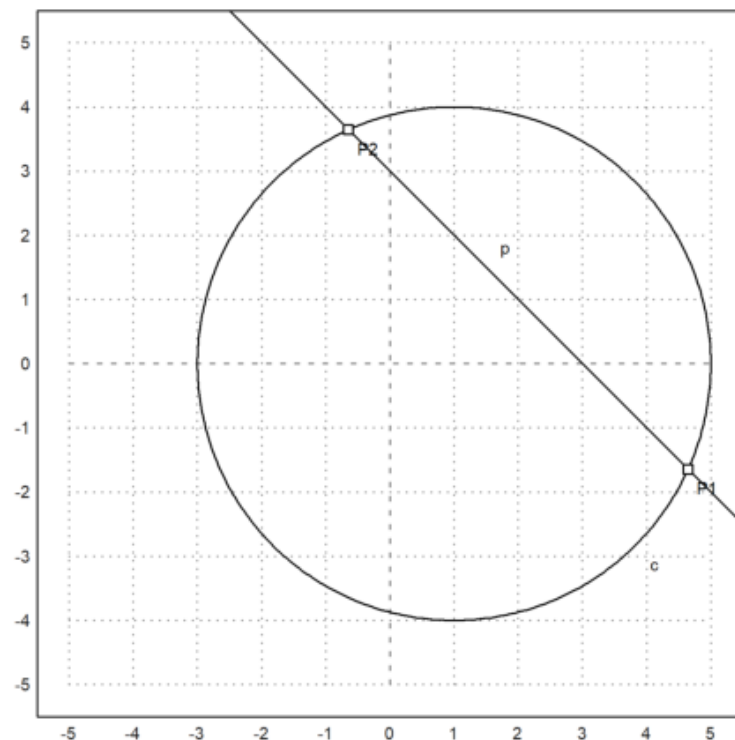
```
>A &:= [1,0]; c=circleWithCenter(A,4);
>B &:= [1,2]; C &:= [2,1]; l=lineThrough(B,C);
>setPlotRange(5); plotCircle(c); plotLine(l);
```

Perpotongan antara sebuah garis dengan lingkaran menghasilkan dua titik dan jumlah titik perpotongan.

```
>{P1,P2,f}=lineCircleIntersections(l,c);
>P1, P2, f
```

```
[4.64575, -1.64575]
[-0.645751, 3.64575]
2
```

```
>plotPoint(P1); plotPoint(P2):
```



Hal yang sama berlaku di Maxima.

```
>c := circleWithCenter(A,4) // lingkaran dengan pusat A jari-jari 4
```

```
[1, 0, 4]
```

```
>l := lineThrough(B,C) // garis l melalui B dan C
```

```
[1, 1, 3]
```

```
>$lineCircleIntersections(l,c) | radcan, // titik potong lingkaran c dan garis l
```

```
[[sqrt(7)+2, 1-sqrt(7)], [2-sqrt(7), sqrt(7)+1]]
```


Akan ditunjukkan bahwa sudut-sudut yang menghadap busur yang sama adalah sama besar.

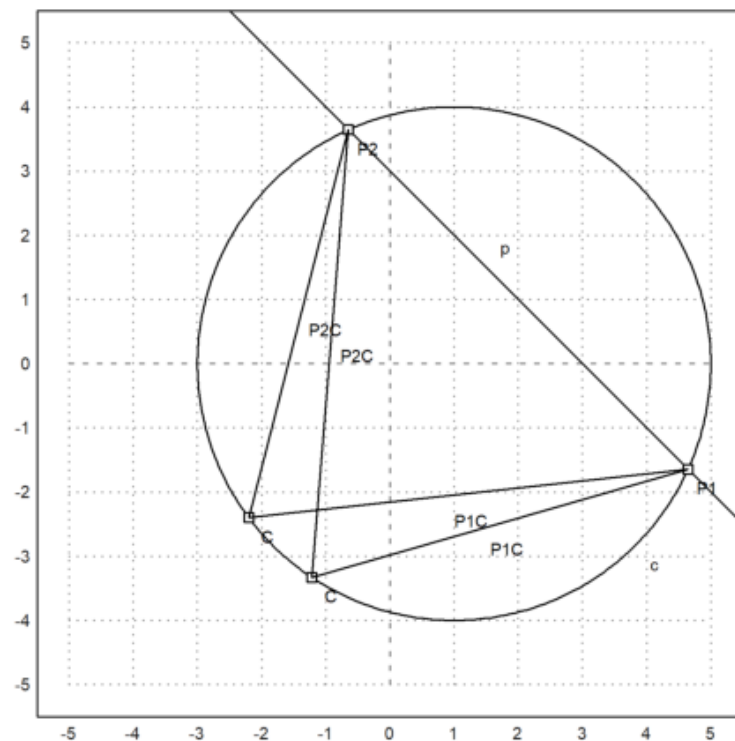
```
>C=A+normalize([-2,-3])*4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,C);  
>degprint(computeAngle(P1,C,P2))
```

69°17'42.68''

```
>C=A+normalize([-4,-3])*4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,C);  
>degprint(computeAngle(P1,C,P2))
```

69°17'42.68''

```
>insimg;
```



Garis Sumbu

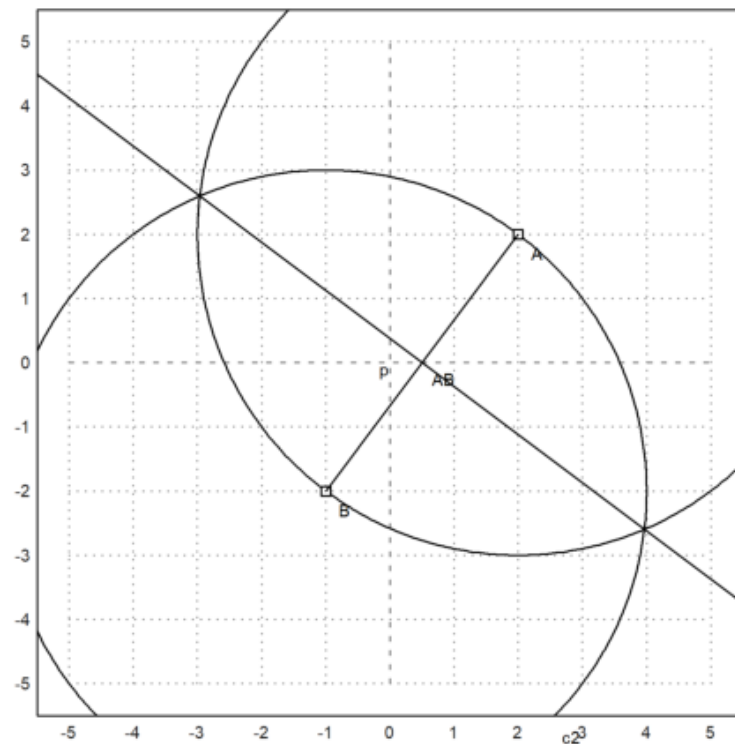
Berikut adalah langkah-langkah menggambar garis sumbu ruas garis AB:

1. Gambar lingkaran dengan pusat A melalui B.
2. Gambar lingkaran dengan pusat B melalui A.
3. Tarik garis melalui kedua titik potong kedua lingkaran tersebut. Garis ini merupakan garis sumbu (melalui titik tengah dan tegak lurus) AB.

```

>A=[2,2]; B=[-1,-2];
>c1=circleWithCenter(A,distance(A,B));
>c2=circleWithCenter(B,distance(A,B));
>{P1,P2,f}=circleCircleIntersections(c1,c2);
>l=lineThrough(P1,P2);
>setPlotRange(5); plotCircle(c1); plotCircle(c2);
>plotPoint(A); plotPoint(B); plotSegment(A,B); plotLine(l):

```



Selanjutnya, kita melakukan hal yang sama di Maxima dengan koordinat umum.

```

>A &= [a1,a2]; B &= [b1,b2];
>c1 &= circleWithCenter(A,distance(A,B));
>c2 &= circleWithCenter(B,distance(A,B));
>P &= circleCircleIntersections(c1,c2); P1 &= P[1]; P2 &= P[2];

```

Persamaan untuk titik potongnya cukup rumit. Namun, kita dapat menyederhanakannya jika kita menyelesaikan persamaan untuk y.

```

>g &= getLineEquation(lineThrough(P1,P2),x,y);
>$solve(g,y)

```

$$\left[y = \frac{-(2b_1 - 2a_1)x + b_2^2 + b_1^2 - a_2^2 - a_1^2}{2b_2 - 2a_2} \right]$$

Ini memang sama dengan garis tegak lurus tengah, yang dihitung dengan cara yang sama sekali berbeda.

```
>$solve(getLineEquation(middlePerpendicular(A,B),x,y),y)
```

$$\left[y = \frac{-(2b_1 - 2a_1)x + b_2^2 + b_1^2 - a_2^2 - a_1^2}{2b_2 - 2a_2} \right]$$

```
>h &=getLineEquation(lineThrough(A,B),x,y);
>$solve(h,y)
```

$$\left[y = \frac{(b_2 - a_2)x - a_1b_2 + a_2b_1}{b_1 - a_1} \right]$$

Perhatikan hasil kali gradien garis g dan h adalah:

$$\frac{-(b_1 - a_1)}{(b_2 - a_2)} \times \frac{(b_2 - a_2)}{(b_1 - a_1)} = -1.$$

Artinya kedua garis tegak lurus.

Contoh 3: Rumus Heron

Rumus Heron menyatakan bahwa luas segitiga dengan panjang sisi-sisi a, b dan c adalah:

$$L = \sqrt{s(s-a)(s-b)(s-c)} \quad \text{dengan } s = (a+b+c)/2,$$

atau bisa ditulis dalam bentuk lain:

$$L = \frac{1}{4} \sqrt{(a+b+c)(b+c-a)(a+c-b)(a+b-c)}$$

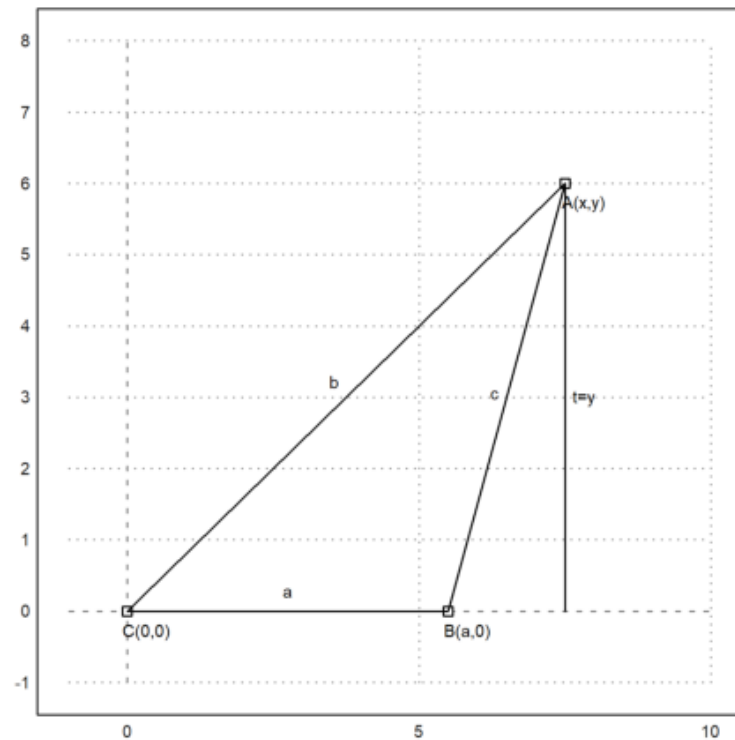
Untuk membuktikan hal ini kita misalkan C(0,0), B(a,0) dan A(x,y), b=AC, c=AB. Luas segitiga ABC adalah

$$L_{\triangle ABC} = \frac{1}{2}a \times y.$$

Nilai y didapat dengan menyelesaikan sistem persamaan:

$$x^2 + y^2 = b^2, \quad (x-a)^2 + y^2 = c^2.$$

```
>setPlotRange(-1,10,-1,8); plotPoint([0,0], "C(0,0)"); plotPoint([5.5,0], "B(a,0)"; ...
> plotPoint([7.5,6], "A(x,y)");
>plotSegment([0,0],[5.5,0], "a",25); plotSegment([5.5,0],[7.5,6], "c",15); ...
>plotSegment([0,0],[7.5,6], "b",25);
>plotSegment([7.5,6],[7.5,0], "t=y",25):
```



```
> assume(a>0); sol := solve([x^2+y^2=b^2, (x-a)^2+y^2=c^2], [x,y])
```

[]

Ekstrak solusi y.

```
> ysol := y with sol[2][2]; $'y:=sqrt(factor(ysol^2))
```

Maxima said:
part: invalid index of list or matrix.
-- an error. To debug this try: debugmode(true);

Error in:
ysol := y with sol[2][2]; \$'y:=sqrt(factor(ysol^2)) ...
^

Kami mendapatkan rumus Heron.

```
> function H(a,b,c) := sqrt(factor((ysol*a/2)^2)); $'H(a,b,c)=H(a,b,c)
```

$$H(a,b,[1,0,4]) = \frac{a |ysol|}{2}$$

```
>$'Luas=H(2,5,6) // luas segitiga dengan panjang sisi-sisi 2, 5, 6
```

$$Luas = |y_{sol}|$$

Tentu saja, setiap segitiga siku-siku merupakan kasus yang sudah dikenal.

```
>H(3,4,5) //luas segitiga siku-siku dengan panjang sisi 3, 4, 5
```

```
Variable or function ysol not found.
Try "trace errors" to inspect local variables after errors.
H:
  useglobal; return a*abs(ysol)/2
Error in:
H(3,4,5) //luas segitiga siku-siku dengan panjang sisi 3, 4, 5 ...
  ^
```

Dan juga jelas bahwa ini adalah segitiga dengan luas maksimum dan dua sisi 3 dan 4.

```
>aspect (1.5); plot2d(&H(3,4,x),1,7): // Kurva luas segitiga sengan panjang sisi 3, 4, x (
```

```
Variable or function ysol not found.
Error in expression: 3*abs(ysol)/2
%ploteval:
  y0=f$(x[1],args());
adaptiveevalone:
  s=%ploteval(g$,t,args());
Try "trace errors" to inspect local variables after errors.
plot2d:
  dw/n,dw/n^2,dw/n,auto;args());
```

Kasus umum juga berlaku.

```
>$solve(diff(H(a,b,c)^2,c)=0,c)
```

```
Maxima said:
diff: second argument must be a variable; found [1,0,4]
-- an error. To debug this try: debugmode(true);

Error in:
$solve(diff(H(a,b,c)^2,c)=0,c) ...
  ^
```

Sekarang mari kita cari himpunan semua titik di mana $b + c = d$ untuk suatu konstanta d . Diketahui bahwa ini adalah elips.

```
>s1 &= subst(d-c,b,sol[2]); $s1
```

```

Maxima said:
part: invalid index of list or matrix.
-- an error. To debug this try: debugmode(true);

Error in:
s1 &= subst(d-c,b,sol[2]); $s1 ...
      ^

```

Dan buat fungsi dari ini.

```
>function fx(a,c,d) &= rhs(s1[1]); $fx(a,c,d), function fy(a,c,d) &= rhs(s1[2]); $fy(a,c,d)
```

Sekarang kita dapat menggambar himpunan tersebut. Sisi b bervariasi dari 1 hingga 4. Diketahui bahwa kita akan mendapatkan elips.

```
>aspect(1); plot2d(&fx(3,x,5),&fy(3,x,5),xmin=1,xmax=4,square=1):
```

Kita dapat memeriksa persamaan umum untuk elips ini, yaitu

$$\frac{(x - x_m)^2}{u^2} + \frac{(y - y_m)^2}{v^2} = 1,$$

di mana (x_m, y_m) adalah pusat, dan u serta v adalah sumbu setengah.

```
>$ratsimp((fx(a,c,d)-a/2)^2/u^2+fy(a,c,d)^2/v^2 with [u=d/2,v=sqrt(d^2-a^2)/2])
```

Kita melihat bahwa tinggi dan oleh karena itu luas segitiga mencapai nilai maksimal ketika $x=0$. Oleh karena itu, luas segitiga dengan $a+b+c=d$ mencapai nilai maksimal jika segitiga tersebut sama sisi. Kita ingin membuktikannya secara analitis.

```
>eqns &= [diff(H(a,b,d-(a+b))^2,a)=0,diff(H(a,b,d-(a+b))^2,b)=0]; $eqns
```

Kami mendapatkan beberapa nilai minimum, yang termasuk dalam segitiga dengan salah satu sisinya 0, dan solusi $a=b=c=d/3$.

```
>$solve(eqns,[a,b])
```

Ada juga metode Lagrange, yang memaksimalkan $H(a,b,c)^2$ terhadap persamaan $a+b+c=d$.

```
>&solve([diff(H(a,b,c)^2,a)=la,diff(H(a,b,c)^2,b)=la, ...
> diff(H(a,b,c)^2,c)=la,a+b+c=d],[a,b,c,la])
```

$$[a = 0, b = -\frac{d}{2}, c = -\frac{d}{2}, la = 0], [a = -\frac{d}{2}, b = 0, c = -\frac{d}{2}, la = 0],$$

$$[a = -\frac{d}{2}, b = -\frac{d}{2}, c = 0, la = 0], [a = -\frac{d}{3}, b = -\frac{d}{3}, c = -\frac{d}{3}, la = -1]$$

Kita dapat membuat plot dari situasi. /
Kita dapat membuat diagram situasi.

Pertama, atur titik-titik di Maxima.

```
>A &= at([x,y],sol[2]); $A
>B &= [0,0]; $B, C &= [a,0]; $C
```

Kemudian atur rentang plot, dan plot titik-titik tersebut.

```
>setPlotRange(0,5,-2,3); ...
>a=4; b=3; c=2; ...
>plotPoint(mxmeval("B"),"B"); plotPoint(mxmeval("C"),"C"); ...
>plotPoint(mxmeval("A"),"A");
```

Gambarlah segmen-segmen tersebut.

```
>plotSegment(mxmeval("A"),mxmeval("C")); ...
>plotSegment(mxmeval("B"),mxmeval("C")); ...
>plotSegment(mxmeval("B"),mxmeval("A"));
```

Hitung garis tegak lurus tengah di Maxima.

```
>h &= middlePerpendicular(A,B); g &= middlePerpendicular(B,C);
```

Dan pusat lingkaran.

```
>U &= lineIntersection(h,g);
```

Kita mendapatkan rumus untuk jari-jari lingkaran luar.

```
>&assume(a>0,b>0,c>0); $distance(U,B) | radcan
```

Mari kita tambahkan ini ke dalam alur cerita.

```
>plotPoint(U()); ...
>plotCircle(circleWithCenter(mxmeval("U"),mxmeval("distance(U,C)"))):
```

Menggunakan geometri, kita dapat menyimpulkan rumus sederhana

$$\frac{a}{\sin(\alpha)} = 2r$$

untuk jari-jari. Kita dapat memeriksa apakah rumus ini benar menggunakan Maxima. Maxima hanya akan memfaktorkan rumus ini jika kita mengkuadratkannya.

```
>$c^2/sin(computeAngle(A,B,C))^2 | factor
```

Contoh 4: Garis Euler dan Parabola

Garis Euler adalah garis yang ditentukan dari segitiga yang tidak sama sisi. Garis ini merupakan garis tengah segitiga dan melewati beberapa titik penting yang ditentukan dari segitiga, termasuk ortosentrum, sirkum-sentrum, pusat massa, titik Exeter, dan pusat lingkaran sembilan titik segitiga.

Untuk demonstrasi, kita menghitung dan menggambar garis Euler pada segitiga.

Pertama, kita mendefinisikan sudut-sudut segitiga dalam Euler. Kita menggunakan definisi yang terlihat dalam ekspresi simbolik.

```
>A:=[-1,-1]; B:=[2,0]; C:=[1,2];
```

Untuk menggambar objek geometris, kita menyiapkan area plot, lalu menambahkan titik-titik ke dalamnya. Semua plot objek geometris ditambahkan ke plot saat ini.

```
>setPlotRange(3); plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C");
```

Kita juga dapat menjumlahkan sisi-sisi segitiga.

```
>plotSegment(A,B,""); plotSegment(B,C,""); plotSegment(C,A,"");
```

Berikut adalah luas segitiga, menggunakan rumus determinan. Tentu saja, kita harus mengambil nilai mutlak dari hasil ini.

```
>$areaTriangle(A,B,C)
```

Kita dapat menghitung koefisien sisi c.

```
>c &= lineThrough(A,B)
```

$[-1, 3, -2]$

Dan juga dapatkan rumus untuk baris ini.

```
>$getLineEquation(c,x,y)
```

Untuk bentuk Hesse, kita perlu menentukan sebuah titik sedemikian rupa sehingga titik tersebut berada di sisi positif dari bentuk Hesse. Dengan memasukkan titik tersebut, kita mendapatkan jarak positif ke garis.

```
>$getHesseForm(c,x,y,C), $at(%, [x=C[1],y=C[2]])
```

Sekarang kita menghitung lingkaran luar segitiga ABC.

```
>LL &= circleThrough(A,B,C); $getCircleEquation(LL,x,y)
>O &= getCircleCenter(LL); $O
```

Gambarlah lingkaran dan pusatnya. Cu dan U adalah simbolik. Kita menghitung ekspresi-ekspresi ini untuk Euler.

```
>plotCircle(LL()); plotPoint(O(),"O"):
```

Kita dapat menghitung irisan ketinggian segitiga ABC (pusat tegak lurus) secara numerik dengan perintah berikut.

```
>H &= lineIntersection(perpendicular(A,lineThrough(C,B)),...
> perpendicular(B,lineThrough(A,C))); $H
```

Sekarang kita dapat menghitung garis Euler segitiga.

```
>el &= lineThrough(H,O); $getLineEquation(el,x,y)
```

Tambahkan ke plot.

```
>plotPoint(H(),"H"); plotLine(el(),"Garis Euler"):
```

Pusat gravitasi harus berada pada garis ini.

```
>M &= (A+B+C)/3; $getLineEquation(el,x,y) with [x=M[1],y=M[2]]
>plotPoint(M(),"M"): // titik berat
```

Teori mengatakan bahwa $MH = 2 \times MO$. Kita perlu menyederhanakan menggunakan radcan untuk mencapai hal ini.

```
>$distance(M,H)/distance(M,O)|radcan
```

Fungsi-fungsi tersebut juga mencakup fungsi untuk sudut.

```
>$computeAngle(A,C,B), degprint(%())
```

```
60°15'18.43''
```

Persamaan untuk pusat lingkaran dalam tidak terlalu bagus.

```
>Q &= lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A))|radcan; $Q
```

Mari kita hitung juga rumus untuk jari-jari lingkaran yang terinskripsi.

```
>r &= distance(Q,projectToLine(Q,lineThrough(A,B)))|ratsimp; $r
>LD &= circleWithCenter(Q,r); // Lingkaran dalam
```

Mari kita tambahkan ini ke plot.

```
>color(5); plotCircle(LD()):
```

Parabola

Selanjutnya akan dicari persamaan tempat kedudukan titik-titik yang berjarak sama ke titik C dan ke garis AB.

```
>p &= getHesseForm(lineThrough(A,B),x,y,C)-distance([x,y],C); $p='0
```

Persamaan tersebut dapat digambar menjadi satu dengan gambar sebelumnya.

```
>plot2d(p,level=0,add=1,contourcolor=6):
```

Ini seharusnya merupakan suatu fungsi, tetapi pemecah masalah default Maxima hanya dapat menemukan solusinya jika kita mengkuadratkan persamaan tersebut. Akibatnya, kita mendapatkan solusi palsu.

```
>akar &= solve(getHesseForm(lineThrough(A,B),x,y,C)^2-distance([x,y],C)^2,y)
```

```
[y = - 3 x - sqrt(70) sqrt(9 - 2 x) + 26, y = - 3 x + sqrt(70) sqrt(9 - 2 x) + 2
```

Solusi pertama adalah

$$akar_1$$

Menambahkan solusi pertama ke grafik menunjukkan bahwa memang itulah jalur yang kita cari. Teori mengatakan bahwa itu adalah parabola yang diputar.

```
>plot2d(&rhs(akar[1]),add=1):  
>function g(x) &= rhs(akar[1]); $'g(x)= g(x) // fungsi yang mendefinisikan kurva di atas  
>T &=[-1, g(-1)]; // ambil sebarang titik pada kurva tersebut  
>dTC &= distance(T,C); $fullratsimp(dTC), $float(%) // jarak T ke C  
>U &= projectToLine(T,lineThrough(A,B)); $U // proyeksi T pada garis AB  
>dU2AB &= distance(T,U); $fullratsimp(dU2AB), $float(%) // jarak T ke AB
```

Ternyata jarak T ke C sama dengan jarak T ke AB. Coba Anda pilih titik T yang lain dan ulangi perhitungan-perhitungan di atas untuk menunjukkan bahwa hasilnya juga sama.

Contoh 5: Trigonometri Rasional

Ini terinspirasi dari pembicaraan N.J. Wildberger. Dalam bukunya “Divine Proportions”, Wildberger mengusulkan untuk mengganti konsep klasik jarak dan sudut dengan quadrance dan spread. Dengan menggunakan konsep ini, memang mungkin untuk menghindari fungsi trigonometri dalam banyak contoh, dan tetap “rasional”.

Di bawah ini, saya memperkenalkan konsep-konsep tersebut dan menyelesaikan beberapa masalah. Saya menggunakan perhitungan simbolik Maxima di sini, yang menyembunyikan keunggulan utama trigonometri rasional bahwa perhitungan dapat dilakukan hanya dengan kertas dan pensil. Anda diundang untuk memeriksa hasilnya tanpa menggunakan komputer.

Intinya, perhitungan simbolik rasional sering menghasilkan hasil yang sederhana. Sebaliknya, trigonometri klasik menghasilkan hasil trigonometri yang rumit, yang hanya dievaluasi sebagai perkiraan numerik.

```
>load geometry;
```

Untuk pengenalan awal, kita menggunakan segitiga siku-siku dengan proporsi Mesir yang terkenal 3, 4, dan 5. Perintah-perintah berikut adalah perintah Euler untuk menggambar geometri datar yang terdapat dalam berkas Euler “geometry.e”.

```
>C&:=[0,0]; A&:=[4,0]; B&:=[0,3]; ...  
>setPlotRange(-1,5,-1,5); ...  
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...  
>plotSegment(B,A,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...  
>insimg(30);
```

Tentu saja,

$$\sin(w_a) = \frac{a}{c},$$

di mana w_a adalah sudut di titik A. Cara biasa untuk menghitung sudut ini adalah dengan mengambil invers fungsi sinus. Hasilnya adalah sudut yang tidak dapat dicetak secara tepat, sehingga hanya dapat dicetak secara aproksimatif.

```
>wa := arcsin(3/5); degprint(wa)
```

```
36°52'11.63''
```

Trigonometri rasional berusaha menghindari hal ini.

Konsep pertama dalam trigonometri rasional adalah kuadran, yang menggantikan jarak. Sebenarnya, kuadran hanyalah kuadrat dari jarak. Dalam pembahasan berikut, a , b , dan c mewakili kuadran dari sisi-sisi.

Teorema Pythagoras menjadi $a + b = c$.

```
>a &= 3^2; b &= 4^2; c &= 5^2; &a+b=c
```

```
25 = 25
```

Konsep kedua dari trigonometri rasional adalah selisih. Selisih mengukur jarak antara dua garis. Nilainya 0 jika garis-garis tersebut sejajar, dan 1 jika garis-garis tersebut tegak lurus. Selisih ini merupakan kuadrat dari sinus sudut antara dua garis tersebut.

Spread garis AB dan AC pada gambar di atas didefinisikan sebagai

$$s_a = \sin(\alpha)^2 = \frac{a}{c},$$

di mana a dan c adalah sisi-sisi segitiga siku-siku dengan salah satu sudutnya di A.

```
>sa &= a/c; $sa
```

Tentu saja, ini lebih mudah dihitung daripada sudut. Namun, Anda kehilangan sifat bahwa sudut dapat ditambahkan dengan mudah.

Tentu saja, kita dapat mengubah nilai perkiraan sudut w_a menjadi sprad, dan mencetaknya sebagai pecahan.

```
>fracprint(sin(wa)^2)
```

```
9/25
```

Hukum kosinus dalam trigonometri klasik dapat diartikan sebagai “hukum silang” berikut.

$$(c + b - a)^2 = 4bc(1 - s_a)$$

Di sini, a , b , dan c adalah kuadran sisi-sisi segitiga, dan s_a adalah sudut di sudut A. Sisi a , seperti biasa, berhadapan dengan sudut A.

Hukum-hukum ini diimplementasikan dalam berkas `geometry.e` yang telah dimuat ke dalam Euler.

```
>$crosslaw(aa,bb,cc,saa)
```

Di kasus ini, kita mendapatkan

```
>$crosslaw(a,b,c,sa)
```

Mari kita gunakan hukum silang ini untuk menemukan selisih di titik A. Untuk melakukannya, kita menghitung hukum silang untuk kuadran a, b, dan c, lalu menyelesaikan persamaan tersebut untuk menemukan selisih yang tidak diketahui sa.

Anda dapat melakukannya secara manual dengan mudah, tetapi saya menggunakan Maxima. Tentu saja, hasil yang kita dapatkan sama dengan yang sudah kita miliki sebelumnya.

```
>$crosslaw(a,b,c,x), $solve(%,x)
```

Kami sudah tahu hal ini. Definisi selisih sudut merupakan kasus khusus dari hukum silang.

Kami juga dapat menyelesaikan ini untuk a, b, c yang umum. Hasilnya adalah rumus yang menghitung selisih sudut segitiga berdasarkan kuadran ketiga sisinya.

```
>$solve(crosslaw(aa,bb,cc,x),x)
```

Kita dapat membuat fungsi dari hasil tersebut. Fungsi semacam itu sudah didefinisikan dalam berkas geometry.e dari Euler.

```
>$spread(a,b,c)
```

Sebagai contoh, kita dapat menggunakannya untuk menghitung sudut segitiga dengan sisi

$$a, \quad a, \quad \frac{4a}{7}$$

Hasilnya adalah bilangan rasional, yang tidak begitu mudah didapatkan jika kita menggunakan trigonometri klasik.

```
>$spread(a,a,4*a/7)
```

Ini adalah sudut dalam derajat.

```
>degprint(arcsin(sqrt(6/7)))
```

67°47'32.44''

Contoh Lain

Sekarang, mari kita coba contoh yang lebih lanjut.
Kita tetapkan tiga sudut segitiga sebagai berikut.

```
>A&:=[1,2]; B&:=[4,3]; C&:=[0,4]; ...
>setPlotRange(-1,5,1,7); ...
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...
>plotSegment(B,A,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...
>insimg;
```

Dengan menggunakan Pythagoras, mudah untuk menghitung jarak antara dua titik. Pertama, saya menggunakan fungsi distance dari berkas Euler untuk geometri. Fungsi distance menggunakan geometri klasik.

```
>$distance(A,B)
```

Euler juga menyediakan fungsi untuk menghitung kuadrat antara dua titik.
Dalam contoh berikut, karena c+b bukan a, segitiga tersebut bukan segitiga siku-siku.

```
>c &= quad(A,B); $c, b &= quad(A,C); $b, a &= quad(B,C); $a,
```

Pertama, mari kita hitung sudut tradisional. Fungsi computeAngle menggunakan metode biasa berdasarkan hasil kali dot dua vektor. Hasilnya adalah perkiraan bilangan floating point.

$$A = \langle 1, 2 \rangle \quad B = \langle 4, 3 \rangle, \quad C = \langle 0, 4 \rangle$$

$$\mathbf{a} = C - B = \langle -4, 1 \rangle, \quad \mathbf{c} = A - B = \langle -3, -1 \rangle, \quad \beta = \angle ABC$$

$$\mathbf{a} \cdot \mathbf{c} = |\mathbf{a}| \cdot |\mathbf{c}| \cos \beta$$

$$\cos \angle ABC = \cos \beta = \frac{\mathbf{a} \cdot \mathbf{c}}{|\mathbf{a}| \cdot |\mathbf{c}|} = \frac{12 - 1}{\sqrt{17}\sqrt{10}} = \frac{11}{\sqrt{17}\sqrt{10}}$$

```
>wb &= computeAngle(A,B,C); $wb, $(wb/pi*180)()
```

```
32.4711922908
```

Dengan menggunakan pensil dan kertas, kita dapat melakukan hal yang sama dengan hukum silang. Kita memasukkan kuadran a, b, dan c ke dalam hukum silang dan menyelesaikan persamaan untuk x.

```
>$crosslaw(a,b,c,x), $solve(%,x), //(b+c-a)^=4b.c(1-x)
```

Artinya, apa yang dilakukan oleh fungsi spread yang didefinisikan dalam “geometry.e”.

```
>sb &= spread(b,a,c); $sb
```

Maxima mendapatkan hasil yang sama menggunakan trigonometri biasa, jika kita memaksanya. Program ini memang menyelesaikan istilah $\sin(\arccos(\dots))$ menjadi hasil pecahan. Kebanyakan siswa tidak dapat melakukan hal ini.

```
>$sin(computeAngle(A,B,C))^2
```

Setelah kita memiliki lebar di B, kita dapat menghitung tinggi h_a di sisi a. Ingat bahwa

$$s_b = \frac{h_a}{c}$$

secara definisi.

```
>ha &= c*sb; $ha
```

Gambar berikut dihasilkan menggunakan program geometri C.a.R., yang dapat menggambar kuadran dan penyebaran.

gambar: (20) Rational_Geometry_CaR.png

Secara definisi, panjang h_a adalah akar kuadrat dari kuadrannya.

```
>$sqrt(ha)
```

Sekarang kita dapat menghitung luas segitiga. Jangan lupa, kita sedang berurusan dengan kuadran!

```
>$sqrt(ha)*sqrt(a)/2
```

Rumus determinan biasa memberikan hasil yang sama.

```
>$areaTriangle(B,A,C)
```

Rumus Heron

Sekarang, mari kita selesaikan masalah ini secara umum!

```
>&remvalue(a,b,c,sb,ha);
```

Kami terlebih dahulu menghitung selisih di titik B untuk segitiga dengan sisi a, b, dan c. Kemudian kami menghitung luas kuadrat (yang disebut “quadrea”), memfaktorkan hasilnya menggunakan Maxima, dan mendapatkan rumus terkenal Heron.

Jujur saja, ini sulit dilakukan dengan pensil dan kertas.

```
>$spread(b^2,c^2,a^2), $factor(%*c^2*a^2/4)
```

Aturan Tiga Sudut

Kekurangan dari sudut-sudut adalah bahwa mereka tidak lagi hanya ditambahkan seperti sudut-sudut. Namun, ketiga sudut segitiga memuaskan aturan “tiga sudut” berikut ini.

```
>&remvalue(sa,sb,sc); $triplespread(sa,sb,sc)
```

Aturan ini berlaku untuk tiga sudut mana pun yang jumlahnya 180° .

$$\alpha + \beta + \gamma = \pi$$

Karena selisih antara

$$\alpha, \pi - \alpha$$

sama, aturan selisih tiga sudut juga berlaku, jika

$$\alpha + \beta = \gamma$$

Karena selisih sudut negatif sama, aturan selisih tiga sudut juga berlaku jika

$$\alpha + \beta + \gamma = 0$$

Misalnya, kita dapat menghitung selisih sudut 60° . Nilainya adalah $3/4$. Persamaan tersebut memiliki solusi kedua, di mana semua selisih sudut sama dengan 0.

```
>$solve(triplespread(x,x,x),x)
```

Jarak antara dua sudut yang jumlahnya 90° jelas sama dengan 1. Jika dua sudut jumlahnya 90° , jarak antara keduanya memenuhi persamaan jarak tiga sudut dengan a, b, dan 1. Dengan perhitungan berikut, kita mendapatkan $a + b = 1$.

```
>$triplespread(x,y,1), $solve(%,x)
```


Karena selisih $180^\circ - t$ sama dengan selisih t , rumus selisih tiga sudut juga berlaku, jika salah satu sudut merupakan penjumlahan atau pengurangan dari dua sudut lainnya.

Jadi, kita dapat menemukan selisih sudut yang digandakan. Perhatikan bahwa ada dua solusi lagi. Kita jadikan ini sebagai fungsi.

```
>$solve(triplespread(a,a,x),x), function doublespread(a) &= factor(rhs(%[1]))
```

$$- 4 (a - 1) a$$

Pembagi Sudut

Inilah situasinya, kita sudah tahu.

```
>C&:=[0,0]; A&:=[4,0]; B&:=[0,3]; ...
>setPlotRange(-1,5,-1,5); ...
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...
>plotSegment(B,A,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...
>insimg;
```

Mari kita hitung panjang pembagi sudut di titik A. Namun, kita ingin menyelesaikan persamaan tersebut untuk nilai a , b , dan c yang umum.

```
>&remvalue(a,b,c);
```

Jadi, kita terlebih dahulu menghitung selisih sudut yang dibagi dua di titik A menggunakan rumus selisih tiga.

Masalah dengan rumus ini muncul lagi. Rumus ini memiliki dua solusi. Kita harus memilih solusi yang benar. Solusi lainnya merujuk pada sudut yang dibagi dua 180° -wa.

```
>$triplespread(x,x,a/(a+b)), $solve(%,x), sa2 &= rhs(%[1]); $sa2
```

Mari kita periksa persegi panjang Mesir.

```
>$sa2 with [a=3^2,b=4^2]
```

Kita dapat mencetak sudut dalam bentuk Euler, setelah mengubah sudut tersebut ke dalam radian.

```
>wa2 := arcsin(sqrt(1/10)); degprint(wa2)
```

$18^\circ 26' 5.82''$

Titik P adalah titik potong antara garis pembagi sudut dengan sumbu y.

```
>P := [0,tan(wa2)*4]
```

```
[0, 1.33333]
```

```
>plotPoint(P,"P"); plotSegment(A,P):
```

Mari kita periksa sudut-sudut dalam contoh spesifik kita.

```
>computeAngle(C,A,P), computeAngle(P,A,B)
```

```
0.321750554397  
0.321750554397
```

Sekarang kita menghitung panjang bisektor AP.

Kita menggunakan teorema sinus pada segitiga APC. Teorema ini menyatakan bahwa

$$\frac{BC}{\sin(w_a)} = \frac{AC}{\sin(w_b)} = \frac{AB}{\sin(w_c)}$$

berlaku pada segitiga mana pun. Kuadratkan, maka menjadi yang disebut “hukum penyebaran”

$$\frac{a}{s_a} = \frac{b}{s_b} = \frac{c}{s_b}$$

di mana a, b, c mewakili kuadran.

Karena spread CPA adalah $1-sa^2$, kita mendapatkan dari itu $bisa/1 = b/(1-sa^2)$ dan dapat menghitung bisa (kuadran dari garis tengah sudut).

```
>&factor(ratsimp(b/(1-sa2))); bisa &= %; $bisa
```

Mari kita periksa rumus ini untuk nilai-nilai Mesir kita.

```
>sqrt(mxmeval("at(bisa,[a=3^2,b=4^2])"), distance(A,P))
```

```
4.21637021356  
4.21637021356
```

Kita juga dapat menghitung P menggunakan rumus penyebaran.

```
>py&=factor(ratsimp(sa2*bisa)); $py
```

Nilai tersebut sama dengan yang kita dapatkan menggunakan rumus trigonometri.

```
>sqrt(mxmeval("at(py,[a=3^2,b=4^2])"))
```

```
1.33333333333
```

Sudut Akord

Perhatikan situasi berikut ini.

```
>setPlotRange(1.2); ...
>color(1); plotCircle(circleWithCenter([0,0],1)); ...
>A:=[cos(1),sin(1)]; B:=[cos(2),sin(2)]; C:=[cos(6),sin(6)]; ...
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...
>color(3); plotSegment(A,B,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...
>color(1); O:=[0,0]; plotPoint(O,"O"); ...
>plotSegment(A,O); plotSegment(B,O); plotSegment(C,O,"r"); ...
>insimg;
```

Kita dapat menggunakan Maxima untuk menyelesaikan rumus penyebaran tiga sudut di pusat O untuk r . Dengan demikian, kita mendapatkan rumus untuk jari-jari kuadrat lingkaran periklim dalam terms kuadran sisi-sisinya.

Kali ini, Maxima menghasilkan beberapa nol kompleks, yang kita abaikan.

```
>&remvalue(a,b,c,r); // hapus nilai-nilai sebelumnya untuk perhitungan baru
>rabc &= rhs(solve(triplespread(spread(b,r,r),spread(a,r,r),spread(c,r,r)),r)[4]); $rabc
```

Kita dapat menjadikannya sebagai fungsi Euler.

```
>function periradius(a,b,c) &= rabc;
```

Mari kita periksa hasilnya untuk titik-titik A, B, dan C.

```
>a:=quadrance(B,C); b:=quadrance(A,C); c:=quadrance(A,B);
```

Jari-jari memang 1.

```
>periradius(a,b,c)
```

```
1
```

Faktanya, selisih CBA hanya bergantung pada b dan c. Ini adalah teorema sudut tali.

```
>$spread(b,a,c)*rabc | ratsimp
```

Sebenarnya, selisihnya adalah $b/(4r)$, dan kita lihat bahwa sudut tali busur b adalah setengah dari sudut pusat.

```
>$doublespread(b/(4*r))-spread(b,r,r) | ratsimp
```

Contoh 6: Jarak Minimal pada Bidang

Catatan pendahuluan

Fungsi yang, untuk suatu titik M di bidang, menetapkan jarak AM antara titik tetap A dan M , memiliki garis tingkat yang cukup sederhana: lingkaran berpusat di A .

```
>&remvalue();  
>A=[-1,-1];  
>function d1(x,y):=sqrt((x-A[1])^2+(y-A[2])^2)  
>fcontour("d1",xmin=-2,xmax=0,ymin=-2,ymax=0,hue=1,...  
>title="If you see ellipses, please set your window square"):
```

dan grafiknya juga cukup sederhana: bagian atas kerucut:

```
>plot3d("d1",xmin=-2,xmax=0,ymin=-2,ymax=0):
```

Tentu saja nilai minimum 0 tercapai di A .

Dua titik

Sekarang kita melihat fungsi $MA+MB$ di mana A dan B adalah dua titik (tetap). Merupakan “fakta yang sudah diketahui” bahwa kurva tingkatnya adalah elips, dengan titik fokus A dan B ; kecuali untuk minimum AB yang konstan pada segmen $[AB]$:

```
>B=[1,-1];  
>function d2(x,y):=d1(x,y)+sqrt((x-B[1])^2+(y-B[2])^2)  
>fcontour("d2",xmin=-2,xmax=2,ymin=-3,ymax=1,hue=1):
```

Grafiknya lebih menarik:

```
>plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):
```

Pembatasan pada garis (AB) lebih terkenal:

```
>plot2d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3):
```

Tiga titik

Sekarang hal-hal menjadi lebih rumit: Tidak banyak yang tahu bahwa $MA+MB+MC$ mencapai minimumnya di satu titik pada bidang, tetapi untuk menentukannya tidak semudah itu:

1) Jika salah satu sudut segitiga ABC lebih dari 120° (misalnya di A), maka minimumnya tercapai di titik tersebut (misalnya $AB+AC$).

Contoh:

```
>C=[-4,1];
>function d3(x,y):=d2(x,y)+sqrt((x-C[1])^2+(y-C[2])^2)
>plot3d("d3",xmin=-5,xmax=3,ymin=-4,ymax=4);
>insimg;
>fcontour("d3",xmin=-4,xmax=1,ymin=-2,ymax=2,hue=1,title="The minimum is on A");
>P=(A_B_C_A)'; plot2d(P[1],P[2],add=1,color=12);
>insimg;
```

2) Namun, jika semua sudut segitiga ABC kurang dari 120° , titik minimum terletak pada titik F di dalam segitiga, yang merupakan satu-satunya titik yang melihat sisi-sisi ABC dengan sudut yang sama (masing-masing 120°):

```
>C=[-0.5,1];
>plot3d("d3",xmin=-2,xmax=2,ymin=-2,ymax=2):
>fcontour("d3",xmin=-2,xmax=2,ymin=-2,ymax=2,hue=1,title="The Fermat point");
>P=(A_B_C_A)'; plot2d(P[1],P[2],add=1,color=12);
>insimg;
```

It is an interesting activity to realize the above figure with a geometry software; for example, I know a soft written in Java which has a "contour lines" instruction...

All of this above have been discovered by a french judge called Pierre de Fermat; he wrote letters to other dilettants like the priest Marin Mersenne and Blaise Pascal who worked at the income taxes. So the unique point F such that $FA+FB+FC$ is minimal, is called the Fermat point of the triangle. But it seems that a few years before, the italian Torricelli had found this point before Fermat did! Anyway the tradition is to note this point F...

Four points

The next step is to add a 4th point D and try and minimize $MA+MB+MC+MD$; say that you are a cable TV operator and want to find in which field you must put your antenna so that you can feed four villages and use as little cable length as possible!

```
>D=[1,1];
>function d4(x,y):=d3(x,y)+sqrt((x-D[1])^2+(y-D[2])^2)
>plot3d("d4",xmin=-1.5,xmax=1.5,ymin=-1.5,ymax=1.5):
>fcontour("d4",xmin=-1.5,xmax=1.5,ymin=-1.5,ymax=1.5,hue=1);
>P=(A_B_C_D)'; plot2d(P[1],P[2],points=1,add=1,color=12);
>insimg;
```

There is still a minimum and it is attained at none of the vertices A, B, C nor D:

```
>function f(x):=d4(x[1],x[2])
>neldermin("f",[0.2,0.2])
```

```
[0.142858, 0.142857]
```

It seems that in this case, the coordinates of the optimal point are rational or near-rational...

Now ABCD is a square we expect that the optimal point will be the center of ABCD:

```
>C=[-1,1];
>plot3d("d4",xmin=-1,xmax=1,ymin=-1,ymax=1):
>fcontour("d4",xmin=-1.5,xmax=1.5,ymin=-1.5,ymax=1.5,hue=1);
>P=(A_B_C_D)'; plot2d(P[1],P[2],add=1,color=12,points=1);
>insimg;
```

Contoh 7: Bola Dandelin dengan Povray

You can run this demonstration, if you have Povray installed, and pvengine.exe in the program path.

First we compute the radii of the spheres.

If you look at the figure below, you see that we need two circles touching the two lines which form the cone, and one line which forms the plane cutting the cone.

We use the geometry.e file of Euler for this.

```
>load geometry;
```

First the two lines forming the cone.

```
>g1 &= lineThrough([0,0],[1,a])
```

```
[- a, 1, 0]
```

```
>g2 &= lineThrough([0,0],[-1,a])
```

```
[- a, - 1, 0]
```

Thenm a third line.

```
>g &= lineThrough([-1,0],[1,1])
```

$[-1, 2, 1]$

We plot everything so far.

```
>setPlotRange(-1,1,0,2);  
>color(black); plotLine(g(),"")  
>a:=2; color(blue); plotLine(g1(),""), plotLine(g2(),""):
```

Now we take a general point on the y-axis.

```
>P &= [0,u]
```

$[0, u]$

Compute the distance to g1.

```
>d1 &= distance(P,projectToLine(P,g1)); $d1
```

Compute the distance to g.

```
>d &= distance(P,projectToLine(P,g)); $d
```

And find the centers of the two circles, where the distances are equal.

```
>sol &= solve(d1^2=d^2,u); $sol
```

There are two solutions.

We evaluate the symbolic solutions, and find both centers, and both distances.

```
>u := sol()
```

$[0.333333, \quad 1]$

```
>dd := d()
```

$[0.149071, \quad 0.447214]$

Plot the circles into the figure.

```
>color(red);  
>plotCircle(circleWithCenter([0,u[1]],dd[1]), "");  
>plotCircle(circleWithCenter([0,u[2]],dd[2]), "");  
>insimg;
```

Plot with Povray

Next we plot everything with Povray. Note that you change any command in the following sequence of Povray commands, and rerun all commands with Shift-Return.

First we load the povray functions.

```
>load povray;  
>defaultpovray="C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe"
```

C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe

We setup the scene appropriately.

```
>povstart (zoom=11,center=[0,0,0.5],height=10°,angle=140°);
```

Next we write the two spheres to the Povray file.

```
>writeln(povsphere([0,0,u[1]],dd[1],povlook(red)));  
>writeln(povsphere([0,0,u[2]],dd[2],povlook(red)));
```

And the cone, transparent.

```
>writeln(povcone([0,0,0],0,[0,0,a],1,povlook(lightgray,1)));
```

We generate a plane restricted to the cone.

```
>gp=g();  
>pc=povcone([0,0,0],0,[0,0,a],1,"");  
>vp=[gp[1],0,gp[2]]; dp=gp[3];  
>writeln(povplane(vp,dp,povlook(blue,0.5),pc));
```

Now we generate two points on the circles, where the spheres touch the cone.

```
>function turnz(v) := return [-v[2],v[1],v[3]]  
>P1=projectToLine([0,u[1]],g1()); P1=turnz([P1[1],0,P1[2]]);  
>writeln(povpoint(P1,povlook(yellow)));  
>P2=projectToLine([0,u[2]],g1()); P2=turnz([P2[1],0,P2[2]]);  
>writeln(povpoint(P2,povlook(yellow)));
```


Then we generate the two points where the spheres touch the plane. These are the foci of the ellipse.

```
>P3=projectToLine([0,u[1]],g()); P3=[P3[1],0,P3[2]];
>writeln(povpoint(P3,povlook(yellow)));
>P4=projectToLine([0,u[2]],g()); P4=[P4[1],0,P4[2]];
>writeln(povpoint(P4,povlook(yellow)));
```

Next we compute the intersection of P1P2 with the plane.

```
>t1=scalp(vp,P1)-dp; t2=scalp(vp,P2)-dp; P5=P1+t1/(t1-t2)*(P2-P1);
>writeln(povpoint(P5,povlook(yellow)));
```

We connect the points with line segments.

```
>writeln(povsegment(P1,P2,povlook(yellow)));
>writeln(povsegment(P5,P3,povlook(yellow)));
>writeln(povsegment(P5,P4,povlook(yellow)));
```

Now we generate a gray band, where the spheres touch the cone.

```
>pcw=povcone([0,0,0],0,[0,0,a],1.01);
>pc1=povcylinder([0,0,P1[3]-defaultpointsize/2],[0,0,P1[3]+defaultpointsize/2],1);
>writeln(povintersection([pcw,pc1],povlook(gray)));
>pc2=povcylinder([0,0,P2[3]-defaultpointsize/2],[0,0,P2[3]+defaultpointsize/2],1);
>writeln(povintersection([pcw,pc2],povlook(gray)));
```

Start the Povray program.

```
>povend();
```

To get an Anaglyph of this we need to put everything into a scene function. This function will be used twice later.

```
>function scene () ...

global a,u,dd,g,g1,defaultpointsize;
writeln(povsphere([0,0,u[1]],dd[1],povlook(red)));
writeln(povsphere([0,0,u[2]],dd[2],povlook(red)));
writeln(povcone([0,0,0],0,[0,0,a],1,povlook(lightgray,1)));
gp=g();
pc=povcone([0,0,0],0,[0,0,a],1,"");
vp=[gp[1],0,gp[2]]; dp=gp[3];
writeln(povplane(vp,dp,povlook(blue,0.5),pc));
P1=projectToLine([0,u[1]],g1()); P1=turnz([P1[1],0,P1[2]]);
writeln(povpoint(P1,povlook(yellow)));
```

```

P2=projectToLine([0,u[2]],g1()); P2=turnz([P2[1],0,P2[2]]);
writeln(povpoint(P2,povlook(yellow)));
P3=projectToLine([0,u[1]],g()); P3=[P3[1],0,P3[2]];
writeln(povpoint(P3,povlook(yellow)));
P4=projectToLine([0,u[2]],g()); P4=[P4[1],0,P4[2]];
writeln(povpoint(P4,povlook(yellow)));
t1=scalp(vp,P1)-dp; t2=scalp(vp,P2)-dp; P5=P1+t1/(t1-t2)*(P2-P1);
writeln(povpoint(P5,povlook(yellow)));
writeln(povsegment(P1,P2,povlook(yellow)));
writeln(povsegment(P5,P3,povlook(yellow)));
writeln(povsegment(P5,P4,povlook(yellow)));
pcw=povcone([0,0,0],0,[0,0,a],1.01);
pcl=povcylinder([0,0,P1[3]-defaultpointsize/2],[0,0,P1[3]+defaultpointsize/2],1);
writeln(povintersection([pcw,pcl],povlook(gray)));
pc2=povcylinder([0,0,P2[3]-defaultpointsize/2],[0,0,P2[3]+defaultpointsize/2],1);
writeln(povintersection([pcw,pc2],povlook(gray)));
endfunction

```

You need red/cyan glasses to appreciate the following effect.

```
>povanaglyph("scene",zoom=11,center=[0,0,0.5],height=10°,angle=140°);
```

Contoh 8: Geometri Bumi

In this notebook, we want to do some spherical computations. The functions are contained in the file "spherical.e" in the examples folder. We need to load that file first.

```
>load "spherical.e";
```

To enter a geographical position, we use a vector with two coordinates in radians (north and east, negative values for south and west). The following are the coordinates for the Campus of the FMIPA UNY.

```
>FMIPA=[rad(-7,-46.467),rad(110,23.05)]
```

```
[-0.13569, 1.92657]
```

You can print this position with `sposprint` (spherical position print).

```
>sposprint(FMIPA) // posisi garis lintang dan garis bujur FMIPA UNY
```

```
S 7°46.467' E 110°23.050'
```

Let us add two more towns, Solo and Semarang.

```
>Solo=[rad(-7,-34.333),rad(110,49.683)]; Semarang=[rad(-6,-59.05),rad(110,24.533)];
>sposprint(Solo), sposprint(Semarang),
```

```
S 7°34.333' E 110°49.683'
S 6°59.050' E 110°24.533'
```

First we compute the vector from one to the other on an ideal ball. This vector is [heading,distance] in radians. To compute the distance on the earth, we multiply with the earth radius at a latitude of 7° .

```
>br=svector(FMIPA,Solo); degprint(br[1]), br[2]*rearth(7°)->km // perkiraan jarak FMIPA-Solo
```

```
65°20'26.60''  
53.8945384608
```

This is a good approximation. The following routines use even better approximations. On such a short distance the result is almost the same.

```
>esdist(FMIPA,Semarang)->" km" // perkiraan jarak FMIPA-Semarang
```

```
88.0114026318 km
```

There is a function for the heading, taking the elliptical shape of the earth into account. Again, we print in an advanced way.

```
>sdegprint(esdir(FMIPA,Solo))
```

```
65.34°
```

The angle of a triangle exceeds 180° on the sphere.

```
>asum=sangle(Solo,FMIPA,Semarang)+sangle(FMIPA,Solo,Semarang)+sangle(FMIPA,Semarang,Solo);
```

```
180°0'10.77''
```

This can be used to compute the area of the triangle. Note: For small triangles, this is not accurate due to the subtraction error in $\text{asum}-\pi$.

```
>(asum-pi)*rearth(48°)^2->" km^2" // perkiraan luas segitiga FMIPA-Solo-Semarang
```

```
2123.67056312 km^2
```

There is a function for this, which uses the mean latitude of the triangle to compute the earth radius, and takes care of rounding errors for very small triangles.

```
>esarea(Solo,FMIPA,Semarang)->" km^2", //perkiraan yang sama dengan fungsi esarea()
```

```
2123.64310526 km^2
```

We can also add vectors to positions. A vector contains the heading and the distance, both in radians. To get a vector, we use `svector`. To add a vector to a position, we use `saddvector`.

```
>v=svector(FMIPA,Solo); sposprint(saddvector(FMIPA,v)), sposprint(Solo),
```

```
S 7°34.333' E 110°49.683'  
S 7°34.333' E 110°49.683'
```

These functions assume an ideal ball. The same on the earth.

```
>sposprint (esadd(FMIPA,esdir(FMIPA,Solo),esdist(FMIPA,Solo))), sposprint(Solo),
```

```
S 7°34.333' E 110°49.683'  
S 7°34.333' E 110°49.683'
```

Let us turn to a larger example, Tugu Jogja dan Monas Jakarta (menggunakan Google Earth untuk mencari koordinatnya).

```
>Tugu=[-7.7833°,110.3661°]; Monas=[-6.175°,106.811944°];  
>sposprint(Tugu), sposprint(Monas)
```

```
S 7°46.998' E 110°21.966'  
S 6°10.500' E 106°48.717'
```

According to Google Earth, the distance is 429.66km. We get a good approximation.

```
>esdist(Tugu,Monas)->" km" // perkiraan jarak Tugu Jogja - Monas Jakarta
```

```
431.565659488 km
```

The heading is the same as the one computed in Google Earth.

```
>degprint(esdir(Tugu,Monas))
```

```
294°17'2.85''
```

However, we do no longer get the exact target position, if we add the heading and distance to the original position. This is so, since we do not compute the inverse function exactly, but take an approximation of the earth radius along the path.

```
>sposprint(esadd(Tugu,esdir(Tugu,Monas),esdist(Tugu,Monas)))
```

```
S 6°10.500' E 106°48.717'
```

The error is not large, however.

```
>sposprint(Monas),
```

```
S 6°10.500' E 106°48.717'
```

Of course, we cannot sail with the same heading from one destination to another, if we want to take the shortest path. Imagine, you fly NE starting at any point on the earth. Then you will spiral to the north pole. Great circles do not follow a constant heading!

The following computation shows that we are way off the correct destination, if we use the same heading during our travel.

```
>dist=esdist(Tugu,Monas); hd=esdir(Tugu,Monas);
```

Now we add 10 times one-tenth of the distance, using the heading to Monas, we got in Tugu.

```
>p=Tugu; loop 1 to 10; p=esadd(p,hd,dist/10); end;
```

The result is far off.

```
>sposprint(p), skmprint(esdist(p,Monas))
```

```
S 6°11.250' E 106°48.372'  
1.529km
```

As another example, let us take two points on the earth at the same latitude.

```
>P1=[30°,10°]; P2=[30°,50°];
```

The shortest path from P1 to P2 is not the circle of latitude 30°, but a shorter path starting 10° further north at P1.

```
>sdegprint(esdir(P1,P2))
```

```
79.69°
```

But, if we follow this compass reading, we will spiral to the north pole! So we must adjust our heading along the way. For rough purposes, we adjust it at 1/10 of the total distance.

```
>p=P1; dist=esdist(P1,P2); ...  
> loop 1 to 10; dir=esdir(p,P2); sdegprint(dir), p=esadd(p,dir,dist/10); end;
```

```
79.69°  
81.67°  
83.71°  
85.78°  
87.89°  
90.00°  
92.12°  
94.22°  
96.29°  
98.33°
```

The distances are not right, since we will add a bit off error, if we follow the same heading for too long.

```
>skmprint(esdist(p,P2))
```

```
0.203km
```

We get a good approximation, if we adjust our heading after each 1/100 of the total distance from Tugu to Monas.

```
>p=Tugu; dist=esdist(Tugu,Monas); ...  
> loop 1 to 100; p=esadd(p,esdir(p,Monas),dist/100); end;  
>skmprint(esdist(p,Monas))
```

0.000km

For navigational purposes, we can get a sequence of GPS position along the great circle to Monas with the function `navigate`.

```
>load spherical; v=navigate(Tugu,Monas,10); ...  
> loop 1 to rows(v); sposprint(v[#]), end;
```

```
S 7°46.998' E 110°21.966'  
S 7°37.422' E 110°0.573'  
S 7°27.829' E 109°39.196'  
S 7°18.219' E 109°17.834'  
S 7°8.592' E 108°56.488'  
S 6°58.948' E 108°35.157'  
S 6°49.289' E 108°13.841'  
S 6°39.614' E 107°52.539'  
S 6°29.924' E 107°31.251'  
S 6°20.219' E 107°9.977'  
S 6°10.500' E 106°48.717'
```

We write a function, which plots the earth, the two positions, and the positions in between.

```
>function testplot ...  
  
    useglobal;  
    plotearth;  
    plotpos(Tugu,"Tugu Jogja"); plotpos(Monas,"Tugu Monas");  
    plotposline(v);  
endfunction
```

Now plot everything.

```
>plot3d("testplot",angle=25, height=6,>own,>user, zoom=4) :
```

Or use `plot3d` to get an anaglyph view of it. This looks really great with red/cyan glasses.

```
>plot3d("testplot",angle=25,height=6,distance=5,own=1,anaglyph=1, zoom=4) :
```

1. Gambarlah segi-n beraturan jika diketahui titik pusat O, n, dan jarak titik pusat ke titik-titik sudut segi-n tersebut (jari-jari lingkaran luar segi-n), r.

Petunjuk:

- Besar sudut pusat yang menghadap masing-masing sisi segi-n adalah $(360/n)$.
- Titik-titik sudut segi-n merupakan perpotongan lingkaran luar segi-n dan garis-garis yang melalui pusat dan saling membentuk sudut sebesar kelipatan $(360/n)$.
- Untuk n ganjil, pilih salah satu titik sudut adalah di atas.
- Untuk n genap, pilih 2 titik di kanan dan kiri lurus dengan titik pusat.
- Anda dapat menggambar segi-3, 4, 5, 6, 7, dst beraturan.

2. Gambarlah suatu parabola yang melalui 3 titik yang diketahui.

Petunjuk:

- Misalkan persamaan parabolanya $y = ax^2 + bx + c$.
- Substitusikan koordinat titik-titik yang diketahui ke persamaan tersebut.
- Selesaikan SPL yang terbentuk untuk mendapatkan nilai-nilai a, b, c.

3. Gambarlah suatu segi-4 yang diketahui keempat titik sudutnya, misalnya A, B, C, D.

- Tentukan apakah segi-4 tersebut merupakan segi-4 garis singgung (sisinya-sisintya

merupakan garis singgung lingkaran yang sama yakni lingkaran dalam segi-4 tersebut).

- Suatu segi-4 merupakan segi-4 garis singgung apabila keempat garis bagi sudutnya

bertemu di satu titik.

- Jika segi-4 tersebut merupakan segi-4 garis singgung, gambar lingkaran dalamnya.
- Tunjukkan bahwa syarat suatu segi-4 merupakan segi-4 garis singgung apabila hasil kal

panjang sisi-sisi yang berhadapan sama.

4. Gambarlah suatu ellips jika diketahui kedua titik fokusnya, misalnya P dan Q. Ingat ellips dengan fokus P dan Q adalah tempat kedudukan titik-titik yang jumlah jarak ke P dan ke Q selalu sama (konstan).

5. Gambarlah suatu hiperbola jika diketahui kedua titik fokusnya, misalnya P dan Q. Ingat ellips dengan fokus P dan Q adalah tempat kedudukan titik-titik yang selisih jarak ke P dan ke Q selalu sama (konstan).