EMT plot2D

Rasyid Salahuddin
22305144016
Matematika E

Sub Bab 1

Menggambar Grafik Fungsi Satu Variabel dalam Bentuk Ekspresi Langsung Ekspresi tunggal

Di dalam program numerik EMT, ekspresi adalah string. Jika ditandai sebagai simbolis, mereka akan mencetak melalui Maxima, jika tidak melalui EMT. Ekspresi dalam string digunakan untuk membuat plot dan banyak fungsi numerik. Untuk ini, variabel dalam ekspresi harus "x".

expresi dalam string

```
>expr := "x^5-x^2-3"
```

```
     x^5-x^2-3
```

plot ekspresi

```
>plot2d(expr,-2,2) :
```

contoh 1

```
>expr := "sin (x-5)"
```

```
     sin (x-5)
```

```
>aspect (1) ; plot2d(expr,-2,2):
```

contoh 2 dan penggunaan grid

```
>aspect(1)plot2d("log(x) + 3",-0.1,2, grid=6):
```

```
     Commands must be separated by semicolon or comma!
     Found: plot2d("log(x) + 3",-0.1,2, grid=6): (character 112)
     You can disable this in the Options menu.
     Error in:
     aspect(1)plot2d("log(x) + 3",-0.1,2, grid=6): ...
              ^
```

contoh 3 dan penggunaan parameter square (atau >square) untuk memilih y-range secara otomatis

```
>aspect(1,1) ; plot2d("x^4-2",-5,5, >square); insimg(15)
>aspect(2) ; plot2d("x^4-2", -5,5 ):
```

contoh 4 dan memberikan nama atau label pada garis sumbu

```
>plot2d("cos(x)", -4, 6, xl="x",yl="y") :
```

Sub Bab 2

Menggambar Grafik Fungsi Satu Variabel yang rumusnya Disimpan dalam Variabel Ekspresi

```
ekspresi
```

```
>expr &= x^5-1
```

$$x^5 - 1$$

plot dari ekspresi diatas

```
>aspect(2); plot2d(expr,-1,1):
```

contoh 1

```
>expr := "x^10-x-5"
```

```
x^10-x-5
```

```
>aspect(2) ; plot2d(expr,-1,1):
```

menggunakan variabel lokal

Ekspresi dapat dievaluasi secara numerik. Variabel x,y,z ditetapkan secara otomatis. Variabel lain dapat ditetapkan berdasarkan parameter yang ditetapkan(variabel lokal) atau melalui variabel global. variabel global adalah variabel yang selalu bisa diakses kapan pun dan di mana pun.

```
>expr &= a*x^5
```

$$a\,x^5$$

menggunakan variabel global

```
>a=6; expr(2.5)
```

```
585.9375
```

menggunakan variabel lokal

```
>expr(2.5,a=6)
```

```
      585.9375
```

evaluasi langsung

```
>"a*x^5"(3,4)
```

```
      1458
```

Oleh karena itu, banyak algoritma EMT yang dapat menggunakan ekspresi dalam x, bukan fungsi. Namun jika parameter tambahan yang tidak bersifat global dilibatkan, fungsi harus diutamakan.

menggunakan variabel global "a"

```
>a=5; plot2d("a*x^3-x",0,1):
>function f(x,a) := a*x^3-x
```

gunakan "a=6" sebagai parameter

```
>plot2d("f",0,1;6):
```

alternatif lain

```
>plot2d({{"f",6}},0,1):
```

alternatif lain

```
>plot2d("f(x,6)",0,1):
```

Sub Bab 3

Menggambar Fungsi Simbolik

Fungsi Plot yang paling penting untuk plot planar adalah plot2d(). Fungsi ini diimplementasikan dalam bahasa Euler dalam file "plot.e", yang dimuat diawal program.

plot2d() menerima ekspresi, fungsi, dan data.

Rentang plot diatur dengan parameter yang ditetapkan ssbagai berikut
- a,b: rentang x (default -2,2)
- -c,d: rentang y (default: skala dengan nilai)
- r: alternatifnya radius di sekitar pusat plot
- cx,cy: koordinat pusat plot (default 0,0)

Keterangan:(menggambar grafik fungsi satu variabel yang fungsinya didefinisikan sebagai fungsi simbolik)
- &: untuk menampilkan variabel pada teks

Berikut adalah beberapa contoh menggunakan fungsi. Seperti biasa di EMT, fungsi yang berfungsi untuk fungsi atau ekspresi lain, jadi kita dapat meneruskan parameter tambahan (selain x) yang bukan variabel global ke fungsi dengan parameter titik koma atau dengan koleksi panggilan.

```
>plot2d("f",0,1;0.4): // plot with a=0.4
>plot2d({{"f",0.2}},0,1);
>plot2d({{"f(x,b)",b=0.1}},0,1):
>function f(x) := x^3-x;...
 plot2d("f",r=1):
>plot2d("exp(-a*x^2)/a"):
```

Berikut merupakan ringkasan dari fungsi yang diterima
- ekspresi atau ekspresi simbolik dalam x
- fungsi atau fungsi simbolis dengan nama sebagai "f"
- fungsi simbolis hanya dengan nama f

Fungsi plot2d() juga menerima fungsi simbolis. Untuk fungsi simbolis, hanya nama saja yang berfungsi.

```
>function f(x) &= diff(x^x,x)
```

$$x^{x} \left(\log(x) + 1\right)$$

```
>plot2d(f,0,2):
>$&expr = sin (x)*exp(-x)
>plot2d(expr,0,3pi):
>plot2d("cos(x)","sin(3*x)"):
```

Sub Bab 4

Menggambar Fungsi Numerik

Fungsi Numerik adalah sebuah fungsi dengan himpunan bilangan cacah sebagai domain dan himpunan mendasar yang melibatkan hubungan matematis antara bilangan yang menjadi domain dan bilangan sebagai kodomain.

```
>
```

Fungsi numerik memiliki 1 atau lebih variabel independen, yang sering dilambangkan sebagai "X". Variabel X adalah nilai atau parameter yang dapat berubah, dan fungsi numerik menggambarkan bagaimana variabel ini memengaruhi variabel dependen. Variabel dependen adalah hasil perhitungan atau keluaran dari fungsi numerik yang bergantung pada nilai atau perubahan dalam variabel independen.

Dalam EMT cara mendefinisikan fungsi menggunakan syntak function. untuk mendefinisikan fungsi numerik menggunakan tanda ":="

Fungsi numerik menjelaskan bagaimana bilangan dalam domain berhubungan dengan bilangan sebagai kodomain, biasanya diberikan dalam bentuk rumus matematik(persamaan) atau aturan yang memetakan setiap domain kedalam kodomain yang sesuai. contoh:

f(x)=2x+3

```
>
```

(x)(variabel dependen) adalah fungsi yang memetakan setiap nilai x(variabel independen)kedalam nilai 2x+3. Terdapat berbagai jenis fungsi yang termasuk ke dalam fungsi numerik, diantaranya:

Fungsi linier dengan bentuk umum
f (x) = ax + b

Fungsi kuadrat dengan bentuk umum

f (x) = ax2 + bx + c

Fungsi eksponensial dengan bentuk umum

f (x) = ax

Fungsi logaritma dengan bentuk umum

f (x) = log a(x)

Fungsi trigonometri dengan bentuk umum

f (x) = sin(x), f (x) = cos(x)

Salah satu cara yang umum digunakan untuk memvisualisasikan fungsi numerik adalah dengan menggambar grafiknya. Grafik ini menggambarkan bagaimana variabel dependen berubah seiring perubahan variabel independen dan membantu dalam memahami sifat-sifat fungsi, seperti titik ekstrim

Contoh soal

```
>function r(x):= abs(x-10)
>function s(x):= abs(sin(x))
>r(-5)
```

```
    15
```

```
>function t(x):=log(x*(2+sin(x/1000)))
>function u(x):=integrate("(sin(x)*exp(-x^2)"0,x)
>function v(x):=logbase((x^2),2)
>plot2d("v"):
```

```
>plot2d("s"):
```

```
>plot2d("t",-2,2):
```

```
>function P(x):=x*cos(x)
>plot2d("P",-2*pi,2*pi):
```

Fungsi plot2d() adalah fungsi serbaguna untuk membuat grafik dalam bidang (grafik 2D). Fungsi ini dapat digunakan untuk membuat grafik fungsi-fungsi satu variabel, grafik data, kurva-kurva dalam bidang, grafik batang (bar plots), grid dari bilangan kompleks, dan grafik implisit dari fungsi dua variabel.

Parameter
x,y : persamaan, fungsi, atau vektor data a,b,c,d : area plot (default a=-2, b=2)
r : jika r diatur, maka a=cx-r, b=cx+r, c=cy-r, d=cy+r r bisa berupa vektor [rx,ry] atau vektor [rx1,rx2,ry1,ry2].
xmin,xmax : rentang parameter untuk kurva
auto : tentukan rentang y secara otomatis (default)
square : jika benar, mencoba menjaga rentang x-y tetap persegi n : jumlah interval (default adalah adaptif)
grid : 0 = tanpa grid dan label, 1 = hanya sumbu,
2 = grid normal (lihat di bawah untuk jumlah garis grid) 3 = di dalam sumbu
4 = tanpa grid
5 = grid penuh termasuk margin 6 = tanda di pinggiran
7 = hanya sumbu
8 = hanya sumbu, sub-ticks frame : 0 = tanpa bingkai
framecolor: warna bingkai dan grid
margin : angka antara 0 dan 0,4 untuk margin di sekitar plot color : Warna kurva. Jika ini adalah vektor warna,akan digunakan untuk setiap baris matriks plot. Dalam hal grafik titik, harus berupa vektor kolom. Jika vektor baris atau matriks penuh warna digunakan untuk grafik titik, akan digunakan untuk setiap titik data.
thickness : ketebalan garis untuk kurva

Nilai ini dapat lebih kecil dari 1 untuk garis yang sangat tipis.
style: Gaya plot untuk garis, penanda, dan isian.

Untuk titik gunakan
"[]", "<>", ".", "..", "...", "*", "+", " ", "-", "o"
"[]", "<>", "o" (bentuk terisi)
"[]w", "<>w", "ow" (tidak transparan)

Untuk garis gunakan
"-", "-", "-.", ".", ".-.", "-.-", "->"

Untuk poligon terisi atau plot batang gunakan
"", "O", "O", "/", "", "/","+", " ", "-", "t"

points : plot titik tunggal sebagai gantinya garis segmen addpoints : jika benar, plot segmen garis dan titik
add : tambahkan plot ke plot yang ada
user : aktifkan interaksi pengguna untuk fungsi delta : ukuran langkah untuk interaksi pengguna
bar : plot batang (x adalah batas interval, y adalah nilai interval) histogram : plot frekuensi x dalam n subinterval
distribusi=n : plot distribusi x dengan n subinterval even : gunakan nilai antar untuk histogram otomatis.
steps : plot fungsi sebagai fungsi langkah (steps=1,2)
adaptive : gunakan plot adaptif (n adalah jumlah minimal langkah) level : plot garis level dari fungsi implisit dua variabel
outline : menggambar batas rentang level.

```
>function s(x):=(x-10)
>function r(x):=abs(sin(x))
>s(-5)

     -15

>function t(x):=log(x*(2+sin(x/1000)))
>function u(x):=integrate("(sin(x)*exp(-x^2)"),0,x)
>function v(x):=logbase((x^2),2)
>plot2d("v"):

>plot2d("s"):

>function P(x):=x*cos(x)
>plot2d("P", -2*pi,2*pi):
```

Sub Bab 5

Menggambar Beberapa Kurva Sekaligus

Dalam subtopik ini, kita akan membahas mengenai cara menggambar beberapa kurva sekaligus. Dalam hal ini kita dapat menggambar beberapa kurva dalam jendela grafik yang berbeda secara bersama-sama. Untuk membuat ini kita dapat menggunakan perintah figure(). Berikut contoh dari menggambar beberapa kurva sekaligus

Menggambar plot fungsi

```
>reset;
>figure(2,2);...
 for n=1 to 4; figure(n); plot2d("x^"+n); end;...
 figure(0):
```

Penjelasan sintaks dari plot fungsi

- reset; Perintah ini berguna untuk menghapus grafik yang telah ada sebelumnya, sehingga kita dapat memulai dari awal untuk menggambar grafik

- figure(2x2); Perintah figure() digunakan untuk membuat jendela grafik dengan ukuran axb. Dalam kasus ini perintah figure(2,2) memiliki makna bahwa jendela grafik yang dibuat berukuran 2x2. Artinya, akan ada empat jendela grafik yang akan ditampilkan dengan tata letak 2 baris dan 2 kolom.
- for n=1 to 4; Perintah ini digunakan untuk melakukan pengulangan (looping) perintah sebanyak empat kali, yaitu dari 1 hingga 4.
- figure(n); Perintah ini digunakan untuk beralih dari jendela grafik satu ke jendela grafik lainnya (jendela grafik ke-n).
- plot2d("x^"+n); Perintah plot2d() digunakan untuk membuat plot fungsi matematika. Dalam hal ini fungsi yang diplot adalah x^n, di mana n adalah nilai dari variabel yang sedang diulang. Dengan kata lain, ini akan membuat plot dari x^1, x^2, x^3, dan x^4 dalam jendela grafik yang sesuai
- end; Perintah ini menandakan akhir dari looping.
- figure(0); Perintah ini digunakan untuk beralih kembali ke jendela grafik utama.

Dari sini dapat kita perhatikan untuk membuat kurva fungsi x^n (x pangkat n) perintahnya tidak ditulis dengan (x^n) melainkan ditulis dengan ("x^"+n). Tanda petik dua ("...") digunakan untuk mengidentifikasi bahwa teks tersebut merupakan ekspresi matematika.
Sedangkan tanda (+) digunakan untuk menggabungkan string dengan nilai yang berubah-ubah atau variabel.

Contoh lain:
Menggambar plot fungsi

```
>reset;
>figure(3,3);...
 for k=1:9; figure(k); plot2d("x^3-x",-2,2,grid=k); end;...
 figure(0):
```

Penjelasan sintaks dari plot fungsi

- reset; Perintah ini berguna untuk menghapus grafik yang telah ada sebelumnya, sehingga kita dapat memulai dari awal untuk menggambar grafik
- figure (3,3); Perintah ini digunakan untuk membuat jendela grafik dengan ukuran 3x3. Artinya, akan ada empat jendela grafik yang akan ditampilkan dengan tata letak 3 baris dan 3 kolom.
- for k=1:9; Perintah ini digunakan untuk melakukan pengulangan (looping) perintah sebanyak sembilan kali.
- figure(n); Perintah ini digunakan untuk beralih dari jendela grafik satu ke jendela grafik lainnya (jendela grafik ke-n).
- plot2d("x^3-x",-2,2,grid=k); Perintah plot2d() digunakan untuk membuat plot fungsi matematika. Dalam hal ini fungsi yang diplot adalah x^3-x, dengan batas sumbu x dari -2 hingga 2. Argumen grid=k digunakan untuk mengaktifkan grid pada jendela grafik ke-k.
- end; Perintah ini menandakan akhir dari looping.
- figure(0); Perintah ini digunakan untuk beralih kembali ke jendela grafik utama.

Dari contoh diatas dapat kita perhatikan bahwa tampilan plot dari yang ke-1 hingga ke-9 memiliki tampilan yang berbeda-beda. Dalam EMT memiliki berbagai gaya plot 2D yang dapat dijalankan menggunakan perintah grid=n dimana n adalah jumlah langkah minimal. Setiap nilai n memiliki tampilan plot adaptif yang berbeda dalam plot 2D, diantaranya yaitu:
0 : tidak ada grid (kisi), frame, sumbu, dan label, hanya kurva saja
1 : dengan sumbu, label-label sumbu di luar frame jendela grafik
2 : tampilan default
3 : dengan grid pada sumbu x dan y, label-label sumbu berada di dalam jendela grafik
4 : tidak ada grid (kisi), sumbu x dan y, dan label berada di luar frame jendela grafik
5 : tampilan default tanpa margin di sekitar plot
6 : hanya dengan sumbu x y dan label, tanpa grid
7 : hanya dengan sumbu x y dan tanda-tanda pada sumbu.
8 : hanya dengan sumbu dan tanda-tanda pada sumbu, dengan tanda-tanda yang lebih halus pada sumbu.
9 : tampilan default dengan tanda-tanda kecil di dalam jendela
10: hanya dengan sumbu-sumbu, tanpa tanda

Contoh lain:
Menggambar plot fungsi

```
>reset;
>aspect(1.2);
>figure(3,4); ...
  figure(2); plot2d("2x^3-x",grid=1); ... // x-y-axis
> figure(3); plot2d("2x^3-x",grid=2); ... // default ticks
>figure(4); plot2d("2x^3-x",grid=3); ... // x-y- axis with labels inside
> figure(5); plot2d("2x^3-x",grid=4); ... // no ticks, only labels
>figure(6); plot2d("2x^3-x",grid=5); ... // default, but no margin
>figure(7); plot2d("2x^3-x",grid=6); ... // axes only
>figure(8); plot2d("2x^3-x",grid=7); ... // axes only, ticks at axis
>figure(9); plot2d("2x^3-x",grid=8); ... // axes only, finer ticks at axis
>figure(10); plot2d("2x^3-x",grid=9); ... // default, small ticks inside
>figure(11); plot2d("2x^3-x",grid=10); ...// no ticks, axes only
>figure(0):
```

Penjelasan sintaks dari plot fungsi

- aspect(1.2); Perintah aspect() digunakan untuk mengatur rasio aspek dari jendela grafik. Hal ini berarti perintah aspect(1.2); akan menghasilkan plot dengan perbandingan rasio panjang dan lebar 2:1.
- figure(3,4); Perintah ini digunakan untuk membuat jendela grafik dengan ukuran 3x4. Jadi, akan ada total 12 jendela grafik yang akan ditampilkan dalam tata letak 3 baris dan 4 kolom.
- figure(1); plot2d("x^3-x",grid=0); ... Adalah perintah untuk beralih ke jendela grafik pertama dan menggambar plot dari fungsi x^3 - x tanpa grid, frame, atau sumbu.
- figure(2); plot2d("x^3-x",grid=1); ... Adalah perintah untuk beralih ke jendela grafik kedua dan menggambar plot dari fungsi x^3 - x dengan grid hanya pada sumbu x dan y.
- figure(3); plot2d("x^3-x",grid=2); ... Adalah perintah untuk beralih ke jendela grafik ketiga dan menggambar plot dari fungsi x^3 - x dengan tampilan default, termasuk tanda-tanda default pada sumbu.
- figure(4); plot2d("x^3-x",grid=3); ... Adalah perintah untuk beralih ke jendela grafik keempat dan menggambar plot dari fungsi x^3 - x dengan grid pada sumbu x dan y, serta label-label sumbu yang ada di dalam jendela.
- figure(5); plot2d("x^3-x",grid=4); ... Adalah perintah untuk beralih ke jendela grafik kelima dan menggambar plot dari fungsi x^3 - x tanpa tanda-tanda sumbu, hanya label-label yang ada.
- figure(6); plot2d("x^3-x",grid=5); ... Adalah perintah untuk beralih ke jendela grafik keenam dan menggambar plot dari fungsi x^3 - x dengan tampilan default, tetapi tanpa margin di sekitar plot.
- figure(7); plot2d("x^3-x",grid=6); ... Adalah perintah untuk beralih ke jendela grafik ketujuh dan menggambar plot dari fungsi x^3 - x hanya dengan sumbu-sumbu (tanpa grid atau label).
- figure(8); plot2d("x^3-x",grid=7); ... Adalah perintah untuk beralih ke jendela grafik kedelapan dan menggambar plot dari fungsi x^3 - x hanya dengan sumbu-sumbu dan tanda-tanda pada sumbu.
- figure(9); plot2d("x^3-x",grid=8); ... Adalah perintah untuk beralih ke jendela grafik kesembilan dan menggambar plot dari fungsi x^3 - x hanya dengan sumbu-sumbu dan tanda-tanda pada sumbu, dengan tanda-tanda yang lebih halus pada sumbu.
- figure(10); plot2d("x^3-x",grid=9); ... Adalah perintah untuk beralih ke jendela grafik kesepuluh dan menggambar plot dari fungsi x^3 - x dengan tanda-tanda default kecil di dalam jendela.
- figure(11); plot2d("x^3-x",grid=10); ... Adalah perintah untuk beralih ke jendela grafik kesebelas dan menggambar plot dari fungsi x^3 - x hanya dengan sumbu-sumbu, tanpa tanda-tanda.
- figure(0); Adalah perintah untuk beralih kembali ke jendela grafik utama atau jendela grafik dengan nomor 0 setelah semua perintah dalam urutan selesai dieksekusi.

Dari ketiga contoh di atas, dapat kita katakan bahwa untuk menggambar beberapa kurva sekaligus itu dapat dilakukan dengan satu baris perintah ataupun dengan cara mendefinisikannya 1 per 1.

Terlihat beberapa jenis grid memiliki tampilan yang mirip atau sama, seperti 1 dan 2, 2 dan 5, 4 dan 9, 7 dan 8, untuk dapat membedakannya secara lebih jelas, ubah grid dari contoh di bawah ini.

```
>reset;
>aspect(1.3);
>figure(1,3);...
```

```
figure (1); plot2d("x^2*exp(-x)",0,10);...
figure (2); plot2d("2*exp(x)",-5,5);...
figure (3); plot2d("exp(x^2)",-2,2);...
figure (0):
```

Contoh lain:

```
>reset;
>aspect(3/4);
>figure(2,1);...
 for a=1:2; figure(a); plot2d("2*x*log(x^2)",0,3,grid=a); end;...
 figure(0):
```

Sub Bab 6

Menggambar Beberapa Kurva pada bidang koordinat yang sama

Plot lebih dari satu fungsi (multiple function) ke dalam satu jendela dapat dilakukan dengan berbagai cara. Salah satu caranya adalah menggunakan >add untuk beberapa panggilan ke plot2d secara keseluruhan, kecuali panggilan pertama.

Berikut contohnya:
menggambar kurva

```
>aspect(); plot2d("cos(x)",r=3); plot2d("x^2",style=".",>add):
```

```
>aspect(2); plot2d("cos(x)-1",-1,6); plot2d("sin(x)-1",style="--",>add):
```

Selain menggunakan >add kita juga bisa menambahkannya secara langsung

Berikut contohnya:
Menggambar kurva

```
>plot2d(["2x+1","x"],0,8):
```

```
>aspect(1.5); plot2d(["sin(2x)","cos(3x)"],0,8):
```

Kegunaan >add yang lain juga bisa untuk menambahkan titik pada kurva.

Berikut contohnya:
Menambahkan sebuah titik di

```
>aspect(); plot2d("x+4",-2,5,); plot2d(2,6,>points,>add):
```

Kita juga bisa mencari titik perpotongan dengan cara berikut:

```
>plot2d(["sin(x)","2x"],r=2,cx=1,cy=1, ...
   color=[black,blue],style=["-","."], ...
   grid=1);
>x0=solve("sin(x)-2x",1);  ...
   plot2d(x0,x0,>points,>add);  ...
   label("sin(x) = 2x",x0,x0,pos="cl",offset=20):
>function f(x,a) := x^2+a*x-x/a; ...
 plot2d("f",-10,10;1,title="a=1"):
> plot2d({{"f",1}},-10,10); ...
 for a=1:10; plot2d({{"f",a}},>add); end:
>function f(x,a) := x^2*exp(-x^2/a); ...
```

```
 plot2d("f",-10,10;5,thickness=2,title="a=5"):
>plot2d({{"f",1}},-8,8); ...
 for a=2:5; plot2d({{"f",a}},>add,thickness=2); end:
>aspect(2.1); &plot2d(1/x,[x,-1,1]):
>x=linspace(-1,1,50);...
 plot2d("1/x"):
```

Sub Bab 7

Menuliskan Label koordinat,label kurva, dan keterangan

kurva(legend) Dalam EMT, untuk menambahkan judul dapat dilakukan dengan title="..."
untuk menambahkan sumbu x dan sumbu y dapat dilakukan dengan x1="...", y1="..."
sebagai contoh:

```
>plot2d("x^2-4*x"):
```

untuk menambahkan judul dapat dilakukan dengan title="..."
untuk menambahkan sumbu x dan sumbu y dapat dilakukan dengan x1="...", y1="..."

```
>plot2d("x^2-4*x",title="FUNGSI y=x^2-4*x",yl="Sumbu y",xl="Sumbu x"):
```

Selain itu juga dapat dengan cara lain seperti contoh berikut:

```
>expr := "x^3-x"; ...
   plot2d(expr,title="y="+expr,xl="Sumbu x",yl="Sumbu y"); ...
   label("(1,0)",1,0);  label("Max",E,expr(E),pos="lc"):
```

Sub Bab 8

Mengatur ukuran gambar,format(style),dan warna kurva

Untuk mengubah ukuran, dapat dilakukan dengan menggunakan aspect="...", semakin besar nilai aspect, maka ukuran kurva akan semakin kecil, begitupun sebaliknya

untuk mengganti style, dapat dipilih dengan berbagai pilihan
style="...", dapat dipilih dari, misal : "-","_","-.",".-.","-.-".

untuk warna dapat dipilih sebagai salah satu warna default
color="...", warna default= red,green,blue,yellow, dll

sebagai contoh:

```
>aspect(1); plot2d("exp(x^2-3)"):
```

ukuran kurva dapat diganti dengan mengganti nilai aspect="...", semakin besar nilai aspect, maka ukuran kurva akan semakin kecil Untuk mengganti warna dapat ditambahkan dengan color="...", sedangkan untuk mengganti format(style) dapat dilakukan dengan menambahkan style="..."

```
>aspect(2); plot2d("exp(x^2-3)", color=red, style="--"):
```

Berikut adalah tampilan warna EMT yang telah ditentukan

```
>aspect (1) ; columnsplot (ones(1,16),lab=0:15,grid=0, color=0:15) :
```

selain menggunakan warna default, untuk mengubah warna dapat juga dengan menggunakan kode warna di atas

sebagai contoh:

```
>aspect(1); plot2d("exp(x^3+2*x)",r=3, color=1, style="--"):
```

Sub Bab 9

Menggambar Sekumpulan Kurva dalam satu perintah plot2d.

Dalam pembahasan sub-bab 9 kali ini akan membahas mengenai bagaimana menggambar sekumpulan kurva dalam satu perintah plot2d. Menggambar sekumpulan kurva dalam satu perintah plot2d adalah teknik yang digunakan untuk memvisualisasikan beberapa fungsi dalam satu grafik. Ini memudahkan perbandingan antara beberapa kurva.

Contoh

```
>plot2d(["x^2","2*x"],-3,3):
```

- Dalam contoh ini, merupakan gambar dua kurva sekaligus, yaitu x^2

dan 2x, pada rentang -3 hingga 3.

- Hasilnya akan menunjukkan grafik dari kedua fungsi tersebut, dan

titik-titik potongan antara keduanya adalah solusi dari persamaan kuadrat.

```
>plot2d(["sin(x)","cos(x)"],0,2pi):
```

- Pada contoh ini, merupakan gambar dua fungsi trigonometri, sin(x) dan cos(x), pada rentang 0 hingga 2p.
- Ini akan menghasilkan dua grafik yang memperlihatkan hubungan antara sin(x) dan cos(x) dalam rentang tersebut.

```
>plot2d(["sin(x)","cos(x)"],0,2pi,color=red:green):
```

Sama seperti contoh kedua, gambar sin(x) dan cos(x) pada rentang 0 hingga 2phi, tetapi Anda juga memberikan warna yang berbeda pada kedua grafik (sin(x) berwarna merah dan cos(x) berwana hijau.

```
>plot2d(["sin(x)","cos(x)"],xmin=0,xmax=2pi):
```

Dalam contoh ini, menggunakan parameter `xmin` dan `xmax` untuk mengatur rentang tampilan grafik pada 0 hingga 2p.

```
>
>plot2d(["cos(x)","sin(3*x)"],xmin=0,xmax=2pi):
```

- ini Merupakan gambar dua fungsi, yaitu cos(x) dan sin(3*x), dalam rentang 0 hingga 2p.
- Penjelasan mencakup konsep bahwa grafik berulang dalam rentang tertentu karena fungsi-fungsi ini memiliki frekuensi, periode, dan amplitudo yang berbeda.

```
>plot2d("cos(x)","sin(3*x)",xmin=0,xmax=2pi):
```

Sintaks diatas lebih menjelaskan bahaimana hubungan periodik grafik fungsi dari 2 fungsi yaitu cos x dan sin 3x dari rentang khusus dimana xmin dari 0 sampai 2pi, hal tersebut dapat terjadi karena fungsi cos x dan fungsi sin 3x memiliki frekuensi, periode, dan amplitudo yang berbeda. grafik akan berulang pada rentang tertentu dan menghasilkan sebuah pola.

```
>x=linspace(0,2pi,1000); plot2d(sin(5x),cos(7x)):
```

sintaks linspace digunakan untuk menghasilkan vektor x dari rentang yang telah ditentukan yaitu o sampai 2pi yang berisi 1000 nilai yang teratur

```
>a:=5.6; f &= exp(-a*x^2)/a;
>plot2d(f,r=1,thickness=2):
```

- Fungsi f(x) yang merupakan hasil dari ekspresi exp(-a*x^2)/a. Fungsi ini memiliki parameter a yang bergantung pada nilai yang diterapkan sebelumnya.
- Menggunakan perintah plot2d untuk menggambar grafik dari fungsi f(x). Parameter r digunakan untuk mengatur rentang plot dan parameter thickness digunakan untuk mengatur ketebalan garis grafik.

```
>plot2d(&diff(f,x),>add,style="--",color=red):
```

ini adalah grafik fungsi f dan grafik turunan pertama dari fungsi f. sintaks r=1 digunakan untuk mengatur rentang yang akan ditampilkan pada plot, r=1 berarti rentang dari -1 sampai 1.
sintaks >add digunakan untuk menambahkan grafik kedalam jendela grafik yang sudah ada sebelumnya.

```
>plot2d("x^2",0,1,steps=1,color=red,n=10):
>plot2d("x^2",>add,steps=2,color=blue,n=10):
```

sintaks steps digunakan untuk mengatur jumlah langkah atau titik-titik yang digunakan dalam plot.
dan sintaks n digunakan untuk mengatur jumlah step yang akan digunakan. semakin bayak n, maka bentuk grafik akan semakin mendekati aslinya.

```
>function f(x) &= x^x;
>plot2d(f,r=1,cx=1,cy=1,color=blue,thickness=2);
>plot2d(&diff(f(x),x),>add,color=red,style="-.-"):
```

sintaks cx=1, cy=1 digunakan untuk mengatur pusat tampilan grafik, maka plot akan diatur dengan titik pusat (1,1).

```
>plot2d("(1-x)^10",0,1);
>for i=1 to 10; plot2d("bin(10,i)*x^i*(1-x)^(10-i)",>add); end;
>insimg;
```

dalam contoh kita menggambar serangkaian plot yang menggambarkan distribusi binomial dengan berbagai nilai i dari 1 hingga 10. kali ini kita menggunakan sintaks untuk melakukan looping pada fungsi yang berasosiasi dengan koefisien binomial dengan kombinasi 10 item. ini memungkinkan untuk memahami bagaimana distribusu binomial berubah dengan berbagai parameter.

```
>x=linspace(0,1,500);
>n=10; k=(0:n)';
>y=bin(n,k)*x^k*(1-x)^(n-k);
>plot2d(x,y):
```

n adalah vektor baris
k adalah vektor kolom
y adalah matrik dari vektor baris dan vektor kolom tersebut dengan menggunakanfungsi binomial.

```
>x=linspace(0,1,200); y=x^(1:10)'; plot2d(x,y,color=1:10):
>n=(1:10)'; plot2d("x^n",0,1,color=1:10):


>
>function f(x,a) := 1/a*exp(-x^2/a); ...
 plot2d("f",-10,10;5,thickness=2,title="a=5"):


>plot2d({{"f",1}},-10,10); ...
 for a=2:10; plot2d({{"f",a}},>add); end:
```

Sub Bab 10

Membuat Gambar Kurva yang Bersifat Interaktif

Kode ini, menggunakan `plot2d` untuk membuat plot dari fungsi matematika `2*x^3-a*x` dengan parameter `a`. Flag `>user` memungkinkan interaksi pengguna. Setelah plot ditampilkan, pengguna dapat melakukan beberapa tindakan interaktif.
Saat plot ditampilkan dengan flag `>user`, pengguna dapat melakukan beberapa tindakan interaktif sebagai berikut:

- Perbesar dengan + atau -: Pengguna dapat memperbesar atau memperkecil plot dengan menggunakan tombol + atau - pada keyboard.
- Pindahkan Plot dengan Tombol Kursor: Pengguna dapat menggeser plot dengan menggunakan tombol kursor (panning).
- Pilih Jendela Plot dengan Mouse: Pengguna dapat memilih area tertentu dalam plot dengan menggunakan mouse.
- Atur Ulang Tampilan dengan Spasi: Jika pengguna menekan tombol spasi, maka tampilan plot akan diatur ulang ke jendela plot.
- Keluar dengan Kembali: Jika pengguna menekan tombol kembali, maka pengguna dapat keluar dari interaksi plot.

```
>plot2d({{"2*x^3-a*x",a=1}},>user,title="Press any key!"); ...
 insimg;
> plot2d("exp(x)*sin(x)",user=true, ...
   title="+/- or cursor keys (return to exit)"):
```

Berikut ini menunjukkan cara interaksi pengguna tingkat lanjut

Ini adalah pemanggilan fungsi plot2d yang digunakan untuk membuat plot dari fungsi matematika exp(x)*sin(x). Parameter user=true menunjukkan bahwa ini adalah plot yang interaktif, yang berarti pengguna dapat berinteraksi dengan plot ini.

title="+/- or cursor keys (return to exit)": Ini adalah judul yang akan ditampilkan di atas plot. Pesan ini memberi petunjuk kepada pengguna tentang bagaimana mereka dapat berinteraksi dengan plot ini. Mereka dapat menggunakan tombol + atau - atau tombol kursor untuk berinteraksi dengan plot, dan tombol return (Enter) untuk keluar dari interaksi.

Berikut ini menunjukkan cara interaksi pengguna tingkat lanjut:
- mousedrag(): Ini adalah fungsi bawaan yang digunakan untuk menunggu event mouse atau keyboard. Fungsi ini dapat mendeteksi kejadian seperti klik mouse, pergerakan mouse, atau penekanan tombol.
- dragpoints(): Fungsi ini memanfaatkan mousedrag() untuk memungkinkan pengguna menyeret titik-titik pada plot. Ini berarti pengguna dapat mengklik dan menarik titik-titik dalam plot sesuai dengan preferensi mereka.

Kita membutuhkan fungsi plot terlebih dahulu. Sebagai contoh, kita interpolasi dalam 5 titik dengan polinomial. Fungsi harus diplot ke area plot tetap.

```
>function plotf(xp,yp,select) ...
   d=interp(xp,yp);
   plot2d("interpval(xp,d,x)";d,xp,r=2);
   plot2d(xp,yp,>points,>add);
   if select>0 then
     plot2d(xp[select],yp[select],color=red,>points,>add);
   endif;
   title("Drag one point, or press space or return!");
 endfunction
```

Perhatikan parameter titik koma di plot2d (d dan xp), yang diteruskan ke evaluasi fungsi interp(). Tanpa ini, kita harus menulis fungsi plotinterp() terlebih dahulu, mengakses nilai secara global.

Sekarang kita menghasilkan beberapa nilai acak, dan membiarkan pengguna menyeret poin.

kode berikut digunakan untuk menghasilkan beberapa nilai acak t dan membiarkan pengguna menyeret titik-titik pada plot dengan menggunakan fungsi dragpoints():

```
>t=-1:0.5:1; dragpoints("plotf",t,random(size(t))-0.5):
```

Ada juga fungsi, yang memplot fungsi lain tergantung pada vektor parameter, dan memungkinkan pengguna menyesuaikan parameter ini.

Pertama kita membutuhkan fungsi plot.

```
>function plotf([a,b]) := plot2d("exp(a*x)*cos(2pi*b*x)",0,2pi;a,b);
```

Kemudian kita membutuhkan nama untuk parameter, nilai awal dan matriks rentang nx2, opsional baris judul.
Ada slider interaktif, yang dapat mengatur nilai oleh pengguna. Fungsi dragvalues() menyediakan ini.

```
>dragvalues("plotf",["a","b"],[-1,2],[[-2,2];[1,10]], ...
   heading="Drag these values:",hcolor=black):
```

Dimungkinkan untuk membatasi nilai yang diseret ke bilangan bulat. Sebagai contoh, kita menulis fungsi plot, yang memplot polinomial Taylor derajat n ke fungsi kosinus.

```
>function plotf(n) ...
 plot2d("cos(x)",0,2pi,>square,grid=6);
 plot2d(&"taylor(cos(x),x,0,@n)",color=blue,>add);
 textbox("Taylor polynomial of degree "+n,0.1,0.02,style="t",>left);
 endfunction
```

Sekarang kami mengizinkan derajat n bervariasi dari 0 hingga 20 dalam 20 pemberhentian. Hasil dragvalues() digunakan untuk memplot sketsa dengan n ini, dan untuk memasukkan plot ke dalam buku catatan.

```
>nd=dragvalues("plotf","degree",3,[0,10],10,y=0.8, ...
    heading="Drag the value:"); ...
 plotf(nd):
```

Berikut ini adalah demonstrasi sederhana dari fungsi tersebut. Pengguna dapat menggambar di atas jendela plot, meninggalkan jejak poin.

```
>function dragtest ...
   plot2d(none,r=1,title="Drag with the mouse, or press any key!");
   start=0;
   repeat
```

```
    {flag,m,time}=mousedrag();
    if flag==0 then return; endif;
    if flag==2 then
      hold on; mark(m[1],m[2]); hold off;
    endif;
  end
endfunction
```

Sub Bab 11

Menggambar Kurva Fungsi Parametrik

Kita telah terbiasa dengan kurva yang didefinisikan oleh sebuah persamaan yang menghubungkan koordinat x dan y Contohnya

Atau

dimana persamaan-persamaan ini tidak dikaitkan dengan panjang kurva s , waktu t, dan besaran lainnya. Besaran besaran ini disebut parameter
persamaan parametrik adalah persamaan yang menyatakan hubungan variabel x, y dituliskan dengan

dengan a<=t<=b tiap nilai t menentukan titik(x,y) pada kurva. Jadi , dengan berubahnya nilai t. titik

bergerak sepanjang kurva yang disebut kurva parametrik

Dalam contoh berikut, kita memplot spiral

Kita perlu menggunakan banyak titik untuk tampilan yang halus

```
>t=linspace(0,1,1000); ...
 plot2d(t*cos(2*pi*t),t*sin(2*pi*t),r=1):
```

```
      r digunakan untuk mengatur radius marker titik-titik yang akan
```

digunakan dalam plot.

Sebagai alternatif, dimungkinkan untuk menggunakan dua ekspresi untuk kurva. Berikut ini plot kurva yang sama seperti di atas.

```
>plot2d("x*cos(2*pi*x)","x*sin(2*pi*x)",xmin=0,xmax=1,r=1):
```

Perintah linspace digunakan untuk membuat array nilai yang terdistribusi secara merata antara dua angka tertentu. Fungsi ini sangat berguna untuk menentukan rentang nilai yang ingin digunakan pada sumbu x atau y ketika membuat plot.

```
      0  : Nilai awal dari rentang.
      1  : Nilai akhir dari rentang.
```

Perintah linspace akan menghasilkan array dengan n elemen yang terdistribusi merata antara start dan stop.

```
>t=linspace(0,1,1000); r=exp(-t); x=r*cos(2pi*t); y=r*sin(2pi*t);
>plot2d(x,y,r=1):
```

exp(-t) menghasilkan nilai yang semakin mendekati nol seiring dengan pertambahan nilai t, karena eksponensial dari nilai negatif semakin mendekati nol saat nilai t semakin besar.
Jadi, r = exp(-t) memberikan suatu fungsi yang menurun dengan nilai t. Dalam konteks program ini, r digunakan untuk mengontrol jari-jari dari kurva spiral dalam plot 2D. Jari-jari ini semakin kecil seiring dengan pertambahan nilai t, menciptakan efek spiral yang semakin rapat ke pusat pada bagian ujung kurva.

Pada contoh berikutnya, kita memplot kurvanya

dengan

```
>t=linspace(0,2pi,1000); r=1+sin(3*t)/2; x=r*cos(t); y=r*sin(t); ...
 plot2d(x,y,>filled,fillcolor=red,style="/",r=1.5):
```

Contoh lain

```
>t=linspace(-3,3,1000); x=2*t+1; y=t^2-1;
>plot2d(x,y,r=8):
>t=linspace(0,2pi,1000); r=3; x=r*cos(t); y=r*sin(t);...
 plot2d(x,y,>filled,fillcolor=green,style="/",r=5):
>t=linspace(-1,1,1000); x=t^2; y=2*t;...
 plot2d(x,y):
>t=linspace(0,2pi,1000); x=3*cos(t); y=2*sin(t);...
 plot2d(x,y,>filled,fillcolor=green,style="/",r=3):
```

Sub Bab 12

Menggambar Kurva Fungsi Implisit

Fungsi implisit adalah fungsi yang memuat lebih dari satu variabel, berjenis variabel bebas dan variabel terikat yang berada dalam satu ruas sehingga tidak bisa dipisahkan pada ruas yang berbeda.

Untuk fungsi implisit, harus berupa fungsi atau ekspresi dari parameter x dan y.

Untuk menggambar himpunan f(x,y)=c untuk satu atau lebih konstanta c, dapat menggunakan plot2d().

Fungsi implisit juga dapat diisi dengan persamaan tingkat

Untuk fungsi ini harus berupa matriks 2xn dimana baris pertama berisi awal dan baris kedua adalah akhir dari setiap interval.

Plot implisit menunjukkan garis level yang menyelesaikan f(x,y)=level, di mana "level" dapat berupa nilai tunggal atau vektor nilai. Jika level="auto", akan ada garis level nc, yang akan menyebar antara fungsi minimum dan maksimum secara merata. Warna yang lebih gelap atau lebih terang dapat ditambahkan dengan >hue untuk menunjukkan nilai fungsi. Untuk fungsi implisit, xv harus berupa fungsi atau ekspresi dari parameter x dan y, atau, sebagai alternatif, xv dapat berupa matriks nilai.

Euler dapat menandai garis level

dari fungsi apapun.

Untuk menggambar himpunan f(x,y)=c untuk satu atau lebih konstanta c, Anda dapat menggunakan plot2d() dengan plot implisitnya di dalam bidang. Parameter untuk c adalah level=c, di mana c dapat berupa vektor garis level. Selain itu, skema warna dapat digambar di latar belakang untuk menunjukkan nilai fungsi untuk setiap titik dalam plot. Parameter "n" menentukan kehalusan plot.

Contoh Soal

```
>aspect(2)
```

```
>plot2d("x^2+y^2-x*y-x",r=1.5,level=0,contourcolor=green):
```

```
>expr := "2*x^2+x*y+3*y^4+y"; // define an expression f(x,y)
>plot2d(expr,level=0,contourcolor=green): // Solutions of f(x,y)=0
>plot2d(expr,level=0:0.5:20,>hue,contourcolor=white,n=200): // nice
```

Parameter >hue digunakan untuk memberikan warna pada kontur sesuai dengan levelnya. Kontur dengan level yang lebih tinggi akan memiliki warna yang berbeda.

```
>plot2d(expr,level=0:0.5:20,>hue,>spectral,n=200,grid=4): // nicer
```

>spectral digunakan untuk mengatur palet warna yang akan digunakan pada kontur. Dalam hal ini, digunakan palet warna "spectral".

```
>x=-2:0.05:1; y=x'; z=expr(x,y);
>plot2d(z,level=0,a=-1,b=2,c=-2,d=1,>hue):
>plot2d("x^3-y^2",>contour,>hue,>spectral):
```

Perintah >contour adalah cara untuk menghasilkan plot kontur, yaitu plot yang menunjukkan garis-garis kontur yang mewakili tingkat-tingkat dari suatu fungsi. Jumlah dan posisi garis kontur akan secara otomatis diatur oleh Euler Math Toolbox berdasarkan distribusi nilai-nilai fungsi.

Penggunaan level memungkinkan Anda secara eksplisit menentukan tingkat kontur yang ingin kita tampilkan pada plot. kita dapat mengatur level kontur sesuai dengan preferensi kita, dan plot akan menampilkan garis kontur pada tingkat-tingkat yang kita tentukan.

```
>plot2d("x^3-y^2",level=0,contourwidth=3,>add,contourcolor=red):
>z=z+normal(size(z))*0.2;
>plot2d(z,level=0.5,a=-1,b=2,c=-2,d=1):
```

normal(size(z)) menghasilkan matriks dengan ukuran yang sama dengan matriks z, dan setiap elemennya diambil dari distribusi normal standar (mean 0, deviasi standar 1). Kemudian, matriks z diubah dengan menambahkan nilai-nilai acak ini, yang telah dikalikan dengan 0.2. Ini menciptakan variasi acak dalam matriks z.

```
>plot2d(expr,level=[0:0.2:5;0.05:0.2:5.05],color=lightgray):
>plot2d("x^2+y^3+x*y",level=1,r=4,n=100):
>plot2d("x^2+2*y^2-x*y",level=0:0.1:10,n=100,contourcolor=white,>hue):
>plot2d(expr,level=[0;1],style="-",color=blue): // 0 <= f(x,y) <= 1
>plot2d("x^4+y^4",r=1.5,level=[0;1],color=blue,style="/"):
>plot2d("x^2+y^3+x*y",level=[0,2,4;1,3,5],style="/",r=2,n=100):
>plot2d("x^2+y^3+x*y",level=-10:20,r=2,style="-",dl=0.1,n=100):
```

dl Parameter ini mengatur tingkat penghalusan pada plot. Semakin kecil nilai ini, semakin halus plotnya.

```
>plot2d("sin(x)*cos(y)",r=pi,>hue,>levels,n=100):
>plot2d("(x^2+y^2-1)^3-x^2*y^3",r=1.3, ...
 style="/",color=red,<outline, ...
 level=[-2;0],n=100):
```

<outline: Parameter ini mengatur plot agar hanya memiliki kontur saja tanpa diisi.

Misal plot solusi dari persamaan

```
>plot2d("x^3-x*y+x^2*y^2",r=6,level=6,n=100):
>plot2d("x^2+y^2-1",level=0):
>plot2d("x^2+y^2-1",r=3,level=0:1:10,n=200):
```

```
>plot2d("x^2+y^2-1",r=3,level=0:1:10,>hue,contourcolor=white):
>plot2d("x^2+y^2-1",r=3,level=0:1:20,>hue,>spectral,n=200,grid=4):
>plot2d("x^2+y^2-1",level=0,a=-2,b=2,c=-2,d=2,>hue):
>plot2d("x^2+y^2-1",>contour,>hue,>spectral):
>plot2d("x^2+y^2-1",level=[0:0.2:5;0.05:0.2:5.05],color=lightgray):
>plot2d("x^2+y^2-1",r=2,level=[0;1],style="-",color=blue): // 0 <= f(x,y) <= 1
>plot2d("x^2+y^2",r=1.5,level=[0;1],color=blue,style="/"):
>plot2d("x^2+y^2-1",level=[0,2,4;1,3,5],style="/",r=2,n=100):
>plot2d("x^2+y^2-1",level=-10:20,r=3,style="-",dl=0.1,n=100):
```

Sub bab 13

Menggambar Grafik Bilangan Kompleks

Bilangan kompleks secara visual dapat direpresentasikan sebagai sepasang angka (a, b) membentuk vektor pada diagram yang disebut diagram Argand, mewakili yang bidang kompleks. Sumbu-x adalah sumbu nyata dan sumbu-y adalah sumbu imajiner.

Menggambar kurva fungsi kompleks sendiri adalah proses visualisasi grafis dari fungsi matematika kompleks (yaitu fungsi yang melibatkan bilangan kompleks, yaitu bilangan dengan bagian real dan imajiner) berperilaku dalam koordinas kompleks. Hal tersebut memungkinkan untuk melihat bagaimana pola, bentuk, dan sifat dari fungsi kompleks tersebut.

Array bilangan kompleks juga dapat diplot. Kemudian titik-titik grid akan terhubung. Jika sejumlah garis kisi ditentukan (atau vektor garis kisi 1x2) dalam argumen cgrid, hanya garis kisi tersebut yang terlihat.

Matriks bilangan kompleks akan secara otomatis diplot sebagai kisi di bidang kompleks.

> Definisi fungsi kompleks, mendefinisikan fungsi kompleks yang dianalisis atau digambarkan. Fungsi ini memiliki variabel kompleks z, yang melibatkan bagian real dan imajiner.
> Selanjutnya kita dapat menggunakan fungsi linspace. Fungsi linspace sendiri adalah salah satu fungsi yang umum digunakan dalam pemrograman, terutama dalam konteks pemrograman numerik dan ilmu data. Ini sering digunakan untuk menghasilkan urutan nilai dalam rentang tertentu dengan jumlah titik yang sama di antara dua titik ujungnya. Penggunaannya tidak terbatas pada pemrosesan sinyal atau elektromagnetik, tetapi bisa digunakan dalam berbagai konteks di mana Anda perlu membuat urutan nilai.
> Penentuan rentang, memilih rentang nilai z yang ingin ditampilkan di dalam plot. Rentang ini mencakup wilayah kompleks tertentu yang ingin diamati.
> Menggunakan sintaks plot2d.
> Penyesuaian plot, mengubah plot sesuai yang diinginkan (mengubah warna, format (style), dan sebagainya).

Dalam contoh berikut, kami memplot gambar lingkaran satuan di bawah fungsi eksponensial. Parameter cgrid menyembunyikan beberapa kurva grid.

Contoh

```
>aspect(); r=linspace(0,1,50); a=linspace(0,2pi,80)'; z=r*exp(I*a);...
 plot2d(z,a=-1.25,b=1.25,c=-1.25,d=1.25,cgrid=10):
```

Penjelasan sintaks

z : sebuah ekspresi atau fungsi yang akan digambar dalam koordinat kompleks.
a,b,c,d : parameter-parameter yang digunakan untuk mengatur jendela tampilan (viewport) dalam koordinat kompleks. Parameter-parameter ini akan menentukan rentang sumbu x dan sumbu y yang akan ditampilkan di dalam plot.
cgrid : parameter ini mengontrol tampilan grid pada plot. Jika cgrid=n, maka grid akan ditampilkan, jika cgrid=0, maka grid akan disembunyikan.

Bentuk lain

```
>aspect(1.25); r=linspace(0,1,50); a=linspace(0,2pi,200)'; z=r*exp(I*a);
>plot2d(exp(z),cgrid=[40,10]):
```

Penjelasan :
Perintah tersebut merupakan perintah untuk menggambar kurva dari fungsi kompleks eksponensial "exp(z)" dalam koordinat kompleks. Dalam perintah tersebut juga menggunakan parameter cgrid dengan nilai [40,10] untuk mengatur grid pada plot.
Dalam sintaks ini,
exp(z) : fungsi eksponensial kompleks yang akan digambar
cgrid=[40,10] : mengatur grid pada plot. cgrid tersebut adalah jumlah garis grid yang akan digunakan pada sumbu x dan sumbu y. Nah di dalam plot ini, akan ada 40 garis grid pada sumbu x dan 10 grid pada sumbu y.

Bentuk lain

```
>r=linspace(0,1,10); a=linspace(0,2pi,40)'; z=r*exp(I*a);
>plot2d(exp(z),>points,>add):
```

Sebuah vektor bilangan kompleks secara otomatis diplot sebagai kurva pada bidang kompleks dengan bagian real dan bagian imajiner.

Penjelasan :
Perintah plot2d di atas adalah perintah untuk menggambar kurva fungsi kompleks dalam koordinat kompleks, namun dengan opsi yang berbeda,
exp(z): fungsi kompleks yang akan digambar
>points : opsi ini mengubah cara plot untuk dilakukan. Dengan menggunakan >points, plot ini akan menggunakan titik-titik diskrit untuk merepresentasikan fungsi ke dalam bentuk titik,titik
>add : sintaks ini menginstrusikan perintah untuk menambahkan plot ini ke plot sebelumnya jika ada.

Contoh

```
>t=linspace(0,2pi,1000); ...
 plot2d(exp(I*t)+exp(10*I*t),r=3):
```

Penjelasan :
Perintah plot2d di atas menggambarkan kurva dari fungsi kompleks yang diberikan dalam koordinat kompleks dengan parameter-parameter tertentu.
Sintaks yang digunakan yaitu,
exp(I*t)+exp(10*I*t) : fungsi kompleks yang akan digambar. Fungsi ini terdiri dari dua bagian yang masing-masing merupakan fungsi kompleks eksponensial. Dengan 10 adalah berapa kali putaran dalam gambar tersebut.
r : parameter r digunakan untuk menentukan rentang nilai dari variabel t. Dalam contoh ini, r=3, yaitu mengatur rentang nilai t dari 3 hingga 3.

Sub Bab 14

Menggambar Daerah Yang Dibatasi Kurva

Plot data benar-benar poligon. Kita juga dapat memplot kurva atau kurva terisi.

Pada subtopik sebelumnya telah kita ketahui dan pelajari bersama bahwa EMT dapat melakukan visualisasi plot mulai dari bentuk ekspresi langsung hingga plot dari fungsi-fungsi. Pada subtopik ini merupakan kelanjutan dari subtopik sebelumnya, bahwa kita dapat membentuk/menggambar daerah dari perpotongan beberapa kurva yang telah didefinisikan. Hal ini dapat bermanfaat untuk membantu dalam menyelesaikan permasalahan dalam matematika, salah satu contohnya seperti optimasi program linear, dimana disajikan beberapa fungsi-fungsi kendala beserta dengan fungsi tujuannya dan perlu divisualisasikan dalam bentuk grafik untuk melihat dimana letak daerah layaknya untuk menentukan nilai optimum.

Dalam EMT ada beberapa perintah yang digunakan untuk menggambar daerah yang dibatasi oleh beberapa kuva, diantaranya yaitu:

- plot2d Digunakan untuk melakukan plotting.
- filled=true Digunakan untuk memberikan isian/arsiran pada daerah/area di bawah kurva saat plotting.
- style="..." Digunakan untuk memilih gaya kurva yang akan digunakan saat plotting. Anda dapat memilih dari beberapa gaya, seperti "#", "/", "\", atau "-". Dan hal ini mempengaruhi tampilan daerah kurva yang terbentuk.
- fillcolor Digunakan untuk menentukan warna isian yang akan digunakan untuk mengiri area di bawah kurva.

Contoh

```
>t=linspace(0,2pi,1000); // parameter for curve
>x=sin(t)*exp(t/pi); y=cos(t)*exp(t/pi); // x(t) and y(t)
>figure(1,2); aspect(16/9)
>figure(1); plot2d(x,y,r=10); // plot curve
>figure(2); plot2d(x,y,r=10,>filled,style="/",fillcolor=red); // fill curve
>figure(0):
```

Penjelasan:

- t=linspace(0,2pi,1000); Pada langkah pertama yaitu mendefinisikan parameter t sebagai serangkaian 1000 titik antara 0 dan 2pi. Parameter t ini akan digunakan sebagai parameter untuk menggambar kurva.
- x=sin(t)*exp(t/pi); y=cos(t)*exp(t/pi); // x(t) and y(t) Kemudian kita definisika dua vektor x dan y yang merupakan koordinat x dan y dari kurva yang akan digambar. Fungsi sin(t)*exp(t/pi) digunakan untuk menghitung komponen x (x(t)), dan cos(t)*exp(t/pi) digunakan untuk menghitung komponen y (y(t)) dari kurva.
- figure(1,2); aspect(16/9) Perintah ini digunakan untuk mengatur tampilan gambar. Perintah figure(1,2) digunakan membuat dua gambar (1 dan 2) dalam satu jendela gambar. Dan perintah aspect(16/9) mengatur rasio aspek gambar menjadi 16:9, yang mempengaruhi bentuk dan ukuran gambar yang akan digambar.
- figure(1); plot2d(x,y,r=10); // membuat plot kurva Perintah ini memilih gambar pertama (1) dan menggunakan perintah plot2d untuk menggambar kurva yang dihitung sebelumnya. Parameter r=10 mengatur lebar garis plot. Ini menghasilkan kurva tanpa adanya isi atau arsiran di dalamnya.
- figure(2); plot2d(x,y,r=10,>filled,style="/",fillcolor=red); // fill curve Selanjutnya pada perintah ini beralih ke gambar kedua (2) dan menggunakan perintah plot2d lagi untuk menggambar kurva yang sama dengan pengisian area di bawahnya. Perintah >filled digunakan untuk mengisi area di bawah kurva, style="/" digunakan untuk mengatur gaya garis menjadi garis miring, dan fillcolor=red digunakan untuk mengatur warna isian menjadi merah.

-figure(0):
Baris perintah ini digunakan untuk mengakhiri gambar dan kembali ke tampilan biasa tanpa gambar. Ini berfungsi untuk menyelesaikan proses penggambaran.

Contoh

```
>x=linspace(0,2pi,1000); plot2d(cos(x),sin(x)*0.5,r=1,>filled,style="\"):
```

Penjelasan:

- x=linspace(0,2pi,100); Mendefinisikan vektor x dengan menggunakan perintah linspace. linspace digunakan untuk membuat vektor dengan 100 titik yang secara merata tersebar antara 0 dan 2phi. Dalam konteks ini, vektor x akan digunakan sebagai parameter saat menggambar kurva.

- plot2d(cos(x),sin(x)*0.5,r=1,>filled,style="\"): Ini merupakan perintah utama yang digunakan untuk menggambar plot. Perintah ini memiliki beberapa parameter sebagai berikut: > cos(x) adalah komponen x dari kurva. Ini adalah hasil dari fungsi kosinus yang dihitung pada vektor x. > sin(x)*0.5 adalah komponen y dari kurva. Ini adalah hasil dari fungsi sinus yang dihitung pada vektor x dan kemudian dikalikan dengan 0,5, yang mengubah amplitudonya. > r=1 mengatur lebar garis plot menjadi 1. > filled digunakan untuk mengisi area di bawah kurva, sehingga menciptakan daerah yang terisi. > style="\" mengatur gaya garis kurva untuk membentuk garis miring yang gunanya menutupi semua bagian kurva dengan garis miring.

Contoh

```
>t=linspace(0,2pi,6); ...
 plot2d(cos(t),sin(t),>filled,style="/",fillcolor=red,r=1.5):
```

Penjelasan:

- t=linspace(0,2pi,6); ... Pada perintah ini, kita definisikan vektor t dengan menggunakan perintah linspace. linspace digunakan untuk membuat vektor dengan 6 titik yang terletak secara merata antara 0 dan 2pi. Dalam konteks ini, vektor t akan digunakan sebagai parameter saat menggambar kurva.

- plot2d(cos(t),sin(t),>filled,style="/",fillcolor=red,r=1.5): Ini adalah perintah utama yang digunakan untuk menggambar plot. Perintah ini memiliki beberapa parameter sebagai berikut: > cos(t) adalah komponen x dari kurva. > sin(t) adalah komponen y dari kurva. > filled digunakan untuk mengisi area di bawah kurva, sehingga menciptakan bentuk yang terisi. Ini berarti daerah di bawah kurva akan diwarnai. > style="/" mengatur gaya garis kurva menjadi garis miring ("/"). > fillcolor=orange mengatur warna isian daerah di bawah kurva menjadi oranye. > r=1.5 mengatur lebar garis plot menjadi 1.5.

Contoh

```
>t=linspace(0,2pi,6); plot2d(cos(t),sin(t),>filled,style="#"):
```

Penjelasan:
- t=linspace(0,2pi,6);
Pada perintah ini, kita definisikan vektor t dengan menggunakan perintah linspace. linspace digunakan untuk membuat vektor dengan 6 titik yang terletak secara merata antara 0 dan 2phi. Dalam konteks ini, vektor t akan digunakan sebagai parameter saat menggambar kurva.

- plot2d(cos(t),sin(t),>filled,style="#"): Ini adalah perintah utama yang digunakan untuk menggambar plot. Perintah ini memiliki beberapa parameter sebagai berikut: > cos(t) adalah komponen x dari kurva. > sin(t) adalah komponen y dari kurva. > filled digunakan untuk mengisi area di bawah kurva, sehingga menciptakan bentuk yang terisi. Ini berarti daerah di bawah kurva akan diisi dengan warna atau pola tertentu. > style="#" mengatur isian kurva menjadi warna solid dengan menggunakan simbol tanda pagar ("#")

Pada contoh ini tidak ada perintah untuk mengatur warna, maka warna yang dihasilkan pada plot ini akan mengikuti pada warna yang disetting pada bagian sebelumnya.

Contoh

Contoh lainnya adalah segi empat, yang kita buat dengan 7 titik pada lingkaran satuan.

```
>t=linspace(0,2pi,7);  ...
 plot2d(cos(t),sin(t),r=1,>filled,style="/",fillcolor=orange):
```

Penjelasan:

- t=linspace(0,2pi,7);: Fungsi linspace digunakan untuk membuat array berisi sejumlah nilai yang merata dalam rentang tertentu. Dalam hal ini, rentangnya adalah dari 0 hingga 2p (dua kali nilai p) dan sebanyak 7 titik akan dihasilkan. Ini akan digunakan sebagai sudut dalam koordinat polar untuk menggambarkan data.
- plot2d(cos(t),sin(t),r=1,>filled,style="/",fillcolor=orange): Ini adalah perintah untuk melakukan plotting data. Terdapat beberapa argumen di sini: > cos(t): Ini adalah nilai kosinus dari setiap elemen dalam array t. Ini akan digunakan sebagai komponen sumbu Y dalam koordinat polar. > sin(t): Ini adalah nilai sinus dari setiap elemen dalam array t. Ini akan digunakan sebagai komponen sumbu X dalam koordinat polar. > r=1: Ini adalah argumen opsional yang menentukan radius plot. Dalam hal ini, radiusnya diatur menjadi 1. > filled: Ini adalah argumen yang menginstruksikan untuk mengisi area di dalam kurva plot. > style="/": Ini adalah argumen yang menentukan gaya garis yang digunakan untuk plot. Di sini, garisnya akan berbentuk garis miring ("/"). > fillcolor=orange: Ini adalah argumen yang menentukan warna pengisian untuk area di dalam kurva plot. Dalam hal ini, warnanya diatur menjadi oren.

Contoh

```
>A=[2,1;1,2;-1,0;0,-1];
>function f(x,y) := max([x,y].A');
>plot2d("f",r=4,level=[0;3],color=red,n=111):
```

Penjelasan:

- A=[2,1;1,2;-1,0;0,-1]; Ini adalah perintah untuk membuat matriks A. Matriks ini memiliki dimensi 4x2, yang berarti memiliki 4 baris dan 2 kolom. Isinya adalah: \begin {bmatrix} 2 \hspace{10pt} 1 \\ 1 \hspace{10pt} 2 \\ \end{bmatrix}
- function f(x,y) := max([x,y].A'); Ini adalah perintah untuk mendefinisikan sebuah fungsi bernama f(x, y). Fungsi ini mengambil dua argumen input, yaitu x dan y. Fungsi ini melakukan operasi berikut:

> [x, y] menghasilkan vektor baris dengan elemen [x, y].
> [x, y].A' adalah perkalian dot (dot product) dari vektor baris [x, y] dengan transpose dari matriks A.
> max([x, y].A') menghitung nilai maksimum dari hasil perkalian dot tersebut.

Dengan kata lain, fungsi `f(x, y)` mengambil vektor `[x, y]` sebagai input, mengalikannya dengan matriks `A`, dan mengembalikan nilai maksimum dari hasil perkalian tersebut.

- plot2d("f",r=4,level=[0;3],color=red,n=111): Ini adalah perintah untuk membuat plot 2D dari fungsi `f(x, y)` yang telah didefinisikan. Rincian perintah ini adalah sebagai berikut:

> "f" adalah nama fungsi yang akan diplot.
> r=4 menentukan rentang plot, yang dalam hal ini adalah [-4, 4] untuk kedua sumbu x dan y.
> level=[0;3] menentukan tingkat kontur (contour levels) yang akan digunakan dalam plot. Ada dua tingkat kontur: 0 dan 3.
> color=green mengatur warna kontur plot menjadi merah.
> n=111 mengendalikan jumlah titik yang digunakan dalam plot.

Hasilnya akan menjadi sebuah grafik kontur 2D dari fungsi `f(x, y)` dengan kontur berwarna merah pada tingkat 0 dan 3, yang mencakup rentang -4 hingga 4 pada kedua sumbu x dan y.

Contoh

```
>t=linspace(0,2pi,1000); x=cos(3*t); y=sin(4*t);
```

```
>plot2d(x,y,<grid,<frame,>filled):
```

Penjelasan:

- t = linspace(0, 2*pi, 1000); Ini adalah perintah untuk membuat vektor t yang berisi 1000 nilai yang merata terdistribusi antara 0 hingga 2pi. Vektor t ini akan digunakan sebagai parameter waktu atau sudut dalam parameterisasi lingkaran. linspace(0, 2*pi, 1000) membuat 1000 titik antara 0 hingga 2pi, memberikan sudut-sudut yang merata di sepanjang satu putaran lingkaran.
- x = cos(3*t); y = sin(4*t); Ini adalah perintah untuk menghitung vektor x dan y yang menggambarkan lintasan dalam koordinat polar.

> x = cos(3*t); menghitung nilai x sebagai hasil dari fungsi kosinus dari 3 kali nilai t. Ini akan menghasilkan osilasi yang lebih cepat pada sumbu x.
> y = sin(4*t); menghitung nilai y sebagai hasil dari fungsi sinus dari 4 kali nilai t. Ini akan menghasilkan osilasi yang lebih cepat pada sumbu y.

- plot2d(x, y, <grid, <frame, >filled); Ini adalah perintah untuk membuat plot dari vektor x dan y. Berikut adalah rincian perintah ini:

x adalah vektor yang digunakan sebagai data untuk sumbu x.
y adalah vektor yang digunakan sebagai data untuk sumbu y.
<grid mengaktifkan garis-garis koordinat (grid) di latar belakang plot, membantu dalam visualisasi.
<frame mengaktifkan bingkai (frame) di sekitar plot.
>filled mengisi area di bawah kurva dengan warna, membuat plot menjadi lebih berwarna.

Contoh

Sebuah vektor interval diplot terhadap nilai x sebagai daerah terisi antara nilai interval bawah dan atas.

Ini dapat berguna untuk memplot kesalahan perhitungan. Tapi itu bisa juga digunakan untuk memplot kesalahan statistik.

```
>t=0:0.1:1; ...
 plot2d(t,interval(t-random(size(t)),t+random(size(t))),style="|");  ...
 plot2d(t,t,add=true):
```

Penjelasan:

- t = 0:0.1:1; Ini adalah perintah untuk membuat vektor t yang berisi nilai-nilai dari 0 hingga 1 dengan interval 0.1. Hasilnya adalah vektor [0, 0.1, 0.2, 0.3, ..., 0.9, 1].
- plot2d(t, interval(t - random(size(t)), t + random(size(t))), style="|"); Ini adalah perintah untuk membuat plot pertama. Rincian perintah ini adalah sebagai berikut:

> interval(t - random(size(t)), t + random(size(t))) adalah interval yang digunakan untuk menggambar "garis" pada plot. Setiap titik pada sumbu x (t) akan dihubungkan oleh dua garis vertikal yang dibuat secara acak di sekitar titik tersebut menggunakan random(size(t)). Hasilnya adalah plot dengan garis-garis vertikal yang mewakili interval acak di sekitar setiap titik pada sumbu x.
> style="|" mengatur gaya plot menjadi garis vertikal ("|").

- plot2d(t, t, add=true); Ini adalah perintah untuk membuat plot kedua dan menambahkannya ke dalam plot yang sudah ada dari perintah sebelumnya. Rincian perintah ini adalah sebagai berikut:

> t adalah sumbu x dan y plot ini, sehingga plot ini akan menjadi plot garis diagonal dengan kemiringan 45 derajat.
> add=true digunakan untuk menambahkan plot ini ke dalam plot sebelumnya, sehingga kedua plot akan ditampilkan dalam satu plot yang sama.

Contoh

Jika x adalah vektor yang diurutkan, dan y adalah vektor interval, maka plot2d akan memplot rentang interval yang terisi dalam bidang. Gaya isian sama dengan gaya poligon.

```
>t=-1:0.01:1; x=~t-0.01,t+0.01~; y=x^3-x;
>plot2d(t,y):
```

Penjelasan:

- t = -1:0.01:1; Ini adalah perintah untuk membuat vektor t yang berisi nilai-nilai dari -1 hingga 1 dengan interval 0.01. Hasilnya adalah vektor t yang berisi nilai-nilai seperti [-1, -0.99, -0.98, ..., 0.99, 1]. Vektor t ini akan digunakan sebagai sumbu x pada plot.
- x = ~t - 0.01, t + 0.01~; Ini adalah perintah yang menghitung vektor x. Tanda ~ digunakan di sini untuk mendefinisikan dua interval, yaitu [~t - 0.01, t + 0.01~]. Ini menghasilkan vektor x yang memiliki dua interval, satu yang kurang dari t - 0.01 dan satu yang lebih dari t + 0.01.
- y = x^3 - x; Ini adalah perintah yang menghitung vektor y sebagai fungsi dari x. Fungsi ini menghitung nilai y dengan memasukkan setiap nilai x ke dalam rumus x^3 - x.
- plot2d(t, y); Ini adalah perintah untuk membuat plot dari fungsi y sebagai fungsi dari t. Rincian perintah ini adalah sebagai berikut: > t adalah sumbu x pada plot, yang berisi vektor t yang telah didefinisikan sebelumnya. > y adalah sumbu y pada plot, yang berisi vektor y yang dihitung dari rumus x^3 - x.

Contoh

```
>expr := "2*x^2+x*y+3*y^4+y"; // define an expression f(x,y)
>plot2d(expr,level=[0;1],style="-",color=blue): // 0 <= f(x,y) <= 1
```

Penjelasan:

- expr := "2*x^2+x*y+3*y^4+y"; Ini adalah perintah untuk mendefinisikan ekspresi matematika yang disimpan dalam variabel expr. Ekspresi ini merupakan suatu fungsi f(x, y) yang tergantung pada dua variabel, yaitu x dan y. Ekspresi ini memiliki bentuk matematika yang terdiri dari berbagai suku, seperti kuadrat dari x, perkalian x*y, kuadrat dari y, dan lainnya.
- plot2d(expr, level=[0;1], style="-", color=blue); Ini adalah perintah untuk membuat plot dari fungsi f(x, y) yang telah didefinisikan sebelumnya. Berikut adalah rincian perintah ini:

> expr adalah ekspresi yang akan digunakan sebagai fungsi yang akan diplotkan. Dalam hal ini, ekspresi 2*x^2+x*y+3*y^4+y adalah fungsi f(x, y) yang telah didefinisikan sebelumnya.

> level=[0;1] mengatur tingkat kontur (contour levels) yang akan digunakan dalam plot. Dalam hal ini, tingkat kontur adalah 0 hingga 1, yang berarti plot akan menunjukkan wilayah di mana f(x, y) memiliki nilai antara 0 hingga 1.

> style="-" mengatur gaya plot menjadi garis berjenis -, yang akan menghasilkan plot kontur.

> color=blue mengatur warna garis plot menjadi biru.

Contoh

```
>plot2d("(x^2+y^2)^2-x^2+y^2",r=1.2,level=[-1;0],style="/"):
```

Penjelasan:

plot2d("(x^2+y^2)^2-x^2+y^2", r=1.2, level=[-1;0], style="/");
Ini adalah perintah untuk membuat plot dari fungsi matematika yang didefinisikan dalam bentuk string: "(x^2+y^2)^2-x^2+y^2". Fungsi ini tergantung pada dua variabel, yaitu x dan y.

(x^2+y^2)^2-x^2+y^2 adalah rumus dari fungsi matematika yang akan diplotkan.

r=1.2 mengatur rentang (range) plot untuk kedua sumbu x dan y. Dalam hal ini, rentangnya adalah [-1.2, 1.2], yang berarti plot akan berada dalam wilayah ini.

level=[-1;0] mengatur tingkat kontur (contour levels) yang akan digunakan dalam plot. Dalam hal ini, ada dua tingkat kontur: -1 dan 0. Ini akan menentukan wilayah kontur dalam plot.

style="/" mengatur gaya plot menjadi garis miring ("/"). Ini akan menghasilkan plot dengan garis-garis miring yang menggambarkan kontur fungsi.

Contoh

```
>plot2d("cos(x)","sin(x)^3",xmin=0,xmax=2pi,>filled,style="/"):
```

Penjelasan:

plot2d("sin(x)^3", "cos(x)", xmin=0, xmax=2*pi, >filled, style="/");
Ini adalah perintah untuk membuat plot dari dua fungsi matematika, yaitu sin(x)^3 dan cos(x), dalam satu plot yang sama. Berikut adalah rincian perintah ini:

"sin(x)^3" adalah ekspresi pertama yang akan diplotkan. Ini adalah fungsi trigonometri sin(x) yang dipangkatkan tiga. Fungsi ini tergantung pada variabel x.

"cos(x)" adalah ekspresi kedua yang akan diplotkan. Ini adalah fungsi trigonometri cos(x). Fungsi ini juga tergantung pada variabel x.

xmin=0 dan xmax=2*pi mengatur rentang (range) plot untuk sumbu x dari 0 hingga 2p. Ini adalah rentang yang akan ditampilkan dalam plot.

>filled mengisi area di bawah kurva fungsi dengan warna, sehingga area di bawah kurva fungsi akan diisi dengan warna.

style="/" mengatur gaya plot menjadi garis miring ("\"). Ini akan menghasilkan plot dengan garis-garis miring.

Sub Bab 15

Menggambar Segi Banyak

Data plot merupakan poligon atau segi banyak. Kita juga dapat membuat kurva atau mengisi kurva.
Fungsi perintah yang digunakan untuk menggambar segi banyak atau poligon.

Membentuk poligon dengan fungsi:
x=linspace(0,2pi,n); plot2d(cos(x),sin(x),r=1,>filled,style="..."):
atau
x=linspace(0,2pi,n); plot2d(sin(x),cos(x),r=1,>filled,style="...",fillcolor=red):

Keterangan
- filled=true, mengisi plot.
- style="...": Pilih dari "#", "/", "\", "\/" dan gaya gaya lainnya.
- fillcolor: untuk memeberikan warna.

Warna isian ditentukan oleh argumen "fillcolor", dan pada <outline opsional mencegah menggambar batas untuk semua gaya kecuali yang default.

Poligon dalam EMT dapat digambar dengan fungsi maksimal. Dengan fungsi maksimal ini, poligon yang dihasilkan dapat berupa poligon tak beraturan.

```
A=[2,1;1,2;-1,0;0,-1];
function f(x,y) := max([x,y].A');
plot2d("f",r=4,level=[0;3],color=green,n=111):
```

Keterangan:
-A adalah titik koordinat dari poligon yang akan dibuat.
-"r" untuk menentukan ukuran bidang koordinat.

Berikut adalah himpunan nilai maksimal dari empat kondisi linear yang kurang dari atau sama dengan 3. Ini merupakan A[k].v<=3 untuk semua baris A. Untuk mendapatkan sudut yang bagus, kita menggunakan n yang relatif besar.

1. Menggambar Segitiga

```
>x=linspace(0,2pi,3); ...
 plot2d(sin(x),cos(x),r=1):
```

Segitiga diatas digambar dari kurva tertutup dengan 3 titik.

Kita dapat membuat segitiga dengan gaya yang berbeda-beda. Seperti pada contoh berikut ini.

```
>x=linspace(0,2pi,3); ...
 plot2d(sin(x),cos(x),>filled,style="/",fillcolor=red,r=1):
>x=linspace(0,2pi,3); ...
 plot2d(sin(x),cos(x),>filled,style="#",fillcolor=blue,r=2):
```

Dua gambar segitiga diatas memiliki gaya yang berbeda, dengan menggunakan fungsi perintah "style=". Gambar segitiga juga dapat dibuat dengan posisi yang berbeda, tergantung pada fungsi yang akan diplot.

2. Menggambar Segiempat

```
>x=linspace(0,2pi,4); ...
 plot2d(cos(x),sin(x),r=1.5):
>x=linspace(0,2pi,4);
>plot2d(cos(x),sin(x),r=2,>filled,outline=1):
```

Gambar diatas merupakan salah satu contoh segiempat yang dapat digambar di EMT. Fungsi perintah yang digunakan masih sama seperti fungsi perintah untuk menggambar segitiga.

Selain fungsi perintah diatas, untuk menggambar segi banyak, dapat menggunakan fungsi maksimum.

```
>A=[2,1;1,2;-1,0;0,-1];
>function f(x,y) := max([x,y].A');
>plot2d("f",r=4,level=[0;3],color=yellow,n=111):
>A=[1,1;-1,1;-1,-1;1,-1];
>function f(x,y) := max([x,y].A');
>plot2d("f",r=1,level=[0;1],color=gray,n=90):
```

Dengan fungsi maksimal ini, kita dapat menggambar segiempat atau segi banyak sebarang.

3. Menggambar Segilima

```
>t=linspace(0,2pi,5); plot2d(sin(t),cos(t),r=1.5):
>t=linspace(0,2pi,5); ...
 plot2d(sin(t),cos(t),r=1.5,>filled,style="\",fillcolor=orange):
>A=[0,5;3,2;1,-4;-1,-4;-3,2];
>function f(x,y) := max([x,y].A');
>plot2d("f",r=1,level=[0;2],color=cyan,n=111):
```

4. Menggambar Segienam

```
>t=linspace(0,2pi,6); ...
 plot2d(cos(t),sin(t),r=1.2):
>t=linspace(0,2pi,6); ...
 plot2d(cos(t),sin(t),>filled,style="/",fillcolor=olive,r=1.2):
```

5. Menggambar dekagon

```
>t=linspace(0,2pi,10); ...
 plot2d(cos(t),sin(t),r=1.2):
>t=linspace(0,2pi,10); ...
```