# EMT untuk Perhitungan Aljabar

Pada notebook ini Anda belajar menggunakan EMT untuk melakukan berbagai perhitungan terkait dengan materi atau topik dalam Aljabar. Kegiatan yang harus Anda lakukan adalah sebagai berikut:
- Membaca secara cermat dan teliti notebook ini;
- Menerjemahkan teks bahasa Inggris ke bahasa Indonesia;
- Mencoba contoh-contoh perhitungan (perintah EMT) dengan cara meng-ENTER setiap perintah EMT yang ada (pindahkan kursor ke baris perintah)
- Jika perlu Anda dapat memodifikasi perintah yang ada dan memberikan keterangan/penjelasan tambahan terkait hasilnya.
- Menyisipkan baris-baris perintah baru untuk mengerjakan soal-soal Aljabar dari file PDF yang saya berikan;
- Memberi catatan hasilnya.
- Jika perlu tuliskan soalnya pada teks notebook (menggunakan format LaTeX).
- Gunakan tampilan hasil semua perhitungan yang eksak atau simbolik dengan format LaTeX. (Seperti contoh-contoh pada notebook ini.)

## Contoh pertama

Menyederhanakan bentuk aljabar:

$$6x^{-3}y^5 \times -7x^2y^{-9}$$

```
>$&6*x^(-3)*y^5*-7*x^2*y^(-9)
```

Menjabarkan:

$$(6x^{-3} + y^5)(-7x^2 - y^{-9})$$

```
>$&showev('expand((6*x^(-3)+y^5)*(-7*x^2-y^(-9))))
```

## Baris Perintah

Baris perintah dalam Euler terdiri atas satu atau beberapa perintah Euler yang diikuti oleh sebuah tanda titik koma ";" atau tanda koma ",". Tanda titik koma menyembunyikan output hasil. Tanda koma setelah peribtah terakhir dapat di abaikan.
Perintah baris berikut hanya akan mencetak hasil ekspresi, bukan penugasan atau format perintah.

```
>r:=2; h:=4; pi*r^2*h/3
```

```
16.7551608191
```

Perintah harus dipisahkan oleh spasi. Baris perintah berikut mencetak dua hasilnya.

```
>pi*2*r*h, %+2*pi*r*h // Ingat tanda % menyatakan hasil perhitungan terakhir sebelumnya
```

```
50.2654824574
100.530964915
```

Baris perintah dijalankan sesuai urutan/susunan yang dibuat pengguna. Sehingga kamua akan mendapat sebuah nilai baru setiap kali menjalankan baris kedua

```
>x := 1;
>x := cos(x) // nilai cosinus (x dalam radian)
```

```
0.155943694764
```

```
>x := cos(x)
```

```
0.857553215846
```

Jika dua baris dihubungkan dengan "...", kedua baris akan selalu dijalankan sekaligus.

```
>x := 1.5; ...
>x := (x+2/x)/2, x := (x+2/x)/2, x := (x+2/x)/2,
```

```
1.41666666667
1.41421568627
1.41421356237
```

Hal ini juga menjadi cara yang baik untuk membagi sebuah perintah panjang yang terdiri dari dua atau lebih baris. Anda bisa menekan Ctrl+Return untuk membagi sebuah baris menjadi dua pada posisi kursor terkini, atau Ctrl+Back untuk menggabungkan baris-baris tersebut.

Untuk melipat semua baris multi-baris, tekan Ctrl+L. Setelah itu, baris-baris berikutnya hanya akan terlihat jika salah satunya mendapatkan fokus. Untuk melipat satu baris multi-baris, mulai baris pertama dengan "%+ "

```
>%+ x=4+5; ...
```

Sebuah baris yang diawali dengan %% akan sepenuhnya tidak terlihat.

```
2
```

Euler mendukung loop dalam baris perintah, asalkan mereka muat dalam satu baris atau beberapa baris. Dalam program, batasan ini tentu tidak berlaku. Untuk informasi lebih lanjut, silakan lihat pengantar berikut.

```
>x=1; for i=1 to 5; x := (x+2/x)/2, end; // menghitung akar 2
```

```
1.5
1.41666666667
1.41421568627
1.41421356237
1.41421356237
```

Tidak masalah menggunakan beberapa baris. Pastikan baris tersebut diakhiri dengan " ...".

```
>x := 1.5; // comments go here before the ...
>repeat xnew:=(x+2/x)/2; until xnew~=x; ...
>   x := xnew; ...
>end; ...
>x,
```

```
1.41421356237
```

Struktur kondisional juga dapat berfungsi.

```
>if E^pi>pi^E; then "Thought so!", endif;
```

```
Thought so!
```

Saat Anda menjalankan perintah, kursor dapat berada di posisi mana saja di baris perintah. Anda bisa kembali ke perintah sebelumnya atau melompat ke perintah berikutnya dengan menggunakan tombol panah. Atau, Anda bisa mengklik bagian komentar di atas perintah untuk berpindah ke perintah tersebut.
Saat Anda memindahkan kursor di sepanjang baris, pasangan tanda kurung buka dan tutup akan disorot. Selain itu, perhatikan baris status. Setelah tanda kurung buka dari fungsi sqrt(), baris status akan menampilkan teks bantuan untuk fungsi tersebut. Jalankan perintah dengan menekan tombol enter.

```
>sqrt(sin(10°)/cos(20°))
```

```
0.429875017772
```

Untuk melihat bantuan untuk perintah terbaru, buka jendela bantuan dengan menekan F1. Di sana, Anda dapat memasukkan teks untuk mencari informasi. Pada baris kosong, bantuan untuk jendela bantuan akan ditampilkan. Anda dapat menekan tombol Escape untuk menghapus baris atau menutup jendela bantuan.
Anda bisa mengklik dua kali pada perintah apa saja untuk membuka bantuan untuk perintah tersebut. Cobalah mengklik dua kali perintah exp di bawah ini di baris perintah.

```
>exp(log(2.5))
```

```
2.5
```

Anda juga dapat menyalin dan menempel di Euler. Gunakan Ctrl-C dan Ctrl-V untuk ini. Untuk menandai teks, seret mouse atau gunakan tombol Shift bersama dengan tombol panah. Selain itu, Anda juga dapat menyalin tanda kurung yang disorot.

## Sintaks Dasar

Euler mengenal fungsi matematika yang biasa. Seperti yang telah Anda lihat di atas, fungsi trigonometri bekerja dalam radian atau derajat. Untuk mengonversi ke derajat, tambahkan simbol derajat (dengan tombol F7) ke nilai, atau gunakan fungsi rad(x). Fungsi akar kuadrat disebut sqrt di Euler. Tentu saja, x^(1/2) juga bisa digunakan.
Untuk menetapkan variabel, gunakan "=" atau ":=". Demi kejelasan, pengantar ini menggunakan bentuk yang terakhir. Spasi tidak masalah. Namun, spasi antara perintah diharapkan ada.
Beberapa perintah dalam satu baris dipisahkan dengan "," atau ";". Titik koma menekan output perintah. Jika ";" tidak ada, maka "," dianggap ada di akhir baris perintah.

```
>g:=9.81; t:=2.5; 1/2*g*t^2
```

```
30.65625
```

EMT uses a programming syntax for expressions. To enter
EMT menggunakan sintaks pemrograman untuk ekspresi. Untuk memasukkan

$$e^2 \cdot \left(\frac{1}{3+4\log(0.6)}+\frac{1}{7}\right)$$

Anda harus menetapkan tanda kurung yang tepat dan menggunakan / untuk pecahan. Perhatikan tanda kurung yang disorot untuk bantuan. Perlu diingat bahwa konstanta Euler e disebut E dalam EMT.

```
>E^2*(1/(3+4*log(0.6))+1/7)
```

```
8.77908249441
```

Untuk menghitung ekspresi yang rumit seperti

$$\left(\frac{\frac{1}{7}+\frac{1}{8}+2}{\frac{1}{3}+\frac{1}{2}}\right)^2 \pi$$

Anda perlu memasukkannya dalam bentuk baris.

```
>((1/7 + 1/8 + 2) / (1/3 + 1/2))^2 * pi
```

```
23.2671801626
```

Hati-hati menempatkan tanda kurung di sekitar sub-ekspresi yang perlu dihitung terlebih dahulu. EMT membantu Anda dengan menyoroti ekspresi yang diakhiri oleh tanda kurung tutup. Anda juga perlu menggunakan nama "pi" untuk huruf Yunani pi.
Hasil dari perhitungan ini adalah angka floating-point. Secara default, hasil ini dicetak dengan akurasi sekitar 12 digit. Dalam baris perintah berikut, kita juga akan belajar bagaimana merujuk ke hasil sebelumnya dalam baris yang sama.

```
>1/3+1/7, fraction %
```

```
0.47619047619
10/21
```

Perintah Euler bisa berupa ekspresi atau perintah primitif. Ekspresi terdiri dari beberapa operator dan fungsi. Jika perlu, ekspresi harus mengandung tanda kurung untuk memaksa urutan eksekusi yang benar. Jika ragu, menambahkan tanda kurung adalah ide yang baik. Perhatikan bahwa EMT menampilkan tanda kurung buka dan tutup saat mengedit baris perintah.

```
>(cos(pi/4)+1)^3*(sin(pi/4)+1)^2
```

```
14.4978445072
```

Beberapa operator numerik di Euler mencakup:

```
+ unary atau operator plus
- unary atau operator minus
*, /
. perkalian matriks
a^b untuk pangkat positif a atau bilangan bulat b (a**b juga
```

berfungsi)

```
n! operator faktorial
```

dan banyak lagi.
Berikut ini beberapa fungsi yang mungkin Anda perlukan. Ada banyak fungsi lainnya.

```
sin,cos,tan,atan,asin,acos,rad,deg
log,exp,log10,sqrt,logbase
bin,logbin,logfac,mod,floor,ceil,round,abs,sign
conj,re,im,arg,conj,real,complex
beta,betai,gamma,complexgamma,ellrf,ellf,ellrd,elle
bitand,bitor,bitxor,bitnot
```

Beberapa perintah memiliki alias, misalnya ln untuk log.

```
>ln(E^2), arctan(tan(0.5))
```

```
2
0.5
```

```
>sin(30°)
```

```
0.5
```

Pastikan untuk menggunakan tanda kurung (kurung bulat) setiap kali ada keraguan mengenai urutan eksekusi! Perhatikan bahwa 2^3^4 berbeda dari (2^3)^4, yang merupakan default di EMT (beberapa sistem numerik melakukannya dengan cara yang berbeda).

```
>2^3^4, (2^3)^4, 2^(3^4)
```

```
2.41785163923e+24
4096
2.41785163923e+24
```

## Bilangan Real

Tipe data utama di Euler adalah angka real. Angka real direpresentasikan dalam format IEEE dengan akurasi sekitar 16 digit desimal.

```
>longest 1/3
```

```
     0.3333333333333333
```

Representasi internal ganda memerlukan 8 byte.

```
>printdual(1/3)
```

```
1.0101010101010101010101010101010101010101010101010101*2^-2
```

```
>printhex(1/3)
```

```
5.5555555555554*16^-1
```

## Strings

String di Euler didefinisikan dengan "...".

```
>"A string can contain anything."
```

```
A string can contain anything.
```

String dapat digabungkan dengan | atau dengan +. Ini juga berlaku untuk angka, yang akan dikonversi menjadi string dalam hal ini.

```
>"The area of the circle with radius " + 2 + " cm is " + pi*4 + " cm^2."
```

```
The area of the circle with radius 2 cm is 12.5663706144 cm^2.
```

Fungsi print juga mengonversi angka menjadi string. Fungsi ini dapat menerima jumlah digit dan jumlah tempat desimal (0 untuk output padat), serta unit jika diperlukan.

```
>"Golden Ratio : " + print((1+sqrt(5))/2,5,0)
```

```
Golden Ratio : 1.61803
```

Ada string khusus none, yang tidak dicetak. String ini dikembalikan oleh beberapa fungsi ketika hasilnya tidak penting. (String ini dikembalikan secara otomatis jika fungsi tidak memiliki pernyataan return.)

```
>none
```

Untuk mengonversi string menjadi angka, cukup evaluasi string tersebut. Ini juga berlaku untuk ekspresi (lihat di bawah).

```
>"1234.5"()
```

```
1234.5
```

Untuk mendefinisikan vektor string, gunakan notasi vector [...].

```
>v:=["affe","charlie","bravo"]
```

```
affe
charlie
bravo
```

Vektor string kosong dilambangkan dengan [none]. Vektor string dapat digabungkan.

```
>w:=[none]; w|v|v
```

```
affe
charlie
bravo
affe
charlie
bravo
```

String dapat mengandung karakter Unicode. Secara internal, string ini menggunakan kode UTF-8. Untuk menghasilkan string seperti itu, gunakan u"..." dan salah satu entitas HTML.
String Unicode dapat digabungkan seperti string lainnya.

```
>u"&alpha; = " + 45 + u"&deg;" // pdfLaTeX mungkin gagal menampilkan secara benar
```

```
 = 45°
```

I

Dalam komentar, entitas yang sama seperti , , dan sebagainya dapat digunakan. Ini bisa menjadi alternatif cepat untuk LaTeX. (Rincian lebih lanjut tentang komentar ada di bawah).

Ada beberapa fungsi untuk membuat atau menganalisis string Unicode. Fungsi strtochar() akan mengenali string Unicode dan menerjemahkannya dengan benar.

```
>v=strtochar(u"&Auml; is a German letter")
```

```
[196,  32,  105,  115,  32,  97,  32,  71,  101,  114,  109,  97,  110,
32,  108,  101,  116,  116,  101,  114]
```

Hasil dari strtochar() adalah vektor angka Unicode. Fungsi sebaliknya adalah chartoutf(), yang mengonversi vektor angka Unicode menjadi string Unicode.

```
>v[1]=strtochar(u"&Uuml;")[1]; chartoutf(v)
```

```
Ü is a German letter
```

Fungsi utf() dapat menerjemahkan string dengan entitas dalam sebuah variabel menjadi string Unicode.

```
>s="We have &alpha;=&beta;."; utf(s) // pdfLaTeX mungkin gagal menampilkan secara benar
```

```
We have =.
```

Memungkinkan juga untuk menggunakan entitas numerik.

```
>u"&#196;hnliches"
```

```
Ähnliches
```

## Nilai Boolean

Nilai Boolean diwakili dengan 1=true atau 0=false di Euler. String dapat dibandingkan, sama seperti angka.

```
>2<1, "apel"<"banana"
```

```
0
1
```

"and" adalah operator "&&" dan "or" adalah operator "||", seperti dalam bahasa C. (Kata-kata "and" dan "or" hanya dapat digunakan dalam kondisi untuk "if").

```
>2<E && E<3
```

```
1
```

Operator Boolean mengikuti aturan dari bahasa matriks.

```
>(1:10)>5, nonzeros(%)
```

```
[0,  0,  0,  0,  0,  1,  1,  1,  1,  1]
[6,  7,  8,  9,  10]
```

Anda dapat menggunakan fungsi nonzeros() untuk mengekstrak elemen-elemen tertentu dari sebuah vektor. Dalam contoh ini, kita menggunakan kondisi isprime(n).

```
>N=2|3:2:99 // N berisi elemen 2 dan bilangan2 ganjil dari 3 s.d. 99
```

```
[2,  3,  5,  7,  9,  11,  13,  15,  17,  19,  21,  23,  25,  27,  29,
31,  33,  35,  37,  39,  41,  43,  45,  47,  49,  51,  53,  55,  57,
59,  61,  63,  65,  67,  69,  71,  73,  75,  77,  79,  81,  83,  85,
87,  89,  91,  93,  95,  97,  99]
```

```
>N[nonzeros(isprime(N))] //pilih anggota2 N yang prima
```

```
[2,  3,  5,  7,  11,  13,  17,  19,  23,  29,  31,  37,  41,  43,  47,
53,  59,  61,  67,  71,  73,  79,  83,  89,  97]
```

## Format Output

Format output default EMT mencetak 12 digit. Untuk memastikan bahwa kita melihat format default, kita dapat mereset formatnya.

```
>defformat; pi
```

```
3.14159265359
```

Secara internal, EMT menggunakan standar IEEE untuk angka ganda dengan akurasi sekitar 16 digit desimal. Untuk melihat jumlah digit penuh, gunakan perintah "longestformat", atau gunakan operator "longest" untuk menampilkan hasil dalam format terpanjang.

```
>longest pi
```

```
      3.141592653589793
```

Berikut ini adalah representasi internal heksadesimal dari angka ganda.

```
>printhex(pi)
```

```
3.243F6A8885A30*16^0
```

Format output dapat diubah secara permanen dengan perintah "format".

```
>format(12,5); 1/3, pi, sin(1)
```

```
0.33333
3.14159
0.84147
```

Format defaultnya adalah format(12).

```
>format(12); 1/3
```

```
0.333333333333
```

Fungsi-fungsi seperti "shortestformat", "shortformat", dan "longformat" bekerja untuk vektor sebagai berikut:
"shortestformat": Menampilkan elemen vektor dengan format terpendek yang memungkinkan untuk menjaga presisi.
"shortformat": Menampilkan elemen vektor dengan format pendek yang mengutamakan keterbacaan sambil tetap menjaga presisi yang cukup.
"longformat": Menampilkan elemen vektor dengan format panjang, memberikan akurasi maksimal dan menampilkan lebih banyak digit desimal.

```
>shortestformat; random(3,8)
```

```
0.66     0.2    0.89    0.28    0.53    0.31    0.44     0.3
0.28    0.88    0.27     0.7    0.22    0.45    0.31    0.91
0.19    0.46   0.095     0.6    0.43    0.73    0.47    0.32
```

Format default untuk skalar adalah format(12). Namun, format ini dapat diubah sesuai kebutuhan.

```
>setscalarformat(5); pi
```

```
3.1416
```

Fungsi "longestformat" juga mengatur format untuk skalar.

```
>longestformat; pi
```

```
3.141592653589793
```

Sebagai referensi, berikut adalah daftar format output yang paling penting:

```
shortestformat shortformat longformat, longestformat
format(length,digits) goodformat(length)
fracformat(length)
defformat
```

shortestformat: Menampilkan angka dengan format terpendek yang memadai untuk presisi.
shortformat: Menampilkan angka dengan format pendek, mengutamakan keterbacaan.
longformat: Menampilkan angka dengan format panjang, memberikan akurasi tinggi.
longestformat: Menampilkan angka dengan format terpanjang, menampilkan semua digit yang tersedia.
format(length,digits): Mengatur format output dengan panjang tertentu dan jumlah digit desimal.
goodformat(length): Menyediakan format yang seimbang antara presisi dan keterbacaan.
fracformat(length): Mengatur format output untuk menampilkan angka dalam bentuk pecahan.
defformat: Mengembalikan format output ke pengaturan default.

Akurasi internal EMT adalah sekitar 16 tempat desimal, sesuai dengan standar IEEE. Angka disimpan dalam format internal ini.

Namun, format output EMT dapat diatur dengan cara yang fleksibel sesuai kebutuhan.

```
>longestformat; pi,
```

```
  3.141592653589793
```

```
>format(10,5); pi
```

```
    3.14159
```

Format default adalah "defformat()".

```
> defformat; // default
```

Ada operator pendek yang mencetak hanya satu nilai. Operator "longest" akan mencetak semua digit valid dari sebuah angka.

```
>longest pi^2/2
```

```
        4.934802200544679
```

Ada juga operator pendek untuk mencetak hasil dalam format pecahan. Kita telah menggunakannya sebelumnya.

```
>fraction 1+1/2+1/3+1/4
```

```
  25/12
```

Karena format internal menggunakan cara biner untuk menyimpan angka, nilai 0.1 tidak akan terwakili secara tepat. Kesalahan ini akan bertambah sedikit demi sedikit, seperti yang terlihat dalam perhitungan berikut.

```
>longest 0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1-1
```

```
   -1.110223024625157e-16
```

Namun, dengan format default "longformat", Anda tidak akan memperhatikan hal ini. Untuk kenyamanan, output dari angka yang sangat kecil akan ditampilkan sebagai 0.

```
>0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1-1
```

```
0
```

## Ekspresi

String atau nama dapat digunakan untuk menyimpan ekspresi matematika, yang dapat dievaluasi oleh EMT. Untuk ini, gunakan tanda kurung setelah ekspresi. Jika Anda berniat menggunakan string sebagai ekspresi, gunakan konvensi untuk menamainya "fx" atau "fxy", dan seterusnya. Ekspresi memiliki prioritas lebih tinggi dibandingkan fungsi.
Variabel global dapat digunakan dalam evaluasi.

```
>r:=2; fx:="pi*r^2"; longest fx()
```

```
      12.56637061435917
```

Parameter akan ditetapkan pada x, y, dan z dalam urutan tersebut. Parameter tambahan dapat ditambahkan menggunakan parameter yang telah ditetapkan.

```
>fx:="a*sin(x)^2"; fx(5,a=-1)
```

```
-0.919535764538
```

Perlu dicatat bahwa ekspresi akan selalu menggunakan variabel global, bahkan jika ada variabel dengan nama yang sama di dalam fungsi. (Jika tidak, evaluasi ekspresi dalam fungsi dapat menghasilkan hasil yang sangat membingungkan bagi pengguna yang memanggil fungsi tersebut.)

```
>at:=4; function f(expr,x,at) := expr(x); ...
>f("at*x^2",3,5) // computes 4*3^2 not 5*3^2
```

```
36
```

Jika Anda ingin menggunakan nilai lain untuk "at" daripada nilai global, Anda perlu menambahkan at=value.

```
>at:=4; function f(expr,x,a) := expr(x,at=a); ...
>f("at*x^2",3,5)
```

```
45
```

Sebagai referensi, perlu dicatat bahwa koleksi panggilan (yang dibahas di tempat lain) dapat berisi ekspresi. Jadi, kita dapat membuat contoh di atas sebagai berikut.

```
>at:=4; function f(expr,x) := expr(x); ...
>f({{"at*x^2",at=5}},3)
```

```
45
```

Ekspresi dalam x sering digunakan seperti halnya fungsi. Perlu dicatat bahwa mendefinisikan fungsi dengan nama yang sama seperti ekspresi simbolik global akan menghapus variabel ini untuk menghindari kebingungan antara ekspresi simbolik dan fungsi.

```
>f &= 5*x;
>function f(x) := 6*x;
>f(2)
```

```
12
```

Sebagai konvensi, ekspresi simbolik atau numerik sebaiknya dinamai fx, fxy, dan seterusnya. Skema penamaan ini sebaiknya tidak digunakan untuk fungsi.

```
>fx &= diff(x^x,x); $&fx
```

$$x^x\,\left(\log x+1\right)$$

Bentuk khusus dari ekspresi memungkinkan penggunaan variabel apa pun sebagai parameter tak bernama untuk evaluasi ekspresi, tidak hanya "x", "y", dan seterusnya. Untuk ini, awali ekspresi dengan "@(variables) ...."

```
>"@(a,b) a^2+b^2", %(4,5)
```

```
@(a,b) a^2+b^2
41
```

Ini memungkinkan untuk memanipulasi ekspresi dalam variabel lain untuk fungsi-fungsi EMT yang memerlukan ekspresi dalam "x".
Cara paling dasar untuk mendefinisikan fungsi sederhana adalah dengan menyimpan rumusnya dalam ekspresi simbolik atau numerik. Jika variabel utama adalah x, ekspresi tersebut dapat dievaluasi seperti fungsi.
Seperti yang terlihat pada contoh berikut, variabel global terlihat selama evaluasi.

```
>fx &= x^3-a*x;  ...
>a=1.2; fx(0.5)
```

```
-0.475
```

Semua variabel lain dalam ekspresi dapat ditentukan dalam evaluasi menggunakan parameter yang ditetapkan.

```
>fx(0.5,a=1.1)
```

```
-0.425
```

Ekspresi tidak harus simbolik. Ini diperlukan jika ekspresi mengandung fungsi-fungsi yang hanya dikenal dalam kernel numerik, bukan di Maxima.

## Matematika Simbolik

EMT melakukan matematika simbolik dengan bantuan Maxima. Untuk detail lebih lanjut, mulailah dengan tutorial berikut, atau telusuri referensi untuk Maxima. Para ahli Maxima harus memperhatikan bahwa ada perbedaan dalam sintaks antara sintaks asli Maxima dan sintaks default ekspresi simbolik di EMT.
Matematika simbolik terintegrasi secara mulus ke dalam Euler dengan menggunakan &. Setiap ekspresi yang diawali dengan & adalah ekspresi simbolik. Ekspresi tersebut dievaluasi dan dicetak oleh Maxima.
Pertama-tama, Maxima memiliki aritmatika "infinite" yang dapat menangani angka-angka yang sangat besar.

```
>$&44!
```

$$2658271574788448768043625811014615890319638528000000000$$

Dengan cara ini, Anda dapat menghitung hasil besar secara tepat. Mari kita hitung:

$$C(44,10)=\frac{44!}{34!\cdot 10!}$$

```
>$& 44!/(34!*10!) // nilai C(44,10)
```

$$2481256778$$

Tentu saja, Maxima memiliki fungsi yang lebih efisien untuk ini (begitu juga dengan bagian numerik dari EMT).

```
>$binomial(44,10) //menghitung C(44,10) menggunakan fungsi binomial()
```

$$2481256778$$

Untuk mempelajari lebih lanjut tentang fungsi tertentu, klik dua kali pada fungsi tersebut. Misalnya, coba klik dua kali pada "&binomial" di baris perintah sebelumnya. Ini akan membuka dokumentasi Maxima sebagaimana disediakan oleh penulis program tersebut.
Anda akan mengetahui bahwa hal berikut ini juga berfungsi:

$$C(x,3)=\frac{x!}{(x-3)!3!}=\frac{(x-2)(x-1)x}{6}$$

```
>$binomial(x,3) // C(x,3)
```

$$\frac{(x-2)\ (x-1)\ x}{6}$$

Jika Anda ingin mengganti x dengan nilai spesifik, gunakan "with".

```
>$&binomial(x,3) with x=10 // substitusi x=10 ke C(x,3)
```

$$120$$

Dengan cara ini, Anda dapat menggunakan solusi dari sebuah persamaan dalam persamaan lainnya.
Ekspresi simbolik dicetak oleh Maxima dalam format 2D. Alasan untuk ini adalah adanya flag simbolik khusus dalam string.
Seperti yang telah Anda lihat dalam contoh sebelumnya dan yang akan datang, jika Anda memiliki LaTeX yang terinstal, Anda dapat mencetak ekspresi simbolik dengan LaTeX. Jika tidak, perintah berikut akan mengeluarkan pesan kesalahan.
Untuk mencetak ekspresi simbolik dengan LaTeX, gunakan $ di depan & (atau Anda dapat menghilangkan &) sebelum perintah. Jangan jalankan perintah Maxima dengan $, jika Anda tidak memiliki LaTeX yang terinstal.

```
>$(3+x)/(x^2+1)
```

$$\frac{x+3}{x^2+1}$$

Ekspresi simbolik diurai oleh Euler. Jika Anda membutuhkan sintaks yang kompleks dalam satu ekspresi, Anda dapat membungkus ekspresi tersebut dalam tanda "...". Menggunakan lebih dari sekadar ekspresi sederhana mungkin saja dilakukan, tetapi sangat tidak disarankan.

```
>&"v := 5; v^2"
```

```
25
```

Untuk melengkapi, perlu dicatat bahwa ekspresi simbolik dapat digunakan dalam program, tetapi harus dibungkus dalam tanda kutip. Selain itu, jauh lebih efektif untuk memanggil Maxima saat waktu kompilasi jika memungkinkan.

```
>$&expand((1+x)^4), $&factor(diff(%,x)) // diff: turunan, factor: faktor
```

$$4\,\left(x+1\right)^3$$

Sekali lagi, % merujuk pada hasil sebelumnya.
Untuk mempermudah, kita simpan solusi ke dalam variabel simbolik. Variabel simbolik didefinisikan dengan "&=".

```
>fx &= (x+1)/(x^4+1); $&fx
```

$$\frac{x+1}{x^4+1}$$

Ekspresi simbolik dapat digunakan dalam ekspresi simbolik lainnya.

```
>$&factor(diff(fx,x))
```

$$\frac{-3\,x^4-4\,x^3+1}{\left(x^4+1\right)^2}$$

Input langsung perintah Maxima juga tersedia. Mulailah baris perintah dengan ::. Sintaks Maxima disesuaikan dengan sintaks EMT (disebut sebagai "compability mode").

```
>&factor(20!)
```

```
                  2432902008176640000
```

```
>::: factor(10!)
```

```
                        8  4  2
                       2  3  5  7
```

```
>:: factor(20!)
```

```
                 18  8  4  2
                2   3  5  7  11 13 17 19
```

Jika Anda seorang ahli Maxima, Anda mungkin ingin menggunakan sintaks asli Maxima. Anda dapat melakukannya dengan :::.

```
>::: av:g$ av^2;
```

```
                          2
                         g
```

```
>fx &= x^3*exp(x), $fx
```

```
                        3  x
                       x  E
```

$$x^3\,e^{x}$$

Variabel seperti itu dapat digunakan dalam ekspresi simbolik lainnya. Perhatikan bahwa dalam perintah berikut, sisi kanan dari &= dievaluasi terlebih dahulu sebelum penugasan ke Fx.

```
>&(fx with x=5), $%, &float(%)
```

```
                          5
                     125 E
```

$$125\,e^5$$

```
18551.64488782208
```

```
>fx(5)
```

```
18551.6448878
```

Untuk evaluasi ekspresi dengan nilai spesifik dari beberapa variabel, Anda dapat menggunakan operator "with".
Baris perintah berikut juga menunjukkan bahwa Maxima dapat mengevaluasi ekspresi secara numerik dengan "float()".

```
>&(fx with x=10)-(fx with x=5), &float(%)
```

```
                10         5
          1000 E   - 125 E
```

```
2.20079141499189e+7
```

```
>$factor(diff(fx,x,2))
```

$$x\,\left(x^2+6\,x+6\right)\,e^{x}$$

Untuk mendapatkan kode Latex untuk suatu ekspresi, Anda dapat menggunakan perintah "tex".

```
>tex(fx)
```

```
x^3\,e^{x}
```

Ekspresi simbolik dapat dievaluasi seperti ekspresi numerik.

```
>fx(0.5)
```

```
0.206090158838
```

Pada ekspresi simbolik, ini tidak dapat berfungsi, karena Maxima tidak mendukungnya. Sebaliknya, gunakan sintaks "with" (bentuk yang lebih baik untuk perintah at(...) pada Maxima.

```
>$&fx with x=1/2
```

$$\frac{\sqrt{e}}{8}$$

Penugasannya dapat juga berupa simbolik.

```
>$&fx with x=1+t
```

$$(t+1)^3\ e^{t+1}$$

Perintah "solve" menyelesaikan ekspresi simbolik untuk variabel pada Maxima. Hasilnya adalah solusi vektor.

```
>$&solve(x^2+x=4,x)
```

$$\left[x=\frac{-\sqrt{17}-1}{2}, x=\frac{\sqrt{17}-1}{2}\right]$$

Bandingkan dengan perintah "solve" numerik pada Euler, yang membutuhkan sebuah nilai awal, dan pilihan nilai target.

```
>solve("x^2+x",1,y=4)
```

```
1.56155281281
```

Nilai numerik dari solusi simbolik dapat dikomputasi dengan evaluasi hasil simbolik. Euler akan membaca tugas x=dll. Jika Anda tidak memerlukan hasil numerik untuk komputasi selanjutnya, Anda juga bisa mengizinkan Maxima untuk menemukan nilai numeriknya.

```
>sol &= solve(x^2+2*x=4,x); $&sol, sol(), $&float(sol)
```

$$\left[x=-\sqrt{5}-1, x=\sqrt{5}-1\right]$$

```
[-3.23607,  1.23607]
```

$$[x=-3.23606797749979, x=1.23606797749979]$$

Untuk memperoleh solusi simbolik yang spesifik,dapat menggunakan "with" dan sebuah index.

```
>$&solve(x^2+x=1,x), x2 &= x with %[2]; $&x2
```

$$\frac{\sqrt{5}-1}{2}$$

$\frac{\sqrt{5}-1}{2}$

Untuk menyelesaikan suatu suistem persamaan, gunakan persamaan vektor. Hasilnya akan berupa penyelesaian vektor

```
>sol &= solve([x+y=3,x^2+y^2=5],[x,y]); $&sol, $&x*y with sol[1]
```

$$2$$

Ekspresi simbolik dapat memiliki flag, yang menunjukkan perlakuan khusus di Maxima. Beberapa flag dapat digunakan sebagai perintah juga, sementara yang lainnya tidak bisa. Flag diakhiri dengan "|" (bentuk yang lebih baik dari "ev(...,flags)")

```
>$& diff((x^3-1)/(x+1),x) //turunan bentuk pecahan
```

$$\frac{3\,x^2}{x+1}-\frac{x^3-1}{\left(x+1\right)^2}$$

```
>$& diff((x^3-1)/(x+1),x) | ratsimp //menyederhanakan pecahan
```

$$\frac{2\,x^3+3\,x^2+1}{x^2+2\,x+1}$$

```
>$&factor(%)
```

$$\frac{2\,x^3+3\,x^2+1}{\left(x+1\right)^2}$$

# Fungsi

Dalam EMT, fungsi adalah program yang didefinisikan dengan perintah "function". Dapat berupa fungsi satu baris atau fungsi multi baris.
Fungsi satu baris dapat berupa numerik atau simbolik. Fungsi numerik satu baris didefinisikan dengan ":="

```
>function f(x) := x*sqrt(x^2+1)
```

Untuk gambaran umum, kami menunjukkan semua definisi yan mungkin untuk fungsi satu-baris. Sebuah fungsi dapat dievaluasi seperti halnya fungsi bawaan Euler lainnya.

```
>f(2)
```

```
4.472135955
```

Fungsi ini juga akan bekerja untuk vektor, mengikuti bahasa matriks Euler, karena ekspresi yang digunakan dalam fungsi tersebut telah divektorisasi.

```
>f(0:0.1:1)
```

```
[0,  0.100499,  0.203961,  0.313209,  0.430813,  0.559017,  0.699714,
0.854459,  1.0245,  1.21083,  1.41421]
```

Fungsi-fungsi dapat dipetakan. Alih-alih ekspresi, kita hanya perlu menyediakan nama fungsi tersebut. Berbeda dengan ekspresi simbolik atau numerik, nama fungsi harus diberikan dalam bentuk string.

```
>solve("f",1,y=1)
```

```
0.786151377757
```

Secara default, jika Anda perlu menimpa fungsibawaan, Anda harus menambahkan kata kunci"overwrite". Menimpa fungsi bawaan berbahaya dan dapat menyebabkan masalah untuk fungsi lain ergantung pada fungsi tersebut.
Anda tetap dapat memanggil fungsi bawaan sebagai "_...", jika fungsi itu masih ada di inti Euler.

```
>function overwrite sin (x) := _sin(x°) // redine sine in degrees
>sin(45)
```

```
0.707106781187
```

Kita sebaiknya menghapus pendefinisian ulang dari sin

```
>forget sin; sin(pi/4)
```

```
0.707106781187
```

## Parameter Default

Fungsi numerik dapat memiliki parameter default

```
>function f(x,a=1) := a*x^2
```

Mengilangkan parameter ini menggunakan nilai default.

```
>f(4)
```

```
16
```

Menyetelnya akan menimpa nilai default.

```
>f(4,5)
```

```
80
```

Parameter yang ditetapkan juga akan menimpa nilai tersebut. Ini digunakan oleh banyak fungsi Euler seperti plot2d, plot3d.

```
>f(4,a=1)
```

```
16
```

Jika sebuah variabel bukan merupakan parameter, maka variabel tersebut harus bersifat global. Fungsi satu baris dapat mengakses variabel global.
Bukan masalah untuk fungsi lain bergantung pada fungsi tersebut.
Anda tetap dapat memanggil fungsi bawaan sebagai "_...", jika fungsi itu masih ada di inti Euler.

```
>function f(x) := a*x^2
>a=6; f(2)
```

```
24
```

Tetapi parameter yang ditetapkan menimpa nilai global.
Jika argumen tidak ada dalam daftar parameter yang telah ditentukan sebelumnya, argumen tersebut harus ditulis dengan ":="!

```
>f(2,a:=5)
```

```
20
```

Fungsi simbolik didefinisikan dengan "&=". Fungsi ini didefinisikan di Euler dan Maxima, dan berfungsi di kedua lingkungan tersebut. Ekspresi yang mendefinisikan fungsi dijalankan melalui Maxima sebelum definisi diterapkan.

```
>function g(x) &= x^3-x*exp(-x); $&g(x)
```

$$x^3-x\,e^{-x}$$

fungsi simbolik dapat digunakan dalam ekpresi simbolik.

```
>$&diff(g(x),x), $&% with x=4/3
```

$$\frac{e^{-\frac{4}{3}}}{3}+\frac{16}{3}$$

$$\frac{e^{-\frac{4}{3}}}{3}+\frac{16}{3}$$

Fungsi simbolik juga dapat digunakan dalam ekspresi numerik. Tentu saja, ini hanya akan berfungsi jika EMT dapat menginterpretasikan semua yang ada di dalam fungsi tersebut.

```
>g(5+g(1))
```

```
178.635099908
```

Fungsi simbolik dapat digunakan untuk mendefinisikan fungsi simbolik atau ekspresi lainnya.

```
>function G(x) &= factor(integrate(g(x),x)); $&G(c) // integrate: mengintegralkan
```

$$\frac{e^{-c}\,\left(c^4\,e^c+4\,c+4\right)}{4}$$

```
>solve(&g(x),0.5)
```

```
0.703467422498
```

Berikut ini juga berfungsi, karena Euler menggunakan ekspresi simbolis dalam fungsi g, jika tidak menemukan variabel simbolik g, dan jika ada fungsi simbolis g.

```
>solve(&g,0.5)
```

```
0.703467422498
```

```
>function P(x,n) &= (2*x-1)^n; $&P(x,n)
```

$$\left(2\,x-1\right)^{n}$$

```
>function Q(x,n) &= (x+2)^n; $&Q(x,n)
```

$$\left(x+2\right)^{n}$$

```
>$&P(x,4), $&expand(%)
```

$$16\,x^4-32\,x^3+24\,x^2-8\,x+1$$

```
>P(3,4)
```

```
625
```

```
>$&P(x,4)+ Q(x,3), $&expand(%)
```

$$16\,x^4-31\,x^3+30\,x^2+4\,x+9$$

```
>$&P(x,4)-Q(x,3), $&expand(%), $&factor(%)
```

$$16\,x^4-33\,x^3+18\,x^2-20\,x-7$$

```
>$&P(x,4)*Q(x,3), $&expand(%), $&factor(%)
```

$$\left(x+2\right)^3\,\left(2\,x-1\right)^4$$

```
>$&P(x,4)/Q(x,1), $&expand(%), $&factor(%)
```

$$\frac{\left(2\,x-1\right)^4}{x+2}$$

$$\frac{16\,x^4}{x+2}-\frac{32\,x^3}{x+2}+\frac{24\,x^2}{x+2}-\frac{8\,x}{x+2}+\frac{1}{x+2}$$

$$\frac{\left(2\,x-1\right)^4}{x+2}$$

```
>function f(x) &= x^3-x; $&f(x)
```

$$x^3-x$$

Dengan tanda &= fungsinya simbolis, dan dapat digunakan dalam ekspresi simbolik lainnya.

```
>$&integrate(f(x),x)
```

$$\frac{x^4}{4}-\frac{x^2}{2}$$

Dengan := fungsinya adalah numerik. Contoh yang baik adalah integral tak tentu seperti

$$f(x)=\int_1^x t^t\,dt,$$

yang tidak dapat dinilai secara simbolis.
Jika kita mendefinisikan kembali fungsi dengan kata kunci "map", fungsi dapat digunakan untuk vektor x. Secara internal, fungsi dipanggil untuk semua nilai x satu kali, dan hasilnya disimpan dalam vektor.

```
>function map f(x) := integrate("x^x",1,x)
>f(0:0.5:2)
```

```
[-0.783431,  -0.410816,  0,  0.676863,  2.05045]
```

Fungsi dapat memiliki nilai default untuk parameter.

```
>function mylog (x,base=10) := ln(x)/ln(base);
```

Sekarang fungsi dapat dipanggil dengan atau tanpa parameter "base".

```
>mylog(100), mylog(2^6.7,2)
```

```
2
6.7
```

Selain itu, dimungkinkan untuk menggunakan parameter yang telah ditetapkan.

```
>mylog(E^2,base=E)
```

```
2
```

Seringkali, kita ingin menggunakan fungsi untuk vektor di suatu tempat, dan untuk elemen individual di tempat lain. Ini memungkinkan dengan parameter vektor.

```
>function f([a,b]) &= a^2+b^2-a*b+b; $&f(a,b), $&f(x,y)
```

$$y^2 - x\,y + y + x^2$$

Fungsi simbolik seperti itu dapat digunakan untuk variabel simbolik.
Namun, fungsi tersebut juga dapat digunakan untuk vektor numerik.

```
>v=[3,4]; f(v)
```

```
17
```

Ada juga fungsi simbolik murni yang tidak dapat digunakan secara numerik.

```
>function lapl(expr,x,y) &&= diff(expr,x,2)+diff(expr,y,2)//turunan parsial kedua
```

```
diff(expr, y, 2) + diff(expr, x, 2)
```

```
>$&realpart((x+I*y)^4), $&lapl(%,x,y)
```

$$0$$

Namun, tentu saja, fungsi simbolik murni dapat digunakan dalam ekspresi simbolik atau dalam definisi fungsi simbolik.

```
>function f(x,y) &= factor(lapl((x+y^2)^5,x,y)); $&f(x,y)
```

$$10\,\left(y^2+x\right)^3\,\left(9\,y^2+x+2\right)$$

Rangkuman :
- &= mendefinisikan fungsi simbolik
- := mendefinisikan fungsi numerik
- &&= mendefinisikan fungsisimbolis murni

## Menyelesaikan Ekspresi

Ekspresi dapat diselesaikan secara numerik dan sibolis.
Untuk menyelesaikan ekspresi sederhana dari suatu variabel, kita dapat menggunakan fungsi solve(). Perlu nilai awal untuk memulai perncarian. Secara internal, solve() menggunakan metode secant.

```
>solve("x^2-2",1)
```

```
1.41421356237
```

Ini juga berfungsi untuk ekspresi simbolis. Ambil fungsi berikut.

```
>$&solve(x^2=2,x)
```

$$\left[ x=-\sqrt{2} , x=\sqrt{2} \right]$$

```
>$&solve(x^2-2,x)
```

$$\left[ x=-\sqrt{2} , x=\sqrt{2} \right]$$

```
>$&solve(a*x^2+b*x+c=0,x)
```

$$\left[ x=\frac{-\sqrt{b^2-4\,a\,c}-b}{2\,a} , x=\frac{\sqrt{b^2-4\,a\,c}-b}{2\,a} \right]$$

```
>$&solve([a*x+b*y=c,d*x+e*y=f],[x,y])
```

$$\left[\left[x=\frac{b\,f-c\,e}{b\,d-a\,e} , y=\frac{c\,d-a\,f}{b\,d-a\,e}\right]\right]$$

```
>px &= 4*x^8+x^7-x^4-x; $&px
```

$$4\,x^8+x^7-x^4-x$$

Sekarang kita mencari titik, dimana polinomialnya adalah 2. Dalam solve(), nilai target default y=0 dapat diubah dengan variabel yang ditetapkan
Kami menggunakan y=2 dan memeriksa dengan mengevaluasi polinomial pada hasil sebulumnya.

```
>solve(px,1,y=2), px(%)
```

```
0.966715594851
2
```

Menyelesaikan ekspresi simbolik dalam bentuk simbolik mengembalikan daftar solusi. Kami menggunakan penyelesaian simbolik solve() yang disediakan oleh Maxima

```
>sol &= solve(x^2-x-1,x); $&sol
```

$$\left[x=\frac{1-\sqrt{5}}{2} , x=\frac{\sqrt{5}+1}{2}\right]$$

Cara termudah untuk mendapatkan nilai numerik adalah dengan mengevaluasi solusi numerik seperti halnya sebuah ekspresi

```
>longest sol()
```

```
    -0.6180339887498949       1.618033988749895
```

Untuk menggunakan solusi secara simbolik dalam ekspresi lain, cara termudah adalah dengan menggunakan "with".

```
>$&x^2 with sol[1], $&expand(x^2-x-1 with sol[2])
```

$$0$$

Menyelesaikan sistem persamaan secara simbolik dapat dilakukan dengan menggunakan vektor persamaan dan pemecah simbolik solve(). Hasilnya adalah daftar dari daftar persamaan.

```
>$&solve([x+y=2,x^3+2*y+x=4],[x,y])
```

$$[[x=-1,y=3]\,,[x=1,y=1]\,,[x=0,y=2]]$$

Fungsi f() dapat mengakses variabel global. Tetapi seringkali kita ingin menggunakan parameter lokal

$$a^x - x^a = 0.1$$

dengan a=3.

```
>function f(x,a) := x^a-a^x;
```

Salah satu cara untuk memalui tambahan parameter ke f() adalah menggunakann list dengan nama fungsi dan parameternya (cara lain dengan parameter titik koma).

```
>solve({{"f",3}},2,y=0.1)
```

```
2.54116291558
```

Ini juga berfunsi dengan beberapa ekspresi. Tetapi, daftar elemen yang telah dinamai harus digunakan. (List lebih lanjut ada di tutorial tentang sintaks EMT)

```
>solve({{"x^a-a^x",a=3}},2,y=0.1)
```

```
2.54116291558
```

## Menyelesaikan Pertidaksamaan

Untuk menyelesaikan pertidaksamaan, EMT tidak akan dapat melakukannya, melainkan dengan bantuan Maxima, artinya secara eksak (simbolik). Perintah Maxima yang digunakan adalah fourier_elim(), yang harus dipanggil dengan perintah "load(fourier_elim)" terlebih dahulu.

```
>&load(fourier_elim)
```

```
        C:/Program Files/Euler x64/maxima/share/maxima/5.35.1/share/f\
ourier_elim/fourier_elim.lisp
```

```
>$&fourier_elim([x^2 - 1>0],[x]) // x^2-1 > 0
```

$$[1 < x] \vee [x < -1]$$

```
>$&fourier_elim([x^2 - 1<0],[x]) // x^2-1 < 0
```

$$[-1 < x, x < 1]$$

```
>$&fourier_elim([x^2 - 1 # 0],[x]) // x^-1 <> 0
```

$$[-1 < x, x < 1] \vee [1 < x] \vee [x < -1]$$

```
>$&fourier_elim([x # 6],[x])
```

$$[x < 6] \vee [6 < x]$$

```
>$&fourier_elim([x < 1, x > 1],[x]) // tidak memiliki penyelesaian
```

$$emptyset$$

```
>$&fourier_elim([minf < x, x < inf],[x]) // solusinya R
```

$$universalset$$

```
>$&fourier_elim([x^3 - 1 > 0],[x])
```

$$\left[1 < x, x^2 + x + 1 > 0\right] \vee \left[x < 1, -x^2 - x - 1 > 0\right]$$

```
>$&fourier_elim([cos(x) < 1/2],[x]) // ??? gagal
```

$$[1 - 2\cos x > 0]$$

```
>$&fourier_elim([y-x < 5, x - y < 7, 10 < y],[x,y]) // sistem pertidaksamaan
```

$$[y - 5 < x, x < y + 7, 10 < y]$$

```
>$&fourier_elim([y-x < 5, x - y < 7, 10 < y],[y,x])
```

$$[max(10, x - 7) < y, y < x + 5, 5 < x]$$

```
>$&fourier_elim((x + y < 5) and (x - y >8),[x,y])
```

$$\left[y + 8 < x, x < 5 - y, y < -\frac{3}{2}\right]$$

```
>$&fourier_elim(((x + y < 5) and x < 1) or  (x - y >8),[x,y])
```

$$[y+8<x] \vee [x<min\,(1,5-y)]$$

```
>&fourier_elim([max(x,y) > 6, x # 8, abs(y-1) > 12],[x,y])
```

```
          [6 < x, x < 8, y < - 11] or [8 < x, y < - 11]
   or [x < 8, 13 < y] or [x = y, 13 < y] or [8 < x, x < y, 13 < y]
   or [y < x, 13 < y]
```

```
>$&fourier_elim([(x+6)/(x-9) <= 6],[x])
```

$$[x=12] \vee [12<x] \vee [x<9]$$

## Bahasa Matriks

Dokumentasi inti EMT berisi diskusi terperinci tentang bahasa matriks Euler.
Vektor dan matriks dimasukkan dalam tanda kurung siku, elemen dibatasi koma, baris dibatasi titik koma.

```
>A=[1,2;3,4]
```

```
              1             2
              3             4
```

```
>C=[1,2,3,4;5,6,7,8]
```

```
              1             2             3             4
              5             6             7             8
```

Produk matriks dilambangkan dengan titik.

```
>b=[3;4]
```

```
              3
              4
```

```
>b' // transpose b
```

```
  [3,  4]
```

```
>inv(A) //inverse A
```

```
            -2             1
           1.5          -0.5
```

```
>A.b //perkalian matriks
```

```
            11
            25
```

```
>A.inv(A)
```

```
             1             0
             0             1
```

The main point of a matrix language is that all functions and operators work element for element.
Poin utama dari bahasa matriks adalah bahwa semua fungsi dan operator berfungsi elemen demi elemen.

```
>A.A
```

```
             7            10
            15            22
```

```
>A^2 //perpangkatan elemen2 A
```

```
             1             4
             9            16
```

```
>A.A.A
```

```
            37            54
            81           118
```

```
>power(A,3) //perpangkatan matriks
```

```
            37            54
            81           118
```

```
>A/A //pembagian elemen-elemen matriks yang seletak
```

```
             1             1
             1             1
```

```
>A/b //pembagian elemen2 A oleh elemen2 b kolom demi kolom (karena b vektor kolom)
```

```
       0.333333      0.666667
           0.75             1
```

```
>A\b // hasilkali invers A dan b, A^(-1)b
```

```
             -2
            2.5
```

```
>inv(A).b
```

```
             -2
            2.5
```

```
>A\A   //A^(-1)A
```

```
              1             0
              0             1
```

```
>inv(A).A
```

```
              1             0
              0             1
```

```
>A*A //perkalin elemen-elemen matriks seletak
```

```
              1             4
              9            16
```

Ini bukan perkalian matriks, tetapi perkalian elemen demi elemen. Hal yang sama berlaku untuk vektor.

```
>b^2 // perpangkatan elemen-elemen matriks/vektor
```

```
              9
             16
```

jika salah satu operan adalah vektor atau skalar, maka dijabarkan dengan cara biasa.

```
>2*A
```

```
              2             4
              6             8
```

Misalnya, jika operan adalah kolom vektor, maka elemennya diterapkan ke semua baris A

```
>[1,2]*A
```

```
                  1                 4
                  3                 8
```

Jika itu adalah vektor baris, maka diterapkan pada semua kolom A

```
>A*[2,3]
```

```
                  2                 6
                  6                12
```

Anda bisa membayangkan perkalian ini seolah-olah vektor baris v telah diduplikasi untuk membentuk matriks dengan ukuran yang sama seperti A

```
>dup([1,2],2) // dup: menduplikasi/menggandakan vektor [1,2] sebanyak 2 kali (baris)
```

```
                  1                 2
                  1                 2
```

```
>A*dup([1,2],2)
```

```
                  1                 4
                  3                 8
```

Ini juga berlaku untuk dua vektor dimana satu adalah vektor baris dan yang lainnya adalah vektor kolom. Kami menghitung i*j untuk i,j dari 1 hingga 5. Caranya adalah dengan mengalikan 1:5 dengan transposnya. Bahasa matriks Euler secara otomatis menghasilkan tabel nilai.

```
>(1:5)*(1:5)' // hasilkali elemen-elemen vektor baris dan vektor kolom
```

```
                  1                 2                 3                 4                 5
                  2                 4                 6                 8                10
                  3                 6                 9                12                15
                  4                 8                12                16                20
                  5                10                15                20                25
```

Sekali lagi, ingat bahwa ini bukan perkalian matriks!

```
>(1:5).(1:5)' // hasilkali vektor baris dan vektor kolom
```

```
  55
```

```
>sum((1:5)*(1:5)) // sama hasilnya
```

```
  55
```

Bahkan operator seperti < atau == bekerja dengan cara yang sama.

```
>(1:10)<6 // menguji elemen-elemen yang kurang dari 6
```

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

misalnya, kita dapat menghitung jumlah elemen yang memenuhi kondisi tertetu dengan fungsi sum().

```
>sum((1:10)<6) // banyak elemen yang kurang dari 6
```

```
5
```

Euler has comparison operators, like "==", which checks for equality.
Euler memiliki operator perbandingan, seperti '==", yang menunjukkan persamaan.
Kita mendapatkan vektor yang terdiri dari 0 dan 1, di mana 1 menunjukkan benar (true).

```
>t=(1:10)^2; t==25 //menguji elemen2 t yang sama dengan 25 (hanya ada 1)
```

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

Dari vektor semacam itu, "nonzeros" memilih elemen-elemen yang tidak bernilai nol.
Dalam kasus ini, kita mendapatkan indeks dari semua elemen yang lebih besar dari 50.

```
>nonzeros(t>50) //indeks elemen2 t yang lebih besar daripada 50
```

```
[8,  9,  10]
```

Tentu saja, kita dapat menggunakan vektor indeks ini untuk mendapatkan nilai yang sesuai dalam

```
>t[nonzeros(t>50)] //elemen2 t yang lebih besar daripada 50
```

```
[64,  81,  100]
```

Sebagai contoh, mari kita cari semua kuadrat dari angka 1 hingga 1000 yang memenuhi syarat 5 modulo 11 dan 3 modulo 13.

```
>t=1:1000; nonzeros(mod(t^2,11)==5 && mod(t^2,13)==3)
```

```
[4,  48,  95,  139,  147,  191,  238,  282,  290,  334,  381,  425,
433,  477,  524,  568,  576,  620,  667,  711,  719,  763,  810,  854,
862,  906,  953,  997]
```

EMT tidak sepenuhnya efektif untuk perhitungan bilangan bulat. Secara internal, ia menggunakan floating point presisi ganda. Namun, sering kali masih sangat berguna.
Kita dapat memeriksa keprimaan. Mari kita cari tahu berapa banyak kuadrat ditambah 1 yang merupakan bilangan prima.

```
>t=1:1000; length(nonzeros(isprime(t^2+1)))
```

```
112
```

Fungsi nonzeros() hanya bekerja untuk vektor. Untuk matriks, ada fungsi mnonzeros().

```
>seed(2); A=random(3,4)
```

```
      0.765761      0.401188      0.406347      0.267829
       0.13673      0.390567      0.495975      0.952814
      0.548138      0.006085      0.444255      0.539246
```

Fungsi mnonzeros() mengembalikan indeks dari elemen-elemen yang tidak bernilai nol.

```
>k=mnonzeros(A<0.4) //indeks elemen2 A yang kurang dari 0,4
```

```
             1             4
             2             1
             2             2
             3             2
```

Indeks-indeks ini dapat digunakan untuk mengatur nilai elemen-elemen tersebut.

```
>mset(A,k,0) //mengganti elemen2 suatu matriks pada indeks tertentu
```

```
      0.765761      0.401188      0.406347             0
             0             0      0.495975      0.952814
      0.548138             0      0.444255      0.539246
```

Fungsi mset() juga dapat digunakan untuk mengatur elemen-elemen pada indeks tertentu dengan entri dari matriks lain.

```
>mset(A,k,-random(size(A)))
```

```
      0.765761      0.401188      0.406347     -0.126917
     -0.122404     -0.691673      0.495975      0.952814
      0.548138     -0.483902      0.444255      0.539246
```

Dan memungkinkan untuk mendapatkan elemen-elemen tersebut dalam sebuah vektor.

```
>mget(A,k)
```

```
 [0.267829,  0.13673,  0.390567,  0.006085]
```

Fungsi lain yang berguna adalah "extrema", yang mengembalikan nilai minimal dan maksimal di setiap baris matriks serta posisi-posisi mereka.

```
>ex=extrema(A)
```

```
      0.267829             4      0.765761             1
       0.13673             1      0.952814             4
      0.006085             2      0.548138             1
```

Kita dapat menggunakan ini untuk mengekstrak nilai maksimal di setiap baris

```
>ex[,3]'
```

```
 [0.765761,  0.952814,  0.548138]
```

Ini, tentu saja, sama dengan fungsi max().

```
>max(A)'
```

```
 [0.765761,  0.952814,  0.548138]
```

Namun, dengan mget(), kita dapat mengekstrak indeks dan menggunakan informasi ini untuk mengambil elemen-elemen pada posisi yang sama dari matriks lain.

```
>j=(1:rows(A))'|ex[,4], mget(-A,j)
```

```
             1             1
             2             4
             3             1
 [-0.765761,  -0.952814,  -0.548138]
```

## Funsi Matriks Lain (Menyusun Matriks)

Untuk menyusun sebuah matriks, kita bisa menumpuk satu matriks ke ats matriks yang lain. Jika keduanya tidak memiliki jumlah kolom yang sama, kolom yang lebih pendek akan diisi dengan 0

```
>v=1:3; v_v
```

```
             1             2             3
             1             2             3
```

Demikian juga, kita dapat melampirkan matriks ke yang lain sisi demi sisi, jika keduanya memiliki jumlah baris yang sama.

```
>A=random(3,4); A|v'
```

```
      0.032444     0.0534171      0.595713      0.564454             1
       0.83916      0.175552      0.396988       0.83514             2
     0.0257573      0.658585      0.629832      0.770895             3
```

Jika mereka tidak punya jumlah baris yan sama, matriks yang lebih pendek diisi dengan 0.
Ada pengecualian untuk aturan ini. Bilangan real yang dilampirkan pada matriks akan digunakan sebagai kolom yang diisi dengan bilangan real tersebut.

```
>A|1
```

```
      0.032444     0.0534171      0.595713      0.564454             1
       0.83916      0.175552      0.396988       0.83514             1
     0.0257573      0.658585      0.629832      0.770895             1
```

Dimungkinkan untuk membuat matriks vektor baris dan vektor kolom

```
>[v;v]
```

```
1          2          3
1          2          3
```

```
>[v',v']
```

```
1          1
2          2
3          3
```

Tujuan utamanya adalah untuk mengintrepetasikan ekpresi vektor untuk vektor kolom.

```
>"[x,x^2]"(v')
```

```
1          1
2          4
3          9
```

Untuk mendapatkan ukuran A, kita dapat menggunakan fungsi berikut.

```
>C=zeros(2,4); rows(C), cols(C), size(C), length(C)
```

```
2
4
[2,  4]
4
```

Untuk vektor, memakai length().

```
>length(2:10)
```

```
9
```

Ada banyak fungsi lain, yang menggeneralisasi matriks.

```
>ones(2,2)
```

```
1          1
1          1
```

Fungsi "ones" bisa juga digunakan untuk satu parameter. Untuk mendapatkan sebuah vektor dengan bilangan lain selain 1, gunakan cara berikut.

```
>ones(5)*6
```

```
[6,  6,  6,  6,  6]
```

Matriks bilangan acak juga bisa digeneralisasi dengan "random"(Distribusi tetap ) atau "normal" (Distribusi Gauß)

```
>random(2,2)
```

```
        0.66566      0.831835
          0.977      0.544258
```

Berikut ini adalah fungsi lain yang berguna, dengan menyusun kembali struktur elemen matris ke dalam matriks lain.

```
>redim(1:9,3,3) // menyusun elemen2 1, 2, 3, ..., 9 ke bentuk matriks 3x3
```

```
              1             2             3
              4             5             6
              7             8             9
```

Dengan fungsi berikut, kita bisa menggunakannya dan fungsi dup untuk menuliskan fungsi rep(), yang mengulang vektor sebanyak n kali.

```
>function rep(v,n) := redim(dup(v,n),1,n*cols(v))
```

Mari kita uji.

```
>rep(1:3,5)
```

```
  [1,  2,  3,  1,  2,  3,  1,  2,  3,  1,  2,  3,  1,  2,  3]
```

Fungsi multdup() menggandakan elemen vektor.

```
>multdup(1:3,5), multdup(1:3,[2,3,2])
```

```
  [1,  1,  1,  1,  1,  2,  2,  2,  2,  2,  3,  3,  3,  3,  3]
  [1,  1,  2,  2,  2,  3,  3]
```

Fungsi flipx() dan flipy() mengembalikan susunan baris atau kolom matriks.
Yaitu, fungsi flipx() membalik secara horisontal.

```
>flipx(1:5) //membalik elemen2 vektor baris
```

```
  [5,  4,  3,  2,  1]
```

Untuk rotasi, Euler memiliki rotleft() dan rotright().

```
>rotleft(1:5) // memutar elemen2 vektor baris
```

```
  [2,  3,  4,  5,  1]
```

Sebuah fungsi khusus yaitu drop(v, i), yang menghapus elemen dengan indeks yang ada di i dari vektor v.

```
>drop(10:20,3)
```

```
  [10,  11,  13,  14,  15,  16,  17,  18,  19,  20]
```

Perhatikan bahwa vektor i dalam drop(v, i) merujuk pada indeks elemen dalam v, bukan nilai elemen tersebut. Jika Anda ingin menghapus elemen, Anda perlu menemukan elemen-elemen tersebut terlebih dahulu. Fungsi indexof(v, x) dapat digunakan untuk menemukan elemen dalam vektor yang sudah terurut.

```
>v=primes(50), i=indexof(v,10:20), drop(v,i)
```

```
[2,  3,  5,  7,  11,  13,  17,  19,  23,  29,  31,  37,  41,  43,  47]
[0,  5,  0,  6,  0,  0,  0,  7,  0,  8,  0]
[2,  3,  5,  7,  23,  29,  31,  37,  41,  43,  47]
```

Seperti yang Anda lihat, tidak ada masalah untuk menyertakan indeks yang di luar rentang (seperti 0), indeks ganda, atau indeks yang tidak terurut.

```
>drop(1:10,shuffle([0,0,5,5,7,12,12]))
```

```
[1,  2,  3,  4,  6,  8,  9,  10]
```

Ada beberapa fungsi khusus untuk mengatur diagonal atau untuk menghasilkan matriks diagonal. Kita mulai dengan matriks identitas.

```
>A=id(5) // matriks identitas 5x5
```

```
1        0        0        0        0
0        1        0        0        0
0        0        1        0        0
0        0        0        1        0
0        0        0        0        1
```

Kemudian kita atur diagonal bawah (-1) dengan 1:4

```
>setdiag(A,-1,1:4) //mengganti diagonal di bawah diagonal utama
```

```
1        0        0        0        0
1        1        0        0        0
0        2        1        0        0
0        0        3        1        0
0        0        0        4        1
```

Perhatikan bahwa kita tidak mengubah matriks A. Kita mendapatkan matriks baru sebagai hasil dari setdiag(). Berikut adalah fungsi yang mengembalikan matriks tri-diagonal.

```
>function tridiag (n,a,b,c) := setdiag(setdiag(b*id(n),1,c),-1,a); ...
>tridiag(5,1,2,3)
```

```
2        3        0        0        0
1        2        3        0        0
0        1        2        3        0
0        0        1        2        3
0        0        0        1        2
```

Diagonal matriks dapat di ekstrak dari matriks. Untuk mendemonstrasikannya, kita susun ulang vektor 1:9 menjadi matriks 3x3.

```
>A=redim(1:9,3,3)
```

```
              1             2             3
              4             5             6
              7             8             9
```

Sekarang kita dapat mengekstraksi diagonalnya.

```
>d=getdiag(A,0)
```

```
  [1,  5,  9]
```

Misalnya, kita dapat membagi matrisks dengan diagonalnya. Bahasa matriks memastikan bahwa vektor kolom d diterapkan ke matriks baris per baaris.

```
>fraction A/d'
```

```
         1         2         3
       4/5         1       6/5
       7/9       8/9         1
```

## Vektorisasi

Hampir semua fungsi di Euler bekerja pada input matriks dan vektor juga, kapanpun hal inimasuk akal. Misalnya, fungsi sqrt() menghitun akar kuadrat dari semua elemen vektor atau matriks.

```
>sqrt(1:3)
```

```
  [1,  1.41421,  1.73205]
```

Jadi, kamu dapat dengan mudah membuat tabel nilai. Ini salah satu cara untuk memplot fungsi (alternatifnya menggunakan ekspresi).

```
>x=1:0.01:5; y=log(x)/x^2; // terlalu panjang untuk ditampikan
```

Dengan ini dan operator koma a:delta:b, vektor nilai fungsi dapat dihasilkan dengan baik.
Dalam contoh berikut, kita menghasilkan vektor nilai t[i] dengan jarak 0.1 dari -1 samapi 1. Lalu kita menghasilkan vektor nilai fungsi.

$$s = t^3 - t$$

```
>t=-1:0.1:1; s=t^3-t
```

```
  [0,  0.171,  0.288,  0.357,  0.384,  0.375,  0.336,  0.273,  0.192,
  0.099,  0,  -0.099,  -0.192,  -0.273,  -0.336,  -0.375,  -0.384,
  -0.357,  -0.288,  -0.171,  0]
```

EMT menjabarkan operator untuk skalar, vektor, dan matriks dengan cara yang jelas.
Misalnya, vektor kolom dikalikan dengan vektor baris akan memperluas menjadi matriks ketika sebuah operator diterapkan. Dalam contoh beriku, adalah transposisi vektor(vektor kolom).

```
>shortest (1:5)*(1:5)'
```

```
         1         2         3         4         5
         2         4         6         8        10
         3         6         9        12        15
         4         8        12        16        20
         5        10        15        20        25
```

Perhatikan, ini sangat berbeda dengan hasil kali matriks. Hasil kali matriks ditandai dengan sebuah titik "." pada EMT.

```
>(1:5).(1:5)'
```

```
  55
```

Secara default, vektor baris dicetak pada format yang lengkap.

```
>[1,2,3,4]
```

```
  [1,   2,   3,   4]
```

Untuk matriks, operator khusus "." Menunjukkan perkalian matriks, dan A' menunjukkan transpose matriks. Matriks A berukuran 1x1 dapat berbentuk seperti hanlya bilangan real.

```
>v:=[1,2]; v.v', %^2
```

```
  5
  25
```

Untuk men-transpos sebuah matriks, kita menggunakan tanda apostrof (')

```
>v=1:4; v'
```

```
                1
                2
                3
                4
```

Jadi, kita dapat menghitung matriks A dikali vektor b.

```
>A=[1,2,3,4;5,6,7,8]; A.v'
```

```
               30
               70
```

Perhatikan bahwa v tetap sebuah vektor baris. Jadi, v'.v berbeda dengan v.v'.

```
>v'.v
```

| | | | |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| 2 | 4 | 6 | 8 |
| 3 | 6 | 9 | 12 |
| 4 | 8 | 12 | 16 |

v.v' menghitung norma Dari v kuadrat untuk vektor baris v. Hasilnya adalah vektor 1x1, yang bekerja seperti bilangan real.

```
>v.v'
```

```
30
```

Ada juga fungsi "norm" (beserta banyak lagi fungsi Aljabar Linier yang lain)

```
>norm(v)^2
```

```
30
```

Operator-operator dan fungsi-fungsi mengikuti bahasa matriks Euler.
Berikut ini rangkuman aturannya :
- Sebuah fungsi yang diterapkan pada sebuah vektor atau matriks diterapkan untuk setiap elemen.
- Sebuah operator yang mengoperasikan Dua matriks berukuran sama diterapkan secara berpasangan dengan elemen matriks
- Jika Dua matriks berbeda dimensi, keduanya dijabarkan/diperluas dengan cara yang praktis, sehingga mempunyai ukuran yang sama

Misalnya, sebuah nilai skalar dikalikan vektor akan mengalikan nilai setiap elemen pada vektor. Atausebuah matriks dikalikan sebuah vektor (dengan * bukan .) melebarkan/meluaskan vektor ke ukuran maaatriks dengan menggandakannya.
Berikut ini kasus khusus dengan operator ^.

```
>[1,2,3]^2
```

```
[1,  4,  9]
```

Ini adalah kasus yang lebih rumit. Sebuah vektor baris dikalikan dengan vektor kolom menjabarkan keduanya dengan menggandakannya.

```
>v:=[1,2,3]; v*v'
```

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 2 | 4 | 6 |
| 3 | 6 | 9 |

Perhatikan bahwa hasil perkalian skalar menggunakan perkalian matriks, bukan "*"!

```
>v.v'
```

```
14
```

Ada banyak fungsi untuk matriks. Kami berikan daftar singkat. Anda harus memeriksa dokumentasi atau informasi lebih banyak pada perintah-perintah ini.

```
sum,prod : menghitung jumlah dan hasil kali baris
cumsum,cumprod : melakukan hal sama secara kumulatif
menghitung nilai extrem tiap baris
extrema : mengganti vektor dengan info ekstrem nya
diag(A,i) : mengembalikan diagonal ke i
setdiag(A,i,v):  mengatur diagonal ke i
id(n) : matriks identitas
det(A) : determinan matriks
charpoly(A) : karakteristik polinomial
eigenvalues(A) : nilai eigen
```

```
>v*v, sum(v*v), cumsum(v*v)
```

```
[1,  4,  9]
14
[1,  5,  14]
```

Operator ":" menghasilkan vektor baris berjarak sama, dipilih dengan langkah ukurannya

```
>1:4, 1:2:10
```

```
[1,  2,  3,  4]
[1,  3,  5,  7,  9]
```

Untuk menggabungkan matriks dan vektor ada operator "|" dan"_".

```
>[1,2,3]|[4,5], [1,2,3]_1
```

```
[1,  2,  3,  4,  5]
            1             2             3
            1             1             1
```

Elemen matriks dintunjukkan dengan "A[i,j]".

```
>A:=[1,2,3;4,5,6;7,8,9]; A[2,3]
```

```
6
```

Untuk vektor baris atau kolom, v[i] merupakan elemen vektor ke-i. Untuk matriks, hal ini mengambalikan seluruh baris matriks ke i

```
>v:=[2,4,6,8]; v[3], A[3]
```

```
6
[7,  8,  9]
```

Indeks juga bisa menjadi vektor baris dari indeks. ":" menunjukkan semuaa indeks.

```
>v[1:2], A[:,2]
```

```
[2,   4]
             2
             5
             8
```

Bentuk singkat dari ":" tidak menyebutkan indeks secara lengkap

```
>A[,2:3]
```

```
             2              3
             5              6
             8              9
```

Untuk tujuan vertorisasi, elemen matriks dapat diakses selayaknya vektor

```
>A{4}
```

```
4
```

matriks juga dapat diratakan, menggunakan funsi "redim()". Fungsi tersebut dijalankan dengan fungsi "flatten()"

```
>redim(A,1,prod(size(A))), flatten(A)
```

```
[1,   2,   3,   4,   5,   6,   7,   8,   9]
[1,   2,   3,   4,   5,   6,   7,   8,   9]
```

Untuk menggunakan matriks sebagai tabel, kita atur ulang format defaultnya, dan menghitung tabel nilai sin dan cos nya. Perhatikan bahwa sudutnya secara default dalam radian.

```
>defformat; w=0°:45°:360°; w=w’; deg(w)
```

```
             0
            45
            90
           135
           180
           225
           270
           315
           360
```

Sekarang kita menambahkan kolom pada matriks

```
>M = deg(w)|w|cos(w)|sin(w)
```

```
      0           0           1           0
     45    0.785398    0.707107    0.707107
     90      1.5708           0           1
    135     2.35619   -0.707107    0.707107
    180     3.14159          -1           0
    225     3.92699   -0.707107   -0.707107
    270     4.71239           0          -1
    315     5.49779    0.707107   -0.707107
    360     6.28319           1           0
```

Menggunakan bahasa matriks, kita bisa menghasilkan beberapa tabel dari beberapa fungsi sekaligus.
Pada contoh berikut, kita menghitung t[j]^i untuk i dari 1 sampai n. kita mendapatkan sebuah matriks, yang tiap barisnya merupakan tabael t^i untuk sebuah i. Misalnya, matriks mempunyai elemen

$$a_{i,j} = t_j^i, \quad 1 \le j \le 101, \quad 1 \le i \le n$$

Sebuah fungsi yang tidak bekerja untuk masukan vektor harus "divaktorisasi". Hal ini dapat dicapai dengan kata kunci "map" pada definisi fungsi. Lalu fungsi akan dievaluasi untuk setiap elemen dari parameter vektornya.
Integrasi numerik "integrate()" bekerja hanya untuk interval terbatas skalar

```
>function map f(x) := integrate("x^x",1,x)
```

Kata kunci "map" memvektorisasi fungsi. Fungsi sekarang bekerja untuk vektor bilangan.

```
>f([1:5])
```

```
[0,  2.05045,  13.7251,  113.336,  1241.03]
```

## Sub-Matriks dan Elemen Matriks

Untuk mengakses elemen matriks, gunakan notasi kurung siku.

```
>A=[1,2,3;4,5,6;7,8,9], A[2,2]
```

```
            1            2            3
            4            5            6
            7            8            9
5
```

Kita bisa mengakses satu baris lengkap suatu matriks

```
>A[2]
```

```
[4,  5,  6]
```

Pada kasus vektor baris atau vektor kolom, hal ini mengembalikan elemen vektornya

```
>v=1:3; v[2]
```

```
2
```

Untuk memastikan, Anda ambil baris pertama untuk matriks A 1xn dan matriks A mxn, stetapkan semua kolom menggunakan index kedua yang kosong.

```
>A[2,]
```

```
[4,  5,  6]
```

Jika index merupakan index vektor, Euler akan mengembalikan baris yang sesuai pada matriks.4
Berikut ini kita ingin baris pertama dan kedua dari A.

```
>A[[1,2]]
```

```
            1           2           3
            4           5           6
```

Kita dapat mengurutkan kembali A dengan menggunakan index vektor. Agar tepat,kita tidak merubah A disini, tetapi menghitung versi urutan ulang dari A

```
>A[[3,2,1]]
```

```
            7           8           9
            4           5           6
            1           2           3
```

Trik index ini bekerja pada kolom juga.
Contoh ini memilih seluruh baris A dan kolom kedua dan ketiga.

```
>A[1:3,2:3]
```

```
            2           3
            5           6
            8           9
```

Untuk mempersingkat, ":" mendandai semua indeks baris dan kolom.

```
>A[:,3]
```

```
            3
            6
            9
```

Sebagai alternatif, biarkan index pertama kosong.

```
>A[,2:3]
```

```
            2           3
            5           6
            8           9
```

Kita juga bisa memperoleh baris terakhir dari A.

```
>A[-1]
```

```
[7,   8,   9]
```

Sekrang kita mengubah elemen A dengan menetaokan sub-matriks A menjadi suatu nilai. Faktanya, hal ini merubah matriks A yang telah tersimpan.

```
>A[1,1]=4
```

```
             4             2             3
             4             5             6
             7             8             9
```

Kita juga bisa menetapkan nilai untuk sebuah baris A

```
>A[1]=[-1,-1,-1]
```

```
            -1            -1            -1
             4             5             6
             7             8             9
```

Kita bahkan bisa menetapkan menjadi sub-matriks jika dia punya ukuran yang tepat.

```
>A[1:2,1:2]=[5,6;7,8]
```

```
             5             6            -1
             7             8             6
             7             8             9
```

Bahkan, beberapa shortcut diperbolehkan.

```
>A[1:2,1:2]=0
```

```
             0             0            -1
             0             0             6
             7             8             9
```

Perhatian : Index diluar batas mengahsiilkan matriks yang kosong, atau pesan eror, tergantung pada pengaturan sistem. Defaultnya sebuah pesan eror. Ingat,, bagaimanapun juga, index negatif mungkin digunakan untuk mengakses elemen matriks dihitung dari akhirnya.

```
>A[4]
```

```
Row index 4 out of bounds!
Error in:
A[4] ...
    ^
```

## Mengurutkan dan Mengacak

Fungsi "sort()" mengurutkan sebuah vektor baris.

```
>sort([5,6,4,8,1,9])
```

```
[1,  4,  5,  6,  8,  9]
```

Fungsi ini sering diperlukan untuk mengetahui index dari vektor yang diurutkan pada vektor aslinya. Hal ini dapat digunakan untuk mengurutkan kembalivektor lain dengan cara yabng sama.
Mari kita acakk sebuah vektor.

```
>v=shuffle(1:10)
```

```
[6,  3,  1,  5,  10,  4,  9,  8,  2,  7]
```

Indeks terdiri atas urutan yang depat dari v.

```
>{vs,ind}=sort(v); v[ind]
```

```
[1,  2,  3,  4,  5,  6,  7,  8,  9,  10]
```

Fungsi ini bekerja pada vektor string juga,

```
>s=["a","d","e","a","aa","e"]
```

```
a
d
e
a
aa
e
```

```
>{ss,ind}=sort(s); ss
```

```
a
a
aa
d
e
e
```

Seperti yang ANda lihat, posisi entri ganda cukup random.

```
>ind
```

```
[4,  1,  5,  2,  6,  3]
```

The function unique returns a sorted list of unique elements of a vector.
Funsi "unique" menunjukkan daftar berurutan dari elemen ysng khas dari vektor.

```
>intrandom(1,10,10), unique(%)
```

```
[9,  3,  8,  3,  5,  4,  10,  2,  5,  1]
[1,  2,  3,  4,  5,  8,  9,  10]
```

Fungsi ini bekerja pada vektor string juga.

```
>unique(s)
```

```
a
aa
d
e
```

# Aljabar Linier

EMT memiliki banyak fungsi untuk menyelesaikan sistem linier, sistem yang jarang, atau masalah regresi. Untuk dsistem linier Ax=b, kamu dapat menggunakan algoritma Gauss, Matriks Invers, atau permodelan linier. Operator A\b menggunakan versi Algoritma Gauss.

```
>A=[1,2;3,4]; b=[5;6]; A\b
```

```
          -4
         4.5
```

Sebagai contoh lain, kita menghasilkan suatu matriks 200x200 dan jumlah barisnya. lalu kita menyelesaikan Ax=b menggunakan invers matriks. Kita mengukur eror sebagai deviasi maksimal semua elemen daari 1, yang sebenarnya solusi yang tepat.

```
>A=normal(200,200); b=sum(A); longest totalmax(abs(inv(A).b-1))
```

```
  3.140820936664568e-13
```

Jika sistem tidak memiliki solusi, permodelan linier meminimalkan norma kesalahan Ax=b.

```
>A=[1,2,3;4,5,6;7,8,9]
```

```
           1             2             3
           4             5             6
           7             8             9
```

Determinan matriks ini adalah 0.

```
>det(A)
```

```
0
```

# Matriks Simbolik

Maxima memiliki matriks simbolik. Tenntunya, Ma=xima dapat digunakan untuk semacam masalah aljabar linier sederhana. Kita dapat mendefinisikan maatriks pada Euler dan MAxima dengan "&:=", dan kemudian menggunakannya pada ekspresi simbolik. Bentuk umum [...] untuk mendefinisikan matriks dapat digunakan pada Euler untuk mendefinisikan matriks simbolik.

```
>A &= [a,1,1;1,a,1;1,1,a]; $A
```

$$\begin{pmatrix} a & 1 & 1 \\ 1 & a & 1 \\ 1 & 1 & a \end{pmatrix}$$

```
>$&det(A), $&factor(%)
```

$$\left(a-1\right)^2\,\left(a+2\right)$$

```
>$&invert(A) with a=0
```

$$\begin{pmatrix} 1 & 0 \\ -\frac{b}{2} & \frac{1}{2} \end{pmatrix}$$

```
>A &= [1,a;b,2]; $A
```

$$\begin{pmatrix} 1 & a \\ b & 2 \end{pmatrix}$$

Seperti pada variabel simbolik, matriks in dapat digunakan pada ekspresi simbolik lainnya.

```
>$&det(A-x*ident(2)), $&solve(%,x)
```

$$\left[x=\frac{3-\sqrt{4\,a\,b+1}}{2} , x=\frac{\sqrt{4\,a\,b+1}+3}{2} \right]$$

$$\left[x=\frac{3-\sqrt{4\,a\,b+1}}{2} , x=\frac{\sqrt{4\,a\,b+1}+3}{2} \right]$$

Nilai eigen dapat juga dihitun secara otomatis. Hasilnya merupakan vektor dengan dua vektor yaitu nilai eigen dan perkaliannya.

```
>$&eigenvalues([a,1;1,a])
```

$$\left[\left[a-1 , a+1\right] , \left[1 , 1\right]\right]$$

Untuk mengekstraksi vektor eigen tertentu dibutuhkan peng-indeks-an yang hati-hati.

```
>$&eigenvectors([a,1;1,a]), &%[2][1][1]
```

$$[[[a-1,a+1],[1,1]],[[[1,-1]],[[1,1]]]]$$

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

Matriks simbolik dapat dievaluasi pada Euler secara numerik seperti pada ekspresi simbolik lainnya,.

```
>A(a=4,b=5)
```

```
           1                 4
           5                 2
```

Pada ekspresi simbolik, gunakan "with"

```
>$&A with [a=4,b=5]
```

$$\begin{pmatrix} 1 & 4 \\ 5 & 2 \end{pmatrix}$$

Akses pada baris matriks simbolik bekerja seperti pada matriks numerik.

```
>$&A[1]
```

$$[1,a]$$

Suatu ekspresi simbolik dapat terdiri atas suatu penugasan. Dan hal tersebut dapat merubah matriks A.

```
>&A[1,1]:=t+1; $&A
```

$$\begin{pmatrix} t+1 & a \\ b & 2 \end{pmatrix}$$

Ada fungsi-fungsi simbolik pada MAxima untuk membuat vektor dan matriks. Untuk ini, dapat merujuk pada dokumentasi MAxima atau tutorial tentang MAxima pada EMT.

```
>v &= makelist(1/(i+j),i,1,3); $v
```

$$\left[\frac{1}{j+1},\frac{1}{j+2},\frac{1}{j+3}\right]$$

```
>B &:= [1,2;3,4]; $B, $&invert(B)
```

$$\begin{pmatrix} -2 & 1 \\ \frac{3}{2} & -\frac{1}{2} \end{pmatrix}$$

$$\begin{pmatrix} -2 & 1 \\ \frac{3}{2} & -\frac{1}{2} \end{pmatrix}$$

Hasilnya dapat dievaluasi secara numerik pada Euler. Untuk informasi lebih lanjut terkait MAxima, lihatlh pada Pengenalan MAxima.

```
>$&invert(B)()
```

```
              -2               1
             1.5            -0.5
```

Euler juga punya fungsi yang sangat kuat "xinv()", yanga membuat usaha yang lebih besar dan mendapat lebih banyak hasil pasti.

Perhatikan, bahwa dengan "&:=" matriks B telah didefinisikan sebagai simbolik pada ekspresi simbolik dan sebagai numerik pada ekspresi numerik. Jadi kita bisa menggunakannya disini.

```
>longest B.xinv(B)
```

```
                         1                       0
                         0                       1
```

Sebagai contoh, nilai eigen A dapat dihitung secara numerik

```
>A=[1,2,3;4,5,6;7,8,9]; real(eigenvalues(A))
```

```
  [16.1168,  -1.11684,  0]
```

Atau secara simbolik. Lihatlah tutorial tentang MAxima untuk detailnya.

```
>$&eigenvalues(@A)
```

$$\left[ \left[ \frac{15-3\,\sqrt{33}}{2} , \frac{3\,\sqrt{33}+15}{2} , 0 \right] , \left[ 1 , 1 , 1 \right] \right]$$

## Nilai Numerik pada Ekspresi Simbolik

Suatu Ekspresi simbolik hanya merupakan string yang mengandung suatu ekspresi. Jika kitat ingin mendefinisikan kedua nilainya untuk ekspresi simbolik dan numerik, kita harus menggunakan "&:=".

```
>A &:= [1,pi;4,5]
```

```
              1        3.14159
              4              5
```

Tetap ada perbedaan antara bentuk numerik dan simbolik. Ketika memindahkan matriks ke bentuk simbolik, pendekatan pecahan pada real akan digunakan,.

```
>$&A
```

$$\begin{pmatrix}1 & \frac{1146408}{364913} \\ 4 & 5\end{pmatrix}$$

Untuk menghindari ini, ada fungsi "mxmset(variable)".

```
>mxmset(A); $&A
```

$$\begin{pmatrix}1 & 3.141592653589793 \\ 4 & 5\end{pmatrix}$$

MAxima juga dapat menghitung dengan angka floating point, dan bahkan dengan anka foating poin besar hingga 32 digit. Namun evaluasinya jauh lebih lambat.

```
>$&bfloat(sqrt(2)), $&float(sqrt(2))
```

$$1.414213562373095$$

Ketepatan angka foating pon yang besar dapat diubah.

```
>&fpprec:=100; &bfloat(pi)
```

```
        3.14159265358979323846264338327950288419716939937510582097494\
  459230781640628620899862803482534211706 8b0
```

Suatu variabel numerik dapat digunakan pada ekspresi simbolik apapun menggunakan "@var"
Perhatikan bahwa ini hany perlu dilakukan, jika variabel telah didefinisikan dengan ":=" atau "=" sebagai variabell numerik.

```
>B:=[1,pi;3,4]; $&det(@B)
```

$$-5.424777960769379$$

## Demo - Suku Bunga

Di bawah ini, kita menggunakan Euler MAth Toolbox (EMT) untuk perhitungan suku bunga. Kita melakukannya secara numerik dan simbolik untuk menunjukkan bagaimana Euler dapat digunakan untuk menyelesaikan masalah kehidupan nyata.
Asumsikan Anda memiliki modal awal 5000 (misalnya dalam dolar)

```
>K=5000
```

```
5000
```

Now we assume an interest rate of 3% per year. Let us add one simple rate and compute the result. Sekarang kita asumsikan suku bunganya 3% per tahun. MAri kita tambahkan satu suku bunga sederhana dan menghitung hasilnya.

```
>K*1.03
```

```
5150
```

Euler akan memahami sintaks berikut ini juga.

```
>K+K*3%
```

```
5150
```

Tetapi lebih mudah untuk menggunakan faktornya

```
>q=1+3%, K*q
```

```
1.03
5150
```

Untuk 10 tahun, kita dapat mengalikan faktornya dengan sederhana dan mendapatkan nilai akhit dengan suku bunga majemuk.

```
>K*q^10
```

```
6719.58189672
```

Untuk tujuan kita ini, kita bisa mengatur format ke 2 digit desimal dibelakang titik.

```
>format(12,2); K*q^10
```

```
    6719.58
```

Mari kita cetak pembulatan 2 digit dalam kalimat yang lengkap.

```
>"Starting from " + K + "$ you get " + round(K*q^10,2) + "$."
```

```
Starting from 5000$ you get 6719.58$.
```

Bagaimana jika kita ingin mengetahui hasil antara tahun pertama hingga tahun ke 9? Bahasa matriks Euler sangat membantu. Anda tidak perlu menulis loop, tetapi cukup memasukkan :

```
>K*q^(0:10)
```

```
Real 1 x 11 matrix

    5000.00     5150.00     5304.50     5463.64     ...
```

BAgaimana cara kerja keajaiban ini? Pertama, ekspresi 0:10 mengembalikan vektor bilanhan bulat.

```
>short 0:10
```

```
 [0,  1,  2,  3,  4,  5,  6,  7,  8,  9,  10]
```

Kemudian semua operator dan fungsi pada Euler dapat diaplikasikan pada vektor elemen demi elemen.

```
>short q^(0:10)
```

```
 [1,  1.03,  1.0609,  1.0927,  1.1255,  1.1593,  1.1941,  1.2299,
 1.2668,  1.3048,  1.3439]
```

merupakan vektor dari faktor q^0 sampai q^10. Vektor ini dikalikan dengan K, dan kita peroleh vektor nilai-nilai.

```
>VK=K*q^(0:10);
```

Tentu saja, cara yang realistis untuk menghitung suku bunga ini adalah dengan membulatkan ke sen terdekat setelah setiap tahun. Mari kita tambahkan fungsi untuk ini.

```
>function oneyear (K) := round(K*q,2)
```

MAri kita bandingkan kedua hasil, dengan dan tanpa pembulatan.

```
>longest oneyear(1234.57), longest 1234.57*q
```

```
                 1271.61
               1271.6071
```

Sekarang tidak ada rumus sederhana untuk tahun ke-n, dan kita harus melakukan iterasi selama tahun-tahun tersebut. Euler menyediakan banyak solusi untuk ini.
Cara termudah adalah menggunakan fungsi "iterate", yang mengulangi fungsi tertentu beberapa kali.

```
>VKr=iterate("oneyear",5000,10)
```

```
 Real 1 x 11 matrix

     5000.00     5150.00     5304.50     5463.64     ...
```

Kita dapat mencetaknya dengan cara yang ramah pengguna, menggunakan format dengan jumlah tempat desimal tetap.

```
>VKr'
```

```
     5000.00
     5150.00
     5304.50
     5463.64
```

```
5627.55
5796.38
5970.27
6149.38
6333.86
6523.88
6719.60
```

Untuk mendapatkan elemen tertentu dari vektor, kita menggunakan indeks di dalam tanda kurung siku.

```
>VKr[2], VKr[1:3]
```

```
5150.00
5000.00     5150.00     5304.50
```

Menariknya, kita juga dapat menggunakan vektor indeks. Ingat bahwa 1:3 menghasilkan vektor [1,2,3].
Mari kita bandingkan elemen terakhir dari nilai yang dibulatkan dengan nilai lengkapnya.

```
>VKr[-1], VK[-1]
```

```
6719.60
6719.58
```

Perbedaannya sangat kecil

## Penyelesaian Persamaan

Sekarang kita akan menggunakan fungsi yan lebih canggih, yang menambahkan sejumlah uang tertentu tiap tahun.

```
>function onepay (K) := K*q+R
```

Kita tidak perlu menentukan nilai q atau R untuk mendefinisikan fungsi. Hanya jika menjalankan perintah, kita harus mendefinisikan nilai-nilai tersebut. Kita pilih R=200.

```
>R=200; iterate("onepay",5000,10)
```

```
 Real 1 x 11 matrix

     5000.00     5350.00     5710.50     6081.82     ...
```

What if we remove the same amount each year?

```
>R=-200; iterate("onepay",5000,10)
```

```
 Real 1 x 11 matrix

     5000.00     4950.00     4898.50     4845.45     ...
```

Kita melihat bahwa jumlah uang berkurang. Jelas, jika kita hanya mendapatkan 150 dari bunga di tahun pertama, tetapi mengeluarkan 200, kita akan kehilangan uang setiap tahun.
Bagaimana kita dapat menentukan berapa tahun uang tersebut akan bertahan? Kita perlu menulis loop untuk ini. Cara termudah adalah dengan melakukan iterasi cukup lama hingga uangnya habis.

```
>VKR=iterate("onepay",5000,50)
```

```
 Real 1 x 51 matrix

     5000.00     5350.00     5710.50     6081.82     ...
```

Dengan menggunakan bahasa matriks, kita dapat menentukan nilai negatif pertama dengan cara berikut.

```
>min(nonzeros(VKR<0))
```

```
         0.00
```

Alasan untuk ini adalah bahwa nonzeros(VKR<0) mengembalikan vektor indeks
i di mana VKR[i]<0, dan min menghitung indeks minimal.
Karena vektor selalu dimulai dengan indeks 1, jawabannya adalah 47 tahun.
Fungsi iterate() memiliki satu trik tambahan. Fungsi ini dapat menerima kondisi akhir sebagai argumen. Kemudian, ia akan mengembalikan nilai serta jumlah iterasi.

```
>{x,n}=iterate("onepay",5000,till="x<0"); x, n,
```

Mari kita coba menjawab pertanyaan yang lebih ambigu. Misalkan kita tahu bahwa nilai tersebut adalah 0 setelah 50 tahun. Berapa suku bunga yang diperlukan?
Ini adalah pertanyaan yang hanya bisa dijawab secara numerik. Di bawah ini, kita akan menurunkan rumus yang diperlukan. Kemudian Anda akan melihat bahwa tidak ada rumus sederhana untuk suku bunga. Namun, untuk saat ini, kita akan mencari solusi numerik.
Langkah pertama adalah mendefinisikan sebuah fungsi yang melakukan iterasi sebanyak n kali. Kita tambahkan semua parameter ke dalam fungsi ini.

```
>function f(K,R,P,n) := iterate("x*(1+P/100)+R",K,n;P,R)[-1]
```

Iterasinya sama seperti di atas:

$$x_{n+1} = x_n \cdot \left(1 + \frac{P}{100}\right) + R$$

Namun, kita tidak lagi menggunakan nilai global dari
R dalam ekspresi kita. Fungsi seperti iterate() memiliki trik khusus di Euler. Anda dapat meneruskan nilai-nilai variabel dalam ekspresi sebagai parameter titik koma, dalam hal ini
P dan Q
Selain itu, kita hanya tertarik pada nilai terakhir. Jadi, kita mengambil indeks [-1].
Mari kita coba sebuah uji coba.

```
>f(5000,-200,3,47)
```

```
       -19.83
```

Sekarang kita dapat menyelesaikan masalah kita

```
>solve("f(5000,-200,x,50)",3)
```

```
3.15
```

Rutinitas solve menyelesaikan persamaan expression=0 untuk variabel
x. Jawabannya adalah 3,15% per tahun. Kita mengambil nilai awal 3% untuk algoritma. Fungsi solve() selalu memerlukan nilai awal.

Kita dapat menggunakan fungsi yang sama untuk menjawab pertanyaan berikut: Berapa banyak yang dapat kita tarik per tahun sehingga modal awal habis setelah 20 tahun, dengan asumsi suku bunga 3% per tahun.

```
>solve("f(5000,x,3,20)",-200)
```

```
-336.08
```

Perhatikan bahwa Anda tidak dapat menyelesaikan jumlah tahun secara langsung, karena fungsi kita mengasumsikan
n sebagai nilai yang bulat

## Solusi Simbolik untuk Masalah Suku Bunga

Kita dapat menggunakan bagian simbolik dari Euler untuk mempelajari masalah ini. Pertama, kita mendefinisikan fungsi onepay() secara simbolik.

```
>function op(K) &= K*q+R; $&op(K)
```

$$R + q\,K$$

Sekarang kita dapat mengiterasi ini

```
>$&op(op(op(op(K)))), $&expand(%)
```

$$q^3\,R + q^2\,R + q\,R + R + q^4\,K$$

KIta melihat pola. Setlah n periode, kita memiliki :

$$K_n = q^n K + R(1 + q + \ldots + q^{n-1}) = q^n K + \frac{q^n - 1}{q - 1} R$$

Formula ini adalah formula untuk jumlah geometri, yang dikenal oleh Maxima.

```
>&sum(q^k,k,0,n-1); $& % = ev(%,simpsum)
```

$$\sum_{k=0}^{n-1} q^k = \frac{q^n - 1}{q - 1}$$

Ini agak rumit. Jumlah dihitung dengan flag "simpsum" untuk menyederhanakannya menjadi kuotien.
Mari kita buat fungsi untuk ini.

```
>function fs(K,R,P,n) &= (1+P/100)^n*K + ((1+P/100)^n-1)/(P/100)*R; $&fs(K,R,P,n)
```

$$\frac{100\,\left(\left(\frac{P}{100}+1\right)^{n}-1\right)\,R}{P}+K\,\left(\frac{P}{100}+1\right)^{n}$$

Fungsi tersebut sama seperti fungsi f kita sebelumnya. Tetapi lebih efektif.

```
>longest fs(5000,-200,3,47), longest fs(5000,-200,3,47)
```

```
      -19.82504734652684
      -19.82504734652684
```

Kita sekarang dapat menggunakannya untuk menanyakan kapan waktu n. Kapan modal kita habis? Perkiraan awal kita adalah 30 tahun.

```
>solve("fs(5000,-330,3,x)",30)
```

```
 20.5061016552
```

Jawaban ini mengatakan bahwa modal akan menjadi negatif setelah 21 tahun.
Kita juga dapat menggunakan sisi simbolik Euler untuk menghitung rumus untuk pembayaran.
Misalkan kita mendapatkan pinjaman sebesar K, dan membayar n cicilan sebesar R (dimulai setelah tahun pertama) meninggalkan utang sisa sebesar Kn (pada saat pembayaran terakhir). Rumus untuk ini jelas adalah:

```
>equ &= fs(K,R,P,n)=Kn; $&equ
```

$$\frac{100\,\left(\left(\frac{P}{100}+1\right)^{n}-1\right)\,R}{P}+K\,\left(\frac{P}{100}+1\right)^{n}=Kn$$

Biasanya rumus ini dberikan dalam bentuk :

$$i=\frac{P}{100}$$

```
>equ &= (equ with P=100*i); $&equ
```

$$\frac{\left(\left(i+1\right)^{n}-1\right)\,R}{i}+\left(i+1\right)^{n}\,K=Kn$$

Kita dapat menyelesaikan suku bunga R secara simbolik

```
>$&solve(equ,R)
```

$$\left[R=\frac{i\,Kn-i\,\left(i+1\right)^{n}\,K}{\left(i+1\right)^{n}-1}\right]$$

Seperti yang Anda lihat dari rumusnya, fungsi ini mengembalikan kesalahan floating point untuk i=0. Namun, Euler tetap menggambarkannya.
Tentu saja, kita memiliki batasan berikut.

```
>$&limit(R(5000,0,x,10),x,0)
```

$$\lim_{x\to 0}R\left(5000 , 0 , x , 10\right)$$

Jelas, tanpa bunga kita harus membayar kembali 10 cicilan sebesar 500.
Persamaan ini juga dapat diselesaikan untuk n. Ini terlihat lebih baik jika kita menerapkan beberapa penyederhanaan.

```
>fn &= solve(equ,n) | ratsimp; $&fn
```

$$\left[ n=\frac{\log \left(\frac{R+i\,K\,n}{R+i\,K}\right)}{\log \left(i+1\right)} \right]$$

---

# TUGAS INDIVIDU ALJABAR

Ummi

## Nurrohmah (24030130095)

Dalam materi aljabar, ada banyak sekali sub-materi yang dapat kita pelajari. Bentuk aljabar sendiri mencangkup polinomial, bentuk pangkat, bentuk akar, logaritma, matriks, dan lainnya.Cakupan aljabar dapat berupa bilangan real, juga bilangan kompleks. Sebagian akan saya bahas pada uraian berikut ini.
Uraian di bawah akan di awali dengan uraian materi (teori) secara singkat, lalu diberikan contoh-contoh soal dari PDF serta sumber lain, yang diselesaikan dengan EMT.
Sumber materi & soal tambahan : Algebra & Trigonometry (Cynthia Young)

## A. Operasi Bentuk Aljabar

Secara umum, operasi bentuk aljabar dapat berupa penjabaran, penyederhanaan dan pemfaktoran. Ada beberapa sifat yang dapat digunakan dalam Operasi bentuk aljabar, diantaranya :
- sifat polinomial

$$ax_1+ax_2+ax_3+...+ax_n=a(x_1+x_2+x_3+...+x_n)$$

$$(a+b)^2=a^2+2ab+b^2$$

$$(a-b)^2=a^2-2ab+b^2$$

$$(a+b)(a-b)=a^2-b^2$$

$$(a+b)^3=a^3+3a^2b+3ab^2+b^3$$

$$(a-b)^3=a^3-3a^2b+3ab^2-b^3$$

$$a^3+b^3=(a+b)(a^2-ab+b^2)$$

$$a^3-b^3=(a-b)(a^2+ab+b^2)$$

- sifat yang berkaitan dengan eksponen (bentuk pangkat) antara lain :

$$a^n = \underbrace{a\cdot a\cdot a\cdots a}_{\text{sebanyak n suku}}$$

$$a^0=1, a\neq 0$$

$$a^{-n}=\frac{1}{a^n}$$

$$a^m\cdot a^n=a^{m+n}$$

$$\frac{a^m}{a^n}=a^{m-n}$$

$$a^m\cdot a^n=a^{m+n}$$

$$(a^m)^n=a^{mn}$$

$$(\frac{a}{b})^n=\frac{a^n}{b^n}$$

Berikut ini beberapa contoh soal terkait operasi bentuk aljabar :
Contoh 1.
Sederhanakan bentuk eksponen berikut :

$$\left(\frac{24a^{10}b^{-8}c^7}{12a^6b^{-3}c^5}\right)^{-5}$$

```
>5$&((24*a^10*b^(-8)*c^7)/(12*a^6*b^(-3)*c^5))^(-5)
```

$$\frac{b^{25}}{32\,a^{20}\,c^{10}}$$

Soal ini termasuk jenis penyederhanaan bentuk aljabar.
Bentuk eksponensial tersebut dapat disederhanakan dengan mengerjakan yang di dalam kurung terlebih dahulu, baru setelah itu dipangkatkan -5. Kita dapat menggunakan sifat berikut :

$$\left(\frac{24a^{10}b^{-8}c^7}{12a^6b^{-3}c^5}\right)^{-5}=\left(\frac{24}{12}a^{10-6}b^{-8-(-3)}c^{7-5}\right)^{-5}=(2a^4b^{-5}c^2)^{-5}=\frac{b^25}{32a^{20}c^{10}}$$

Contoh 2
Carilah bentuk lain dari bentuk aljabar berikut

$$(x-1)(x^2+x+1)(x^3+1)$$

```
>$&expand((x-1)*(x^2+x+1)*(x^3+1))
```

$$x^6 - 1$$

Contoh 2 merupakan contoh menjabarkan bentuk aljabr, yaitu dengan mengalikan suku-sukunya dengan sifat distributif perkalian :

$$(a + b)(c + d) = ac + ad + bc + bd$$

kita dapat mengalikan (x-1)(x^2+x+1) terlebih dahulu. kemudian hasilnya dikalikan kembali dengan (x^3+1)
Contoh 3.
Faktorkanlah bentuk aljabar berikut :

$$4z^2 - 81$$

```
>$&factor(4*z^2-81)
```

$$(2\,z - 9)\,(2\,z + 9)$$

Contoh ini merupakan kasus pemfaktoran yang dapat diselesaikan dengan sifat :

$$a^2 - b^2 = (a + b)(a - b)$$

Contoh 4.
Bagaimana bentuk pemfaktoran dari bentuk aljabar berikut ini?

$$x^6 - 2x^5 + x^4 - x^2 + 2x - 1$$

```
>$&factor(x^6-2*x^5+x^4-x^2+2*x-1)
```

Pemfaktoran di atas dapat diselesaikan dengan cara Horner, tetapi cukup panjang. Dengan menggunakan EMT menjadi lebih cepat.
Contoh 5.
Volume suatu wadah diketahui dapat dituliskan sebagai

$$V = x^3 + 7x^2 + 12x$$

Nyatakan volume tersebut dalam bentuk faktorisasi polinomial dengan vaariabel x.

```
>$&factor(x^3+7*x^2+12*x)
```

$$x\,(x + 3)\,(x + 4)$$

Contoh soal di atas merupakan contoh pemfaktoran yang dikaitkan dengan geometri.

## Fungsi Matematika dan Operasinya

Ada beberapa jenis fungsi dalam aljabar, mulai dari fungsi linier, fungsi kuadrat, fungsi konstan, fungsi akar kuadrat, fungsi nilai absolut, fungsi hiperbolik, dan masih banyak lainnya. Fungsi memiliki beberapa macam operasi, antara lain :

$$(f+g)(x) = f(x) + g(x)$$

$$(f-g)(x) = f(x) - g(x)$$

$$(f \cdot g)(x) = f(x) + \cdot g(x)$$

$$\left(\frac{f}{g}\right)(x) = \frac{f(x)}{g(x)}$$

Ada pula operasi komposisi fungsi

$$f(x) \circ g(x) = f(g(x))$$

$$g(x) \circ f(x) = g(f(x))$$

Berikut ini merupakan contoh soal tentang fungsi matematika :
Contoh 1.
Tentukan nilai f(3), jika

$$f(x) = \sqrt{3x+7}$$

```
>function f(x):=sqrt(3*x+7);
```

```
>f(3)
```

```
4
```

f(3) dapat diperoleh dengan substitusi nilai x=3 ke fungsi f(x)
Contoh 2.
Tentukan nilai (f+g)(x) jika

$$f(x) = 2x + 1$$

$$g(x) = \frac{x-4}{3x+2}$$

```
>function f(x)&=2*x+1; function g(x)&=(x-4)/(3*x+2);
>compose_function([f(x);g(x)])
```

```
Variable or function x not found.
Error in:
compose_function([f(x);g(x)]) ...
                    ^
```

## Persamaan dan Sistem Persamaan

Suatu persamaan kuadrat dapat diselesaikan dengan pemfaktoran, melengkapkan kuadrat sempurna, atau dengan rumus abc.
Suatu sistem persamaan terdiri atas 2 atau lebih persamaan. Bisa diselesaikan dengan Eliminasi, Substitusi, atau campuran. TEtapi ada kalanya beberapa tipe soal, penyelesaiannya menjadi lebih rumit.
Berikut ini contoh soal PErsamaan dan Sistem Persamaan
Contoh 1.
Selesaikan persamaan di bawah ini :

$$x^2 - 2x = 15$$

```
>$& x^2-2*x=15, $&solve %
```

$$x^2 - 2\,x = 15$$

$$[x = -3, x = 5]$$

Contoh 2.
Carilah penyelesaian persamaan berikut ini :

$$x - 3\sqrt{x} - 4 = 0$$

```
>$& x-3*sqrt(x)-4=0, $&solve %
```

$$x - 3\,\sqrt{x} - 4 = 0$$

$$\left[x = 3\,\sqrt{x} + 4\right]$$

Contoh 3.
Bagaimana penyelesaian persamaan trigonometri berikut?

$$cos2x - sinx = 1$$

```
>$& cos(2*x)-sin(x)=1, $&solve %
```

$$\cos\left(2\,x\right) - \sin x = 1$$

$$\left[\cos\left(2\,x\right) = \sin x + 1\right]$$

Contoh 4.
Diketahui :

$$\begin{cases} 4x + 2y = 11 \\ 3x - y = 2 \end{cases}$$

Berapa nilai x dan y?

```
>$&solve([4*x+2*y=11,3*x-y=2],[x,y])
```

$$\left[\left[x=\frac{3}{2},y=\frac{5}{2}\right]\right]$$

Contoh 5.
Carilah penyelesain sistem persamaan berikut :

$$\begin{cases} x+2y-3z=9 \\ 2x-y+2z=-8 \\ 3x-y-4z=3 \end{cases}$$

```
>N=[1,2,-3;2,-1,2;3,-1,-4]
```

```
            1             2            -3
            2            -1             2
            3            -1            -4
```

```
>H=[9;-8;3]
```

```
            9
           -8
            3
```

```
>A=N\H
```

```
           -1
            2
           -2
```

## Pertidaksamaan dan Sistem Pertidaksamaan

Contoh 1.
Selesaikan pertidaksamaan berikut

$$3-2x<7$$

```
>&load(fourier_elim)
```

```
        C:/Program Files/Euler x64/maxima/share/maxima/5.35.1/share/f\
  ourier_elim/fourier_elim.lisp
```

```
>$&fourier_elim([3-2*x<7],[x])
```

$$[-2 < x]$$

Contoh 2.
Selesaikan pertidaksamaan berikut ini

$$\frac{x+4}{x} < -\frac{1}{3}$$

```
>$&fourier_elim([((x+4)/x)<(-1/3)],[x])
```

$$[-3 < x, x < 0]$$

Contoh 3.
Carilah penyelesaian sistem pertidaksamaan berikut :

$$1.4x^2 - 7.2x + 5.3 > -8.6x + 3.7$$

```
>$&fourier_elim([1.4*x^2-7.2*x+5.3>-8.6*x+3.7],[x])
```

$$\left[1.4\,x^2 + 1.399999999999999\,x + 1.6 > 0\right]$$

Contoh 4.
Carilah penyelesaian sistem pertidaksamaan berikut ini

$$\begin{cases} y < x^2 + 4x \\ y \leq 3 - x \end{cases}$$

```
>$&fourier_elim([y<x^2+4*x,y<=3-x],[x,y])
```

$$\left[x = 3 - y, -y + 4\,(3 - y) + (3 - y)^2 > 0\right] \vee \left[x < 3 - y, -y + x^2 + 4\,x > 0\right]$$