TI_EMTaljabar_Prima Rosalina

Nama: Prima Rosalina
NIM: 24030130035
Kelas: Pendidikan Matematika C 2024

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)
```

$$-\frac{42}{x\,y^4}$$

Menjabarkan:

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

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

$$expand\left(\left(-\frac{1}{y^9}-7\,x^2\right)\,\left(y^5+\frac{6}{x^3}\right)\right)=-7\,x^2\,y^5-\frac{1}{y^4}-\frac{6}{x^3\,y^9}-\frac{42}{x}$$

Baris Perintah

Baris perintah Euler terdiri dari satu atau beberapa perintah Euler diikuti dengan titik koma ";" atau koma ",". Titik koma mencegah pencetakan hasil. Koma setelah perintah terakhir dapat dihilangkan.

Baris perintah berikut hanya akan mencetak hasil ekspresi, bukan tugas atau perintah format.

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

```
16.7551608191
```

Perintah harus dipisahkan dengan yang kosong. 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 dieksekusi dalam urutan yang ditekan pengguna kembali. Jadi Anda mendapatkan nilai baru setiap kali Anda menjalankan baris kedua.

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

```
0.540302305868
```

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

```
0.857553215846
```

Jika dua garis terhubung dengan "..." kedua garis akan selalu dieksekusi secara bersamaan.

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

```
1.41666666667
1.41421568627
1.41421356237
```

Ini juga merupakan cara yang baik untuk membagi perintah panjang ke dalam dua baris atau lebih. Anda dapat menekan Ctrl+Return untuk memisahkan satu baris menjadi dua di posisis kursor saat ini, atau Ctrl+Back untuk menggabungkan garis.

Sedangkan untuk menyembunyikan semua multi-garis tekan Ctrl + L. Kemudian garis-garis berikutnya hanya akan terlihat jika salah satunya memiliki fokus. Untuk menyembunyikan satu multi-baris, mulailah baris pertama dengan "%+".

```
>%+ x=4+5; ...
  // Baris ini tidak akan terlihat setelah kursor tidak berada di baris tersebut.
```

Baris yang diawali dengan %% tidak akan terlihat sama sekali.

```
81
```

Euler Math Toolbox (EMT) mendukung perulangan (loops) di baris perintah selama perulangan tersebut muat dalam satu baris tunggal atau multi-baris. Dalam program, pembatasan ini tidak berlaku tentunya. 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 apa-apa untuk menggunakan multi-line. Pastikan baris 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 bersyarat juga berfungsi.

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

```
Thought so!
```

Saat Anda mengeksekusi sebuah perintah, kursor dapat berada di posisi manapun di baris perintah. Anda dapat kembali ke perintah sebelumnya atau melompat ke perintah berikutnya menggunakan tombol panah. Atau, Anda bisa mengklik bagian komentar di atas perintah untuk langsung menuju perintah tersebut.

Saat Anda memindahkan kursor sepanjang garis, pasangan tanda kurung buka dan tutup akan disorot. Perhatikan juga status line. Setelah tanda kurung buka pada fungsi sqrt(), status line akan menampilkan teks bantuan untuk fungsi tersebut. Jalankan perintah dengan menekan tombol Return.

```
>sqrt(sin(10°)/cos(20°))
```

```
0.429875017772
```

Untuk melihat bantuan untuk perintah terbaru, buka jendela bantuan (help window) dengan menekan F1. Di sana, Anda dapat memasukkan teks untuk dicari. Pada baris kosong, bantuan untuk jendela bantuan itu sendiri akan ditamppilkan. Anda dapat menekan Escape untuk menghapus baris atau untuk menutup jendela bantuan.

Anda juga dapat klik ganda (double click) pada perintah manapun untuk membuka bantuan untuk perintah tersebut. Coba double click perintah exp di bawah ini pada baris perintah.

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

```
2.5
```

Anda juga dapat copy paste di EMT. Gunakan Ctrl+C dan Ctrl+V untuk ini. Untuk menandai teks, geser mouse atau gunakan Shift bersamaan dengan tombol kursor manapun. Selain itu, Anda juga dapat menyalin tanda kurung yang disorot.

Sintaks Dasar

EMT mengenal fungsi-fungsi matematika yang umum. Seperti yang telah Anda lihat sebelumnya, fungsi trigonometri bekerja dalam radian atau derajat. Untuk mengonversi ke derajat, tambahkan simbol derajat (dengan tombol F7) ke nilai atau gunakan fungsi rad(). Fungsi akar kuadrat disebut sqrt di EMT. Tentu saja, x^(1/2) jga bisa digunakan.

Untuk menetapkan variabel, gunakan "=" atau ":=". Demi kejelasan, pengantar ini menggunakan bentuk yang terakhir. Spasi tidak berpengaruh, tetapi spasi antar perintah tetap diharapkan.

Beberapa perintah dalam satu baris dipisahkan dengan "," atau ";". Tanda titik koma menyembunyikan hasil keluaran dari perintah tersebut. Di akhir baris perintah, koma diasumsikan secara otomatis jika tanda titik koma tidak ada.

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

```
30.65625
```

EMT menggunakan sintaks pemrograman untuk ekspresi. Untuk mengetik

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

Anda harus mengatur tanda kurung dengan benar dan menggunakan "/" untuk pecahan. Perhatikan tanda kurung yang disorot untuk bantuan. Perhatikan bahwa konstanta Euler e diberi nama E dalam EMT.

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

```
8.77908249441
```

Untuk menghitung ekspresi rumit seperti

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

Anda harus memasukkannya dalam bentuk baris.

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

```
23.2671801626
```

Letakkan tanda kurung dengan hati-hati di sekitar sub-ekspresi yang perlu dihitung terlebih dahulu. EMT membantu Anda dengan menyorot ekspresi bahwa braket penutup selesai. Anda juga harus memasukkan nama "pi" untuk huruf Yunani pi.

Hasil dari perhitungan ini adalah bilangan floating point. Secara default dicetak dengan akurasi sekitar 12 digit. Di baris perintah berikut, kita juga belajar bagaimana kita bisa merujuk ke hasil sebelumnya dalam baris yang sama.

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

```
0.47619047619
10/21
```

Perintah Euler dapat berupa ekspresi atau perintah primitif. Ekspresi terbuat dari operator dan fungsi. Jika diperlukan, hal tersebut harus berisi tanda kurung untuk memaksa urutan eksekusi yang benar. Jika ragu, memasang braket atau tanda kurung adalah ide yang bagus. Perhatikan bahwa EMT menunjukkan tanda kurung buka dan tutup saat mengedit baris perintah.

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

```
14.4978445072
```

Operator numerik Euler meliputi

```
+ unary atau operator plus
- unary atau operator minus
* operator perkalian
/ operator pecahan
. produk matriks
a^b daya untuk positif a atau bilangan bulat b (a**b juga berfungsi)
n! operator faktorial
```

dan masih banyak lagi.

Berikut adalah beberapa fungsi yang mungkin Anda butuhkan. Ada banyak lagi.

```
sin, cos, tan, atan, asin, acos, rad, deg
log, exp, log10, sqrt, logbase
bin, logbin, logfac, mod, lantai, ceil, bulat, abs, tanda
conj, re, im, arg, conj, nyata, kompleks
beta, betai, gamma, complexgamma, ellrf, ellf, ellrd, elle
bitand, bitor, bitxor, bitnot
```

Beberapa perintah memiliki alias, mis. 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 tentang urutan eksekusi! Berikut ini tidak sama dengan (2^3)^4, yang merupakan default untuk 2^3^4 di EMT (beberapa sistem numerik melakukannya dengan cara lain).

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

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

Bilangan Asli

Tipe data utama dalam Euler adalah bilangan real. Real direpresentasikan dalam format IEEE dengan akurasi sekitar 16 digit desimal.

```
>longest 1/3
```

```
0.3333333333333333
```

Representasi ganda internal membutuhkan 8 byte.
Representasi ganda adalah format penyimpanan untuk floating-point yang menggunakan 64 bit(8 byte).

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

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

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

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

Perbedaan 'printdual' dan 'printhex' adalah 'printdual' yakni mencetak representasi internal dari sebuah bilangan floating-point dalam format presisi ganda (pendekatan yang sangat dekat dengan nilai aslinya tetapi tidak persis sama), meskipun ia tergantung pada konteks bahasa pemograman tertentu. Sedangkan 'printhex' yakni representasi dari nilai floating-point dalam bentuk heksadesimal(basis 16), heksadesimal ini adalah cara yang lebih ringkas untuk menampilkan nilai biner karena setiap digit heksadesimal mempresentasikan empat digit biner.

String

Sebuah string dalam Euler didefinisikan dengan "..."

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

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

String dapat digabungkan dengan | atau dengan +. Ini juga berfungsi dengan angka, yang dikonversi menjadi string dalam kasus itu.

```
>"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.
```

Pada String fungsi print mengonversi angka menjadi string. Ini dapat mengambil sejumlah digit dan sejumlah tempat (0 untuk keluaran padat), dan secara optimal satu unit.

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

```
Golden Ratio : 1.61803
```

Ada string khusus bernama 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 mengevaluasinya. Ini bekerja untuk ekspresi juga (lihat dibawah).

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

```
1234.5
```

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

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

```
affe
charlie
bravo
```

Vektor pada string kosong dilambangkan dengan [none]. Dan vektor string dapat digabungkan dengan '|'.

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

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

String dapat berisi karakter Unicode. Secara internal, string ini berisi 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
```

```
α = 45°
```

I

Dalam komentar, entitas yang sama seperti α β, dan lain-lain dapat digunakan. Ini bisa menjadi alternatif cepat untuk LaTeX. (Detail 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]
```

Perintah ini menghasilkan array atau daftar angka berupa vektor angka yang mewakili karakter dalam string dalam bentuk kode Unicode.
Fungsi kebalikannya adalah chartoutf().

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

```
Ü is a German letter
```

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

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

```
We have α=β.
```

Memungkinkan juga untuk menggunakan entitas numerik.

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

Ähnliches

Nilai Boolean

Nilai boolean direpresentasikan dengan 1=true atau 0=false dalam euler. String dapat dibandingkan, seperti halnya angka.

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

```
0
1
```

"dan" adalah operator "&&" dan "atau" adalah operator "||", seperti dalam bahasa C. (Kata-kata "dan" dan "atau" hanya dapat digunakan dalam kondisi "jika".

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

```
1
```

Operator Boolean mematuhi aturan 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 mengambil 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 Keluaran (Output Format)

Format keluaran default EMT mencetak 12 digit. Untuk memastikan kita melihat format default, kita menggunakan reset format.

```
>defformat; pi
```

```
3.14159265359
```

Secara internal, EMT menggunakan standar IEEE (Institute of Electrical and Electronics Engineers) untuk bilangan ganda dengan sekitar 16 digit desimal. Untuk melihat bentuk digit penuh, gunakan perintah

"longestformat" atau gunakan operator "longest" untuk memunculkannya.

```
>longest pi
```

```
            3.141592653589793
```

Berikut ini adalah repesentasi heksadesimal internal dari bilangan ganda.

Heksadesimal adalah sistem bilangan yang menggunakan basis 16. Di mana angka 0 hingga 9 (untuk mewakili nilai 0 hingga 9) dan huruf A hingga F (untuk mewakili nilai 10 hingga 15).

```
>printhex(pi)
```

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

Format keluaran dapat diubah secara permanen dengan perintah format.

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

```
          0.33333
          3.14159
          0.84147
```

Format standarnya adalah 12.

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

```
  0.333333333333
```

Fungsi-fungsi seperti shortestformat, shortformat, dan longformat bekerja untuk vektor dengan cara berikut.

```
>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, ini dapat diubah.

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

```
  3.1416
```

Fungsi longestformat juga mengatur format skalar.

```
>longestformat; pi
```

```
  3.141592653589793
```

Sebagai referensi, berikut adalah daftar format keluaran yang paling penting.

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

Akurasi internal EMT sekitar 16 angka desimal sesuai dengan standar IEEE. Angka-angka disimpan dalam format internal ini.

Namunn, format keluaran EMT dapat diatur dengan cara yang fleksibel.

```
>longestformat; pi,
```

```
3.141592653589793
```

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

```
3.14159
```

Format default adalah defformat().

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

Ada operasi singakt yang hanya mencetak satu nilai. Operasi longest akan mencetak seluruh digit valid dari sebuah angka.

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

```
4.934802200544679
```

Ada juga operator singkat untuk mencetak hasil dalam format pecahan. Kita sudah 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 direpresentasikan secara tepat. Kesalahan ini akan terakumulasi sedikit, seperti yang terlihat pada 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, hal ini tidak akan terlihat. Untuk kenyamanan, keluaran 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
```

String atau nama dapat digunakan untuk menyimpan ekspresi matematika yang dapat dievaluasi oleh EMT. Untuk ini, gunakan tanda kurung setelah ekspresi. Jika berniat menggunakan string sebagai ekspresi, gunakan konversi untuk menamainya "fx" atau "fxy" dan sebagainya. Ekspresi memiliki prioritas lebih tinggi daripada fungsi.

Variabel global dapat digunakan dalam proses evaluasi.

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

```
            12.56637061435917
```

Parameter ditetaapkan ke x, y, dan z secara berurutan. Parameter tambahan dapat ditambahkan menggunakan parameter yang sudah ditetapkan.

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

```
     -0.919535764538
```

Perlu diperhatikan bahwa ekspresi akan selalu menggunakan variabel global, meskipun ada variabel dalam fungsi dengan nama sama.
(Jika tidak, evaluasi ekspresi bisa 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" selain nili 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, kami menekankan bahwa call collections (dibahas di bagian 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 pada x sering digunakan seperti fungsi.
Perlu dicatat bahwa mendefinisikan fungsi dengan nama yang sama seperti simbolik global akan menghapus variabel tersebut untuk menghindari kebingungan antara ekspresi simbolik dan fungsi.

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

```
     12
```

Sebagai konversi, ekspresi simbolik atau numerik sebaiknya dinamai fx, fxy, dan sebagainya. Skema penamaan ini tidak boleh digunakan untuk fungsi.

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

$$x^x\ (\log x+1)$$

Bentuk khusus dari ekspresi memungkinkan variabel apapun digunakan sebagai parameter tanpa nama untuk evaluasi ekspresi, tidak hanya x, y, dan sebagainya. Untuk ini, mulailah ekspresi dengan @(variables) ... .

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

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

Hal ini memungkinkan manipulasi ekspresi dalam variabel lain untuk fungsi-fungsi EMT yang memerlukan ekspresi dalam x.

Cara paling sederhana untuk mendefinisikan sebuah fungsi sederhana adalah dengan menyimpan rumusnya dalam ekspresi simbolik atau numerik. Jika variabel utamanya adalah x, ekspresi tersebut dapat dievaluasi seperti sebuah fungsi.

Seperti yang Anda lihat pada contoh berikut, variabel global terlihat selama proses evaluasi.

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

```
-0.475
```

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

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

```
-0.425
```

Sebuah ekspresi tidak harus simbolik. Hal ini diperlukan jika ekspresi mengandung fungsi-fungsi yang hanya dikenal di kernel numerik, bukan di Maxima.

Matematika Simbolik

EMT melakukan matematika simbolik dengan bantuan Maxima. Untuk detailnya, mulailah dengan tutorial berikut, atau jelajahi referensi untuk Maxima. Para ahli Maxima perlu mencatat bahwa terdapat perbedaan sintaks antara sintaks asli Maxima dan sintaks default ekspresi simbolik di EMT.

Matematika simbolik terintegrasi secara mulus ke dalam Euler dengan &. Setiap ekspresi yang diawali dengan & adalah ekspresi simbolik. Ekspresi ini dievaluasi dan dicetak oleh Maxima.

Pertama-tama, Maxima memiliki aritmetika "tak terbatas" yang dapat menangani 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 bagian numerik dari EMT).

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

$$2481256778$$

Untuk mempelajari lebih lanjut tentang fungsi tertentu, klik ganda pada fungsi tersebut. Misalnya, coba klik ganda pada "&binomial" di baris perintah sebelumnya. Ini akan membuka dokumentasi Maxima sebagaimana disediakan oleh pembuat program tersebut.

Anda akan mempelajari bahwa hal berikut juga bisa dilakukan:

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

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

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

Jika Anda ingin mengganti x dengan nilai tertentu, gunakan with.

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

$$120$$

Dengan cara ini, Anda dapat menggunakan solusi dari suatu persamaan dalam persamaan lain.

Ekspresi simbolik dicetak oleh Maxima dalam bentuk 2D. Hal ini disebabkan oleh flag simbolik khusus dalam string.

Seperti yang mungkin sudah Anda lihat pada contoh sebelumnya dan berikutnya, jika Anda memiliki LaTeX terpasang, Anda dapat mencetak ekspresi simbolik dengan LaTeX. Jika tidak, perintah berikut akan menghasilkan 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 terpasang.

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

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

Ekspresi simbolik diparse oleh Euler. Jika Anda membutuhkan sintaks kompleks dalam satu ekspresi, Anda dapat membungkus ekspresi tersebut dengan "...". Menggunakan lebih dari sekadar ekspresi sederhana memungkinkan, tetapi sangat tidak disarankan.

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

25

Sebagai pelengkap, kami mencatat bahwa ekspresi simbolik dapat digunakan dalam program, tetapi perlu dibungkus dalam tanda kutip. Selain itu, akan jauh lebih efektif untuk memanggil Maxima pada saat compile time jika memungkinkan.

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

$$x^4+4\,x^3+6\,x^2+4\,x+1$$

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

Sekali lagi, % merujuk pada hasil sebelumnya.

Untuk mempermudah, kita menyimpan 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 mode kompatibilitas).

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

2432902008176640000

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

$$2^8\,3^4\,5^2\,7$$

```
>:: 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 semacam ini dapat digunakan dalam ekspresi simbolik lainnya. Perlu dicatat, bahwa pada perintah berikut, sisi kanan dari &= dievaluasi terlebih dahulu sebelum dilakukan penugasan ke Fx.

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

```
                                   5
                              125 E
```

$$125\,e^5$$

```
                          18551.64488782208
```

```
>fx(5)
```

```
  18551.6448878
```

Untuk evaluasi suatu ekspresi dengan nilai-nilai tertentu dari 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))
```

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

Untuk mendapatkan kode LaTeX dari sebuah ekspresi, Anda dapat menggunakan perintah tex.

```
>tex(fx)
```

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

Ekspresi simbolik dapat dievaluasi sama seperti ekspresi numerik.

```
>fx(0.5)
```

```
0.206090158838
```

Dalam ekspresi simbolik, hal ini tidak berfungsi, karena Maxima tidak mendukungnya. Sebagai gantinya, gunakan sintaks with (bentuk yang lebih rapi dari perintah at(...) di Maxima).

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

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

Penugasan juga dapat bersifat simbolik.

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

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

Perintah solve menyelesaikan ekspresi simbolik untuk suatu variabel di Maxima. Hasilnya berupa vektor solusi.

```
>$&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 di Euler, yang membutuhkan nilai awal, dan opsional nilai target.

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

```
1.56155281281
```

Nilai numerik dari solusi simbolik dapat dihitung dengan evaluasi hasil simbolik. Euler akan membaca penugasan seperti x= dan sebagainya. Jika Anda tidak memerlukan hasil numerik untuk perhitungan lebih lanjut, Anda juga dapat membiarkan Maxima menemukan nilai numeriknya.

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

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

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

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

Untuk mendapatkan solusi simbolik tertentu, Anda dapat menggunakan with beserta indeks.

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

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

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

Untuk menyelesaikan sistem persamaan, gunakan vektor persamaan. Hasilnya adalah vektor solusi.

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

$$[[x=2,y=1],[x=1,y=2]]$$

$$2$$

Ekspresi simbolik dapat memiliki bendera, yang menandakan perlakuan khusus dalam Maxima. Beberapa bendera dapat digunakan sebagai perintah, sementara yang lain tidak. Bendera ditambahkan 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(%)
```

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

Fungsi

Dalam EMT, fungsi adalah program yang didefinisikan dengan perintah “function”. Fungsi dapat berupa fungsi satu baris atau fungsi multi-baris.
Fungsi satu baris dapat bersifat numerik atau simbolik. Fungsi numerik satu baris didefinisikan dengan “:=”.

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

Untuk gambaran umum, kami menampilkan semua definisi yang mungkin untuk fungsi satu baris. Sebuah fungsi dapat dievaluasi sama seperti fungsi Euler bawaan.

```
>f(2)
```

```
  4.472135955
```

Fungsi ini juga akan berfungsi untuk vektor, mengikuti bahasa matriks Euler, karena ekspresi yang digunakan dalam fungsi tersebut telah di-vectorize.

```
>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 dapat digambar. Alih-alih menggunakan ekspresi, kita hanya perlu memberikan nama fungsi.

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 mengganti fungsi bawaan, Anda harus menambahkan kata kunci “overwrite”. Mengganti fungsi bawaan dapat berbahaya dan dapat menyebabkan masalah bagi fungsi lain yang bergantung padanya.

Anda masih dapat memanggil fungsi bawaan sebagai “_...”, jika fungsi tersebut berada di inti Euler.

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

```
  0.707106781187
```

Sebaiknya kita menghapus redefinisi 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
```

Jika parameter ini diabaikan, nilai default akan digunakan.

```
>f(4)
```

```
  16
```

Menetapkannya akan menimpa nilai default.

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

```
  80
```

Parameter yang ditetapkan juga akan menimpanya. Hal ini digunakan oleh banyak fungsi Euler seperti plot2d, plot3d.

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

```
  16
```

Jika sebuah variabel bukan parameter, maka harus global. Fungsi satu baris dapat melihat variabel global.

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

```
  24
```

Namun, parameter yang ditetapkan akan menimpa nilai global.

Jika argumen tidak ada dalam daftar parameter yang telah ditentukan, maka harus dideklarasikan dengan ":="!

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

```
  20
```

Fungsi simbolik didefinisikan dengan "&=". Fungsi ini didefinisikan di Euler dan Maxima, dan bekerja di kedua lingkungan. Ekspresi yang mendefinisikannya dieksekusi melalui Maxima sebelum definisi dilakukan.

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

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

Fungsi simbolik dapat digunakan dalam ekspresi simbolik.

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

$$x\,e^{-x}-e^{-x}+3\,x^2$$

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

Fungsi simbolik juga dapat digunakan dalam ekspresi numerik. Tentu saja, hal ini hanya akan berhasil jika EMT dapat menafsirkan semua yang ada di dalam fungsi.

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

```
  178.635099908
```

Fungsi simbolik dapat digunakan untuk mendefinisikan fungsi atau ekspresi simbolik 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
```

Hal berikut juga berfungsi, karena Euler menggunakan ekspresi simbolik dalam fungsi g, jika tidak menemukan variabel simbolik g, dan jika ada fungsi simbolik g.

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

```
  0.703467422498
```

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

$$(2\,x-1)^n$$

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

$$(x+2)^n$$

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

$$(2\,x-1)^4$$

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

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

625

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

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

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

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

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

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

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

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

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

$$16\,x^7+64\,x^6+24\,x^5-120\,x^4-15\,x^3+102\,x^2-52\,x+8$$

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

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

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

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

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

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

$$x^3 - x$$

Dengan &=, fungsi bersifat simbolik, dan dapat digunakan dalam ekspresi simbolik lainnya.

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

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

Dengan :=, fungsi bersifat numerik. Contoh yang baik adalah integral tertentu seperti

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

yang tidak dapat dievaluasi secara simbolik.

Jika kita mendefinisikan ulang fungsi dengan kata kunci map, fungsi tersebut dapat digunakan untuk vektor x. Secara internal, fungsi dipanggil untuk semua nilai x sekaligus, dan hasilnya disimpan dalam sebuah 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 parameternya.

```
>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 ditetapkan.

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

```
2
```

Seringkali, kita ingin menggunakan fungsi untuk vektor di satu tempat, dan untuk elemen individu di tempat lain. Hal ini dimungkinkan dengan parameter vektor.

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

$$b^2 - a\,b + b + a^2$$

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

Fungsi simbolik semacam ini dapat digunakan untuk variabel simbolik.

Namun, fungsi tersebut juga dapat digunakan untuk vektor numerik.

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

```
17
```

Ada juga fungsi murni simbolik, 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)
```

$$y^4 - 6\,x^2\,y^2 + x^4$$

$$0$$

Namun tentu saja, fungsi tersebut dapat digunakan dalam ekspresi simbolik atau dalam pendefinisian fungsi simbolik.

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

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

Ringkasan

- &= mendefinisikan fungsi simbolik,
- := mendefinisikan fungsi numerik,
- &&= mendefinisikan fungsi murni simbolik.

Menyelesaikan Ekspresi

Ekspresi dapat diselesaikan secara numerik maupun simbolik.
Untuk menyelesaikan ekspresi sederhana dengan satu variabel, kita dapat menggunakan fungsi solve().
Fungsi ini membutuhkan nilai awal untuk memulai pencarian. Secara internal, solve() menggunakan metode secant.

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

```
1.41421356237
```

Hal ini juga berfungsi untuk ekspresi simbolik. Ambil contoh fungsi berikut.

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

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

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

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

```
>$&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])
```

$$\left[\left[x=-\frac{c\,e}{b\,(d-5)-a\,e},y=\frac{c\,(d-5)}{b\,(d-5)-a\,e}\right]\right]$$

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

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

Sekarang kita mencari titik di mana polinomial sama dengan 2. Dalam solve(), nilai target default y=0 dapat diubah dengan variabel yang ditetapkan.
ita menggunakan y=2 dan memeriksanya dengan mengevaluasi polinomial pada hasil sebelumnya.

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

```
  0.966715594851
  2
```

Menyelesaikan ekspresi simbolik dalam bentuk simbolik akan mengembalikan daftar solusi. Kita menggunakan pemecah 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 paling mudah untuk mendapatkan nilai numerik adalah dengan mengevaluasi solusi secara numerik sama seperti mengevaluasi sebuah ekspresi.

```
>longest sol()
```

```
      -0.6180339887498949       1.618033988749895
```

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

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

$$\frac{\left(\sqrt{5}-1\right)^2}{4}$$

$$0$$

Menyelesaikan sistem persamaan secara simbolik dapat dilakukan dengan menggunakan vektor persamaan dan pemecah simbolik solve(). Hasilnya berupa 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 melihat variabel global. Namun 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 mengoper parameter tambahan ke f() adalah dengan menggunakan daftar yang berisi nama fungsi dan parameternya (cara lainnya adalah menggunakan parameter titik koma).

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

```
  2.54116291558
```

Hal ini juga berfungsi untuk ekspresi. Namun, dalam hal ini, harus digunakan elemen daftar yang bernama. (Lebih lanjut tentang daftar terdapat pada 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
```

$$\left[1

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

$$\left[-1

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

$$\left[-1

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

$$\left[x<6 \right] \lor \left[6

```
>$&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])
```

$$\left[10 \right] \lor \left[x<1 , -x^2-x-1>0 \right] $$

```
>$&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
```

$$\left[y-5

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

$$\left[{\it max}\left(10 , x-7\right)

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

$$\left[y+8

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

$$\left[y+8

```
>&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])
```

$$\left[x=12 \right] \lor \left[12

Bahasa Matriks

Dokumentasi inti EMT memuat diskusi mendetail tentang bahasa matriks di Euler.

Vektor dan matriks dimasukkan dengan tanda kurung siku, elemen dipisahkan dengan koma, dan baris dipisahkan dengan titik koma.

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

```
                1              2
                3              4
```

Perkalian matriks ditandai 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
```

Inti dari bahasa matriks adalah bahwa semua fungsi dan operator bekerja 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, melainkan perkalian elemen demi elemen. Hal yang sama berlaku untuk vektor.

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

```
                9
               16
```

Jika salah satu operand adalah vektor atau skalar, maka akan diperluas secara alami.

```
>2*A
```

```
                2              4
                6              8
```

Misalnya, jika operand adalah vektor kolom, elemennya akan diterapkan ke semua baris A.

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

```
                1              4
                3              8
```

Jika itu adalah vektor baris, maka akan diterapkan ke semua kolom A.

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

```
                2              6
                6             12
```

Kita dapat membayangkan perkalian ini seolah-olah vektor baris v diduplikasi untuk membentuk matriks dengan ukuran yang sama dengan 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
```

Hal ini juga berlaku untuk dua vektor di mana satu adalah vektor baris dan yang lain vektor kolom. Kita menghitung i*j untuk i,j dari 1 hingga 5. Triknya adalah dengan mengalikan 1:5 dengan transposenya. 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 tertentu dengan fungsi sum().

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

```
 5
```

Euler memiliki operator perbandingan, seperti ==, yang digunakan untuk memeriksa kesetaraan.

Kita akan mendapatkan vektor berisi 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 ini, "nonzeros" memilih elemen yang tidak 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-nilai yang sesuai di t.

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

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

Sebagai contoh, mari kita temukan semua kuadrat dari angka 1 hingga 1000, yang modulo 11 sama dengan 5 dan modulo 13 sama dengan 3.

```
>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, EMT menggunakan floating point presisi ganda. Namun, ini sering sangat berguna.

Kita dapat memeriksa bilangan prima. 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, digunakan 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 ini mengembalikan indeks dari elemen-elemen yang bukan 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 menetapkan elemen-elemen ke suatu nilai.

```
>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 menetapkan elemen pada indeks tertentu ke nilai-nilai 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 dimungkinkan untuk mengambil elemen-elemen tersebut ke dalam sebuah vektor.

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

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

Fungsi lain yang berguna adalah extrema, yang mengembalikan nilai minimum dan maksimum di setiap baris matriks beserta posisinya.

```
>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 mengambil nilai maksimum 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 mengambil indeks dan menggunakan informasi ini untuk mengambil 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]
```

Fungsi Matriks Lainnya (Membangun Matriks)

Untuk membangun matriks, kita dapat menumpuk satu matriks di atas matriks lain. Jika keduanya tidak memiliki jumlah kolom yang sama, matriks yang lebih pendek akan diisi dengan 0.

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

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

Demikian pula, kita dapat menempelkan matriks satu sama lain secara berdampingan, 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 keduanya tidak memiliki jumlah baris yang sama, matriks yang lebih pendek akan diisi dengan 0.

Ada pengecualian untuk aturan ini. Bilangan real yang ditempelkan 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 dari vektor baris dan vektor kolom.

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

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

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

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

Tujuan utama dari ini adalah untuk menafsirkan vektor ekspresi sebagai vektor kolom.

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

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

Untuk mendapatkan ukuran A, kita dapat menggunakan fungsi-fungsi berikut.

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

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

Untuk vektor, ada fungsi length().

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

```
  9
```

Ada banyak fungsi lain yang menghasilkan matriks.

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

```
                1              1
                1              1
```

Ini juga dapat digunakan dengan satu parameter. Untuk mendapatkan vektor dengan angka selain 1, gunakan cara berikut.

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

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

Selain itu, matriks angka acak dapat dihasilkan dengan random (distribusi uniform) atau normal (distribusi Gaussian).

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

```
          0.66566       0.831835
            0.977       0.544258
```

Berikut adalah fungsi lain yang berguna, yang menyusun ulang elemen-elemen matriks menjadi 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 dapat menggunakan ini dan fungsi dup untuk menulis 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-elemen dari sebuah 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() membalik urutan baris atau kolom dari sebuah matriks.
Artinya, fungsi flipx() membalik secara horizontal.

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

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

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

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

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

Fungsi khusus adalah drop(v,i), yang menghapus elemen dengan indeks i dari vektor v.

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

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

Catatan: vektor i dalam drop(v,i) mengacu pada indeks elemen dalam vektor v, bukan pada nilai elemen itu sendiri.

Jika ingin menghapus elemen berdasarkan nilainya, maka elemen tersebut harus dicari terlebih dahulu. Fungsi indexof(v,x) dapat digunakan untuk menemukan elemen x di dalam vektor v 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 terlihat, penggunaan **`drop(v, i)`** tidak akan menimbulkan masalah meskipun:

- terdapat indeks di luar jangkauan (misalnya **0** atau lebih besar dari panjang vektor),
- terdapat indeks ganda (duplikasi),
- ataupun indeks tidak berurutan.

Fungsi tetap berjalan, hanya saja elemen yang sesuai indeks valid saja yang akan benar-benar dihapus.

```
>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) menjadi 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
```

Perlu dicatat bahwa kita tidak mengubah matriks A. Kita mendapatkan matriks baru sebagai hasil dari setdiag().

Berikut adalah sebuah fungsi yang mengembalikan matriks tridiagonal.

```
>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 dari sebuah matriks juga dapat diekstrak dari matriks tersebut. Untuk mendemonstrasikan hal ini, kita menyusun kembali vektor 1:9 menjadi matriks 3x3.

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

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

Sekarang kita dapat mengekstrak diagonalnya.

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

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

Misalnya, kita dapat membagi matriks dengan diagonalnya. Bahasa matriks memastikan bahwa vektor kolom d diterapkan pada matriks baris demi baris.

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

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

Vektorisasi

Hampir semua fungsi dalam Euler juga bekerja untuk masukan matriks dan vektor, kapan pun hal itu masuk akal.

Misalnya, fungsi sqrt() menghitung akar kuadrat dari semua elemen vektor atau matriks.

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

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

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

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

Dengan ini dan operator titik dua a:delta:b, vektor nilai dari fungsi dapat dengan mudah dihasilkan.

Dalam contoh berikut, kita menghasilkan sebuah vektor nilai t[i] dengan jarak 0,1 dari -1 hingga 1.

Kemudian kita menghasilkan sebuah vektor nilai dari 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 memperluas operator untuk skalar, vektor, dan matriks dengan cara yang jelas.

Misalnya, sebuah vektor kolom dikalikan dengan vektor baris menghasilkan matriks, jika suatu operator diterapkan. Pada contoh berikut, v' adalah vektor transpose (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
```

Perlu dicatat bahwa ini sangat berbeda dari hasil kali matriks. Hasil kali matriks dalam EMT dinotasikan dengan sebuah titik “.”

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

```
55
```

Secara bawaan, vektor baris ditampilkan dalam format yang ringkas.

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

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

Untuk matriks, operator khusus . menyatakan perkalian matriks, dan A' menyatakan transpose. Matriks berukuran 1x1 dapat digunakan sama seperti bilangan real.

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

```
5
25
```

Untuk mentranspos matriks kita menggunakan tanda apostrof.

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

```
1
2
3
4
```

Jadi kita dapat menghitung matriks A dikalikan vektor b.

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

```
                30
                70
```

Perhatikan bahwa v masih berupa 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 v kuadrat untuk vektor baris v. Hasilnya adalah vektor 1x1, yang berfungsi sama seperti bilangan real.

```
>v.v'
```

```
  30
```

Ada juga fungsi norm (bersama dengan banyak fungsi lainnya dalam Aljabar Linear).

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

```
  30
```

Operator dan fungsi mengikuti bahasa matriks dari Euler.

Berikut adalah ringkasan aturannya:

- Suatu fungsi yang diterapkan pada vektor atau matriks akan diterapkan pada setiap elemennya.
- Suatu operator yang bekerja pada dua matriks dengan ukuran sama akan diterapkan secara berpasangan pada elemen-elemen dari matriks tersebut.
- Jika kedua matriks memiliki dimensi berbeda, keduanya diperluas dengan cara yang masuk akal sehingga memiliki ukuran yang sama.

Sebagai contoh, sebuah nilai skalar dikalikan dengan vektor berarti nilai tersebut dikalikan dengan setiap elemen vektor. Atau sebuah matriks dikalikan dengan vektor (dengan *, bukan .) memperluas vektor tersebut ke ukuran matriks dengan menduplikasinya.

Berikut adalah kasus sederhana dengan operator ^.

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

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

Berikut adalah kasus yang lebih rumit. Sebuah vektor baris dikalikan dengan vektor kolom diperluas dengan cara menduplikasi keduanya.

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

```
                1               2               3
                2               4               6
                3               6               9
```

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

```
>v.v'
```

```
  14
```

Ada banyak fungsi untuk matriks. Berikut adalah daftar singkatnya. Untuk informasi lebih lanjut mengenai perintah-perintah ini, sebaiknya melihat dokumentasi.

- sum, prod : menghitung jumlah dan hasil kali dari baris-baris
- cumsum, cumprod : melakukan hal yang sama secara kumulatif
- menghitung nilai ekstrem dari setiap baris
- extrema : mengembalikan sebuah vektor dengan informasi nilai ekstrem
- diag(A,i) : mengembalikan diagonal ke-i
- setdiag(A,i,v): menetapkan diagonal ke-i
- id(n) : matriks identitas
- det(A): determinan
- charpoly(A) : polinomial karakteristik
- eigenvalues(A) : nilai eigen

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

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

Operator ":" menghasilkan vektor baris dengan jarak antar elemen yang sama, dengan opsi menentukan ukuran langkah.

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

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

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

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

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

Elemen-elemen dari sebuah matriks direferensikan 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]** adalah elemen ke-i dari vektor. Untuk matriks, ini mengembalikan seluruh baris ke-i dari matriks.

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

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

Indeks juga dapat berupa vektor baris dari indeks. ":" menyatakan semua indeks.

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

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

Bentuk singkat dari : adalah dengan menghilangkan indeks sepenuhnya.

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

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

Untuk keperluan vektorisasi, elemen-elemen dari sebuah matriks dapat diakses seolah-olah mereka adalah vektor.

```
>A{4}
```

```
  4
```

Sebuah matriks juga dapat diratakan menggunakan fungsi redim(). Hal ini diimplementasikan dalam 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, mari kita atur kembali ke format default, lalu hitung tabel nilai sinus dan kosinus. Perlu dicatat bahwa sudut secara default dinyatakan 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 sebuah 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
```

Dengan menggunakan bahasa matriks, kita dapat 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, di mana setiap baris adalah tabel t^i untuk satu nilai i. Artinya, matriks tersebut memiliki 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 input berupa vektor harus “divektorisasi”. Hal ini dapat dilakukan dengan kata kunci map dalam definisi fungsi. Dengan demikian, fungsi tersebut akan dievaluasi untuk setiap elemen dari parameter vektor.

Integrasi numerik integrate() hanya bekerja untuk batas interval skalar. Oleh karena itu, kita perlu melakukan vektorisasi terhadapnya.

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

Kata kunci map melakukan vektorisasi fungsi. Dengan ini, fungsi akan dapat bekerja untuk vektor bilangan.

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

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

Sub-Matriks dan Elemen Matriks

Untuk mengakses sebuah 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 dapat mengakses satu baris lengkap dari sebuah matriks.

```
>A[2]
```

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

Dalam kasus vektor baris atau kolom, ini akan mengembalikan sebuah elemen dari vektor.

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

```
2
```

Untuk memastikan bahwa yang diambil adalah baris pertama dari matriks berukuran 1×n maupun m×n, tentukan semua kolom dengan menggunakan indeks kedua yang kosong.

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

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

Jika indeks berupa vektor indeks, Euler akan mengembalikan baris-baris matriks yang bersesuaian.

Di sini kita ingin mengambil baris pertama dan kedua dari matriks A.

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

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

Kita bahkan dapat menyusun ulang A dengan menggunakan vektor indeks. Lebih tepatnya, kita tidak mengubah A secara langsung, melainkan menghitung versi A yang sudah disusun ulang.

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

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

Trik indeks juga dapat digunakan pada kolom.

Contoh berikut memilih semua baris dari A serta kolom kedua dan ketiga.

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

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

Sebagai singkatan, : menyatakan semua indeks baris atau kolom.

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

```
                3
                6
                9
```

Sebagai alternatif, biarkan indeks pertama kosong.

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

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

Kita juga dapat mengambil baris terakhir dari A.

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

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

Sekarang mari kita ubah elemen-elemen dari A dengan memberikan nilai pada sebuah submatriks dari A. Hal ini benar-benar akan mengubah matriks A yang tersimpan.

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

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

Kita juga dapat memberikan sebuah nilai pada satu baris dari A.

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

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

Kita bahkan dapat memberikan nilai pada sebuah submatriks jika ukurannya sesuai.

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

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

Selain itu, beberapa bentuk singkatan juga diperbolehkan.

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

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

Peringatan:
Indeks yang berada di luar batas akan menghasilkan matriks kosong, atau pesan kesalahan, tergantung pada pengaturan sistem. Secara default, hasilnya adalah pesan kesalahan. Namun, perlu diingat bahwa indeks negatif dapat digunakan untuk mengakses elemen matriks dengan menghitung dari bagian akhir.

```
>A[4]
```

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

Pengurutan dan Pengacakan

Fungsi sort() mengurutkan sebuah vektor baris.

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

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

Sering kali diperlukan untuk mengetahui indeks dari vektor yang sudah diurutkan terhadap vektor aslinya. Hal ini dapat digunakan untuk menyusun ulang vektor lain dengan cara yang sama.

Sekarang mari kita acak sebuah vektor.

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

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

Indeks menyimpan urutan yang benar dari v.

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

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

Hal ini juga berlaku untuk vektor string.

```
>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 terlihat, posisi dari entri ganda bersifat agak acak.

```
>ind
```

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

Fungsi unique mengembalikan daftar elemen unik dari sebuah vektor yang sudah diurutkan.

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

```
[4,  4,  9,  2,  6,  5,  10,  6,  5,  1]
[1,  2,  4,  5,  6,  9,  10]
```

Hal ini juga berlaku untuk vektor string.

```
>unique(s)
```

```
a
aa
d
e
```

Aljabar Linear

EMT memiliki banyak fungsi untuk menyelesaikan sistem linier, sistem sparse, atau masalah regresi. Untuk sistem linier Ax=b, Anda dapat menggunakan algoritma Gauss, matriks invers, atau fitting 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 membuat matriks 200×200 dan menghitung jumlah tiap barisnya. Kemudian kita menyelesaikan Ax = b menggunakan matriks invers. Kita mengukur kesalahan sebagai deviasi maksimal dari semua elemen terhadap 1, yang tentu saja adalah solusi yang benar.

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

```
8.790745908981989e-13
```

Jika sistem tidak memiliki solusi, fitting linier meminimalkan norma dari kesalahan Ax - b

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

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

Determinannya matriks ini adalah 0.

```
>det(A)
```

```
0
```

Matriks Simbolik

Maxima memiliki matriks simbolik. Tentu saja, Maxima dapat digunakan untuk masalah aljabar linear sederhana. Kita dapat mendefinisikan matriks untuk Euler dan Maxima dengan &:=, lalu menggunakannya dalam ekspresi simbolik. Bentuk biasa [...] untuk mendefinisikan matriks juga dapat digunakan di Euler untuk mendefinisikan matriks simbolik.

```
>A &= [a,1,1;1,a,1;1,1,a]; $A
>$&det(A), $&factor(%)
>$&invert(A) with a=0
>A &= [1,a;b,2]; $A
```

Seperti semua variabel simbolik, matriks-matriks ini dapat digunakan dalam ekspresi simbolik lainnya.

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

Nilai eigen juga dapat dihitung secara otomatis. Hasilnya adalah sebuah vektor yang berisi dua vektor: nilai eigen dan multiplikitasnya.

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

Untuk mengekstrak sebuah vektor eigen tertentu diperlukan penentuan indeks yang teliti.

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

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

Matriks simbolik dapat dievaluasi secara numerik di Euler sama seperti ekspresi simbolik lainnya.

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

```
                 1              4
                 5              2
```

Dalam ekspresi simbolik, gunakan with.

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

Akses ke baris matriks simbolik bekerja sama seperti pada matriks numerik.

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

Sebuah ekspresi simbolik dapat berisi sebuah penugasan. Dan itu akan mengubah matriks A.

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

Terdapat fungsi simbolik di Maxima untuk membuat vektor dan matriks. Untuk hal ini, lihat dokumentasi Maxima atau tutorial tentang Maxima dalam EMT.

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

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

Hasilnya dapat dievaluasi secara numerik di Euler. Untuk informasi lebih lanjut tentang Maxima, lihat pengantar tentang Maxima.

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

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

Euler juga memiliki fungsi xinv() yang lebih kuat, melakukan perhitungan lebih mendalam, dan menghasilkan hasil yang lebih akurat.

Perlu dicatat, dengan &:=, matriks B telah didefinisikan sebagai simbolik dalam ekspresi simbolik dan sebagai numerik dalam ekspresi numerik. Jadi kita dapat menggunakannya di sini.

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

```
                          1                          0
                          0                          1
```

Sebagai contoh, nilai eigen dari 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. Lihat tutorial tentang Maxima untuk detailnya.

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

Nilai Numerik dalam Ekspresi Simbolik

Ekspresi simbolik hanyalah sebuah string yang berisi sebuah ekspresi. Jika kita ingin menetapkan nilai baik untuk ekspresi simbolik maupun ekspresi numerik, kita harus menggunakan "&:=".

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

```
                1        3.14159
                4              5
```

Masih ada perbedaan antara bentuk numerik dan simbolik. Saat mentransfer matriks ke bentuk simbolik, pendekatan pecahan untuk bilangan real akan digunakan.

```
>$&A
```

Untuk menghindari hal ini, terdapat fungsi "mxmset(variable)".

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

Maxima juga dapat menghitung dengan bilangan titik mengambang, bahkan dengan bilangan titik mengambang besar hingga 32 digit. Namun, evaluasinya jauh lebih lambat.

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

Presisi dari bilangan titik mengambang besar dapat diubah.

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

```
          3.14159265358979323846264338327950288419716939937510582097494\
 459230781640628620899862803482534211706 8b0
```

Variabel numerik dapat digunakan dalam ekspresi simbolik mana pun dengan "@var".

Perlu dicatat bahwa ini hanya diperlukan jika variabel telah didefinisikan dengan ":=" atau "=" sebagai variabel numerik.

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

Demo - Suku Bunga

Berikut, kita menggunakan Euler Math Toolbox (EMT) untuk perhitungan suku bunga. Kita akan melakukannya secara numerik dan simbolik untuk menunjukkan bagaimana Euler dapat digunakan untuk menyelesaikan masalah kehidupan nyata.

Misalkan Anda memiliki modal awal sebesar 5000 (misalnya dalam dolar).

```
>K=5000
```

```
  5000
```

Sekarang kita asumsikan suku bunga 3% per tahun. Mari kita tambahkan satu tingkat bunga sederhana dan hitung hasilnya.

```
>K*1.03
```

```
  5150
```

Euler juga akan memahami sintaks berikut.

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

```
  5150
```

Namun, lebih mudah menggunakan faktor.

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

```
  1.03
  5150
```

Untuk 10 tahun, kita dapat langsung mengalikan faktor-faktor tersebut dan mendapatkan nilai akhir dengan suku bunga majemuk.

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

```
  6719.58189672
```

Untuk keperluan kita, kita dapat mengatur format menjadi 2 angka di belakang koma.

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

```
       6719.58
```

Mari kita cetak hasil tersebut dibulatkan menjadi 2 angka di belakang koma dalam satu kalimat 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 dari tahun 1 hingga tahun 9? Untuk hal ini, bahasa matriks Euler sangat membantu. Anda tidak perlu menulis loop, cukup masukkan saja

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

```
  Real 1 x 11 matrix
       5000.00      5150.00      5304.50      5463.64      ...
```

Bagaimana keajaiban ini bekerja? Pertama, ekspresi 0:10 mengembalikan sebuah vektor bilangan bulat.

```
>short 0:10
```

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

Kemudian semua operator dan fungsi di Euler dapat diterapkan pada vektor secara elemen per elemen. Jadi,

```
>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]
```

adalah sebuah vektor faktor dari
q^0 hingga

10
Vektor ini dikalikan dengan
K, sehingga kita mendapatkan vektor nilai-nilainya.

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

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

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

Mari kita bandingkan kedua hasil tersebut, 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, sehingga kita harus melakukan loop selama beberapa tahun. Euler menyediakan banyak solusi untuk hal ini.

Cara termudah adalah menggunakan fungsi iterate, yang menjalankan sebuah fungsi tertentu sejumlah 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 lebih ramah, menggunakan format kita dengan jumlah angka 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 mengambil elemen tertentu dari vektor, kita menggunakan indeks dalam tanda kurung siku.

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

```
         5150.00
         5000.00      5150.00      5304.50
```

Mengejutkan, 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.

Menyelesaikan Persamaan

Sekarang kita menggunakan fungsi yang lebih lanjut, yang menambahkan sejumlah suku bunga setiap tahun.

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

Kita tidak perlu menentukan nilai q atau R saat mendefinisikan fungsi. Hanya saat menjalankan perintah, kita harus menetapkan nilai-nilai ini. Kita pilih R = 200.

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

```
    Real 1 x 11 matrix
         5000.00      5350.00      5710.50      6081.82      ...
```

Bagaimana jika kita mengurangi jumlah yang sama setiap tahun?

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

```
    Real 1 x 11 matrix
```

```
      5000.00      4950.00      4898.50      4845.45      ...
```

Kita melihat bahwa uang berkurang. Jelas, jika kita hanya mendapatkan 150 dari bunga pada tahun pertama, tetapi mengurangi 200, kita akan kehilangan uang setiap tahun.

Bagaimana kita dapat menentukan berapa tahun uang tersebut akan bertahan? Kita harus menulis sebuah loop untuk ini. Cara termudah adalah melakukan iterasi cukup lama.

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

```
    Real 1 x 51 matrix
      5000.00      4950.00      4898.50      4845.45      ...
```

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

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

```
        48.00
```

Alasannya adalah bahwa nonzeros(VKR<0) rmengembalikan vektor indeks i, di mana VKR[i]<0, dan min menghitung indeks minimal.

Karena vektor selalu mulai dari indeks 1, jawabannya adalah 47 tahun.

Fungsi iterate() memiliki satu trik lagi. Fungsi ini dapat mengambil kondisi akhir sebagai argumen. Maka fungsi akan mengembalikan nilai serta jumlah iterasi yang dilakukan.

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

```
       -19.83
        47.00
```

Mari kita coba menjawab pertanyaan yang lebih ambigu. Misalkan kita tahu bahwa nilai uang menjadi 0 setelah 50 tahun. Berapa suku bunganya?

Ini adalah pertanyaan yang hanya dapat dijawab secara numerik. Di bawah ini, kita akan menurunkan rumus yang diperlukan. Anda akan melihat bahwa tidak ada rumus mudah untuk suku bunga. Namun, saat ini kita menargetkan solusi numerik.

Langkah pertama adalah mendefinisikan sebuah fungsi yang melakukan iterasi sebanyak n kali. Kita menambahkan 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 sebelumnya:

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

Namun, sekarang kita tidak lagi menggunakan nilai global dari
R dalam ekspresi kita. Fungsi seperti iterate() memiliki trik khusus di Euler. Anda dapat melewatkan nilai variabel dalam ekspresi sebagai parameter dipisahkan titik koma, dalam hal ini
P dan
R.

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 menggunakan nilai awal 3% untuk algoritmanya. Fungsi solve() selalu memerlukan nilai awal.

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

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

```
         -336.08
```

Perlu dicatat bahwa Anda tidak dapat menyelesaikan untuk jumlah tahun, karena fungsi kita mengasumsikan
n sebagai nilai 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)
```

Sekarang kita dapat melakukan iterasi terhadap fungsi ini.

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

Kita melihat sebuah pola. Setelah
?
periode, kita memiliki

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

Rumus ini adalah rumus jumlah geometri, yang sudah dikenal oleh Maxima.

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

Ini agak rumit. Jumlah tersebut dievaluasi dengan flag simpsum untuk mereduksinya menjadi sebuah pecahan.

Mari kita buat sebuah 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)
```

Fungsi ini melakukan hal yang sama seperti fungsi f kita sebelumnya. Namun, fungsinya lebih efektif.

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

```
          -19.82504734650985
          -19.82504734652684
```

Sekarang kita dapat menggunakannya untuk menanyakan waktu n. Kapan modal kita habis? Tebakan awal kita adalah 30 tahun.

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

```
           20.51
```

Jawaban ini menunjukkan bahwa modal akan negatif setelah 21 tahun.

Kita juga dapat menggunakan sisi simbolik Euler untuk menghitung rumus pembayaran.

Misalkan kita mendapatkan pinjaman sebesar K, dan melakukan n pembayaran sebesar R (dimulai setelah tahun pertama) sehingga tersisa utang K_n pada saat pembayaran terakhir. Rumus untuk hal ini jelas adalah

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

Biasanya rumus ini diberikan dalam bentuk

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

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

Kita dapat menyelesaikan secara simbolik untuk suku bunga R.

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

Seperti yang terlihat dari rumus, fungsi ini menghasilkan kesalahan titik mengambang untuk i = 0. Namun, Euler tetap dapat memplotnya.

Tentu saja, kita memiliki batas berikut.

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

Jelas, tanpa bunga kita harus membayar 10 kali angsuran sebesar 500.

Persamaan ini juga dapat diselesaikan untuk n. Tampak lebih rapi jika kita menerapkan beberapa penyederhanaan padanya.

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

PENGERJAAN SOAL-SOAL ALJABAR

Soal-soal berikut diambil dari file PDF yang terlampir pada Besmart. Namun, ada juga beberapa yang saya cantumkan dari sumber-sumber selain Besmart.

Operasi Bentuk-Bentuk Aljabar

- Penyederhanaan

Penyederhaan bentuk aljabar adalah proses mengubah ekspresi aljabar menjadi bentuk yang lebih sederhana tanpa mengubah nilainya. Tujuannya agar lebih mudah untuk dihitung dan dibaca.
Aturan yang digunakan seperti menggabungkan suku-suku yang sejenis, operasi pangkat, perkalian dan pembagian.
Contoh 1:

$$(2x+3y+z-7)+(4x-2y-z+8)+(3x+y-2z-4)$$

```
>$&ratsimp((2*x+3*y+z-7)+(4*x-2*y+8)+(3*x+y-2*z-4))
```

Dari bentuk aljabar tersebut, ketika semua suku-suku sejenis kita jumllahkan, hasil yang diperoleh yaitu:

$$9x+2y-z-3$$

Meskipun urutannya berbeda, tetapi hasilnya tetap sama sehingga tidak mengubah nilai.

Contoh 2:

$$(x^4-3x^2+4x)-(3x^3+x^2-5x+3)$$

```
>$&ratsimp((x^4-3*x^2+4*x)-(3*x*3+x^2-5*x+3))
```

Contoh 3:

$$(5x^2+4xy-3y^2+2)-(9x^2-4xy+2y^2-1)$$

```
>$&ratsimp((5*x^2+4*x*y-3*y^2+2)-(9*x^2-4*x*y+2*y^2-1))
```

Contoh 4:

$$(3x^2-2x-x^3+2)-(5x^2-8x-x^3+4)$$

```
>$&ratsimp((3*x^2-2*x-x^3+2)-(5*x^2-8*x-x^3+4))
```

Contoh 5:

$$(2x^4-3x^2+7x)-(5x^3+2x^2-3x+5)$$

```
>$&ratsimp((2*x^4-3*x^2+7*x)-(5*x^3+2*x^2-3*x+5))
```

- Penjabaran

Penjabaran adalah mengubah bentuk aljabar yang dikalikan menjadi bentuk penjumlahan/pengurangan (membuka kurung).

Contoh 1:

$$(a+b+c)^2$$

```
>$&expand((a+b+c)^2)
```

Contoh 2:

$$[(2x-1)^2-1]^2$$

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

Contoh 3:

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

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

Contoh 4:

$$(-5m^4n^2)(6m^2n^3)$$

```
>$&expand((-5*m^4*n^2)*(6*m^2*n^3))
```

Contoh 5:

$$(6xy^3)(9x^4y^2)$$

```
>$&expand((6*x*y^3)*(9*x^4*y^2))
```

- Pemfaktoran

Pemfaktoran adalah mengubah bentuk aljabar menjadi perkalian beberapa faktor untuk mempermudah perhitungan atau mneyederhanakan bentuk aljabar.

Contoh 1:

$$7pq^4-7py^4$$

```
>$&factor(7*p*q^4-7*p*y^4)
```

Contoh 2:

$$a^3+24a^2+144a$$

```
>$&factor(a^3+24*a^2+144*a)
```

Contoh 3:

$$18a^b-15ab^2$$

```
>$&factor(18*a^2*b-15*a*b^2)
```

Contoh 4:

$$x^{2n}+5x^{n}-24$$

```
>$&factor(x^(2*n)+5*x^n-24)
```

Contoh 5:

$$12a^{2}-28=5a$$

```
>$&factor(12*a^2-28=5*a)
```

Polinomial

Polinomial atau suku banyak adalah ekspresi matematika yang terdiri dari variabel dan koefisien yang digabungkan menggunakan operasi penjumlahan, pengurangan, dan perkalian. Singkatnya, polinomial merupakan bentuk aljabar yang memiliki dua atau lebih suku dengan pangkat bulat non negatif.

Bentuk umum polinomial:

$$a_3x^{n}+a_2x^{n-1}+a_1x^{n-2}+ax+a$$

```
>$a3*x^n+a2*x^(n-1)+a1*x^(n-2)+ax+a
```

Derajat polinomial diambil dari pangkat tertinggi variabel dalam suatu ekspresi polinomial. Derajat polinomial membantu mengklasifikasikan polinomial, seperti:
> Derajat 1: polinomial linear (grafiknya gariss lurus)
> Derajat 2: Polinomial kuadratik (grafiknya parabola)
> Derajat 3: Polinomial kubik
> Derajat 4: Polinomial kuartik

Berdasarkan jumlah suku, polinomial dapat diklasifikasikan menjadi beberapa jenis.
> Monomial: polinomial dengan satu suku.
> Binomial: polinomial dengan dua suku.
> Trinomial: polinomial dengan tiga suku.
> Polinomial dengan lebih dari tiga suku biasanyaa disebut sebagai polinomial saja.

Contoh non polinomial:

$$x^{-3}$$

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

< Penjumlahan polinomial >

$$5x^{3}+3x^{2}+4x+1$$

$$7x^{2}+2x+5$$

a. Penjumlahan secara langsung di EMT

```
>$(5*x^3+3*x^2+4*x+1)+(7*x^2+2*x+5)
```

b. Menggunakan vektor koefisiennya (panjang vektor harus sama)

```
>p=[5,3,4,1]
```

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

```
>q=[0,7,2,5]
```

```
[0,  7,  2,  5]
```

```
>p+q
```

```
[5,  10,  6,  6]
```

Bisa juga menggunakan fungsi polyadd()

```
>polyadd(p,q)
```

```
[5,  10,  6,  6]
```

Note:
Panjang vektor polinomial yang dijumlahkan harus sama. Jika tidak maka hasil yang dikeluarkan (output) akan salah.

< Pengurangan polinomial >

$$3x^2 - 7x - 2$$

$$5x^3 + 3x^2 + 4x - 6$$

a. Pengurangan secara langsung di EMT

```
>$(3*x^2-7*x-2)-(5*x^3+3*x^2+4*x-6)
```

b. Menggunakan vektor koefisiennya

```
>r=[0, 3, -7, -2]
```

```
[0,  3,  -7,  -2]
```

```
>s=[5, 3, 4, -6]
```

```
[5,  3,  4,  -6]
```

```
>r-s
```

```
[-5,  0,  -11,  4]
```

Bisa juga menggunakan fungsi polyadd() dengan cara mendapatkan vektor negatif dari s terlebih dahulu dengan dikalikan -1

```
>negs=-1*s
```

```
[-5,  -3,  -4,  6]
```

```
>polyadd(r, negs)
```

```
[-5,  0,  -11,  4]
```

Note: sama seperti yang di penjumlahan tadi panjang vektor harus sama

< Perkalian polinomial >

```
>$expand((6*x-3)*(2*x+5))
```

Menggunakan fungsi polymult()

```
>t=[6, -3]
```

```
[6,  -3]
```

```
>u=[2, 5]
```

```
[2,  5]
```

```
>polymult(t, u)
```

```
[12,  24,  -15]
```

Note:
Fungsi polymult() terbatas hanya dirancang untuk mengambil 2 argumen sehingga tidak bisa mengalikan 3 polinomial sekaligus.

< Pembagian polinomial >

$$v = x^3 - x^2 - x + 1$$

$$w = x - 1$$

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

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

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

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

```
>polydiv(v,w)
```

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

Bentuk hasil polinomialnya yaitu

$$x^2 - 1$$

dengan sisa 0

Note:
Fungsi polydiv() dirancang untuk dapat menampilkan hasil sisa dari pembagian polinomial, tapi pada beberapa perangkat hasil sisa tidak muncul dan hasil pembagiannya juga salah.

Contoh soal:

```
>$p=x^3-x^2+x-1
>$q=x^3-x^2-x+1
>p=[1,-1,1,-1]
```

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

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

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

```
>polyadd(p,q)
```

```
[2,  -2,  0,  0]
```

```
>negs=-1*q
```

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

```
>polyadd(p,negs)
```

```
[0,  0,  2,  -2]
```

```
>polymult(p,q)
```

```
[1,  -2,  1,  0,  -1,  2,  -1]
```

```
>$u=2*x^4+7*x^3+x-12
>$v=x+3
>u=[-12,1,0,7,2]
```

```
[-12,  1,  0,  7,  2]
```

```
>v=[3,1]
```

```
[3,  1]
```

```
>polydiv(u,v)
```

```
[10,  -3,  1,  2]
```

Note:
Ternyata fungsi polydiv() di EMT itu, untuk mencari hasil pembagiannya kita harus mengurutkan polinomial dari belakang. Jadi konstanta ada diurutan paling depan dan seterusnya sehingga yang di polinomial berada di depan ketika diurutkan menjadi di belakang. Untuk pembagi maupun yang dibagi urutannya sama, yaitu dibalik. Nah, nanti akan ketemu hasilnya. Nanti hasilnya juga urutannya sama ketika ditulis menjadi polinomial lagi yang konstanta diletakkan di belakang. Jadi misal, di situ kita dapat hasil itu [10,-3,1,2]. Nah, dalam bentuk polinomial kita tulis menjadi

$$2x^3 + x^2 - 3x + 10$$

Oh iya, di EMT kita tidak bisa melihat berapa hasil sisa dari pembagian ini karena defaultnya memang hanya menampilkan h (quotient/hasil bagi).

Eksponen

Eksponen dalam konteks aljabar adalah bentuk pangkat yang melibatkan variabel (huruf) sebagai basis maupun eksponen.

Sifat eksponen:
> Jika a^n = a^m dan a>0, a tidak sama dengan 1 maka m=n
> Jika a^n*a^m=1, maka n+m=0
> Eksponen hanya berlaku jika variabel basis bernilai real dan tidak nol untuk pangkat negatif.

Contoh soal:

Menentukan nilai x dari persamaan eksponen berikut.

$$3^{x+1} + 3^x = 108$$

```
>$&solve(3^(x+1)+3^x-108=0,x)
>$&float(solve(3^x=27,x))
```

Contoh 2:

$$3^{x^2+4x} = \frac{1}{27}$$

```
>$&solve(3^(x^2+4*x)=1/27,x)
```

Contoh 3:

$$4^{3x-5} = 16$$

```
>$&solve(4^(3*x-5)-16=0,x)
```

Logaritma

Logaritma merupakan invers atau kebalikan dari eksponen yang digunakan untuk menentukan besaran pangkat pada sebuah bilangan pokok. Secara umum bentuk logaritma terdiri dari tiga bagian yaitu basis, numerus, dan hasil logaritma.

Bentuk umum logaritma:

$$^{a}logb = c \quad \Leftrightarrow \quad a^{c} = b$$

Namun, di EMT hanya mengerti defisini operasional (komputasi) yakni :

$$^{a}logb = \frac{log(b)}{log(a)}$$

Contoh:

$$2\log_6(16) - 3\log_6(4) + \log_6(9)$$

```
>hasil = 2*(log(16)/log(6)) - 3*(log(4)/log(6)) + log(9)/log(6)
```

2

```
>$ratsimp(2*(log(16)/log(6)) - 3*(log(4)/log(6)) + log(9)/log(6))
>$float(2*(log(16)/log(6)) - 3*(log(4)/log(6)) + log(9)/log(6))
```

Tentukan nilai x yang memenuhi persamaan

$$\log_2(5x - 9) = \log_5(5x - 9)$$

```
>&solve(log(5*x-9)/log(2) = log(5*x-9)/log(5),x)
```

[x = 2]

Tentukan nilai x dari

$$2log50 = 3log25 + log(x - 2)$$

```
>$&solve(2*log(50)=3*log(25)+log(x-2),x)
```

Vektor

Vektor merupakan besaran yang mempunyai arah.
Secara geometri, setiap vektor dinyatakan secara geometris sebagai
segmen garis berarah pada bidang atau ruang, dengan notasi garis berpanah.

Jika u,v dan w adalah vektor-vektor di dalam ruang vektor (ruang 2 atau ruang 3) dan k dan l adalah skalar, maka hubungan yang berikut akan berlaku;

> u + v = v + u
> (u + v) + w = u + (v + w)

> u + 0 = 0 + u = 0
> u + (-u)= 0
> k * (l * u) = (k * l) * u
> k (u + v) = k * u + k * v
> (k + l) * u = k * u + l * u

Contoh:

$$a = \begin{bmatrix} -3 \\ 2 \\ 1 \end{bmatrix} b = \begin{bmatrix} 2 \\ -1 \\ 5 \end{bmatrix} c = \begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix}$$

Tentukan vec[a]+vec[2*b].

```
>$&a:=[-3;2;1]
>$b:=[2;-1;5]
>$& a+2*b
```

< Panjang vektor (norm) >

Misal [u]=(u1,u2)dan [w]= (w1,w2,w3) vector di R^2 dan R^3,maka panjang (norm) [u] dan [w] adalah

```
>$&u:= sqrt (u1^2 + u2^2 +u3^2)
```

Contoh:

$$v = \begin{bmatrix} 11 & 13 & 17 \end{bmatrix}$$

```
>V$&:=[11,13,17]
```

```
    [11,  13,  17]
```

```
>norm([11,13,17])
```

```
    24.062418831
```

```
>$&u= sqrt((11^2)+(13^2)+(17^2))
```

< Hasil kali titik >

Hasil kali titik merupakan operasi antara dua buah vektor yang akan menghasilkan skalar.
Pada software EMT menggunakan perintah dot(u,v) atau u*v

Contoh:

Jika vector u= [4,9,6] dan vektor v =[2,10,7]. Tentukan u*v.

```
>$&u:=[4,9,6]
>$&v:=[2,10,7]
>$&dot(u,v)
>$&(u*v)
```

Lanjutan Pengerjaan Soal-Soal

Mohon maaf, seharusnya bagian ini ada di bagian setelah operasi bentuk aljabar dan sebelum polinomial. Namun, ketika saya menambah baris perintah baru di bagian tersebut, semua hasilnya berubah dari hasil

awal sehingga saya memutuskan untuk melanjutkan pengerjaan soal-soalnya di sini. Sekali lagi mohon maaf Bapak karena tidak urut sesuai perintah yang tertulis di Besmart.

Sederhanakan bentuk aljabar berikut ini.

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

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

Dengan menggunakan EMT, kita mendapatkan hasil dari bentuk aljabar tersebut adalah

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

Hitung hasil dari bentuk operasi matematika berikut.

$$\frac{4(8-6)^2-4\cdot 3+2\cdot 8}{3^1+19^0}$$

```
>((4*(8-6)^2-4*3+2*8)/(3^1+19^0))
```

5

Jika kita kerjakan secara manual akan seperti berikut.

$$\frac{4(8-6)^2-4\cdot 3+2\cdot 8}{3^1+19^0}$$

$$=\frac{4(2)^2-12+16}{3+1}$$

$$=\frac{16-12+16}{4}$$

$$=\frac{20}{4}$$

$$=5$$

Nah, hasilnya juga sama seperti yang dikerjakan oleh EMT.

Kalikan bentuk aljabar berikut. Asumsikan semua eksponen adalah bilangan asli.

$$(a^n+b^n)^2$$

```
>$&expand((a^n+b^n)^2)
```

Nah, hasilnya juga mirip dengan yang sudah kita tahu bahwa (a+b)^2 hasilnya adalah

$$a^2 + 2ab + b^2$$

Faktorkan selisih kuadrat berikut ini.

$$4z^2 - 81$$

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

Jika dihitung secara manual akan seperti berikut.

$$4z^2 - 81 = 0$$

$$4z^2 - 81 = 0$$

$$(2z)^2 - 9^2 = 0$$

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

Tentukan nilai x dari persamaan berikut.

$$9(2x + 8) = 20 - (x + 5)$$

```
>$&solve(9*(2*x+8)=20-(x+5))
```

Jika kita hitung secara manual akan seperti berikut.

$$9(2x + 8) = 20 - (x + 5)$$

$$18x + 72 = 20 - x - 5$$

$$18x + x = 20 - 5 - 72$$

$$19x = -57$$

$$x = -57/3$$

$$x = -3$$

(Sesuai dengan hasil EMT)

Sederhanakan bentuk pecahan aljabar berikut.

$$\frac{y^5-5y^4+4y^3}{y^3-6y^2+8y}$$

```
>$&ratsimp((y^5-5*y^4+4*y^3)/(y^3-6*y^2+8*y))
```

Jadi, bentuk sederhana dari operasi aljabar tersebut adalah

$$\frac{y^3-y^2}{y-2}$$

atau bisa disederhanakan lagi menjadi

$$\frac{y^2(y-1)}{y-2}$$

Tentukan hasil dari operasi berikut.

$$\frac{5}{4z}-\frac{3}{8z}$$

```
>$&expand((5/(4*z))-(3/(8*z)))
```

Oke, jadi

$$\frac{5}{4z}-\frac{3}{8z}$$

$$=\frac{5\cdot 2}{4z\cdot 2}-\frac{3}{8z}$$

$$=\frac{10-3}{8z}$$

$$=\frac{7}{8z}$$

Tentukan hasil pembagian dari bentuk aljabar berikut.

$$\frac{3x+12}{2x-8}\div\frac{(x+4)^2}{(x-4)^2}$$

```
>$&solve(((3*x+12)/(2*x-8))/((x+4)^2/(x-4)^2))
```

Berdasarkan perhitungan EMT, nilai x yang didapat bernilai 4.

Tentukan hasil dari akar berikut.

$$\sqrt[4]{48x^6y^4}$$

```
>$&simplify((48*x^6*y^4)^(1/4))
```

Tentukan masing-masing nilai dari fungsi berikut. Jika diketahui

$$f(x) = 3x + 1$$

$$g(x) = x^2 - 2x - 6$$

(f o g)(-1)

```
>3*( (-1)^2 - 2*(-1) - 6 ) + 1
```

```
-8
```

(g o f)(-2)

```
>(3*(-2)+1)^2 - 2*(3*(-2)+1) - 6
```

```
29
```

Hitung penjumlahan dari bilangan kompleks berikut.

$$(-5 + 3i) + (7 + 8i)$$

```
>(-5+3i)+(7+8i)
```

```
2+11i
```

$$\frac{3}{5 - 11i}$$

```
>3/(5-11i)
```

```
0.10274+0.226027i
```

$$x^2 + x - \sqrt{2} = 0$$

```
>$&solve(x^2+x-sqrt(2)=0)
```