

# Pengantar Komputasi Numerik dan Simbolik dengan Jasyzca

Saifuddin Arief

[Saifuddin.Arief@rocketmail.com](mailto:Saifuddin.Arief@rocketmail.com)

## Lisensi Dokumen:

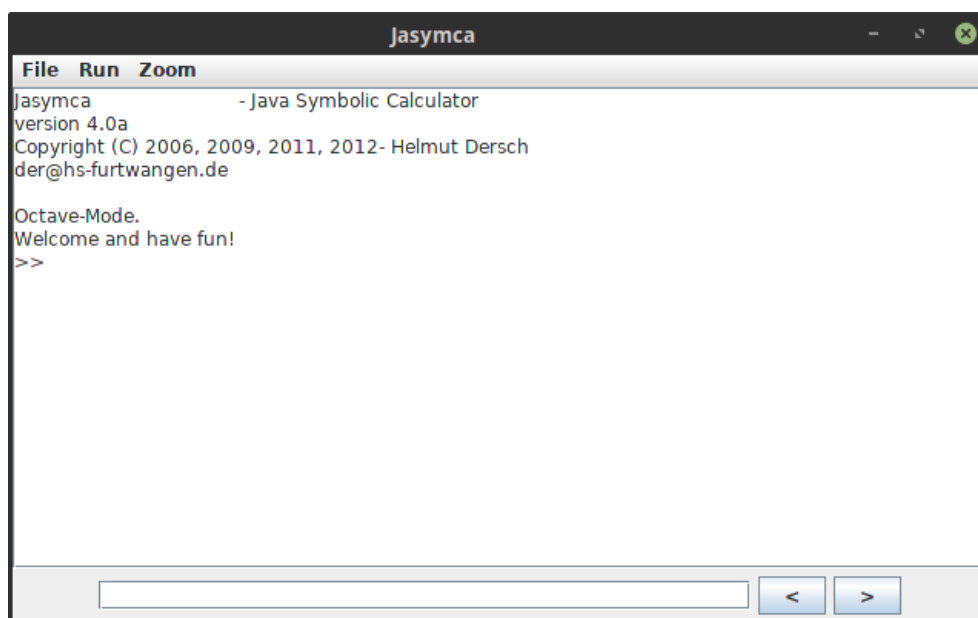
Copyright © 2003-2019 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Jasyzca - Java Symbolic Calculator adalah aplikasi matematika yang dapat digunakan secara interaktif untuk melakukan komputasi numerik dan simbolik. Aplikasi tersebut dikembangkan sebagai alat bantu pengajaran komputasi matematika. Jasyzca dapat digunakan untuk menyelesaikan suatu perhitungan yang sederhana maupun persoalan matematika yang lebih rumit. Fitur-fitur yang disediakan oleh Jasyzca untuk komputasi numerik sudah memadai namun untuk komputasi simbolik kemampuannya relatif terbatas hanya dapat menangani beberapa fungsi dan operasi matematika yang bersifat elementer. Jasyzca dapat digunakan sistem operasi Windows, Linux, MacOS dan Android. File instalasi dan dokumentasi Jasyzca dapat diperoleh pada situs <https://webuser.hs-furtwangen.de/~dersch/jasyzca2/indexEN.html>.

## Interaksi dengan Jasyzca

Setelah Jasyzca dijalankan maka pada layar komputer akan muncul jendela Jasyzca seperti yang terlihat pada Gambar 1.



Gambar 1. Jasyzca

Seperti yang terlihat pada gambar 1, pada bagian atas jendela Jasyima akan ditampilkan beberapa keterangan mengenai versi Jasyimca serta mode yang digunakan oleh Jasyimca. Mode default yang digunakan oleh Jasyimca adalah Octave-Mode. Selain mode tersebut, Jasyima juga dapat dijalankan dengan Maxima-Mode.

Pada artikel hanya akan diberikan penggunaan Jasyimca dengan Octave-Mode. Penjelasan mengenai penggunaan Jasyimca dengan Maxima-Mode dapat dilihat pada dokumentasinya.

Perintah yang diberikan pada Jasyimca dimasukkan lewat papan input yang terletak pada bagian bawah jendelanya. Sebagai contoh, untuk melakukan perhitungan  $1.23 + 4.56$  maka ekspresi matematika tersebut diketikkan pada papan input kemudian ditekan tombol enter untuk menjalankan perintahnya. Perintah yang diberikan dan hasil perhitungannya akan ditampilkan pada jendela Jasyimca sebagai berikut

```
>> 1.23 + 4.56
ans = 5.7900
>>
```

Pada contoh di atas, `ans` adalah nama variabel default yang digunakan oleh Jasyimca untuk menyimpan suatu hasil perhitungan yang tidak disimpan ke dalam sebuah variabel. Kemudian simbol `>>` yang muncul kembali menunjukkan bahwa Jasyimca siap untuk menerima perintah berikutnya.

Apabila diperlukan, perintah `help` dapat digunakan untuk mendapatkan penjelasan mengenai suatu perintah atau fungsi tertentu. Sebagai contoh, penjelasan tentang fungsi `atan` dapat diperoleh dengan menjalankan perintah `help(atan)` atau `help atan`.

```
>> help atan
Usage: y=atan(x)
Arctangent of x in radian.
See also: atand,atan2.
```

Untuk keluar dari Jasyimca dapat dilakukan melalui menu `File - Quit` atau dengan menekan tombol `×` yang terletak pada bagian kanan atas jendela Jasyimca.

## Operator, Fungsi Matematika dan Variabel

Operator untuk perhitungan aritmatika yaitu `+`, `-`, `*`, `/` dan `^`. Simbol tersebut masing-masing melambangkan operasi penjumlahan, pengurangan, perkalian, pembagian dan pemangkatan. Di dalam Jasyimca juga terpasang fungsi-fungsi matematika, seperti `sqrt`, `abs`, `exp`, `sin`, `sind`, `cos`, `cosd` dan lain sebagainya.

```
>> 1010*(sqrt(9.81^2 + 7^2)*((0.07 + 0.0214)*cos(35.5/180*pi)))
ans = 1112.5*cos(0.19722*pi)
>> float(ans)
ans = 905.71
>> 1010*(sqrt(9.81^2 + 7^2)*((0.07 + 0.0214)*cosd(35.5)))
ans = 905.71
```

Variabel `pi` merupakan variabel khusus yang merepresentasikan  $\pi = 3.1415927$ . Jasyimca tidak akan menampilkan nilai aproksimasi dari variabel `pi` secara langsung. Untuk menampilkan aproksimasi desimal dari ekspresi matematika yang di dalamnya terdapat variabel `pi` dapat digunakan fungsi `float`.

Seperti yang diilustrasikan pada contoh di atas, terdapat dua kelompok fungsi trigonometri yang disediakan oleh Jasyimca, yaitu fungsi trigonometri dengan argumen radian dan fungsi trigonometri dengan argumen derajat. Apabila pada akhiran fungsi terdapat huruf `d`, seperti `cosd`, `sind` maka argumennya menggunakan satuan derajat.

Nilai suatu obyek atau hasil perhitungan dapat disimpan ke dalam suatu variabel dengan statemen sebagai berikut:

```
x = ekspresi
```

dimana x adalah nama variabel. Dalam penggunaan variabel, huruf kecil dan huruf besar adalah berbeda. Nilai dari suatu ekspresi akan ditampilkan pada baris berikutnya kecuali jika ditambahkan tanda titik koma (;) pada akhir ekspresinya.

Untuk memperjelas perintah-perintah yang dibuat, dapat ditambahkan suatu baris komentar. Baris komentar dapat ditulis sebagai suatu baris tersendiri atau ditulis di belakang suatu statemen. Kumpulan karakter yang terletak setelah tanda % dianggap sebagai baris komentar.

```
>> luas = 12.5*8
luas = 100
>> r = 10;          % radius lingkaran
>> A = pi*r^2      % luas lingkaran
A = 100*pi
>> float(A)
ans = 314.16
```

Variabel-variabel yang telah dibuat akan disimpan dalam ruang kerja. Untuk melihat nama-nama variabel yang telah dibuat, gunakan perintah who. Variabel yang tersimpan dalam ruang kerja dapat dihapus dengan perintah clear.

```
>> who
A: 100*pi
luas: 100
ans: 314.16
r: 10
>> clear r          % menghapus variabel luas dan tinggi
>> clear            % menghapus semua variabel
```

## Bilangan Kompleks

Bilangan kompleks  $z = x + iy$  dapat dinyatakan dengan  $z = x + yi$  atau  $z = x + yj$ , dimana  $i$  dan  $j$  adalah variabel khusus yang merepresentasikan konstanta  $\sqrt{-1}$ . Operasi aritmatika terhadap bilangan kompleks dapat dilakukan dengan menggunakan notasi yang sama dengan notasi pada bilangan real.

```
>> z1 = 6 - 8i;
>> z2 = 3 + i;
>> p = z1 + z2
p = 9-7i
>> q = z1 - z2
q = 3-9i
>> z1*z2
ans = 26-18i
>> z2/z1
ans = 0.1+0.3i
```

## Matrik dan Vektor

Salah satu kelebihan Jasympca yaitu kemampuannya dalam menangani berbagai macam operasi manipulasi terhadap data yang berupa suatu matrik. Vektor dan skalar merupakan bentuk khusus dari suatu matrik. Vektor adalah suatu matrik yang hanya mempunyai satu baris atau satu kolom saja, sementara itu skalar adalah suatu matrik yang hanya terdiri dari satu elemen saja.

Pembuatan data matrik dan vektor secara manual dilakukan dengan menggunakan operator kurung siku ([ ]). Dimana elemen-elemen matrik atau vektor dimasukkan diantara kedua kurung siku tersebut. Elemen yang lainnya yang terletak pada satu baris dapat dipisahkan dengan tanda koma (,) atau tanda spasi. Kemudian antara baris yang satu dengan yang lainnya dipisahkan dengan tanda titik koma (;) atau tombol Enter.

```
>> X = [1 2 3; 4 5 6; 7 8 9];  
>> v = [1 2 3 4 5];           % vektor baris  
>> w = [3; 5; 2];             % vektor kolom
```

Suatu vektor baris dimana nilai elemen-elemennya berubah secara konstan dari suatu nilai awal sampai nilai akhir tertentu dapat dibuat dengan sintak i:j:k. Apabila nilai j sama satu maka notasi tersebut dapat ditulis dengan notasi yang lebih singkat yaitu i:k.

```
>> i = 1:10  
i = [ 1 2 3 4 5 6 7 8 9 10 ]  
>> n = 0:0.25:1  
n = [ 0 0.25 0.5 0.75 1 ]  
>> m = 1:2:10  
m = [ 1 3 5 7 9 ]
```

Jasymca juga menyediakan sejumlah fungsi yang dapat digunakan untuk membuat matrik-matrik khusus, seperti yang ditunjukkan pada contoh-contoh di bawah ini.

```
>> A = zeros(3,4)           % matrik nol  
A =  
 0 0 0 0  
 0 0 0 0  
 0 0 0 0  
>> B = ones(2,5)           % matrik satuan  
B =  
 1 1 1 1 1  
 1 1 1 1 1  
>> Y = eye(3)              % matrik identitas  
Y =  
 1 0 0  
 0 1 0  
 0 0 1  
>> D = diag(1:4)          % matrik diagonal  
D =  
 1 0 0 0  
 0 2 0 0  
 0 0 3 0  
 0 0 0 4
```

## Operasi Berbasis Vektor

Di dalam Jasymca, secara umum operasi-operasi matematika terhadap obyek matrik dan vektor dapat dilakukan dengan mudah tanpa harus menggunakan suatu perulangan.

```
>> x = float(0:pi/4:pi)  
x = [ 0 0.7854 1.5708 2.3562 3.1416 ]  
>> cos(x)  
ans = [ 1 0.70711 6.1232E-17 -0.70711 -1 ]  
>> u = [1 2 3];  
>> y = exp(u)  
y = [ 2.7183 7.3891 20.086 ]  
>> z = log(y)  
z = [ 1 2 3 ]
```

Operasi Aljabar Linier, seperti penjumlahan, pengurangan dan perkalian, juga dapat dilakukan dengan sangat mudah tanpa harus menggunakan suatu ekspresi perulangan secara eksplisit. Pada operasi aljabar argumen-argumennya harus mempunyai dimensi yang kompatibel, jika dimensinya tidak kompatibel maka operasinya tidak dapat dieksekusi dan akan muncul suatu pesan kesalahan.

```
>> X = [9 8 5; 1 3 0; 2 4 6];
>> Y = [3 2 1; 4 5 6; 9 8 7];
>> A = X + Y
A =
    12    10     6
     5     8     6
    11    12    13
>> B = X - Y
B =
     6     6     4
    -3    -2    -6
    -7    -4    -1
>> p = [1 2; 3 8];
>> z = A + p
aF: Wrong argument type:
     1     2
     3     8
>> j = [6;4];
>> k = ones(3);
>> g = p*j
g =
    14
    50
>> h = p*k
aF: Matrix dimensions wrong.
```

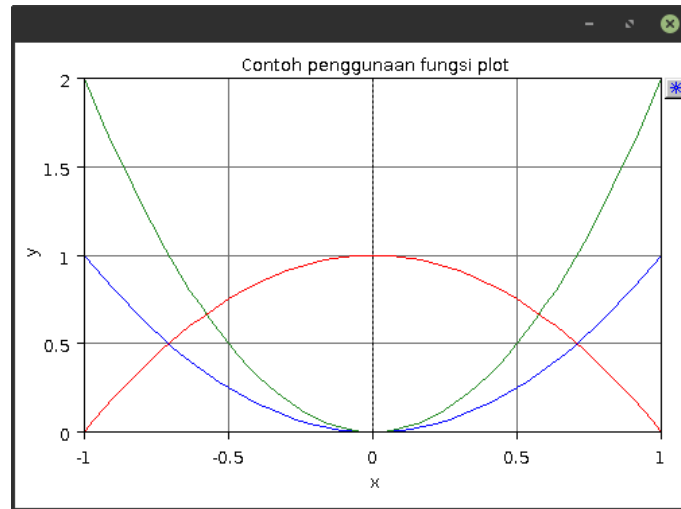
Di dalam Jasymca dapat juga dilakukan operasi perkalian dan pembagian terhadap obyek matrik dan vektor dengan melakukan operasi secara elemen dengan elemen. Notasi untuk perkalian dan pembagian secara elemen dengan elemen adalah `.*` dan `./`.

```
>> x = [1 2; 3 4];
>> y = [5 6; 7 8];
>> u = x.*y
u =
     5    12
    21    32
>> z = y./x
z =
     5     3
    2.3333  2
```

## Visualisasi Data

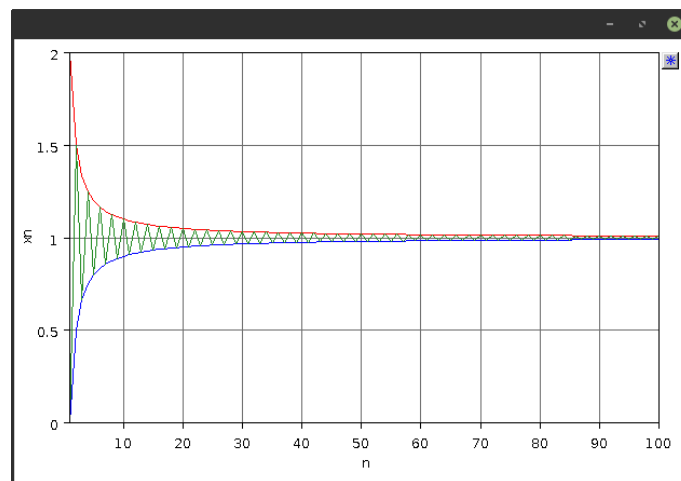
Jasymca dapat digunakan untuk melakukan visualisasi data secara dua dimensi. Grafik yang dihasilkan oleh perintah-perintah grafis akan ditampilkan pada jendela tersendiri.

```
>> % Contoh penggunaan fungsi plot (lihat Gambar 2)
>> x = linspace(-1,1,61);
>> y1 = x.^2; y2 = 1 - y1; y3 = 2*y1;
>> plot(x,y1)
>> hold
Current plot held.
>> plot(x,y2,'r')
>> plot(x,y3,'g')
>> title('Contoh penggunaan fungsi plot')
>> xlabel('x'); ylabel('y')
```



Gambar 2. Contoh penggunaan fungsi plot

```
>> n = 1:100;
>> x1 = (n - 1)./n; x2 = (n + 1)./n; x3 = (n + (-1).^n)./n;
>> plot(n, x1)
>> hold
Current plot held.
>> plot(n, x2, 'r')
>> plot(n, x3, 'g')
>> xlabel("n"); ylabel("xn")
```



Gambar 3. Contoh penggunaan fungsi plot

## Perulangan dan Kondisional

Jasymca menyediakan fitur yang dapat digunakan untuk mengatur jalannya eksekusi suatu program dengan menggunakan statemen perulangan `for`, `while` dan kondisional `if-elseif-else`.

```
>> # Contoh sistem perulangan
>> H = zeros(4,4)
H =
  0  0  0  0
  0  0  0  0
  0  0  0  0
  0  0  0  0
```

```
>> for i=1:4
>   for j=1:4
>     H(i,j) = 1/(i+j-1);
>   end
> end
>> H
H =
    1    0.5    0.33333    0.25
    0.5    0.33333    0.25    0.2
    0.33333    0.25    0.2    0.16667
    0.25    0.2    0.16667    0.14286

>> # Contoh statemen perulangan while
>> a = 12.34; x = a/2; tol = 1e-6;
>> while abs(a - x^2) > tol; x = (x + a/x)/2; end
>> x
x = 3.5128

>> # Contoh statemen perulangan kondisional
>> function y = H(x)
>   if (x >= 0)
>       y = 1;
>   else
>       y = 0;
>   end
> end
>> H(-10)
y = 0
>> H(3)
y = 1
```

## Fungsi

Fungsi merupakan kumpulan dari statemen-statementen Jasymca yang dapat melakukan suatu komputasi atau perhitungan tertentu. Fungsi dapat dibuat dalam bentuk suatu file atau secara inline pada jendela perintah. Bentuk umum dari suatu fungsi adalah sebagai berikut:

```
function out = fname(in1,in2,...)
    statemen-statementen
endfunction
```

dimana *fname* adalah nama fungsi yang dibuat, dan *in1*, *in2* adalah argumen-argumen input serta *out* adalah argumen output.

Suatu file fungsi harus dibuat dengan menggunakan program text editor, seperti Notepad, Notepad2, atau Geany. Nama suatu file fungsi harus sama dengan nama fungsi yang terdapat di dalamnya. Di belakang setiap statemen harus diberi tanda titik-koma (;). Contoh sebuah file fungsi diberikan pada gambar 4.

Agar sebuah file fungsi dapat dibaca Jasymca maka foldernya harus ditambahkan terlebih dahulu melalui menu File - Add Path atau dengan perintah `addpath`.

Contoh pembuatan suatu fungsi secara inline adalah sebagai berikut:

```
>> function y = foo(x)
>   y = x - exp(-x);
> end
```

```

bisect.m - /home/ariefs/Documents - Geany
File Edit Search View Document Project Build Tools Help
bisect.m x
1 function x = bisect(f,x1,x2,tol)
2 % Penentuan akar persamaan nonlinier dengan metode Bisection
3 if f(x1) == 0.0; x = x1; return; end;
4 if f(x2) == 0.0; x = x2; return; end;
5 if f(x1)*f(x2) > 0; error('Akar tidak terdapat di dalam selang [x1,x2]'); end;
6 n = ceil(log(abs(x2 - x1)/tol)/log(2.0)); % Iterasi maksimum
7 for k = 1:n
8     x3 = 0.5*(x1 + x2);
9     if f(x3) == 0.0; x=x3; return; end;
10    if f(x1)*f(x3) < 0.0
11        x1 = x3;
12    else
13        x2 = x3;
14    end;
15 end;
16 x = (x1 + x2)/2;
17 end;
line: 17 / 17 col: 4 sel: 0 INS TAB mode: LF encoding: UTF-8 filetype: Matlab/Octave scope: bisect(f,x1,x2,tol)
    
```

Gambar 4. Contoh sebuah file fungsi

Fungsi yang telah dibuat dapat digunakan sebagaimana fungsi yang telah terpasang pada Jasmca. Berikut ini adalah contoh penggunaan fungsi `bisect` dan `f` yang telah dibuat.

```

>> foo(1)
ans = 0.63212
>> xsol = bisect(@foo, 0,1,1e-6)
xsol = 0.56714
fx0 = -5.1246e-007
    
```

Sayangnya, fasilitas pemrograman pada Jasmca masih belum sempurna. Oleh karena itu bagi yang tertarik dengan pemrograman sebaiknya menggunakan program GNU Octave atau Scilab.

## Komputasi Numerik

Di dalam Jasmca, berbagai macam persoalan komputasi numerik dapat diselesaikan dengan mudah, seperti yang ditunjukkan pada contoh-contoh di bawah ini.

Sistem persamaan linear  $Ax = b$  dapat diselesaikan dengan mudah menggunakan operator pembagian kiri (`\`) atau dengan fungsi `linsolve(A,b)`.

```

>> A = [1 1 -1; 6 0 2; 6 -4 0]
A =
     1     1    -1
     6     0     2
     6    -4     0
>> b = [0; 10; 24]
b =
     0
    10
    24
>> x = A\b
x =
     2
    -3
    -1
>> linsolve(A,b)
ans = [ 2 -3 -1 ]
    
```



Nilai determinan dan inverse matrik dapat dihitung dengan menggunakan fungsi `det` dan `inv`.

```
>> det(A)
ans = 44
>> inv(A)
ans =
    0.18182    9.0909E-2    4.5455E-2
    0.27273    0.13636   -0.18182
   -0.54545    0.22727   -0.13636
```

Penyelesaian suatu persamaan nonlinier  $f(x) = 0$  dapat dilakukan dengan fungsi `fzero`, seperti yang ditunjukkan pada contoh di bawah ini.

```
>> fzero('x - exp(-x)',0,1)
x = 0.56714
```

Diperoleh bahwa akar dari  $f(x) = x - \exp(-x) = 0$ , adalah  $x = 0.5671$ .

Terdapat banyak persamaan integral tertentu yang sulit atau bahkan tidak dapat diselesaikan secara analitis, salah satunya adalah seperti yang terdapat pada contoh berikut ini.

$$Q = \int_0^5 \frac{\sin(3x)}{\sqrt{x^2 + x + 1}} dx$$

Nilai suatu integral tertentu dapat dihitung dengan menggunakan fungsi `quad`. Nilai integral  $Q$  dapat dihitung dengan perintah sebagai berikut.

```
>> Q = quad('sin(3*x)./sqrt(x.^2 + x + 1)',0,5)
Q = 0.36487
```

## Komputasi Simbolik

Jasymca juga dapat melakukan komputasi secara simbolik meskipun hanya terdapat pada beberapa fungsi dan operasi yang bersifat elementer. Komputasi simbolik yang dijelaskan pada bagian disini menggunakan Octave-Mode. Dalam komputasi simbolik, variabel yang akan digunakan secara simbolik harus dinyatakan terlebih dahulu dengan perintah `syms`.

Operasi simbolik untuk fungsi polinomial telah diimplementasikan dengan baik dalam Jasymca, seperti yang ditunjukkan pada contoh di bawah ini.

```
>> syms x
>> x + 2*x + 5
ans = 3*x+5
>> (x + 2)^2
ans = x^2+4*x+4
>> y = (x + 2)^2
y = x^2+4*x+4
>> allroots(y)
ans = [ -2 -2 ]
>> a = [2 -2 6 -6];
>> p = polyval(a,x)
p = 2*x^3-2*x^2+6*x-6
>> allroots(p)
ans = [ 1 -1.7321i 1.7321i ]
```

Mengacu pada contoh-contoh di atas, terlihat pada pembuatan dan operasi matematika pada fungsi polinomial dapat dilakukan dengan mudah. Pada contoh ini, fungsi `polyval(a,x)` digunakan mengevaluasi suatu polinomial dengan koefisien  $a$  pada  $x$ . Kemudian fungsi `aallroots(p)` digunakan untuk mencari akar dari fungsi polinomial  $p$ .

Penyelesaian sebuah persamaan nonlinear dapat dilakukan dengan menggunakan fungsi `solve`.

```
>> syms x
>> solve(x^2 - 13*x + 10, x)
ans = [ 1/2*sqrt(129)+13/2 -1/2*sqrt(129)+13/2 ]
>> solve(x^2 + 10, x)
ans = [ 1*i*sqrt(10) -1*i*sqrt(10) ]
```

Harap diperhatikan bahwa fungsi `solve` kemampuan terbatas, pada sejumlah kasus outputnya hanya bersifat parsial atau bahkan tidak dapat memberikan penyelesaian, seperti pada contoh berikut ini.

```
>> solve(x - exp(-x), x)
aF: Can not solve equation.
```

Jasymca juga melakukan operasi komputasi simbolik untuk kalkulus meskipun terbatas pada fungsi-fungsi elementer saja.

```
>> syms x
>> y = integrate(x^2 + x - 3, x)
y = 0.33333*x^3+0.5*x^2-3*x
>> diff(y)
ans = x^2+x-3
>> dy = diff(tan(x),x)
dy = 1/cos(x)^2
>> z = integrate(1/(1 + x^2), x)
z = atan(x)
>> diff(z, x)
ans = 1/(x^2+1)
```

Untuk komputasi integral tertentu dari suatu fungsi juga dapat dilakukan dengan menyatakan fungsinya secara simbolik kemudian menghitungnya dengan fungsi `romberg`.

```
>> syms x
>> Q = romberg(sin(3*x)/sqrt(x^2 + x + 1),x,0,5)
Q = 0.36487
```

Terlihat bahwa nilai  $Q = 0.36487$ , sama seperti yang diperoleh pada perhitungan dengan fungsi `quad`.

Secara umum, komputasi simbolik yang terdapat dalam Jasymca masih sangat terbatas. Bagi yang tertarik dengan komputasi simbolik dapat menggunakan freeware computer algebra system yang lebih powerful seperti Maxima, Reduce, Xcas dan SymPy.

## Daftar Pustaka

Dersch, H. 2009. Jasymca 2.0 - Symbolic Calculator for Java