

PRAKTIKUM 10

ANIMASI

- 1. MINGGU KE : 12
- 2. PERALATAN : LCD, E-LEARNING
- 3. SOFTWARE : MAPLE
- 4. TUJUAN

Mahasiswa dapat memahami animasi dua dimensi dan tiga dimensi dalam Maple.

5. TEORI PENGANTAR DAN LANGKAH KERJA

Maple mempunyai kemampuan untuk membangun animasi dalam dua dimensi atau tiga dimensi.

PLOT(ANIMATE(.....))

Atau

PLOT3D(ANIMATE(...))

Di dalam fungsi ANIMATE adalah rangkaian frame. Perhatikan contoh berikut:

```
> lprint(plots[animate](x*t, x=-
1..1, t=1..3, numpoints=3, frames=3));
PLOT(ANIMATE([CURVES([[ -1., -1.], [0., 0.], [1.000000000,
1.]], COLOUR(RGB,1.00000000,0.,0.))], [CURVES([[ -1., -2.], [0., 0.],
[1.000000000, 2.]], COLOUR(RGB,1.00000000,0.,0.))], [CURVES([[ -1., -3.],
[0., 0.], [1.000000000,
3.]], COLOUR(RGB,1.00000000,0.,0.))]), AXESLABELS(x, ``), VIEW(-1. ..
1., DEFAULT))
```

Titik-titik fungsi berikut adalah bentuk parametrik dari *kurva* $(x, y) = (1 + \cos(t\pi/180))^2, 1 + \cos(t\pi/180)\sin(t\pi/180))$.

```
> points:=t->evalf(
>   [(1+cos(t/180*Pi))*cos(t/180*Pi),
>   (1+cos(t/180*Pi))*sin(t/180*Pi)]:
```

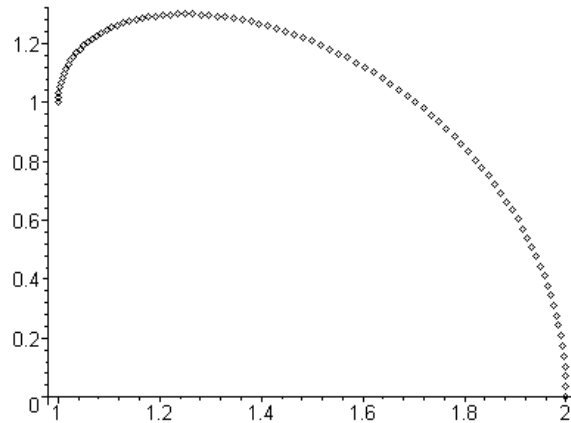
Misalnya,

```
> points(2);
```

```
[1.998172852 .0697777335]
```

Barisan titik-titik dapat diplot.

```
> PLOT (POINTS (seq(points(t), t=0..90))) ;
```



Selanjutnya membuat animasi. Buat setiap frame yang terdiri dari poligon yang diputar di pusat, (0, 0), dan barisan titik di kurva.

```
> frame:=n->[POLYGONS ([[0,0], seq(points(t), t=0..60*n)],  
>   COLOR(RGB, 1.0/n, 1.0/n, 1.0/n))]:
```

Animasi terdiri dari enam frame.

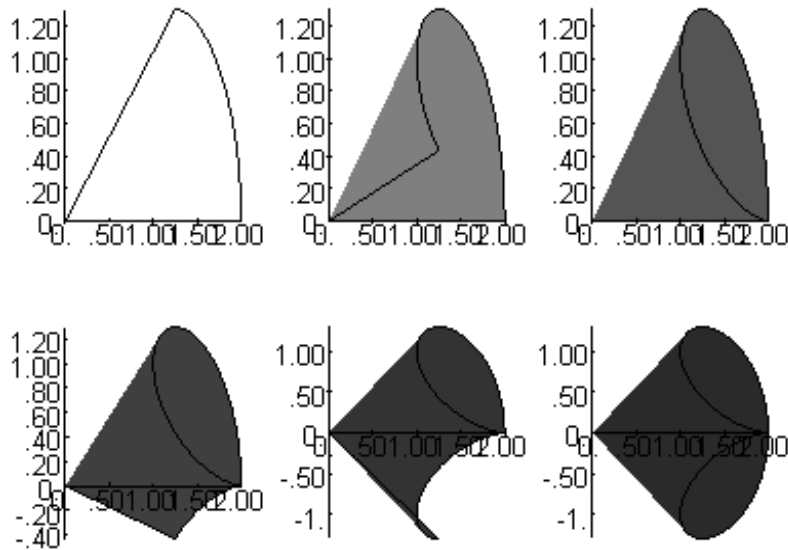
```
> PLOT (ANIMATE (seq(frame(n), n=1..6))) ;
```

Perintah display dari paket plot dapat menampilkan animasi bentuk statis.

```
> with(plots):
```

Warning, the name changecoords has been redefined

```
> display(PLOT (ANIMATE (seq(frame(n), n=1..6)))) ;
```



Dengan prosedur varyAspect berikut memperlihatkan berbagai permukaan dengan aspek rasio.

```
> with(plottools):
```

Warning, the name arrow has been redefined

```
> varyAspect:=proc(p)
```

```
>   local n,opts;
```

```
>   opts:=convert([args[2..nargs]],PLOT3Doptions);
```

```
>   PLOT3D(ANIMATE(seq([stellate(p,n/sqrt(2))],n=1..4)),
```

```
>     op(opts));
```

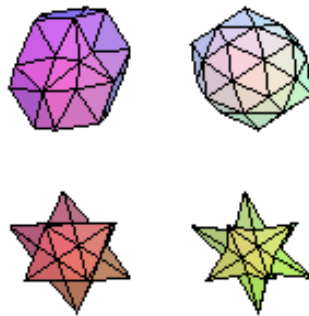
```
> end proc;
```

Cobakan prosedur ini pada dodecahedron.

```
> varyAspect(dodecahedron(),scaling=constrained);
```



```
> display(varyAspect(dodecahedron(), scaling=constrained));
```



Objek grafis dari paket `plottools` dapat dikombinasikan dengan `display` dari rangkaian pilihan animasi objek fisika. Prosedur `springPlot` membuat per dari plot tiga dimensi helix, membuat kotak dan mengkopir kotak ini dan menggerakkan satu dari kotak ke berbagai lokasi tergantung nilai u . Akhirnya membuat bola dan memutarnya ke lokasi di atas kotak dengan ketinggian berbagai parameter.

```
> springPlot := proc(n)
>   local u, curve, springs, box, tops, bottoms,
>     helix, ball, balls;
>   curve:=(u,v)->spacecurve(
>     [cos(t), sin(t), 8*sin(u/v*Pi)*t/200],
>     t=0..20*Pi,
```

```

> color=black, numpoints=200, thickness=3):
> springs := display( [seq(curve(u,n), u=1..n) ],
> insequence=true ):
> box := cuboid( [-1,-1,0], [1,1,1], color=red ):
> ball := sphere( [0,0,2], grid=[15, 15], color=blue ):
> tops := display( [seq(
> translate( box, 0, 0, sin(u/n*Pi)*4*Pi/5),
> u=1..n) ], insequence=true ):
> bottoms := display( [ seq( translate(box, 0, 0, -1),
> u=1..n) ], insequence=true ):
> balls := display( [seq(translate( ball, 0, 0,
> 4*sin( (u-1)/(n-1)*Pi )+ 8*sin(u/n*Pi)*Pi/10),
> u=1..n) ], insequence=true ):
> display( springs, tops, bottoms, balls,
> style=patch, orientation=[45,76],
> scaling=constrained );
> end proc:
> with(plots): with(plottools):
Warning, the name arrow has been redefined

Warning, the name arrow has been redefined

> springPlot(6);

```



```

> display( springPlot(6) );

```

