**Laporan Tugas Besar 1**

**Dasar Pemodelan dan Simulasi**



Disusun Oleh:

1. Pratama Yoga S. - 1301170073
2. Darmawan Putra N. - 1301174154
3. Tri Jaka Pamungkas - 1301170306

Kelas IF-40-07

**FAKULTAS TEKNIK INFORMATIKA**

**TELKOM UNIVERSITY**

**BANDUNG**

**2019**

Bola Pantul

Bola pantul merupakan praktik dari teori tumbukan, dimana benda akan jatuh, memantul di permukaan, kemudian kembali melayang di udara.

Pada keadaan awal, benda akan dijatuhkan dari ketinggian tertentu, yang mana kemudian benda tersebut akan memantul kembali ke udara. Pada saat bola berada di permukaan, akan terjadi tumbukan antara benda dengan permukaan yang menyebabkan benda tersebut terpantul kembali ke atas. Namun pada saat benda berada di permukaan, akan terjadi tumbukan lenting sebagian yang diakibatkan oleh koefisien restitusi benda tersebut. Sehingga menyebabkan pantulan yang tidak sempurna dan terjadinya perlambatan laju gerak vertikal dari benda tersebut. Selanjutnya bola akan kembali terpantul ke atas dengan mencapai ketinggian tertentu dan jatuh kembali sehingga akan terjadi pantulan berikutnya.

Dari teori diatas, dapat disimpulkan bahwa teori yang akan digunakan yaitu teori gerak vertikal dan teori tumbukan. Rumus-rumus yang digunakan diantaranya:

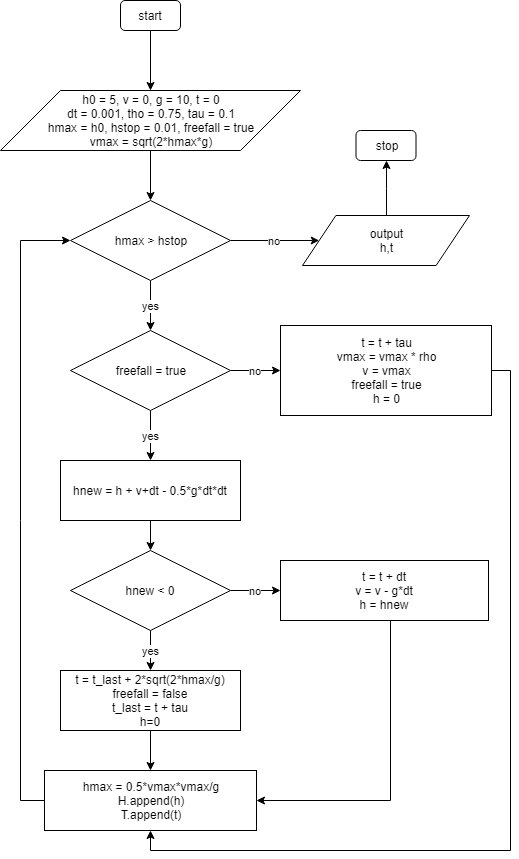
* Rumus gerak vertikal
* Rumus lama waktu benda melayang
* Rumus kecepatan setelah tumbukan

Asumsi:

* Ketinggian awal () = 5m
* Koefisien restitusi (rho / ) = 0.75
* Lama benturan antara benda dan permukaan (tau) = 0.1s

Penghitungan akan berhenti bilamana ketinggian benda berada kurang dari 1cm diatas permukaan.

Flowchart:



Code:

|  |
| --- |
| h0 = 5 # m v = 0 # m/s, current velocity g = 10 # m/s/s t = 0 # starting time dt = 0.001 # time step rho = 0.75 # coefficient of restitution tau = 0.10 # contact time for bounce hmax = h0 # keep track of the maximum height h = h0 hstop = 0.01 # stop when bounce is less than 1 cm freefall = True # state: freefall or in contact  t\_last = -sqrt(2\*h0/g) # time we would have launched to get to h0 at t=0 vmax = sqrt(2 \* hmax \* g) # [ FREEFALL ] H = [] T = []  while(hmax > hstop):  if(freefall):  hnew = h + v\*dt - 0.5\*g\*dt\*dt #H after bounce  if(hnew<0):  t = t\_last + 2\*sqrt(2\*hmax/g) # t every bounce [PARABOLA {freefall} ]  freefall = False  t\_last = t + tau  h = 0  else:  t = t + dt  v = v - g\*dt #V every move [ GLBB ]  h = hnew  else:  t = t + tau  vmax = vmax \* rho  v = vmax  freefall = True  h = 0  hmax = 0.5\*vmax\*vmax/g #hmax [ PARABOLA ] tetha = 90   H.append(h)  T.append(t)  print("stopped bouncing at t=%.3f\n"%(t))  plt.figure() plt.plot(T, H) plt.xlabel('x') plt.ylabel('y') plt.title('bouncing ball') |
|  |