SK500x PENGANTAR SAINS KOMPUTASI

Catatan Kuliah
Menggunakan R

Ikang FADHLI
ikanx101.com

30 August 2022

Contents

List of Figures

Pertemuan I

1 *Unconstrained Growth and Decay*

Pada pembahasan kali ini kita akan membahas model dimana *rate* perubahan itu proporsional dengan kondisi sekarang. Sebelumnya, mari kita bahas konsep dasar tentang **perubahan** sebagai berikut:

1.1 Konsep Dasar

1.1.1 *Rate of Change*

Misalkan dalam suatu gerak benda, y menandakan posisi dinotasikan sebagai fungsi (s) terhadap waktu (t).

Misalkan suatu mobil bergerak dari $t = 0$ h pada $s(0) = 0$ km sampai $t = 10$ h, $s(10) = 70$ km.

Rata-rata kecepatan bisa didefinisikan sebagai perubahan posisi (Δs) terhadap perubahan waktu (Δt). Pada kasus di atas:

$$\text{average velocity} = \frac{\Delta s}{\Delta t} = \frac{70-0}{10-0} = 7\frac{km}{h}$$

Jika kita memperkecil perubahan Δ, kita akan dapatkan ***instantaneous rate of change***.

1.1.2 *Instantaneous Rate of Change*

Dari sini baru muncul yang namanya *derivative* atau turunan.

1.2 *Differential Equation*

Kita diperkenalkan kepada **Malthusian Model** untuk pertumbuhan penduduk, yakni:

$$\frac{dP}{dt} \sim P$$

$$\frac{dP}{dt} = rP, r \text{ is growth rate}$$

1.2.1 Contoh Kasus

Misalkan pada $t = 0$, populasi bakteri ada sebanyak 100 dengan *instantaneous growth rate* sebesar 10% di mana unit waktu yang digunakan adalah jam. Kita bisa tuliskan:

$$\frac{dP}{dt} = 0.1P, P(0) = 100$$

1.3 *Finite Difference Equation*

Komputer tidak bisa menyelesaikan masalah kontinu, oleh karena itu dibutuhkan pendekatan diskrit. Maka untuk kasus model di atas, kita bisa membuat persamaan beda agar kita bisa menghitungnya.

Bentuk umumnya adalah sebagai berikut:

$$\text{new value} = \text{old values} + \text{change in value}$$

1.3.1 Algoritma

```
initialize
  sim_length
  population
  rate
  dt
compute:
  rate_per_step = rate * dt
  num_iter = sim_length / dt
for i: 1 to num_iter
  population = population + rate_per_step * population
  t = i * dt
  print(t, population)
```

1.3.2 Penyelesaian Contoh Kasus

Kita akan selesaikan contoh kasus dengan algoritma yang ada pada bagian sebelumnya.

```
# define
sim_length = 1
population = 100
rate = 0.1
dt = .05
```

```
# compute
rate_per_step = rate * dt
num_iter = sim_length / dt

for(i in 1:num_iter){
  population = population + rate_per_step * population
  t = i*dt
  print(paste("Populasi = ",population," pada t = ",t))
}
```

```
## [1] "Populasi =  100.5  pada t =  0.05"
## [1] "Populasi =  101.0025  pada t =  0.1"
## [1] "Populasi =  101.5075125  pada t =  0.15"
## [1] "Populasi =  102.0150500625  pada t =  0.2"
## [1] "Populasi =  102.525125312813  pada t =  0.25"
## [1] "Populasi =  103.037750939377  pada t =  0.3"
## [1] "Populasi =  103.552939694073  pada t =  0.35"
## [1] "Populasi =  104.070704392544  pada t =  0.4"
## [1] "Populasi =  104.591057914507  pada t =  0.45"
## [1] "Populasi =  105.114013204079  pada t =  0.5"
## [1] "Populasi =  105.639583270099  pada t =  0.55"
## [1] "Populasi =  106.16778118645  pada t =  0.6"
## [1] "Populasi =  106.698620092382  pada t =  0.65"
## [1] "Populasi =  107.232113192844  pada t =  0.7"
## [1] "Populasi =  107.768273758808  pada t =  0.75"
## [1] "Populasi =  108.307115127602  pada t =  0.8"
## [1] "Populasi =  108.84865070324  pada t =  0.85"
## [1] "Populasi =  109.392893956757  pada t =  0.9"
## [1] "Populasi =  109.93985842654  pada t =  0.95"
## [1] "Populasi =  110.489557718673  pada t =  1"
```

Berikut adalah hasil simulasi untuk t yang lebih panjang lagi.

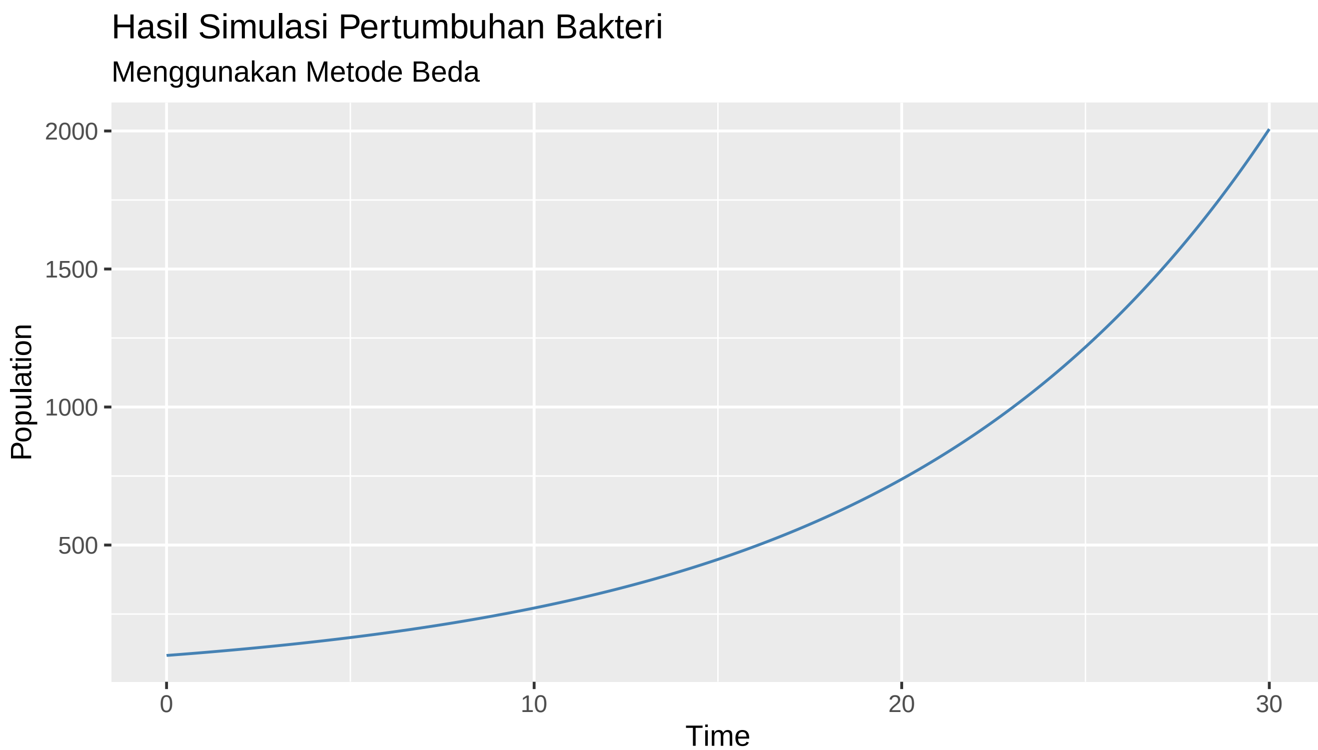

Figure 1: Hasil Simulasi Pertumbuhan Bakteri