PR III Pengantar Sains Komputasi

20921004 - Mohammad Rizka Fadhli

10 November 2022

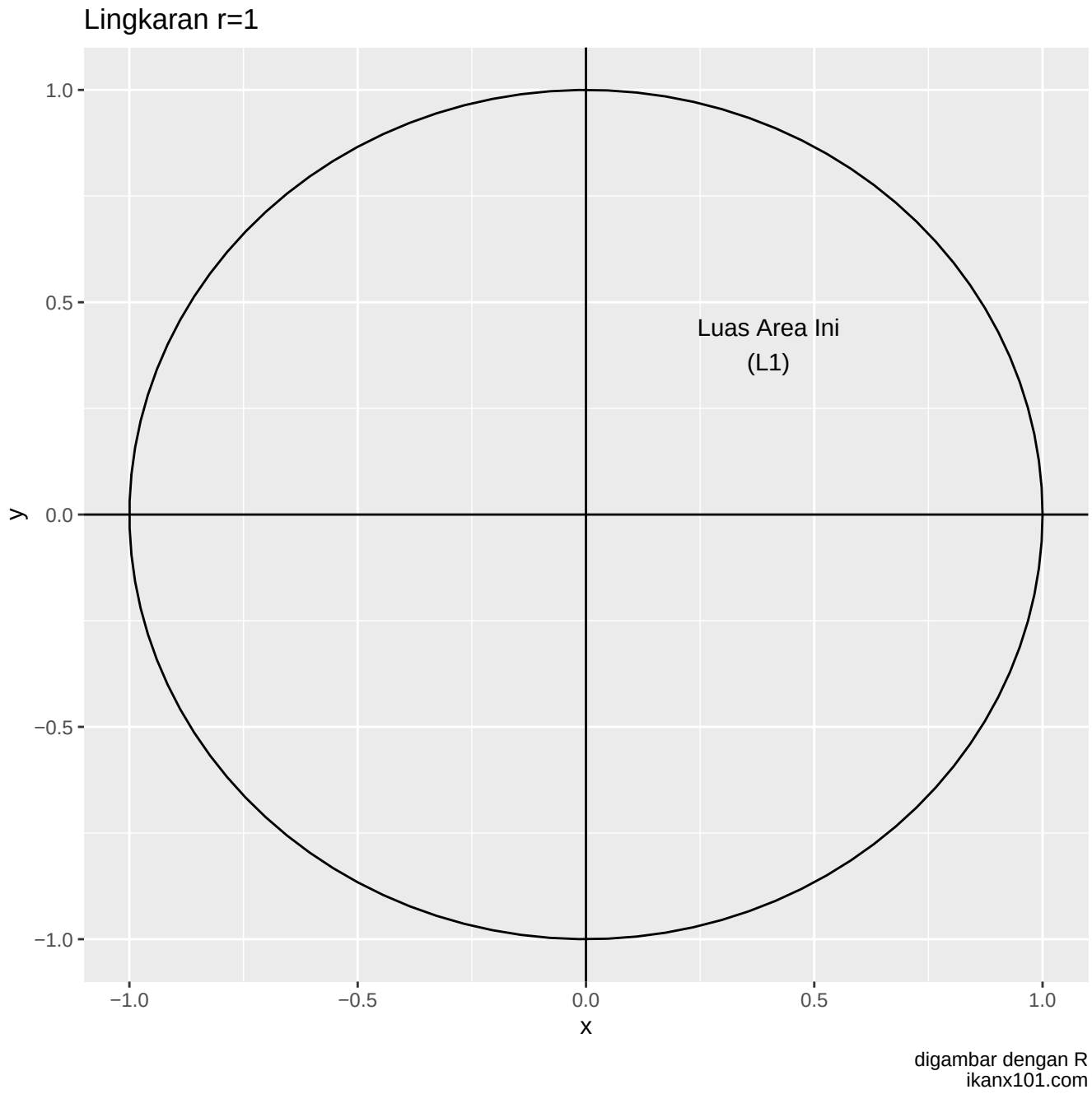
Cara Menghitung π

Bagi matematikawan, ada banyak cara menghitung nilai pi. Ada yang cara deterministik dan ada cara probabilistik.

Kali ini saya akan mencoba menghitung nilai π dengan cara kedua yakni dengan pendekatan simulasi MonteCarlo. Bagaimana cara kerjanya? *Yuk* perhatikan dengan seksama.

Lingkaran dengan $r = 1$

Saya mulai dari lingkaran dengan $r = 1$ berikut ini:



Dari gambar di atas, luas area pada x di range $[0, 1]$ saya tuliskan sebagai berikut:

$$L1 = \frac{1}{4} * L_{lingkaran} = \frac{1}{4} * \pi * r^2$$

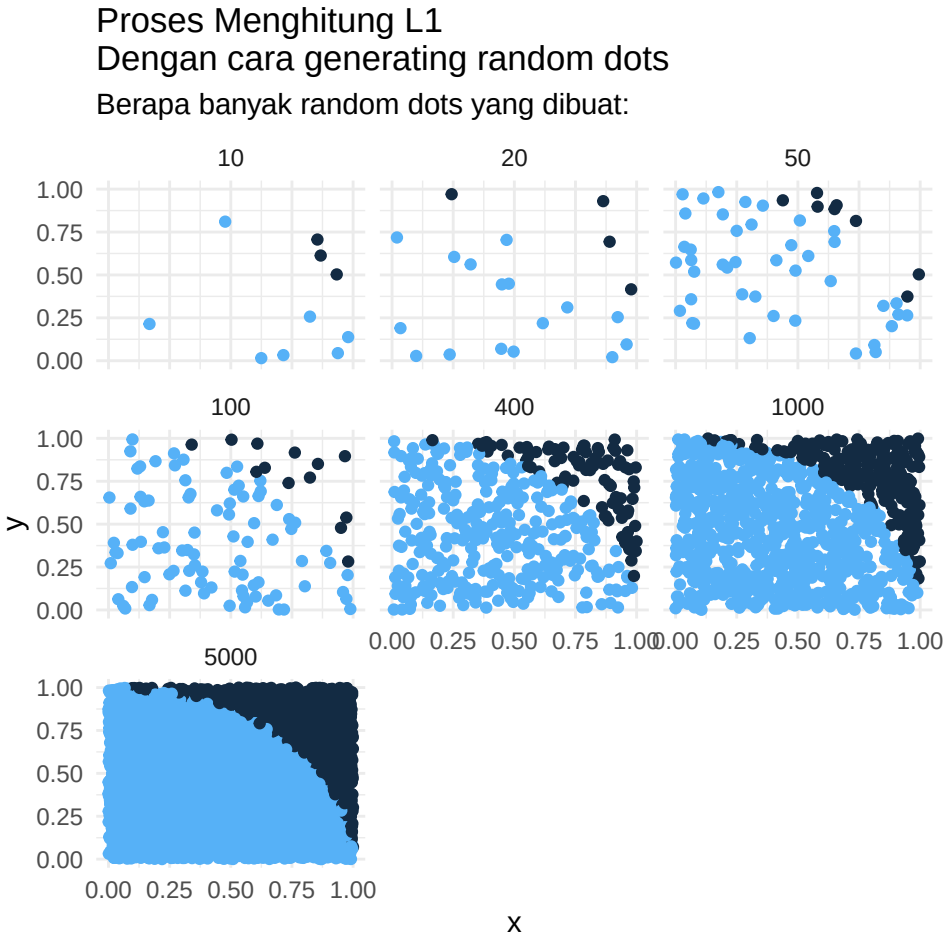
$$L1 = \frac{1}{4} * \pi * (1^2) = \frac{1}{4} * \pi$$

Mencari Nilai $L1$

Kunci untuk mencari nilai π adalah dengan menghitung $L1$.

> Bagaimana menghitung $L1$? Untuk menghitungnya saya akan gunakan metode yang tidak biasa, yakni dengan melakukan *generating random dots* di area $x \in [0, 1]$ dan $y \in [0, 1]$. Setiap titik yang memenuhi persyaratan $x^2 + y^2 \leq 1$ akan saya tandai sebagai `inner` dan diluar itu akan saya tandai sebagai `outer`.

Perhatikan grafik di bawah ini:



Jika dilihat dari grafik di atas, semakin banyak *dots* yang saya buat, semakin banyak area $L1$ yang ter-*cover*. Akibatnya semakin akurat saya menghitung $L1$.

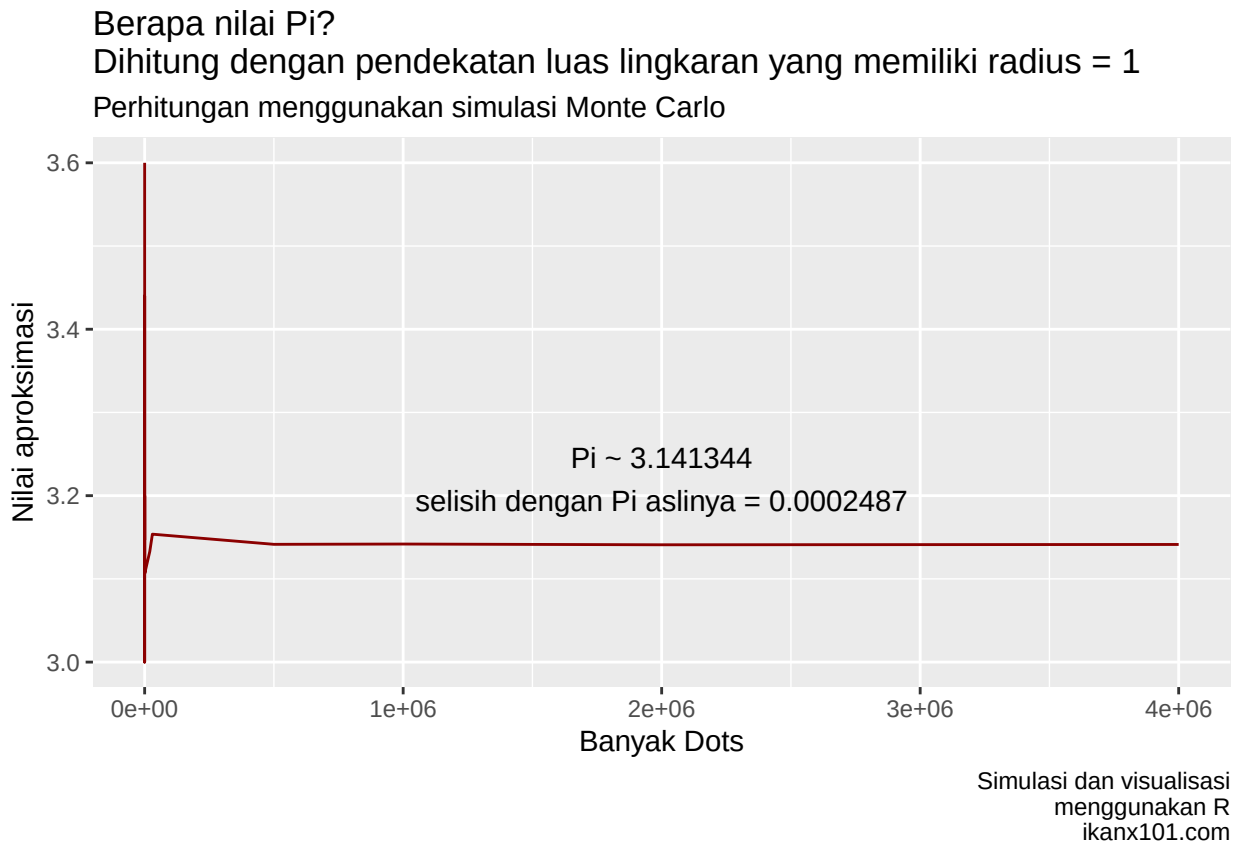
Luas $L1$ dapat saya tuliskan sebagai:

$$L1 \approx \frac{count(dots_{inner})}{count(all.dots)}$$

Lalu: $\pi = 4 * L1$

Berikut algoritma dan hasil perhitungan saya:

```
hitung_pi = function(n){
  x = runif(n)
  y = runif(n)
  data = data.frame(x,y)
  data =
    data %>%
    distinct() %>%
    mutate(jatuh = x^2 + y^2,
           ket = ifelse(jatuh <= 1, 1,0))
  return(4 * sum(data$ket)/length(data$ket))
}
```



Ternyata hasil perhitungan saya dengan membuat 4 juta *dots* lebih akurat dibandingkan pendekatan $\frac{22}{7}$.