



CODELABS
BUILD SOCIETY WITH TECHNOLOGY

ANALISIS ALGORITMA

Notasi Asimptotik

Disusun Oleh:

Adam Mukharil Bachtiar

Teknik Informatika UNIKOM

adfbipotter@gmail.com



AGENDA PERKULIAHAN

- ➡ Penjelasan Order of Growth
- ➡ Notasi Big Oh
- ➡ Notasi Big Theta
- ➡ Notasi Big Omega



Penjelasan Order of Growth

LATAR BELAKANG

“Dalam prakteknya, kompleksitas $T(n)$ secara detil bukanlah menjadi hal yang sangat penting. **Variabel n** yang akan menjadi penentu laju pertumbuhan.”

PEMBUKTIAN

Contoh:

$$T_1(n) = n ; T_2(n) = 2n ; T_3(n) = n^2$$

Perhitungan Laju Pertumbuhan:

n	n	2n	n ²
10	10	20	100
100	100	200	10000
1000	1000	2000	1000000

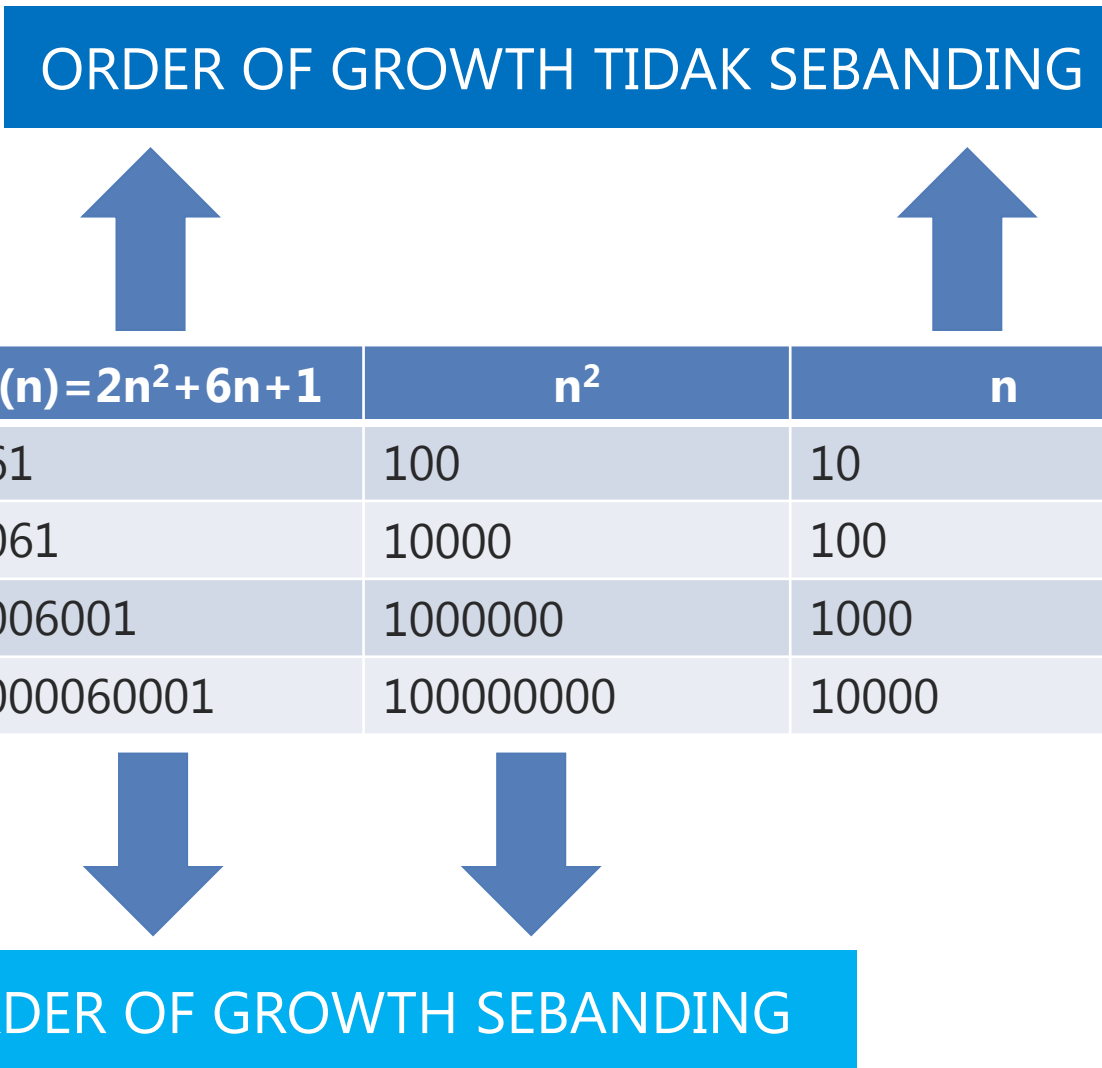
Terlihat bahwa pada $T_1(n)$ dan $T_2(n)$ memiliki laju pertumbuhan yang sebanding (sama) sedangkan untuk $T_3(n)$ memiliki laju pertumbuhan yang lebih tinggi dibanding $T_1(n)$ dan $T_2(n)$.

DEFINISI ORDER OF GROWTH

"Istilah yang dapat digunakan untuk pola **varian jumlah input** dalam suatu pengujian algoritma."

CONTOH ORDER OF GROWTH

ORDER OF GROWTH TIDAK SEBANDING



n	$t(n)=2n^2+6n+1$	n^2	n
10	261	100	10
100	2061	10000	100
1000	2006001	1000000	1000
10000	2000060001	100000000	10000

ORDER OF GROWTH SEBANDING

KESIMPULAN ORDER OF GROWTH

" $t(n) = 2n^2 + 6n + 1$ memiliki orde n^2 karena order of growthnya sebanding dengan $T(n) = n^2$. Oleh karena itu dibutuhkan **pengelompokan kompleksitas algoritma** (agar kompleksitas tidak terlalu tersebar dan spesifik)."

NOTASI PENGELOMPOKAN ALGORITMA

$t(n)$ dan $g(n)$:

Setiap fungsi non negatif dan didefinisikan pada sekumpulan angka alami.

$t(n)$:

Kompleksitas running time suatu algoritma.

$g(n)$:

Beberapa fungsi sederhana untuk membandingkan perhitungan dengan $t(n)$.

Notasi Big Oh (\mathcal{O})

PENJELASAN BIG OH (O)

Definisi $O(g(n))$:

Kumpulan semua fungsi yang order of growth-nya **lebih kecil atau sama dengan** $g(n)$.

Contoh:

$$\Rightarrow n \in O(n^2); 100n + 5 \in O(n^2)$$

$$\Rightarrow \frac{1}{2} n (n-1) \in O(n^2)$$

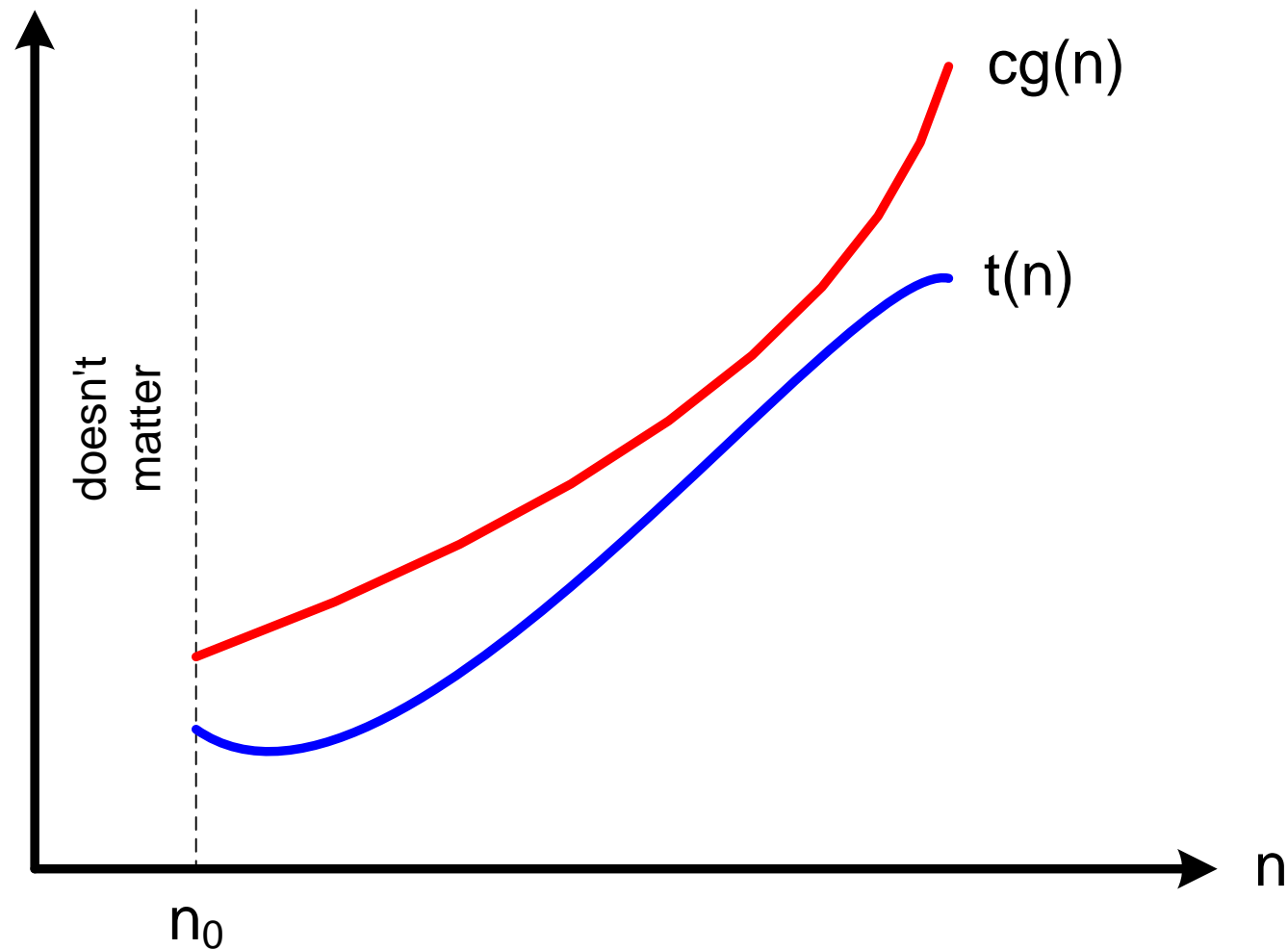
$$\Rightarrow n^3 \notin O(n^2); 0.0001n^3 \notin O(n^2); n^4 + n + 1 \notin O(n^2)$$

DEFINISI BIG OH (O) \rightarrow FORMAL

"Fungsi $t(n)$ dikatakan berada di dalam $O(g(n))$, dinyatakan dengan $t(n) \in O(g(n))$, apabila $t(n)$ dibatasi atas oleh beberapa konstanta dikali $g(n)$ untuk seluruh n , jika ada beberapa konstanta c positif dan beberapa bilangan bulat tak negatif N_0 sedemikian sehingga:

$$t(n) \leq cg(n) \text{ untuk semua } n \geq n_0"$$

ILUSTRASI $t(n) \in O(g(n))$



PEMBUKTIAN BIG OH (O) CARA 1

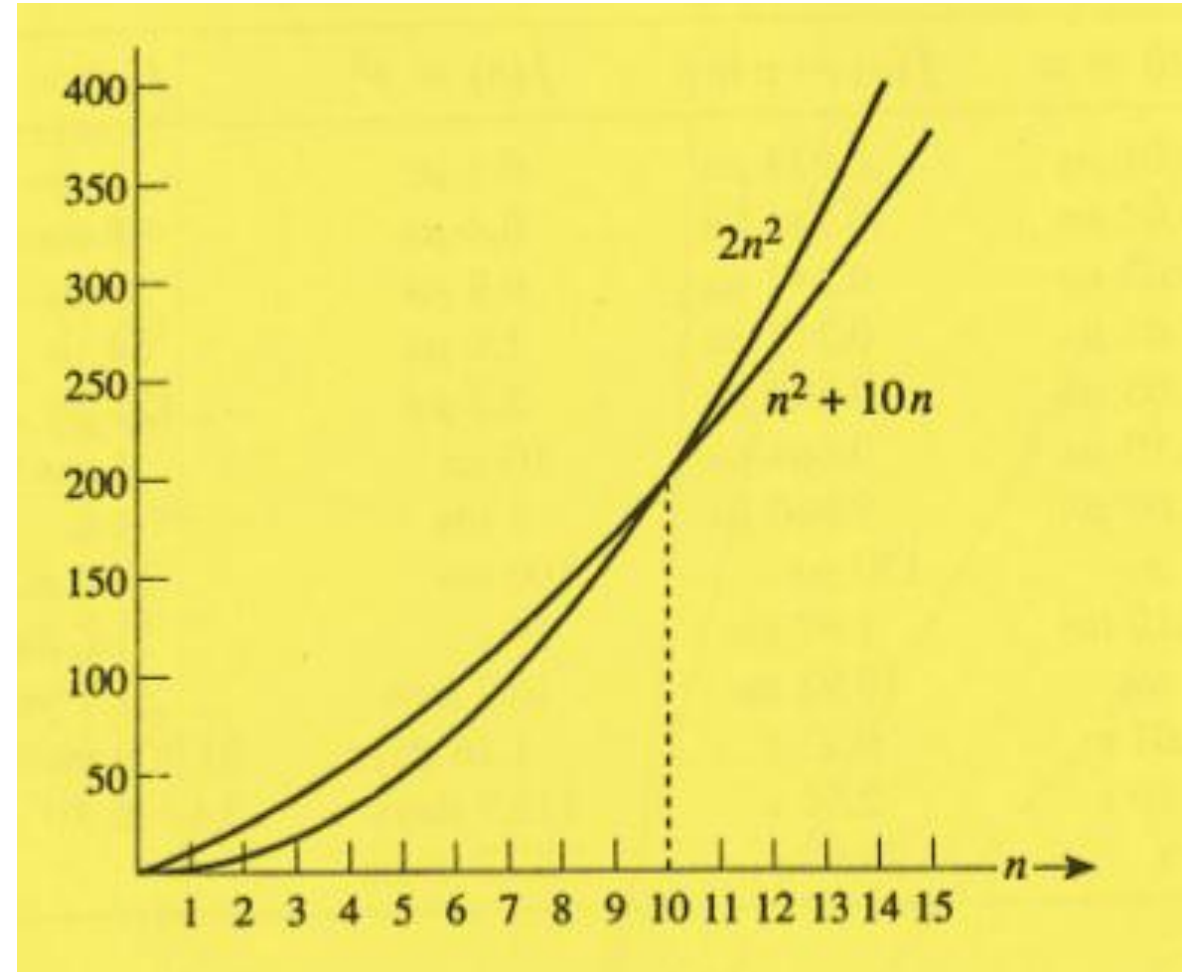
Buktikan Bahwa:

$$n^2 + 10n \in O(n^2)$$

Solusi:

$$n^2 + 10n \leq n^2 + n^2$$

$$n^2 + 10n \leq 2n^2, \text{ for } c=2 \text{ and } n_0=10$$



PEMBUKTIAN BIG OH (O) CARA 2

Buktikan Bahwa:

$$n^2 + 10n \in O(n^2)$$

Solusi:

$$n^2 + 10n \leq n^2 + 10n^2, \text{ for } c=1 \text{ and } n_0=10$$

$$n^2 + 10n \leq 11n^2, \text{ for } c=11 \text{ and } n_0=0$$

PENGELOMPOKAN ALGORITMA

Kelompok Algoritma	Nama
$O(1)$	Konstan
$O(\log n)$	Logaritmik
$O(n)$	Lanjar
$O(n \log n)$	$n \log n$
$O(n^2)$	kuadratik
$O(n^3)$	Kubik
$O(2^n)$	Eksponensial
$O(n!)$	faktorial

Urutan spektrum kompleksitas
algoritma:

$$\underbrace{O(1) < O(\log n) < O(n) < O(n \log n) < O(n^2) < O(n^3) < \dots}_{\text{Algoritma polynomial}} < \underbrace{O(2^n) < O(n!)}_{\text{Algoritma eksponensial}}$$

Algoritma polynomial

Algoritma eksponensial

LATIHAN

Buktikan Bahwa:

$$\Rightarrow n \in O(n^2)$$

$$\Rightarrow \frac{1}{2}n(n-1) \in O(n^2)$$

$$\Rightarrow 100n+5 \in O(n^2)$$

$$\Rightarrow 2n^2+6n+1 \in O(n^2)$$

$$\Rightarrow 6 \cdot 2^n + 2n^2 \in O(2^n)$$

Notasi Big Omega (Ω)

PENJELASAN BIG OMEGA (Ω)

Definisi $\Omega(g(n))$:

Kumpulan semua fungsi yang order of growth-nya **lebih besar atau sama dengan** $g(n)$.

Contoh:

$$\Rightarrow n^3 \in \Omega(n^2)$$

$$\Rightarrow \frac{1}{2} n (n-1) \in \Omega(n^2)$$

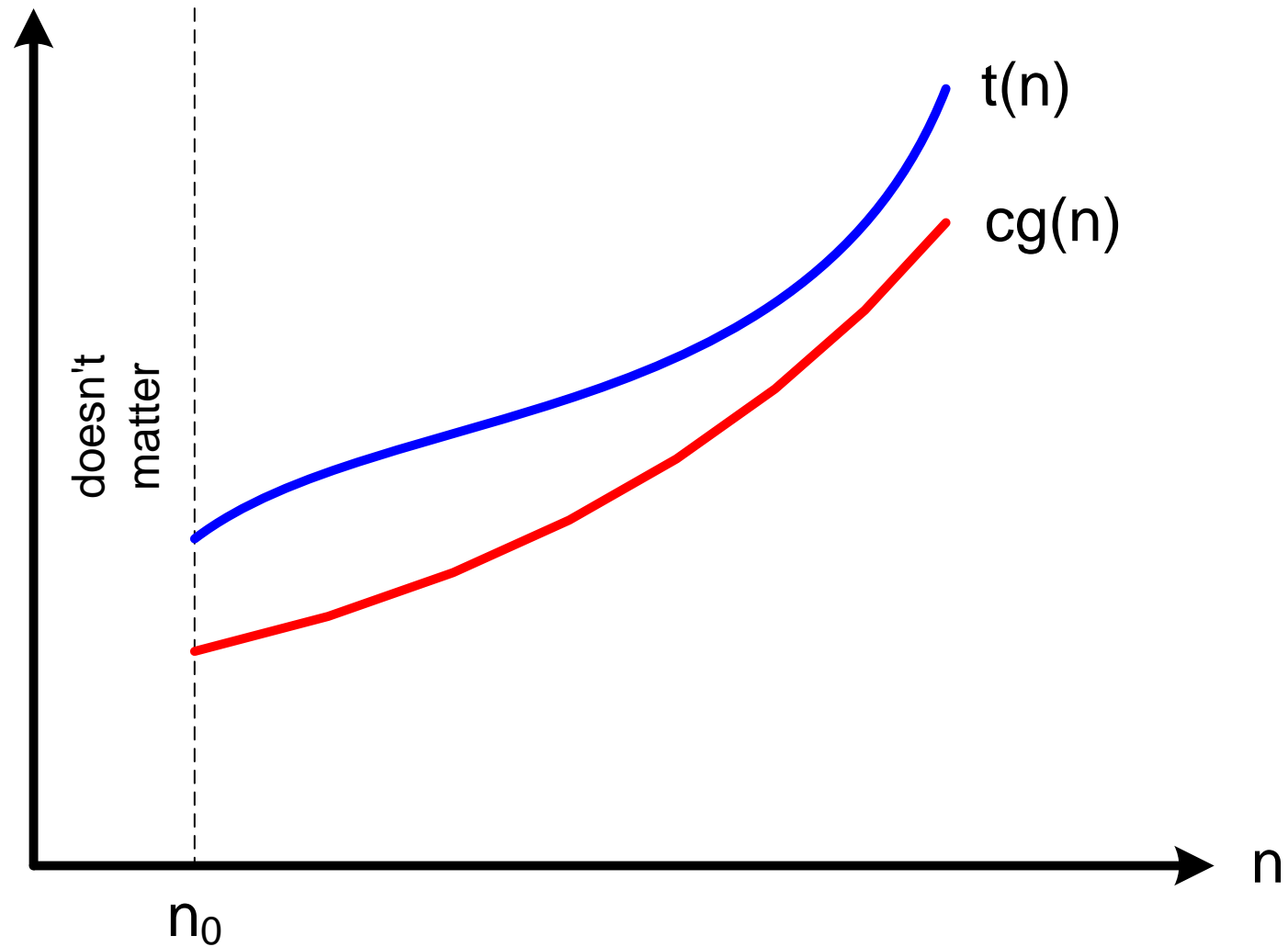
$$\Rightarrow 100n + 5 \notin \Omega(n^2)$$

DEFINISI BIG OMEGA (Ω) \rightarrow FORMAL

"Fungsi $t(n)$ dikatakan berada di dalam $\Omega(g(n))$, dinyatakan dengan $t(n) \in O(g(n))$, apabila $t(n)$ dibatasi bawah oleh beberapa konstanta dikali $g(n)$ untuk seluruh n , jika ada beberapa konstanta c positif dan beberapa bilangan bulat tak negatif N_0 sedemikian sehingga:

$$t(n) \geq cg(n) \text{ untuk semua } n \geq n_0"$$

ILUSTRASI $t(n) \in \Omega(g(n))$



PEMBUKTIAN BIG OMEGA (Ω)

Buktikan Bahwa:

$$n^3 \in \Omega(n^2)$$

Solusi:

$$n^3 \geq n^2 \text{ for } c=1 \text{ and all } n \geq 0$$

PEMBUKTIAN BIG OMEGA (Ω)

Buktikan Bahwa:

$$10n^2 + 4n + 2 \in \Omega(n^2)$$

Solusi:

$$10n^2 + 4n + 2 \geq n^2 \text{ for } c=1 \text{ and all } n \geq 0$$

Notasi Big Theta (Θ)

PENJELASAN BIG THETA (Θ)

Definisi $\Theta(g(n))$:

Kumpulan semua fungsi yang order of growth-nya **sama dengan** $g(n)$.

Contoh:

$$\Rightarrow an^2 + bn + c; a > 0 \in \Theta(n^2); n^2 + \sin n \in \Theta(n^2)$$

$$\Rightarrow \frac{1}{2} n (n-1) \in \Theta(n^2); n^2 + \log n \in \Theta(n^2)$$

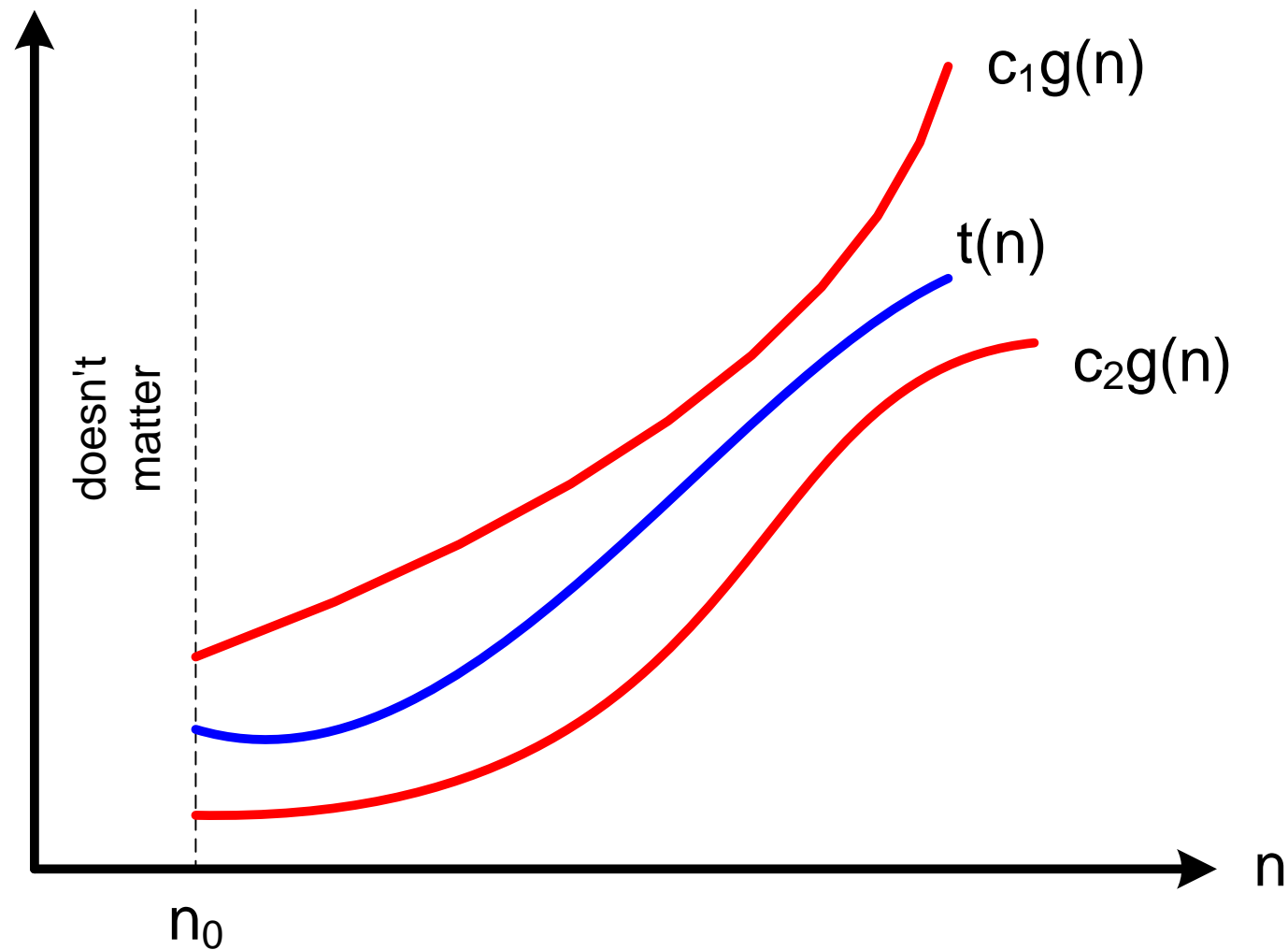
$$\Rightarrow 100n + 5 \notin \Theta(n^2); n^3 \notin \Theta(n^2)$$

DEFINISI BIG THETA (Θ) \rightarrow FORMAL

"Fungsi $t(n)$ dikatakan berada di dalam $\Theta(g(n))$, dinyatakan dengan $t(n) \in O(g(n))$, apabila $t(n)$ dibatasi atas dan dibatasi bawah oleh beberapa konstanta dikali $g(n)$ untuk seluruh n , jika ada beberapa konstanta c positif dan beberapa bilangan bulat tak negatif N_0 sedemikian sehingga:

$$c_2 g(n) \leq t(n) \leq c_1 g(n) \text{ for all } n \geq n_0"$$

ILUSTRASI $t(n) \in \Theta(g(n))$



PEMBUKTIAN BIG THETA (Θ)

Buktikan Bahwa:

$$\frac{1}{2}n(n-1) \in \Theta(n^2)$$

Solusi:

➡ Batas atas $\frac{1}{2}n(n-1) =$

$$\frac{1}{2}n^2 - \frac{1}{2}n \leq \frac{1}{2}n^2 \text{ (for all } n \geq 0)$$

➡ Batas bawah : $\frac{1}{2}n(n-1) =$

$$\frac{1}{2}n^2 - \frac{1}{2}n \geq \frac{1}{2}n^2 - \frac{1}{2}n^2 \text{ (for all } n \geq 2) = \frac{1}{4}n^2$$

$$\Rightarrow c_1 = \frac{1}{2}, c_2 = \frac{1}{4}, n_0 = 2$$

TERIMA KASIH