

Regresi

PENGANTAR KECERDASAN BUATAN

Dr. Prima Dewi Purnamasari
Program Studi Teknik Komputer FTUI



Halo!
Selamat datang
kembali di kuliah
pengantar kecerdasan
buatan!

Dr. Prima Dewi Purnamasari

Sebelumnya:
Pembelajaran Mesin

Dr. Prima Dewi Purnamasari



Sekarang:
Regresi

Dr. Prima Dewi Purnamasari

**Capaian
pembelajaran**

**Mampu menjelaskan metode regresi
dalam pembelajaran mesin**



Tipe pembelajaran machine learning



Supervised Learning

Data yang dipelajari memiliki label input dan output yang jelas (berupa nilai kontinyu, kategori atau kelas)



MACHINE LEARNING

Unsupervised Learning

Data yang dipelajari hanya memiliki label input tanpa output yang jelas



Reinforcement Learning

Data didapatkan berdasarkan reward dari lingkungan mesin



Supervised Learning

Data yang dipelajari memiliki label input dan output yang jelas (berupa nilai kontinu, kategori atau kelas)

Label berupa kategori/kelas

Klasifikasi



APEL



PISANG



ANGGUR



Rp.300juta



Rp.600juta



Rp.700juta

Label berupa nilai kontinu

Regresi



Regresi dengan metode statistika



Diketahui harga rumah



300 jt



600 jt



700 jt






INPUT

Harga???

OUTPUT



Misalnya hanya diperhatikan luas rumah saja

INPUT	Luas (m ²)			OUTPUT
	80	200	220	Harga (Rp)
				
	300 jt	600 jt	700 jt	



120

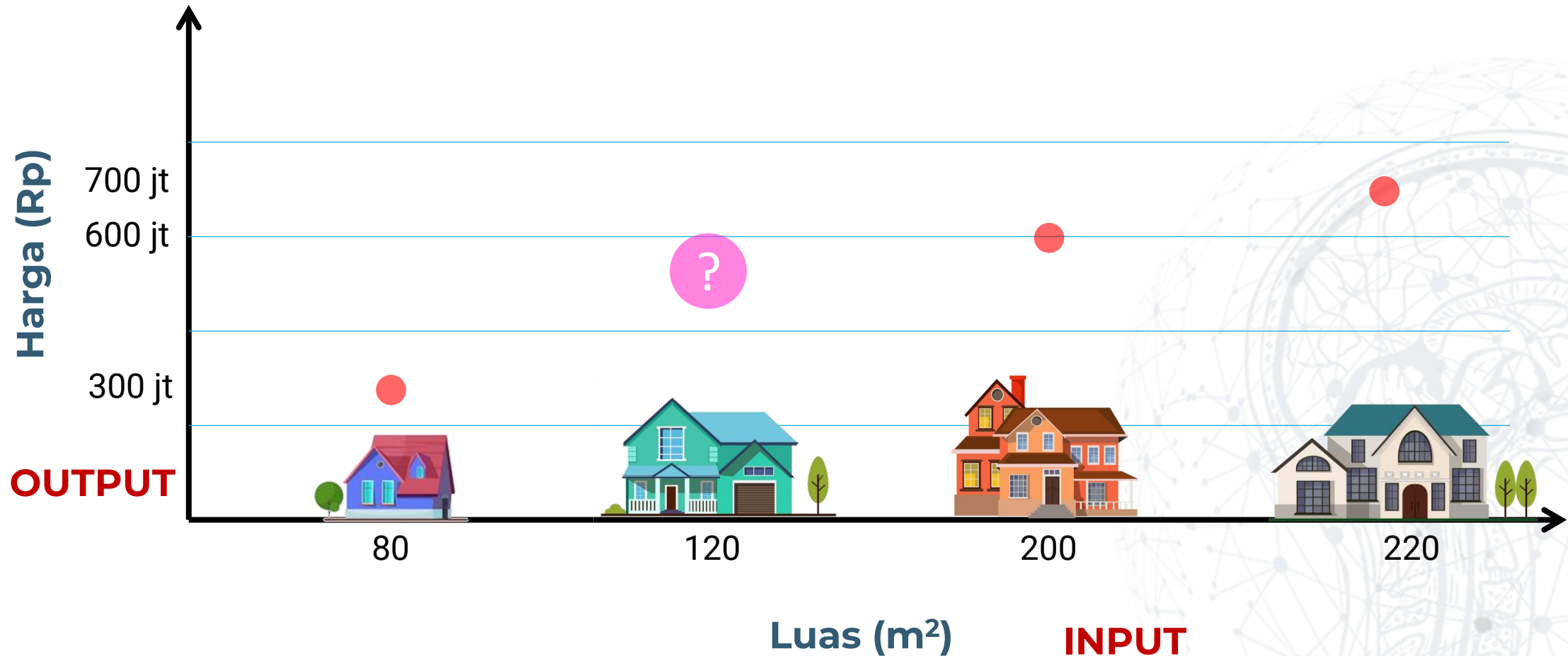
INPUT

Harga???

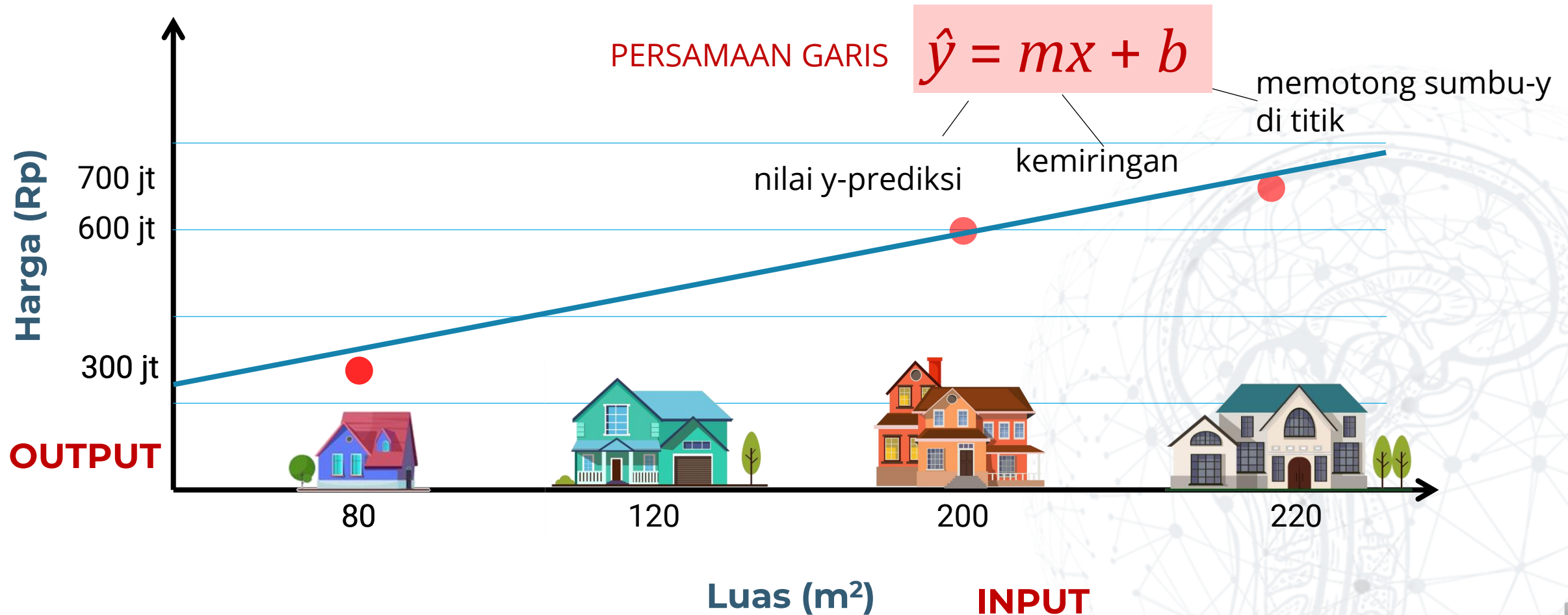
OUTPUT



Cari hubungan antara
input (*independent variable*) dan
output (*dependent/response variable*)



Hubungan yang paling mudah untuk dibuat: GARIS



Metode regresi statistika (*simple linear regression*)

$$\hat{y} = mx + b$$

$$m = \frac{n \sum xy - (\sum x)(\sum y)}{n \sum x^2 - (\sum x)^2}$$

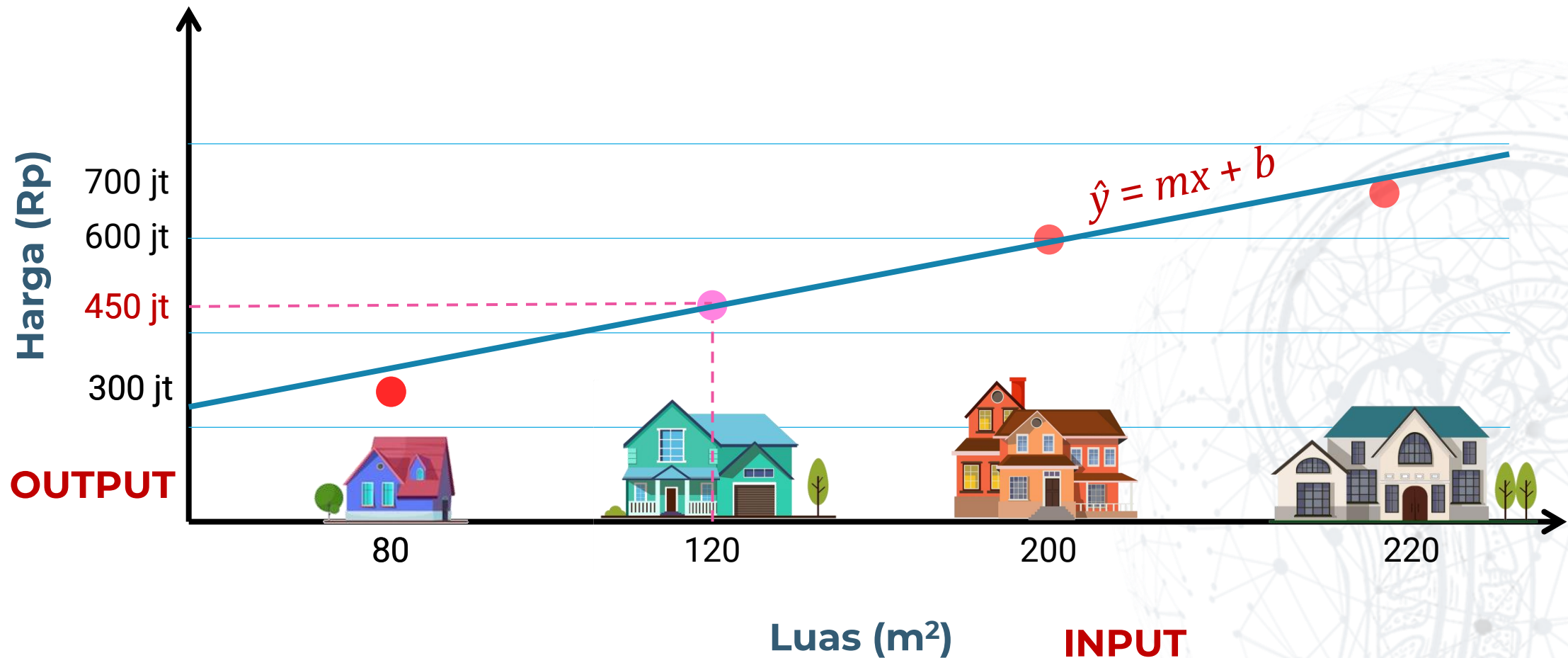
$$b = \bar{y} - m\bar{x} = \frac{\sum y}{n} - m \frac{\sum x}{n}$$

Dengan:

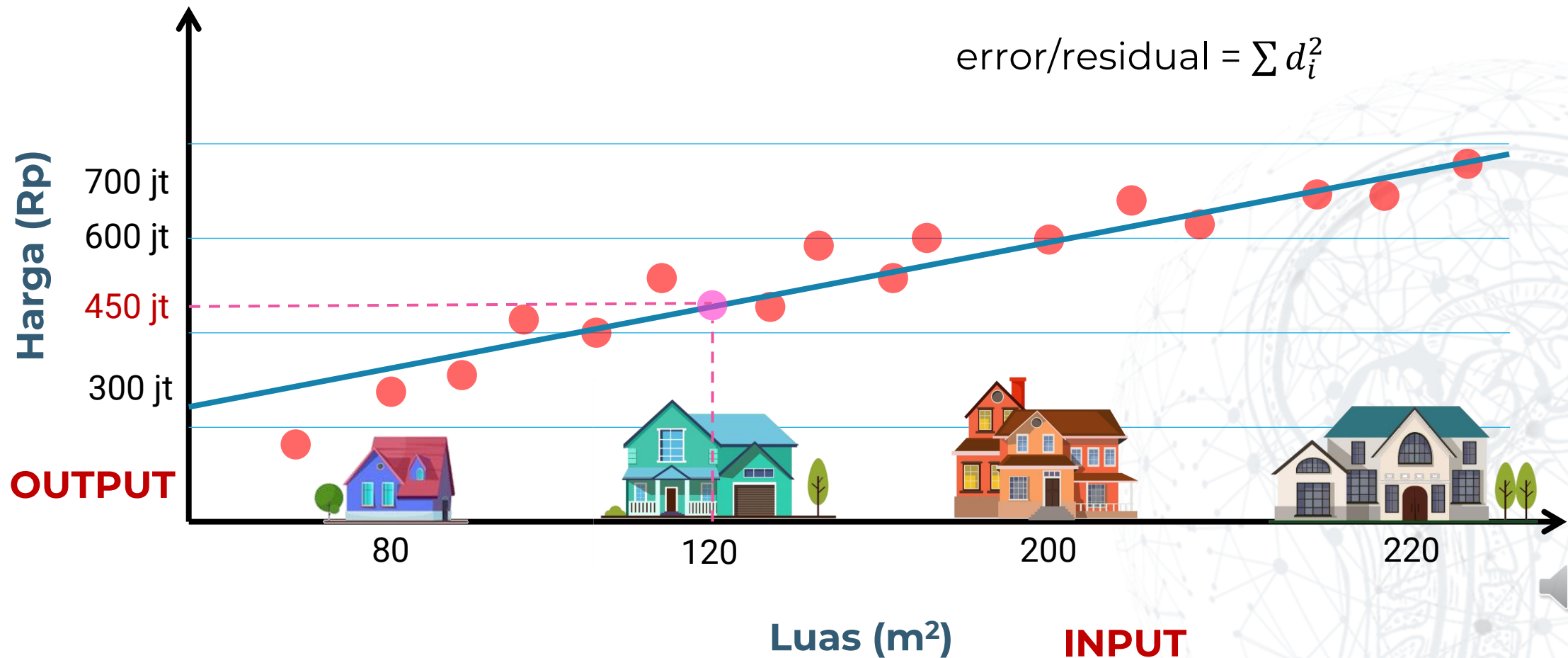
- data adalah pasangan x & y
- x = independent variable
- y = dependent variable
- \bar{x} = rata-rata x
- \bar{y} = rata-rata y
- n = jumlah data
- \hat{y} = y hasil prediksi
- m = gradien garis
- b = intercept pada sumbu y



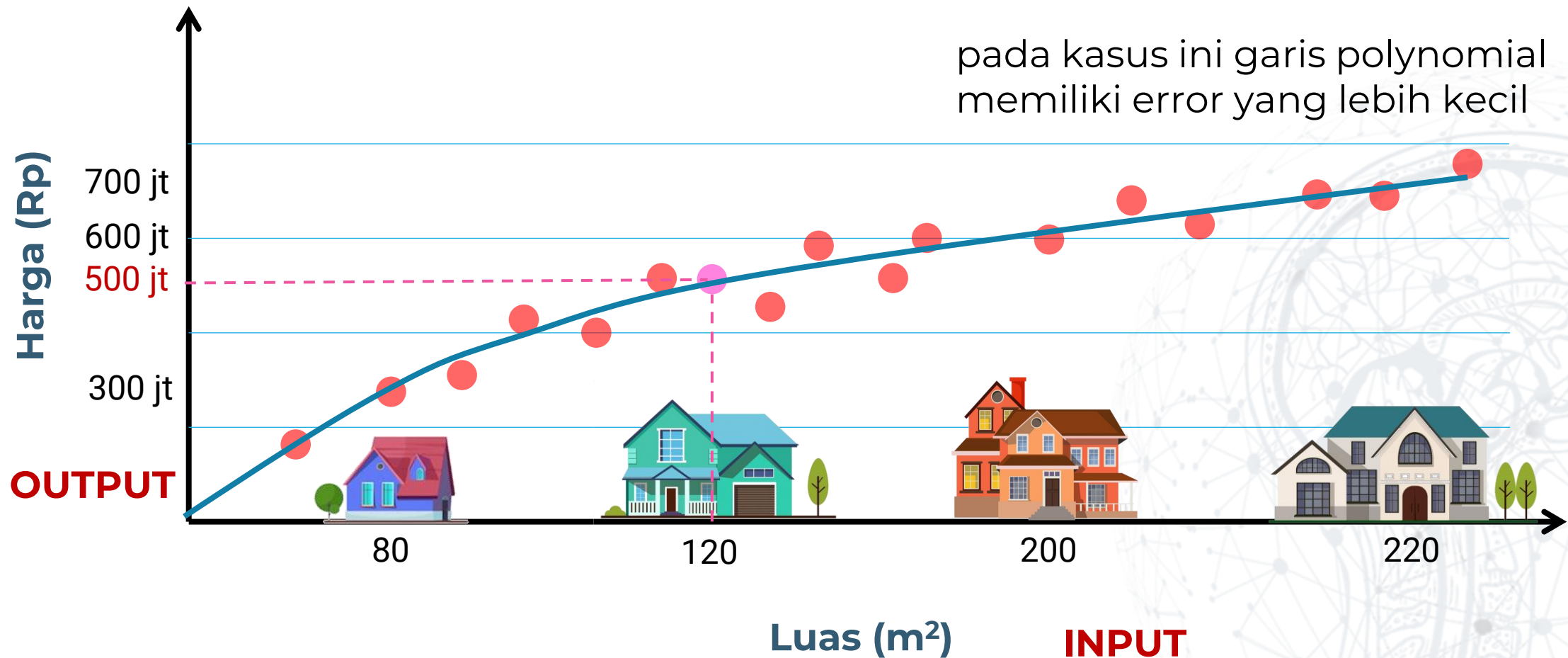
Dari persamaan garis (m dan b) dapat dicari nilai \hat{y} jika diketahui x



**Garis terbaik = garis yang memiliki
error/residual terkecil**



Ada kalanya garis lurus **bukan** pilihan terbaik



Bagaimana jika dependent variable ada banyak?



Bagaimana jika dependent variable ada banyak?

**Matriks semakin besar
→ Biaya komputasi semakin mahal**



REGRESI dengan MACHINE LEARNING

Semakin kompleks permasalahan yang ingin dipecahkan, maka secara komputasi akan semakin “mahal”.

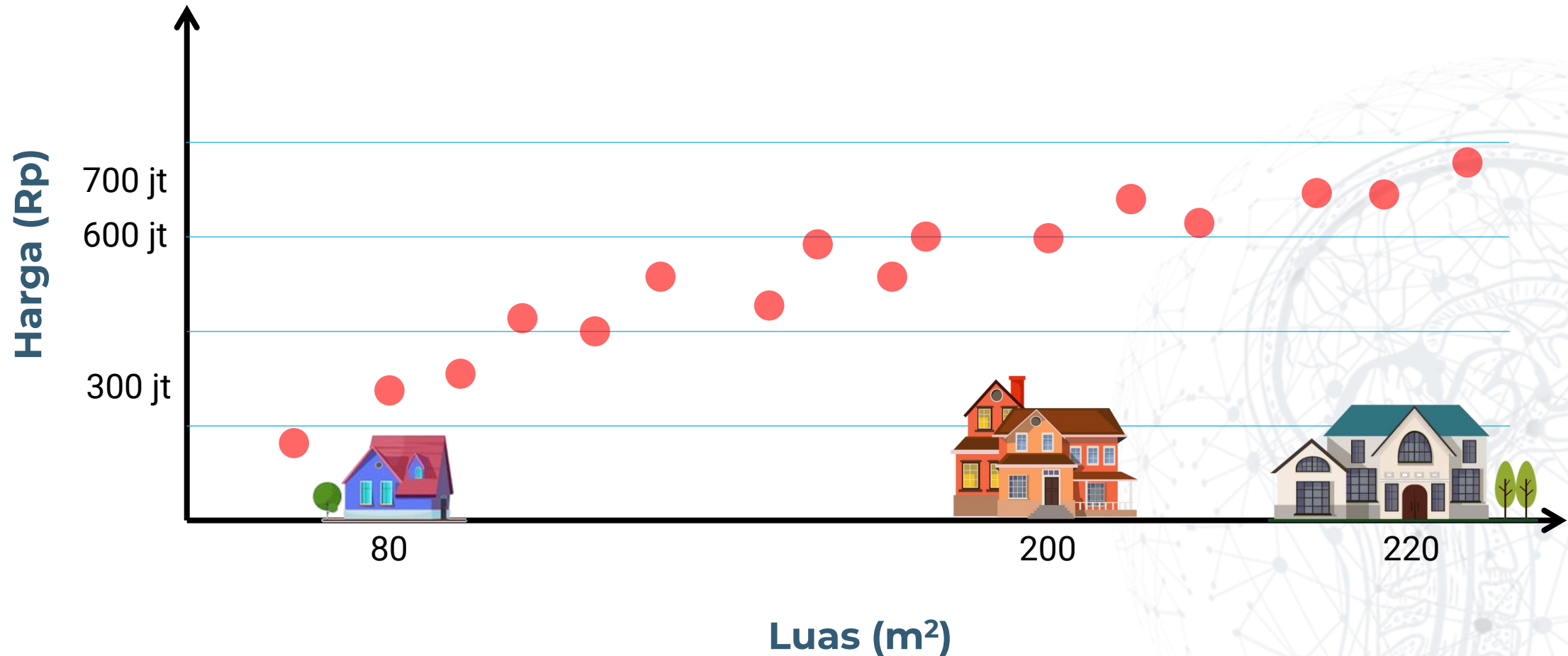
Oleh karena itu pendekatan aproksimasi atau optimasi dengan machine learning jadi lebih “murah” dibanding pendekatan statistika



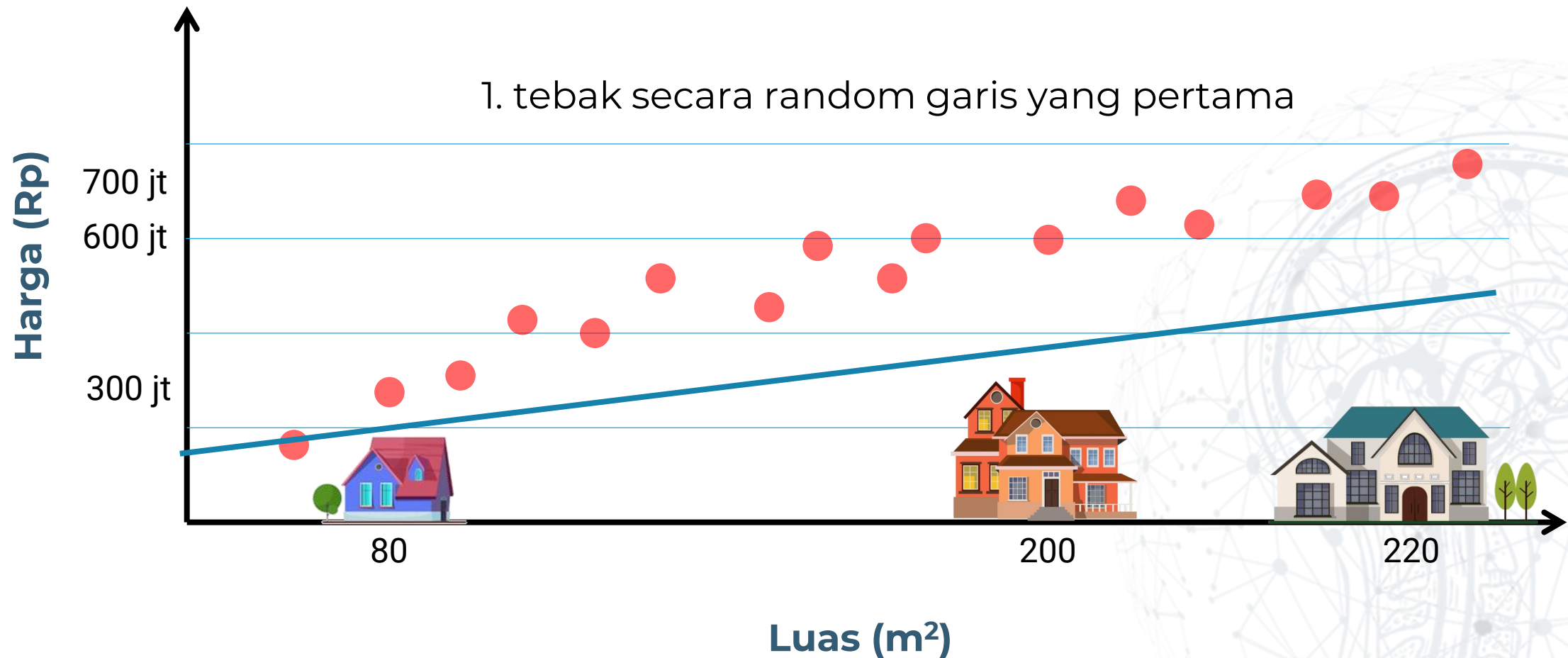
Salah satu teknik dasar dalam melakukan optimasi dalam machine learning adalah
GRADIENT DESCENT



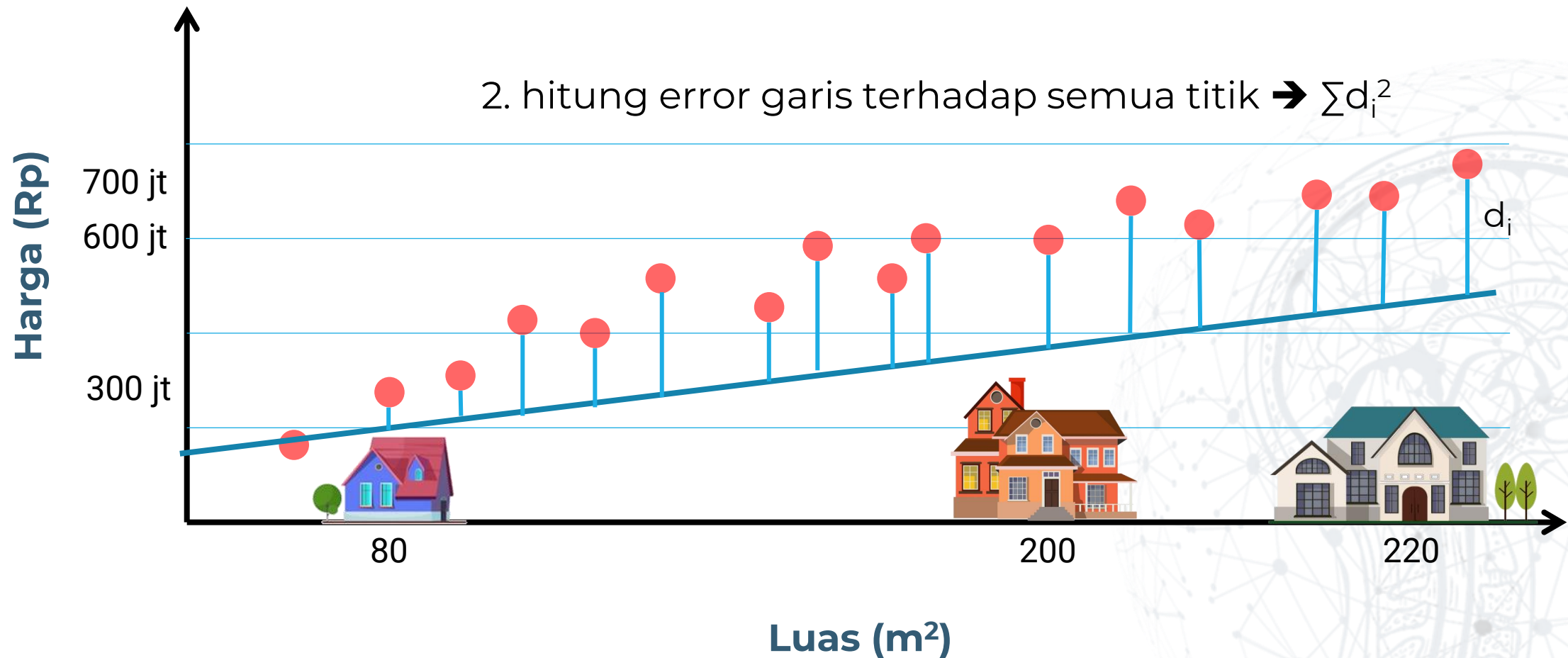
GRADIENT DESCENT: mencari gradien berulang kali agar diperoleh error yang terkecil



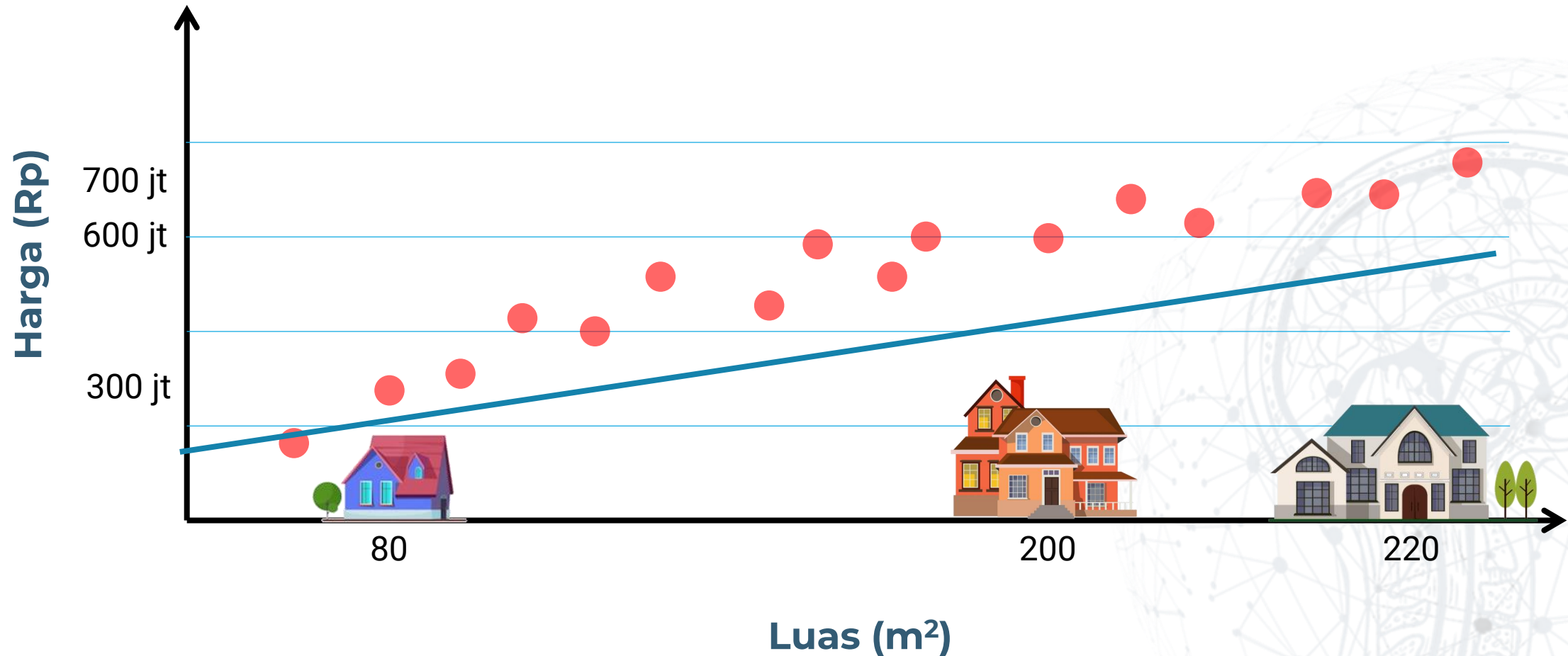
GRADIENT DESCENT: mencari gradien berulang kali agar diperoleh error yang terkecil



GRADIENT DESCENT: mencari gradien berulang kali agar diperoleh error yang terkecil

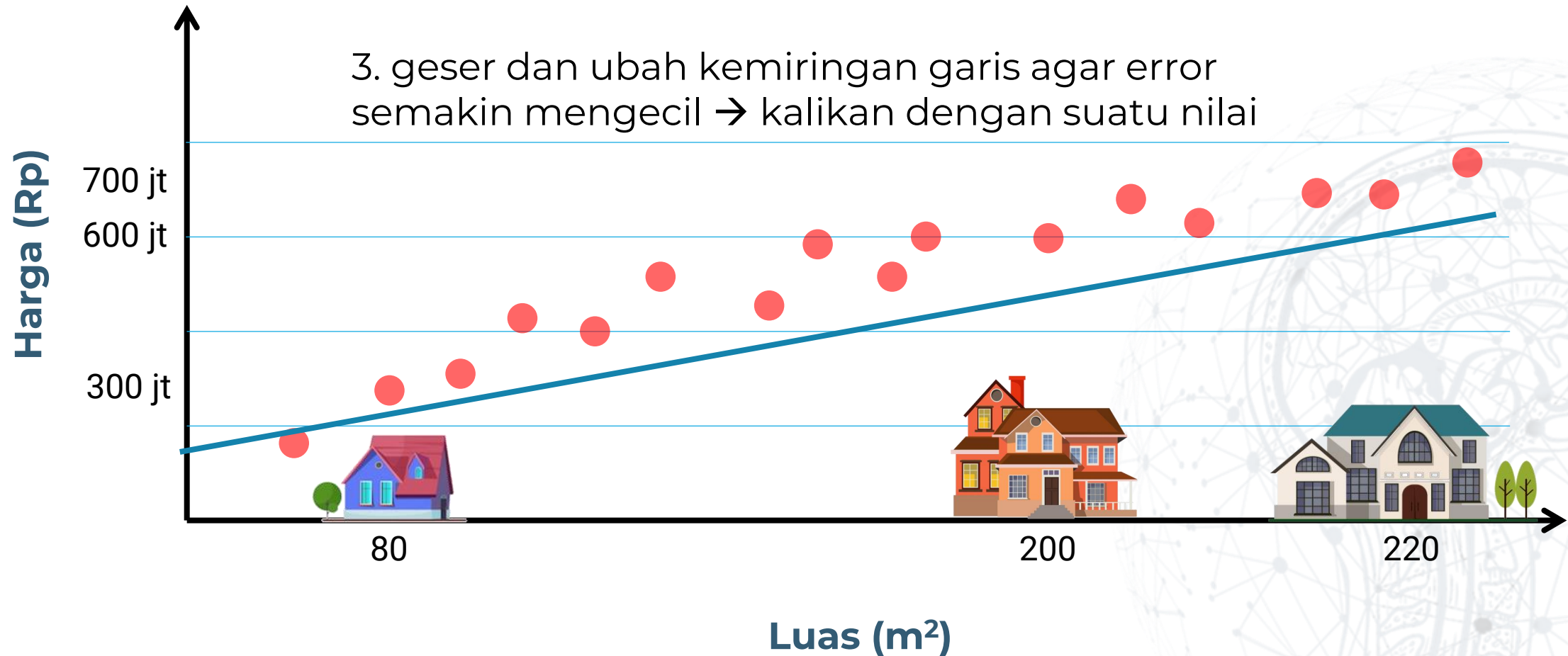


GRADIENT DESCENT: mencari gradien berulang kali agar diperoleh error yang terkecil

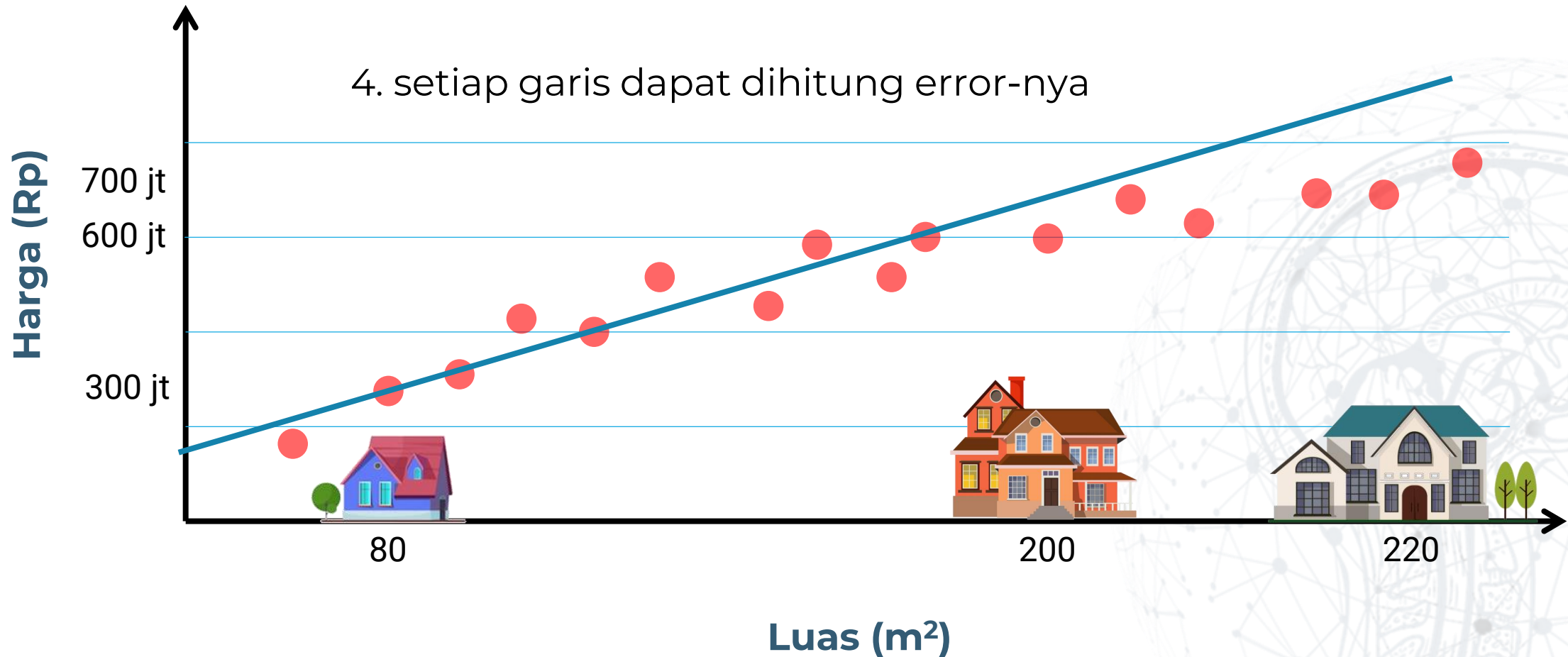


GRADIENT DESCENT: mencari gradien berulang kali agar diperoleh error yang terkecil

3. geser dan ubah kemiringan garis agar error
semakin mengecil → kalikan dengan suatu nilai

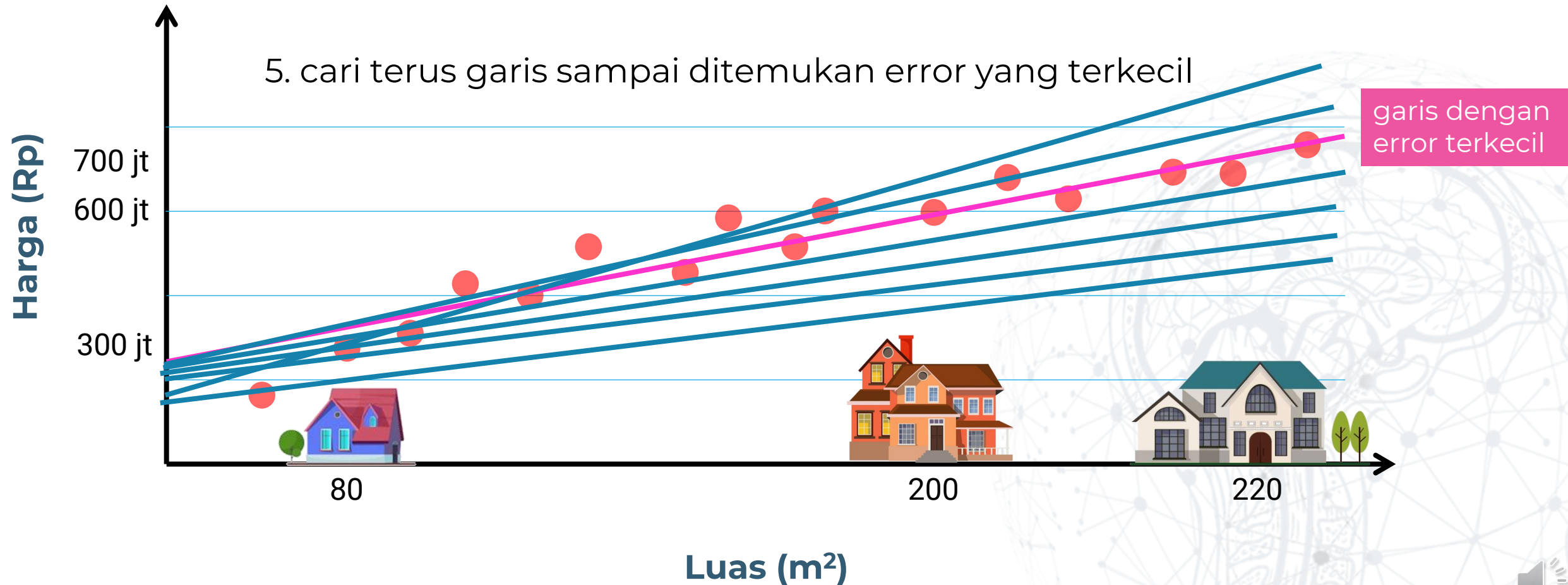


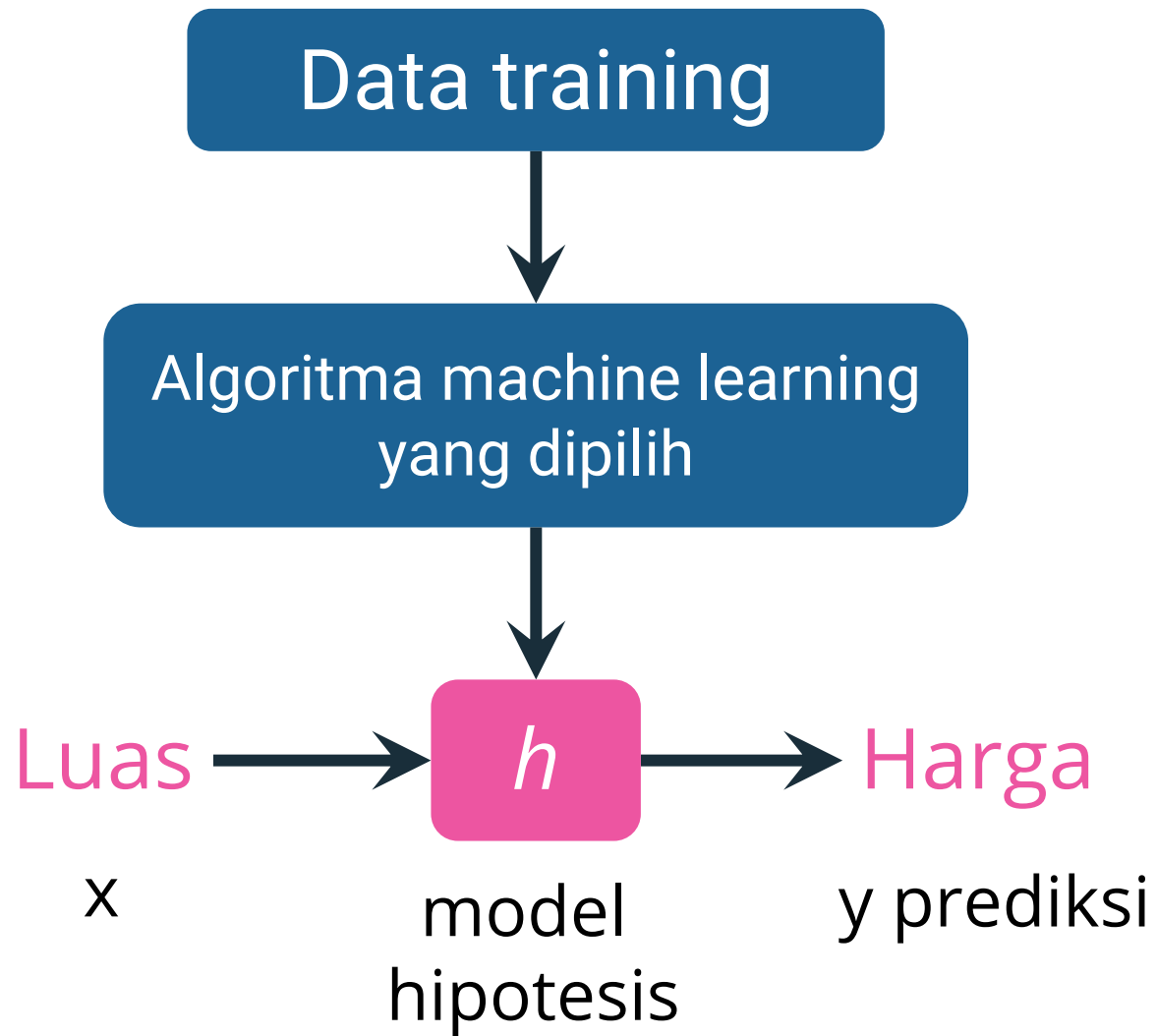
GRADIENT DESCENT: mencari gradien berulang kali agar diperoleh error yang terkecil



GRADIENT DESCENT: mencari gradien berulang kali agar diperoleh error yang terkecil

5. cari terus garis sampai ditemukan error yang terkecil





h memetakan x ke y



Bagaimana menyatakan h ?

tergantung dari keinginan kita
ingin memodelkan data ke bentuk apa

Regresi linier dengan 1 variabel

persamaan garis biasa

$$\hat{y} = mx + b$$

supaya dapat dikembangkan ke lebih
dari 1 variabel independen

$$\hat{y} = \theta_0 + \theta_1 x$$

dapat juga ditulis sebagai fungsi
hipotesis

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$



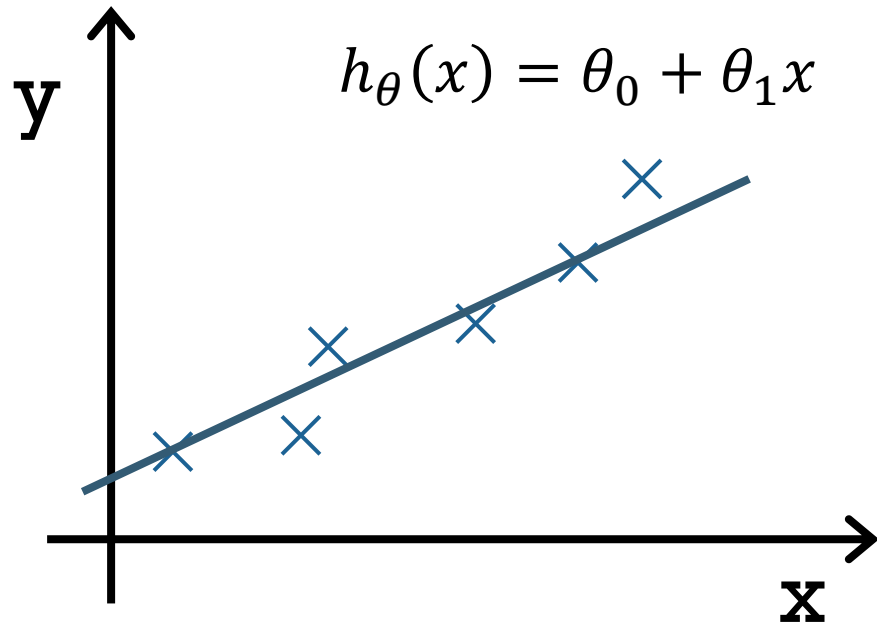
$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

**Bagaimana cara mencari θ_i
tanpa metode SLR?**



IDE:

Cari θ_0 dan θ_1 sehingga $h_{\theta}(x)$ mendekati y dari data training yang dimiliki (x,y)



cost function

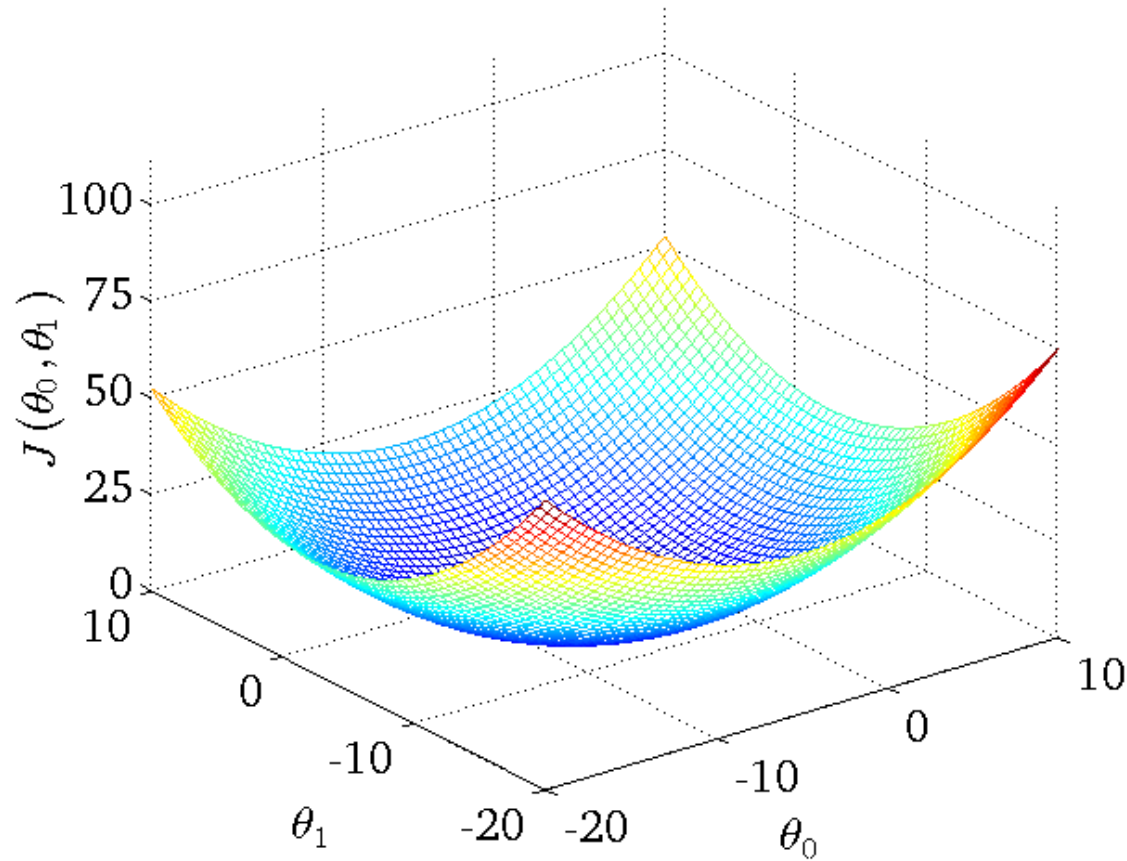
$$J(\theta_0, \theta_1) = \frac{1}{m} \sum (h_{\theta}(x) - y)^2$$

Mean squared
error (MSE)



$$\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$$

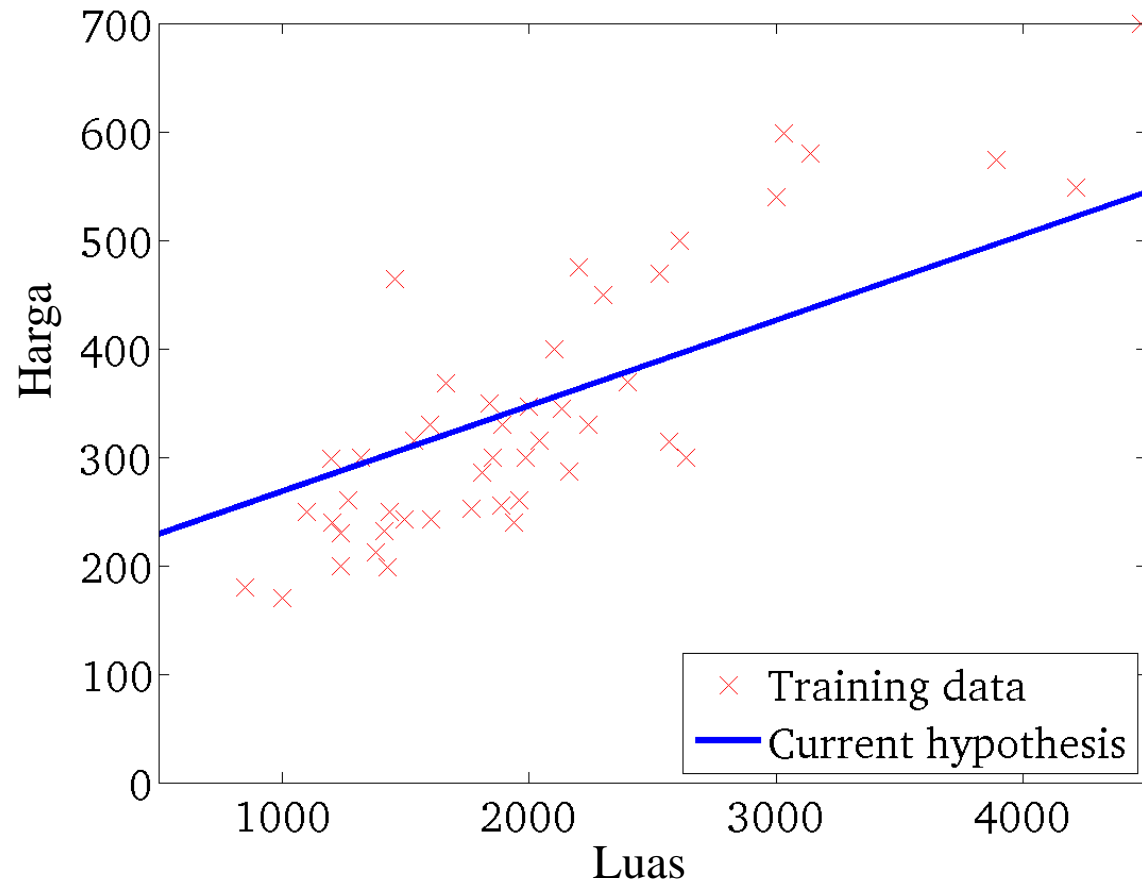




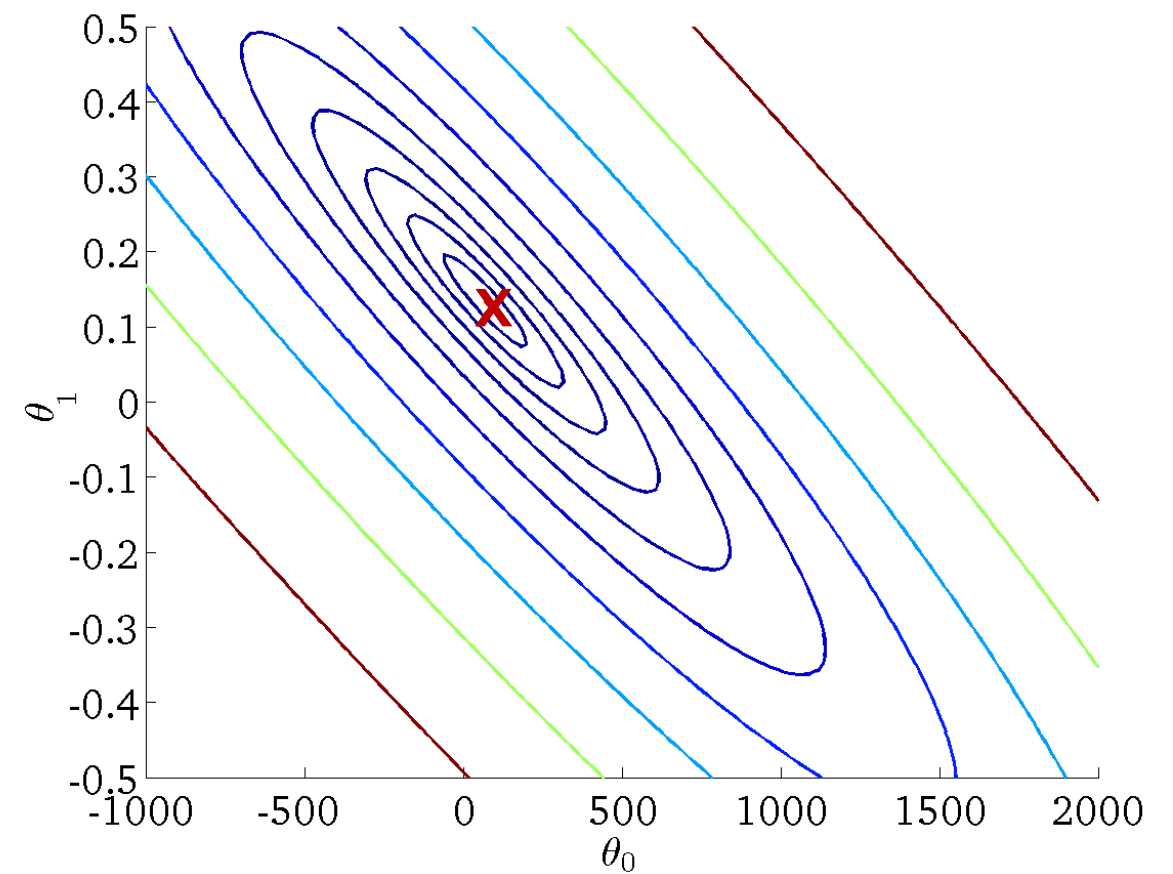
**Bagaimana cara
mencari titik
 $J(\theta_0, \theta_1)$
yang paling
minimal
(kalau bisa 0)**



Line plot Luas vs Harga



3D mesh plot cost function $J(\theta_0, \theta_1)$



Algoritma dasar

- Mulai dengan nilai inisial θ_0, θ_1
- Hitung cost: $J(\theta_0, \theta_1) = \frac{1}{m} \sum (h_{\theta}(x) - y)^2$
- Ubah θ_0, θ_1
- Lakukan sampai didapatkan $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$



ALGORITMA GRADIENT DESCENT

ulangi sampai konvergen: {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

Learning rate

untuk $j = 0$ dan $j = 1$

assignment
a:=b



ulangi sampai konvergen: {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum (h_{\theta}(x) - y)$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum (h_{\theta}(x) - y) \cdot x$$

}

$$\frac{d}{d\theta_0} \cdot J(\theta_0, \theta_1)$$

$$\frac{d}{d\theta_1} \cdot J(\theta_0, \theta_1)$$

θ_0 dan θ_1 harus diupdate secara simultan



Jika independent variabel banyak

**implementasikan algoritma ke dalam
program dengan bentuk matriks**



Bagaimana cara
menentukan
learning rate α ?



Tidak ada panduan baku untuk memilih α

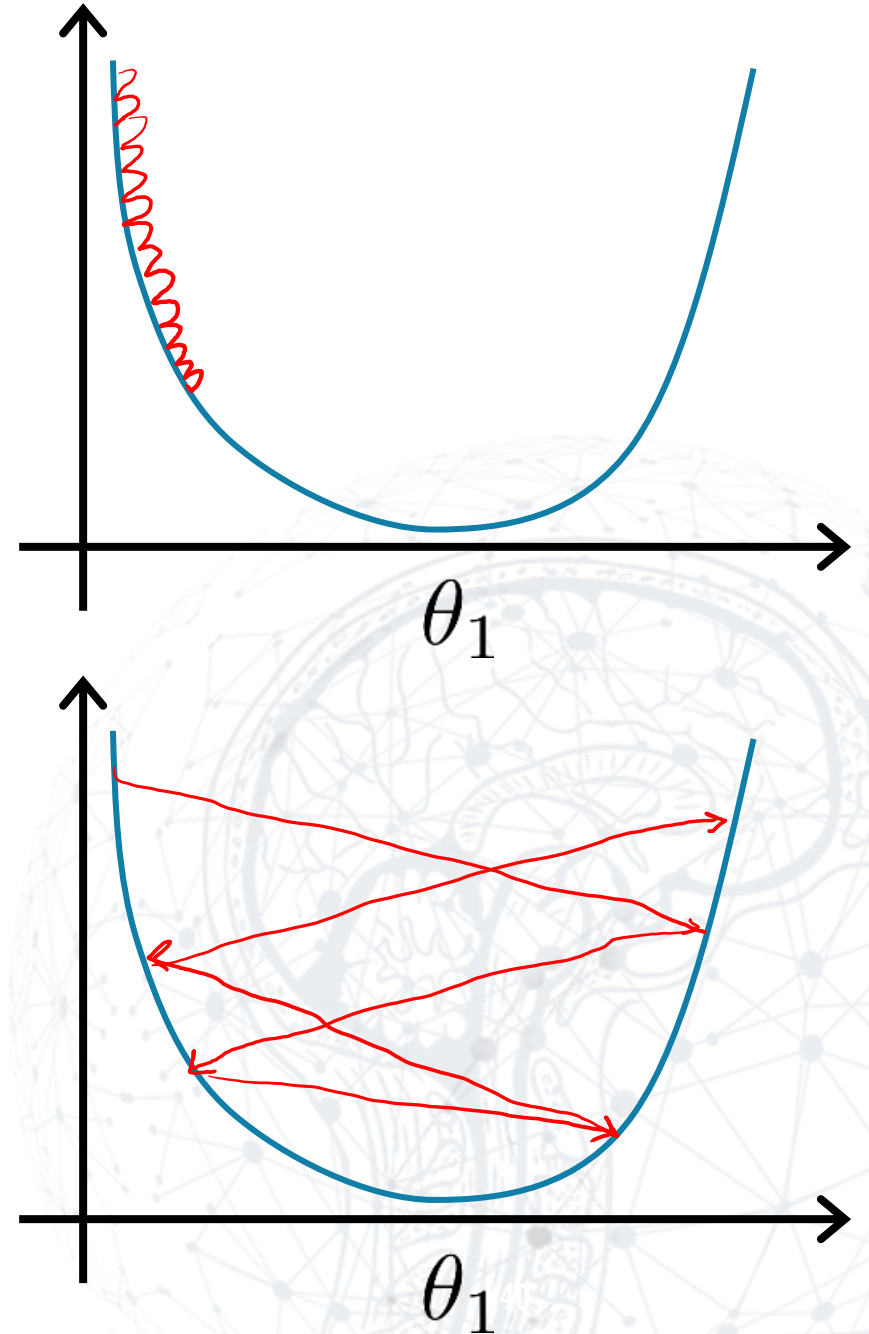
$\alpha \rightarrow$ hyperparameter

ditebak dan dicari secara heuristik (coba-coba) atau menggunakan metode optimasi

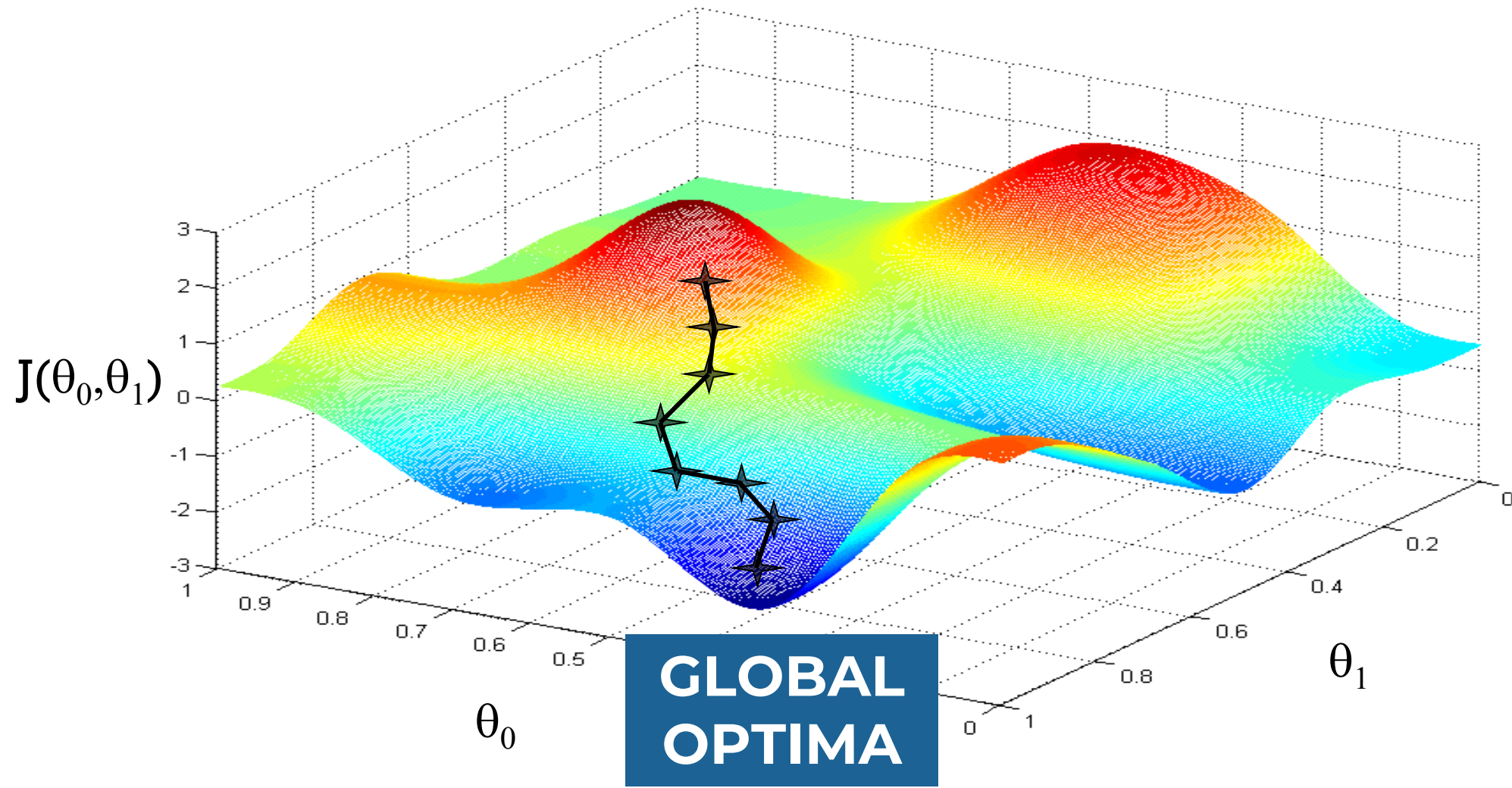


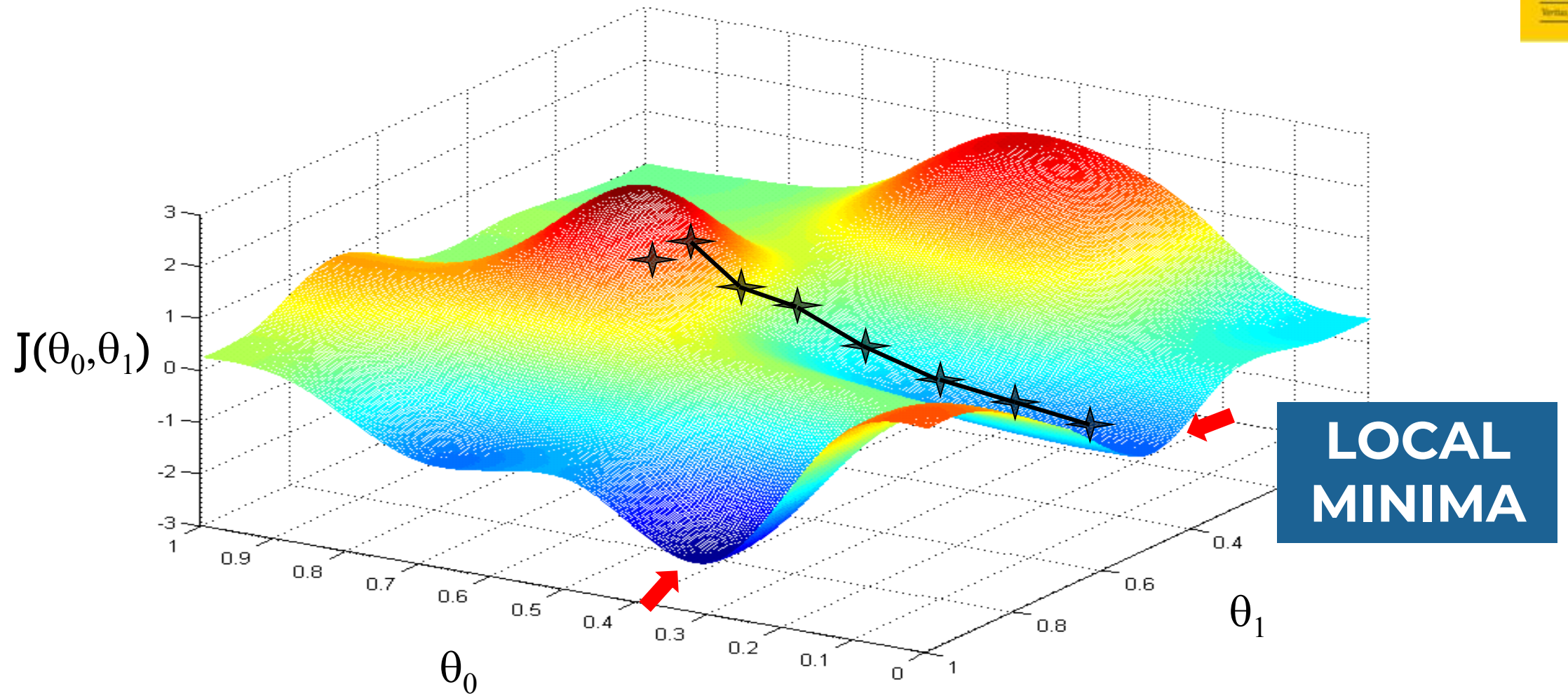
Jika **α terlalu kecil**, proses gradient descent akan terlalu lama \rightarrow error hanya sedikit turunnya di setiap epoch

Jika **α terlalu besar** gradient descent bisa jadi overshoot local minima. Bisa jadi gagal mencapai konvergen (atau bahkan menjadi divergen).



Sampai kapan
pembelajaran
harus dilakukan?





Oleh karena itu biasanya dalam machine learning yang diawali dengan **inisialisasi random** maka harus **diulang beberapa kali** dengan harapan akan diperoleh local minima yang mendekati global optima

EPOCH

**1 epoch =
1 kali pembelajaran untuk satu set data
training**

100 epoch = 100 kali belajar



Proses pembelajaran dapat dihentikan jika sudah mencapai **epoch** tertentu atau sampai diperoleh **loss/error** di bawah standar yang ditetapkan

STOPPING CRITERIA

Stochastic Gradient Descent

Nilai θ_j di-update untuk setiap pasangan data training x dan y .
Juga disebut incremental gradient descent

Batch Gradient Descent



Nilai θ_j di-update ketika sudah belajar 1 set data (1 epoch)

Mini batch sekelompok data digunakan dalam 1x iterasi, misal batch_size = 16 artinya 16 buah data digunakan untuk update θ_j

Bagaimana cara
implementasinya
ke dalam program
Python
?

Kesimpulan

- Pada regresi, mesin akan memprediksi nilai kontinyu
- Metode gradient descent pada regresi machine learning dilakukan dengan mencari parameter garis yang memiliki error minimum
- 1 set pembelajaran disebut epoch
- Metode gradient descent: stochastic, batch, mini batch



Selanjutnya:
Klasifikasi