

Convex Hull Trick

Bukan Convex Hull Biasa!

William Gozali
Pelatnas 3 TOKI 2015

Soal – SPOJ ACQUIRE

- Ada N tanah ($1 \leq N \leq 10^5$)
- Tanah ke- i berukuran $w[i] \times h[i]$ ($1 \leq w[i], h[i] \leq 10^5$)
- Harga tanah == luas tanah
- Ada program diskon, jika membeli tanah $\{a, b, c, \dots\}$, harganya:
$$\max(w[a], w[b], w[c], \dots) * \max(h[a], h[b], h[c], \dots)$$
- Kita mau membeli semua tanah, tentukan pengelompokkan cara pembelian tanah, supaya total harganya minimum!

Contoh

$$N = 5$$

- $100*1$
- $2*1$
- $15*15$
- $20*5$
- $1*100$

Solusi:

- $100*1, 2*1$, harga=100
- $1*100$, harga=100
- $15*15, 20*5$, harga=300

Total = 500 (minimum)

Observasi 1...

Jika ada tanah dengan ukuran:

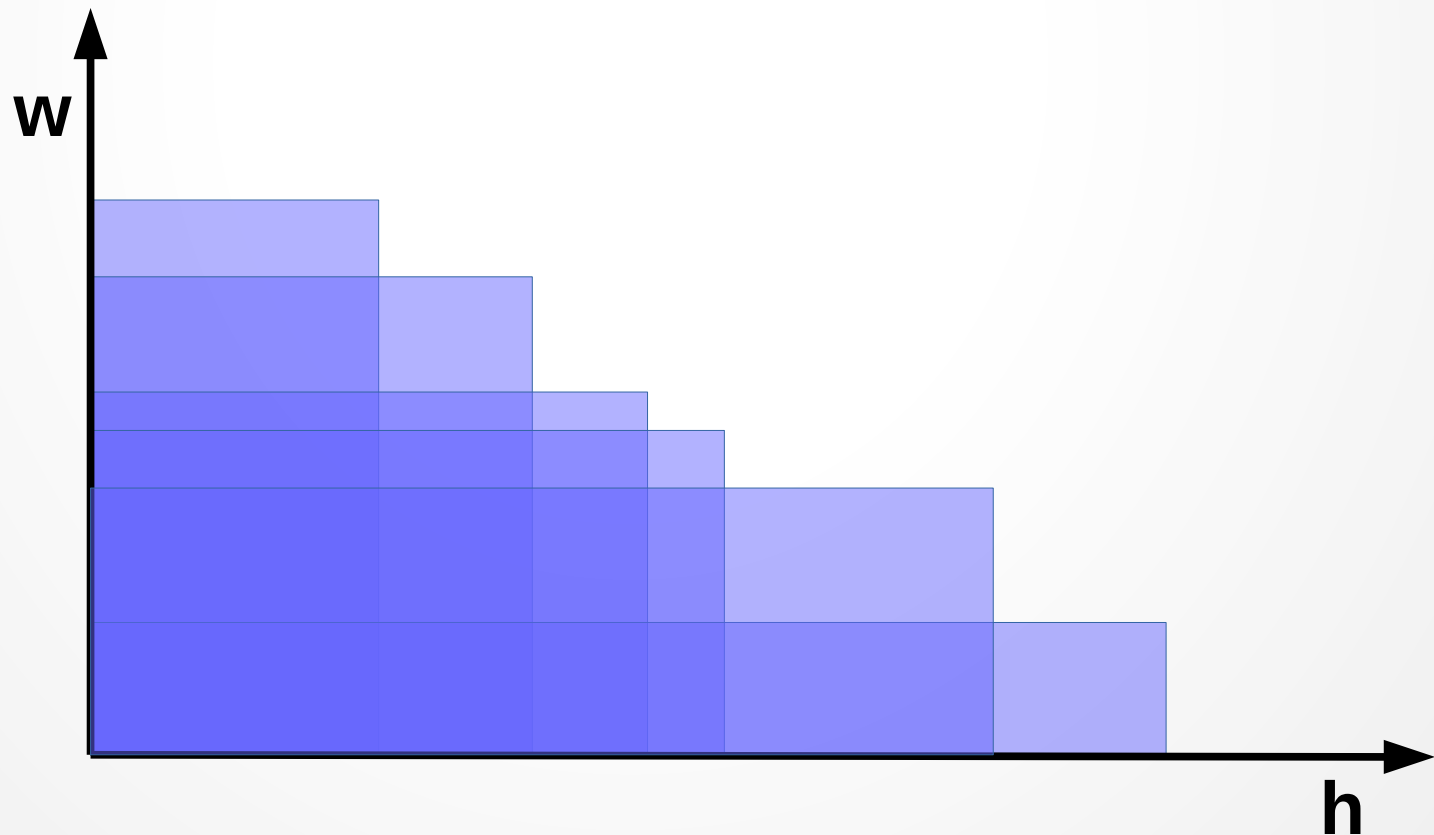
- 10×3
- 8×1

Maka tanah 8×1 “gratis”, karena bisa diserap ke tanah 10×3

Jadi bisa kita sort ukuran tanah berdasarkan w, lalu buang semua tanah “gratis”. Pembuangan secara efisien bisa menggunakan struktur data BIT

Observasi 2...

Jika semua tanah “gratis” sudah diserap:



Observasi 2... (lanj.)

Yang tersisa adalah tanah-tanah $w[1]*h[1]$, $w[2]*h[2]$, $w[3]*h[3]$, ... dengan sifat:

- $w[1] < w[2] < w[3] < \dots$
- $h[1] > h[2] > h[3] > \dots$

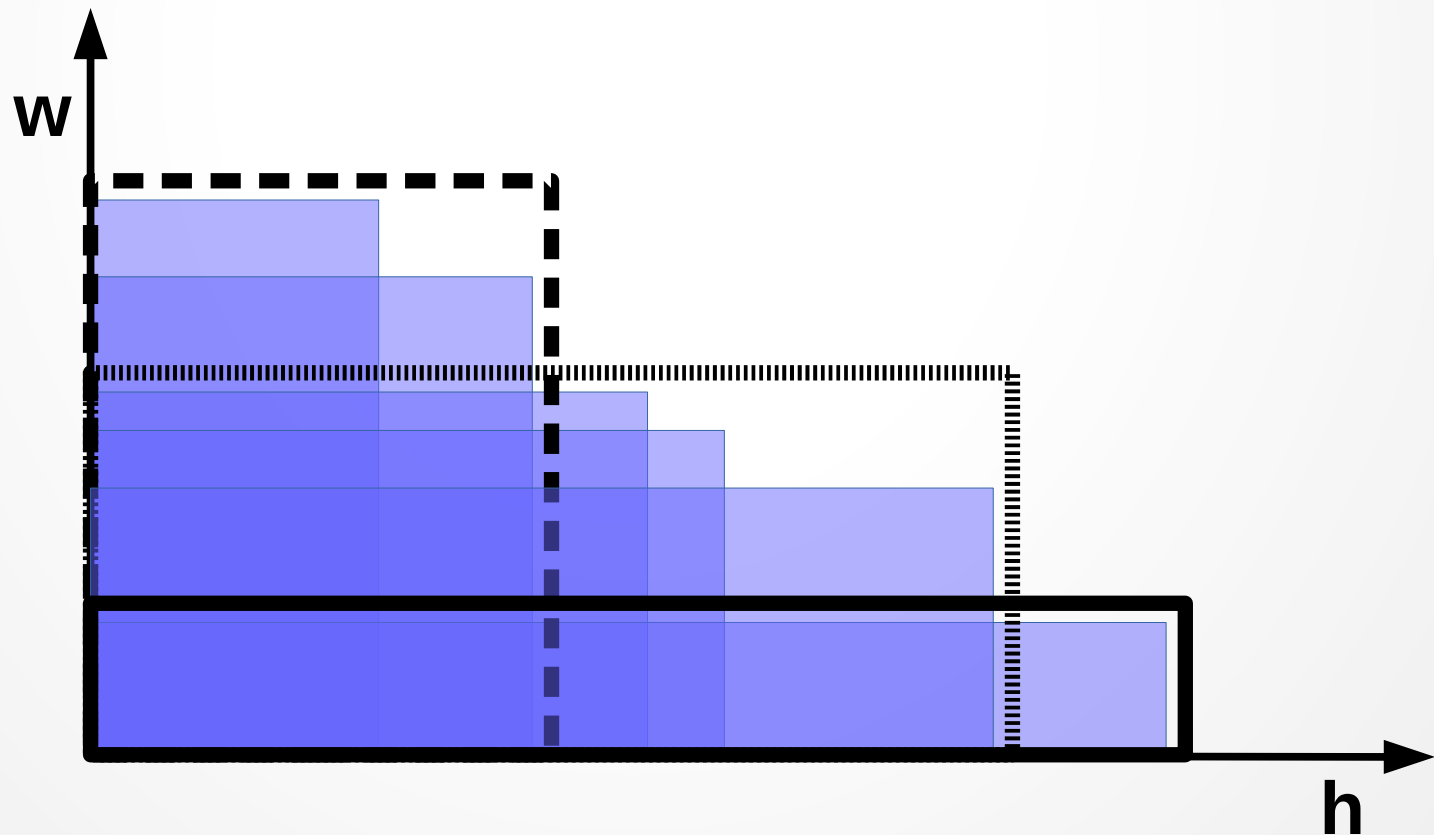
Kenapa?

- Karena jika ada $i < j$, lalu $w[i] < w[j]$ dan $h[i] \leq h[j]$, maka $w[i]*h[i]$ harusnya sudah diserap oleh $w[j]*h[j]$

Jadi tidak mungkin ada kasus begitu

Observasi 3...

- Untuk meminimalkan harga, pengelompokkan pasti secara konsekutif (jika sudah sorted berdasarkan w atau h)



Solusi DP $O(N^2)$

- Berdasarkan 3 observasi itu, bisa dibuat DP:

$$dp(u) = \min_{1 \leq i \leq u} (h[i] * w[u] + dp(i-1))$$

- Dengan $dp(u)$ = biaya minimal pembelian tanah 1, 2, 3, ..., u
- Tentu saja $dp(0) = 0$
- Ingat bahwa:
 - $w[1] < w[2] < w[3] < \dots$
 - $h[1] > h[2] > h[3] > \dots$

Sudut Pandang Berbeda

- Modelkan setiap fungsi biaya sebagai fungsi garis:

$$dp(u) = \min_{1 \leq i \leq u} (h[i] * w[u] + dp(i-1))$$

$$dp(u) = \min_{1 \leq i \leq u} (m_i x + c_i)$$

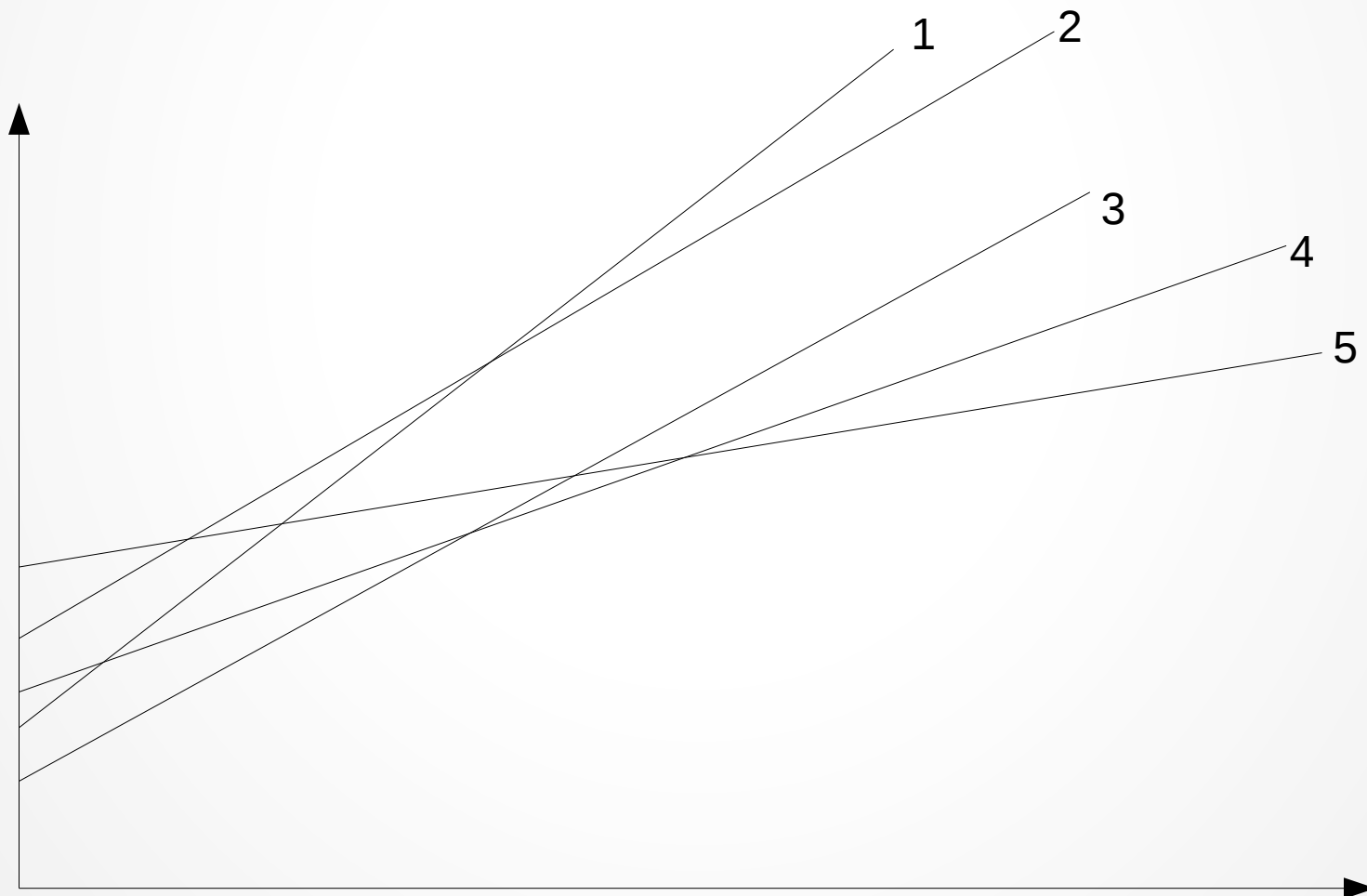

Sudut Pandang Berbeda (lanj.)

$$dp(u) = \min_{1 \leq i \leq u} (m_i x + c_i)$$

- Kini masalah yang dihadapi adalah:
 - Diberikan sejumlah persamaan garis, dari yang ke-1 sampai ke-(x-1)
 - Persamaan garis ke-i punya gradien m_i dan konstanta c_i
 - Cari nilai y terkecil untuk suatu nilai x !

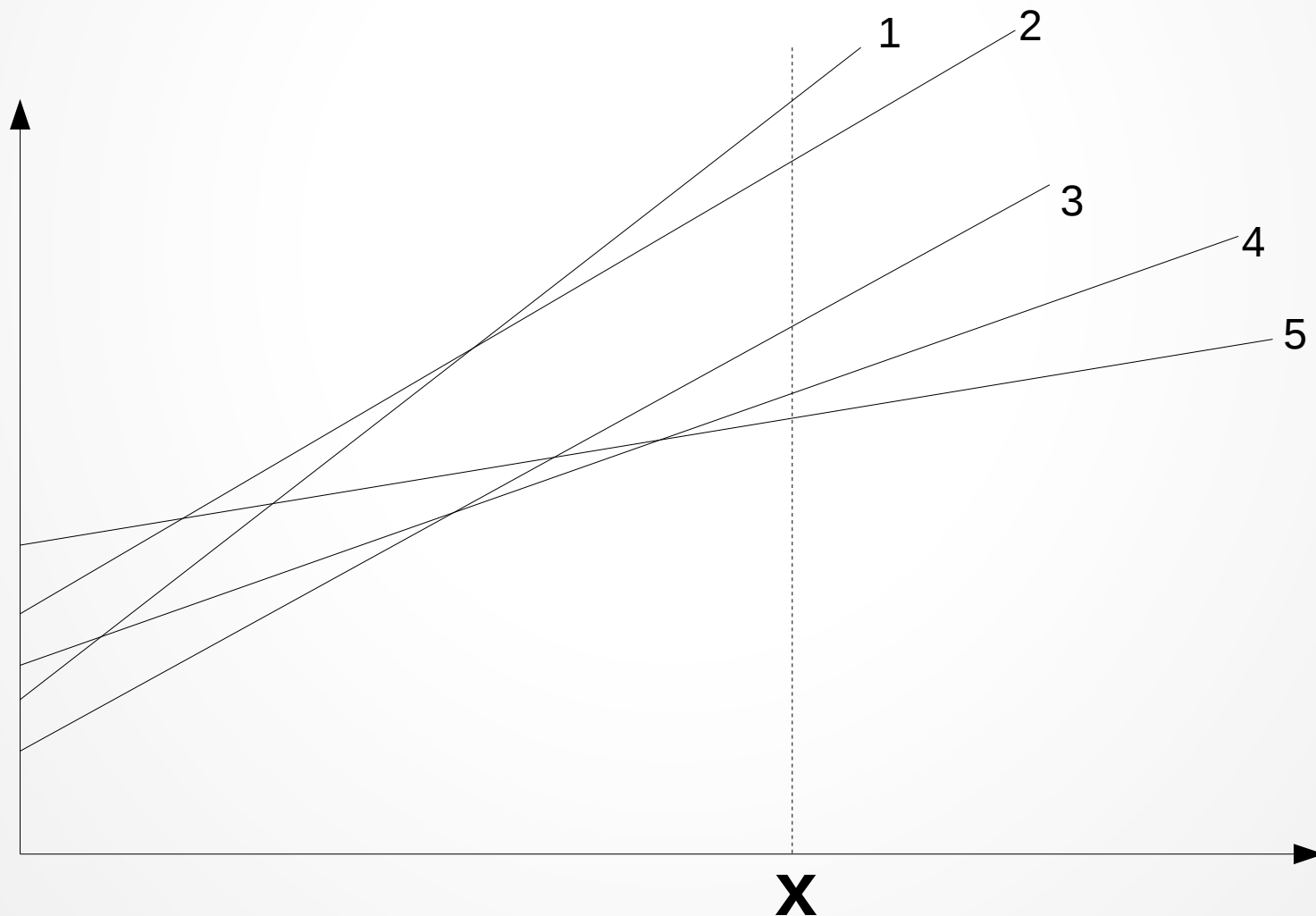
Sudut Pandang Berbeda (lanj.)

$$dp(u) = \min_{1 \leq i \leq u} (m_i x + c_i)$$



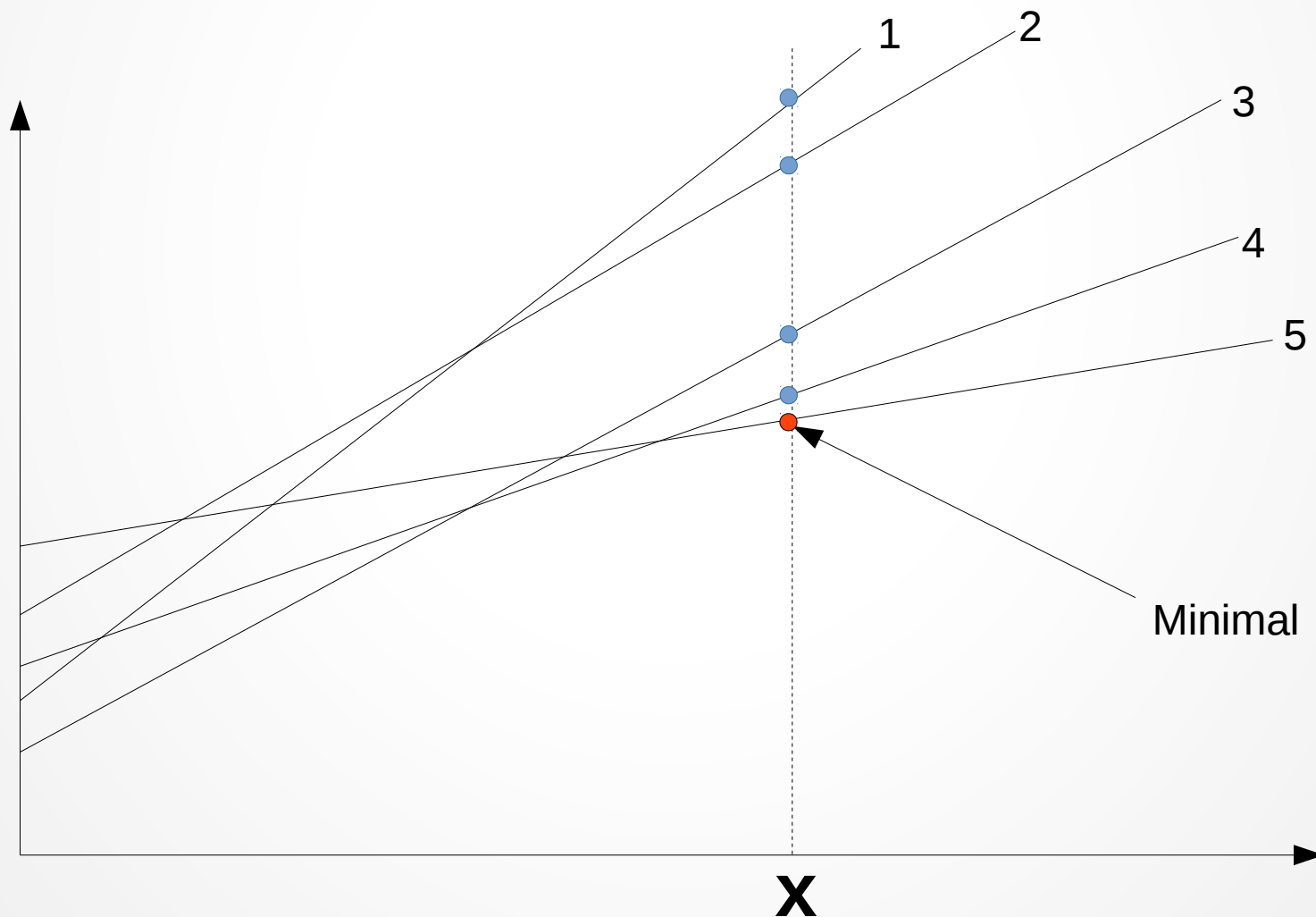
Sudut Pandang Berbeda (lanj.)

$$dp(u) = \min_{1 \leq i \leq u} (m_i x + c_i)$$



Sudut Pandang Berbeda (lanj.)

$$dp(u) = \min_{1 \leq i \leq u} (m_i x + c_i)$$

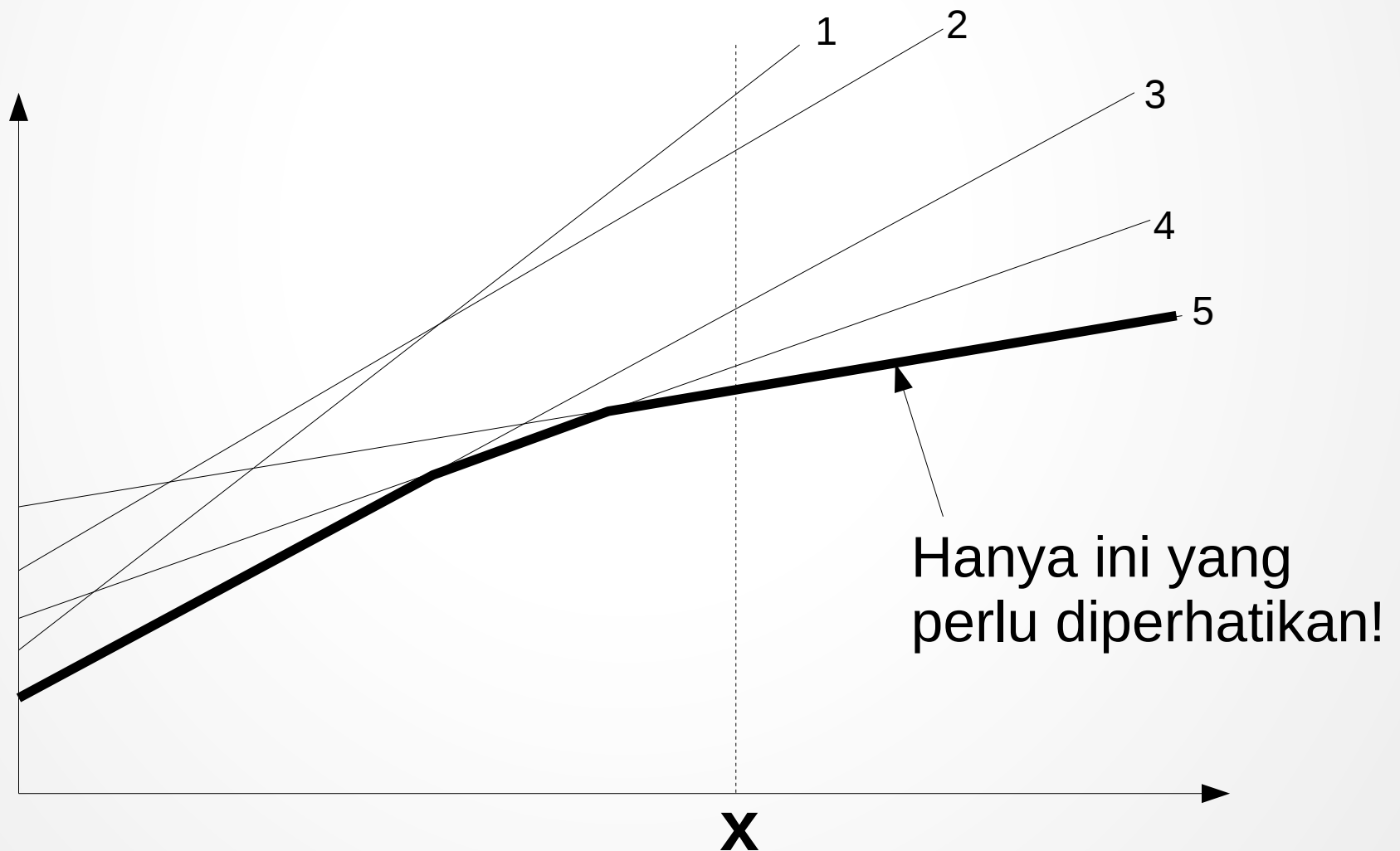


Apakah Lebih Efisien?

- Iterasi satu per satu pada seluruh persamaan garis sama saja $O(N^2)$, seperti solusi DP sebelumnya
- Tapi, sebenarnya hanya beberapa garis saja yang perlu diperhatikan

Garis-Garis penting

$$dp(u) = \min_{1 \leq i \leq u} (m_i x + c_i)$$

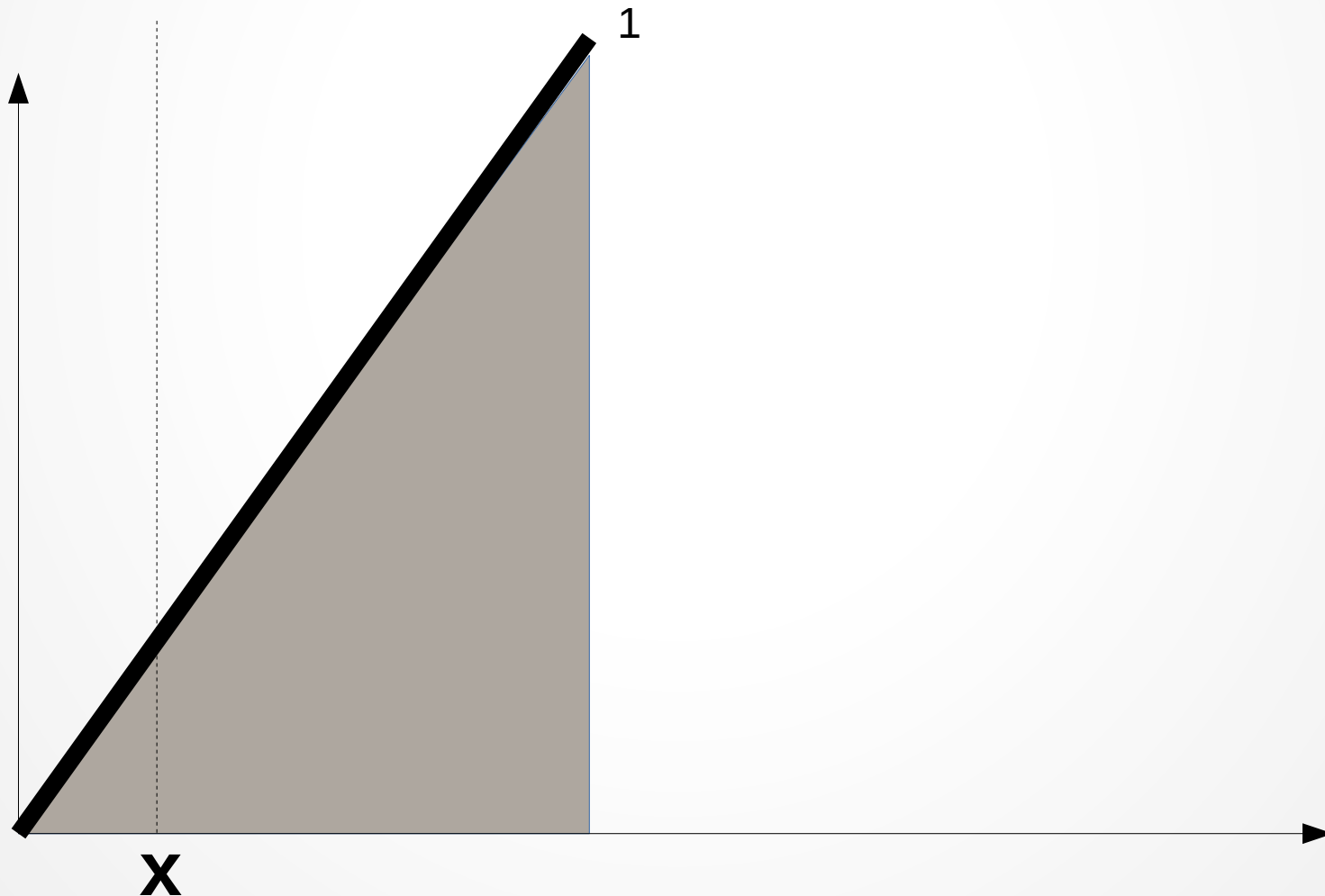


Observasi 4...

- Ketika mengisi $dp(1)$, $dp(2)$, $dp(3)$, ..., nilai x yang diuji untuk pencarian nilai y minimum selalu meningkat, sementara gradien garis selalu menurun.
- Ingat bahwa pada $dp(u)$:
 $x = w[u]$ dan $w[1] < w[2] < w[3] < \dots$
 $m_i = h[i]$ dan $h[1] > h[2] > h[3] > \dots$
- Kita bisa memelihara “convex hull” sambil mengisi tabel DP
- Istilah lebih tepat \rightarrow lower envelope

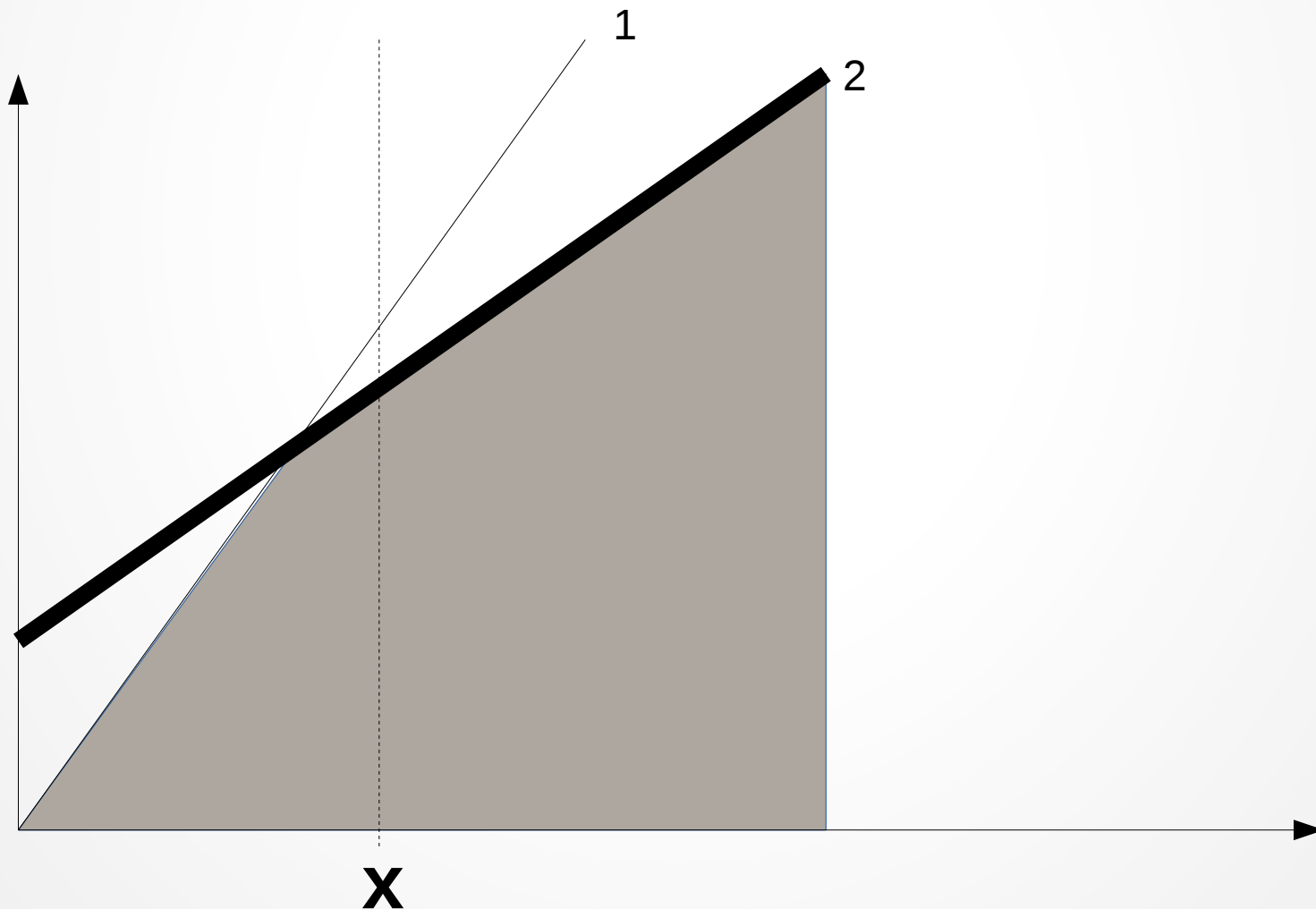
Contoh Eksekusi

- $dp(1)$



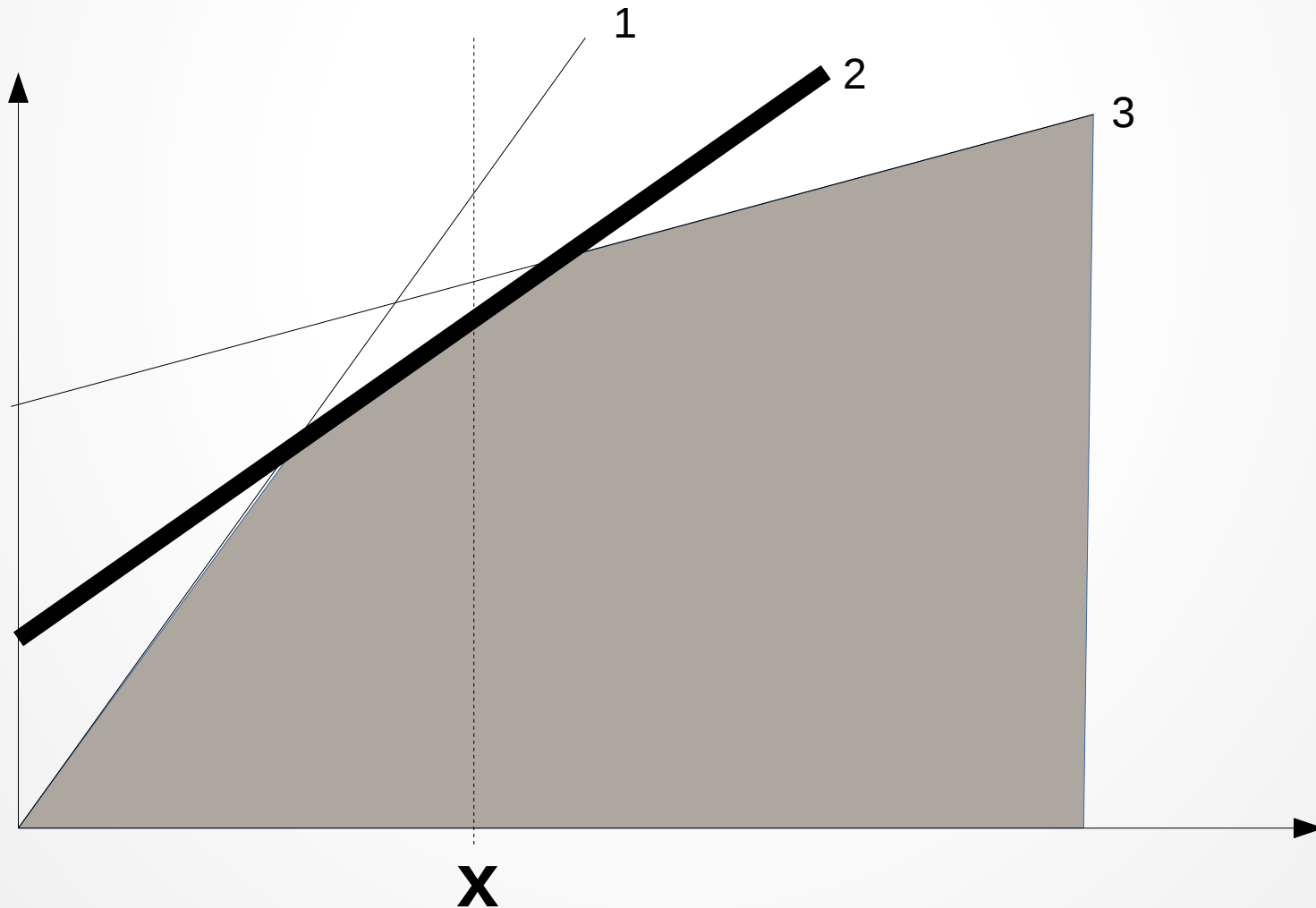
Contoh Eksekusi

- $dp(2)$



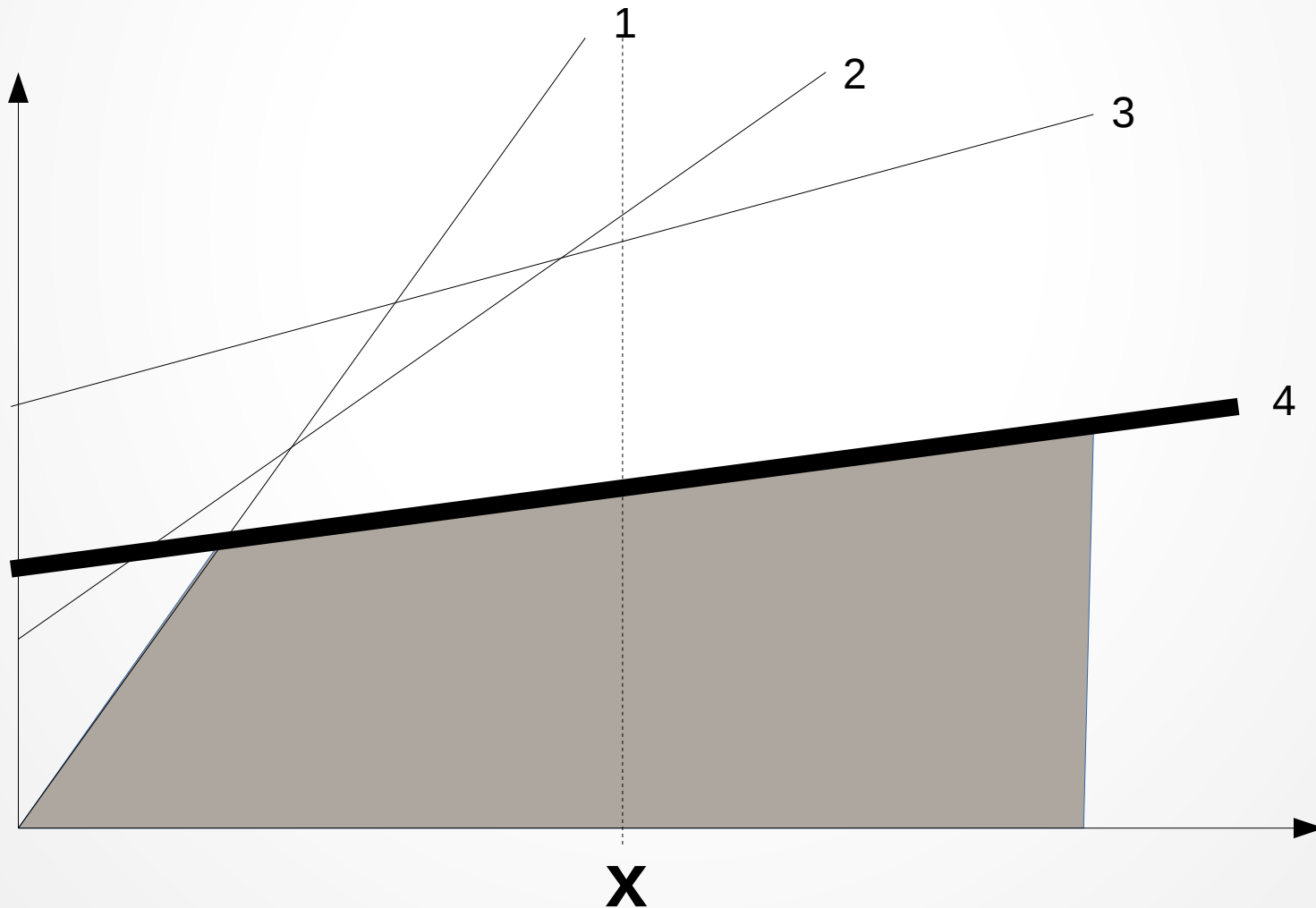
Contoh Eksekusi

- $dp(3)$



Contoh Eksekusi

- $dp(4)$



Observasi 5...

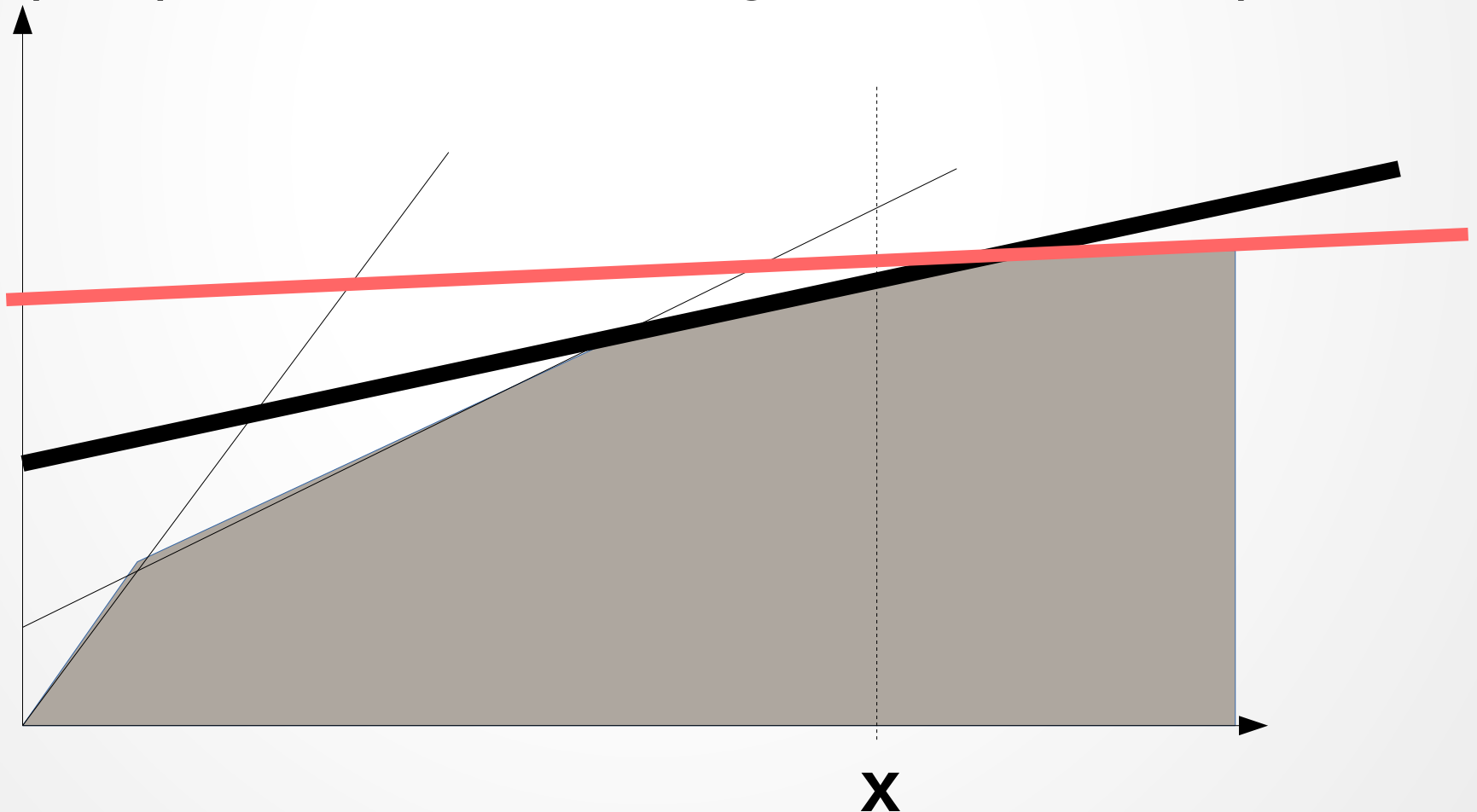
- Lower envelope terdiri dari garis-garis dengan gradien menurun

Observasi 6...

- Setiap sebelum mengisi $dp(u)$, persamaan garis yang baru ditambahkan dan lower-envelope bisa jadi berubah

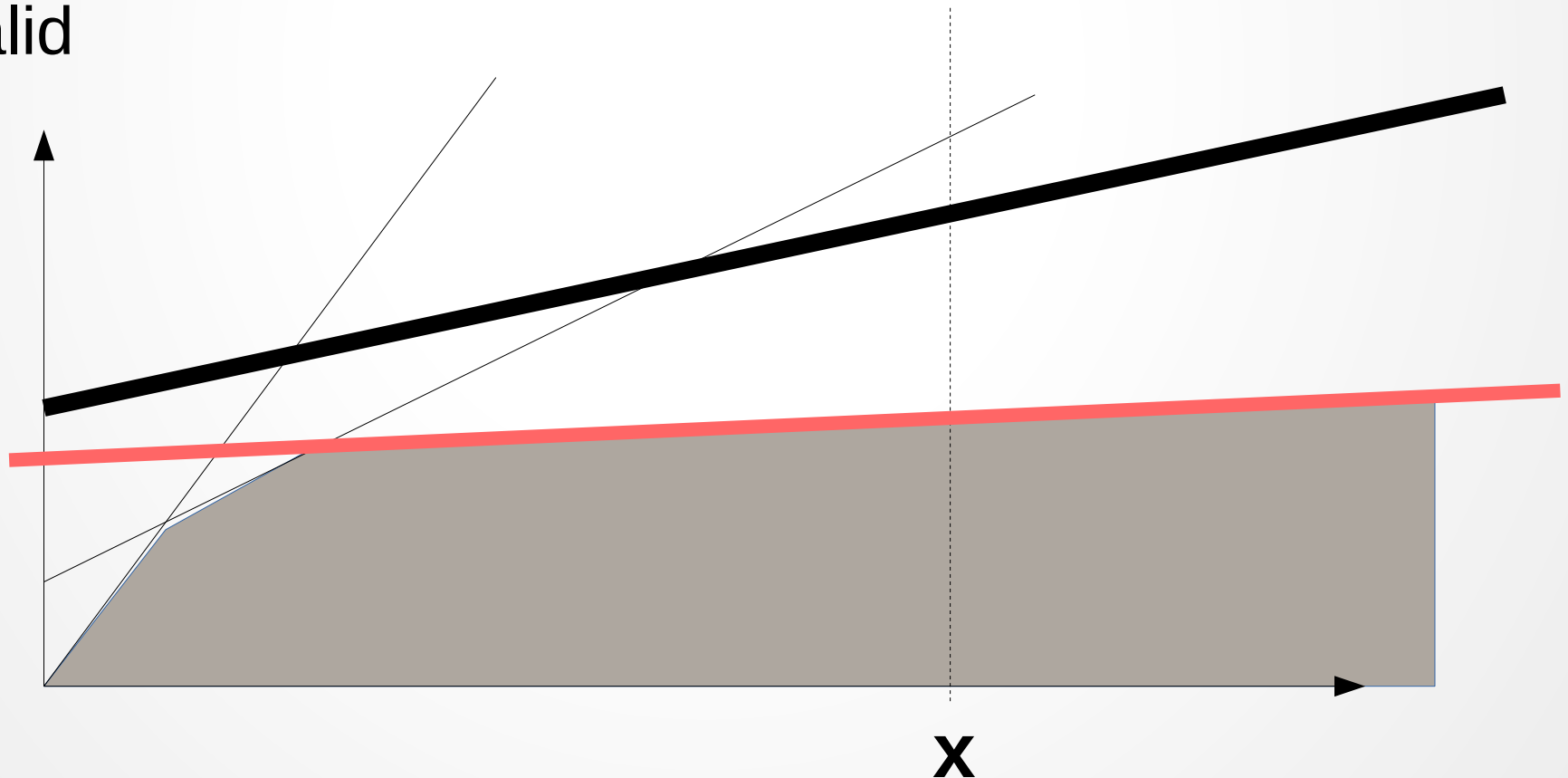
Gradien lebih Kecil

- Persamaan garis itu mungkin menjadi bagian dari lower-envelope pada masa depan
- Simpan pada daftar calon bagian lower-envelope



Gradien lebih Kecil (lanj.)

- Tapi tidak selalu demikian, bisa saja garis yang baru ini “memakan” lower-envelope yang sudah terbentuk
- Cara yang lebih tepat adalah melakukan “pop” pada bagian lower-envelope terkanan sampai bentuknya valid



Gradien Lebih Kecil (lanj.)

- Jadi untuk menyimpan daftar garis-garis yang tergabung pada lower-envelope, bisa digunakan stack yang monotonic decreasing (pada gradien garisnya)
- Setiap mau “push” garis baru, “pop” dulu garis-garis di atas stack yang tidak valid (sudah tidak lagi tergabung pada lower-envelope)

Implementasi

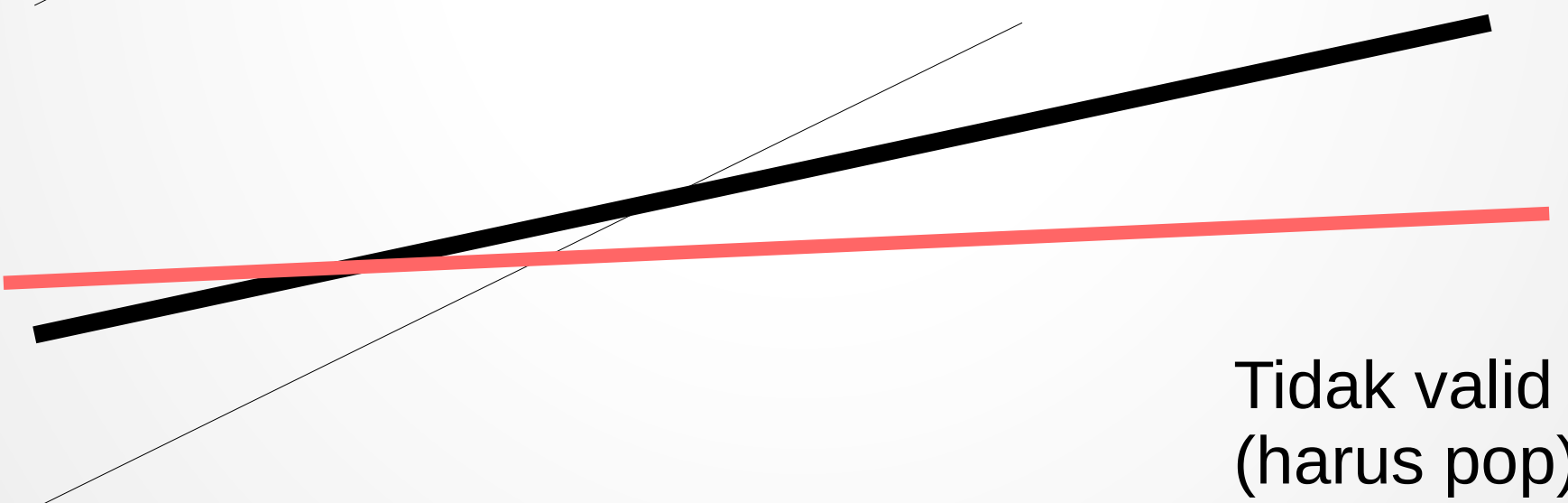
- Bagaimana cara menentukan apakah top-of-stack pada lower-envelope masih valid ketika akan ditambahkan garis baru?

Implementasi (lanj.)



The diagram illustrates a valid implementation of a linear transformation. It features a thick black line representing the transformation, which is parallel to a thin red line representing the identity transformation. A thin grey line, representing the transformation matrix, is shown intersecting the black line at a point. The word "Valid" is written to the right of the black line.

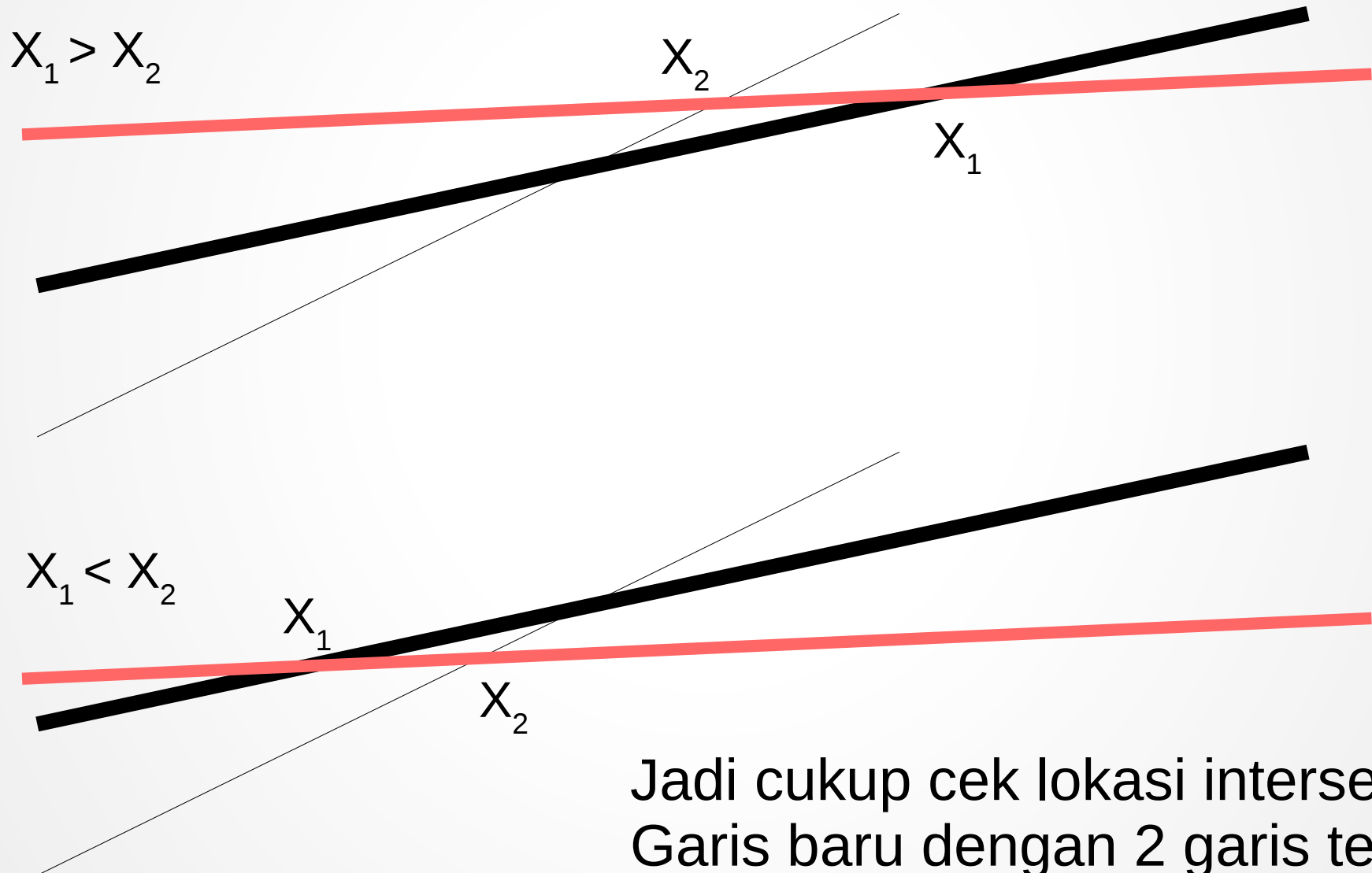
Valid



The diagram illustrates an invalid implementation of a linear transformation. It features a thick black line representing the transformation, which is not parallel to a thin red line representing the identity transformation. A thin grey line, representing the transformation matrix, is shown intersecting the black line at a point. The text "Tidak valid (harus pop)" is written to the right of the black line.

Tidak valid
(harus pop)

Implementasi (lanj.)



Jadi cukup cek lokasi intersection
Garis baru dengan 2 garis teratas
pada stack

DP $O(N)$

- Isi tabel DP mulai dari 1, 2, 3, ..., sampai N, sambil memelihara lower-envelope
- Kompleksitas DP saja: $O(N)$
- Ditambah sorting di awal menjadi $O(N \log N)$

Convex Hull Trick

- Sifat untuk “hanya perlu memperhatikan lower-envelope” biasa disebut sebagai convex hull trick
- Varian lain adalah memperhatikan upper-envelope, atau cara melakukan update pada envelope-nya berbeda

Tips Mengerjakan Soal

- Jika ada soal DP yang butuh optimasi, coba pikirkan convex hull trick sebagai alternatif
- Cari tahu mana yang bisa dimodelkan sebagai gradien, konstanta, dan variabel x
 - Variabel x biasanya berasosiasi dengan parameter/state pada DP
 - Gradien merupakan pengali dari x
 - Konstanta merupakan nilai konstan lainnya yang ditambahkan ke variabel x