

Pengenalan Analisis dan Strategi Algoritma

Sukmawati Nur Endah

Departemen Informatika UNDIP

Pentingnya Mata Kuliah

- ▶ Orang komputer harus menguasai Algoritma
- ▶ Harus mampu mengembangkan sesuai dengan masalah yang ada
- ▶ Tidak saja harus benar, tetapi juga harus efisien baik dari waktu maupun ruang memori
- ▶ Goal : mahasiswa mampu memilih dan mengembangkan algoritma yang tepat dengan mempertimbangkan keefisiensinya (kemangkusannya)

Apakah Algoritma itu??



Definisi Algoritma

- ▶ Urutan langkah-langkah dalam menyelesaikan masalah secara sistematis
- ▶ Langkah-langkah yang mentransformasikan dari input menjadi output
- ▶ Tiga hal dalam mendefinisikan algoritma
 - ▶ Masalah : Persoalan yang akan diselesaikan
 - ▶ Input : Contoh data atau keadaan yang jadi masalah
 - ▶ Output : Hasil akhir dari data atau keadaan sesudah algoritma diimplementasikan pada input
- ▶ Hasil akhir dianggap sebagai penyelesaian masalah



Contoh Mendefinisikan Algoritma

- ▶ Masalah : Pengurutan sekumpulan nilai secara acak
- ▶ Input : Serangkaian data berukuran n
 - ▶ Misal : [4,3,1,2,5,6] → bentuk numerik
[“Dita”, “Alex”, “Eko”, “Chandra”, “Bintang”, “Fifi”]
→ bentuk string
- ▶ Output : Serangkaian data berukuran n yang terurut dengan urutan $a_1 \leq a_2 \leq a_3 \leq \dots \leq a_{n-1} \leq a_n$
 - ▶ Misal : [1,2,3,4,5,6]
[“Alex”, “Bintang”, “Chandra”, “Dita”, “Eko”, “Fifi”]

Solusi Algoritma

- Dapat menggunakan Algoritma Insertion Sort
- Implementasi dengan bahasa pemrograman Python

```
def insertion_sort(data):  
    for i in range(0, len(data)):  
        insert_val = data[i]  
        hole_pos = i  
  
        while hole_pos > 0 and insert_val < data[hole_pos - 1]:  
            data[hole_pos] = data[hole_pos - 1]  
            hole_pos = hole_pos - 1  
  
        data[hole_pos] = insert_val
```

Bagaimana menentukan Algoritma itu baik?



► Benar

- algoritma dapat menyelesaikan masalah dengan tepat, sesuai dengan definisi masukan / keluaran algoritma yang diberikan.

► Efisien

- algoritma dalam menyelesaikan masalah tidak memberatkan bagian lain dari aplikasi.
- Sebuah algoritma yang tidak efisien biasanya menggunakan sumber daya (memori, CPU) yang besar dan waktu komputasi yang tinggi.

► Mudah diimplementasikan

- sebuah algoritma yang baik harus dapat dimengerti dengan mudah sehingga implementasi algoritma dapat dilakukan siapapun dengan pendidikan yang tepat, dalam waktu yang masuk akal.

Algoritma yang Baik

- ▶ Pada prakteknya tiga hal tersebut tidak dapat selalu tercapai
- ▶ Biasanya di syarat kedua dan ketiga yang tidak selalu didapatkan
- ▶ Sebagai programmer wajib mengusahakan ketiganya

Bagaimana membuktikan Algoritma itu benar?



- ▶ Sebuah algoritma dikatakan telah benar jika :
 - ▶ algoritma tersebut dapat **memberikan keluaran yang benar** jika menerima masukan sesuai dengan definisi algoritma tersebut
 - ▶ algoritma tersebut terbukti akan **selalu dapat diterminasi (berakhir)**

- ▶ Pembuktikan kebenaran:
 - ▶ Induksi Matematika,
 - ▶ Pembuktian kontradiktif,
 - ▶ Pembuktian kontrapositif, dan
 - ▶ Metode Formal

- ▶ Masing-masing metode mempunyai kelebihan dan kekurangannya serta kasus penggunaan yang berbeda-beda

Induksi Matematika

- ▶ Merupakan alat pembuktian matematis yang digunakan untuk membuktikan pernyataan atau proses yang melibatkan perhitungan bilangan asli yang berulang.
 - ▶ Contoh : perhitungan deret aritmatika, deret geometris, ataupun sigma bilangan
- ▶ Pembuktian menggunakan induksi matematika dilakukan dengan dua langkah, yaitu:
 - ▶ Melakukan pembuktian kasus dasar (*base case*), yaitu membuktikan bahwa sebuah pernyataan (fungsi) matematika atau algoritma bernilai benar jika diaplikasikan pada bilangan pertama yang sah sesuai dengan spesifikasi fungsi atau algoritma tersebut.
 - ▶ Melakukan induksi, yaitu membuktikan bahwa kebenaran dari fungsi $P(k+1)$ jika kebenaran fungsi $P(k)$ diketahui.

Contoh pembuktian dengan induksi matematika

► Contoh 1. Deret Aritmatika

- Buktikan bahwa deret aritmatika $1+2+3+\dots+n=n(n+1)/2$ adalah benar untuk semua bilangan bulat $n \geq 1$

► Contoh 2. Pembuktian Hipotesa

- Buktikan hipotesa bahwa fungsi matematika $n^3 - n$ habis dibagi 6 untuk semua bilangan bulat $n \geq 2$.

Pembuktian contoh 1

- ▶ Ubah pernyataan matematika tersebut menjadi sebuah fungsi matematika:
 - ▶ $P(k)=1+2+3+\dots+n=k(k+1)/2$
- ▶ Pembuktian kasus dasar
 - ▶ Karena pernyataan matematika pada soal menyatakan bahwa pernyataan benar untuk semua bilangan bulat $k \geq 1$, maka untuk pembuktian kasus dasar harus membuktikan bahwa $P(1)$ adalah benar untuk ruas kiri maupun ruas kanan dari $P(k)$.

$$P(1)=1=1(1+1)/2$$

$$1=1(2)/2$$

$$1=2/2$$

$$1=1$$

- ▶ karena hasil akhir dari ruas kanan dan ruas kiri adalah sama (1), maka dapat dikatakan bahwa kasus dasar telah terbukti

Lanj. Pembuktian Contoh 1

► Induksi

- Untuk pembuktian induksi, kita harus membuktikan bahwa $P(k) \rightarrow P(k+1)$ bernilai benar.
- Langkah pertama : tuliskan fungsi matematis dari $P(k+1)$ terlebih dahulu:

$$P(k+1) = 1 + 2 + \dots + k + (k+1) = \frac{(k+1)((k+1)+1)}{2}$$

- Pembuktian akan dilakukan dengan melakukan penurunan pada ruas kiri agar menjadi sama dengan ruas kanan:

$$\begin{aligned} 1 + 2 + \dots + k + (k+1) &= (1 + 2 + \dots + k) + (k+1) \\ &= \frac{k(k+1)}{2} + (k+1) \\ &= \frac{k(k+1) + 2(k+1)}{2} \\ &= \frac{k^2 + 3k + 2}{2} \\ &= \frac{(k+1)(k+2)}{2} \\ &= \frac{(k+1)((k+1)+1)}{2} \end{aligned}$$

ruas kiri dari $P(k+1)$ telah sama dengan ruas kanannya, sehingga dapat dikatakan bahwa tahap induksi telah berhasil dibuktikan benar.

- Dengan pembuktian kasus dasar dan induksi yang bernilai benar, dapat disimpulkan bahwa $P(n)$ bernilai benar untuk $n \geq 1$

Latihan

- Buatlah pembuktian dengan induksi matematis untuk contoh 2

Bagaimana induksi matematika dapat membuktikan algoritma?

- ▶ Karena dalam algoritma biasanya melakukan perhitungan bilangan atau data secara berulang
- ▶ Contoh : Algoritma menghitung perkalian dua buah bilangan bulat positif

```
def kali(m, n):  
    if m < 0:  
        return -1 # error  
    else:  
        i = 0  
        result = 0  
  
        while(m != i):  
            result = result + n  
            i = i + 1  
  
        return result
```

Lanjutan

- Permasalahan diatas dapat dituliskan :

$$f(m, n) = \sum_{i=1}^n m; n \geq 0$$

$$m \times n = \underbrace{m + m + m + \dots + m}_{n \text{ kali}}$$

- Hal ini dapat dibuktikan dengan induksi matematika

Pemodelan masalah

- ▶ Sebuah algoritma dapat dituliskan menjadi fungsi matematika
- ▶ Algoritma maupun fungsi matematika adalah sebuah *model*,
 - ▶ digunakan untuk menggambarkan masalah yang ditemui pada dunia nyata, dan ingin diselesaikan, baik dengan menggunakan matematika ataupun program komputer.
- ▶ Dengan memiliki model masalah:
 - ▶ lebih mudah mengerti masalah yang akan diselesaikan,
 - ▶ akan menyebabkan solusi yang ditawarkan menjadi lebih baik.
- ▶ Pertanyaannya :
 - ▶ Bagaimana membuat model yang benar dari masalah-masalah yang ada?

Jenis-jenis model

- ▶ Model Numerik
- ▶ Model Simbolik
- ▶ Model Spasial
- ▶ Model Logis
- ▶ Model Statistik
- ▶ Model Pseudocode

Model Numerik

- Untuk mendeskripsikan jumlah atau ukuran dari sesuatu
- Model numerik menggunakan angka (1, 2, 3, dst) untuk mendeskripsikan suatu hal
- Misal:



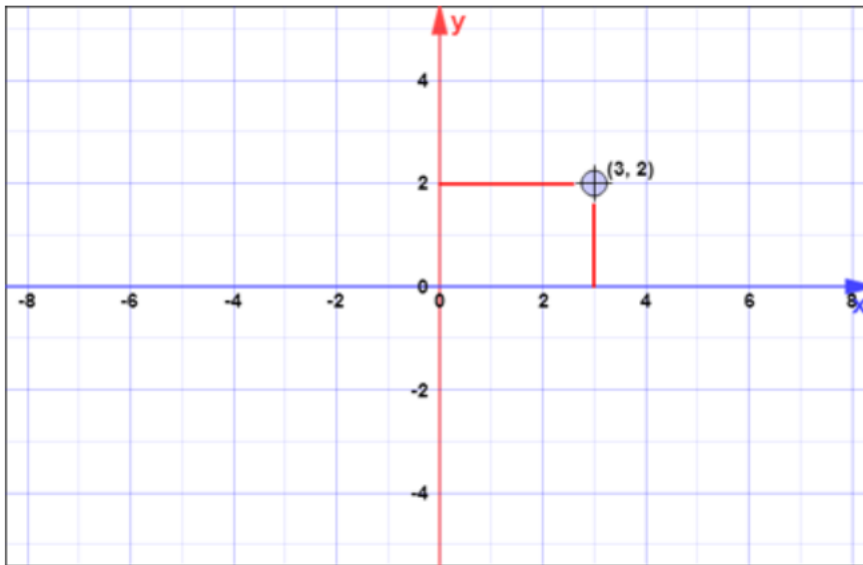
Model Numerik Sapi

Model Simbolik

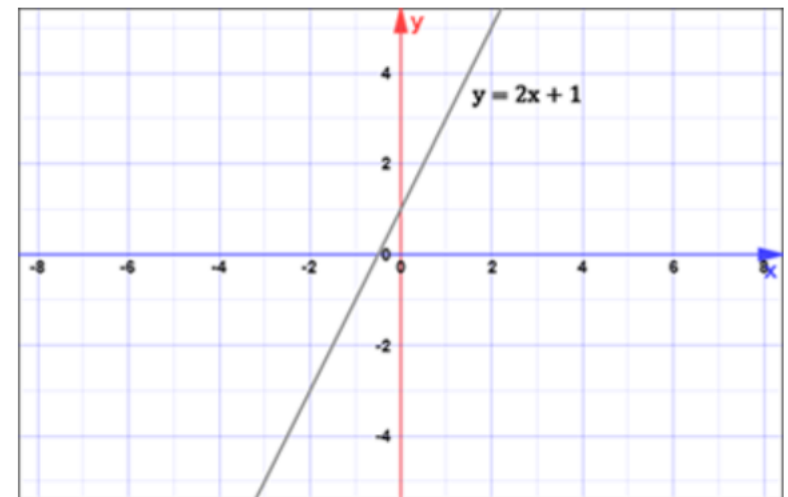
- ▶ Terdapat empat buah simbol dasar untuk pemrosesan angka, yaitu $+$, $-$, \times atau $*$, dan \div atau $/$
- ▶ Simbol = juga digunakan untuk menandakan kesamaan nilai antara ruas kiri dan ruas kanan
 - ▶ Misal : $15 = 5 * 3$
- ▶ Aturan-aturan umum untuk operator numerik yaitu:
 - ▶ Hukum Kumulatif, di mana $a+b=b+a$ dan $a*b=b*a$.
 - ▶ Hukum Asosiatif, di mana $a+(b+c)=(a+b)+c$ dan $a*(b*c)=(a*b)*c$.
 - ▶ Hukum Distribusi, di mana $a*(b+c)=(a*b)+(a*c)$.
 - ▶ Hukum Invers, yaitu $a+(-a)=0$ dan $a * 1/a = 1$.
 - ▶ Hukum Identitas, yaitu $a+0=a$ dan $a*1=a$.
 - ▶ Perkalian dengan 0, yaitu $a*0=0$.
- ▶ Dalam model simbolik juga dikenal namanya variabel dan konstanta. Misal:
 - ▶ Luas Segitiga = $\frac{1}{2} * a * t$
 - ▶ Variabel : a dan t , konstanta : $\frac{1}{2}$

Model spasial

- Digunakan untuk masalah-masalah yang berhubungan dengan representasi dunia nyata seperti perhitungan jarak dua objek atau pencarian jalur terdekat untuk kendaraan.
- Dapat digambarkan dengan peta, graph, dan gambar-gambar teknis lainnya

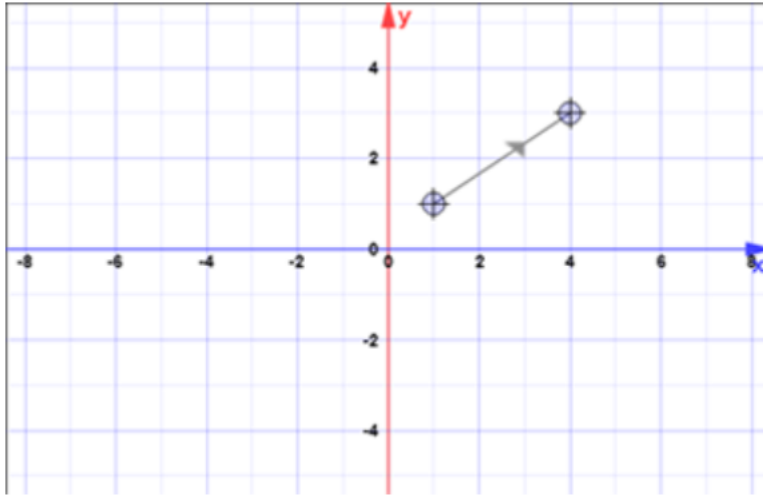


Titik pada Kartesius

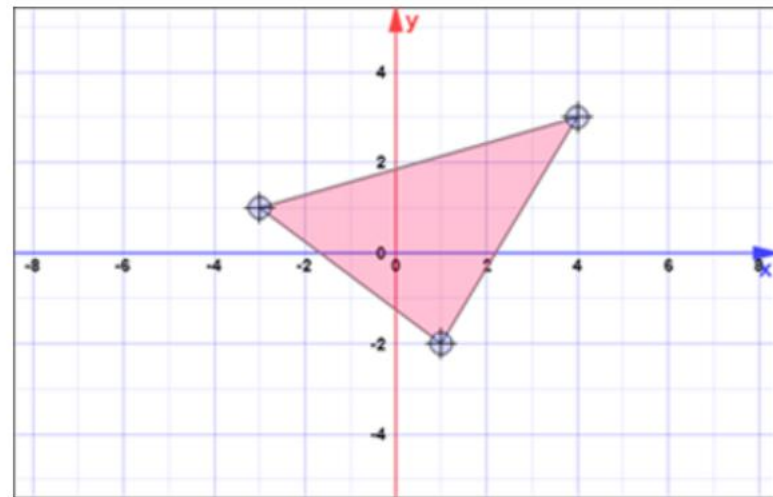


Garis pada Kartesius

Model spasial



Garis Berarah pada Kartesius



Bidang pada Kartesius

gambar di atas dapat direpresentasikan sebagai matriks berikut:

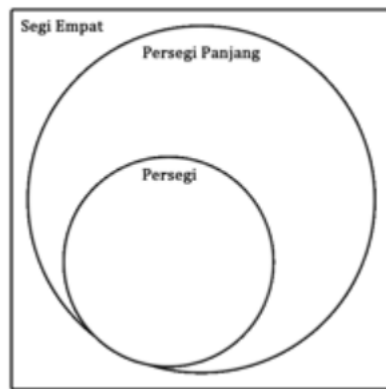
$$\begin{bmatrix} -3 & 4 & 1 \\ 1 & 3 & -2 \end{bmatrix}$$

Model Logis

- ▶ Model logis merupakan cara memodelkan masalah berdasarkan logika matematika.
- ▶ Terdapat empat cabang utama dari logika matematika, yaitu teori himpunan, teori model, teori rekursif, dan teori pembuktian.
- ▶ Masing-masing teori memiliki cara pemodelan yang berbeda-beda, untuk merepresentasikan masalah yang berbeda.
 - ▶ Misal untuk teori himpunan dapat dimodelkan untuk masalah basis data
 - ▶ masalah dapat digambarkan dalam bentuk diagram Venn (termasuk atribut atau bukan)

Model Logis

- Contoh diagram Venn



Contoh Diagram Venn

- Bagaimana jika ditambahkan dengan trapesium?

Model Statistik

- ▶ model statistik digunakan untuk menganalisa tren terhadap sampel data yang relevan untuk meniadakan ketidak pastian atau keadaan khusus.
- ▶ Dengan mengambil keadaan rata-rata dari sekumpulan data, didapatkan **kecenderungan** dari sebuah keadaan jika dihadapkan dengan keadaan umumnya.
- ▶ Misal : Prakiraan cuaca

Model Pseudocode

- ▶ Pseudocode memberikan langkah-langkah penyelesaian masalah dengan menggunakan bahasa manusia, dengan sedikit batasan sesuai dengan konstruk logika komputer.
- ▶ Pseudocode tidak memiliki konstruk untuk bahasa pemrograman tertentu, sehingga pseudocode harus bisa diimplementasikan dengan bahasa pemrograman apapun.
- ▶ Contoh pseudocode sederhana:

```
for i = 1 to 5 do  
    print i  
end for
```

Pembangunan model

- ▶ Apakah masalah yang dihadapi merupakan masalah yang memerlukan solusi matematis?
 - ▶ Jika masalahnya merupakan masalah numerik (perhitungan angka) atau logis, maka jawabannya sudah pasti “ya”.
 - ▶ Jika solusi dari masalah berupa *pendapat*, maka kemungkinan jawabannya adalah “tidak”.
- ▶ Fakta-fakta relevan apa saja yang diketahui?
 - ▶ Masalah umum yang dihadapi saat akan membangun solusi adalah informasi yang *terlalu banyak*, yang terkadang mencuri fokus kita dari akar masalah.
 - ▶ Pisahkan antara fakta (informasi) yang relevan dari keseluruhan informasi yang didapatkan.
- ▶ Fakta atau informasi tambahan apa yang kita perlukan untuk menyelesaikan masalah? Di mana atau bagaimana cara agar kita mendapatkan fakta-fakta tersebut?

Lanj. Pembangunan Model

- ▶ Adakah langkah atau metode *alami* untuk menyelesaikan masalahnya?
 - ▶ Metode alami artinya metode yang umumnya digunakan oleh manusia. Misalnya, untuk menghitung total dari sekumpulan nilai, kita dapat menambahkan seluruh bilangan yang ada di dalam kumpulan nilai tersebut.
- ▶ Apakah fakta-fakta yang ada dapat direpresentasikan oleh simbol matematis dan dikategorikan menjadi fakta yang “diketahui” dan “tidak diketahui”?
- ▶ Apakah terdapat model *lama* yang dapat digunakan atau disesuaikan untuk menyelesaikan masalah kita?
- ▶ Jika terdapat model yang telah dikembangkan sebelumnya untuk masalah kita, apakah model tersebut dapat diaplikasikan pada komputer?
- ▶ Bagaimana kita dapat mengaplikasikan model dari solusi kita sehingga model tersebut dapat dibuat menjadi program komputer dengan mudah?

Contoh kasus :Perhitungan bunga pinjaman

- ▶ Sebuah perusahaan kredit ABC ingin membuat sebuah program komputer yang mengembangkan sistem untuk menghitung total jumlah yang harus dibayar oleh peminjam uang per tahunnya.
- ▶ Bunga pinjaman yang diberikan ABC adalah sebesar 15% per tahunnya.
- ▶ Bagaimana membangun modelnya?

Langkah-langkah pembangunan model

- ▶ **Apakah masalah yang dihadapi merupakan masalah yang memerlukan solusi matematis?**
 - ▶ Ya. Perhitungan total bunga bunga jelas akan melibatkan matematika.
- ▶ **Fakta-fakta relevan apa saja yang diketahui?**
 - ▶ Bunga pinjaman sebesar 15% per tahun.
- ▶ **Fakta atau informasi tambahan apa yang diperlukan untuk menyelesaikan masalah?**
 - ▶ Beberapa fakta tambahan yang *harus ada* tetapi tidak disebutkan secara eksplisit pada deskripsi masalah:
 - ▶ Jumlah pinjaman awal. Untuk menghitung total pinjaman dengan bunganya jelas kita harus mengetahui jumlah pinjaman awal terlebih dahulu.
 - ▶ Lama pinjaman. Tanpa adanya lama pinjaman, kita tidak dapat mengetahui dengan pasti total bunga yang harus ditambahkan.
- ▶ **Adakah langkah atau metode alami untuk menyelesaikan masalahnya?**
 - ▶ Ya, lakukan perhitungan bunga tiap tahunnya, dan tambahkan hasil kalkulasi tersebut sampai tahun pinjaman terakhir

Langkah-langkah pembangunan model

- ▶ Apakah fakta-fakta yang ada dapat direpresentasikan oleh simbol matematis?
 - ▶ Dari fakta-fakta yang kita dapatkan pada langkah kedua dan ketiga, kita dapat mendefinisikan simbol matematis seperti berikut:
 - let b =bunga
 - let p =jumlah pinjaman
 - let t =waktu pinjaman (per tahun)
 - let T =total pinjaman
- ▶ Apakah terdapat model lama yang dapat digunakan untuk menyelesaikan masalah?
 - ▶ Ya, perhitungan bunga majemuk yang dimodelkan dengan rumus: $T=p(1+b)^t$

Langkah-langkah pembangunan model

- ▶ Apakah model yang ada sebelumnya pada langkah 6 dapat diaplikasikan pada komputer?
 - ▶ Kemungkinan tidak, karena perhitungan bunga majemuk merupakan perhitungan yang tidak banyak diketahui orang (terutama pada bidang *pemrograman*), dan juga memiliki banyak aturan kompleks yang harus dimengerti terlebih dahulu.
 - ▶ Karena kasus yang sederhana, dapat dikembangkan algoritma sendiri dengan iterasi.
 - ▶ Untuk tahun pertama, peminjam akan berhutang sebanyak:
$$T = p + (15\% * p)$$
 - ▶ untuk tahun kedua hutangnya akan bertambah menjadi:
$$T = T + (15\% * T)$$
 - ▶ di mana T adalah nilai baru dari T . Jika dikembangkan, maka model matematis akhir yang kita dapatkan adalah:
$$T = T + (15/100 * T)$$
yang akan dijalankan sebanyak \$t\$ kali, dengan nilai \$T\$ yang bertambah setiap iterasinya.

Langkah-langkah pembangunan model

- Implementasikan dalam bentuk pseudocode

```
b = 15
T = 0

READ p, t

T = p

for i = 1 to t do
    T = T + (15 / 100 * T)
end for

WRITE T
```

- Bagaimana kita dapat mengaplikasikan model dari solusi kita sehingga model tersebut dapat dibuat menjadi program komputer dengan mudah?
 - Pseudocode yang ada sudah sangat jelas, dan baris per barisnya dapat diimplementasikan secara langsung menggunakan bahasa pemrograman apapun

Contoh implementasi dengan Python

```
b = 15
T = 0
p = input("Masukkan jumlah pinjaman: ")
t = input("Masukkan lama pinjaman: ")

T = int(p)

for i in range(1, int(t)):
    T = T + (15 / 100 * T)

print("Total pinjaman yang harus dibayarkan adalah: " + str(T))
```

Latihan

- ▶ Seseorang ingin membuat program komputer yang mampu menghitung jumlah kata unik yang ada dalam daftar kata dari sebuah naskah atau artikel.
- ▶ Buatlah pengembangan modelnya!