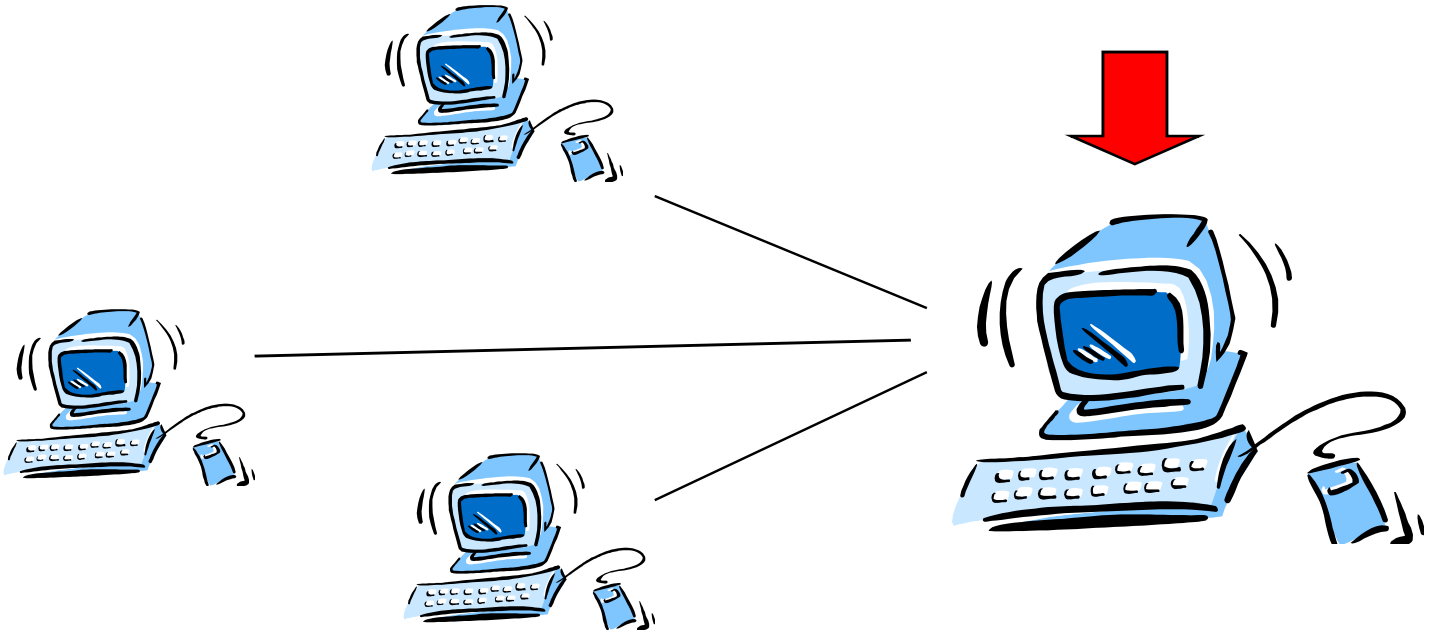
PENGANTAR STRATEGI DAN ANALISA ALGORITMA

wijanarto

Algorithm: himpunan instruksi yang menggambarkan bagaimana melakukan pekerjaan atau proses .

Programming: C,pascal



Algorithm – Definisi

From: Websters Revised Unabridged Dictionary, 1913:

Algorism (Al"go*rism Al"go*rithm)

- n. [OE. algorism, algrim, augrim, OF. algorisme, F. algorithme (cf. Sp. algoritmo, OSp. alguarismo, LL. algorismus)
- Al-Khowarezmi aslinya Abu Ja'far Mohammed ben Musa, ahli arithmetic awal abad 9, bukunya dalam bhs latin algorismus.

Algorithm

- Urutan instruksi untuk menentukan langkah yang di perlukan dalam menyelesaikan suatu tugas.
- Muhammad ibn Musa al-Khwarizmi

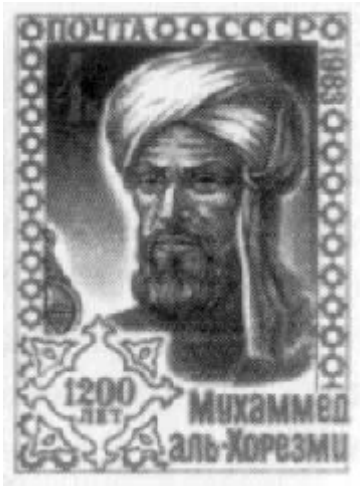
  Berasal dari Khowarezm (sekarang Khiva di Uzbekistan)

Source: http://www.atlapedia.com/online/maps/political/Kazakh_etc.htm

Algorithm– Sejarah

Muhammad ibn Musa Al-Khwarizmi

- Circa 160-230 A.H. (*anno Hegirae*)
- Circa 780-850 C.E. (Common Era)



Algorithm –Sejarah (lanj)

Muhammad ibn Musa Al-Khwarizmi

http://www-groups.dcs.st-andrews.ac.uk/~history/Mathematicians/Al-Khwarizmi.html

- Buku arithmetic:
  - Hindu numeration, decimal numbers, use of zero, method for finding square root
  - Latin translation (c.1120 CE): "*Algoritmi de numero Indorum*"
- Buku aljabar
  - *Hisab al-jabr w'al-muqabala*

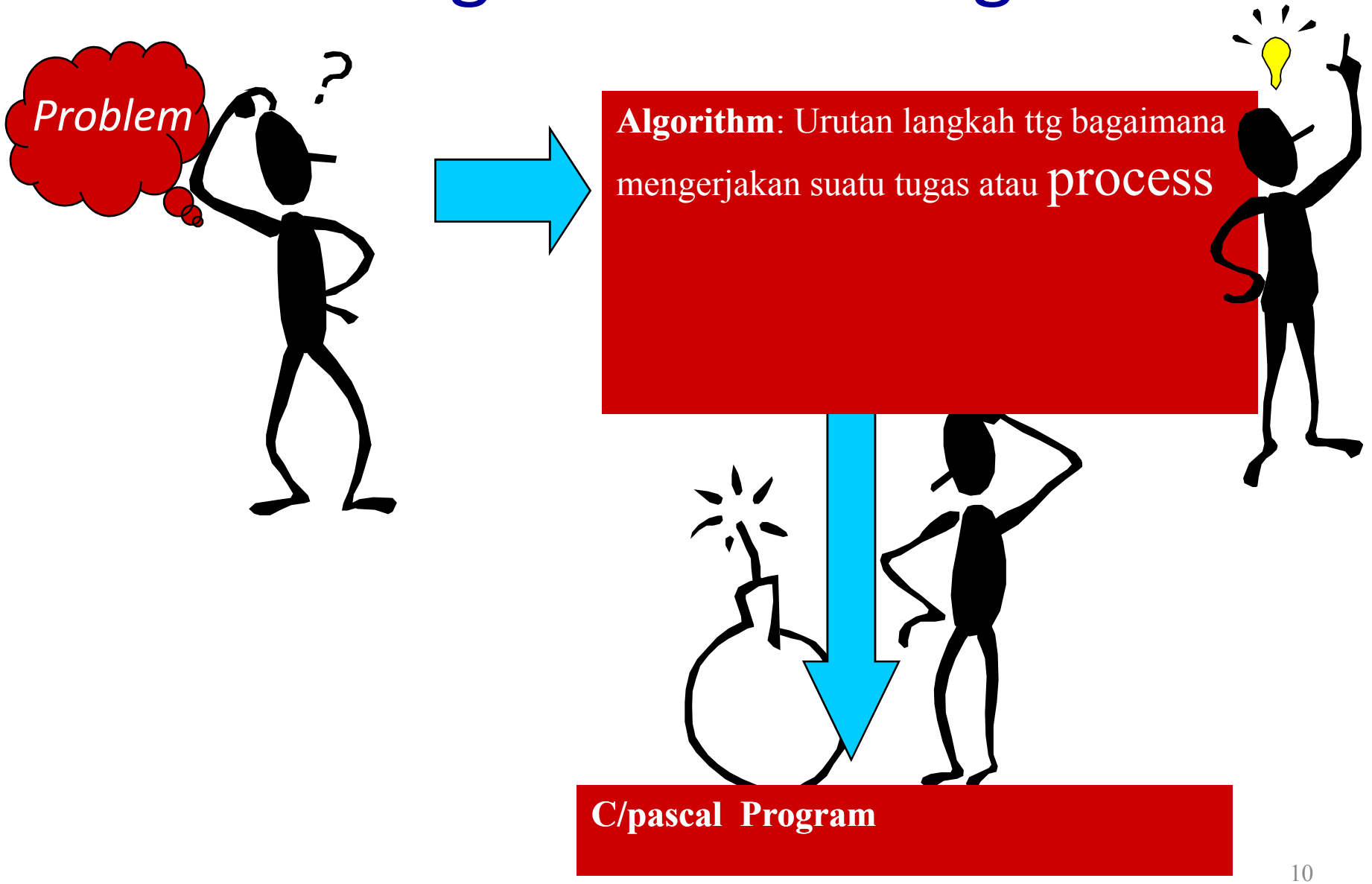
Algorithm – Working Definition

- Urutan langkah yang menggambarkan bagaimana melakukan suatu pekerjaan

Algorithm -- Contoh

- Resep masakan
- Instruksi Assembly
- Aturan main game
- Instruksi VCR
- Deskripsi teknik bela diri
- Arahan dari A to B
- Pola jahitan
- Manual perbaikan mobil

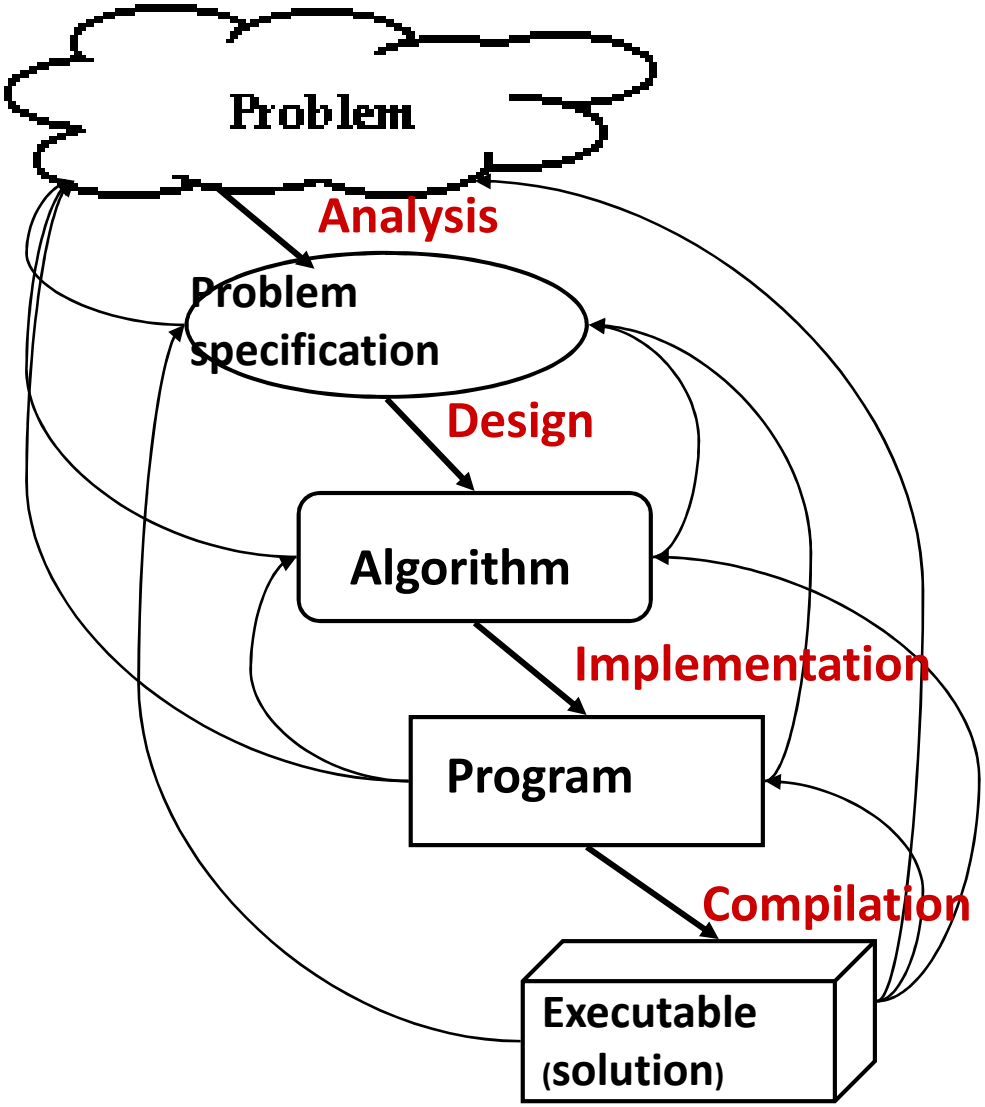
Dari Algorithm ke Program



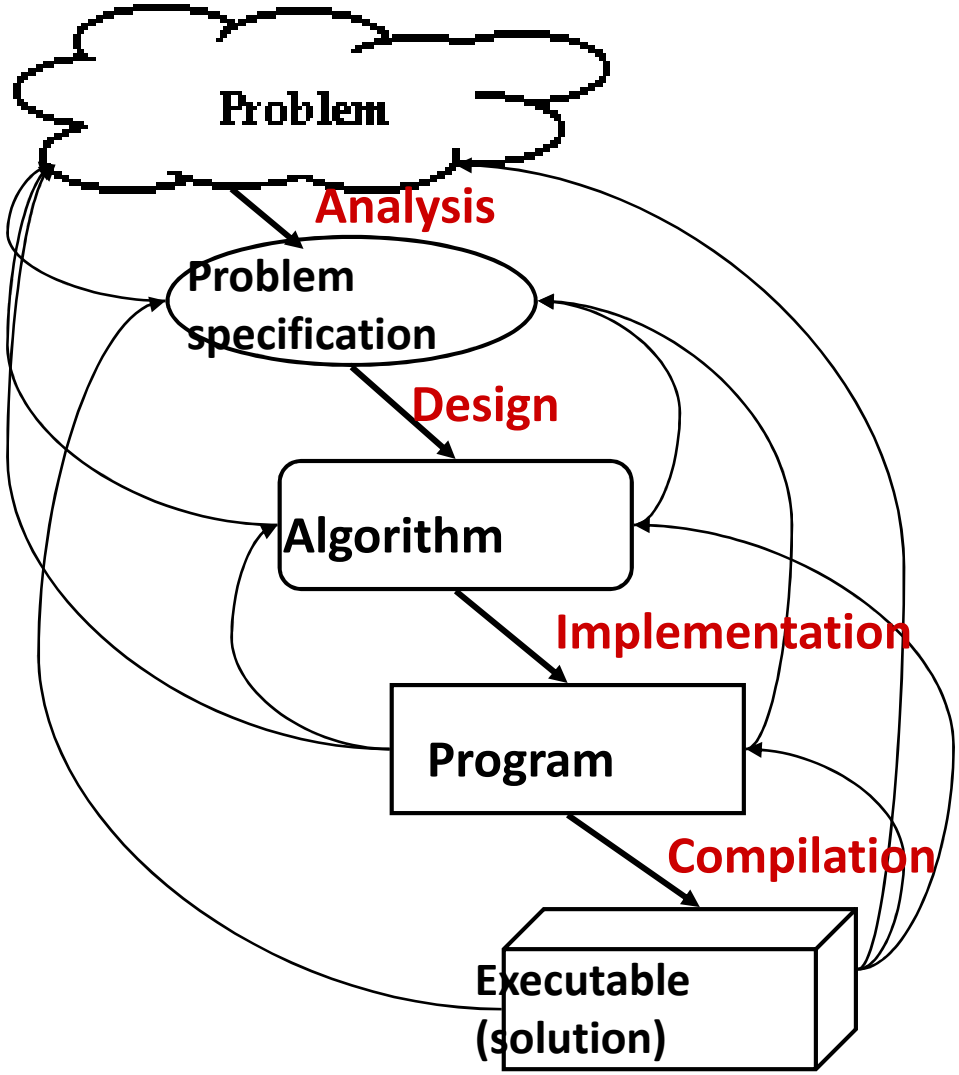
Bgmn kita memecahkan masalah ?

- Langsung kerjakan
- Tebak dan untung-untungan
- Trial error
- Pengalaman
- "Scientifically“ = dg cara ilmiah

Process memecahkan masalah



Process memecahkan masalah



"Dok, kepala saya sakit"

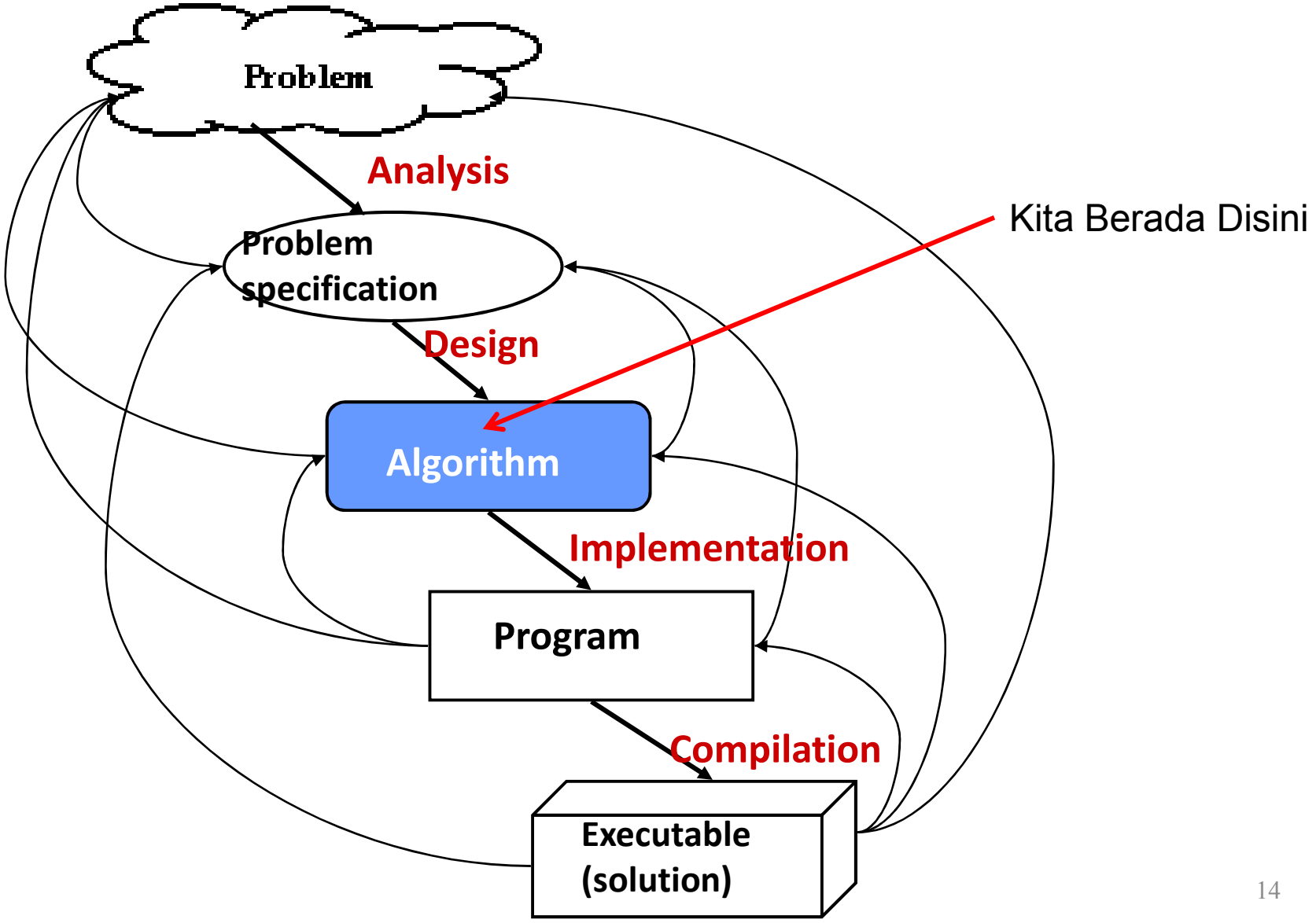
Pasien ada penyumbatan pd anterior parietal lobe.

1. Steril gergaji
2. bius
3. Potong batok kepala
4. Ambil spoon...
5. *etc., etc.*

```
steril(gergaji,alcohol);
raise_hammer();
lower hammer(fast);
start(saw);
/* etc. etc. */
```

0100111010110010101010101001
0101010101001100101010101010
0101101001110101010101001001
0111010011110101010111110101
0100011010000110 1...

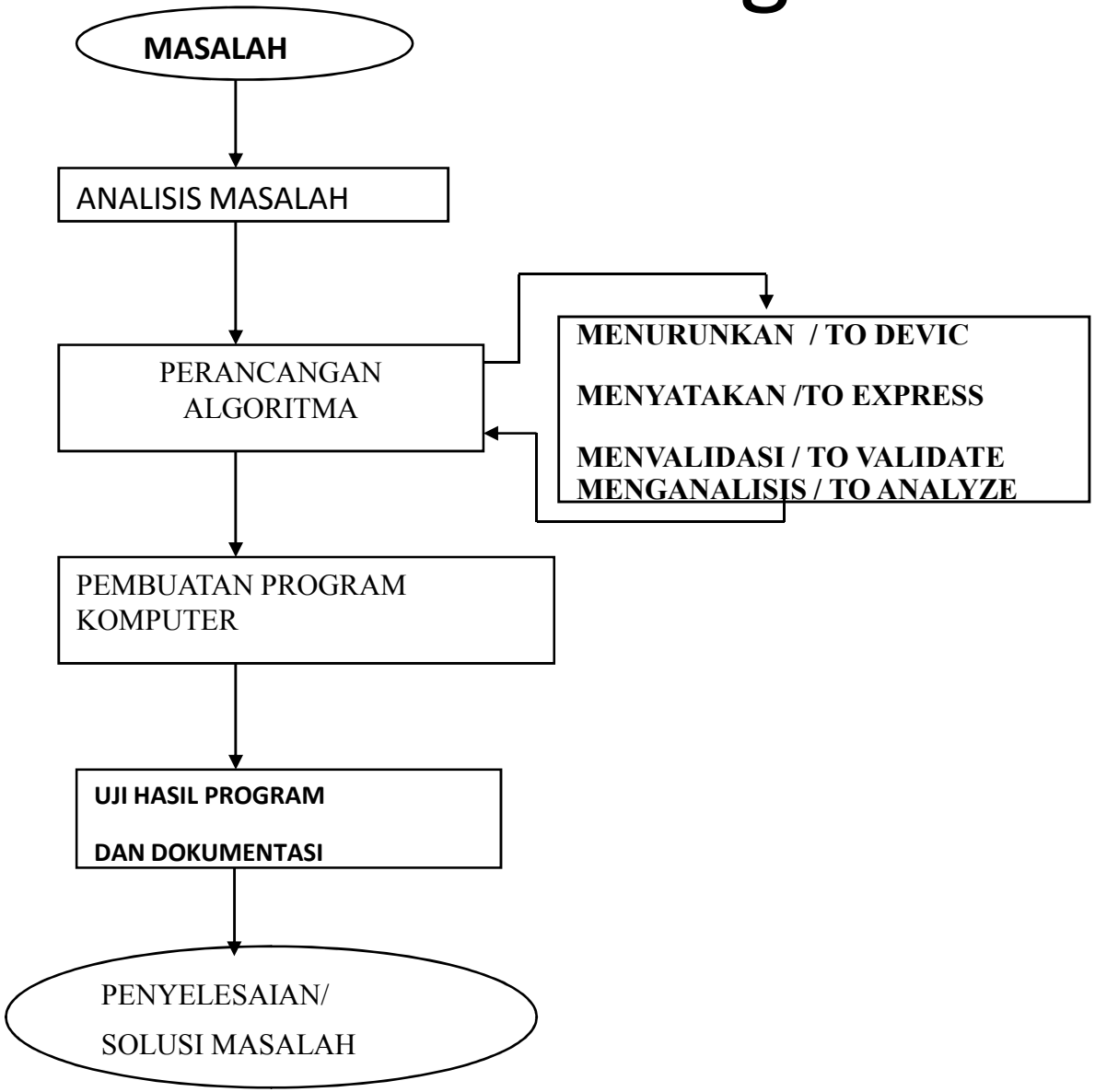
Process memecahkan masalah



Tahapan Penyelesaian Masalah
Versi lain

- Komputer diciptakan untuk membantu menyelesaikan masalah manusia. Masalah yang dimaksud tentunya masalah yang dapat diselesaikan dengan menggunakan computer (computerize problems). Adapun secara umum, langkah-langkah penyelesaian masalah dengan komputer adalah sebagai berikut

Diagram PS


MASALAH
ANALISIS MASALAH
PERANCANGAN ALGORITMA
MENURUNKAN / TO DEVIC
MENYATAKAN /TO EXPRESS
MENVALIDASI / TO VALIDATE
MENGANALISIS / TO ANALYZE
PEMBUATAN PROGRAM KOMPUTER
UJI HASIL PROGRAM
DAN DOKUMENTASI
PENYELESAIAN/
SOLUSI MASALAH

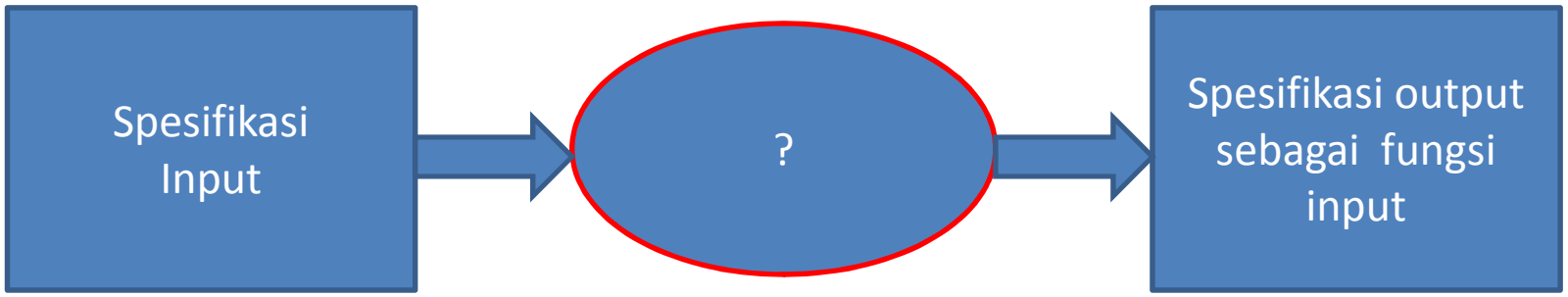
Tahapan Pemecahan Masalah

- **Analisis Masalah (40%)**
  - Analisis masalah adalah kegiatan mempelajari, mendalami masalah hingga mendapatkan ide-ide penyelesaian masalah (ide global).
- **Perancangan Algoritma (30%)**
  - Perancangan algoritma adalah pembuatan algoritma dimulai dari ide-ide penyelesaian masalah hingga terciptanya algoritma dalam bentuk standar (a.l. pseudocode).
- **Pembuatan Program Komputer (20%)**
  - Mentransfer algoritma menjadi kode program, yang sebelumnya perlu ditentukan struktur datanya.
- **Pengujian Hasil Program (5%)**
  - Running program untuk mengetahui apakah ada kesalahan, baik kesalahan sintax, running atau output/hasil.
- **Pembuatan Dokumentasi Program (5%)**
  - Pembuatan dokumentasi meliputi dokumentasi dalam program atau manual petunjuk pemakaian dan pemeliharaan program.

Struktur Data dan Algoritma

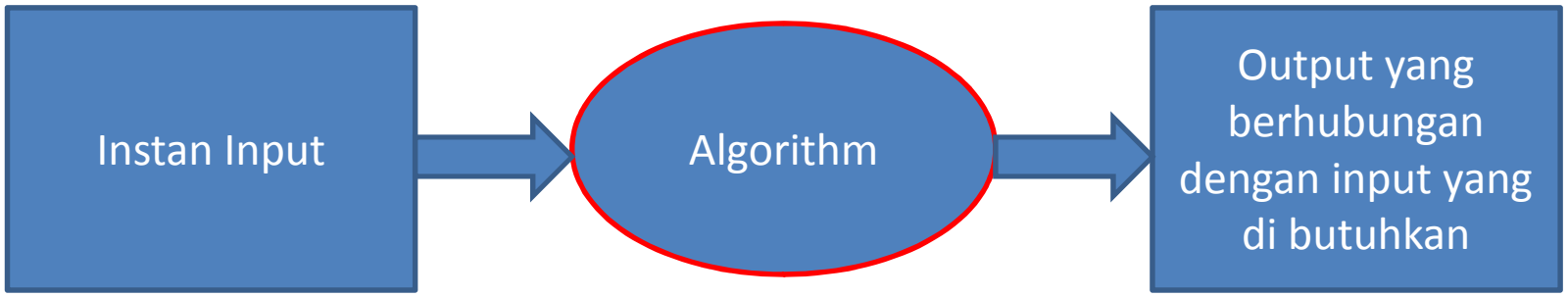
- Algoritma
  - Prosedur komputasional
  - Instruksi yang di kerjakan Step by step
- Program
  - Implementasi algortima dengan bahasa tertentu
- Struktur Data
  - ORGANISASI DATA yang di perlukan untuk menyelesaikan masalah

Masalah Algoritmik



- Sejumlah instan input yang tidak terbatas yang memenuhi spesifikasi, misal terurut menaik dari suatu bilangan bulat lebih besar dari nol
  - 1,20,,908,909,100000,10000000

Solusi Algoritmik



- Algoritma menggambarkan aksi dari instan input

Bagaimanakah Algoritma yang bagus ?

- Effisien
  - Running time
  - Space yang di pakai
- Efisiensi sebagai ukuran dari fungsi input
  - Jumlah bit dalam sejumlah input
  - Sejumlah elemen data (angka)

Mengukur running time

- Bagaimana kita mengukur running time algoritma
- Studi Eksperimen
  - Tulis program yang mengimplementasikan suatu algoritma
  - Jalankan program dengan input data yang berbeda-beda
  - Pakai pengukuran waktu

Waktu dalam c (gcc)

```
/*fungsi utk mendapatkan waktu dalam milisecond*/
void    get_current_time( double *time_returned )
    {
    struct timeval tp;

    gettimeofday (&tp, NULL);

    *time_returned = tp.tv_usec;
    *time_returned += tp.tv_sec*1000000;
    *time_returned /= 1E6;
}
```

Keterbatasan Studi eksperimen

- Harus dapat di implementasikan untuk mengetahui running time
- Keterbatasan input
- HW dan SW yang berbeda

Psuedo code

Algoritma ArrMax (A,n)

Input: Array A yang menyimpan n integer

Output: Nilai maximum dalam array A

```
max←a[0]
for i←1 to n-1 do
  if max<a[i]then max←a[i]
→ max
```

Notasi Algoritmik

- Eskpresi
  - **Simbol** matematika standar
    - Untuk ekspresi boolean dan aritmatika
  - ← untuk assignment
  - <,>,≤,≥,≠,= untuk relasi kesetaraan
- Deklarasi method/fungsi
  - nama(param1,param2) : Bla(X)
- Hasil Balik Fungsi
  - → X atau return X

Notasi Algoritmik

- Konstruksi program
  - `if(kondisi)then(aksi1)else(aksi lain)`
  - `while(kondisi)…do`
  - `repeat …until(kondisi)`
  - `for … do atau i Traversal 1..N`
  - `index array a[i],a[i,j]`
  - `Assignment : Y=10 atau X←10`

Strategi dan Analisis Algoritma

- **Strategi** adalah rencana yang cermat mengenai kegiatan untuk mencapai sasaran khusus (KBBI).
- **Algoritma** adalah urutan langkah-langkah untuk memecahkan suatu masalah.
- **Strategi algoritmik** adalah kumpulan metode atau teknik untuk memecahkan masalah guna mencapai tujuan yang ditentukan, yang dalam hal ini deskripsi metode atau teknik tersebut dinyatakan dalam suatu urutan langkah-langkah penyelesaian.

Strategi dan Analisis Algoritma

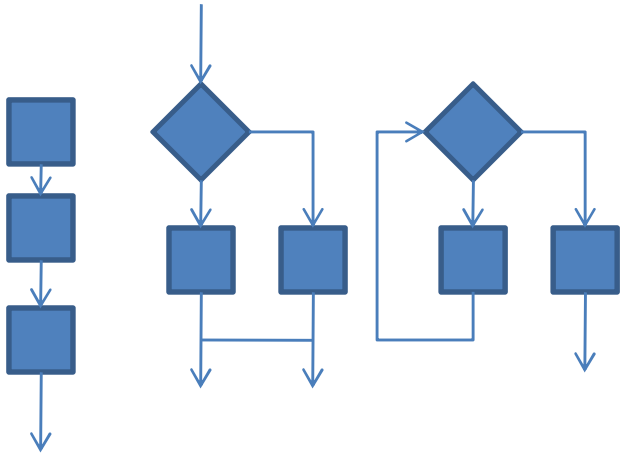
- **Analisis** berhubungan dengan cara pandang kita terhadap seberapa cepat algoritma itu bekerja (waktu tempuh) dan perlu berapa banya sumber daya (memori)
- Dengan menganalisis suatu algoritma, diharapkan akan mendapatkan strategi yang tepat untuk memilih algoritma dalam menyelesaikan masalah.

Analisis Algoritma

- **Analisis algoritma** adalah salah satu tahapan dari perancangan algoritma, Sedangkan perancangan algoritma adalah salah satu tahapan dari proses pemecahan masalah dengan komputer. Sebelum pembahasan analisis algoritma, terlebih dahulu kita bahas tahapan pemecahan masalah dengan komputer.

Struktur Dasar Algoritma

- 3 Bentuk Standar Algoritma
  - Sekuensial
  - Pencabangan (branching)
  - Kalang (looping)



Aspek Algoritma

- Valid
  - Algoritma harus benar, artinya memberikan keluaran yang di kehendaki
- Efektifitas
  - Tepat guna, hasil guna, akurat, mencapa tujuan, aman
- Efisiensi
  - Hemat, Cepat dengan space memori yang kecil
- Berhingga langkahnya
- Logis dan terstruktur

Analisis Algoritma

- Memori
  - Di pengaruhi oleh variabel
  - Stack : tempat menyimpan data sebenarnya
- Kecepatan
  - Cache memori (HW)
  - Jumlah langkah (algoritma)
  - Operator dan operand

Strategi Algoritma

- Bagaimana merencanakan atau menentukan suatu model penyelesaian masalah, untuk itu perlu di buat disain yang memenuhi aspek algoritma yang di maksud.
- Bagaimana memilih model penyelesian masalah yang sudah ada sehingga di pandang dapat dengan tepat untuk itu

Efektifitas dan efisiensi

- Kecenderungan efektifitas adalah
  - Kebenaran algoritma tersebut
  - Merespon terhadap inputan apapun
- Efisiensi
  - Space yang di perlukan
  - Waktu/banyak langkah atau waktu tempuh yang di gunakan oleh :
    - Banyaknya statement
    - Jenis statement/struktur
    - Banyak operasi aritmatik dan logik
    - Procedure dan fungsi call
      - Built-in function
      - User define
        - » Recursif
        - » Non recursive

Kompleksitas Algoritma

- Kebutuhan space dan waktu tempuh inilah yang di sebut dengan kompleksitas algoritma
- Space sering unpredicted, terutama jika banyak menggunakan pointer/ tipe data dinamis
- Seringkali space tidak disertakan dalam penentuan kompleksitas algoritma, kecuali jika tipe data yang terlibat adalah **statis**.

Problem ?

- Kompleksitas algoritma memiliki batas bawah dan batas atas yang di sebut dengan ORDER
- Bagaimana menentukan kompleksitas (dalam bentuk fungsi) ???
- Bagaimana menentukan ORDERnya ????
- Untuk menentukan 2 hal tersebut di perlukan Mathematical Background.

Background Matematika

- Hanya review, karena sudah di ajarkan di mata kuliah lain (Kalkulus)
- Induksi Matematika
- Fungsi, barisan dan deret
- Limit barisan

Induksi Matematika

- Buktikan bahwa statement P(n) adalah bilangan asli untuk $n \in N$. P(n) benar, kecuali
  - Terdapat $n_0 \in N$ sehingga $P(n_0)$ benar
  - Untuk setiap $n \in n_0$, jika P(n) benar maka P(n+1) juga benar

Induksi Matematika

- P(a) Benar, ini di sebut sebagai langkah basis.
- Jika P(n) benar maka P(n+1) benar untuk $n \geq a$, ini di sebut sebagai langkah induksi.

Induksi Matematika

$P(n)=1+2+3+4+\ldots+n=\frac{1}{2}n(n+1)$

$P(1)=1=\frac{1}{2}.1(1+1)$, adalah BENAR (**basis**)

$n \geq 1$ dan P(n) adalah BENAR, maka

$P(n)=1+2+\#+4+\ldots+n=\frac{1}{2}n(n+1)$ adalah BENAR

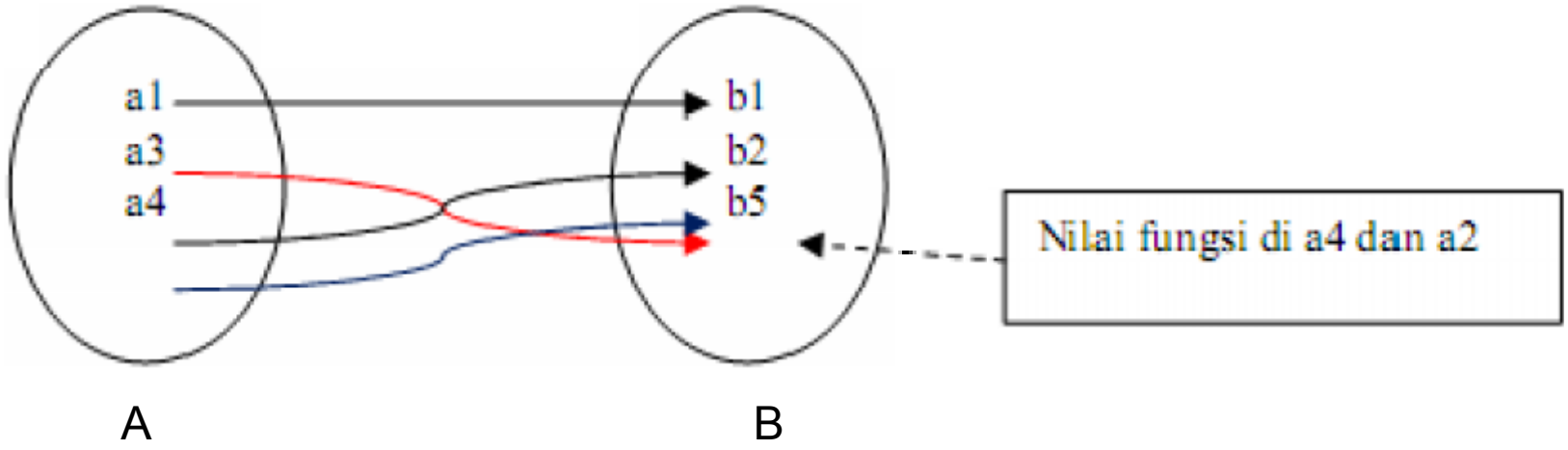
$P(n+1)=1+2+\#+4+\ldots+n+(n+1)=\frac{1}{2}n(n+1)+(n+1)$ (**induksi**)

$$=\frac{1}{2}(n+1)(n+2)$$

$$=\frac{1}{2}(n+1)((n+1)+1)$$, juga BENAR

Fungsi

- Pemetaan suatu nilai dari domain ke range
  F: A→B



Fungsi

- Penyajian Fungsi
- Model Matematika

$$F(n)\begin{cases} 1 & n \le 1 \\ n * F(n-1) & n > 1 \end{cases}$$

- Algoritma

```
F(n)
   if(n≤1)then →1
   else →(n*F(n+1))
```

Barisan

- Merupakan fungsi dengan himpunan asal berupa bilangan asli, jika *a* merupakan nilai fungsi *f* di *n*, maka $a_n=f(n)$ dan dapat ditulis $\{ a_n | n=1,2,3,... \}$ disebut dengan BARISAN.
- Contoh :

$$a_n = f(n) = \frac{1}{n^2}$$

Barisannya adalah $\frac{1}{1^2}, \frac{1}{2^2}, \frac{1}{3^2}, \frac{1}{4^2}, \ldots$

$$a_n = f(n) = \begin{cases} n & , n = 1,2 \\ 3(f(n-2))^2 & , n \geq 3 \end{cases}$$

$a_1=1$, $a_2=2$, $a_3=3(f(1))^2=3$

$a4=3(f(2))^2=3X2^2=12$

$a5=3(f(3))^2=3X3^2=27,\ldots$

barisannya adalah 1, 2, 3, 12, 27, 432,.....

Deret

- Merupakan barisan dalam bentuk **jumlahan barisan**.
- Di ketahui barisan a1,a2,a3,a4,…, a,…
  - DERET1= $a_1, a_1+a_2, a_1+a_2+a_3, \ldots$
  - DERET2=$a_1, a_2, a_1+a_3, a_2+a_4, a_1+a_3+a_5, a_2+a_6, \ldots$

Maka deret1 dapat di tulis sebagai $S_n = \sum_{i=1}^{n} a_i$ dan

Deret2 dapat di tulis sebagai $S_n = \begin{cases} \sum_{i=1}^{n} a_{2i} & , n = genap \\ \sum_{i=1}^{\frac{n+1}{2}} a_{2i-1} & n = gasal \end{cases}$

Limit Barisan

- Misalkan di ketahui suatu barisan $\{a_n\}$ dengan $a=f(n)$, maka barisan a_n di katakan konvergen ke L apabila :

$$\lim_{n\to\infty} f(n) = L \text{ atau}$$

a_n cenderung ke nilai L, dimana $L \in R$

Limit Barisan

- Contoh

barisan $\left\{\frac{n}{n+1}\right\}: \frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{4}{5}, \ldots$

$f(n) = \frac{n}{n+1}$

$$\lim_{n\to\infty} f(n) = \lim_{n\to\infty} \frac{n}{n+1} : \frac{n}{n} = \lim_{n\to\infty} \frac{1}{1+\frac{1}{n}} = \frac{1}{1+0} = 1, \text{ ingat } \frac{1}{\infty} = 0.$$

Jadi $\left\{\frac{n}{n+1}\right\}$ konvergen ke 1

Teorema

Teorema

Misalkan f(n) konvergen ke L1 dan g(n) konvergen ke L2, maka

F(n)±g(n)	konvergen ke L1±L2
k.f(n)	konvergen ke L1
f(n)*g(n)	konvergen ke L1*L2
$\frac{f(n)}{g(n)}$	konvergen ke $\frac{L1}{L2}$, untuk L2≠0

Khusus untuk perkalian dan pembagian (* dan /) jika salah satu fungsinya tidak konvergen, *BELUM* tentu hasilnya tidak konvergen, bias saja hasilnya konvergen.

contoh

f(n)=n, $g(n)=\frac{1}{n+1}$

$\lim_{n\to\infty} f(n) = \lim_{n\to\infty} n = \infty$, tidak konvergen

$\lim_{n\to\infty} g(n) = \lim_{n\to\infty} \frac{1}{n+1} = \frac{1}{\infty} = 0$, konvergen ke 0

An=f(n)*g(n)=n* $\frac{1}{n+1} = \frac{n}{n+1}$, jelas konvergen ke 1, karena $\frac{\infty}{\infty} = 1$

Jadi barisan $\{\frac{n}{n+1}\}$konvergen ke 1.

Latihan

1. Jelaskan bagaimana masalah seharusnya di selesaikan
2. Dapatkah anda gambarkan komponen penyelesaian masalah, berikan contoh aktual.
3. Tuliskan pengertian konsep strategi algoritma dan analisa algoritma serta hubungan diantaranya.
4. Dapatkah anda menuliskan struktur dasar algoritma, berikan contohnya
5. Jelaskan konsep mengenai algoritma, program, pemrograman.
6. Sebut dan jelaskan aspek-aspek suatu algoritma dan bagaimana pengukuran dilakukan.
7. Tuliskan konsep dasar suatu fungsi, deret dan barisan.
8. Apa peranan struktur data dalam analisa algoritma.