Pertemuan - 1

Pengenalan Algoritme

Pengertian Algoritme

- **Algoritme:**
  - Satu set aturan untuk menyelesaikan masalah dalam jumlah langkah yang terbatas.
  - Urutan langkah-langkah penyelesaian masalah yang disusun secara sistematis dan logis

Pengertian Program

- **Program:**
  - Implementasi algoritme pada komputer menggunakan bahasa pemrograman tertentu
  - Jadi algoritme dapat dipandang sebagai abtraksi/ inti sari dari program

Manfaat penggunaan algoritme

- Algoritma independen (bebas) dari bahasa pemrograman dan komputer yang melaksanakannya.
- Notasi algoritma dapat diterjemahkan ke dalam berbagai bahasa pemrograman.
- Dapat memperkirakan jumlah waktu dan memori yang diperlukan tanpa harus diimplementasikan (dengan teknik matematika)

Sejarah

- Berasal dari kata *algorism* yang berarti proses menghitung dengan angka arab
- Abu Ja'far Muhammad Ibnu Musa Al-Khuwarizmi menulis buku yang berjudul *Kitab Al Jabar Wal Muqabala* yang artinya "Buku pemugaran dan pengurangan" (*The book of restoration and reduction).*
- Ilmuwan Persia (abad 9), masuk Indonesia 80-an

Penyajian Algoritme

Algoritme bisa dibuat dengan:

- Teknik tulisan:
  - Bahasa Inggris atau bahasa lain
  - Pseudocode.
- Teknik gambar : *Flow chart*

Contoh : Algoritma Menggunakan Kalkulator

Mulai

Nyalakan kalkulator

Kosongkan Kalkulator

Ulangi

Input harga

Tekan tombol Plus (+)

Sampai semua harga diinput

Tampilkan total harga

Matikan kalkulator

Selesai

Karakteristik Algoritme (Donal E. Knuth)

- Input: algoritma dapat memiliki nol atau lebih inputan dari luar.
- Output: algoritma harus memiliki minimal satu buah output keluaran.
- *Definiteness (*pasti*):* algoritma memiliki instruksi-instruksi yang jelas dan tidak ambigu.
- *Finiteness* (ada batas): algoritma harus memiliki titik berhenti (*stopping role*).
- *Effectiveness* (tepat dan efisien): algoritma sebisa mungkin harus dapat dilaksanakan dan efektif.

Contoh:
menghitung luas dan keliling lingkaran

Algoritme:

- Baca jari-jari lingkaran (r)
- Tentukan konstanta phi = 3.14
- Hitung luas dan keliling
  - L = phi*r*r
  - K = 2*phi*r
- Tampilkan L (luas) dan K (keliling)

Contoh:
menghitung rata-rata tiga bilangan

Algoritme:

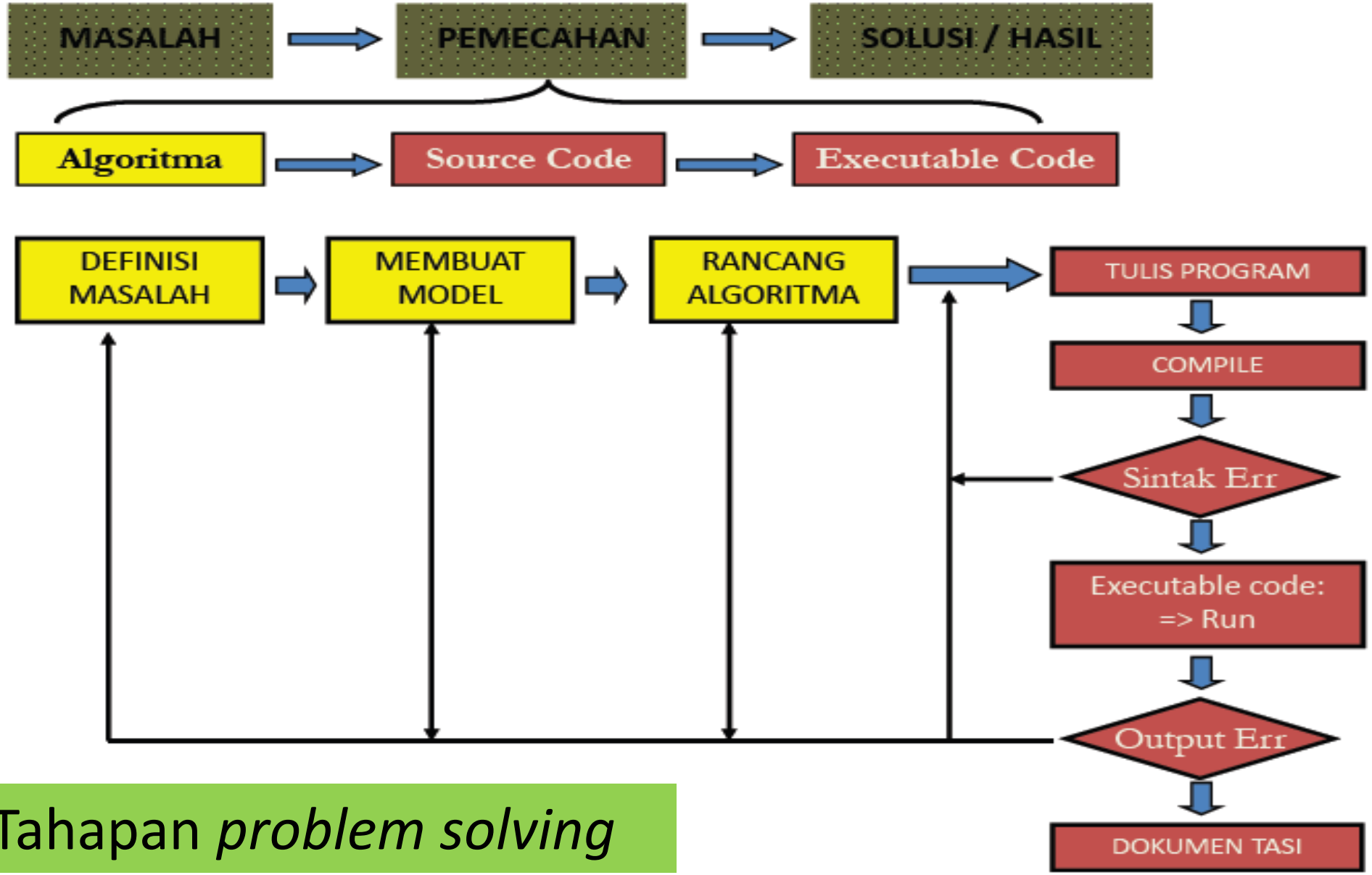
- Baca bilangan a, b, dan c
- Jumlahkan ketiga bilangan tersebut (x = a+b+c)
- Bagi jumlah tersebut dengan 3 (y = x/3)
- Tampilkan hasilnya (y)

Tahapan *problem solving* menggunakan komputer

- Pendefinisian masalah -> *user requirement*
  - Tentukan apa yang menjadi masalah
  - Tentukan data input yang diperlukan
  - Tentukan output yang diinginkan
- Perancangan algoritme -> gunakan teknik/ strategi yang sesuai

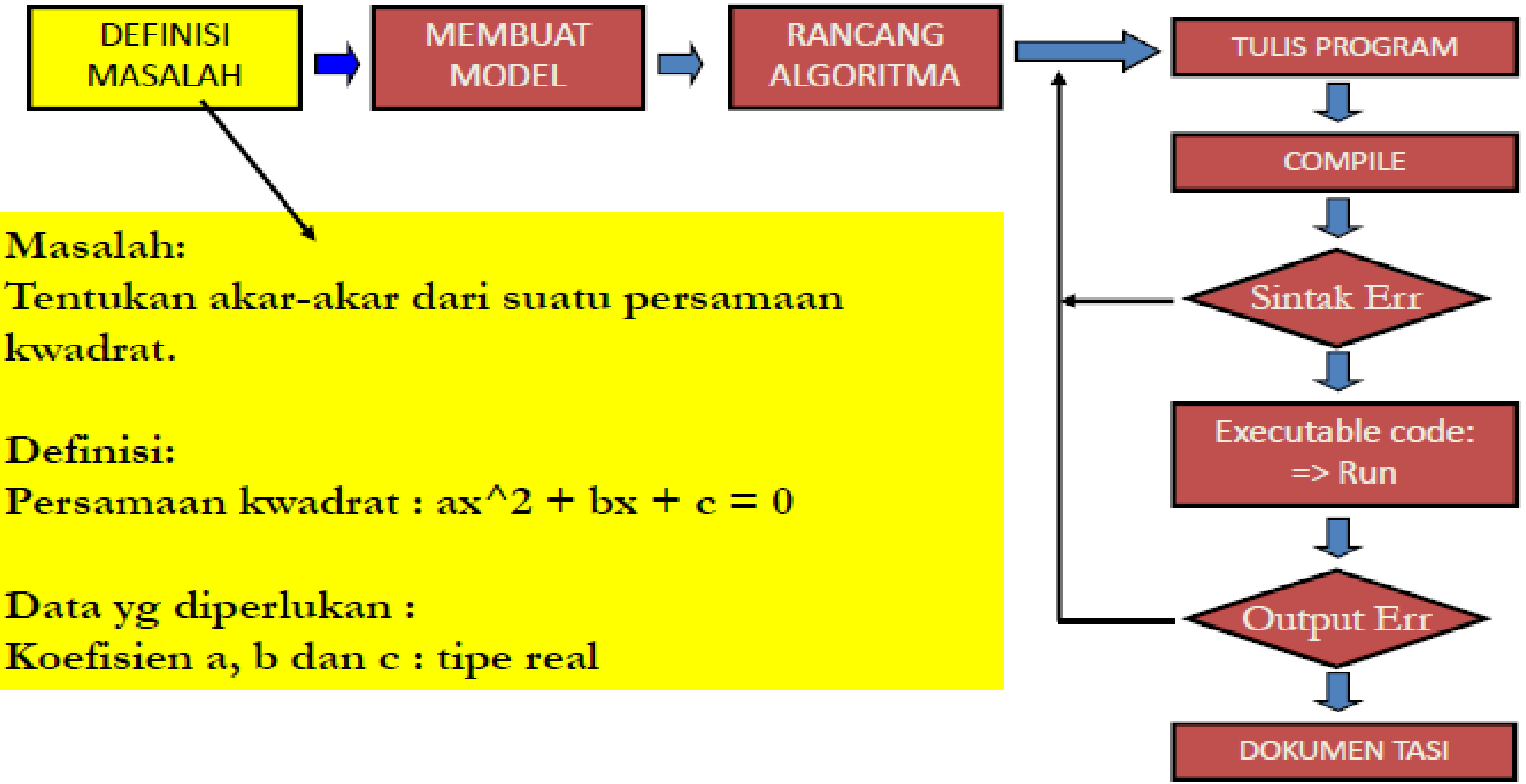
Tahapan *problem solving* menggunakan komputer

- Implementasi : pilih bhs pemrograman yg sesuai
- Test program : *syntax & logic error*, *accuracy*
- Dokumentasi hasil
- Pemeliharaan
  - Memperbaiki kekurangan yang ditemukan kemudian
  - Memodifikasi, karena perubahan spesifikasi

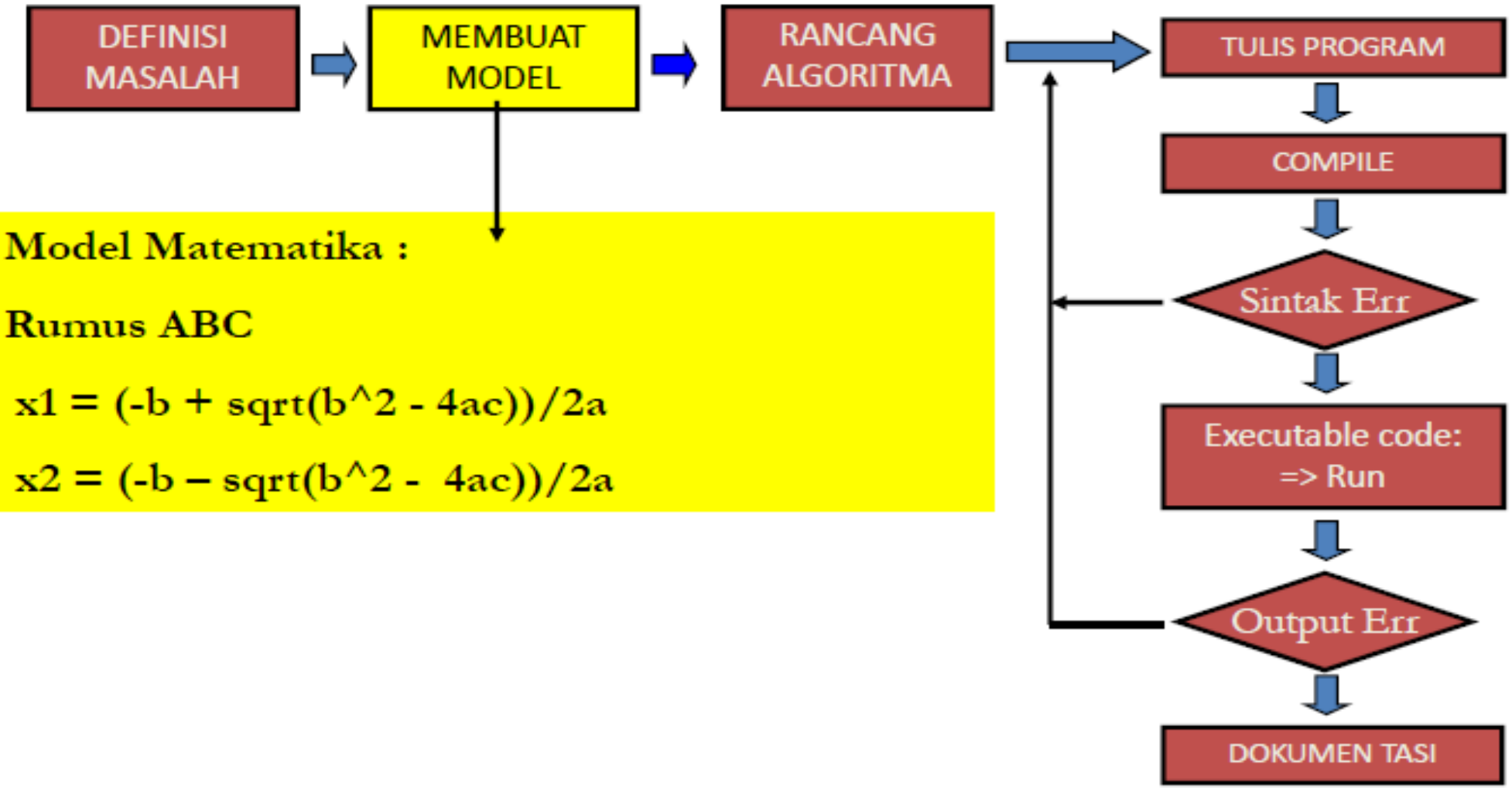


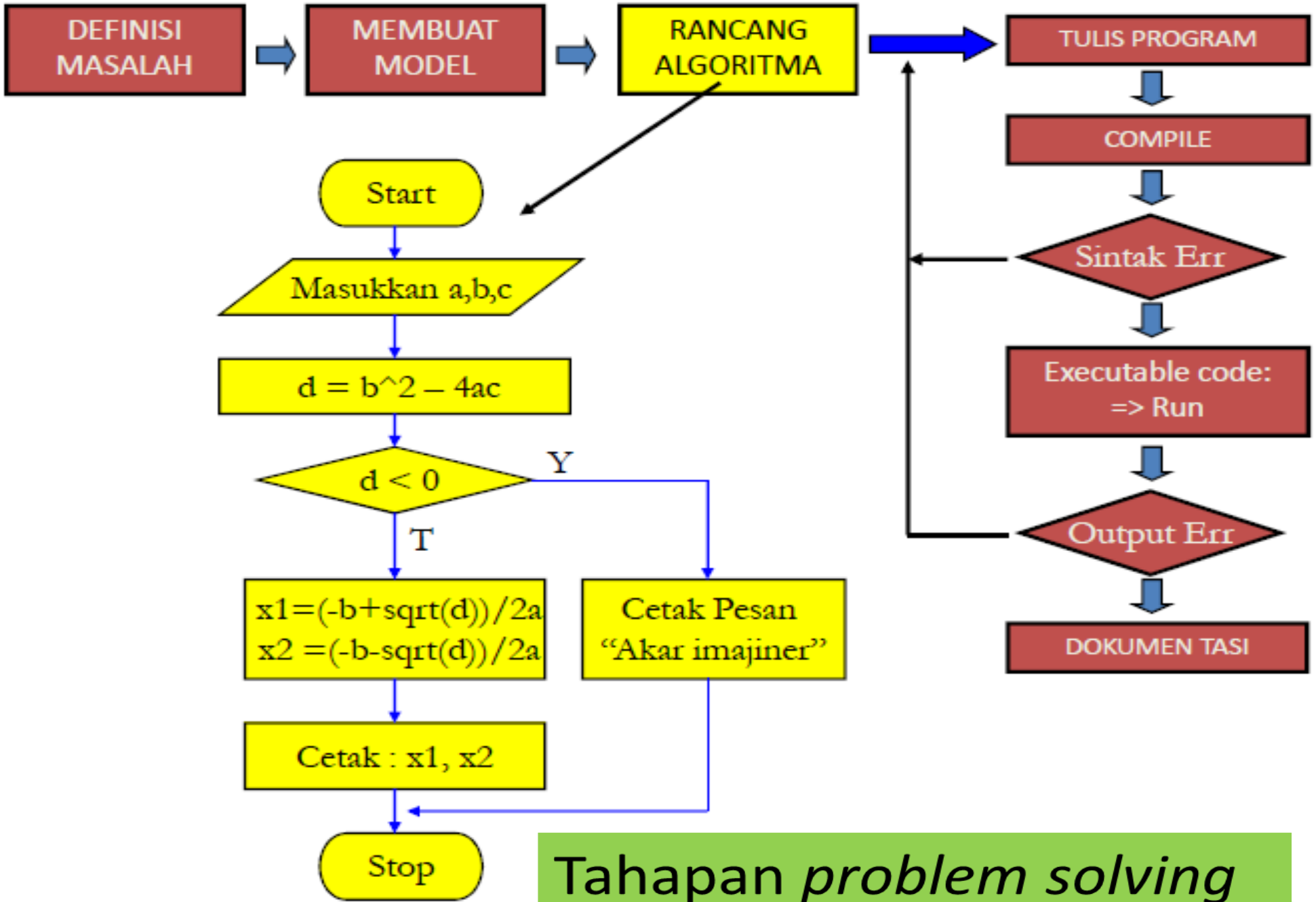
Tahapan *problem solving*

Tahapan *problem solving*



Tahapan *problem solving*

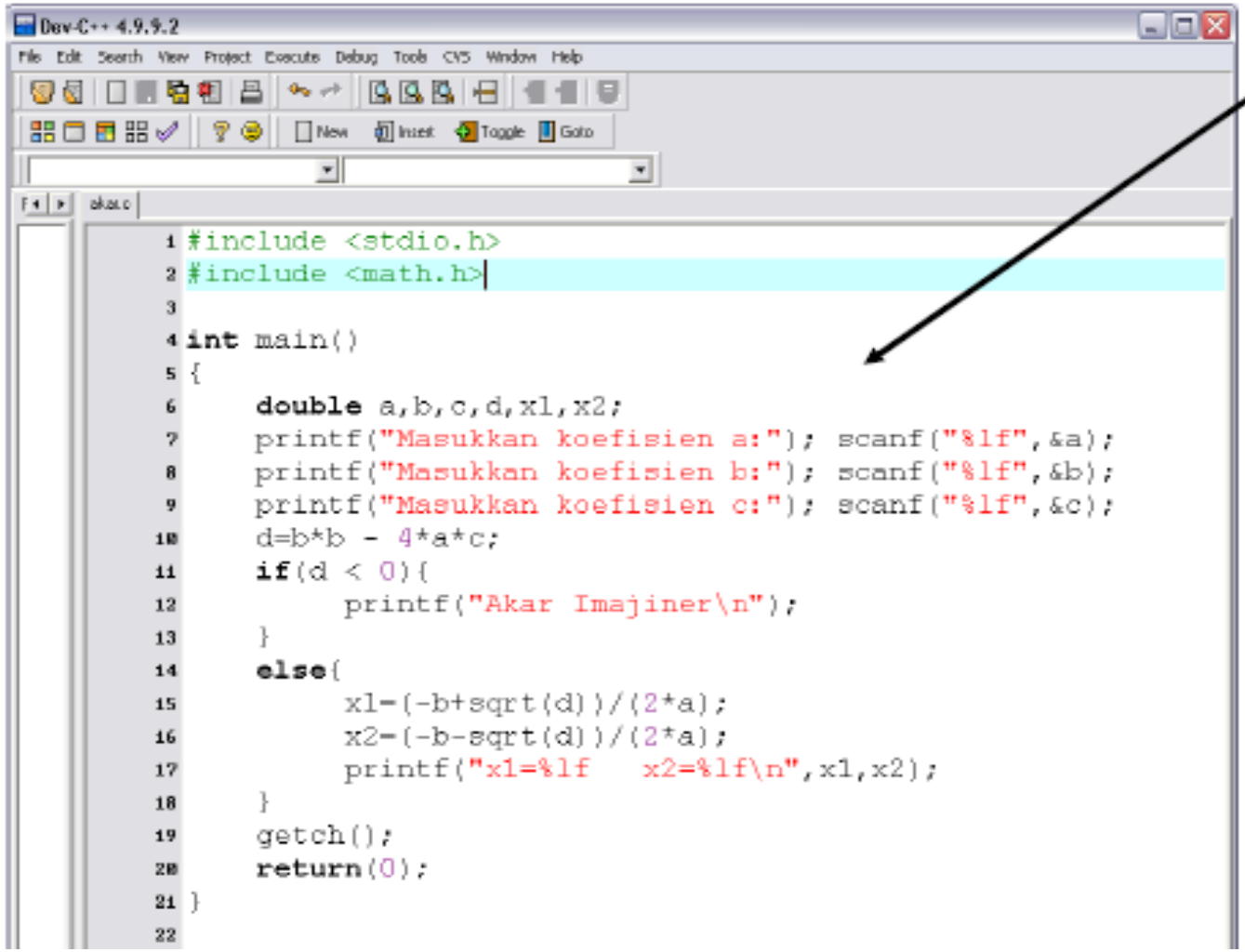




Tahapan *problem solving*

Tahapan *problem solving*

DEFINISI MASALAH → MEMBUAT MODEL → RANCANG ALGORITMA → TULIS PROGRAM → COMPILE → Sintak Err → Executable code: => Run → Output Err → DOKUMEN TASI

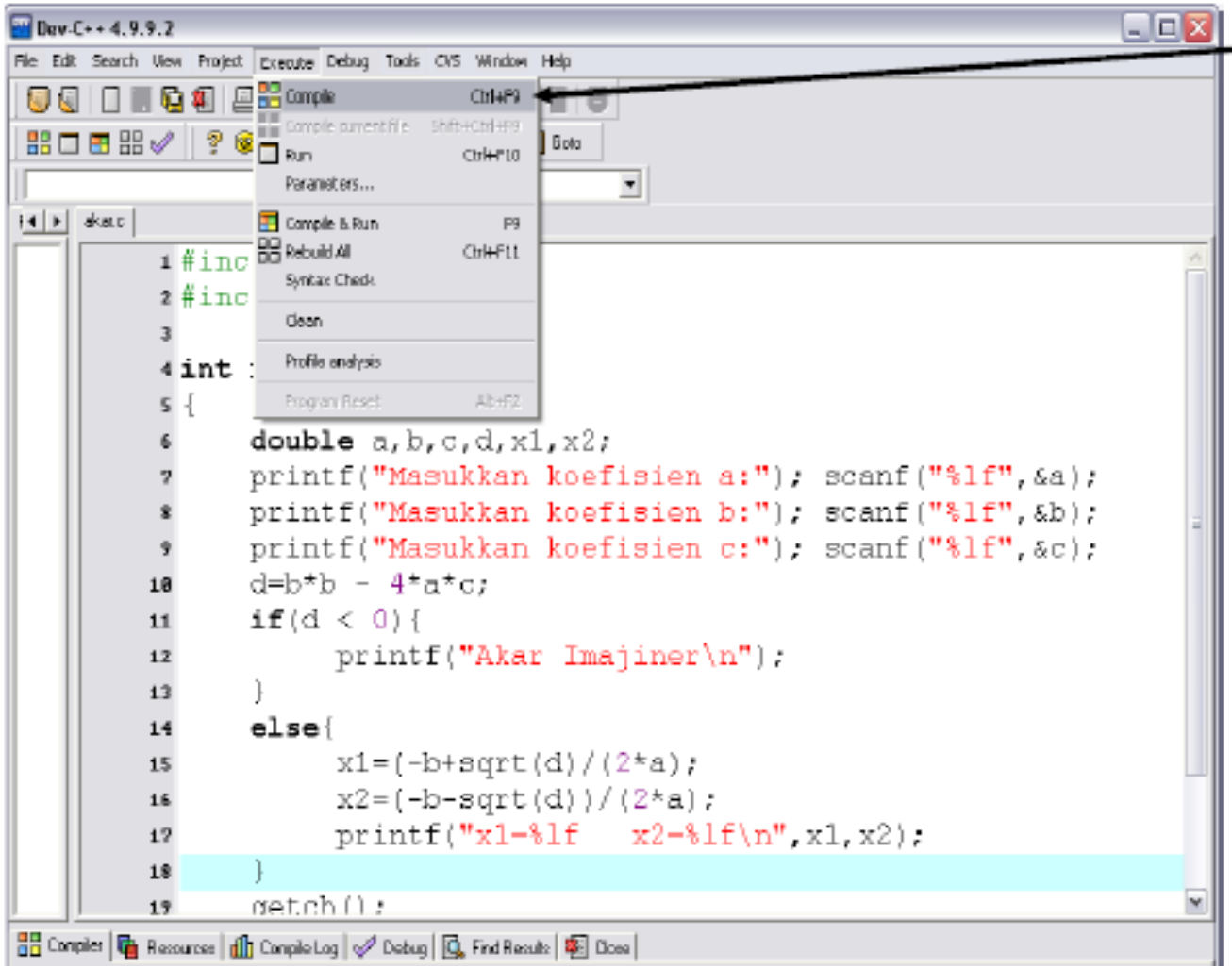



```c
#include <stdio.h>
#include <math.h>

int main()
{
    double a,b,c,d,x1,x2;
    printf("Masukkan koefisien a:"); scanf("%lf",&a);
    printf("Masukkan koefisien b:"); scanf("%lf",&b);
    printf("Masukkan koefisien c:"); scanf("%lf",&c);
    d=b*b - 4*a*c;
    if(d < 0){
        printf("Akar Imajiner\n");
    }
    else{
        x1=(-b+sqrt(d))/(2*a);
        x2=(-b-sqrt(d))/(2*a);
        printf("x1=%lf   x2=%lf\n",x1,x2);
    }
    getch();
    return(0);
}
```

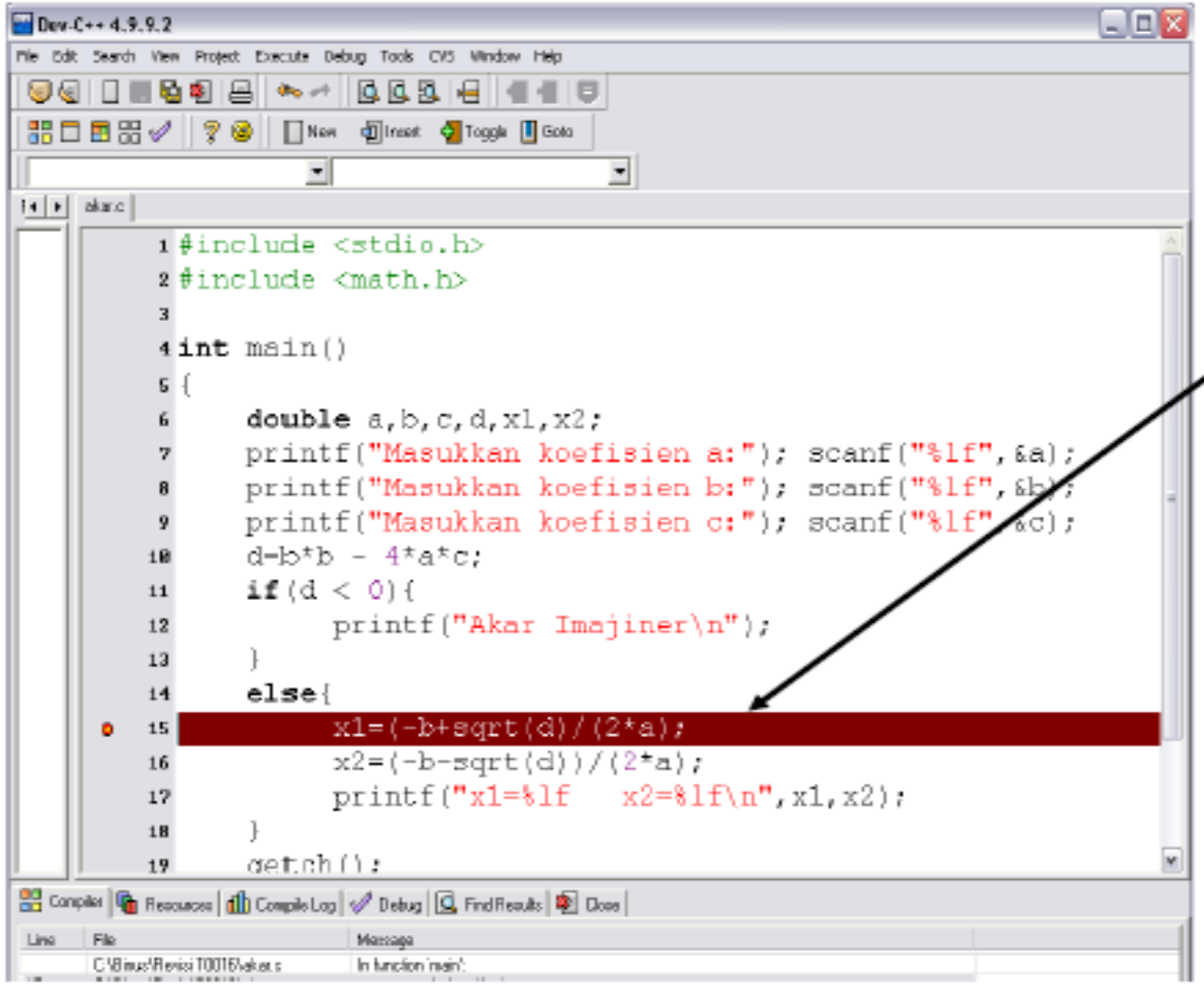
Tahapan *problem solving*

DEFINISI MASALAH → MEMBUAT MODEL → RANCANG ALGORITMA → TULIS PROGRAM → COMPILE → Sintak Err → Executable code: => Run → Output Err → DOKUMEN TASI



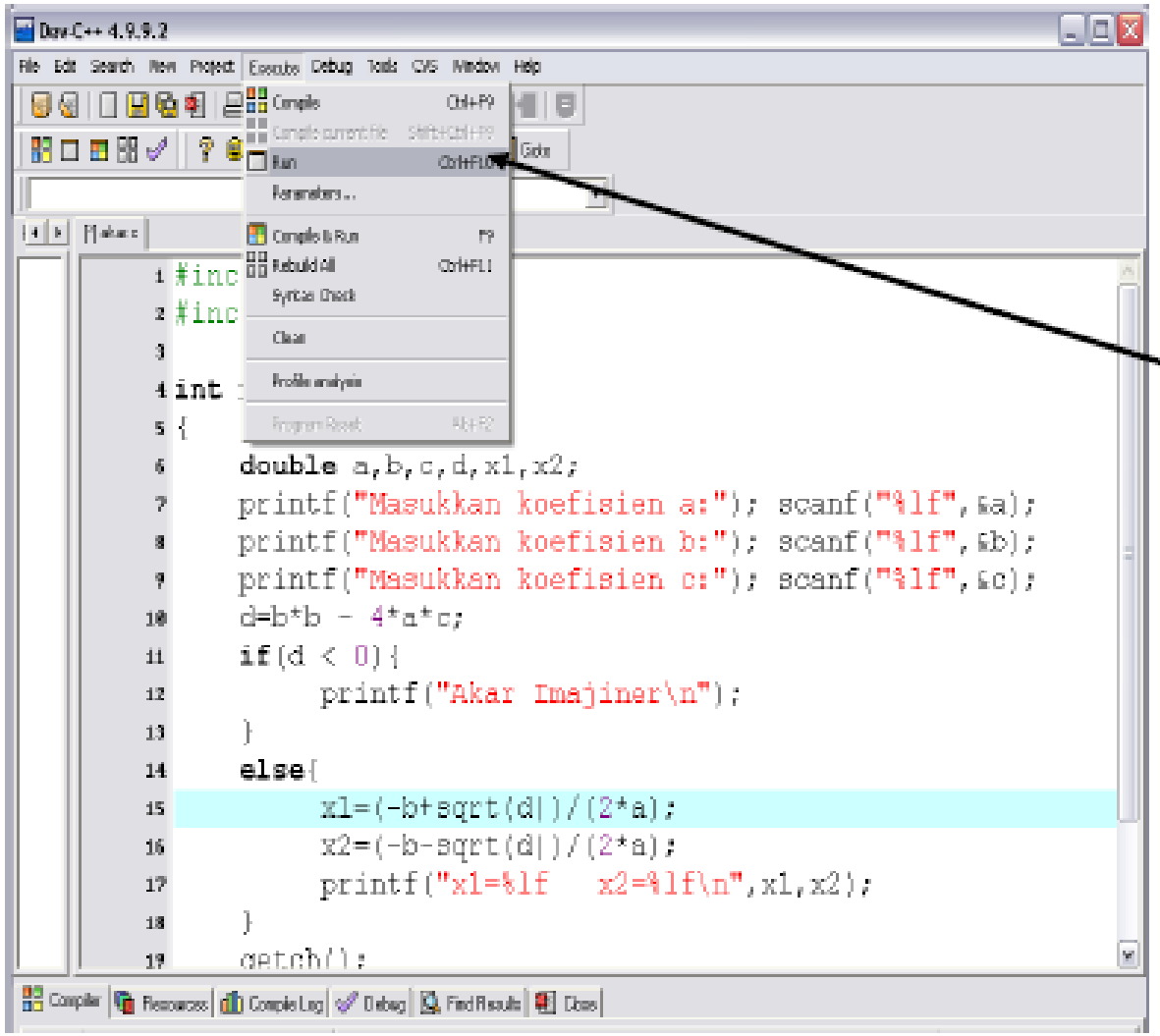
Tahapan *problem solving*

DEFINISI MASALAH → MEMBUAT MODEL → RANCANG ALGORITMA → TULIS PROGRAM → COMPILE → Sintak Err → Executable code: => Run → Output Err → DOKUMEN TASI

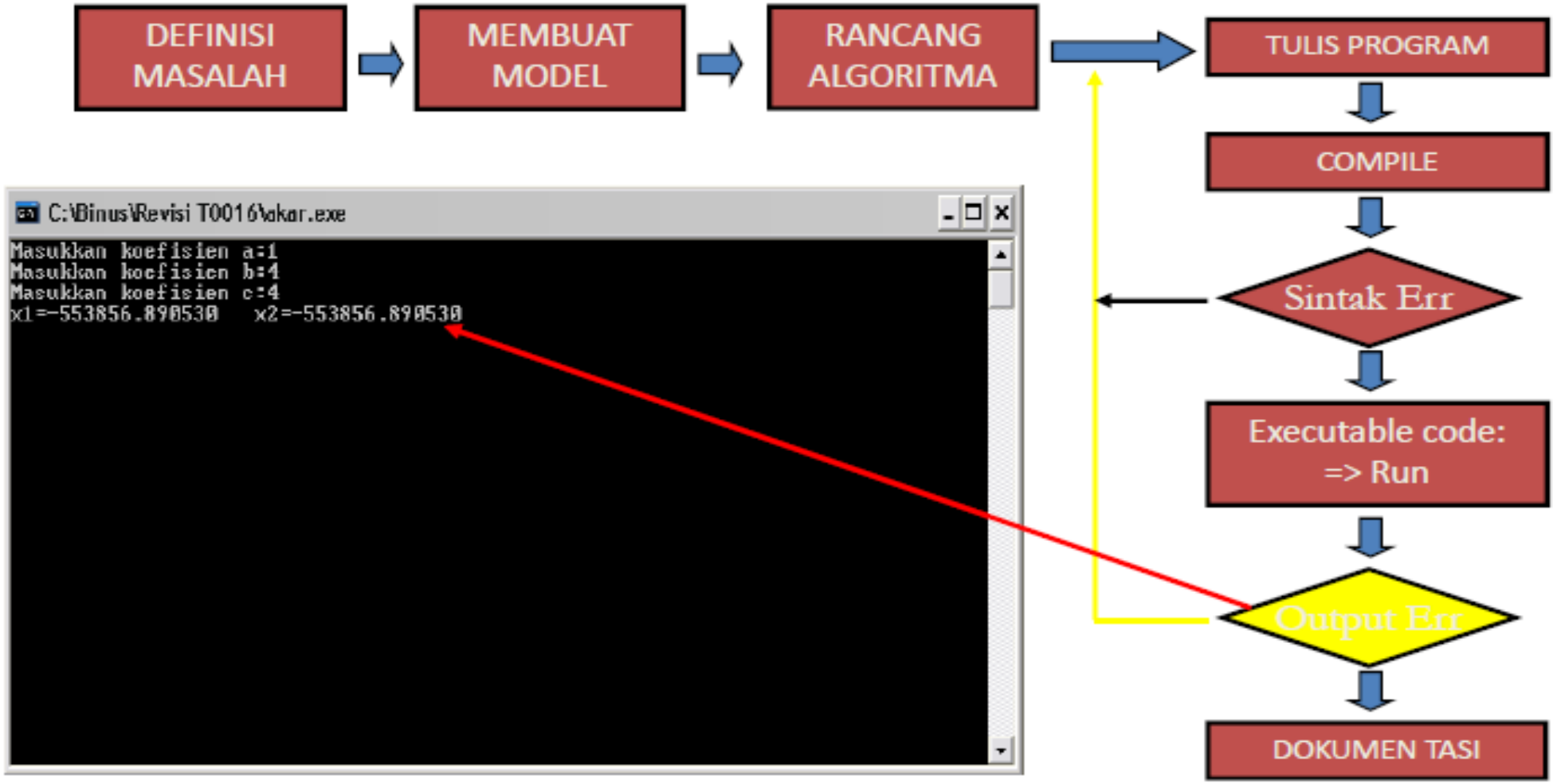


Tahapan *problem solving*

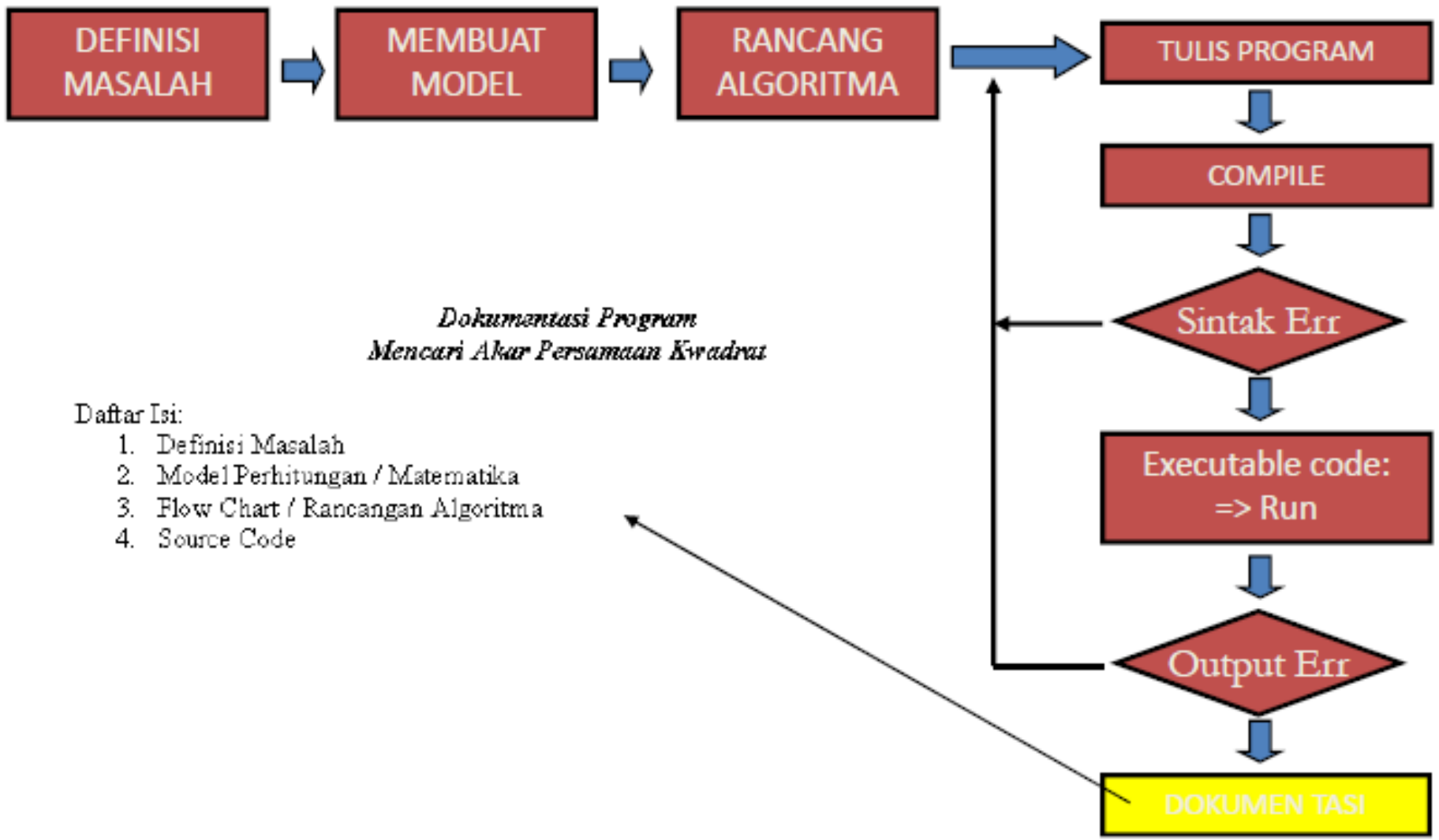
DEFINISI MASALAH → MEMBUAT MODEL → RANCANG ALGORITMA → TULIS PROGRAM → COMPILE → Sintak Err → Executable code: => Run → Output Err → DOKUMEN TASI



Tahapan *problem solving*



Tahapan *problem solving*



Kriteria algoritme yang baik

- Algoritme harus benar, artinya akan memberikan keluaran (*output*) yang dikehendaki atas sejumlah masukan (*input*) yang diberikan
- Algoritme harus mampu memberikan hasil yang sedekat mungkin dengan nilai yang sebenarnya.
- Algoritme harus efisien dalam penggunaan waktu dan memori

- Sifatnya general; bukan hanya untuk menyelesaikan satu kasus saja, tapi juga untuk kasus lain yang lebih general.
- Bisa dikembangkan (*expandable); h*aruslah sesuatu yang dapat dikembangkan lebih jauh berdasarkan perubahan *requirement* yang ada.
- Mudah dimengerti; agar lebih mudah dikoreksi dan di-*maintenance (*kelola*).*
- Portabilitas yang tinggi (*portability); bisa dengan mudah* diimplementasikan di berbagai *platform* komputer.
- Harus *terminate;* harus ada kriteria berhenti.

Teknik/ strategi penyelesaian masalah

- *Top-down* (umum ke khusus)
- *Bottom-up (kecil ke besar)*
- *Exhaustive search* (mencoba semua kemungkinan)
- *Divide and conquer* (dipecah-pecah menjadi beberapa bagian)
- *Greedy* (mencari optimum lokal)
- *Dynamic programming* (memanfaatkan memori)
- *Backtracking* (penelusuran lintasan)
- *Graph* (*vertex & edge*)

Latihan

1. Jelaskan beda antara algoritme dengan program komputer
2. Berikan contoh pemanfaatan algoritme dalam kehidupan kita sehari-hari
3. Uraikan tahapan penyelesaian masalah menggunakan komputer
4. Kriteria apa yang dapat digunakan untuk menilai baik tidaknya sebuah algoritme
5. Rancang sebuah algoritme untuk mengonversi satuan derajat ke satuan derajat yang lain (celcius ke Fahrenheit dan kelvin dan sebaliknya).


Sekian