

# ALGORITMA INDERAJA KELAUTAN

## I. PENDAHULUAN

### 1.1. Asal Usul Kata *Algoritma*

Kata *algoritma* TIDAK berasal dari kata Yunani (Greek) seperti *algos* yang berarti kesakitan akan tetapi merupakan pemelesetan (corruption) dari nama ahli matematika berkebangsaan Persia yaitu “Abu ‘Abd Allah Muhammad ibn Musa al-Khwarizmi” yang diterjemahkan sebagai “Mohammad, bapak dari Adbdulla, anak dari Musa, si Kwarizmian” (Kwarizm merupakan sebuah kota tua yang sekarang terletak di Provinsi Xorazm, Uzbekistan). Al-Khwarizmi dikenal sebagai penulis terkenal risalah *Kitab al-jabr wa'l-Muqabala* dimana kata aljabar moderen diturunkan.

Orang barat membaca *Al-Khuwarizmi* menjadi *Algorism* yang sebenarnya tidak berhubungan dengan kata Yunani *arithmos* yaitu angka akan tetapi dulunya digunakan untuk menerangkan sistem desimal moderen untuk menulis dan memanipulasi angka-angka khususnya dalam angka-angka arab.

Perubahan kata dari *Algorism* menjadi *Algorithm* muncul karena kata *Algorism* sering dipelesetkan (*corrupted*) dengan *Arithmetic*, sehingga akhiran *-sm* berubah menjadi *-thm*. Mengingat perhitungan dengan angka Arab sudah menjadi hal yang biasa, maka lambat laun kata *Algorithm* berangsur-angsur dipakai sebagai metode perhitungan (komputasi) secara umum, sehingga kehilangan makna kata aslinya. Istilah *Algorithm* dalam Bahasa Indonesia disebut menjadi *Algoritma*.

### 1.2. Definisi Algoritma

*Algoritma merupakan suatu urutan langkah-langkah (steps) yang disusun secara logis dan sistematis untuk menyelesaikan suatu masalah dan dapat dieksekusi.*

Dalam defenisi diatas istilah “komputer” tidak terlihat sehingga sebuah algoritma tidak selalu berhubungan dengan komputer. Dalam kehidupan sehari-hari banyak terdapat proses yang dinyatakan dalam suatu algoritma. Contohnya, cara membuat masakan yang dinyatakan dalam suatu resep juga dapat disebut sebagai sebuah algoritma. Pada setiap resep selalu ada urutan langkah-langkah membuat masakan. Bila langkah-langkahnya tidak logis, maka tidak akan dapat dihasilkan masakan yang diinginkan. Ibu-ibu yang mencoba suatu resep masakan akan membaca (mengikuti) satu per satu langkah-langkah pembuatannya lalu ia mengerjakan proses itu sesuai dengan yang ia baca. Secara umum, pihak (benda) yang mengerjakan proses disebut pemroses (*processor*). Pemroses tersebut dapat berupa manusia, komputer, robot atau alat-alat elektronik lainnya. Pemroses melakukan suatu proses dengan melaksanakan atau "mengeksekusi" algoritma yang menjabarkan proses tersebut.

Melaksanakan Algoritma berarti mengerjakan langkah-langkah di dalam Algoritma tersebut. Pemroses mengerjakan proses sesuai dengan algoritma yang diberikan kepadanya. Juru masak membuat kue berdasarkan resep yang diberikan kepadanya, pianis memainkan lagu berdasarkan papan not balok. Karena itu suatu Algoritma

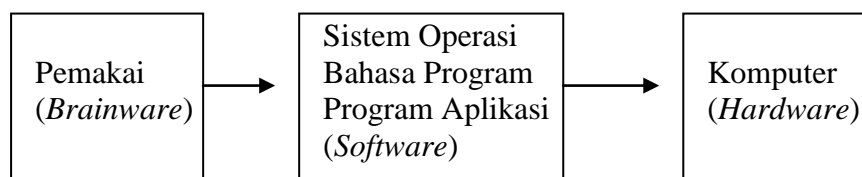
harus dinyatakan dalam bentuk yang dapat dimengerti oleh pemroses. Jadi suatu pemroses harus :

1. Mengerti setiap langkah dalam Algoritma
2. Mengerjakan operasi yang bersesuaian dengan langkah tersebut.

Kata *logis* merupakan kata kunci dalam sebuah Algoritma karena setiap langkah dalam Algoritma harus logis (jelas dan pasti) dan harus dapat ditentukan bernilai salah atau benar. Sebuah algoritma diperbolehkan tanpa ada input tetapi minimal harus ada 1 output. Jumlah langkah (steps) dalam sebuah algoritma harus berhingga atau dengan kata lain harus ada akhir proses.

Umumnya sebuah algoritma sering dikaitkan dengan perhitungan dan mengingat proses perhitungan dilakukan dengan cepat oleh sebuah komputer maka secara umum sekarang ini sebuah algoritma menjadi selalu berkaitan dengan komputer. Komputer merupakan suatu alat elektronika yang mempunyai kemampuan untuk melakukan perhitungan dan membuat keputusan logika dengan jumlah waktu yang jauh lebih cepat dari kemampuan manusia. Pada saat ini, sebuah PC (personal computer) dapat melakukan puluhan juta penjumlahan dalam waktu satu detik. Sebuah output kadangkala memerlukan lebih dari satu algoritma khususnya dalam Penginderaan Jarak Jauh yang umumnya memerlukan banyak algoritma untuk menghasilkan sebuah output sehingga untuk menghasilkan sebuah output yang benar dan cepat maka algoritma dalam Penginderaan Jarak Jauh selalu berhubungan dengan komputer.

Komputer memproses data dengan menggunakan kumpulan instruksi (langkah) yang disebut program. Program ini memandu komputer untuk melaksanakan kerja (mengeksekusi perintah) yang secara sistematis dan berstruktur disusun oleh seseorang yang disebut *programmer*. Program yang disusun secara sistematis dan logis dan dapat menyelesaikan masalah (menghasilkan output) adalah merupakan sebuah algoritma. Secara umum, ada tiga elemen penting yang harus dimiliki mulai dari pembuatan sebuah algoritma (*brainware*), program untuk menjalankan algoritma (*software*), dan alat untuk pemrosesan algoritma (*hardware*) yang secara struktural dapat digambarkan pada Gambar 1.1. dibawah ini.



Gambar 1.1. Tiga elemen penting dalam pemecahan masalah dengan komputer

Perangkat keras (*hardware*) merupakan berbagai peralatan yang terdapat di dalam suatu sistem komputer seperti papan ketik (*keyboard*), layar (*monitor*), disk, memori, dan unit pemroses (CPU=*Central Processing Unit*). Perangkat keras sebuah komputer terdiri dari enam unit (logika) yang saling berhubungan membentuk suatu organisasi komputer yaitu:

- (i) **Unit input**, yang digunakan untuk menerima masukan data dan/atau program komputer dari berbagai alat masukan (input devices) dan menempatkannya ke unit lain sedemikian rupa sehingga dapat diproses.

- (ii) **Unit output**, yang digunakan untuk menyimpan data atau informasi yang telah diproses oleh komputer dan menempatkannya pada berbagai alat output (output devices) sehingga dapat digunakan oleh pengguna.
- (iii) **Unit memori**, yang sering disebut sebagai memori utama digunakan untuk menyimpan atau menampung masukan dari unit input, sehingga data (informasi) tersebut selalu tersedia untuk diproses pada saat dibutuhkan. Unit ini juga menampung hasil proses komputer sampai diperlukan oleh unit output.
- (iv) **ALU(Arithmetic and Logic Unit)**, yang bertugas untuk melakukan operasi aritmetika (perhitungan) seperti penjumlahan, pengurangan, perkalian, dan pembagian. Disamping itu, ALU juga bertugas untuk melakukan keputusan dari suatu operasi logika dasar maupun operasi boolean sesuai dengan instruksi program seperti sama dengan ( $=$ ), tidak sama dengan ( $\neq$ ), kurang dari ( $<$ ), kurang atau sama dengan ( $\leq$ ), lebih besar dari ( $>$ ), dan lebih besar atau sama dengan ( $\geq$ ), dan lain-lain.
- (v) **CPU (Central Processing Unit)** merupakan "brain" atau "processor" dari sebuah komputer yang merupakan koordinator dan penanggung jawab bagi operasi unit-unit lainnya. Unit ini menyampaikan informasi pada unit input, menentukan kapan informasi harus dibaca ke dalam unit memori, dan pada unit output, kapan harus mengirim informasi dari memori ke alat-alat output.
- (vi) **Secondary storage unit**, yaitu media penyimpanan data dan/atau program yang non-volatile berupa *Flash Drive, Optical Disc, Magnetic Disk, Magnetic Tape*. Media ini biasanya daya tampungnya cukup besar dengan harga yang relatif murah.

Program untuk menjalankan suatu komputer disebut dengan perangkat lunak (*software*). Dalam perangkat lunak termasuk sistem operasi, bahasa pemrograman, dan program aplikasi. Sistem operasi (*operating system*) atau *OS* adalah perangkat lunak sistem yang bertugas untuk melakukan layanan inti dan umum maupun kontrol dan manajemen perangkat keras (*hardware*) serta operasi-operasi dasar sistem, termasuk menjalankan software aplikasi seperti program-program pengolah kata dan browser web. Bagian kode yang melakukan tugas-tugas inti dan umum tersebut dinamakan dengan *kernel* suatu Sistem Operasi.

Sistem Operasi bisa juga berfungsi sebagai penghubung antara lapisan *hardware* dan lapisan *software*. Sistem Operasi ini melakukan semua tugas-tugas penting dalam komputer, menjamin aplikasi-aplikasi yang berbeda dapat berjalan secara bersamaan dengan lancar, menjamin aplikasi software lainnya dapat menggunakan memori, melakukan input dan output terhadap peralatan lain, dan memiliki akses kepada sistem file. Bilamana beberapa program aplikasi berjalan secara bersamaan, maka Sistem Operasi akan mengatur jadwal yang tepat, sehingga sedapat mungkin semua proses yang menggunakan prosesor (CPU) akan berjalan dengan lancar dan tidak saling mengganggu.

Dalam banyak kasus, Sistem Operasi menyediakan suatu pustaka dari fungsi-fungsi standar, dimana aplikasi lain dapat memanggil fungsi-fungsi itu, sehingga dalam setiap pembuatan program baru, tidak perlu membuat fungsi-fungsi tersebut dari awal.

Secara umum sebuah Sistem Operasi terdiri dari 5 (lima) bagian:

- (1) Mekanisme *boot* yaitu meletakkan kernel ke dalam memory
- (2) *Kernel* yaitu inti dari sebuah Sistem Operasi
- (3) *Command Interpreter* atau *shell* yang bertugas membaca input dari pengguna
- (4) *Library* (pustaka) yang menyediakan kumpulan fungsi dasar dan standar yang dapat dipanggil oleh aplikasi lain
- (5) *Driver* untuk berinteraksi dengan hardware eksternal, sekaligus untuk mengontrol mereka.

Dewasa ini, sebagian besar Sistem Operasi membolehkan beberapa aplikasi berjalan secara simultan pada waktu yang bersamaan yang disebut sebagai *Multi-tasking Operating System*. Beberapa contoh Sistem Operasi yang berkembang saat ini adalah *MS-DOS*, *Microsoft Windows*, *Unix*, *Linux*, dan *Mac-OS*. Dalam kuliah ini akan dipelajari Sistem Operasi *MS-DOS*, *Microsoft Windows*, dan *Linux*. Sedangkan bahasa pemrograman akan digunakan *Interactive Data Language (IDL) software* yang dikeluarkan oleh *ITTs Inc. Software* ini akan dipelajari dan dikembangkan untuk menghasilkan sebuah algoritma untuk aplikasi penginderaan jarak jauh.

### **1.3. Algoritma Merupakan Jantung Ilmu Komputer/Informatika**

Algoritma adalah jantung ilmu komputer atau informatika. Banyak cabang ilmu komputer yang diacu dalam terminologi algoritma. Akan tetapi, seperti dijelaskan sebelumnya sebuah algoritma tidak selalu identik dengan ilmu komputer. Komputer hanya berfungsi sebagai alat pemroses. Agar proses dapat dilakukan oleh komputer dengan baik, algoritma harus ditulis dalam notasi bahasa pemrograman yang logis dan sistematis serta dapat dieksekusi yang disebut program. Sehingga, sebuah program adalah perwujudan atau implementasi teknis sebuah atau lebih algoritma yang ditulis dalam bahasa pemrograman tertentu sehingga dapat dieksekusi oleh komputer.

Bagi orang yang baru belajar komputer, pemrograman merupakan hal yang sangat sulit dilakukan, sehingga pemrograman dilakukan secara sistematis yang dimulai dari perancangan atau desain langkah-langkah yang tepat untuk memberikan jawaban dari suatu masalah yang dihadapi. Langkah-langkah yang disusun secara sistematis, berstruktur, dan terurut untuk menjawab suatu persoalan dengan menggunakan bahasa manusia inilah yang disebut algoritma. Untuk dapat diproses oleh komputer dengan menggunakan sistem operasi dan *software* tertentu maka instruksi atau langkah-langkah tersebut harus ditulis dalam bahasa pemrograman komputer tertentu.

Belajar membuat sebuah program tidak sama dengan belajar sebuah bahasa program. Belajar membuat sebuah program adalah belajar tentang metodologi pemecahan masalah, kemudian menuangkannya dalam suatu notasi tertentu yang mudah dibaca dan dipahami. Sedangkan belajar sebuah bahasa program berarti belajar memakai suatu bahasa dan aturan-aturan tata bahasanya, instruksi-instruksinya (*commands*), tata cara pengoperasian *compiler*-nya, dan

memanfaatkan instruksi-instruksi tersebut untuk membuat program yang ditulis hanya dalam bahasa itu saja.

Contoh algoritma tanpa menggunakan komputer untuk menyelesaikannya:

Algoritma A: membuat minuman teh manis

1. Tuangkan air panas kedalam cangkir.
2. Tuangkan satu sendok gula kedalam cangkir.
3. Masukkan satu bungkus teh kedalam cangkir.
4. Aduk isi cangkir sampai teh dan gula larut dalam airnya.

Algoritma B: membuat minuman teh manis

1. Tuangkan satu sendok gula kedalam cangkir.
2. Masukkan satu bungkus teh kedalam cangkir.
3. Seduhkan air panas kedalam cangkir.
4. Aduk isi cangkir sampai teh dan gula larut dalam airnya.

Dari contoh Algoritma A dan B diatas, tampak bahwa untuk menyelesaikan suatu masalah yang sama, setiap orang dapat membuat langkah-langkah (algoritma) yang berbeda.

Contoh algoritma dengan bantuan komputer untuk menyelesaikannya:

Algoritma penentuan nilai mutu dan kelulusan mata kuliah inderaja kelautan

1. Baca nilai ujian tengah semester (UTS)
2. Baca nilai ujian akhir semester (UAS)
3. Hitung nilai rata-rata ujian yaitu  $(UTS+UAS)/2.0$
4. Jika nilai rata-rata  $\geq 85$  maka cetak nilai mutu A dan Lulus
5. Jika nilai  $75 \leq \text{rata-rata} < 85$  maka cetak nilai mutu B dan Lulus
6. Jika nilai  $65 \leq \text{rata-rata} < 75$  maka cetak nilai mutu C dan Lulus Bersyarat
7. Jika nilai rata-rata  $< 65$  maka cetak nilai mutu D dan TIDAK LULUS
8. Cetak semua nilai mutu dan kelulusan mahasiswa

Algoritma dapat disajikan dalam 2 (dua) bentuk:

(1) Secara tulisan:

- melalui **struktur bahasa/deskripsi**
- melalui *Pseudocode*

Contoh membuat algoritma menghitung luas lingkaran:

Misal: input =r, output=L

Proses:  $L=\pi*r^2$

Algoritma secara struktur bahasa/deskripsi:

1. tetapkan nilai  $\pi=3.14$
2. bacalah nilai r (jari-jari)
3. hitung luas (L)
4. Tulis L

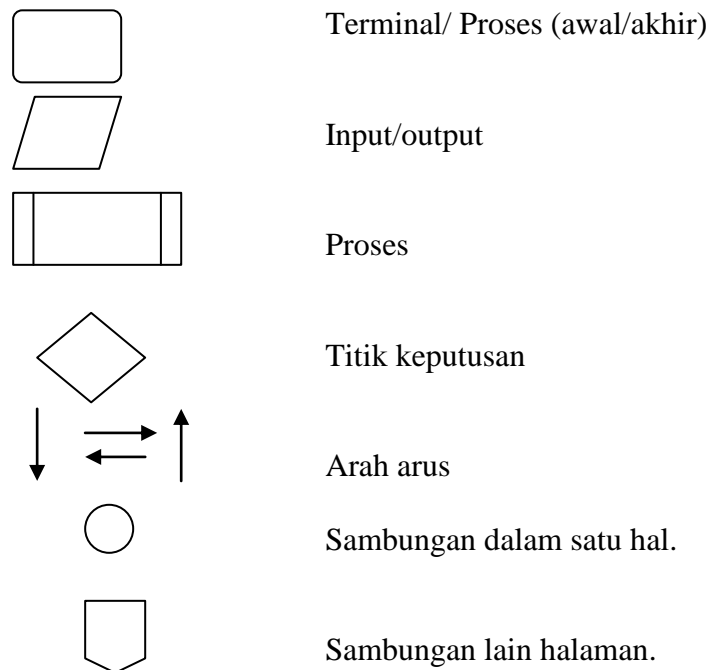
Algoritma Pseudocode:

1.  $\pi=3.14$
2. input, r

3.  $L = \pi * r^2$
4. Print, L

(2) Secara **teknik gambar (flowchart)**

Simbol-simbol yang umum digunakan dalam teknik gambar (flowchart) dalam pembuatan sebuah algoritma adalah sebagai berikut:



Simbol-simbol ini dapat disatukan sesuai dengan fungsinya dan membentuk sebuah bagan alur algoritma. Pembuatan flowchart umumnya dilakukan sebelum menulis program lengkap dalam bentuk pseudocode. Flowchart ini juga dapat berfungsi sebagai guide (arah langkah-langkah) dalam penulisan program lengkap dalam bentuk pseudocode.

### 1.4. Paradigma Pemrograman

Berdasarkan pembahasan sebelumnya bahwa pemrograman merupakan inti dalam menyelesaikan (menjawab) masalah dalam sebuah algoritma yang berkaitan dengan komputer. Seseorang dalam membuat sebuah algoritma atau menyelesaikan sebuah masalah mempunyai cara pandang (metode) tertentu untuk menyelesaikan masalah tersebut yang umum disebut sebagai paradigma pemrograman. Untuk menyelesaikan masalah yang sama bisa diselesaikan dengan cara (metode) yang berlainan sehingga banyak paradigma yang ada dalam pemrograman seperti contoh berikut ini:

#### 1. Pemrograman Prosedural

Pemrograman yang berdasarkan urutan, sekuensial, prosedural dalam penyelesaian sebuah masalah. Program disini merupakan suatu rangkaian prosedur (instruksi) untuk memanipulasi data dan dikerjakan secara berurutan.

## 2. Pemrograman Fungsional

Pemrograman yang berdasarkan teori fungsi matematika sebagai dasar utama program dalam penyelesaian masalah.

## 3. Pemrograman Terstruktur

Pemrograman yang dilakukan secara berurutan dan terstruktur yang dibagi-bagi menjadi prosedur dan fungsi dalam menyelesaikan sebuah masalah.

## 4. Pemrograman Modular

Pemrograman yang dibentuk dalam berbagai modul yang merupakan kumpulan dari prosedur dan fungsi yang berdiri sendiri dalam menyelesaikan sebuah masalah.

## 5. Pemrograman Berorientasi Obyek

Pemrograman yang didasarkan pada prinsip obyek, dimana obyek memiliki data/variabel/property dan method/prosedur yang dapat dimanipulasi. Objek dalam hal ini merupakan representasi objek dalam dunia nyata.

## 6. Pemrograman Berorientasi Fungsi

Pemrograman yang berfokus pada suatu fungsi tertentu dalam menyelesaikan sebuah masalah.

## 7. Pemrograman Deklaratif

Pemrograman yang didasarkan pada penyelesaian masalah sebagai hasil inferensi terhadap fakta dan aturan yang diberikan. Program disini lebih mendeskripsikan suatu masalah dengan pernyataan daripada memecahkan masalah dengan implementasi algoritma.

Sebuah algoritma harus mengikuti kriteria berikut ini:

1. **Input:** sebuah algoritma harus memiliki input (masukan) dan nilai input ini dapat bernilai nol atau lebih yang berasal dari masukan dari luar.
2. **Output:** sebuah algoritma harus memiliki minimal satu buah output (keluaran).
3. **Definiteness** (kepastian): sebuah algoritma memiliki instruksi-instruksi yang jelas (pasti) dan tidak ambigu.
4. **Finiteness** (ada batas): sebuah algoritma harus memiliki titik berhenti (END atau stopping role).
5. **Effectiveness** (tepat dan efisien): sebuah algoritma harus efisien dan dapat dieksekusi (dilaksanakan).

Berdasarkan prosesnya, algoritma dibedakan menjadi:

1. **Sequence Process:** algoritma/instruksi yang dikerjakan secara sekuensial atau berurutan.
2. **Selection Process:** algoritma/instruksi yang dikerjakan dengan seleksi jika memenuhi kriteria tertentu
3. **Iteration Process:** algoritma/instruksi yang dikerjakan jika memenuhi suatu kondisi tertentu.
4. **Concurrent Process:** beberapa algoritma/instruksi yang dikerjakan secara bersama.

## 1.5. Hal-hal Penting dalam Pengembangan Algoritma

Untuk menghasilkan sebuah algoritma yang baik maka hal-hal berikut ini merupakan bagian dari pengembangan algoritma tersebut:

**1. Mendefinisikan dan menentukan masalah.** Hal pertama yang perlu dilakukan dalam pengembangan sebuah algoritma adalah pendefinisian atau penentuan masalah. Tentukan masalahnya, apa saja yang harus diselesaikan dengan menggunakan komputer, dan apa masukan (input) serta keluarannya (output).

**2. Menemukan solusi.** Setelah masalah dapat didefinisikan atau ditemukan, maka langkah berikutnya adalah menentukan langkah-langkah untuk memecahkan permasalahan tersebut atau menghasilkan jawaban (solusi). Jika masalah terlalu kompleks, maka sebaiknya masalah tersebut dipecah menjadi modul-modul yang lebih kecil agar lebih mudah diselesaikan. Dengan penggunaan sub-modul ini maka program utama akan menjadi lebih singkat dan mudah dilihat.

**3. Memilih algoritma.** Pilihlah algoritma yang benar-benar sesuai dan efisien untuk dapat menyelesaikan permasalahan tersebut.

**4. Menulis program.** Dalam menulis program, sebaiknya dipilih bahasa komputer yang mudah dipelajari, mudah digunakan, sudah dikuasai, dan memiliki tingkat kompatibilitas tinggi dengan perangkat keras/platform komputer lainnya.

Sebuah algoritma harus ditulis dengan jelas dan cara terbaik untuk menuliskannya adalah dengan menggunakan pseudocode. Pseudocode menggunakan struktur formal bahasa pemrograman dan matematika dalam merinci algoritma tersebut kedalam kalimat pada setiap langkah. Kalimat-kalimat ini ditulis menggunakan matematik, Bahasa Inggris murni atau percampuran dari keduanya. Persisnya bagaimana seseorang membuat struktur pseudocode merupakan pilihan seseorang akan tetapi tujuan akhir penulisan program harus jelas dan presise. Untuk menghasilkan program yang baik langkah-langkah berikut ini perlu diperhatikan:

- (i) Gunakan kata kunci standar keharusan pemrograman (contohnya *if/then/else*, *while*, *for*, *repeat/until*, *case*, *return*) dan notasi (*variable*  $\leftarrow$  *value*, *array[index]*, *pointer*  $\rightarrow$  *field*, *function(args)*, *etc.*)
- (ii) Struktur blok harus terlihat dengan jelas dalam algoritma dengan membuat “indent” dengan hati-hati dan konsisten pada satu kesatuan perintah.
- (iii) Jangan membuat kata kunci dalam font atau gaya yang berbeda. Mengganti jenis font atau gaya penulisan dalam algoritma khususnya dalam kata-kata kunci dapat mengiring pembaca pada pengertian yang berbeda dalam arti yang sesungguhnya sama.
- (iv) Sebaiknya setiap pernyataan (perintah) termuat dalam satu baris, dan setiap baris harus berisi satu pernyataan, kecuali pada pernyataan pendek seperti  $i + 1; j - 1; k = 0$ .
- (v) Letakkan setiap struktur pernyataan (*for*, *while*, *if*) di barisnya masing-masing. Urutan struktur pernyataan ini sangat penting sekali dan harus dibuat sejelas mungkin.
- (vi) Gunakan istilah singkat tapi yang membantu menghafal algoritma dan variabel nama.

- (vii) Selalu tambahkan catatan tambahan (*non commands*) untuk menerangkan arti dari setiap sub-program untuk mempermudah pembaca (*programmer*) pada saat pemeriksaan (*debugging*).

Deskripsi yang baik dari suatu algoritma mengungkap struktur internal, menyembunyikan rincian yang tidak relevan, dan dapat diimplementasikan dengan mudah oleh setiap programmer dalam berbagai bahasa pemrograman, biarpun kadang-kadang tidak dipahami kenapa algoritma tersebut bekerja. Pseudocode yang baik seperti kode yang baik, membuat algoritma lebih mudah dipahami dan dianalisa, juga membuat kesalahan lebih mudah untuk ditandai.

**5. Menguji program.** Setelah program selesai ditulis, maka langkah berikutnya adalah menguji program tersebut dengan segala macam kemungkinan yang ada, termasuk error yang ditimbulkan sehingga program tersebut akan benar-benar handal dan layak digunakan.

**6. Menulis dokumentasi.** Menulis dokumentasi (komentar) pada bagian program sangat penting agar pada suatu saat jika kita akan melakukan perubahan atau membaca *source code* yang sudah kita tulis dapat kita ingat-ingat lagi dan kita akan mudah membacanya. Komentar kecil yang ditulis adalah tentang apa maksud kode tersebut, untuk apa, variabel apa saja yang digunakan, untuk apa, dan parameter-parameter yang ada pada suatu prosedur dan fungsi.

**7. Merawat program.** Program yang sudah jadi perlu dirawat untuk mencegah munculnya *bug* yang sebelumnya tidak terdeteksi atau bilamana pengguna membutuhkan fasilitas baru yang berbeda dari sebelumnya.

## 1.6. Running Time Algoritma/Program

Running time sebuah algoritma/program adalah waktu yang dibutuhkan sebuah algoritma dengan bantuan komputer untuk menyelesaikan suatu permasalahan. Semakin baik dan efisien sebuah algoritma maka running time yang dibutuhkan menjadi semakin singkat. Dengan meningkatnya kemampuan dan kualitas berbagai komputer maka diharapkan running time sebuah algoritma akan menjadi lebih efisien.

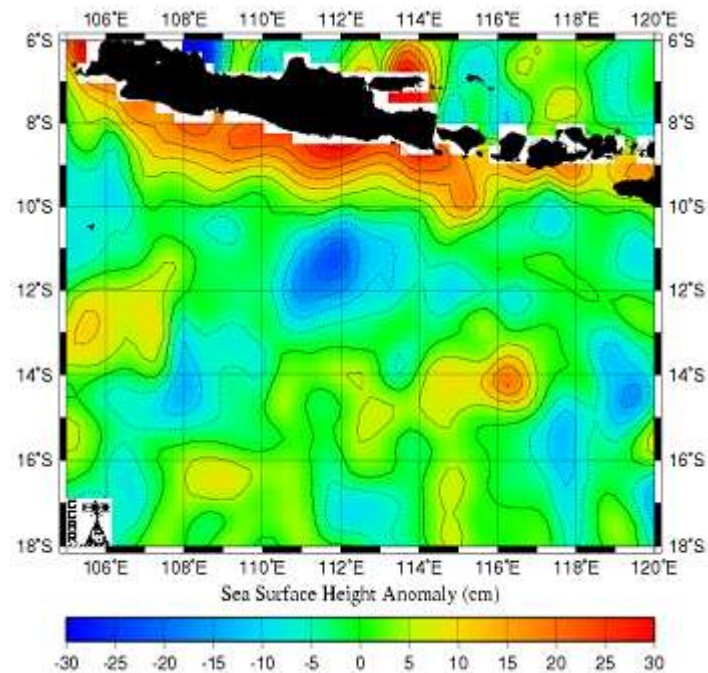
## 1.7. Kenapa Kita Perlu Mempelajari Algoritma dalam Inderaja Kelautan?

Kesahihan algoritma dalam Penginderaan Jauh Kelautan sangat menentukan tingkat akurasi nilai hasil unsur-unsur yang diduga dibandingkan dengan nilai sebenarnya di lapangan. Dalam bidang ini, sangat banyak faktor-faktor yang mempengaruhi keberhasilan dan keakuratan algoritma dalam menduga nilai terduga sehingga inti keberhasilan (akurasi) pendugaan nilai hasil unsur terduga dalam bidang ini sangat ditentukan oleh kesahihan algoritma yang diaplikasikan kedalamnya. Dengan kata lain, mempelajari penginderaan jauh kelautan berarti mempelajari algoritma yang diaplikasikan didalamnya.

Secara umum dalam kelas algoritma komputer akan dipelajari dua keterampilan yang sangat penting yaitu: cara untuk berpikir tentang algoritma dan cara untuk berbicara tentang algoritma. Dalam program ini diharapkan juga dapat membantu mahasiswa mengembangkan intuisi yang bersifat algoritma. Bagaimana berbagai algoritma dapat

bekerja? Bilamana kita melihat sebuah masalah untuk pertama kalinya, maka kita harus berpikir bagaimana untuk menyelesaikan masalah itu? Bagaimana kita mengetahui teknik apa yang akan bekerja dengan baik, dan teknik mana yang bekerja paling baik? Bagaimana kita dapat menilai apakah satu algoritma lebih baik daripada yang lain? Bagaimana kita dapat menyatakan bahwa kita memiliki solusi (jawaban) terbaik untuk penyelesaian masalah tersebut?

Tujuan lain dari program ini adalah untuk membantu mahasiswa mengembangkan bahasa logaritma yang dalam hal ini adalah bahasa Interactive Data Language (IDL) yang diterbitkan oleh ITTS solution. Kita tidak cukup hanya untuk memahami bagaimana untuk memecahkan suatu masalah, namun harus juga dapat menjelaskan solusi untuk orang lain, dapat mengerti dan memperbaiki *source code* orang lain. Disamping itu dalam program ini akan dipelajari juga proses pengembangan algoritma yang tercakup dalam penginderaan jauh kelautan. Berikut ini adalah sebuah contoh anomali ketinggian muka air laut di bagian selatan Laut Jawa yang diestimasi dari data satelit TOPEX/POSEIDON. Estimasi ketinggian muka air laut dari satelit secara konseptual adalah simple, namun dalam kenyataannya proses estimasi ini sangatlah kompleks, dimana untuk menghasilkan estimasi anomali ketinggian muka air laut seperti pada Gambar 1.2 membutuhkan sekitar 60 buah algoritma dan berbagai pengukuran lapangan untuk menghasilkan estimasi yang akurat ketinggian muka air laut yang berhubungan dengan dinamika arus laut.



Gambar 1.2. Anomali Ketinggian Muka Air Laut Hasil Estimasi dari Satelit TOPEX/POSEIDON Wilayah Selatan Lautan Hindia untuk Tanggal 15 November 1998 (courtesy Robert Leben, CCAR).

## Soal Latihan Algoritma

1. Buatlah algoritma untuk menghitung konversi suhu.dari Celcius menjadi Reamur dan Fahrenheit.

Input: suhu dalam Celcius

Proses:  $R = 4/5 * C$  dan  $F = 9/5 * C + 32$

Output: suhu dalam Reamur dan Fahrenheit

2. Buatlah algoritma untuk mencari sisi miring dari suatu segitiga siku-siku, jika diketahui panjang sisi yang membentuk sudut siku-siku.

Input: a dan b, yaitu panjang sisi pembentuk sudut siku-siku

Proses:  $c = a^2 + b^2$

Ouput: sisi miring (c)

3. Buatlah algoritma untuk menentukan suatu bilangan genap atau ganjil dari bilangan 1 ...20

Input: suatu bilangan

Ouput: genap / ganjil

4. Buatlah algoritma untuk menentukan suatu bilangan adalah bilangan prima.