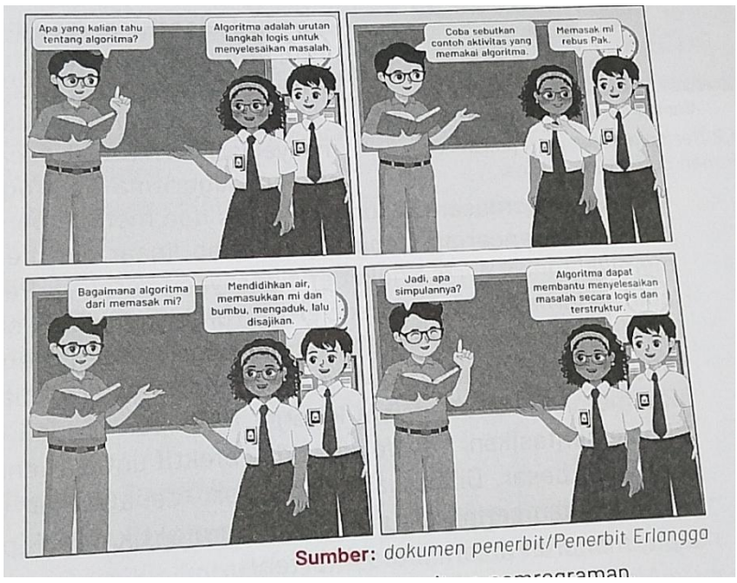
BAB III

ALGORITMA PEMROGRAMAN

Algoritma sering ditemui dalam kehidupan sehari-hari, seperti ketika memasak mi rebus. Algoritma adalah urutan langkah-langkah logis yang digunakan untuk menyelesaikan masalah. Pada dunia digital, algoritma berguna untuk menyelesaikan masalah yang diimplementasikan dalam bentuk bahasa pemrograman. Untuk lebih memahami mengenai algoritma dan penerapannya dalam masalah kehidupan sehari-hari ataupun dalam pemrograman, simak, pelajari, dan lakukan aktivitas yang tersedia pada bab ini.



Sumber: dokumen penerbit/Penerbit Erlangga
Gambar 3.1 Percakapan mengenai algoritma pemrograman

A. Algoritma Tingkat Menengah

Algoritma adalah langkah-langkah terstruktur yang berfungsi untuk menyelesaikan permasalahan. Algoritma sangat erat kaitannya dengan pemrograman karena algoritma menjadi dasar dalam menyusun instruksi-instruksi dalam program komputer untuk menyelesaikan suatu permasalahan. Perhatikan penjelasan pada materi berikut untuk memahami lebih lanjut keterkaitan antara algoritma dan pemrograman.

1. Memahami Konsep Berpikir Algoritma

Algoritma berasal dari nama matematikawan muslim, yaitu Al Khawarizmi. Algoritma adalah serangkaian langkah terstruktur dan logis untuk menyelesaikan masalah atau mencapai tujuan. Pada konteks ilmu komputer dan matematika, algoritma berfungsi sebagai instruksi sistematis yang dapat diterapkan di berbagai bidang. Tiap individu dapat memiliki algoritma yang berbeda untuk menyelesaikan masalah yang sama asalkan hasil akhirnya tetap konsisten.

Algoritma memiliki kemampuan untuk menyelesaikan masalah secara efektif dan efisien. Algoritma pemrograman dapat dibedakan berdasarkan fungsionalitas dan metode penyelesaiannya. Algoritma pencarian, seperti pencarian linear dan pencarian biner memiliki fungsi serta tingkat efisiensi yang berbeda. Pencarian linear lebih sederhana, tetapi kurang efisien untuk dataset besar. Sementara itu, algoritma pencarian biner lebih cepat, tetapi memerlukan data yang terurut. Selain algoritma pencarian, algoritma pengurutan juga memiliki variasi dalam kinerja. Meskipun mudah dipahami dan diimplementasikan, *Bubble sort* tidak efektif untuk menangani jumlah data yang besar. Di sisi lain, *quick sort* menawarkan efisiensi yang lebih baik dan sering digunakan dalam praktik, meskipun kinerjanya dapat menurun dalam situasi tertentu.

Algoritma pemrograman dinamis dan graf merupakan dua pendekatan penting dalam menyelesaikan masalah kompleks dalam pemrograman. Pemrograman dinamis dengan teknik *memoization* meningkatkan efisiensi perhitungan untuk berbagai masalah, seperti Fibonacci dan *Knapsack*. Di sisi lain, algoritma graf, seperti Dijkstra dan BFS, membantu dalam analisis data dengan menemukan jalur terpendek dalam graf berbobot dan tidak berbobot. Pemilihan algoritma dalam kondisi tepat sangat bergantung pada kebutuhan spesifik dari

Algoritma pemrograman dinamis dan graf merupakan dua pendekatan penting dalam menyelesaikan masalah kompleks dalam pemrograman. Pemrograman dinamis dengan teknik *memoization* meningkatkan efisiensi perhitungan untuk berbagai masalah, seperti Fibonacci dan *Knapsack*. Di sisi lain, algoritma graf, seperti Dijkstra dan BFS, membantu dalam analisis data dengan menemukan jalur terpendek dalam graf berbobot dan tidak berbobot. Pemilihan algoritma dalam kondisi tepat sangat bergantung pada kebutuhan spesifik dari



Sumber: *shutterstock.com/Trump card*

Gambar 3.2 Bermain catur merupakan salah satu penerapan berpikir algoritma.

masalah yang dihadapi. Sebagai contoh, algoritma Dijkstra efektif digunakan untuk graf tanpa bobot negatif, sedangkan BFS ideal untuk graf tanpa bobot.

Perbandingan berbagai jenis algoritma menunjukkan bahwa mereka memiliki kegunaan dan kekuatan tertentu sesuai dengan konteks masalah. Algoritma pencarian dan pengurutan merupakan dasar, sedangkan algoritma pemrograman dinamis dan graf menawarkan solusi untuk tantangan yang lebih kompleks. Memahami manfaat dan kelemahan tiap algoritma adalah kunci dalam memilih metode yang paling efektif untuk pengembangan perangkat lunak dan analisis data. Perhatikan ciri-ciri, karakter, dan sifat algoritma berikut.

Algoritma memiliki ciri-ciri khas, seperti *input*, *output*, instruksi yang jelas, jumlah langkah terbatas, serta efektif dan efisien dalam menyelesaikan masalah. Karakter algoritma mencakup adanya hasil akhir, langkah-langkah yang terstruktur, dan proses yang harus selesai dalam waktu tertentu. Sifat algoritma menuntut kejelasan instruksi. *Input* dan *output* dalam algoritma harus memiliki batasan serta tujuan yang jelas. Penjelasan lebih lanjut mengenai ciri-ciri, karakter, dan sifat algoritma dapat dipelajari pada kode QR di samping.

Contoh Algoritma dalam Kehidupan Sehari Hari

1. Buka situs *e-banking*.
2. *Login* dengan *password*/*pin* atau sidik jari.
3. Pilih *transfer*.
4. Tulis nomor rekening tujuan dan besar nominal uang.
5. Memasukkan kode keamanan atau *pin*
6. Proses *transfer* dijalankan setelah persetujuan data.
7. Laporan bukti *transfer* akan ditampilkan di layar.

2. Mengaplikasikan Berpikir Algoritma

Berpikir algoritma merupakan salah satu elemen dalam berpikir komputasional. Berpikir komputasional terdiri atas enam tahapan yang sistematis. Tahap pertama adalah deteksi masalah untuk memahami dan mengidentifikasi sumber permasalahan. Tahap ini diikuti oleh dekomposisi, yaitu proses memecah masalah besar menjadi bagian-bagian yang lebih sederhana agar fokus penyelesaiannya menjadi lebih terarah. Representasi visual dari pemecahan masalah membantu dalam memprioritaskan elemen penting sehingga memudahkan proses penyelesaian. Setelah itu, penyusunan algoritma dilakukan untuk merumuskan langkah-langkah logis dalam menyelesaikan masalah,

yang kemudian diikuti dengan implementasi solusi dalam praktik. Tahap terakhir adalah evaluasi, yang penting untuk menilai efektivitas solusi dan melakukan penyesuaian yang diperlukan.

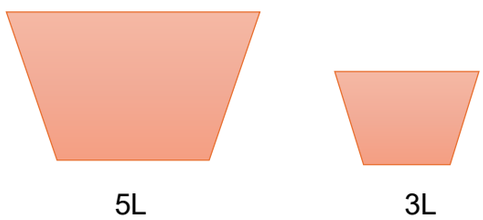
Proses pemecahan masalah dimulai dengan identifikasi masalah dan dilanjutkan dengan penyusunan algoritma. Setelah itu, tahap implementasi melibatkan penerapan algoritma yang telah disusun untuk menghasilkan solusi. Hubungan antara algoritma, masalah, dan solusi dapat dijelaskan pada gambar berikut.



Sumber: dokumen penulis
Gambar 3.3 Tahapan mengaplikasikan berpikir algoritma.

Secara keseluruhan algoritma yang digunakan untuk memecahkan masalah melibatkan langkah-langkah sistematis yang jelas dan terurut. Studi kasus dengan menggunakan contoh nyata dalam kehidupan sehari-hari merupakan salah satu cara merangsang berpikir algoritma. Perhatikan studi kasus berikut sebagai latihan dalam mengaplikasikan berpikir algoritma.

a. Studi kasus perbandingan algoritma memindahkan air



Sumber: dokumen penulis/HS
Gambar 3.4 Ilustrasi ember yang digunakan.

Rancanglah algoritma dan program untuk mengambil air sebanyak 4 L dari sebuah danau, dengan hanya menggunakan dua ember yang memiliki kapasitas masing-masing 5 L dan 3 L. Bagaimana cara untuk mendapatkan tepat 4 L air tanpa menggunakan alat tambahan?

Untuk menyelesaikan studi kasus ini, terdapat beberapa tahapan, yaitu sebagai berikut.

1) **Masalah** Masalah dalam studi kasus ini adalah untuk mendapatkan 4 L air dari danau dengan hanya ada ember berkapasitas 5 L dan 3 L seperti pada ilustrasi. Jadi diperlukan algoritma untuk mendapatkan jumlah alir yang sesuai dari ember yang ada.
2) Terdapat dua opsi algoritma yang dapat dipilih seperti yang terdapat pada tabel berikut.

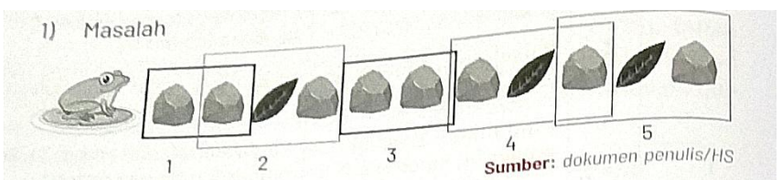
No	Algoritma 1	Algoritma 2
1	Isi ember 5 L	Isi ember 3 L penuh air
2	Pindahkan isinya ke ember 3 L	Pindahkan air dari ember 3 L ke dalam ember 5 L
3	Selanjutnya, buang air yang di ember 3 L	Isi kembali ember 3 L penuh air
4	Pindahkan sisa air di ember 5 L ke ember 3 L	Pindahkan air dari ember 3 L ke ember 5 L hingga penuh
5	Selanjutnya, isi penuh lagi ember yang 5 L	Kosongkan seluruh air dari ember 5 L
6	Selanjutnya, pindahkan sebagian ke ember 3 L yang sudah terisi 2 L	Tuangkan air dari ember 3 L ke ember 5 L
7	Sisa air di ember 5 L adalah 4 L	Penuhi ember dengan 3 L dengan air
8		Pindah air dari ember 3 L ke dalam ember 5 L

3) Solusi dari perbandingan algoritma

Berdasarkan algoritma yang telah dibuat pada tahapan sebelumnya, terdapat dua algoritma yang dapat menyelesaikan masalah. Algoritma yang dipilih sebagai solusi adalah algoritma yang paling efisien. Algoritma 1 dipilih sebagai solusi karena hanya membutuhkan 7 langkah untuk menyelesaikan masalah. Oleh karena itu, untuk mendapatkan 4 L air dari danau, dapat menggunakan langkah-langkah yang ada di algoritma 1.

b. Studi kasus perbandingan algoritma untuk masalah katak menyeberangi sungai

Untuk menyelesaikan studi kasus ini, terdapat beberapa tahapan, yaitu sebagai berikut.



Sumber: dokumen penulis/HS

Gambar 3.6 Ilustrasi langkah-langkah yang dapat diambil.

Kotak nomor 2, 4, dan 5 sudah menjadi jalur yang pasti, jadi tidak ada perbedaan dengan jalur yang lain agar katak tidak melompat mengenai daun. Selanjutnya hitung cara yang digunakan pada kotak 1 dan 3. Pada kotak 1 ada 2 cara, yaitu ke batu 1 atau langsung ke batu 2. Pada kotak 3 ada 3 cara, yaitu batu 4 loncat ke batu 5, 6, dan 7; batu 4 loncat ke batu 5 dan 7; atau batu 4 loncat ke batu 6 dan 7. Jadi diperoleh 6 cara yang dapat digunakan oleh katak.

2 Algoritma

Untuk memudahkan penulisan algoritma, tiap benda diberi nomor 0-11. 0 adalah posisi katak dan 11 adalah batu terakhir. Terdapat 6 opsi algoritma yang dapat dipilih seperti yang terdapat pada tabel berikut.

Tabel 3.2 Langkah yang dapat diambil katak.

No	Perbandingan Langkah Algoritma	Jumlah Langkah
1	0 > 1> 2> 4> 5> 6> 7> 9 >11	**8**
2	0 > 1> 2> 4> 5> 7> 9 >11	**7**
3	0 > 1> 2> 4> 6> 7> 9 >11	**7**
4	0 > 2> 4> 5> 6> 7> 9 >11	**7**
5	0 > 2> 4> 5> 7> 9 >11	**6**
6	0 > 2> 4> 6> 7> 9 >11	**6**

3 Solusi dari perbandingan algoritma

Berdasarkan algoritma yang telah dibuat pada tahapan sebelumnya, algoritma yang dipilih sebagai solusi adalah algoritma yang paling efisien. Algoritma 5 dan 6 dipilih sebagai solusi karena hanya membutuhkan 6 langkah untuk menyelesaikan masalah