



Perkenalan

Tim Olimpiade Komputer Indonesia

Perkenalan

Selamat datang di topik **Pemrograman Kompetitif Dasar!**

Anda diharapkan telah menguasai pemrograman dasar dan mampu:

- Mengetahui setidaknya satu bahasa pemrograman.
- Mampu membaca dan menulis program.
- Mampu memahami **pseudocode**.



Pseudocode

- Merupakan bahasa informal serupa dengan bahasa pemrograman untuk mendeskripsikan program.
- Biasa digunakan pada materi pembelajaran algoritma.
- Pseudocode sendiri bukanlah bahasa pemrograman sungguhan.



Contoh Pseudocode

INSERTIONSORT(*A*)

```
1  for i = 2 to A.length
2      j = i
3      while (j > 1) and (A[j] < A[j - 1])
4          SWAP(A[j], A[j - 1])
5          j = j - 1
```

- Memahami pseudocode pada awalnya mungkin sulit.
- Seiring berjalannya waktu, Anda akan terbiasa dan memahami betapa mudahnya menggunakan pseudocode.



Tentang Pemrograman Kompetitif

Pemrograman Kompetitif

Competitive programming is solving well-defined problems by writing computer programs under specified limits.

-Ashar Fuadi-



Tentang Pemrograman Kompetitif (lanj.)

- Pemrograman kompetitif sering dijadikan ajang "adu otak" dan asah kemampuan *problem solving*.
- Peserta ditantang untuk:
 - Menganalisa permasalahan
 - Merancang algoritma solusi
 - Menuliskannya dalam bentuk program



Ajang Pemrograman Kompetitif - Nasional

- Di Indonesia, pemrograman kompetitif menjadi konsep dalam Olimpiade Sains Nasional (OSN) bidang komputer/informatika.
- Selain OSN, terdapat kompetisi tingkat nasional yang diselenggarakan beberapa Universitas di Indonesia, seperti CompFest (UI), NPC (ITS), BNPCHS (Binus), dan ILPC (Ubaya).
- Sebagai sarana berlatih, ada juga TOKI Open Contest yang biasa dilaksanakan per bulan.



Ajang Pemrograman Kompetitif - Internasional

- International Olympiad in Informatics (IOI) merupakan kompetisi bagi siswa SMA dari seluruh dunia.
- Untuk mahasiswa, terdapat ACM-ICPC (ACM International Collegiate Programming Contest) yang pesertanya terdiri dari tim-tim beranggotakan tiga orang.



Ajang Pemrograman Kompetitif - Lainnya

- Terdapat pula kompetisi tingkat regional yang diselenggarakan bagi negara-negara dalam suatu bagian, seperti Asia-Pacific Informatics Olympiad (APIO).
- Untuk sekedar hobi dan latihan rutin, Anda dapat mengikuti [Codeforces](#), [Topcoder](#), dan [Codechef](#).



Contoh Pemrograman Kompetitif

- Masalah yang diberikan adalah *well-defined problems*.
- *Well-defined problem* adalah sebuah masalah yang telah terdefinsi dengan baik, seperti asumsi yang diperlukan dan batasan masalah.
- Solusi atas masalah ditulis dalam bentuk program program dan memenuhi batas-batas yang ditentukan.
- Batas yang ditentukan: waktu, memori, dan lainnya.



Contoh Soal Pemrograman Kompetitif

- Terdapat N buah ruangan yang dinomori dari 1 sampai N
- Ruangan ke- i memiliki sebuah lampu dan sebuah tombol.
- Bila tombol itu ditekan, keadaan lampu pada seluruh ruangan ke- x akan berubah (dari mati menjadi menyala, atau sebaliknya), yang mana x habis dibagi i .
- Contoh, bila $N = 10$ dan tombol di ruangan ke-2 ditekan, maka keadaan lampu pada ruangan 2, 4, 6, 8, dan 10 akan berubah.
- Bila pada awalnya seluruh lampu berada pada keadaan mati, dan tombol pada setiap ruangan ditekan tepat sekali, bagaimanakah keadaan lampu pada ruangan ke- N ?



Contoh Soal Pemrograman Kompetitif (lanj.)

- Batas waktu: 1 detik.
- Batas memori: 32 MB.
- Diketahui $1 \leq N \leq 10^{14}$.



Solusi Naif

Salah satu cara penyelesaiannya adalah dengan mensimulasikan skenario pada deskripsi soal:

- Mulai dengan ruangan ke-1, dipastikan keadaan lampu pada ruangan ke- N akan berubah (N habis dibagi 1).
- Lanjut ke ruangan ke-2, periksa apakah 2 habis membagi N . Bila ya, ubah keadaan lampunya.
- Lanjut ke ruangan ke-3, periksa apakah 3 habis membagi N . Bila ya, ubah keadaan lampunya.
- ... dan seterusnya sampai ruangan ke- N .



Solusi Naif (lanj.)

- Setelah selesai disimulasikan, tinggal keadaan lampu ruangan ke- N dan cetak jawabannya.
- Kompleksitas solusi ini adalah $O(N)$, dan hanya akan bekerja untuk nilai N yang kecil.
- Untuk N yang lebih besar, misalnya $N = 10^9$, kemungkinan besar diperlukan waktu lebih dari 1 detik.



Solusi yang Lebih Baik

- Dengan observasi, yang sebenarnya perlu dilakukan adalah memeriksa banyaknya pembagi dari N .
- Apabila banyaknya pembagi ganjil, berarti pada akhirnya lampu di ruangan ke- N akan menyala.
- Bila genap, berarti lampu di ruangan ke- N akan tetap mati.



Solusi yang Lebih Baik (lanj.)

- Untuk mencari banyaknya pembagi dari N dengan lebih efisien, lakukan faktorisasi prima terlebih dahulu.
- Misalkan $N = 12$, maka faktorisasi primanya adalah $2^2 \times 3$.
- Untuk menjadi pembagi dari 12, suatu bilangan hanya boleh:
 - Memiliki faktor 2 maksimal sebanyak 2.
 - Memiliki faktor 3 maksimal sebanyak 1.
 - Tidak boleh memiliki faktor lainnya.



Solusi yang Lebih Baik (lanj.)

Sebagai contoh, berikut daftar seluruh pembagi dari 12:

- $1 = 2^0 \times 3^0$
- $2 = 2^1 \times 3^0$
- $3 = 2^0 \times 3^1$
- $4 = 2^2 \times 3^0$
- $6 = 2^1 \times 3^1$
- $12 = 2^2 \times 3^1$



Solusi yang Lebih Baik (lanj.)

- Banyaknya pembagi dari 12 sebenarnya sama saja dengan banyaknya kombinasi yang bisa dipilih dari 2^0 , 2^1 , 2^2 dan 3^0 , 3^1 .
- Banyaknya kombinasi sama dengan mengkalikan banyaknya elemen pada tiap-tiap himpunan.
- Sehingga banyaknya cara ada $3 \times 2 = 6$ cara.



Solusi yang Lebih Baik (lanj.)

- Cara ini juga berlaku untuk nilai N yang lain.
- Misalnya $N = 172.872 = 2^3 \times 3^2 \times 7^4$.
- Berarti banyak pembaginya adalah $4 \times 3 \times 5 = 60$.



Solusi yang Lebih Baik (lanj.)

- Secara umum, banyaknya pembagi dari:

$$N = a_1^{p_1} \times a_2^{p_2} \times a_3^{p_3} \times \dots \times a_k^{p_k}$$

adalah:

$$(1 + p_1) \times (1 + p_2) \times (1 + p_3) \times \dots \times (1 + p_k)$$

- Jadi cukup faktorkan N , lalu periksa banyak pembaginya.
- Faktorisasi bilangan bisa diimplementasikan dengan efisien dan 1 detik cukup untuk N sampai 10^{14} .



Manfaat Pemrograman Kompetitif

- Mengasah kemampuan menganalisa permasalahan dan pemecahan masalah.
- Bertemu dengan teman-teman sehobi!
- Kadang, soal *interview* untuk masuk ke perusahaan teknologi terkemuka juga membutuhkan kemampuan *problem solving*.



Dan Tentunya...

Menantang dan menyenangkan!



Latihan

- Sebagai pemanasan, silakan mengerjakan soal latihan yang diberikan.
- Baca juga beberapa materi pengayaan yang diberikan.

