



9th NPLC

National Programming & Logic Competition



Guide

Book

1. Panduan Soal “Competitive Programming”

1.1 Penjelasan Umum Soal Programming

Pada soal Competitive Programming, soal hanya berupa soal pemrograman, Anda diminta untuk mengirimkan jawaban berupa file coding (script). Jawaban bisa dalam bentuk Java, Pascal, dan C++. Bahasa ini pun memiliki potensi untuk bertambah.

Pada babak penyisihan (online competition), soal terdiri atas 6 soal dalam bentuk pemrograman yang harus dijawab dalam waktu yang ditentukan. Pada babak ini, tim anda diminta untuk berkumpul di salah satu tempat yang terkoneksi internet. Jawaban dikirimkan ke website NPLC dimana anda harus login terlebih dahulu dengan menggunakan username dan password yang sudah diberikan. Jawaban yang dinilai adalah jawaban yang tersimpan di database server.

Pada babak final (online competition), soal terdiri atas 8 soal dalam bentuk pemrograman yang harus dijawab dalam waktu yang ditentukan. Pada babak ini, tim anda diminta untuk berkumpul dalam zoom meeting yang sudah disediakan panitia. Pengiriman jawaban tetap menggunakan website nplc.

1.2 Problem Statement

Menjelaskan detail masalah yang harus diselesaikan.

1.3 Input output specification

Menjelaskan detail mengenai format input dan output termasuk kriteria / batasan input yang harus dapat dijalankan oleh program.

Catatan:

Program yang telah disubmit harus sama dengan spesifikasi format output. Terlalu banyak spasi/huruf besar akan menghasilkan wrong answer. Input dan output bersifat case-sensitive.

1.4 Input output example

Menjelaskan detail contoh input dan output yang harus dihasilkan program.

1.5 Time limit

Menjelaskan batasan maksimum waktu yang diperbolehkan selama program dijalankan.

1.6 Penyerahan Jawaban

Problem akan dianggap benar bila jawaban tersebut diterima oleh judges dan judges menjawab benar. Untuk menentukan jawaban, judges akan meng-compile dan menguji problem dengan menggunakan tes input rahasia (hanya diketahui oleh judges) yang telah disiapkan oleh judges. Setiap jawaban yang dikirim akan mendapat respon dari judges berupa :

- **Yes - Accepted**

Program menghasilkan keluaran/output yang benar, tidak melebihi batasan waktu yang ditentukan, dan tanpa error. Problem ini diterima oleh judges dan problem dikatakan sudah berhasil diselesaikan.

- **Yes - Solved**

Program menghasilkan keluaran/output yang benar, namun sudah pernah mendapatkan jawaban benar sebelumnya atau sudah accepted.

- **No - Wrong Answer**

Program tidak menghasilkan output yang benar.

- **No - Runtime Error**

Program crash ketika memproses input yang diberikan judges.

- **No - Time Limit Exceeded**

Program melebihi batas waktu yang ditentukan ketika dijalankan.

- **No - Compile Error**

Source code yang diberikan tidak bisa di - compile.

Catatan:

Akan ada kemungkinan respon lain dari judges selain dari 5 respon di atas.

Bergantung pada situasi yang ditentukan oleh judges.

1.7 Scoreboard

Papan scoreboard adalah papan penilaian yang menunjukkan ranking tim dan jawaban dari pertanyaan programming (Programming question). Jika ada yang mau ditanyakan perihal problem tersebut, setiap tim hanya diperbolehkan untuk bertanya melalui klarifikasi seperti yang dijelaskan pada poin 1.7.

1.8 Peringkat

Peringkat tim didasarkan jumlah problems yang diselesaikan. Jika terdapat tim yang menyelesaikan jumlah problems yang sama, maka tim akan diperingkat berdasarkan waktu tercepat (waktu yang paling sedikit adalah waktu yang paling bagus). Total waktu adalah jumlah dari waktu yang dibutuhkan untuk setiap problem (dalam menit). Waktu yang dibutuhkan untuk menyelesaikan suatu problem adalah waktu dimulainya kontes hingga jawaban

diterima oleh penguji ditambah penalti waktu yang telah ditentukan untuk setiap jawaban yang ditolak oleh penguji (respon selain Yes – Accepted).

Format pengumpulan jawaban dan peraturan umum jawaban programming

1. Java (.java)

- Jangan menggunakan Package
Karena jika menggunakan package, program bisa di-compile tapi tidak bisa dirun, sehingga akan mendapatkan NO - Runtime Error.
- Pastikan nama class adalah "class Main", bukan "Public class Main".
- Pastikan nama classnya "class Main". Tidak boleh nama lain.
- Gunakan "\r\n" dalam pembuatan baris baru / new line Contoh:
System.out.print("\r\n") atau System.out.println("\r"), **bukan**
System.out.println("") atau System.out.print("\n").
- Java tidak dapat membaca tipe data long dengan Scanner nextInt().
Ingat, tipe data int hanya bisa menyimpan hingga $2^{31} - 1$. Jika input yang diberikan lebih besar dari itu dan dibaca dengan Scanner nextInt(), maka Java akan mengeluarkan exception, Runtime Error.

2. C++ (.cpp)

- Jangan menggunakan void main()
Compiler yang juri gunakan mengharuskan fungsi main() memiliki return value berupa int, sehingga harus ditulis dengan int main().
- Jangan menggunakan getch() atau system("pause")
Jika maksud dari penggunaan perintah ini adalah menghentikan program pada akhir eksekusi, maka saat submit perintah ini harus dihapus terlebih dahulu. Karena getch() dapat menyebabkan Time Limit Exceed (program akan menunggu user memberi input melalui keyboard, yang tidak akan dilakukan juri) dan system("pause") akan menyebabkan Wrong Answer (ada output tambahan "Press any key to continue . . ." yang tidak diharapkan).
- Gunakan "\r\n" dalam pembuatan baris baru / new line Contoh:
printf ("%ld\r\n", res);
- Jangan memanggil fungsi tanpa meng-include header yang diperlukan

Jika anda menggunakan IDE Dev C++, maka anda harus berhati-hati karena Dev C++ bisa meng-include secara default beberapa header yang diperlukan oleh program anda, tapi tidak demikian jika anda meng-compile dari command line (yang dilakukan juri). Sehingga submission yang seperti ini akan mendapatkan NO – Compile Error.

- Jangan menggunakan `strrev`, `fflush(stdin)`
Karena juri mengcompile dari command line, maka jika menggunakan perintah ini akan mendapatkan NO-Compile Error.
- C/C++, konstanta yang terlalu besar
Di C/C++ jika kita ingin menuliskan suatu konstanta yang melebihi batas `int`, kita harus menambahkan “LL” di akhir angka tersebut. Contoh: 1000000000000000000LL. Jika LL tidak ada, maka konstanta itu akan dianggap integer dan nilainya berubah (itu sebabnya compiler akan memberikan warning).

3. Pascal (.pas)

- Dilarang menggunakan `clrscr`;
- `int main` harus return 0
- Jangan menggunakan `readln` / `readkey` (atau sejenisnya) di akhir program. Penggunaan `readln` dalam membaca inputan diperbolehkan, asalkan tidak menggunakan `readln` untuk menghentikan program untuk mengeluarkan output.
- Gunakan “`#13#10`” dalam pembuatan baris baru / new line. Contoh: `write(#13#10)` atau `writeln(#13)`, penggunaan dengan kalimat `write('Hello world'+#13#10)` atau `writeln('Hello world'+#13)`, bukan `writeln()` maupun `writeln('Hello world')`.

Jika maksud dari penggunaan perintah ini adalah menghentikan program pada akhir eksekusi, maka saat submit perintah ini harus dihapus terlebih dahulu. Karena akan menghasilkan Time Limit Exceed (program akan menunggu user memberi input melalui keyboard, yang tidak akan dilakukan juri).

Contoh Penulisan :

```
1 begin
2     write('6th NPLC');
3 end.
```

a. Penilaian soal Programming

1. Setiap jawaban soal pemrograman wajib dikirimkan dalam bentuk source code menggunakan salah satu bahasa yang diperbolehkan.
2. Source code yang dikirimkan berukuran maksimal **100 KB**.
3. Setiap program jawaban menerima masukan dari standard input (stdin) dan memberikan keluaran melalui standard output (stdout).
4. Standard error (stderr) akan diabaikan oleh sistem, sehingga dapat digunakan sebagai sarana pengujian sementara. Tetapi ingat bahwa penggunaan stderr akan memiliki efek pada performa runtime anda.
5. Program jawaban harus mengembalikan kode keluaran (exit code) 0. Jika exit code tidak bernilai 0, maka program akan dianggap sebagai **Run Time Error**.
6. Program jawaban akan diuji terhadap beberapa kasus uji. Setiap kasus uji memiliki batasan waktu eksekusi dan penggunaan memori yang berbeda-beda (batasan dicantumkan pada soal).
7. Pengumpulan jawaban terhadap satu soal dinyatakan **Accepted** apabila jawaban yang dikumpulkan berhasil menyelesaikan **semua** kasus uji pada soal tersebut. Tidak ada nilai parsial.
8. Tim dianggap menyelesaikan sebuah soal apabila pengumpulan terakhir jawaban untuk soal tersebut dinyatakan **Accepted**.
9. Setelah tim menyelesaikan sebuah soal, pengumpulan jawaban untuk soal yang sama akan diabaikan.
10. Tim akan dikenai penalti sebesar jumlah waktu yang berlalu sejak kompetisi dimulai sampai pengumpulan jawaban yang dinyatakan **Accepted** untuk setiap soal yang berhasil diselesaikan.
11. Tim juga akan dikenai penalti waktu tambahan sebesar **10 menit** untuk setiap pengumpulan jawaban yang tidak dinyatakan **Accepted** pada setiap soal yang berhasil diselesaikan pada akhirnya.
12. Kedua penalti waktu pada no. 10 dan 11 hanya akan dihitung ketika soal yang bersangkutan berhasil diselesaikan. Berapapun penalti waktu yang dikenakan untuk satu soal tidak akan dihitung jika soal tersebut tidak berhasil diselesaikan.
13. Tim akan diurutkan berdasarkan total soal yang diselesaikan, kemudian total penalti waktu yang paling sedikit, dan akhirnya berdasarkan waktu pengumpulan jawaban **Accepted** terakhir yang lebih awal.
14. Program jawaban tim dilarang untuk:
 - a. Mengakses jaringan
 - b. Forking
 - c. Membuka dan membuat file
 - d. Menyerang sistem keamanan server
 - e. Mengeksekusi program lain

- f. Mengganti hak akses (izin) file system
 - g. Membaca informasi file system
 - h. Membuat system call
 - i. dan hal-hal lain yang mengindikasikan hacking server kontes atau menggunakan program lain untuk menyelesaikan soal.
15. Tim yang melanggar aturan no. 14 secara sengaja akan didiskualifikasi.
16. Keputusan juri adalah mutlak dan tidak dapat diganggu gugat.

Online Competition

1. Benar: @ pertanyaan 10.
2. Salah: @ pertanyaan 0.
3. Setiap kali salah penalti 20 menit. (Tidak mempengaruhi skor).

b. Hal yang harus diperhatikan

Selama babak penyisihan berlangsung, dimohon tim tidak mengajukan pertanyaan-pertanyaan (klarifikasi) yang "aneh" dari peserta:

1. "Kenapa runtime error?" atau "salah dimana?" atau "input yang dijudge apa?" Respon yang hanya bisa diberikan juri untuk setiap submission hanya YES atau NO beserta statusnya. Juri tidak akan memberitahukan letak kesalahan dari program peserta. Penjurian dilakukan secara black box, submission dari peserta di compile, dan kemudian dijalankan dengan input yang sudah disiapkan. Juri sama sekali tidak membaca source code yang dikirimkan dalam menentukan respon yang harus diberikan. Setelah respon diberikan memang kadang-kadang juri membuka source code untuk melihat apa kesalahan code tersebut, namun apa yang juri temukan tidak akan diberitahukan kepada tim yang bersangkutan.
2. "Kenapa di scoreboard sudah ada submission di menit ke-x yang sudah digrade tapi submission kami di menit ke-y ($y < x$) belum digrade?"
Well, yang ini bukan pertanyaan aneh, bisa dimaklumi. Yang melakukan penjurian (tentu) lebih dari satu orang. Submission peserta akan diperiksa berdasarkan siapa yang submit duluan, tapi lama proses pemeriksaan bisa berbeda-beda tergantung dari juri dan code yang diperiksa. Jadi ada kemungkinan submission yang lebih telat mendapat respon lebih cepat karena submission yang lebih cepat tadi masih diperiksa, namun perbedaan waktunya seharusnya tidak signifikan.
3. "Kenapa hasilnya wrong answer, padahal output sudah benar?"
Kesalahan umum yang judge temui pada saat trial adalah whitespace (spasi, enter baris) berlebih yang dikeluarkan oleh output program peserta, serta ketidaksesuaian program peserta dengan format yang telah diberikan panitia melalui Guidebook ini (lihat **Bagian 1.8: Format pengumpulan**

jawaban dan peraturan umum jawaban). Ingatlah bahwa output dengan testcase harus sesuai agar dapat dibaca benar oleh sistem, diasumsikan bahwa logika pemrograman dan hal-hal yang lain sudah benar.

c. Tips

1. Jangan lupa untuk menuliskan "import" atas semua class yang anda gunakan di file Java anda.
2. Hapus tulisan "package" di file Java anda.
3. class untuk menulis PSVM (java) tidak boleh public (caranya dengan menghapus public).
4. Nama class untuk menulis PSVM di Java harus Main.
5. Input dan output bersifat case-sensitive. Jadi kalau ada perbedaan satu karakter saja akan mengakibatkan jawaban anda tidak diterima. Penambahan spasi, enter, atau karakter lain di luar yang diminta di problem juga mengakibatkan jawaban anda tidak diterima.
6. Untuk melihat problem mana yang lebih mudah, anda bisa melihat problem mana yang lebih dahulu diselesaikan oleh tim lain melalui scoreboard agar anda tidak menghabiskan waktu yang berlebihan di sebuah problem.
7. Saat submit jawaban jangan lupa untuk menyamakan bahasa yang dipakai dengan file yang dikirim.
8. Agar lebih aman, gunakanlah template yang telah disediakan (lihat 1.8), anda tinggal mengubahnya sesuai kebutuhan.