



# Software Development Selection

## Task

**Deadline: 12 Februari 2026, 23.59 WIB**

---

### Aturan Pengerjaan

#### Pengerjaan mandiri

Peserta wajib mengerjakan tugas secara mandiri dengan pemahaman penuh terhadap kode yang dibuat. Peserta hanya perlu mengerjakan minimal **5 dari 10 soal**, apabila peserta mengerjakan 10 soal maka akan menjadi nilai plus. Penggunaan **Chat GPT diperbolehkan**, namun peserta harus memahami logika dari solusi yang dihasilkan.

#### Pengerjaan kode

- Bahasa Pemrograman : Bebas memilih bahasa yang disukai, namun penggunaan JavaScript akan mendapatkan nilai tambahan.
- Buat sebuah Repository GitHub dan pastikan repository github dibuka secara **publik**
- Buat repository dengan format nama berikut: **Hands On EISD\_SOFTDEV\_Nama Lengkap\_NIM**
- Setiap soal harus memiliki file kode terpisah dengan format: `soalX.extension` (*X adalah nomor soal, extension sesuai bahasa pemrograman*) Contoh: `soal1.js`, `soal2.py`, `soal3.java`
- Pengerjaan task noncoding dijadikan file README

#### Fokus Penilaian

- Pemahaman Masalah & Ketepatan Solusi: Memahami kebutuhan soal dan menerjemahkannya ke dalam solusi yang sesuai aturan
- Logika Pemrograman & Alur Berpikir: Seberapa runtut dan logis alur berpikir yang kamu gunakan dalam menyelesaikan masalah?



- Pengolahan Data & Penanganan Edge Case: Bagaimana kemampuan kamu dalam memvalidasi input agar sesuai dengan ketentuan?



## Noncoding

### 1. Soal noncoding (WAJIB)

EISD Lab baru saja merilis website pendaftaran Study Group untuk batch terbaru. Karena ingin cepat dipakai, versi awalnya dibuat sederhana: peserta tinggal isi Nama, Email, Pilihan Kelas, lalu klik Daftar.

Di hari pertama pendaftaran, panitia menemukan beberapa kejadian :

- Ada peserta yang panik karena halaman sempat loading lama, lalu menekan tombol Daftar berkali-kali yang mengakibatkan data masuk lebih dari sekali.
- Ada yang mengisi asal: nama ".", email "aaa", atau bahkan mengosongkan form mengakibatkan data berantakan.
- Di akhir hari, beberapa kelas yang kuotanya 25 orang ternyata terisi 30+ karena sistem tetap menerima pendaftaran.
- Saat panitia mau rekap, muncul pertanyaan: "Kalau data dobel dan data tidak valid begini, mana yang dianggap resmi?"

Anggap kamu menjadi orang yang diminta membantu memikirkan perbaikan versi berikutnya, tapi waktu sangat terbatas dan panitia minta kamu menentukan prioritas. Jika kamu hanya boleh memperbaiki satu hal pada *website* tersebut, apa yang akan kamu perbaiki terlebih dahulu dan mengapa?



## Easy

### 2. Array Reading Dasar

Membaca data *array* dan menentukan kondisi sederhana

Contoh: hitung, pilih, bandingkan

Panitia EISD Lab menyimpan data pendaftaran *study group* sebagai berikut:

Nama Study Group	Kuota	Jumlah Terdaftar
Software Development	20	18
UI/UX Design	15	15
Technopreneur	25	10
Intelligence System	10	10

Buat sebuah program/*function* yang membaca data di atas dan:

- Menampilkan nama *study group* yang masih tersedia (jumlah terdaftar < kuota)
- Menampilkan nama *study group* yang sudah penuh (jumlah terdaftar  $\geq$  kuota)

### 3. Conditional Logic

Menentukan output berdasarkan kondisi tertentu

Dalam sistem pendaftaran Study Group EISD Lab, status pendaftaran ditentukan berdasarkan kuota maksimal dan jumlah peserta terdaftar. Panitia ingin sistem memberikan informasi kondisi pendaftaran dengan aturan berikut:

1. Jika jumlah terdaftar kurang dari (kuota - 3) = Pendaftaran diterima
2. Jika jumlah terdaftar berada di 3 slot terakhir sebelum penuh = Hampir penuh
3. Jika jumlah terdaftar sama dengan kuota = Pendaftaran ditutup
4. Jika jumlah terdaftar melebihi kuota = Pendaftaran ditolak



Buatlah sebuah program dengan ketentuan sebagai berikut:

- Menerima input kuota maksimal
- Menerima input jumlah peserta terdaftar
- Menampilkan status pendaftaran sesuai aturan di atas

#### 4. Validasi Input Sederhana

Cek data valid / tidak valid, string kosong, angka minimum, format dasar

Pada sistem pendaftaran Lab, peserta diminta mengisi beberapa data dasar sebelum melakukan pendaftaran. Data yang diinput seharusnya lengkap dan masuk akal agar tidak menyulitkan panitia saat melakukan rekap.

Data yang dimasukkan peserta meliputi:

- Nama peserta
- Email
- Pilihan Divisi
- Jumlah peserta terdaftar saat ini

Sayangnya, masih ditemukan data seperti nama yang dikosongkan, *email* tanpa format yang benar, atau jumlah peserta bernilai nol.

Buat sebuah program atau *function* yang memeriksa apakah data yang diinput sudah valid atau belum dengan ketentuan berikut:

- Nama tidak boleh kosong
- Email tidak boleh kosong dan harus mengandung karakter @
- Pilihan *study group* tidak boleh kosong
- Jumlah peserta terdaftar harus berupa angka dan minimal bernilai 1

Program menampilkan:

- "Data valid" jika seluruh data memenuhi ketentuan
- "Data tidak valid" jika terdapat satu atau lebih data yang tidak sesuai



## 5. Counting & Aggregation

Menghitung total, rata-rata, atau jumlah tertentu dari data  
loop ringan, accumulator

Panitia EISD Lab ingin melihat gambaran umum jumlah peserta yang sudah mendaftar di setiap Study Group. Data jumlah peserta disimpan dalam sebuah array yang berisi informasi berikut:

- Web Development Dasar: 18 peserta
- Logic & Algorithm: 15 peserta
- Frontend Fundamental: 10 peserta
- Backend Introduction: 10 peserta

Agar data ini lebih mudah dianalisis, panitia membutuhkan perhitungan sederhana dari data tersebut.

Buat sebuah program yang:

1. Menghitung total seluruh peserta yang sudah terdaftar
2. Menghitung rata-rata jumlah peserta per Study Group
3. Menampilkan hasil perhitungan tersebut ke layar

Gunakan perulangan sederhana dan variabel penampung (accumulator) dalam proses perhitungannya.

## 6. Duplicate Detection

Menentukan apakah ada data yang muncul lebih dari sekali  
(*array, object, set*)

Pada proses pendaftaran Study Group EISD Lab, sistem menyimpan nama peserta yang berhasil melakukan pendaftaran. Karena tidak ada pembatasan klik tombol Daftar, beberapa peserta tercatat lebih dari satu kali di dalam sistem.



Daftar nama peserta yang tersimpan adalah sebagai berikut :

- Andi@gmail.com
- Budi@gmail.com
- Sari@gmail.com
- Andi@gmail.com
- Rina@gmail.com
- Budi@gmail.com

Buat sebuah program yang:

- Memeriksa apakah terdapat email yang muncul lebih dari satu kali
- Menampilkan daftar email yang terdeteksi sebagai data duplikat

Jika tidak ditemukan data yang sama, program cukup menampilkan pesan bahwa tidak ada data duplikat.



## Medium

### 7. Data Mapping

Mengubah bentuk data

array → object, object → array, rename field

Panitia EISD Lab menyimpan data pendaftar study group dalam bentuk data berikut:

Nama	Status	Nilai Seleksi
Alya	active	85
Bima	inactive	90
Citra	active	70
Doni	active	95
Eka	inactive	60

Buat sebuah program yang melakukan langkah berikut:

- Memfilter pendaftar dengan ketentuan:
  1. Status = active
  2. Nilai seleksi minimal 80
- Dari hasil filter tersebut, ubah bentuk data menjadi data baru dengan format:  
nama: <nama peserta>  
nilai: <nilai seleksi>

### 8. Filtering with Multiple Conditions

Memfilter data dengan lebih dari satu aturan

contoh: status + kategori + nilai minimum

Sebuah platform kursus online sedang melakukan evaluasi peserta untuk menentukan siapa saja yang dapat mengikuti kelas lanjutan. Setiap





peserta memiliki data berupa nama, status akun, kategori kelas, dan nilai evaluasi akhir.

Data peserta yang tersimpan adalah sebagai berikut:

Nama	Status Akun	Kategori	Nilai Evaluasi
Alya	active	Web	85
Bima	active	Data	75
Citra	inactive	Web	901
Doni	active	Web	95
Eka	active	Data	80

Pihak pengelola ingin menampilkan hanya peserta yang benar-benar memenuhi syarat untuk mengikuti kelas lanjutan.

Buat sebuah program yang melakukan penyaringan data peserta dengan ketentuan berikut:

- Status akun active
- Kategori kelas adalah Web
- Nilai evaluasi minimal 80

Setelah proses penyaringan, tampilkan nama peserta dan nilai evaluasinya sebagai hasil akhir.



## Hard

### 9. *Status Transition*

Data memiliki status yang bisa berubah berdasarkan aksi

Contoh: open → closed, pending → approved

Pada hari pertama pendaftaran Study Group EISD Lab, panitia menerapkan sistem *first come first served* dengan kuota terbatas.

Diketahui:

1. Kuota maksimal peserta: 3 orang
2. Urutan pendaftar masuk ke sistem adalah sebagai berikut:  
[Dea, Ijat, Arra, Budi, Andi]

Buat sebuah program yang mensimulasikan proses pendaftaran dengan aturan:

- Peserta diproses sesuai urutan
- Selama kuota masih tersedia, peserta diterima
- Jika kuota sudah penuh:
  - peserta berikutnya ditolak
  - proses pendaftaran berhenti

Output yang Diharapkan:

- Daftar peserta yang diterima
- Daftar peserta yang ditolak

### 10. *Simulation Logic*

Mensimulasikan proses bertahap

input → proses berulang → kondisi berhenti

Sebuah aplikasi peminjaman buku perpustakaan melayani peminjaman secara bergiliran. Setiap anggota hanya boleh meminjam satu buku, dan jumlah buku yang tersedia terbatas.

Pada awal proses, tersedia 4 eksemplar buku. Permintaan peminjaman masuk ke sistem dengan urutan sebagai berikut:

- Alya
- Bima



- Citra
- Doni
- Eka
- Farah

Sistem akan memproses permintaan satu per satu sesuai urutan masuk. Setiap permintaan yang berhasil akan mengurangi jumlah buku yang tersedia.

Jika buku sudah habis, sistem tidak akan memproses permintaan selanjutnya.

Buat sebuah program yang mensimulasikan proses peminjaman buku dengan ketentuan:

- Permintaan diproses satu per satu sesuai urutan
- Selama buku masih tersedia, peminjaman berhasil
- Ketika buku sudah habis:
  - Permintaan berikutnya dinyatakan gagal
  - Proses simulasi dihentikan

**"Programming is not the pursuit of perfect code, but the craft of turning complex problems into well reasoned solutions."**

Selamat Mengerjakan! Jawabanmu mencerminkan caramu melihat masalah.