**TUGAS JOBSHEET 1**

**Ditulis untuk memenuhi tugas matakuliah Pemrograman Jaringan**

**[202513430087]**

**Dosen Pengampu:**

**Randi Proska Sandra, S.Pd, M.Sc.**

****

**Disusun Oleh:**

**Florensius Panca Gati**

**(23343006)**

**PROGRAM STUDI INFORMATIKA**

**DEPARTEMEN TEKNIK ELEKTRONIKA**

**FAKULTAS TEKNIK**

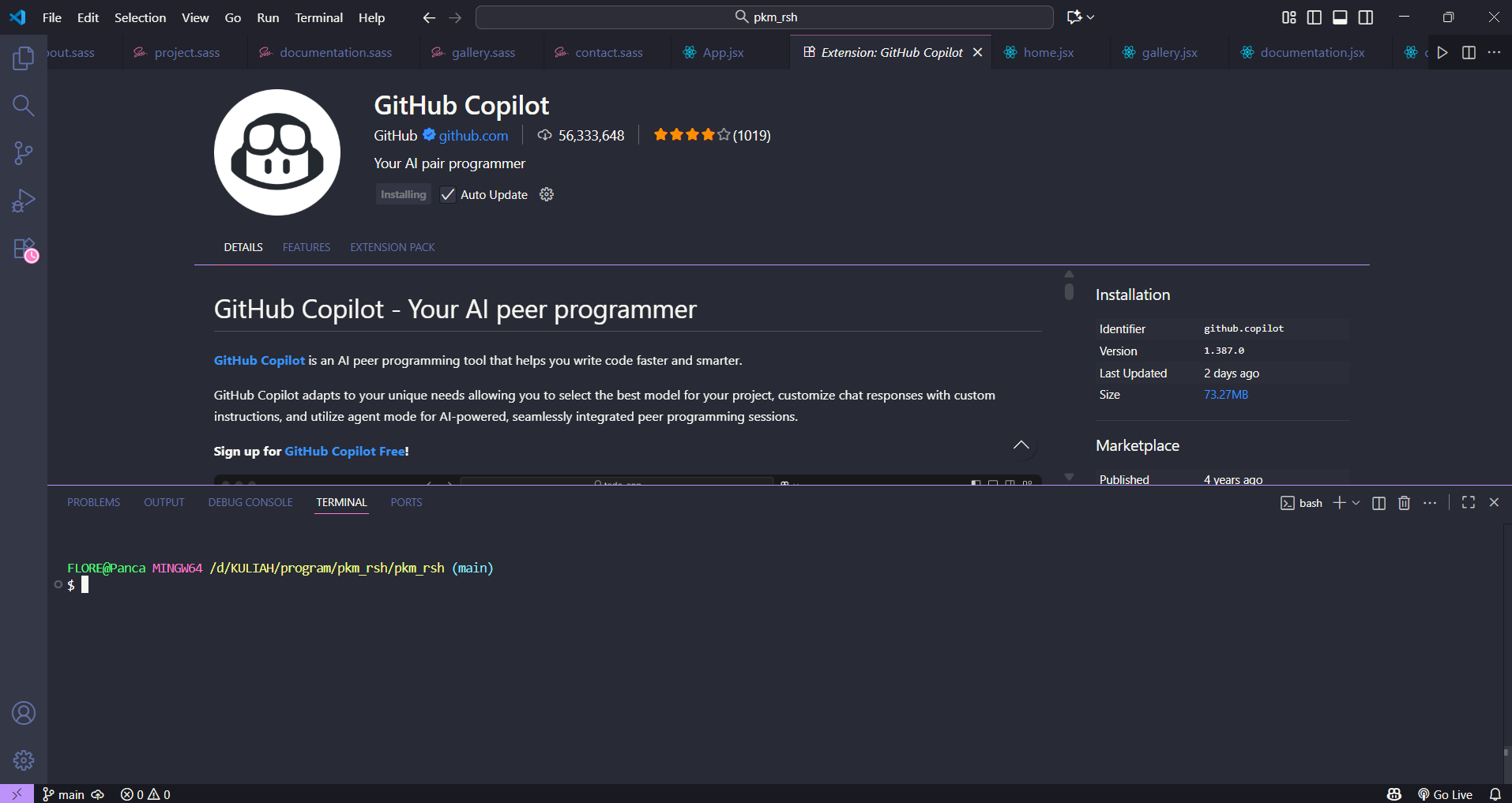
**UNIVERSITAS NEGERI PADANG**

**2025**

**LATIHAN**

1. **Instalasi VSCode**
2. Silakan download visual studio code melalui link berikut ini <https://code.visualstudio.com/>
3. Setelah diunduh, jalankan installer (VSCodeUserSetup-{versi}.exe). Ini hanya akan memakan waktu sebentar.
4. Secara default, VS Code diinstal pada direktori C:\Users\{Username}\AppData\Local\Programs\Microsoft VS Code

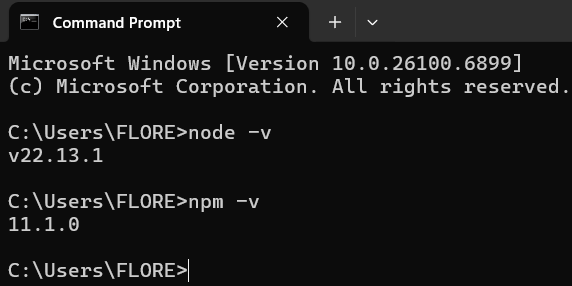
Penulis sudah melakukan seluruh proses ini sudah dilakukan pada semester-semester sebelumya.



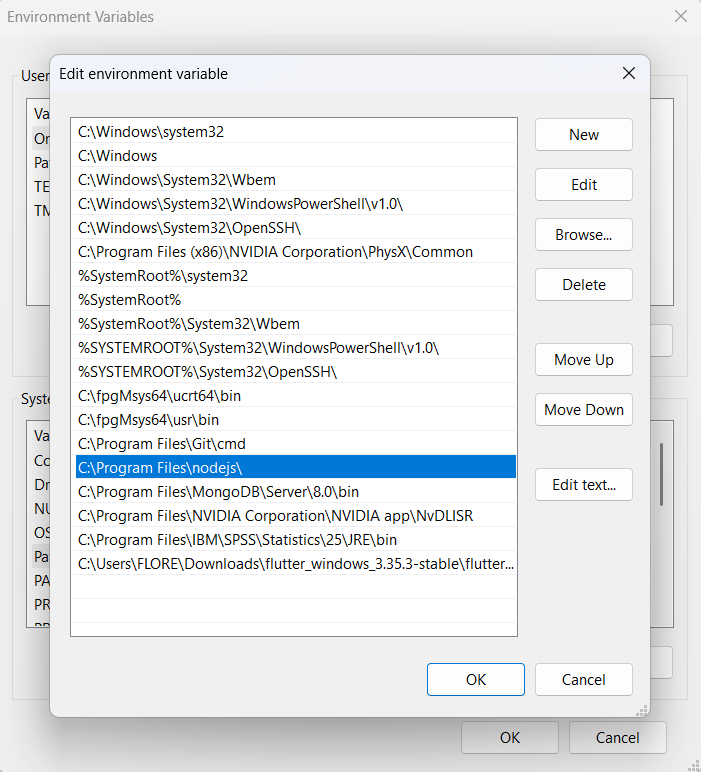
1. **Instalasi NodeJS**
2. Kunjungi situs resmi Node.js di https://nodejs.org/ dan unduh versi Node.js yang sesuai dengan sistem operasi Anda (Windows, macOS, atau Linux).



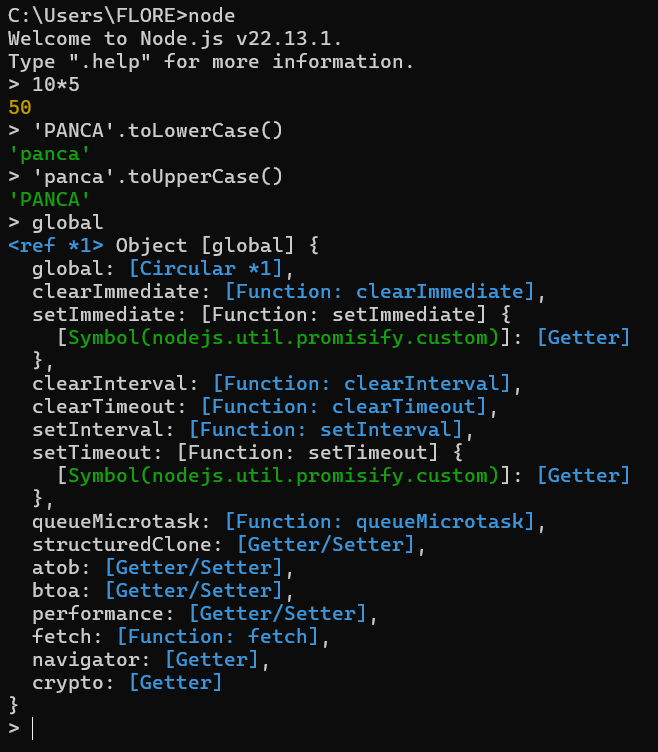
1. Setelah unduhan selesai, jalankan berkas instalasi yang telah diunduh. Pada Windows, biasanya akan berupa berkas dengan ekstensi .msi, dan pada macOS berkas dengan ekstensi .pkg. Di Linux, mungkin perlu menggunakan manajer paket atau unduh dari sumber yang sesuai.
2. Ikuti langkah-langkah yang ada dalam proses instalasi. Hal ini dapat mencakup pemilihan direktori instalasi, persetujuan terhadap syarat dan ketentuan, dan konfigurasi lainnya.
3. Setelah mengkonfirmasi pilihan-pilihan instalasi, tunggulah sampai proses instalasi selesai.
4. Setelah instalasi selesai, buka terminal atau command prompt (tergantung pada sistem operasi) dan ketikkan perintah node -v. Hal ini akan mengeluarkan versi Node.js yang telah diinstal. Selain itu, juga dapat menguji dengan menjalankan perintah npm -v untuk memastikan bahwa npm (Node Package Manager) juga diinstal.



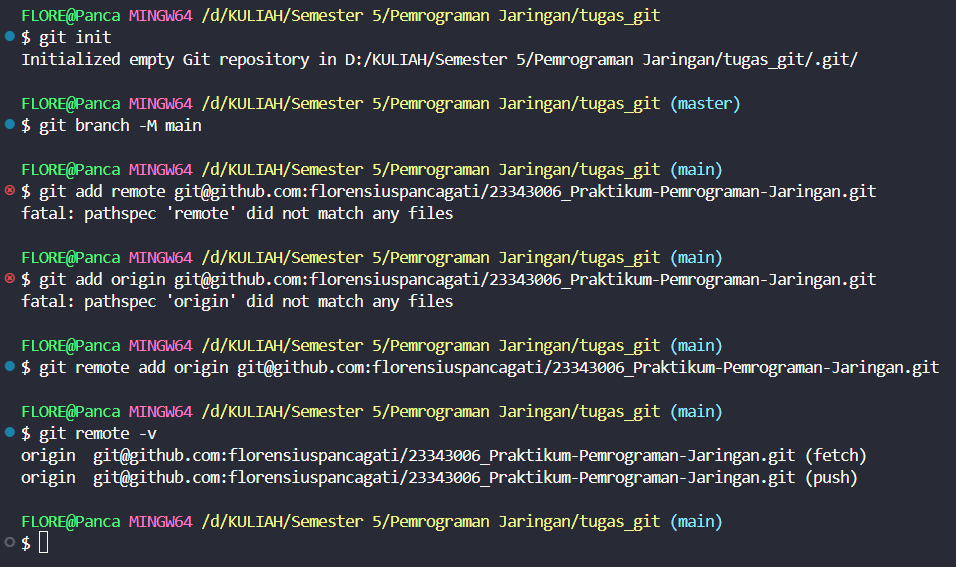
1. Jika node.js belum terdeteksi silakan set PATH node.js pada environment variables. Anda bisa melakukannya dengan perintah berikut pada command prompt. (pastkan lokasi direktori node.js sesuai dengan folder instalasi) SET PATH=C:\Program Files\Nodejs;%PATH%
2. Atau membuka setting environment variables dengan cara membuka Control Panel -> System and Security -> System -> Advanced System Settings -> Environment Variables
3. Edit variabel path pada user variables dan system variables dan tambahkan direktori instalasi node.js C:\Program Files\nodejs



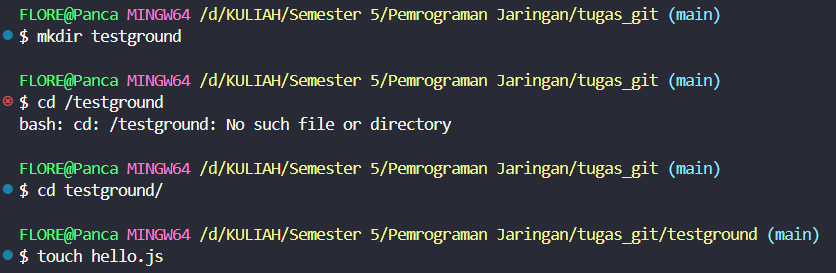
1. **Tes Perintah NodeJS pada CMD**
2. Silakan masukan perintah berikut pada command prompt untuk uji coba penggunaan node.js

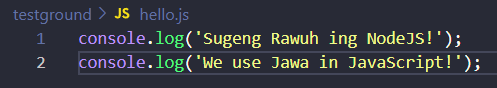


1. Silakan gunakan perintah process.exit() untuk keluar
2. Program Hello-World
3. Silakan buat sebuah folder dengan nama PBJ1 atau sesuai keinginan anda dan simpan pada direktori yang anda inginkan. Jangan memberikan spasi pada nama folder anda.
4. Bukalah visual studio code dan kemudian bukalah folder tersebut pada visual studio code



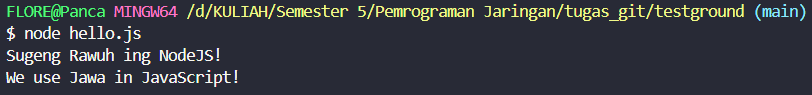
1. Setelah terbuka buatlah folder baru pada folder tersebut melalui visual studio code dengan nama testground dan dalam folder tersebut
2. Buatlah file baru dengan nama hello.js dalam folder yang anda buat pada Langkah 3 dan cobalah ketikan kode berikut ini



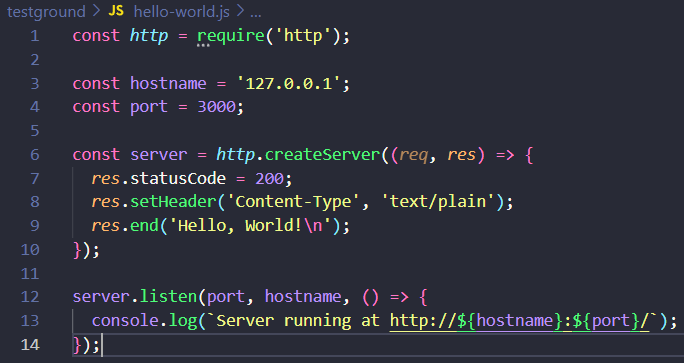


1. Jalankan file tersebut dan lihat bagaimana hasilnya. Anda juga dapat menjalankan program tersebut melalui terminal pada visual studio code dengan memasukan perintah berikut PS

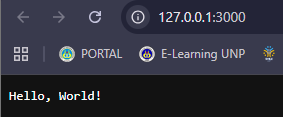
C:\Users\{NamaPengguna}\Desktop\{namaFolder}\testground> node hello.js



1. Buatlah sebuah file baru lagi dengan nama hello-world.js pada folder yang sama dan cobalah ketikan kode berikut ini



1. Jalankan program tersebut dan akan muncul informasi Server running at <http://127.0.0.1:3000/>
2. Bukalah url tersebut melalui browser anda dan akan muncul tampilan HelloWorld



**TUGAS**

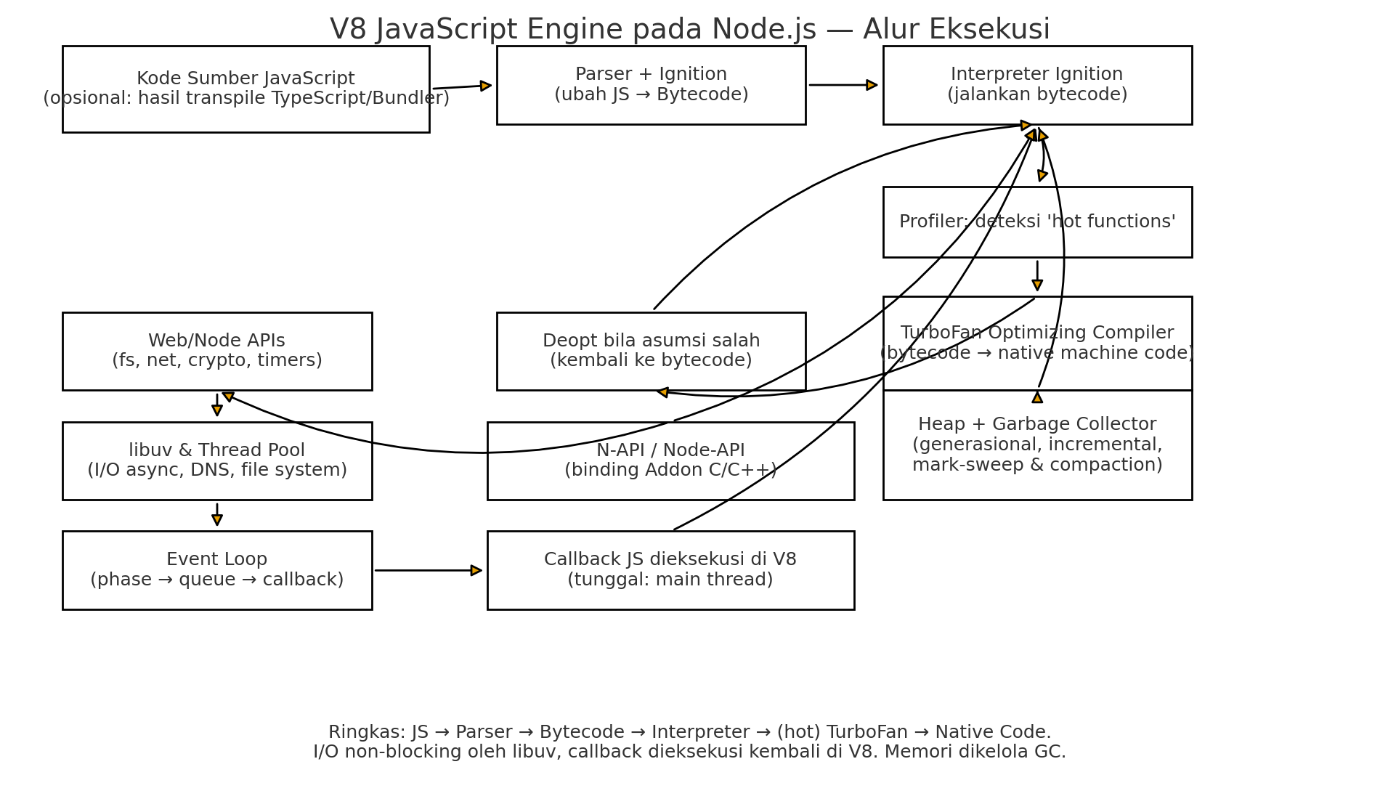
1. Jelaskan tentang JavaScript Engine, V8 dan bagaimana perbedaan antara Node.js dan JavaScript Engine pada browser Google Chrome?
2. **V8 JavaScript Engine pada NodeJS**

NodeJS adalah runtime environment untuk running Bahasa pemrograman JavaScript pada server (diluar browser). NodeJS berjalan pada V8 JavaScript runtime engine. V8 enggine mirip seperti mingw pada C++, ia merubah kode JavaScript menjadi kode mesin yang lebih cepat. V8 ini ditulis menggunakan kode C++.

Keunggulan NodeJS:

* Efisiensi skalabilitas
* Responsivitas tinggi
* Dukungan multiplatform
* Integritas kode client-side dan server-side
* Ekosistem NPM yang kaya

Berikut Adalah diagram flow V8 JavaScript Engine pada NodeJS:



Node.js menggunakan V8 sebagai mesin eksekusi JavaScript di sisi server. Prosesnya dimulai dari parser yang mengubah kode JS menjadi bytecode melalui Ignition, kemudian dijalankan interpreter. Fungsi yang sering dipanggil dioptimalkan oleh TurboFan menjadi native machine code untuk meningkatkan performa. Node.js tidak memiliki DOM atau rendering, tetapi menyediakan *Node APIs* seperti fs, http, crypto, dan timers. I/O asynchronous ditangani oleh **libuv** melalui thread pool, sementara hasil operasi asinkron dikirim kembali ke event loop untuk menjalankan callback JavaScript di main thread. Node.js juga mendukung **N-API** untuk integrasi dengan library native C/C++.

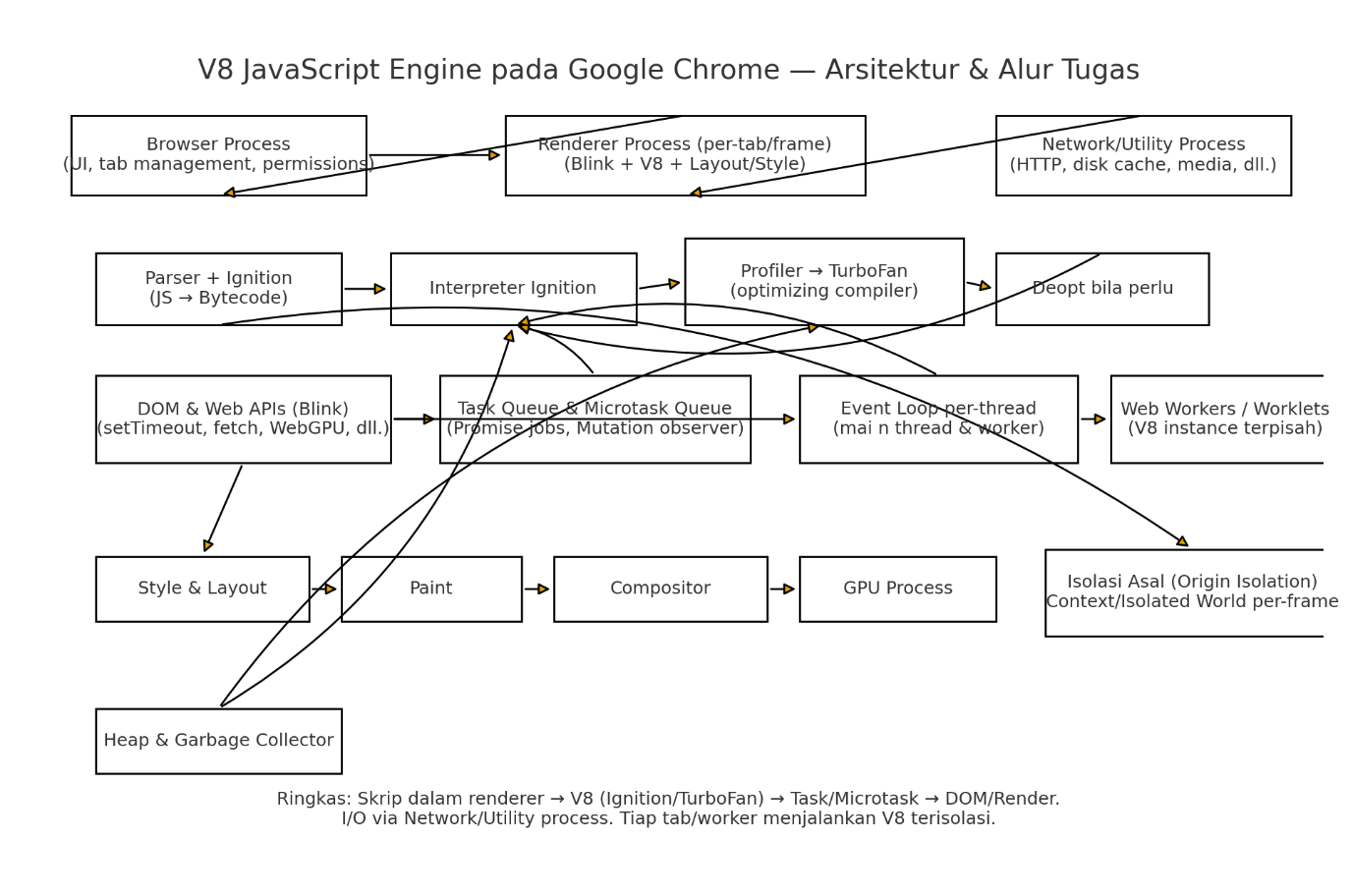
1. **V8 JavaScript Engine pada browser Google Chrome**

V8 pada chrome sama dengan pada NodeJS. Dengan adanya V8 engine pada chrome, memungkinkan user untuk running kode JavaScript pada console chrome.

Keunggulan V8 JavaScript Engine pada browser Google Chrome:

* Peforma tinggi
* Optimisasi dinamis
* Efisiensi memori
* Integrasi dengan browser
* Keamanan tinggi
* Dukungan modern JavaScript
* Multi-threading dengan worker
* Kompatibilitas dengan webassembly

Berikut Adalah diagram flow V8 JavaScript Engine pada Chrome:



Di Chrome, V8 menjadi bagian dari **renderer process** yang bekerja bersama engine Blink. Sama seperti Node.js, JavaScript dijalankan melalui Ignition (interpreter) dan TurboFan (compiler), dengan deoptimization bila asumsi optimisasi salah. Namun, integrasinya jauh lebih kompleks: V8 terhubung langsung dengan **DOM, Web APIs, dan rendering pipeline** (style, layout, paint, compositor, GPU). Setiap tab/frame memiliki instance V8 sendiri (isolation), sehingga keamanan lebih tinggi. Chrome juga mendukung **Web Workers dan Worklets**, yang menjalankan V8 terpisah di thread berbeda, serta **WebAssembly** untuk kode non-JS. Event loop di Chrome menangani task queue & microtask queue untuk memastikan UI tetap responsif