

Panduan Lengkap gstack (v1.91.100.0)

Dokumentasi AI Virtual Engineering Team untuk Claude Code dan AI Coding Agents

Dibuat oleh Garry Tan (President dan CEO Y Combinator) | Repositori GitHub: <https://github.com/garrytan/gstack>

1. Apa itu gstack?

gstack adalah kumpulan perkakas (toolset) dan keahlian (skills) berbasis Markdown dan CLI yang dirancang khusus untuk mengubah Claude Code (serta AI coding agent lainnya seperti Codex, OpenClaw, dan Cursor) menjadi satu tim rekayasa perangkat lunak virtual yang lengkap.

Alih-alih memberikan instruksi kosong tanpa panduan, gstack menerapkan alur kerja pengembangan terstruktur:

Think -> Plan -> Build -> Review -> Test -> Ship -> Reflect

Target Pengguna dan Nilai Utama

1. Founders, Tech Leads, dan Solo Developers: Membangun produk dari konsep awal hingga rilis produksi dengan standar rekayasa perangkat lunak yang ketat.
 2. Pengguna Claude Code: Menghadirkan peran spesialis terarah seperti CEO, Product Designer, Engineering Manager, QA Lead, dan Security Officer.
 3. Standar Kualitas Tinggi: Mencegah penulisan kode tanpa uji coba, tampilan antarmuka generik, dan kegagalan pada lingkungan produksi.
-

2. Informasi Versi, SHA dan Pembaruan

- Versi Rilis Saat Ini: v1.91.100.0
 - Latest Commit Hash (SHA-1): 15ebd85cdde6c9a1b2fb707e8573aec18e8f7438
 - Ringkasan Pembaruan Terbaru (Changelog):
 - v1.91.100.0: Resolusi kalkulasi harga gbrain LongMemEval untuk call plan penuh dan perbaikan bootstrapper Capy.
 - Skill baru /eval-plan: pra-registrasi evaluasi berbayar tier-1 dengan lane-check, verifikasi bukti, dan coordinator contract.
 - Skill baru /test-audit: penemuan duplicate atau low-value test suites dan dead test-only code.
 - Unattended Runner /autoplan: runner mandiri untuk sesi autoplan dengan pluggable outside voice dan checklists review.
 - Perbaikan Ekosistem Windows: penanganan line endings (setup CRLF repair), pemulihan false positive hook Git Bash, dan dukungan temp path 8.3.
 - Dukungan AntigraVity CLI (agy): integrasi host agy dan sinkronisasi batasan credential boundary.
-

3. Panduan Instalasi Lengkap (Manual & AI)

A. Metode 1: Instalasi Menggunakan AI (Claude Code / AI Agent Otomatis)

Jika Anda menggunakan Claude Code atau AI Agent lain di terminal proyek, Anda cukup menyalin prompt berikut dan mengirimkannya ke AI. AI akan mengakses repositori GitHub <https://github.com/garrytan/gstack> dan melakukan seluruh konfigurasi secara mandiri:

Prompt untuk AI Agent:

"Tolong pasang gstack dari repositori <https://github.com/garrytan/gstack> dengan menjalankan perintah: `git clone --single-branch --depth 1 https://github.com/garrytan/gstack.git ~/claude/skills/gstack && cd ~/claude/skills/gstack && ./setup` kemudian tambahkan daftar routing skill gstack ke CLAUDE.md proyek ini."

Keuntungan metode ini: AI akan otomatis mengunduh berkas, melakukan kompilasi biner, dan mengonfigurasi routing skill proyek tanpa Anda harus mengetik perintah terminal satu per satu.

B. Metode 2: Instalasi Manual Terminal dari Awal hingga Selesai (Tanpa AI)

Ikuti tahapan berurutan berikut di terminal komputer Anda:

1. Tahap 1: Verifikasi Prasyarat Runtime

Pastikan Git dan Bun (atau Node.js di Windows) sudah terpasang:

```
git --version
```

```
bun --version || node --version
```

2. Tahap 2: Clone Repositori gstack

Unduh repositori resmi gstack ke folder direktori skills Claude:

```
git clone --single-branch --depth 1 https://github.com/garrytan/gstack.git ~/claude/skills/gstack
```

3. Tahap 3: Masuk Direktori & Jalankan Setup

Kompilasi binary internal gstack dengan menjalankan:

```
cd ~/.claude/skills/gstack && ./setup
```

4. Tahap 4: Hubungkan ke Proyek Anda (Routing Skill)

Masuk ke folder proyek kerja Anda dan daftarkan routing skill ke berkas CLAUDE.md:

```
~/.claude/skills/gstack/bin/gstack-team-init optional
```

5. Tahap 5: Uji Coba Skill Pertama

Buka Claude Code di proyek Anda dan jalankan perintah:

```
/office-hours
```

Jika muncul 6 pertanyaan kritis produk dari simulasi YC Office Hours, instalasi berhasil 100%.

6. Tahap 6 (Opsional): Setup AI Agent Non-Claude (Codex, Cursor, OpenClaw)

Jika Anda menggunakan coding agent lain, jalankan perintah:

```
cd ~/.claude/skills/gstack && ./setup --host codex
```

7. Tahap 7: Cara Memperbarui Versi gstack

Perbarui berkas gstack ke versi rilis terbaru kapan saja dengan perintah:

```
~/.claude/skills/gstack/bin/gstack-upgrade
```

4. Katalog Lengkap Skill gstack dan Kegunaannya

1. Tahap Ideasi dan Strategi Produk (Think and Scope)

- `/office-hours` : Simulasi YC Office Hours. Menguji ide produk dengan 6 pertanyaan kritis untuk menantang asumsi awal, mengeksplorasi alternatif solusi, dan menyusun dokumen spesifikasi awal sebelum penulisan kode dimulai.
- `/plan-ceo-review` : Tinjauan dari sudut pandang CEO dan Founder. Mengevaluasi arah masalah pada level strategi bisnis dengan 4 mode pilihan: Expansion, Selective Expansion, Hold Scope, dan Reduction.
- `/spec` : Penulisan dokumen spesifikasi teknis yang terstruktur, presisi, dan siap dieksekusi oleh developer maupun AI.

2. Desain dan Pengalaman Visual (Design and DX)

- `/design-consultation` : Mitra desain untuk merancang Design System dari awal (meliputi tipografi, sistem warna, tata letak, dan panduan anti-slop).
- `/design-shotgun` : Eksplorasi desain dengan membuat 4 sampai 6 variasi tampilan mockup sekaligus, menampilkan perbandingan visual di browser, dan merekam preferensi gaya desain pengguna.
- `/design-html` : Konversi mockup menjadi kode HTML dan CSS bersih yang siap pakai untuk framework React, Svelte, atau Vue.
- `/plan-design-review` dan `/design-review` : Audit kualitas visual desain dan perbaikan langsung pada inkonsistensi elemen antarmuka.
- `/plan-devex-review` dan `/devex-review` : Audit Developer Experience (DX), mengukur Time To Hello World (TTHW), dan menguji kemudahan dokumentasi teknis.

3. Arsitektur, Debugging dan Code Review (Plan and Build)

- `/plan-eng-review` : Peran Engineering Manager untuk mengunci arsitektur teknis dengan diagram alur data, state machine, penanganan error, serta matriks pengujian dan keamanan.
- `/autoplan` : Eksekusi pipeline review otomatis berlapis (CEO review, Design review, hingga Engineering review) dalam satu langkah terpadu.
- `/eval-plan` : Pra-registrasi evaluasi berbayar (bars, arms, decoys, stop rules), dry-run gratis, dan pilot tiga item berbayar sebelum persetujuan.
- `/deslop-shared-libs` : Audit arsitektur dan ekstraksi duplikasi kode bersama (shared-code extraction) untuk mencegah redundansi antar pustaka.
- `/investigate` : Pelacakan akar masalah (root cause debugging) secara sistematis dengan meneliti aliran data sebelum modifikasi kode dilakukan.
- `/review` : Tinjauan kode tingkat Staff Engineer untuk menemukan potensi bug yang lolos dari CI tetapi berisiko gagal di produksi.
- `/codex` : Evaluasi independen menggunakan OpenAI Codex CLI untuk memberikan sudut pandang kedua terhadap kode.

4. Keamanan dan Jaminan Kualitas (Test and Security)

- `/qa` : QA Lead otonom yang menguji aplikasi langsung di browser Chromium asli, menemukan bug interaktif, memperbaiki kode, dan menghasilkan skenario regresi otomatis.
- `/qa-only` : Pelaksanaan pengujian aplikasi di browser dengan keluaran laporan bug komprehensif tanpa memodifikasi kode sumber.
- `/test-audit` : Penilaian standar kualitas dan efisiensi test suite proyek sebelum melangkah ke proses staging atau rilis.
- `/cso` : Audit keamanan menyeluruh berbasis OWASP Top 10 dan model ancaman STRIDE dengan bukti skenario eksploitasi nyata.

5. Rilis, Deployment dan Pemantauan (Ship and Canary)

- `/ship` : Release Engineer untuk sinkronisasi branch utama, verifikasi seluruh test suite, merapikan riwayat commit, push, dan pembuatan Pull Request.
- `/land-and-deploy` : Otomasi penggabungan PR, pemantauan status CI/CD hingga sukses, dan verifikasi endpoint di server produksi.
- `/canary` : Pemantauan paska-rilis untuk mendeteksi error pada console log, penurunan performa, atau gangguan endpoint.
- `/benchmark` : Pengukuran performa web seperti Core Web Vitals, waktu respon, dan ukuran aset sebelum dan sesudah rilis.

6. Dokumentasi dan Retrospektif (Docs and Reflect)

- `/document-release` : Pembaruan dokumentasi repositori secara otomatis agar selalu sinkron dengan kode yang baru dirilis.
- `/document-generate` : Pembuatan dokumen teknis baru mengikuti kerangka kerja Diataxis (Tutorial, Panduan Praktis, Referensi, dan Penjelasan).
- `/retro` : Pembuatan laporan retrospektif berkala mencakup performa tim, tren stabilitas pengujian, dan riwayat pengiriman fitur.

7. Alat Diagram dan Utilitas Tambahan

- `/make-pdf` : Generator dokumen PDF berkualitas tinggi dari Markdown dengan rendering diagram vektor.
- `/diagram` : Konversi deskripsi teks menjadi diagram Mermaid, Excalidraw, dan format SVG.
- `/guard`, `/careful`, `/freeze` : Pembatasan eksekusi terhadap perintah berbahaya (seperti `rm -rf` atau `drop table`) dan penguncian cakupan folder kerja.
- `/learn` : Pengelolaan memori lintas sesi mengenai pola arsitektur, preferensi kode, dan catatan penting proyek.

5. Ringkasan File dan Akses Cepat

- Panduan Format PDF: Tersedia untuk dibaca secara offline atau dicetak ([gstack-guide.pdf](#)).
- Panduan Format Markdown (.MD): Tersedia untuk dibaca di editor teks atau dijadikan referensi AI prompt ([gstack-guide.md](#)).
- Tautan Repositori GitHub: <https://github.com/garrytan/gstack>