

Kiat dan alat AI #1 — Oktober 2026

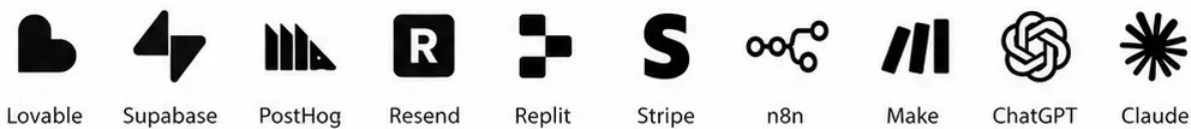
28 Sep 2026 · 12 menit dibaca · Oleh [Viktor Zaika](#)

Tiga kombinasi alat praktis untuk menguji ide, membangun MVP, dan mengotomatiskan pekerjaan, dilengkapi prompt, diagram, serta panduan unduhan.

- Perkakas AI
- Manajemen produk
- MVP
- Otomatisasi
- Tanpa kode

prompts&stacks

for your growth



mampu membuat proyek Anda sendiri tidak lagi kemampuan dari insinyur saja manajer produk yang baik harus mampu membangun PoCs, bahkan MVP dan tes tangan dan untuk memiliki sesuatu yang membantu mereka untuk mengkomunikasikan ide-ide. Tidak hanya itu tapi hampir semua orang cerdas teknologi saat ini harus menggunakan beberapa pipa produktivitas pribadi yang meningkatkan kinerja mereka, memberi mereka lebih banyak waktu luang dalam sehari, membantu mereka untuk berpikir, meningkatkan kecanggihan online mereka dll. Jadi di sini saya akan memberi Anda yang paling penting, mudah dibaca dan mudah untuk menggunakan tips, yang akan sangat membantu Anda dalam skenario ini:

 Dapat Disukai Halaman + UI	 Supabase Data + Auth	 PostHog Acara + replay	 Kirim Ulang Surel
 Balas Aplikasi backend	 n8n Aliran kerja	 Membuat Integrasi	 Stripe Pembayaran

Alat-alat yang dibahas di bawah ini, tekan kartu apa saja untuk dokumentasi produk tersebut atau halaman integrasi ikon yang dapat dipindah-pindahkan dari Lovable; yang lainnya dari Simple Ikon.

1.kau ingin menguji ide, bukan membangun perusahaan

Anggap saja kau punya ide untuk beberapa SaaS, pasar, alat AI, apapun.

Tumpukan saya untuk ini akan menjadi:

Lovable + Supabase + PostHog + Resend

Dapat Disukai untuk halaman dan mungkin beberapa interaksi yang sangat dasar. Supabase untuk menyimpan signups dan apa pun backend kecil yang Anda butuhkan. PostHog untuk benar-benar memahami apa yang orang lakukan bukan hanya melihat pada counter halaman. Kirim Ulang untuk surel konfirmasi, daftar tunggu, undangan akses awal dll.

PostHog sangat berguna di sini karena Anda dapat pergi dari nomor lalu lintas dasar ke tujuan konversi dan corong dan bahkan sesi replay, berarti Anda benar-benar dapat melihat di mana orang klik, di mana mereka bingung dan di mana mereka menghilang.

Sebuah twist benar-benar berguna di sini adalah untuk palsu bagian yang mahal sebelum membangunnya.

Anggap saja ide Anda adalah layanan AI yang menganalisis PDF dan mengembalikan laporan yang rumit. Halaman pendaratan Anda sudah dapat memungkinkan pengguna mengunggah PDF dan tekan Analyze. Di balik layar Anda dapat menerima berkas, melakukan 10 permintaan pertama secara manual, dan email hasilnya kembali.

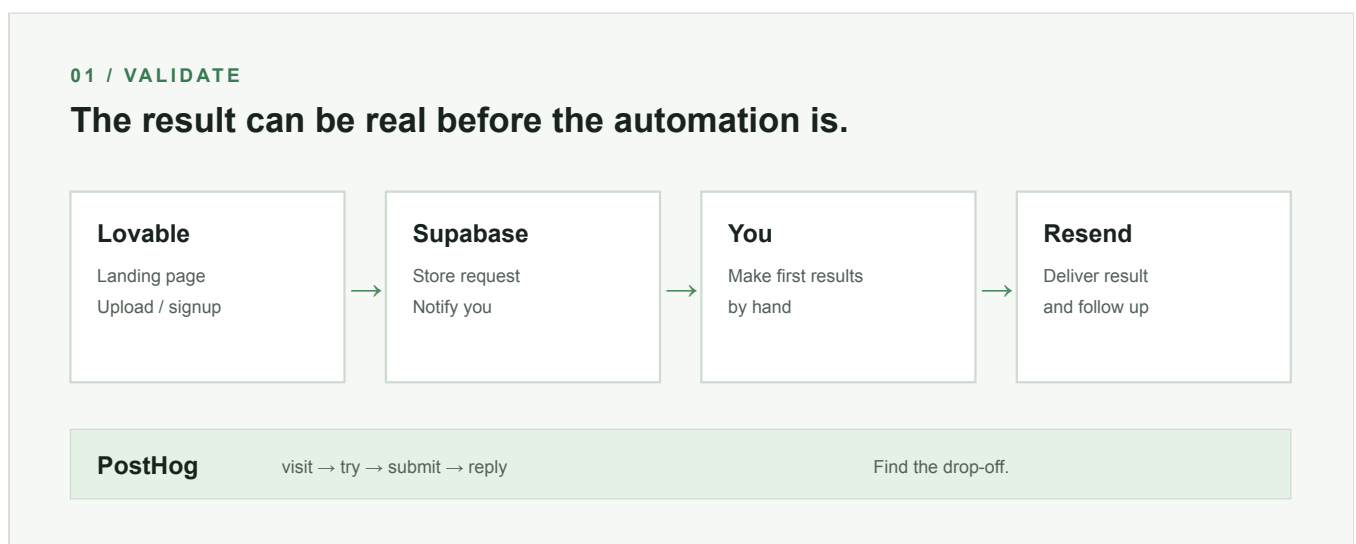
Ya, otomatisasi palsu.

Tapi kau menguji apakah orang benar-benar menginginkan hasilnya sebelum menghabiskan dua minggu membangun sistem yang membuatnya secara otomatis.

Untuk skenario tertentu:

Halaman pendaratan → signup/upload → Supabase → pemberitahuan kepada Anda → pembuatan hasil secara manual → Kirim surel

Setelah orang benar-benar mulai menggunakannya, automate tengah.



Satu kemungkinan aliran kerja validasi. Permintaan tes hasil buatan tangan sebelum otomatisasi mahal ada. Lovableés Supabase integrasi, Kanal PostHog, dan Kirim ulang API surel Diagram oleh Viktor Zaika. tekan gambar untuk melihat ukuran penuh.

Hal lain yang berguna: melacak peristiwa, tidak hanya kunjungan.

Jangan hanya mengukur:

1.400 orang berkunjung

Ukuran:

382 diklik cobalah

117 mulai formulir

83 menyelesaikannya

19 menjawab ke email

Itu memberitahu Anda cara yang lebih tentang apakah ide ini menarik. Kalkulator corong konversi dapat menunjukkan di mana penurunan terbesar terjadi sebelum Anda menempatkan langkah-langkah ke dalam corong PostHog.

Validate ide Anda

Salin prompt ↗

Saya ingin memvalidasi ide produk ini:

Saya TIDAK ingin membangun produk lengkap belum.

Mendesain kemungkinan terkecil arsitektur validasi menggunakan Lovable, Supabase, PostHog dan Resend.

Jelaskan:

1. apa yang seharusnya dibangun
2. apa yang dapat secara manual dipalsukan di belakang layar
3. data apa yang harus disimpan di Supabase
4. Peristiwa PostHog mana yang harus saya lacak
5. corong konversi apa yang harus aku ciptakan.
6. email apa yang harus dikirim melalui Resend
7. Apa yang akan dihitung sebagai bukti yang cukup untuk melanjutkan ke MVP sebuah Jaga arsitektur dengan sengaja sederhana. Jangan memperkenalkan layanan tambahan kecuali ada alasan yang sangat bagus.

Anda benar-benar membutuhkan aplikasi kerja sekarang

Saat kau butuh pengguna untuk masuk, simpan sesuatu, kembalilah besok, mungkin bayar, mungkin upload file, maka kau sudah membuat aplikasi.

Di sini saya akan membagi menjadi dua jalan.

Kasus A: normal SaaS atau aplikasi web

Lovable + Supabase + Stripe + Resend + PostHog

Ini cukup jujur untuk sejumlah besar MVP. hubungkan sebuah projek Supabase yang anda miliki ke Lovable; Lovable saat ini juga menawarkan backend built-in sendiri secara baku.

Supabase memberi Anda aktual Postgres di bawah, ditambah Keamanan tingkat aut dan baris, jadi frontend Anda yang dihasilkan tidak harus menjadi model keamanan Anda.

Dan ini adalah salah satu hal teknis yang membosankan yang aplikasi buatan AI dapat mengacaukan mengerikan: otorisasi bukanlah hal yang sama dengan menyembunyikan tombol.

Jika Alice log in, dia tidak boleh dapat mengubah beberapa permintaan dan tiba-tiba membaca catatan Bob.

Itu yang mana Supabase RLS sebenarnya penting. sementara RLS menentukan baris mana yang diijinkan pengguna untuk mengaksesnya. Panduan Supabase RLS menjelaskan bagaimana database hibah dan kebijakan bekerja sama.

Untuk operasi sisi server, panggilan API, Stripe webhook atau beberapa langkah generasi AI, Fungsi Tepi Supabase adalah lapisan berikutnya yang cukup nyaman. Mereka menjalankan sisi server TypeScript dan dapat mengintegrasikan dengan Auth, Postgres dan API eksternal.

Seorang MVP yang sangat normal bisa oleh karena itu terlihat seperti:

frontend Lovable → Supabase Auth → Postgres dengan RLS → Fungsi Tepi → API eksternal

Lalu:

Stripe webhook → Fungsi Edge → langganan update dalam Postgres

Dan:

aksi pengguna → Fungsi Tepi → Kirim Ulang

Itu sudah cukup arsitektur yang nyata.

Kasus B: backend itu sendiri adalah produk

Jika Anda membangun sesuatu dengan pekerja, API yang tidak biasa, menggores, memproses file, antrian, paket Python atau logika backend aneh, saya mungkin akan pindah ke Balas Daripada memaksa semuanya melalui pembangun frontend.

Replit Agent dapat menghasilkan aplikasi dari bahasa alami, sementara Replit juga menyediakan integrasi penyebaran dan basis data di lingkungan yang sama.

Misalnya:

Anda ingin membangun alat pemantau pesaing.

Masukan:

URL pesaing

Sistem:

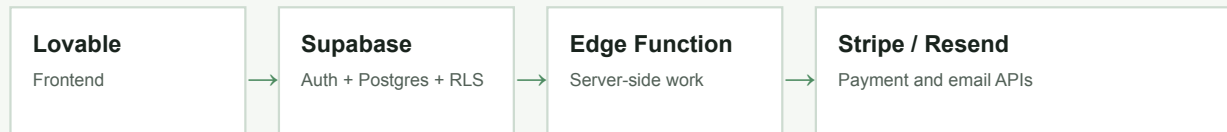
mengambil halaman → dibandingkan dengan versi sebelumnya → Tanyakan LLM apa yang berubah secara materi → simpan hasil → kirim pemberitahuan

Itu sudah jauh lebih alami aplikasi backend dari UI cukup dihasilkan.

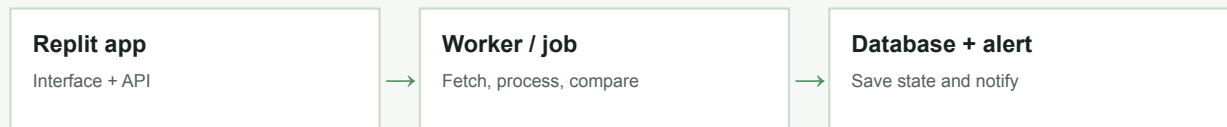
02 / BUILD

Choose the architecture by the hard part.

A / WEB SAAS



B / BACKEND-HEAVY APP



Dua jalur ilustratif MVP, dipilih oleh apa yang sebenarnya dibutuhkan oleh produk. Jalur aplikasi web Lovable Supabase connection dan Fungsi Tepi; jalur backend-berat menggunakan Replit AgentPanah adalah contoh arsitektur, bukan klaim bahwa produk-produk ini terintegrasi secara otomatis Diagram oleh Viktor Zaika.

Satu aturan yang sangat berguna di sini:

Jangan taruh kunci API rahasia Anda di ujung depan.

Kunci OpenAI, rahasia Stripe, tombol Resend, apa pun. Jika browser dapat melihatnya, asumsikan orang lain dapat melihatnya juga.

Menempatkan operasi ini di belakang titik akhir server atau fungsi. Panduan keamanan data Supabase. tutup batas sisi server ini.

Aku ingin membuat MVP ini:

Bantu aku memilih antara:

A. Lovable + Supabase

B. Replit

C. kombinasi keduanya

Jangan memilih alat mana yang lebih Anda sukai. Pilih berdasarkan persyaratan teknis yang sebenarnya.

Pertama mengidentifikasi:

1. persyaratan frontend

2. Otentikasi

3 basis data entitas dan hubungan

Aturan otorisasi 4.

5. API eksternal

6 latar belakang atau panjang pekerjaan berjalan

Penyimpanan file 7.

Delapan pembayaran.

9 email transaksional

10 analisis.

Kemudian mengusulkan arsitektur terkecil yang masih secara teknis waras.

Katakan padaku operasi mana yang harus dilakukan sisi server dan rahasia mana yang harus tidak pernah terungkap di frontend.

Jika menggunakan Supabase, mengusulkan tabel dan aturan RLS juga.

Kau ingin berhenti melakukan sesuatu secara manual setiap hari.

Ini mungkin kategori yang paling berguna bagi kebanyakan orang.

Kau tidak perlu produk.

Kau perlu pipa.

Default saya di sini mungkin akan:

t8n

Kemudian menghubungkan apa pun yang sudah Anda gunakan di sekitarnya.

Google Drive, Gmail, Slack, Telegram, Notion, Airtable, Supabase, OpenAI, Claude, API dll.

Jika Anda ingin sesuatu yang lebih mudah dan sangat visual, Membuat adalah alternatif yang sangat wajar dan mendukung ribuan integrasi, termasuk koneksi HTTP generik ketika layanan yang Anda butuhkan tidak secara langsung didukung.

Dan bagian yang menarik dimulai ketika Anda berhenti berpikir:

pemicu → aksi

dan mulai berpikir:

masukan → mengerti → Putuskan → bertindak → ingat

Sebagai contoh, di sini sebuah jalur pribadi yang benar-benar berguna:

Gmail → n8n → LLM → mengklasifikasikan → tenggat waktu ekstrak → membuat tugas → pemberitahuan Telegram

Tapi jangan minta LLM untuk hanya membaca email saya dan memutuskan semuanya.

Buat kembali data terstruktur:

```
{  
  "important": true,  
  "category": "invoice",  
  "deadline": "2026-10-14",  
  "action_required": "pay invoice",  
  "confidence": 0.94  
}
```

Copy

Kini aliran kerja dapat menggunakan keluaran yang dapat diandalkan. N8n's Terstruktur Keluaran Parser adalah salah satu cara untuk menegaskan bentuk hasil.

Itu adalah detail teknis kecil yang membuat automasi AI jauh lebih rapuh.

Satu lagi:

Jangan biarkan AI melakukan tindakan yang tidak dapat diubah secara langsung kapanpun Anda dapat menghindarinya.

Misalnya:

email → AI berpikir ini adalah spam → DELETE

Buruk.

Lebih baik:

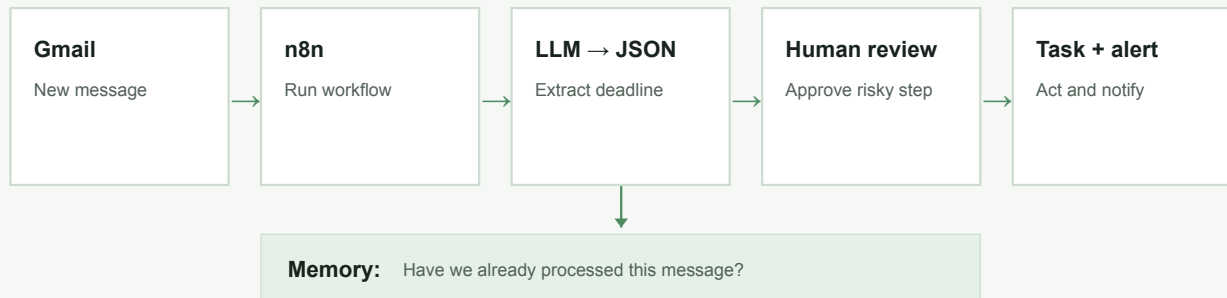
email → AI berpikir ini adalah spam → label mungkin spam → Anda mengkonfirmasi

Sama dengan mengirim email, menerbitkan konten, menghapus berkas, memperbarui data produksi dll.

Letakkan langkah perkenan manusia dimana harga dari keputusan yang salah tinggi.

03 / AUTOMATE

Input → understand → decide → act → remember



Sebuah aliran kerja ilustrasi sehari-hari. Keluaran terstruktur memungkinkan langkah-langkah selanjutnya menggunakan bidang; ulasan manusia Diagram oleh Viktor Zaika.

Beberapa jalur pipa yang sebenarnya berguna

Jalur pipa riset pribadi

RSS / websites / newsletters → n8n → ekstrak teks → deduplikasi → LLM ratresi relevansi → simpan yang menarik → satu digest Telegram

Alih-alih membaca 80 hal, Anda membaca 7.

Jalur pipa pertemuan

Pertemuan kalender berakhir → transkrip muncul → LLM ekstraks keputusan dan tindakan → menciptakan tugas → kirim ringkasan pendek

Baris pipa isi

Ide dalam Telegram → n8n → save to Notion → LLM mengkategorikan itu → menghasilkan pertanyaan penelitian → menunggu

Perhatikan bahwa saya tidak mengatakan bahwa ciri khas saya adalah yang terbaik.

Bagian yang berguna mungkin hanya menghapus pekerjaan organisasi bodoh di sekitar menulis itu.

observasi online

Google Alert / Reddit search / X atau eksternal APIs / review produk → workflow → hapus duplikat → klasifikasi sentimen/topik hanya memberitahu ketika sesuatu yang menarik terjadi

Jadi, bukannya mengoogle dirimu sendiri, perusahaan atau pesaing setiap beberapa hari, sistem melakukan bagian itu.

4. Bagian yang benar-benar menarik: biarkan AI menghasilkan data, bukan prosa

Ini mungkin salah satu perbedaan terbesar antara otomasi yang menyenangkan dan sesuatu yang dapat Anda andalkan.

Jangan tanya:

Baca tiket dukungan ini dan katakan padaku apa yang harus dilakukan.

Tanya:

```
Analyze this support ticket and return ONLY valid JSON using this schema:  
category: billing | bug | feature_request | account | other  
urgency: 1 to 5  
requires_human: boolean  
customer_sentiment: positive | neutral | negative  
summary: maximum 200 characters
```

Copy

Sekarang bagian lain dari pipa Anda benar-benar dapat rute itu.

urgensi $\geq 4 \rightarrow$ Slack

antrian tagihan $\rightarrow$ tagihan

Memerlukan_manusia = palsu $\rightarrow$ menghasilkan draft

LLM jauh lebih berguna di dalam perangkat lunak ketika mereka diperlakukan sebagai salah satu fungsi sedikit tidak dapat diandalkan dalam sistem deterministik yang lebih besar, daripada beberapa orang ajaib mengendalikan semuanya.

04 / DECIDE

Give each part of the job to the right kind of logic.

Normal code

```
price > 1000
```

Exact comparisons, IDs, math, access
control, deduplication

LLM + schema

```
{ urgency: 4, category: "billing" }
```

Messy text $\rightarrow$ structured output
Human approves expensive actions

Gunakan kode normal untuk aturan yang tepat dan LLM untuk input fuzzy. Sisi JSON mengilustrasikan jenis ruas sebuah langkah keluaran terstruktur Diagram oleh Viktor Zaika.

Dan memberikan memori kerja Anda

Anggap saja kau membuat jalur pencarian AI.

Tanpa memori:

Senin menemukan artikel X.

Selasa menemukan artikel X lagi.

Rabu menemukan seseorang membahas artikel X dan dengan bangga memberitahu Anda lagi.

Simpan apa yang sudah kau proses.

Ini dapat secara harfiah menjadi salah satu tabel Supabase:

```
source_url  
hash  
first_seen  
summary  
topic
```

Copy

Kemudian sebelum memproses sesuatu yang mahal:

Apa aku sudah melihat ini?

Jika ya, lewatkan saja.

Tiba-tiba pipamu terasa 5 kali lebih pintar.

Jangan otomatis membuat semuanya AI

Kedengarannya bodoh di pos bernama AI Tips&Tools, tapi itu penting.

Jika ini berhasil:

```
price > 1000
```

Copy

jangan menggantinya dengan:

```
Ask GPT whether this price appears relatively expensive in the context of the transact
```

Copy

Kumohon.

Kode normal lebih cepat, lebih murah dan deterministik.

Gunakan LLM di mana masukan fuzzy.

Memahami bahasa.

Klasifikasi di mana aturan menjadi mengerikan.

Summarisasi.

Ekstraksi dari dokumen berantakan.

Mencocokkan sesuatu dengan makna.

Menjangkitkan sesuatu.

Tinggalkan matematika, ID, perbandingan yang tepat, akses kontrol dan yang paling membosankan aturan bisnis membosankan kode normal.

Membosankan adalah fantastis ketika bekerja setiap kali.

Tiga tumpukan yang aku ingat

Uji apakah ada yang peduli

Lovable + Supabase + PostHog + Resend

Membangun sesuatu yang dapat digunakan orang-orang

Lovable + Supabase untuk normal web SaaS

atau

Balikkan saat logika backend menjadi bagian yang menarik

Kemudian tambahkan Stripe, Resend dan PostHog ketika Anda benar-benar membutuhkannya.

Otomatiskan pekerjaan Anda sendiri

n8n + alat yang telah anda gunakan + LLM + beberapa basis data kecil untuk memori

Dan mungkin yang paling berguna prompt dari seluruh posting ini:

Otomatiskan pekerjaan Anda

Salin prompt ↗

Saat ini saya melakukan ini secara manual:

[DESCRIBE THE PROCESS IN NORMAL HUMAN LANGUAGE]

Mendesain otomatisasi untuk itu.

Pisahkan aliran kerja ke:

Langkah menentukan yang seharusnya menggunakan logika normal,

Langkah kabur di mana LLM sebenarnya masuk akal,

Tindakan yang harus membutuhkan persetujuan manusia,

negara yang perlu disimpan sehingga workflow ingat apa yang sudah dilakukan.

Lebih suka n8n dan layanan yang sudah saya gunakan. Pertahankan jumlah bagian bergerak sekecil mungkin.

Untuk setiap langkah LLM, definisikan keluaran JSON terstruktur yang tepat itu harus kembali.

Juga katakan padaku bagaimana alur kerja ini bisa gagal dan apa yang harus saya log jadi saya bisa debug nanti.

Karena ini adalah intinya, bukan berarti kau harus menjadi insinyur, tapi juga tidak berarti teknik berhenti berpengaruh.

Sekarang Anda dapat membangun sejumlah hal yang tidak masuk akal tanpa mengetahui segala sesuatu di bawahnya. dan di mana Anda harus benar-benar membiarkannya freestyle.