

Bab 1: Cara Berpikir Founder

Membangun startup bukan dimulai dari logo, nama perusahaan, atau pitch deck. Semua itu penting pada waktunya, tetapi bukan fondasi pertama. Fondasi pertama adalah cara berpikir founder.

Founder adalah orang yang memulai dan bertanggung jawab membangun arah awal sebuah usaha. Dalam konteks startup, founder bukan hanya “pemilik ide”. Founder adalah orang yang bersedia masuk ke wilayah yang belum jelas, mencari masalah nyata, menguji solusi, berbicara dengan pelanggan, mengambil keputusan dengan data yang belum sempurna, dan tetap belajar ketika rencana awal ternyata salah.

Ini berbeda dari gambaran populer bahwa founder harus selalu percaya diri, karismatik, dan punya visi besar sejak hari pertama. Kepercayaan diri bisa membantu, tetapi terlalu percaya diri juga bisa berbahaya. Visi bisa memberi arah, tetapi visi yang tidak diuji dapat berubah menjadi ilusi. Karena itu, cara berpikir founder yang sehat bukan “saya pasti benar”, melainkan:

> “Saya punya dugaan. Saya akan mengujinya secepat dan semurah mungkin. Jika pasar memberi bukti yang berbeda, saya akan belajar dan menyesuaikan.”

Bab ini membangun fondasi mental tersebut: memahami risiko, ketidakpastian, eksperimen, kecepatan belajar, ketahanan, dan perbedaan antara bisnis kecil, startup, serta perusahaan teknologi yang dapat tumbuh cepat.

Startup Bukan Sekadar Bisnis Baru

Setiap startup adalah bisnis baru, tetapi tidak setiap bisnis baru adalah startup.

Jika seseorang membuka kedai kopi di area kampus, ia sedang membangun bisnis. Ia perlu memilih lokasi, menentukan menu, menghitung biaya bahan baku, mempekerjakan barista, melayani pelanggan, dan mencari laba. Ini bisnis nyata dan berharga. Namun model dasarnya sudah cukup dikenal: orang datang, membeli minuman, membayar, lalu kembali jika puas.

Startup berbeda karena biasanya belum tahu dengan jelas model bisnis mana yang akan berhasil. Steve Blank dan Bob Dorf mendefinisikan startup sebagai organisasi sementara yang dibentuk untuk mencari model bisnis yang dapat diulang dan dapat diskalakan (Blank & Dorf, 2012). Kata pentingnya adalah mencari.

Model bisnis adalah cara sebuah usaha menciptakan nilai, memberikan nilai kepada pelanggan, dan menangkap sebagian nilai itu sebagai pendapatan. Contohnya:

- Kedai kopi menciptakan nilai melalui minuman, tempat, dan pengalaman.
- Aplikasi kasir untuk UMKM menciptakan nilai melalui pencatatan transaksi yang lebih rapi.
- Marketplace menciptakan nilai dengan mempertemukan penjual dan pembeli.
- Platform belajar daring menciptakan nilai dengan menyediakan materi dan pengalaman belajar.

Pada bisnis yang sudah umum, banyak komponen model bisnis sudah diketahui. Pada startup, banyak komponen masih berupa pertanyaan:

- Siapa pelanggan pertama yang benar-benar membutuhkan solusi ini?
- Masalah apa yang paling menyakitkan bagi mereka?
- Apakah mereka bersedia membayar?
- Berapa harga yang masuk akal?
- Bagaimana cara menjangkau mereka secara konsisten?
- Apakah produk ini bisa tumbuh tanpa biaya yang selalu naik sebanding dengan jumlah pelanggan?

Eric Ries juga menekankan bahwa startup bekerja dalam kondisi ketidakpastian ekstrem, bukan sekadar membuat produk baru (Ries, 2011). Artinya, startup tidak hanya bertanya, "Bisakah kita membuat ini?" tetapi juga, "Apakah ini layak dibuat, untuk siapa, dan dengan model bisnis apa?"

Contoh sederhana: membuat aplikasi pencatat keuangan pribadi mungkin secara teknis bisa dilakukan. Tetapi pertanyaan startup adalah:

- Apakah orang yang dituju benar-benar merasa sulit mengatur uang?
- Apakah mereka sudah memakai Excel, catatan ponsel, atau aplikasi lain?
- Mengapa mereka akan pindah?
- Apakah mereka mau membayar, atau hanya mau memakai jika gratis?
- Apakah masalahnya cukup sering terjadi sehingga mereka akan memakai aplikasi ini berulang?

Founder yang baik tidak berhenti pada kemampuan membuat produk. Ia belajar memahami apakah produk itu penting bagi pasar.

Risiko dan Ketidakpastian: Dua Hal yang Sering Tertukar

Dalam kehidupan sehari-hari, kita sering memakai kata “risiko” untuk semua hal yang belum pasti. Namun dalam membangun startup, berguna untuk membedakan risiko dan ketidakpastian.

Risiko adalah keadaan ketika kemungkinan hasil buruk dapat diperkirakan dengan cukup jelas. Misalnya, seorang pemilik toko membeli 100 kotak minuman untuk dijual. Ia tahu harga beli, harga jual, tanggal kedaluwarsa, dan kira-kira permintaan harian. Tetap ada kemungkinan barang tidak habis, tetapi beberapa angka bisa dihitung.

Ketidakpastian adalah keadaan ketika kita belum tahu kemungkinan hasilnya, bahkan kadang belum tahu faktor apa saja yang menentukan hasil tersebut. Frank Knight membedakan risiko yang dapat dihitung dari ketidakpastian yang tidak mudah diberi probabilitas pasti (Knight, 1921). Startup banyak berhadapan dengan jenis kedua.

Misalnya, Anda ingin membuat platform yang menghubungkan guru les privat dengan siswa SMA. Beberapa pertanyaan masih tidak jelas:

- Apakah orang tua lebih percaya platform baru atau rekomendasi tetangga?
- Apakah guru les mau membayar komisi?
- Apakah siswa memilih berdasarkan harga, lokasi, reputasi, atau kecocokan gaya mengajar?
- Apakah masalah terbesar sebenarnya mencari guru, menjadwalkan les, atau memastikan kualitas pengajaran?

Ini bukan sekadar risiko yang bisa dihitung dari awal. Ini ketidakpastian yang perlu dipelajari.

Perbedaan ini mengubah cara founder bekerja. Jika masalahnya risiko, kita bisa membuat perhitungan dan rencana. Jika masalahnya ketidakpastian, kita perlu melakukan pencarian dan eksperimen.

Contoh:

- Jika Anda sudah tahu bahwa biaya server per pengguna adalah Rp2.000 per bulan dan harga langganan Rp20.000, itu risiko operasional yang bisa dihitung.
- Jika Anda belum tahu apakah pengguna mau membayar Rp20.000 per bulan, itu ketidakpastian pasar yang perlu diuji.

Kesalahan umum founder pemula adalah memperlakukan ketidakpastian seperti risiko biasa. Mereka membuat proyeksi keuangan lima tahun seolah-olah semua variabel sudah diketahui. Padahal yang lebih penting pada tahap awal adalah membuktikan asumsi-asumsi dasar.

Asumsi: Batu Bata yang Belum Teruji

Sebelum ada fakta, biasanya ada asumsi.

Asumsi adalah pernyataan yang kita anggap benar untuk sementara, tetapi belum terbukti cukup kuat. Setiap ide startup dibangun dari banyak asumsi. Founder yang matang tidak malu mengakui asumsi. Justru ia berusaha menuliskannya agar bisa diuji.

Misalnya, ide Anda adalah aplikasi pengingat minum air untuk pekerja kantoran. Beberapa asumsi di balik ide itu mungkin:

1. Pekerja kantoran sering lupa minum air.
2. Lupa minum air adalah masalah yang cukup penting bagi mereka.
3. Mereka mau mengunduh aplikasi khusus untuk masalah itu.
4. Notifikasi dari aplikasi akan membantu, bukan mengganggu.
5. Sebagian pengguna bersedia membayar fitur premium.
6. Aplikasi ini bisa menjangkau pengguna dengan biaya pemasaran yang masuk akal.

Jika semua asumsi itu benar, ide tersebut mungkin punya peluang. Tetapi jika asumsi kedua salah—misalnya orang memang lupa minum air tetapi tidak menganggapnya masalah penting—maka produk mungkin sulit tumbuh. Jika asumsi keempat salah—notifikasi dianggap mengganggu—maka produk bisa cepat dihapus.

Founder pemula sering jatuh cinta pada solusi. Founder yang lebih matang jatuh cinta pada proses menemukan kebenaran.

Ini bukan berarti founder tidak boleh punya intuisi. Intuisi berguna sebagai titik awal. Namun intuisi harus diperlakukan sebagai hipotesis, bukan kebenaran final.

Hipotesis adalah dugaan yang dirumuskan cukup jelas sehingga bisa diuji. Contoh hipotesis yang lebih baik:

> “Pekerja kantoran usia 22–35 tahun yang bekerja lebih dari 8 jam di depan laptop sering lupa minum air dan bersedia memakai aplikasi pengingat jika notifikasinya dapat disesuaikan dengan jadwal kerja.”

Hipotesis ini lebih mudah diuji daripada kalimat “orang butuh aplikasi kesehatan”.

Eksperimen: Cara Belajar Tanpa Membakar Terlalu Banyak Uang

Karena startup bergerak dalam ketidakpastian, founder membutuhkan cara belajar yang cepat dan murah. Cara itu disebut eksperimen.

Eksperimen adalah tindakan terencana untuk menguji sebuah hipotesis. Dalam startup, eksperimen tidak selalu berarti eksperimen laboratorium seperti dalam ilmu fisika atau kimia. Eksperimen bisa berupa wawancara pelanggan, landing page, prototipe, pre-order, demo manual, atau penjualan langsung.

Eric Ries mempopulerkan siklus build-measure-learn, yaitu membangun sesuatu yang cukup untuk diuji, mengukur respons, lalu belajar dari hasilnya (Ries, 2011). Tujuannya bukan membuat produk besar secepat mungkin, melainkan mendapatkan pembelajaran yang valid secepat mungkin.

Misalnya, Anda ingin membuat layanan langganan makanan sehat untuk pekerja kantor. Anda bisa langsung menyewa dapur, membuat aplikasi, merekrut kurir, dan membeli bahan baku besar-besaran. Tetapi itu mahal. Alternatif yang lebih bijak adalah eksperimen kecil:

1. Buat menu contoh selama satu minggu.
2. Tawarkan kepada 20 pekerja kantor di satu gedung.
3. Minta mereka membayar untuk paket percobaan.
4. Catat berapa orang tertarik, berapa yang benar-benar membayar, menu apa yang dipilih, dan keluhan apa yang muncul.
5. Ulangi dengan penyesuaian.

Eksperimen seperti ini tidak membuktikan seluruh masa depan bisnis. Namun ia memberi bukti lebih kuat daripada sekadar bertanya, “Kalau ada makanan sehat, mau beli tidak?” Banyak orang akan menjawab “mau” untuk bersikap sopan. Komitmen nyata seperti membayar, mendaftar, atau memakai berulang jauh lebih kuat daripada pujian.

Dalam startup, pujian tidak selalu berarti permintaan pasar. Kalimat “idenya bagus” belum tentu berarti “saya mau membayar sekarang”.

Kecepatan Belajar Lebih Penting daripada Kecepatan Sibuk

Banyak founder bangga karena sibuk. Mereka membuat logo, membuat akun media sosial, mengikuti banyak acara, mengedit pitch deck, berdiskusi panjang tentang fitur, dan mengadakan rapat. Kesibukan terasa seperti kemajuan. Namun dalam startup, pertanyaan pentingnya adalah:

> “Apa yang kita pelajari minggu ini yang mengurangi ketidakpastian terbesar?”

Kecepatan belajar adalah kemampuan tim untuk mengubah asumsi menjadi bukti, lalu memakai bukti itu untuk mengambil keputusan. Ini berbeda dari sekadar bekerja cepat.

Contoh kesibukan yang belum tentu produktif:

- Menghabiskan tiga minggu memilih warna aplikasi, padahal belum tahu apakah pelanggan mau memakai aplikasi itu.
- Membuat 30 fitur, padahal belum tahu satu fitur utama mana yang benar-benar penting.
- Membuat pitch deck investor, padahal belum pernah berbicara dengan calon pelanggan.

Contoh kecepatan belajar yang baik:

- Dalam satu minggu, mewawancarai 15 calon pelanggan yang sesuai target.
- Menemukan bahwa masalah utama bukan harga, melainkan kepercayaan.
- Mengubah eksperimen berikutnya untuk menguji bukti sosial, testimoni, atau jaminan kualitas.
- Mendapatkan 5 pelanggan pertama yang bersedia membayar.

Rita McGrath dan Ian MacMillan memperkenalkan gagasan discovery-driven planning, yaitu perencanaan untuk situasi yang sangat tidak pasti dengan cara mengidentifikasi asumsi penting dan mengujinya secara bertahap (McGrath & MacMillan, 1995). Ini sejalan dengan cara kerja startup: rencana tetap perlu, tetapi rencana harus diperlakukan sebagai alat belajar, bukan kitab suci.

Founder perlu bertanya secara rutin:

- Asumsi paling berbahaya apa yang belum terbukti?
- Bukti apa yang akan membuat kita lebih percaya?
- Eksperimen termurah apa yang bisa dilakukan?
- Jika hasilnya negatif, keputusan apa yang akan kita ambil?

Kecepatan belajar juga menuntut kejujuran. Jika data menunjukkan bahwa orang tidak tertarik, founder tidak boleh hanya mencari alasan untuk mempertahankan ego. Data yang buruk tetapi jujur lebih berguna daripada laporan indah yang menipu diri sendiri.

Founder Harus Nyaman dengan Ketidaknyamanan

Membangun startup berarti sering berada di antara dua keadaan: belum cukup bukti untuk merasa aman, tetapi harus tetap mengambil tindakan.

Anda mungkin belum tahu apakah produk akan diterima. Belum tahu apakah co-founder cocok dalam tekanan. Belum tahu apakah pelanggan akan membayar. Belum tahu apakah investor tertarik. Belum tahu apakah kompetitor akan bergerak lebih cepat.

Karena itu, founder perlu membangun ketahanan.

Ketahanan bukan berarti tidak pernah takut, tidak pernah lelah, atau selalu optimis. Ketahanan adalah kemampuan untuk tetap berpikir jernih, belajar, dan bertindak ketika menghadapi tekanan. Dalam praktik startup, ketahanan terlihat dari hal-hal sederhana:

- Tetap berbicara dengan pelanggan walau banyak yang menolak.
- Mau memperbaiki produk tanpa tersinggung oleh kritik.
- Bisa mengakui kesalahan keputusan.
- Tidak langsung menyerah karena satu eksperimen gagal.
- Tidak memaksakan ide yang jelas-jelas tidak mendapat respons pasar.

Angela Duckworth dan rekan-rekannya membahas konsep grit, yaitu ketekunan dan konsistensi minat terhadap tujuan jangka panjang (Duckworth et al., 2007). Namun dalam startup, ketekunan perlu dipahami dengan hati-hati. Ketekunan yang sehat bukan berarti keras kepala mempertahankan solusi yang salah. Ketekunan yang sehat berarti setia pada upaya menciptakan nilai, sambil tetap fleksibel terhadap cara mencapainya.

Misalnya, Anda ingin membantu UMKM mengelola keuangan. Awalnya Anda membuat aplikasi pencatatan otomatis. Setelah diuji, ternyata banyak pemilik warung tidak mau mengisi data sendiri. Jika Anda keras kepala, Anda akan terus memaksa aplikasi yang tidak dipakai. Jika Anda tangguh secara sehat, Anda akan kembali ke masalah dasarnya: UMKM butuh laporan sederhana, tetapi tidak punya waktu mencatat. Mungkin solusinya bukan aplikasi mandiri, melainkan layanan pencatatan berbasis WhatsApp, integrasi kasir, atau pendampingan ringan.

Founder yang tahan banting bukan founder yang tidak berubah. Founder yang tahan banting adalah founder yang tidak berhenti belajar.

Visi Penting, Tetapi Visi Harus Bertemu Realitas

Visi adalah gambaran tentang perubahan besar yang ingin diwujudkan. Visi memberi energi. Tanpa visi, founder mudah terjebak pada pekerjaan kecil harian dan kehilangan arah.

Contoh visi:

- Membantu pelaku UMKM mengelola bisnis dengan lebih profesional.
- Membuat pendidikan berkualitas lebih mudah diakses.
- Mengurangi pemborosan makanan di kota besar.
- Membantu klinik kecil melakukan administrasi dengan lebih rapi.

Namun visi harus dibedakan dari solusi awal. Visi bisa bertahan lama, sementara solusi dapat berubah.

Misalnya, visi Anda adalah “membantu mahasiswa menemukan kesempatan magang yang tepat”. Solusi awal mungkin aplikasi pencari magang. Setelah riset, Anda menemukan bahwa masalah terbesar bukan mencari lowongan, tetapi membuat CV, memahami proses seleksi, dan mendapat rekomendasi. Maka solusi bisa berubah menjadi platform persiapan karier, program mentoring, atau layanan kurasi kandidat.

Jika founder terlalu melekat pada solusi awal, ia bisa mengabaikan masalah yang lebih nyata. Tetapi jika founder punya visi yang cukup luas, ia dapat menyesuaikan solusi tanpa kehilangan arah.

Dalam pendekatan effectuation, Saras Sarasvathy menunjukkan bahwa wirausahawan berpengalaman sering memulai bukan dari prediksi masa depan yang sempurna, melainkan dari sarana yang mereka miliki: siapa mereka, apa yang mereka tahu, dan siapa yang mereka kenal (Sarasvathy, 2001). Mereka juga cenderung memikirkan kerugian yang sanggup ditanggung, bukan hanya keuntungan maksimum. Ini berguna bagi founder pemula.

Pertanyaan praktisnya:

- Apa kemampuan yang sudah saya miliki?
- Siapa orang yang bisa saya ajak bicara minggu ini?
- Masalah apa yang dekat dengan pengalaman saya?
- Eksperimen apa yang bisa saya lakukan tanpa mempertaruhkan semuanya?
- Kerugian apa yang masih sanggup saya tanggung jika eksperimen gagal?

Founder tidak perlu menunggu semua sumber daya lengkap. Tetapi founder perlu tahu batas aman agar keberanian tidak berubah menjadi kecerobohan.

Berpikir dalam “Affordable Loss”, Bukan Hanya “Potensi Untung”

Banyak orang memulai bisnis dengan bertanya, “Kalau berhasil, saya bisa untung berapa?” Pertanyaan ini wajar. Namun pada tahap awal startup, pertanyaan lain sering lebih sehat:

> “Jika eksperimen ini gagal, kerugian apa yang masih bisa saya terima?”

Inilah gagasan affordable loss: kerugian yang masih sanggup ditanggung. Konsep ini terkait dengan effectuation, yaitu cara berpikir kewirausahaan yang tidak hanya mencoba memprediksi masa depan, tetapi juga mengendalikan langkah kecil yang dapat diambil sekarang (Sarasvathy, 2001).

Contoh:

Anda ingin menguji bisnis kelas online untuk persiapan kerja. Ada dua pilihan:

- Pilihan A: menyewa studio mahal, membeli kamera profesional, membuat 50 video, dan membayar iklan besar sebelum ada pelanggan.

- Pilihan B: membuat satu kelas live kecil untuk 20 peserta, memakai alat sederhana, meminta pembayaran awal, lalu belajar dari respons peserta.

Pilihan A mungkin terlihat lebih “serius”, tetapi kerugiannya besar jika asumsi pasar salah. Pilihan B lebih cocok untuk tahap awal karena kerugian terbatas dan pembelajaran cepat.

Affordable loss bukan berarti selalu berpikir kecil. Ini berarti menyesuaikan ukuran taruhan dengan tingkat bukti. Ketika bukti masih lemah, taruhan harus kecil. Ketika bukti makin kuat, taruhan boleh diperbesar.

Prinsip ini sangat penting untuk founder muda, pekerja yang membangun startup sambil bekerja, atau siapa pun yang punya tanggung jawab keluarga. Keberanian bukan berarti mengabaikan realitas keuangan. Keberanian yang matang adalah berani menguji, tetapi tidak mempertaruhkan hidup secara membabi buta.

Bisnis Kecil, Startup, dan Perusahaan Teknologi yang Dapat Tumbuh Cepat

Agar tidak salah arah, founder perlu memahami tiga kategori yang sering bercampur: bisnis kecil, startup, dan perusahaan teknologi yang dapat tumbuh cepat.

Bisnis kecil

Bisnis kecil biasanya melayani pasar yang cukup jelas dengan model bisnis yang relatif dikenal. Contohnya:

- Warung makan.
- Salon lokal.
- Bengkel motor.
- Toko perlengkapan sekolah.
- Jasa laundry di dekat kos mahasiswa.

Bisnis kecil bisa sangat menguntungkan dan penting bagi ekonomi. Namun pertumbuhannya sering terkait langsung dengan sumber daya fisik dan operasional. Jika laundry ingin melayani dua kali lebih banyak pelanggan, mungkin perlu mesin tambahan, pegawai tambahan, dan tempat lebih besar.

Startup

Startup adalah organisasi yang masih mencari model bisnis yang dapat diulang dan diskalakan (Blank & Dorf, 2012). Startup belum tentu memakai teknologi canggih, tetapi banyak startup memakai teknologi karena teknologi dapat membantu eksperimen, distribusi, dan skala.

Contoh startup:

- Platform yang membantu toko kecil memesan stok dari distributor dengan lebih efisien.
- Aplikasi untuk menghubungkan pasien dengan klinik terdekat.
- Sistem langganan software akuntansi untuk UMKM.
- Marketplace jasa perbaikan rumah.

Ciri pentingnya bukan “ada aplikasi”, melainkan masih ada pencarian: siapa pelanggan terbaik, bagaimana model pendapatan, bagaimana akuisisi pelanggan, dan apakah model tersebut bisa diulang.

Perusahaan teknologi yang dapat tumbuh cepat

Sebagian startup menjadi perusahaan teknologi yang dapat tumbuh cepat. Pertumbuhan cepat biasanya mungkin ketika produk memiliki biaya tambahan per pengguna yang relatif rendah, distribusi digital, otomatisasi tinggi, atau efek jaringan.

Efek jaringan terjadi ketika nilai produk meningkat karena semakin banyak orang yang memakai produk tersebut. Contoh sederhana: aplikasi pesan menjadi lebih berguna jika banyak teman Anda juga menggunakannya. Marketplace juga bisa memiliki efek jaringan: lebih banyak penjual menarik lebih banyak pembeli, dan lebih banyak pembeli menarik lebih banyak penjual. Shapiro dan Varian membahas karakteristik ekonomi informasi seperti biaya produksi awal yang besar, biaya reproduksi yang rendah, serta pentingnya jaringan dalam ekonomi digital (Shapiro & Varian, 1999).

Namun tidak semua aplikasi memiliki efek jaringan. Aplikasi kalkulator di ponsel tidak otomatis menjadi lebih berguna karena banyak orang lain memakainya. Jadi founder perlu hati-hati menggunakan istilah ini.

Contoh perbandingan:

- Warung makan: makin banyak pelanggan biasanya perlu lebih banyak bahan, meja, karyawan, dan kapasitas dapur.
- Software akuntansi berbasis cloud: setelah sistem dibangun, menambah pelanggan baru mungkin tetap butuh biaya server dan dukungan pelanggan, tetapi tidak selalu naik sebanding dengan jumlah pelanggan.

- Marketplace: jika berhasil, pertumbuhan bisa cepat karena pengguna saling menambah nilai, tetapi membangun kepercayaan dan likuiditas awal sangat sulit.

Tidak ada kategori yang “lebih mulia”. Bisnis kecil yang sehat lebih baik daripada startup yang membakar uang tanpa pelanggan. Yang penting adalah founder memahami permainan yang sedang dimainkan.

Jika Anda membangun bisnis lokal, ukurannya mungkin laba stabil dan operasional rapi. Jika Anda membangun startup, ukuran awalnya adalah pembelajaran cepat dan bukti bahwa model bisnis bisa diulang. Jika Anda membangun perusahaan teknologi yang ingin tumbuh besar, Anda juga harus memikirkan skala, infrastruktur, distribusi, dan pertahanan kompetitif.

Founder Berpikir dari Pelanggan, Bukan dari Ego

Salah satu perubahan mental terbesar bagi founder adalah memindahkan pusat perhatian dari “ide saya” ke “masalah pelanggan”.

Pelanggan adalah orang atau organisasi yang memiliki masalah, membutuhkan solusi, dan dalam banyak kasus bersedia memberikan sesuatu sebagai imbalan: uang, waktu, data, perhatian, atau perubahan perilaku.

Founder pemula sering berkata:

> “Saya punya ide aplikasi. Menurut saya ini keren.”

Founder yang lebih matang bertanya:

> “Siapa yang punya masalah ini? Seberapa sering masalah itu terjadi? Apa yang mereka lakukan sekarang? Mengapa solusi sekarang belum cukup? Apa bukti bahwa mereka mau mencoba cara baru?”

Contoh:

Ide: aplikasi antrean digital untuk klinik kecil.

Cara berpikir dari ego:

- “Aplikasi ini akan punya fitur booking, chat dokter, rekam medis, pembayaran, dan rating.”
- “Kalau fiturnya lengkap, pasti klinik mau.”

Cara berpikir dari pelanggan:

- “Klinik kecil paling kesulitan di bagian mana: antrean, pencatatan pasien, pembayaran, atau komunikasi?”
- “Siapa yang paling terganggu: pemilik klinik, staf administrasi, dokter, atau pasien?”
- “Apakah klinik sudah memakai WhatsApp, buku tulis, Excel, atau software lama?”
- “Jika kami hanya menyelesaikan satu masalah dulu, masalah mana yang paling bernilai?”
- “Apakah pemilik klinik bersedia membayar Rp300.000 per bulan untuk solusi ini?”

Founder yang berpikir dari pelanggan tidak berarti selalu menuruti semua permintaan pelanggan. Pelanggan sering tahu rasa sakitnya, tetapi belum tentu tahu solusi terbaik. Tugas founder adalah mendengarkan masalah dengan serius, lalu merancang solusi yang masuk akal.

Optimisme Perlu Ditemani Disiplin

Startup membutuhkan optimisme. Tanpa optimisme, sulit memulai sesuatu yang belum pasti. Namun optimisme harus ditemani disiplin.

Optimisme tanpa disiplin menghasilkan kalimat seperti:

- “Nanti juga viral.”
- “Kalau aplikasinya jadi, pengguna pasti datang.”
- “Pasarnya besar, jadi kita pasti dapat bagian.”
- “Investor akan tertarik karena idenya bagus.”

Disiplin mengubah kalimat itu menjadi pertanyaan yang bisa diuji:

- “Channel apa yang paling mungkin membawa 100 pengguna pertama?”
- “Berapa persen pengunjung landing page yang mendaftar?”
- “Dari 20 orang yang bilang tertarik, berapa yang benar-benar membayar?”
- “Apa bukti bahwa masalah ini cukup mendesak?”
- “Apa metrik yang menunjukkan pengguna kembali memakai produk?”

Daniel Kahneman menjelaskan bahwa manusia sering memakai jalan pintas mental dalam mengambil keputusan, dan jalan pintas ini dapat membantu tetapi juga dapat menghasilkan bias (Kahneman, 2011). Dalam startup, bias yang sering muncul adalah melihat tanda kecil sebagai bukti besar, hanya mencari informasi yang mendukung ide sendiri, atau terlalu yakin pada prediksi.

Contoh bias:

Anda mewawancarai lima teman. Semuanya berkata ide Anda bagus. Anda lalu menyimpulkan pasar siap. Padahal teman mungkin tidak ingin mengecewakan Anda, dan mereka belum tentu pelanggan target. Bukti yang lebih kuat adalah orang asing dalam segmen target bersedia mencoba, membayar, atau memakai produk berulang.

Disiplin founder berarti membuat sistem agar tidak mudah tertipu oleh harapan sendiri. Misalnya:

- Menentukan kriteria sukses eksperimen sebelum eksperimen dimulai.
- Membedakan pujian dari komitmen.
- Mencatat data, bukan hanya mengandalkan ingatan.
- Mencari alasan mengapa ide bisa gagal.
- Mengundang kritik dari orang yang memahami pasar.

Founder tidak harus pesimis. Founder perlu optimis dalam tujuan, tetapi skeptis dalam bukti.

Keputusan Founder Sering Diambil dengan Data yang Belum Lengkap

Dalam perusahaan besar, banyak keputusan dapat dibuat dengan data historis. Dalam startup, data sering belum tersedia. Pelanggan masih sedikit. Produk masih berubah. Pasar belum terbukti. Karena itu, founder harus belajar mengambil keputusan dengan informasi yang tidak sempurna.

Ini bukan alasan untuk asal menebak. Justru karena data belum lengkap, founder perlu membedakan tiga tingkat keyakinan:

1. Dugaan: kita merasa mungkin benar, tetapi belum ada bukti kuat.
2. Indikasi: ada tanda awal, tetapi belum cukup untuk keputusan besar.
3. Bukti kuat: ada pola berulang dari perilaku nyata pelanggan.

Contoh:

- Dugaan: “Mahasiswa butuh aplikasi manajemen tugas kelompok.”
- Indikasi: “Dari 20 mahasiswa yang diwawancarai, 14 mengeluh koordinasi tugas kelompok sulit.”
- Bukti lebih kuat: “50 kelompok memakai prototipe selama satu bulan, 30 kelompok tetap aktif setiap minggu, dan 10 kelompok bersedia membayar untuk fitur premium.”

Founder harus menyesuaikan ukuran keputusan dengan kekuatan bukti. Jangan merekrut tim besar hanya berdasarkan dugaan. Jangan membangun sistem rumit hanya berdasarkan indikasi. Naikkan komitmen ketika bukti makin kuat.

Cara berpikir ini akan muncul berkali-kali di bab-bab berikutnya: validasi ide, MVP, penjualan awal, product-market fit, dan growth.

Ketika Gagal, Tanyakan “Apa yang Dipelajari?”

Kegagalan dalam startup bukan selalu tanda bahwa founder buruk. Kadang kegagalan adalah hasil alami dari menguji sesuatu yang tidak pasti. Namun kegagalan hanya berguna jika menghasilkan pembelajaran.

Ada kegagalan yang mahal dan tidak mendidik:

- Membangun produk selama satu tahun tanpa pernah bicara dengan pelanggan.
- Menghabiskan uang iklan tanpa tahu metrik yang diukur.
- Merekrut banyak orang sebelum model bisnis jelas.
- Menolak semua kritik sampai uang habis.

Ada juga kegagalan yang sehat:

- Menjalankan landing page selama satu minggu dan menemukan bahwa pesan produk tidak dipahami.
- Menawarkan harga tertentu dan mengetahui pelanggan menolak karena nilai belum jelas.
- Membuat prototipe sederhana dan menemukan bahwa fitur utama tidak dipakai.
- Menguji segmen pelanggan dan menyadari segmen lain lebih mendesak.

Kegagalan sehat memiliki tiga ciri:

1. Taruhannya terbatas.
2. Hipotesisnya jelas.
3. Ada keputusan setelah hasil keluar.

Misalnya, Anda menguji layanan konsultasi pajak untuk freelancer. Hipotesisnya: “Freelancer berpenghasilan di atas Rp10 juta per bulan bersedia membayar Rp250.000 untuk sesi konsultasi pajak satu jam.” Eksperimen: hubungi 50 freelancer yang sesuai segmen, tawarkan sesi berbayar, dan ukur jumlah yang membayar. Jika hanya 1 orang tertarik, Anda belajar bahwa segmen, harga, pesan, atau masalahnya perlu diperiksa ulang.

Ini jauh lebih baik daripada membuat platform lengkap selama enam bulan lalu baru mengetahui bahwa orang belum mau membayar.

Founder Perlu Membangun Identitas sebagai Pembelajar

Pada tahap awal, identitas paling berguna bagi founder bukan “saya CEO”, melainkan “saya pembelajar utama perusahaan”.

CEO terdengar tinggi. Pembelajar terdengar sederhana. Tetapi dalam startup, pembelajaran adalah pekerjaan inti. Founder harus belajar tentang pelanggan, pasar, produk, teknologi, penjualan, keuangan, hukum, perekrutan, dan diri sendiri.

Identitas sebagai pembelajar membuat founder lebih mudah:

- Bertanya tanpa malu.
- Mengakui tidak tahu.
- Menerima data yang mengecewakan.
- Mencari mentor.
- Membaca dan menguji konsep.
- Mengubah keputusan ketika bukti berubah.

Amy Edmondson meneliti psychological safety, yaitu kondisi ketika anggota tim merasa aman untuk menyampaikan pertanyaan, kesalahan, dan kekhawatiran tanpa takut dipermalukan atau dihukum secara interpersonal (Edmondson, 1999). Dalam tim startup, suasana seperti ini penting karena pembelajaran cepat membutuhkan kejujuran. Jika anggota tim takut menyampaikan kabar buruk, founder akan terlambat mengetahui masalah.

Contoh:

Seorang engineer tahu bahwa fitur yang diminta founder akan memakan waktu tiga minggu dan belum jelas dampaknya. Jika budaya tim tidak aman, ia mungkin diam dan mengerjakan saja. Jika budaya tim sehat, ia bisa berkata, "Sebelum kita bangun fitur ini, apakah kita sudah punya bukti bahwa pengguna membutuhkannya?" Pertanyaan seperti ini dapat menghemat waktu dan uang.

Founder menentukan nada awal budaya. Jika founder marah setiap kali ada kritik, tim akan belajar menyembunyikan masalah. Jika founder menghargai kejujuran, tim akan lebih cepat menemukan kenyataan.

Praktik Mental Harian Founder

Cara berpikir founder tidak dibentuk dari membaca satu bab saja. Ia terbentuk dari kebiasaan harian. Berikut beberapa praktik yang dapat langsung digunakan.

Tulis asumsi sebelum bertindak

Sebelum membuat fitur, kampanye, atau keputusan besar, tulis asumsi di baliknya.

Contoh:

> "Kami akan membuat fitur reminder otomatis karena kami percaya pengguna lupa menyelesaikan tugas setelah membuat daftar."

Lalu tanyakan:

- Apakah kita punya bukti pengguna lupa?
- Apakah reminder adalah solusi terbaik?
- Bagaimana kita tahu fitur ini berhasil?

Cari bukti perilaku, bukan hanya opini

Opini berguna, tetapi perilaku lebih kuat.

Opini:

> "Saya tertarik ikut kelas ini."

Perilaku:

> “Saya membayar uang pendaftaran.”

Opini:

> “Aplikasi ini sepertinya berguna.”

Perilaku:

> “Saya memakai aplikasi ini tiga kali minggu ini.”

Opini:

> “Saya akan rekomendasikan ke teman.”

Perilaku:

> “Saya benar-benar mengirim link ke lima teman.”

Buat eksperimen kecil

Jika belum yakin, jangan langsung membangun besar. Buat versi kecil yang menguji inti masalah.

Contoh:

- Sebelum membuat aplikasi marketplace, coba hubungkan penjual dan pembeli secara manual.
- Sebelum membuat software HR lengkap, tawarkan template dan konsultasi sederhana.
- Sebelum membuat kursus besar, jalankan satu kelas live kecil.
- Sebelum membuka cabang fisik, uji permintaan dengan pre-order.

Tetapkan batas kerugian

Sebelum memulai eksperimen, tentukan batas:

- Berapa uang maksimal yang boleh hilang?
- Berapa waktu maksimal yang akan digunakan?
- Apa tanda bahwa eksperimen harus dihentikan?
- Apa tanda bahwa eksperimen layak diperbesar?

Ini melatih founder untuk berani tanpa sembrono.

Lakukan refleksi mingguan

Setiap minggu, jawab lima pertanyaan:

1. Apa asumsi terbesar yang kami uji minggu ini?
2. Apa bukti yang kami dapat?
3. Apa yang mengejutkan?
4. Apa keputusan yang berubah karena bukti itu?
5. Apa eksperimen paling penting minggu depan?

Jika tidak ada keputusan yang berubah, mungkin tim hanya sibuk, bukan belajar.

Kesalahan Cara Berpikir yang Sering Membunuh Startup Awal

Beberapa startup tidak gagal karena kurang kerja keras. Mereka gagal karena cara berpikir yang keliru sejak awal.

Mengira ide bagus otomatis menjadi bisnis bagus

Ide bagus belum tentu menghasilkan bisnis bagus. Bisnis membutuhkan pelanggan, distribusi, harga, biaya yang masuk akal, dan kemampuan menjalankan operasi.

Contoh: ide aplikasi donasi makanan terdengar baik. Tetapi bisnisnya perlu menjawab siapa yang membayar, bagaimana logistiknya, bagaimana menjaga keamanan makanan, dan bagaimana menyeimbangkan pasokan serta permintaan.

Menunggu produk sempurna sebelum menjual

Produk yang terlalu lama disembunyikan bisa menjadi jebakan. Founder merasa aman karena belum ditolak pasar. Tetapi penolakan awal sering justru mempercepat pembelajaran.

Produk awal tidak harus sempurna, tetapi harus cukup jelas untuk menguji nilai utama.

Mencari investor sebelum mencari bukti

Pendanaan dapat membantu startup tumbuh, tetapi uang investor bukan pengganti validasi pasar. Jika model bisnis belum jelas, uang besar bisa memperbesar kesalahan.

Pertanyaan investor yang baik biasanya kembali ke dasar:

- Siapa pelanggan?
- Apa masalahnya?
- Apa bukti mereka membutuhkan solusi?
- Bagaimana Anda mendapatkan pelanggan?
- Berapa biaya mendapatkan pelanggan?
- Bagaimana bisnis ini tumbuh?

Menyamakan teknologi dengan nilai

Teknologi hanyalah alat. Nilai terjadi ketika pelanggan mengalami hasil yang lebih baik.

Contoh: memakai AI dalam produk tidak otomatis membuat produk bernilai. Jika AI membantu customer support menjawab lebih cepat, itu nilai. Jika AI hanya menjadi fitur yang tidak dipakai, itu dekorasi.

Tidak mau memilih fokus

Founder pemula sering ingin melayani semua orang. Padahal tahap awal membutuhkan fokus. Jika target pelanggan terlalu luas, pesan menjadi kabur, produk menjadi terlalu umum, dan eksperimen sulit dibaca.

Lebih baik menang di pasar kecil yang jelas daripada terlihat menarik bagi semua orang tetapi tidak penting bagi siapa pun.

Inti Cara Berpikir Founder

Pada akhirnya, cara berpikir founder dapat diringkas menjadi beberapa prinsip.

Pertama, startup adalah proses pencarian, bukan sekadar proses eksekusi. Anda belum hanya membangun perusahaan; Anda sedang mencari model bisnis yang benar.

Kedua, ketidakpastian tidak bisa diselesaikan hanya dengan opini dan rencana. Ia perlu diuji melalui eksperimen.

Ketiga, pelanggan adalah sumber realitas. Ide boleh datang dari founder, tetapi validasi datang dari pasar.

Keempat, belajar cepat lebih penting daripada terlihat sibuk. Kemajuan bukan jumlah fitur, rapat, atau desain, melainkan berkurangnya ketidakpastian penting.

Kelima, ketahanan berarti tetap belajar di bawah tekanan. Bukan keras kepala mempertahankan solusi, tetapi konsisten mencari cara menciptakan nilai.

Keenam, ukuran taruhan harus mengikuti kekuatan bukti. Saat bukti masih lemah, eksperimenlah secara kecil. Saat bukti menguat, barulah naikkan komitmen.

Jika bab ini dipahami dengan baik, bab-bab berikutnya akan terasa lebih masuk akal. Kita akan mulai dari masalah, pelanggan, riset, validasi, MVP, penjualan, dan seterusnya. Semua itu bukan tahapan administratif. Semuanya adalah cara untuk menjawab pertanyaan inti:

> “Apakah kita sedang membangun sesuatu yang benar-benar dibutuhkan, dapat dijual, dapat dijalankan, dan dapat tumbuh?”

References

Blank, S., & Dorf, B. (2012). *The Startup Owner’s Manual: The Step-by-Step Guide for Building a Great Company*. K&S Ranch.

Duckworth, A. L., Peterson, C., Matthews, M. D., & Kelly, D. R. (2007). *Grit*:

Document information

Bab 1: Cara Berpikir Founder

Project	Membangun Startup dari Nol
Document	Document 1.5
Author	ningsumarti
Verifier	Not verified
Downloaded	July 07, 2026 07:14 KST
Status	Working
Document link	https://theorytrace.com/projects/membangun-startup-dari-nol/documents/bab-1-cara-b-erpikir-founder/