

# Rintisan

STARTUP • TALENTA DIGITAL • IDE • INOVASI

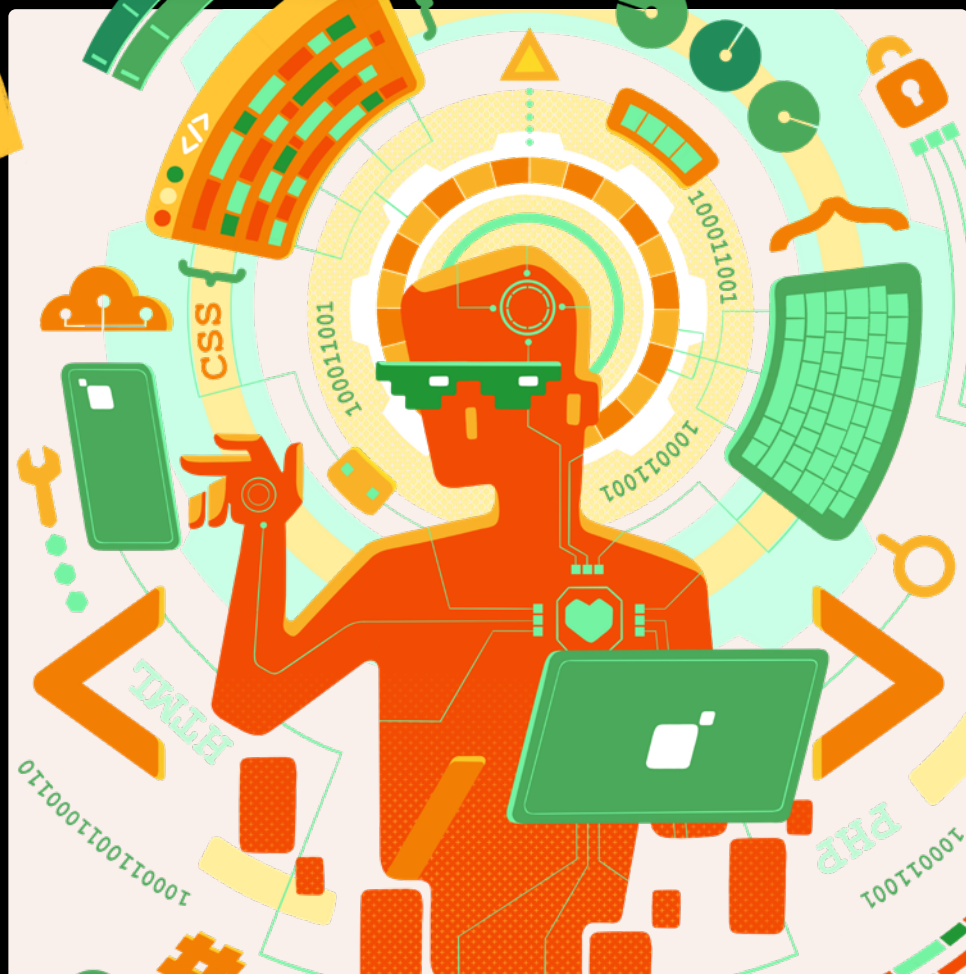
# Volume 18

# Hacker

# Seluk Beluk Menjadi Chief Technology Officer di Startup

## Prinsip Manajemen Produk

## Apa Saja Kemampuan Penting Bagi *Data Scientist*?



GERAKAN NASIONAL  
**1000**  
SEKTOR DIGITAL





# Susunan Redaksi

PENGARAH

**Semuel Abrijani Pangerapan**

PEMBINA

**Bonifasius Wahyu Pudjianto**

PENANGGUNG JAWAB & PEMIMPIN REDAKSI

**Sonny Sudaryana**

EDITOR

**Fadhila Hasna Athaya**  
**Maria Sattwika Duhita**  
**Putranto Adhi Nugroho**

PENULIS

**Aulia Mahiranissa**  
**Mayasti Dwidya Nastiti**  
**Sofy Nito Amalia**

DESAIN & LAYOUT

**Bagus Septa Pratama**  
**Rizka Irjayanti**

ILUSTRASI COVER

**Ngrh Yudha**

ILUSTRASI ARTIKEL

**Gerardus Aloysius**

PRODUKSI & SIRKULASI

**Anka Raharja**  
**Fahmi Riskian**

GERAKAN NASIONAL  
**1000**  
STARTUP DIGITAL



BUKU INI TIDAK UNTUK DIPERJUALBELIKAN

# Hacker: Membuat Solusi jadi *Scalable* dengan Teknologi

Menjalankan sebuah *startup* itu tidak hanya melulu tentang solusi yang ditawarkan, namun juga tentang bagaimana mengemas solusi tersebut.

Tanpa produk, tidak ada *startup*. Pada tahap awal untuk membangun sebuah *startup*, *Hacker* pasti akan menerima tekanan paling dahsyat. Mereka ditantang menciptakan MVP atau produk purwarupa, untuk segera dapat diluncurkan ke pengguna dengan cepat menjadi sangat penting untuk kesuksesan perusahaan.

Tidak mudah untuk bisa menjadi *Hacker*. Selain harus bisa menguasai bahasa pemrograman, mereka juga harus dapat memiliki standar yang tinggi dalam pengembangan produk, mengetahui teknologi mana yang cocok untuk diimplementasikan dalam produk, hingga memahami bagaimana membangun tim dari sekumpulan *programmer* yang memiliki cara dan etos kerja yang berbeda-beda.

Salah satu yang tersulit, *Hacker* harus dapat mengelola ekspektasi dua *co-founder* lainnya, *Hustler* dan *Hipster*. Dengan banyaknya aplikasi yang ada di pasaran, tentu tidak sulit untuk mencari referensi aplikasi yang keren untuk jadi inspirasi. Yang mungkin dua *co-founder* lainnya belum pahami, di balik sebuah aplikasi yang nyaman digunakan, ada banyak kerja keras di baliknya.

Perlu untuk diketahui setiap *co-founder*, termasuk *Hacker*, sebuah produk tidak akan pernah sempurna. Perlu iterasi terus menerus untuk menjawab kebutuhan-kebutuhan baru yang ada. Maka dari itu, jago *ngoding* saja

tidak cukup. Ada serangkaian skillset lainnya yang harus *Hacker* miliki.

***Hacker* harus dapat terbuka dengan masukan dan mau belajar hal-hal baru yang dapat membantu bagaimana agar produk dapat dikembangkan agar menjadi lebih baik lagi.**

Agar kamu lebih mengenal peran dari seorang *Hacker*, kami sudah siapkan topik esensial bagi teman-teman *Hacker* yang berada pada tingkatan *startup* di *early-stage startup*, mulai dari apa saja tanggung jawab CTO, apa saja prinsip penting dalam manajemen produk, bagaimana membangun MVP, membuat *product roadmap*, hingga mengenal SCRUM sebagai salah satu metode manajemen proyek.

Kami harap, Buku Saku Rintisan edisi *Hacker* kali ini dapat menambah pemahaman tentang apa hal-hal dasar yang sebaiknya dikuasai para *Hacker*, kemudian bagi kamu para *Hustler* atau *Hipster*, juga bisa lebih mengenal hal-hal apa saja yang menjadi tanggung jawab seorang *Hacker*.

Tim Redaksi

# Memastikan Teknologi Jadi Sumber Solusi Negeri

Berapa banyak dari kita yang saat ini memanfaatkan rapat secara daring? Memanfaatkan transfer uang tanpa harus ke bank, melakukan transaksi tanpa harus mengeluarkan uang kertas?

Atau mungkin memanfaatkan aplikasi yang memberikan notifikasi kesehatan badan kita dan masih banyak yang lainnya lagi. Semua ini bisa terjadi berkat teknologi digital yang terkoneksi.

Dunia saat ini sedang menikmati teknologi digital dan konektivitas yang sangat luar biasa. Entah ada berapa banyak masalah yang sudah terselesaikan dengan bantuan teknologi ini bukan? Belum berhenti bahkan makin bermunculan solusi-solusi lainnya, mulai dari yang memperbaharui teknologi lama, juga menciptakan inovasi baru.

Temuan-temuan solusi inilah yang sedang dibutuhkan dan dinanti-nantikan negara kita. Karena masih sangat banyak masalah di negeri ini yang dapat diselesaikan dengan bantuan teknologi digital.

**Di sinilah peran kalian sebagai talenta digital, programer, dan *developer*, harus mulai aktif dan peka dengan perubahan dan peluang ini.**

Kita semua harus sama-sama bekerja keras memastikan agar teknologi ini menjadi sumber solusi atas segala masalah yang ada di negeri kita. Ide, gagasan, dan terobosan harus menjadi bekal dalam setiap pemikiran kita. Kolaborasi juga sangat dibutuhkan dalam membantu proses pencaharian solusi-solusi tersebut.

Talenta digital pun harus juga sadar akan pentingnya menjaga nilai-nilai dan

etika masyarakat. Karena kita tahu bahwa teknologi digital dan konektivitas ini sanggup membuka seluruh akses ke pelbagai penjuru dunia.

Dengan teknologi digital, siapapun bisa bikin solusi. Yang diperlukan adalah kemampuan untuk menjadi peka dengan situasi, untuk tahu bagaimana kita bisa menyelesaikan masalah lama dengan cara baru, dengan inovasi teknologi.

Teknologi dengan kehidupan sehari-hari kini sudah tidak dapat dipisahkan lagi. Pada hakikatnya, teknologi memang ditujukan untuk membantu kegiatan kita agar lebih mudah sehingga kita dapat fokus pada hal yang lebih esensial. Namun, dengan besarnya potensi yang dihadirkan oleh teknologi, bukan berarti tidak ada dampak buruk yang menghantui. Adopsi internet yang pesat sejak pandemi, memaksa berbagai kalangan untuk segera menggunakan teknologi. Sayangnya, tidak semua memiliki kecakapan yang memadai.

Pemerintah selalu berupaya untuk menghadirkan ruang digital yang aman bagi semua. Itu diwujudkan dengan adanya regulasi terkait internet dan juga rangkaian program pemberdayaan bagi masyarakat. Namun, tentu pemerintah tidak dapat bergerak sendiri. Ada banyak pihak yang berkontribusi dalam membangun ruang digital kita, termasuk Anda. Saya harap bersama kita dapat wujudkan Indonesia yang semakin digital, semakin maju.

Pada hakikatnya, teknologi memang ditujukan untuk membantu kegiatan kita agar lebih mudah sehingga kita dapat fokus pada hal yang lebih esensial.



**Semuel A. Pangerapan**

Direktur Jenderal Aplikasi Informatika  
Kementerian Komunikasi dan Informatika RI

# Daftar Isi

## SURAT DARI REDAKSI

<i>Hacker: Membuat Solusi jadi Scalable dengan Teknologi</i>	2
--------------------------------------------------------------	---

## PRAKATA DIRJEN APTIKA

Memastikan Teknologi Jadi Sumber Solusi Negeri	4
------------------------------------------------	---

## IN-DEPTH

Prinsip Manajemen Produk	8
--------------------------	---

## IN-DEPTH

Mengenal <i>Scrum</i> : Metode Manajemen Proyek Favorit <i>Startup Founder</i>	14
--------------------------------------------------------------------------------	----

## REKOMENDASI

Buku, Film, & <i>Podcast</i>	22
------------------------------	----

## IN-DEPTH

Apa Saja Kemampuan Penting Bagi <i>Data Scientist</i> ?	26
---------------------------------------------------------	----

## IN-DEPTH

Seluk Beluk Menjadi <i>Chief Technology Officer</i> di <i>Startup</i>	34
-----------------------------------------------------------------------	----

IN-DEPTH

Hal yang <i>Founder Startup</i> Wajib Pahami Sebelum Membangun <i>Product Roadmap</i>	42
---------------------------------------------------------------------------------------	----

IN-DEPTH

<i>Full Stack Developer</i> atau <i>Specialized Developer</i> ?	48
-----------------------------------------------------------------	----

IN-DEPTH

Mengenal MVP Tipe <i>Concierge</i> dan <i>The Wizard of Oz</i>	52
----------------------------------------------------------------	----

PROFIL STARTUP

LingoTalk: <i>Startup</i> untuk Belajar Bahasa	58
------------------------------------------------	----

CAKAP STARTUP

Empati, Visi, dan Konsistensi: Andreas Sanjaya	66
------------------------------------------------	----

SEKOLAH BETA

Kelas Daring Serial <i>Hacker: Product Engineer</i>	68
-----------------------------------------------------	----

DIREKTORI

<i>Startup Hacker</i>	72
-----------------------	----

# Prinsip Manajemen Produk



Salah satu hal yang sangat menarik dari manajemen produk adalah tidak ada informasi yang bisa jadi praktik terbaik terkait pengembangan produk. Manajemen pengembangan produk adalah salah satu hal yang selalu diperbarui secara terus-menerus.

Jadi, daripada kita mengejar pemahaman tentang praktik terbaik dalam manajemen produk yang selalu mengalami perubahan, akan lebih bijaksana jika kita memahami prinsip-prinsip dari manajemen produk. Prinsip ini dirasa punya sifat lebih

stabil dan tidak terpengaruh oleh waktu, sehingga akan lebih relevan jika diaplikasikan pada timmu atau organisasi yang lebih besar. Seperti apa prinsip manajemen produknya? Kita bahas satu-persatu, yuk!

1



## Mulai dengan ‘Mengapa’

Mulai dengan ‘mengapa’, atau *start with why*, merupakan judul TED Talk dari Simon Sinek yang sangat populer. Jadi, sebelum kita menuju ke bagaimana cara menyelesaikan sesuatu, kita harus memulai pertanyaan dengan ‘mengapa’. Urutannya seperti ini:

MENGAPA

→  
APA

→  
BAGAIMANA

- Mengapa kita harus membuat produk ini?
- Apa tujuan paling besar dari pekerjaan ini?
- Apa yang mau kita wujudkan?
- Apa visi kita dengan menghadirkan produk ini?
- Bagaimana produk ini dapat meningkatkan kehidupan pelanggan kita?
- Bagaimana mereka dapat menggunakan produk ini?

## Memahami Masalahnya



Untuk apa membuat banyak fitur tapi ternyata tidak dapat menyelesaikan masalah pengguna?

Untuk itu, prinsip manajemen produk selanjutnya adalah kita, sebagai pembuat produk, benar-benar memahami secara

dalam tentang masalah yang sedang kita pecahkan. Secara teori, sepertinya terlihat mudah, ya. Tetapi pada praktiknya, ada banyak orang yang terlalu mengejar ide fitur tanpa mau meluangkan waktu untuk mendalami masalah.

- Siapa pelanggan kita saat ini?
- Siapa saja yang dapat menjadi calon pelanggan kita?
- Apa saja masalah terbesar mereka?
- Apakah mereka peduli terhadap masalah mereka?  
Atau justru cuek?
- Apakah calon pelanggan yang punya masalah ini mau mencari solusi dan membeli solusi tersebut agar masalahnya selesai?

Beberapa daftar pertanyaan di atas bisa menjadi sebuah panduan rute di mana kamu bisa memahami masalah pelangganmu secara lebih dalam. Selanjutnya, yang penting untuk diterapkan, tentu saja dengan menggunakan empati. Pembuat produk harus bisa menempatkan diri pada posisi pelanggan. Bagaimana rasanya menjadi pelanggan, ya? Apa saja yang mereka lihat? Apa saja yang mereka butuhkan dan inginkan?

Itu artinya, kamu mencoba melihat dari sudut pandang mereka.

Tidak hanya memahami masalah dari sudut pandang pelanggan saja, selanjutnya kamu bisa menanyakan perspektif masalah dari tim produk. Kombinasi perspektif dari pelanggan serta anggota tim akan menjadi perpaduan yang bermanfaat untuk pembuatan atau pengembangan produk.

## 3



## Memberdayakan Tim

Produk yang hebat dibuat oleh tim yang hebat pula. Seluruh anggota tim bekerja sama untuk dapat memecahkan masalah pelanggan melalui kolaborasi lintas divisi. Tim ini berbeda dari anggota tim seperti tim fitur atau tim pengiriman, karena tim ini dibentuk khusus dengan tujuan memecahkan masalah.

Tim yang fokus untuk memecahkan masalah ini harus memiliki nilai bahwa pelanggan adalah prioritas (*customer centricity*), kecepatan memberikan umpan balik, dekat dengan pelanggan, serta rasa empati yang tinggi.

## 4



## Mengetahui Ada Risiko Ketidakpastian

Membuat produk sama halnya dengan menanggung risiko yang besar. Bisa saja ternyata *prototype* gagal, bisa saja ternyata masalah yang didapat bukan akar masalah pelanggan, atau mungkin bisa saja ternyata fitur tidak tepat dan tidak perlu dibuat. Semuanya penuh dengan ketidakpastian. Untuk itulah, para pembuat produk harus bisa merangkul ketidakpastian. Itu artinya,

harus ikhlas jika ternyata teori dan praktiknya gagal dan tidak memberikan nilai apa pun. Dan tentu saja rasanya akan berat. Untuk itulah mawas diri terhadap risiko kegagalan bisa dijadikan pondasi mental ketika akan membuat produk. Oleh karena itu, sangat penting untuk melakukan validasi pada setiap ide yang muncul agar bisa meminimalkan risiko ketidakpastian.



## Menyeimbangkan Masukan, Hasil, dan Pembelajaran

Menghitung dan fokus pada hasil atau keluaran sudah terasa kuno dalam manajemen produk. Tidak ada lagi istilah, berapa banyak fitur yang sudah kita keluarkan?

Alih-alih hanya terpaku pada hasil saja, kita sebaiknya dapat menyeimbangkan masukan, hasil, serta proses pembelajaran. Maksudnya adalah kita perlu mempertimbangkan

masukan, atau kualitas informasi dan data yang kita gunakan untuk membuat keputusan produk. Sedangkan proses pembelajaran adalah bagaimana cara kita memasukkan informasi dan data dari hasil yang sudah dilakukan, kemudian kembali ke dalam proses pengembangan produk.

**Prinsip manajemen produk ini bukanlah sebuah patokan yang pasti harus dilakukan. Namun, dengan prinsip ini, kamu bisa lebih punya gambaran serta membantumu dalam menyelesaikan praktik-praktik pembuatan produk. Bagaimana pun, akan ada banyak cara untuk bisa menerapkan prinsip manajemen produk dan disesuaikan dengan nilai startupmu sendiri, budaya yang ada, proses yang dijalani, serta produk yang sedang kamu buat.**

# SATU APLIKASI UNTUK RAIH INSPIRASI

Akses rangkaian tahapan #1000StartupDigital  
sambil berjejaring dengan calon *founder*  
di seluruh Indonesia dari ponselmu!

UNDUH APLIKASI "1000 STARTUP DIGITAL":

[1000startupdigital.id/aplikasi](https://1000startupdigital.id/aplikasi)





IN DEPTH

# Mengenal *Scrum*: Metode Manajemen Proyek Favorit *Startup Founder*

Buat kamu yang baru memulai *startup*, pasti akan kewalahan awalnya karena ada begitu banyak hal yang harus dikerjakan dalam membangun produk. Belum lagi, kamu harus melakukan review berkala untuk memastikan hal yang dikerjakan memang sesuai dengan apa yang direncanakan. Nah, kamu harus kenalan sama yang namanya *Scrum*. Selain digunakan dalam *startup*, mulai banyak organisasi lain yang menerapkan ini, seperti korporasi, lembaga pemerintahan, dan yayasan *nonprofit*.

## ASAL USUL SCRUM



Metode *Scrum* ini lahir dari perpaduan dari pendekatan *agile* dan rugby.

Pendekatan *agile* adalah sebuah inovasi baru dari pola standar pengembangan proyek IT bernama *waterfall* yang mengharuskan kita selalu terikat pada perencanaan awal. Dengan pendekatan *agile*, proses manajemen proyek bisa lebih fleksibel dan bisa lebih mudah melakukan perubahan serta beradaptasi

pada kebutuhan yang mungkin muncul saat pengerjaan proyek. *Scrum* sebagai metode manajemen proyek membagi pengerjaan proyek dalam rangkaian *sprint* singkat untuk dapat merespons lebih cepat pada perubahan yang diperlukan. Sehingga, dapat dibilang *Scrum* adalah pengembangan pendekatan *agile* lebih lanjut.

Penamaan metode *Scrum* terinspirasi dari olahraga rugby. *Scrum* merupakan sebuah istilah dalam rugby yang menandakan permainan harus dimulai kembali setelah terjadi pelanggaran. Sebelum dimulai kembali, para pemain berkumpul erat dengan kepala menunduk dan berusaha menguasai bola. Ide awal *Scrum* berasal dari artikel Harvard Business Review 1986 “*The New New Product Development Game*”, yang ditulis oleh Hirotaka Takeuchi dan Ikujiro Nonaka, dan dikembangkan lebih lanjut oleh Jeff Sutherland dan Ken Schwaber.

# PRINSIP-PRINSIP UTAMA SCRUM

## Fleksibilitas dan kemampuan beradaptasi.

Metode *Scrum* dirancang untuk memudahkan tim dapat menjawab kebutuhan yang terus berkembang secara cepat dan efektif. Produk akan dites setelah satu *sprint* selesai, dan *backlog* (daftar pekerjaan yang belum dikerjakan) akan selalu diperbarui, dan tim berdiskusi tentang produk setiap hari.



## Kerja tim yang konsisten.

Salah satu aspek dasar dalam metode ini adalah komunikasi tim. Sebelum memulai hari, diadakan rapat singkat terkait produk yang dikembangkan. Dalam memulai *sprint* baru, tim dan *product owner* mendiskusikan rencana iterasi yang akan diterapkan. Di akhir *sprint*, akan ada presentasi untuk mengambil kesimpulan dari hasil yang didapat.





## Dikerjakan dalam bentuk rangkaian *sprint*.

*Sprint* adalah kondisi di mana tim menetapkan target yang harus dicapai dalam waktu iterasi singkat (1-3 minggu). Dalam kurun waktu tersebut, sudah ada tugas-tugas yang ditetapkan. Di akhir *sprint*, klien akan dipresentasikan produk yang telah dihasilkan.

## Objektif utama adalah hasil untuk klien.

Klien harus mendapatkan produk sesuai dengan *goal* dan spesifikasi yang disepakati, apapun itu keadaannya.



## Tim dengan keahlian beragam.

Satu tim *Scrum* harus terdiri atas orang dengan beragam kemampuan yang saling melengkapi untuk membangun sebuah produk.

# TIGA PERAN PENTING DALAM SCRUM



## Product Owner

*Product owner* adalah “jembatan” antara produk dan pengguna. Mereka lah yang memahami apa yang dibutuhkan pengguna. Karena itu, *product owner* yang mengarahkan tim: mengelola prioritas sebuah tugas, menentukan seberapa penting tugas tersebut, dan menguji versi produk baru setelah sebuah *sprint* selesai dilaksanakan.

*Product owner* bertanggung jawab membuat daftar *backlog* produk, yaitu catatan berisi semua informasi penting tentang produk dan proses kerja yang telah dilakukan untuk membangun produk.

Biasanya, peran ini dijalankan oleh perwakilan klien, dan jika tim mengembangkan produk internal untuk *startup*/organisasi, maka itu adalah seseorang dari pimpinan manajemen. Dalam *startup* dengan tim kecil, fungsi klien dilakukan oleh CEO (tetapi peran klien dan *scrum-master* harus dipisahkan).





## Scrum-Master

*Scrum-Master* mengoordinasikan proses kerja tim dan penggunaan alat *Scrum*. Sebelum *sprint* pertama dimulai, *scrum-master* melatih *product owner* dan seluruh tim pengembangan untuk bekerja sesuai prinsip-prinsip *Scrum*. Setelah dimulai, *Scrum-master* akan mengontrol seluruh proses.

*Scrum-Master* memoderasi semua pertemuan pembahasan produk, memastikan bahwa tim selalu memiliki sumber daya yang diperlukan, menangkap ide dan saran, serta bertindak sebagai mediator jika terjadi perselisihan dan diskusi dalam tim. Pada saat yang sama, penting bagi *Scrum-Master* untuk mengetahui batasan dalam proses moderasi agar tidak berubah menjadi “pengawas” atau manajer biasa.



## Tim

Tim adalah gabungan dari beberapa orang yang bergerak menuju tujuan *sprint* dan bertanggung jawab untuk pengembangan produk. Disarankan sebuah tim *scrum* itu beranggotakan tim yang kecil. Kata Jeff Bezos, indikator tim yang kecil itu dimana ada 2 loyang pizza cukup dimakan untuk bersama. Kriteria penting kedua untuk sebuah tim adalah adanya beragam kemampuan, mencakup semua spesialis yang diperlukan untuk mengembangkan produk.

# BAGAIMANA MEMULAI SCRUM?



Ada beberapa langkah yang perlu dilakukan untuk menjalankan metode *Scrum*.

## 1 | BENTUK TIM

Pastikan tim terdiri atas orang-orang dengan berbeda kemampuan untuk dapat membangun produk.

## 2 | PILIH SCRUM-MASTER

Pastikan orang ini memiliki pemahaman yang baik tentang metode *Scrum* dan memiliki kemampuan untuk memoderasi.

## 3 | PILIH PRODUCT OWNER

Perlu digarisbawahi bahwa sebelum mengimplementasikan metode *Scrum*, setiap pihak yang terlibat harus memahami bagaimana *Scrum* bekerja, tidak terkecuali *product owner*. Keterlibatan *product owner* dalam proses *sprint* dan komunikasi secara berkala dengan tim oleh *product owner* sangatlah penting.

## 4 | SUSUN PRODUCT BACKLOG

*Product owner* bertanggung jawab untuk menyusun *product backlog*. Para *developer* dalam tim harus memahami apa yang tertulis dalam *product backlog* untuk memastikan apa spesifikasi yang dibutuhkan dan informasi yang disajikan sudah cukup. Setelah itu, daftar pekerjaan dalam *product backlog* akan dijalankan dalam rangkaian *sprint*.

## 5 | RENCANAKAN SPRINT

Durasi dari sebuah *sprint* dapat berbeda-beda sesuai dengan kebutuhan, namun umumnya tidak kurang dari 7 hari dan tidak lebih dari 30 hari. Setiap *sprint* dimulai dengan diskusi, di mana *sprint backlog* ditentukan dan tugas dibagikan pada masing-masing anggota tim. Penting untuk diingat bahwa sebaiknya tidak menambahkan tugas baru dalam ruang lingkup pekerjaan saat *sprint* telah berjalan.

# KELEBIHAN DAN KEKURANGAN SCRUM



## KELEBIHAN

- ☑ Fleksibilitas dan kemampuan beradaptasi yang cepat terhadap metrik atau kebutuhan baru.
- ☑ Kemungkinan untuk mengubah spesifikasi produk, bahkan melakukan pivot.
- ☑ Dirancang untuk memenuhi tenggat waktu.
- ☑ Penggunaan sumber daya tim secara efektif.
- ☑ Membangun kekompakan klien dan tim.
- ☑ Pertukaran informasi yang konstan.
- ☑ Reaksi cepat terhadap perubahan apa pun.



## KEKURANGAN (YANG DAPAT DIANTISIPASI)

- ☒ Tim harus diajari cara menggunakan metode tersebut.
- ☒ Sulit untuk menemukan spesialis yang tepat untuk tim.
- ☒ Sangat sulit untuk menemukan *Scrum-master* yang berpengalaman.
- ☒ Ada risiko mengubah pertemuan harian menjadi pertemuan panjang yang tidak masuk akal (tanpa moderasi yang tepat).
- ☒ Satu tim tidak akan dapat mengerjakan beberapa produk sekaligus.

Yang pasti, keuntungan utama yang diberikan *Scrum* adalah kemampuan untuk bekerja dengan cepat dan efisien pada suatu produk di kondisi yang sangat tidak pasti, seperti untuk menghadapi pesaing ataupun tantangan baru. Itulah mengapa metode *Scrum* adalah alat yang hebat untuk industri *startup*, di mana versi pertama suatu produk biasanya dibangun melalui banyak *pivot* dan mungkin terlihat sangat berbeda pada rencana awal pengembangannya.

Jadi, siap untuk menjalankan *Scrum*?

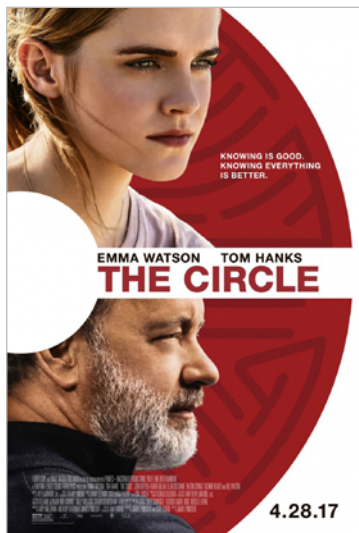
# Buku, Film, & *Podcast*

Telah dikurasi, tinggal  
diresapi, dinikmati, dan dibagi  
pada teman-teman lainnya!



Pada edisi kali ini, kami telah mengumpulkan karya-karya yang bisa kamu jadikan sebagai inspirasi dalam menciptakan ide-ide kreatif yang solutif dan *out of the box*. Mari disimak!

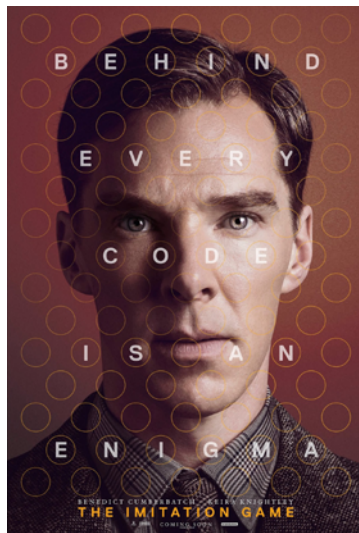
# Film



## THE CIRCLE

2017

Menceritakan tentang Mae Holland, seorang karyawan di perusahaan teknologi besar layaknya Google dan Facebook yang bernama The Circle. Suatu hari, pemilik perusahaan menugaskan Mae untuk menciptakan inovasi baru yang dapat menghapus kebebasan dan privasi manusia. Dalam film ini ditunjukkan bagaimana teknologi punya kuasa besar dalam kehidupan, begitu pula internet yang dapat menjadi pisau bermata dua.

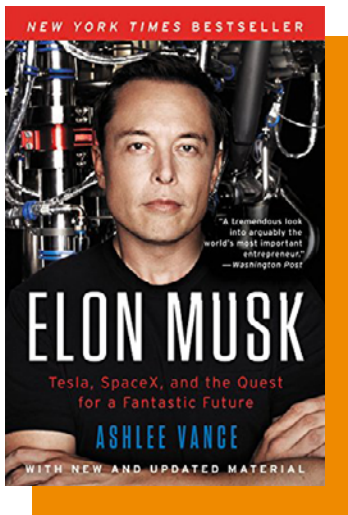


## THE IMITATION GAME

2014

Tertarik dengan *machine learning*? Jika iya, kamu wajib nonton film yang satu ini nih! Berkisah tentang Alan Turing, seorang ahli matematika dan komputer yang berhasil menciptakan mesin untuk memecahkan German Enigma Code. Keberhasilan tersebut sekaligus menyelamatkan nyawanya selama Perang Dunia II.

# Buku



## ELON MUSK: TESLA, SPACEX, AND THE QUEST FOR A FANTASTIC FUTURE ASHLEE VANCE

Ashlee Vance menggunakan gaya tulisan yang menarik dan terperinci untuk mengisahkan perjuangan orang terkaya di dunia, Elon Musk. Buku ini diawali dengan kisah Musk yang memimpikan mobil listrik dan pesawat luar angkasa. Dengan motivasi yang kuat, Musk memberanikan diri ke Amerika Serikat untuk melanjutkan studi dan impiannya. Namun, perjalanan Musk tidak selalu mulus, karena dia sempat didepak dari jabatan CEO. Jatuh bangun kariernya tersebut membuat Musk makin bertekad untuk merealisasikan mimpinya saat kecil, yakni dengan mendirikan SpaceX dan menjadi CEO Tesla.



## DEEP TECH: DEMYSTIFYING THE BREAKTHROUGH TECHNOLOGIES THAT WILL REVOLUTIONIZE EVERYTHING ERIC REDMOND

Deep Tech meluncurkan tujuh revolusi global secara bersamaan, di antaranya AI, *augmented and virtual reality*, *blockchain*, *cryptocurrency*, dan *Internet of Things*. Revolusi inilah yang digadang akan mendorong ekonomi global. Eric Redmond menjelaskan dalam bukunya tentang risiko tersembunyi dari tiap revolusi teknologi. Di dalamnya juga terdapat cara agar kamu fasih berteknologi dan cerdas dalam mengambil peluang besar.

# Podcast



## DEVELOPER TEA

Dipandu oleh Jonatan Cutrell, *Developer Tea* merupakan saluran podcast dengan episode pendek sekitar 15 sampai 20 menit. Didalamnya membahas berbagai topik seputar pemrograman dan pengembangan aplikasi yang bertujuan untuk membantu para *developer* lebih unggul dalam karier mereka dan berdampak positif pada dunia teknologi.

[HTTPS://DEVELOPERTEA.COM](https://developtea.com)



## CODENEWBIE

Podcast CodeNewbie ditujukan untuk kamu yang baru terjun di ranah teknologi dan pemrograman. Dalam tiap episodenya yang berdurasi 30 sampai 55 menit, terdapat panduan informatif yang mencakup semua dasar-dasar bidang teknologi dan spektrumnya yang luas, seperti *web development* dan *data science*.

[HTTPS://WWW.CODENEWBIE.ORG/  
PODCAST](https://www.codenewbie.org/podcast)

# Apa Saja Kemampuan Penting Bagi *Data Scientist*?

Data telah berkembang menjadi aset penting di organisasi, seiring dengan transformasi digital di berbagai lini dan terbukti mampu menambah nilai di berbagai aspek operasional. Untuk menganalisis data dan mengubahnya menjadi suatu hal yang bernilai, dibutuhkan keahlian khusus yang berbeda dari kemampuan menyimpan dan mengelola data. Karenanya, kemampuan *data scientist* dibutuhkan dalam hal ini.



Umumnya, *data scientist* berfokus mengungkap informasi dan wawasan dalam jumlah besar dari data terstruktur maupun data tidak terstruktur. *Data scientist* membantu organisasi untuk mendapatkan jawaban atas pertanyaan bisnis, melihat tren, dan membuat prediksi yang tepat.

Ada 14 kemampuan yang perlu kamu asah untuk menjadi *data scientist*, baik itu *hard skill* maupun *soft skill*. Apa saja?

## Hard Skill

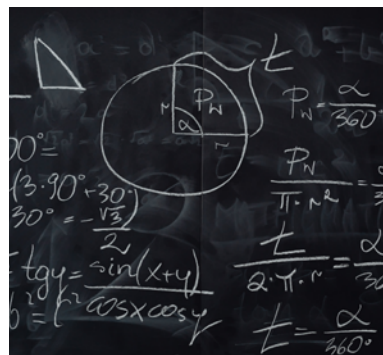
### (Keterampilan Teknis)

Agar dapat mengolah data jadi wawasan yang bernilai, seorang *data scientist* harus memiliki kemampuan untuk melontarkan pertanyaan yang tepat. Untuk mencapai itu, ada beberapa keterampilan teknis yang biasanya dibutuhkan:



## 1. STATISTIK

Karena *data scientist* secara berkala menerapkan konsep dan teknik statistik, *data scientist* harus memiliki pemahaman statistik yang baik. Familiar dengan analisis statistik, kurva distribusi, probabilitas, standar deviasi, varians, dan elemen statistik lainnya membantu *data scientist* mengumpulkan, mengatur, menganalisis, menafsirkan, dan menyajikan data.



## 2. KALKULUS MULTIVARIABEL DAN ALJABAR LINIER

Bagi *data scientist*, memiliki kemampuan menerapkan konsep matematika untuk memahami dan mengoptimalkan fungsi yang cocok dengan model pada data set menjadi sangat penting. Jika kemampuan ini tidak dikuasai, akan sulit untuk membuat model yang dapat menghasilkan prediksi yang akurat. Memiliki keterampilan kalkulus dan aljabar juga merupakan keharusan jika ingin menguasai *machine learning*.

## 4. PREDICTIVE MODELING

Mampu menggunakan data untuk membuat prediksi dan memodelkan berbagai skenario dan hasil adalah bagian utama di bidang *Data science*. Analitik prediktif mencari pola dalam kumpulan data yang ada atau baru untuk memperkirakan peristiwa, perilaku, dan hasil di masa mendatang. Hal tersebut dapat diterapkan pada berbagai kasus penggunaan di berbagai industri, seperti analisis pelanggan, pemeliharaan peralatan, dan diagnosis medis. Dengan banyaknya potensi penggunaan dan manfaatnya, *predictive modeling* dinilai sebagai keterampilan yang sangat berharga bagi para *data scientist*.

## 5. MACHINE LEARNING DAN DEEP LEARNING

Walau sebenarnya seorang *data scientist* tidak perlu bekerja dengan teknologi AI, semakin banyak perusahaan mencari *data scientist* untuk mengimplementasikan aplikasi *machine learning*. Untuk melakukannya, perusahaan membutuhkan seseorang yang dapat melatih algoritma *machine learning* untuk mempelajari kumpulan data dan mencari pola, anomali, atau wawasan yang dapat digunakan untuk membangun model analitik. Akibatnya, kebutuhan untuk mempekerjakan *data scientist* yang terampil dalam *machine learning* semakin meningkat. Keterampilan dalam *deep learning*, yaitu metode yang lebih maju yang menggunakan *neural network* untuk membuat model analitik yang kompleks, juga dapat membantu seorang *data scientist* lebih menonjol dibanding *data scientist* lainnya.





## 6. DATA WRANGLING DAN DATA PREPARATION

Data scientist sering mengatakan bahwa lebih dari 80 persen waktu yang mereka habiskan untuk pekerjaan *data science* adalah menyiapkan data untuk analisis. Untuk itu, jika *data scientist* dapat melakukan *data profiling* dasar, pembersihan data, dan pemodelan data dasar, akan besar manfaatnya bagi mereka. Hal tersebut akan memungkinkan mereka untuk menangani masalah kualitas data dan ketidaksempurnaan dalam kumpulan data, seperti isian yang hilang atau salah diberi label, hingga masalah pemformatan. Keterampilan mengolah data juga termasuk melakukan pengumpulan data dari berbagai sumber dan memilah format data yang berbeda, serta memfilter, mengubah, dan menambah data untuk aplikasi analitik. Agar mencapai hal tersebut, *data scientist* harus terbiasa menggunakan *data warehouse* umum dan *data lake environment*, termasuk *database* relasional dan NoSQL, serta *platform* data besar seperti Apache Spark dan Hadoop.

## 7. MODEL DEPLOYMENT DAN PRODUCTION

Data scientist harus dapat memilih algoritma yang tepat dan kemudian menggunakan data pelatihan untuk pendekatan pembelajaran terawasi atau menjalankan algoritma untuk secara otomatis menemukan kluster atau pola dalam pembelajaran yang tidak terawasi. Setelah model menghasilkan hasil yang diinginkan, *data scientist* harus menerapkannya dalam *production environment* untuk membantu organisasi mereka membuat keputusan bisnis praktis secara berkelanjutan.

## 8. VISUALISASI DATA

Saat bekerja dengan kumpulan data besar yang besar dan berisi tipe data yang berbeda, mampu memvisualisasikan data secara efektif saat menyajikan hasil analitik adalah keterampilan yang penting bagi seorang *data scientist*. Mereka harus memiliki kemampuan untuk menceritakan data agar dapat menyoroti dan menjelaskan wawasan yang mereka hasilkan, dan visualisasi data adalah fondasi dasar bagi mereka mengomunikasikan wawasan tersebut kepada pimpinan dan pemangku kepentingan lainnya. Karenanya, seorang *data scientist* sebaiknya memahami penggunaan Tableau, D3.js atau berbagai alat visualisasi data lainnya.

# Soft Skill

## (Keterampilan Interpersonal dan Non Teknis)

Selain keterampilan teknis, *soft skill* juga penting bagi *data scientist* karena mereka harus dapat menerjemahkan temuan mereka dan menjelaskannya pada rekan-rekan lain yang membutuhkan wawasan dari temuan tersebut. Berikut adalah beberapa *soft skill* yang penting untuk dikuasai:



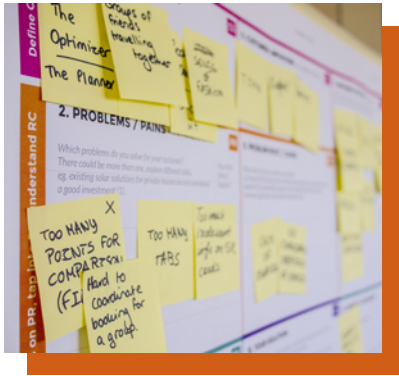
### 9. CRITICAL THINKING (BERPIKIR KRITIS)

*Data scientist* harus mampu menilai kumpulan data dan hasil analitik untuk membentuk penilaian tentang validitas dan relevansinya. Melihat data dengan pandangan skeptis membantu *data scientist* mencapai kesimpulan yang akurat dan tidak bias.



### 10. BUSINESS KNOWLEDGE

Dalam berbagai organisasi, tim *data scientist* merupakan bagian yang melengkapi bidang lain, tidak hanya terikat pada bidang IT saja. Karena itu, mereka perlu memiliki pengetahuan tentang bisnis dan industri perusahaan tempat mereka bekerja. Hal ini akan membantu mereka untuk dapat menghasilkan pertanyaan terkait analisis data yang lebih tepat, sehingga membantu perusahaan menemukan cara baru dalam menggunakan data mereka untuk memprioritaskan permasalahan yang penting untuk dipecahkan.



## 11. PROBLEM SOLVING (PENYELESAIAN MASALAH)

Sering kali, *data scientist* diminta menemukan sekeping informasi dari tumpukan data. Untuk melakukan hal tersebut, mereka perlu datang dengan hipotesis terkait permasalahan yang ingin dipecahkan dan mencoba memvalidasi permasalahan tersebut menggunakan data yang dianalisis. Saat mengerjakan proses pengolahan data, mereka perlu berpikir tajam dalam memecahkan masalah dan mencari tahu bagaimana berbagai temuan mereka masuk ke dalam perhitungan dan menentukan data apa yang harus dimasukkan atau ditinggalkan. Wuih, sepertinya berat, ya?



## 12. CURIOSITY (RASA INGIN TAHU)

Rasa ingin tahu dan memiliki keinginan untuk terus belajar adalah keterampilan yang harus dimiliki seorang *data scientist*. Pikiran yang selalu ingin tahu mampu menyaring sejumlah besar data untuk menemukan jawaban dan wawasan. Data adalah hal yang dapat terus berubah, sehingga *data scientist* tidak boleh berpuas diri tentang bagaimana cara mereka mengolah data atau membatasi diri pada kesimpulan yang saat ini mereka peroleh.



### 13. KOLABORASI

Dapat bekerja dalam bagian tim yang lebih besar juga penting. *Data scientist* seringkali perlu berkolaborasi di antara satu sama lain, hingga lintas divisi bahkan pimpinan dalam perusahaan.



### 14. KOMUNIKASI

*Data scientist* yang bekerja dengan data setiap hari pasti akan memahami data dan temuannya lebih baik daripada siapa pun. Untuk itu, mereka harus dapat mengomunikasikan pemahaman tentang data dan menjelaskan hasil analitik sehingga tim dan pimpinan dapat menggunakan informasi tersebut untuk membuat keputusan yang baik.

Wah, banyak ya hal yang harus dipelajari dan dikuasai oleh seorang *data scientist*! Untuk bisa jadi seorang *data scientist* yang jago, kamu tidak bisa hanya bergantung pada ilmu yang kamu pelajari di sekolah atau kampus atau *bootcamp* koding saja. Kamu harus dapat belajar dari berbagai sumber untuk dapat memahami *data science* karena teknologi selalu berkembang setiap harinya. Semoga artikel ini dapat membantumu dalam menentukan hal apa yang harus kamu kuasai berikutnya ya.

"Satu-satunya cara mempelajari bahasa pemrograman baru adalah dengan mencoba menuliskannya."

- Dennis Ritchie, *Computer Scientist*

# Seluk Beluk Menjadi *Chief Technology Officer* di *Startup*



Teknologi kini menjadi kunci kesuksesan banyak perusahaan, apalagi sejak pandemi terjadi di 2020. Dengan penerapan teknologi, perusahaan dapat melakukan pemetaan sumber daya, menganalisis kebutuhan pasar, memprediksi pendapatan, dan banyak hal lainnya dengan lebih baik. Sejak saat itu, peran *Chief Technology Officer* (CTO) menjadi peran fundamental dalam perusahaan di berbagai lini industri.

Pada umumnya, peran CTO adalah memastikan bagaimana perusahaan memiliki keunggulan kompetitif dari penerapan teknologi dan mampu mengembangkannya ke skala lebih besar. Namun, sebenarnya, peran CTO bervariasi tergantung dari lingkup usaha dan skala perusahaan. Dalam artikel ini, kita akan menggali bagaimana evolusi peran dan tanggung jawab CTO seiring perkembangan *startup*.

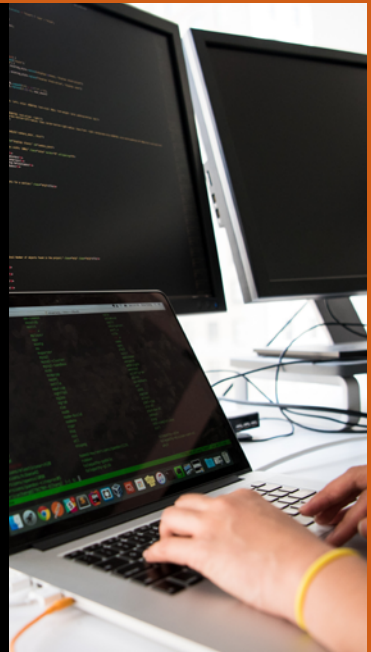
Berikut adalah deskripsi peran dan tanggung jawab CTO di *startup* pada umumnya:

## 1. PEMILIHAN PLATFORM DAN DESAIN TEKNIS

CTO bertanggung jawab mempertahankan fokus strategis proyek dan mengubah konsep menjadi kenyataan.

Pada *startup* yang masih baru dirintis, *founder* atau *co-founder* dapat menjalankan fungsi CTO. CTO juga dapat disebut sebagai *product manager* dengan tujuan utama mengelola tim teknologi dan membuat keputusan bisnis terkait teknologi yang akan digunakan, seperti perencanaan desain teknologi, tata letak arsitektur produk, dan pemilihan *platform* pengembangan.

Pada perusahaan besar, CTO dapat mengelola beberapa tim pengembangan produk, bahkan termasuk tim yang melayani kebutuhan teknologi perusahaan.

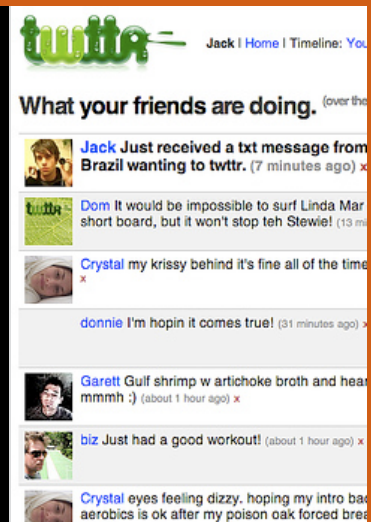


## 2. PENGEMBANGAN MVP

*Minimum Viable Product (MVP)* adalah produk dengan fitur esensial yang dibuat untuk menguji hipotesis kebutuhan pengguna dan menggali feedback mereka.

Ada perbedaan besar antara peran dan tanggung jawab CTO dalam *startup* baru dan perusahaan besar terkait pengembangan MVP.

Pada perusahaan besar, CTO mengurus manajerial tim *developer*, tetapi tidak mengurus masalah hingga level engineering harian. Sementara, pada *startup* baru, CTO harus menangani tugas-tugas teknologi mengenai iterasi produk.





### 3. MEREKRUT TIM DAN MENGELOLA PERTUMBUHAN TIM

CTO *startup* bertanggung jawab untuk merekrut anggota tim sesuai yang dibutuhkan. Salah satu yang menyulitkan bagi CTO untuk mencari tim adalah kebanyakan *developer* memiliki preferensi untuk fokus pada area yang terbatas, sementara *startup* membutuhkan *developer* yang serba bisa.



### 4. CYBERSECURITY

Celah dari aspek keamanan dapat ditemukan di *database* perusahaan, situs *web*, atau alat digital lain yang digunakan tim. Produk yang sedang dikembangkan oleh perusahaan mungkin juga memerlukan algoritma keamanan untuk menjaga data pengguna tetap pribadi dan terenkripsi. Selain mengurus tim dan mengembangkan produk, menjaga keamanan dari sistem perusahaan juga menjadi tanggung jawab CTO.

## 5. QUALITY ASSURANCE (QA) DAN PRODUCT TESTING

*Startup* jarang memiliki divisi QA yang terpisah. Ini berarti bahwa pengujian produk harus dibagi antara *developer* dan anggota tim lainnya. Tanggung jawab CTO adalah memutuskan siapa yang bertanggung jawab atas proses *debug*. Jadi, ketika *bug* ditemukan, info tersebut biasanya diteruskan ke CTO. Kemudian, CTO perlu memutuskan bagaimana menyelesaikan masalah tersebut. CTO juga diharapkan dapat memberikan instruksi untuk perbaikan *bug*.

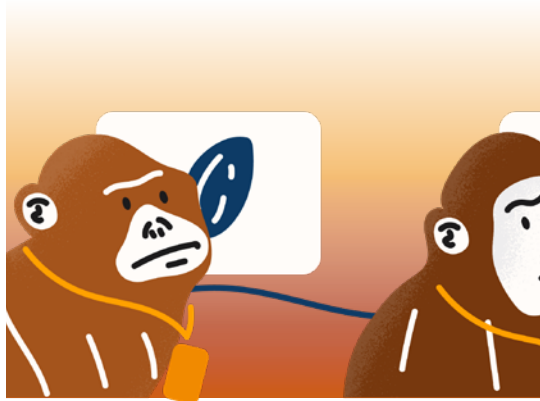


## 6. INOVASI DAN ROADMAP PRODUK

Pada tahap awal, CTO membangun *roadmap* pengembangan produk dengan mempertimbangkan tujuan dan strategi perusahaan. Dalam merumuskannya, CTO mempertimbangkan berbagai aspek pengembangan produk digital, mulai dari penggunaan teknologi, UX, keamanan, anggaran yang dibutuhkan, proses operasional, hingga koordinasi tim.



# Evolusi Peran dan Tanggung Jawab CTO



## 1. *Pre-seed*

### Tidak ada *tech specialist* kecuali CTO.

Dalam tahap ini, kita berbicara tentang *startup* yang mencoba meluncurkan produk pertamanya. Di sini, CTO adalah *engineer* “serba bisa” yang harus siap membangun arsitektur produk digital tanpa bantuan dan tetap menjadi spesialis utama untuk implementasinya. CTO bertanggung jawab atas keamanan data, manajemen QA, dan

masalah penerapan. CTO juga harus meninjau kompleksitas teknis dari ide produk dan menyarankan pendekatan teknologi untuk mewujudkannya.

## 2. *Startup*

### 5 - 10 *tech specialist* di divisi teknologi.

Selain mengerjakan tugas pada tahap *pre-seed*, di tahap ini, CTO harus dapat merekrut tim baru serta melakukan proses *onboarding*, pelatihan awal, dan mengelola kinerja tim tersebut. Kemudian, CTO harus membuat pembagian kerja yang sesuai dan memilih siapa orang yang tepat dalam menjalankan tugas tersebut.

Setelah MVP dibuat, CTO berfokus untuk membuat produk lebih baik. CTO akan

merencanakan fitur baru, mengembangkan tim untuk memenuhi kebutuhan dari hal teknis produk yang berkembang, dan mengembangkan infrastruktur produk agar optimal.

Jadi, peran CTO berubah menjadi gabungan antara visioner teknologi, spesialis SDM, dan manajer tim, bukan menjadi *developer* yang lebih ahli.



### 3. *Growth*

#### 50 - 100 *tech specialist* di divisi teknologi.

CTO bertugas mengawasi beberapa divisi tim teknologi, meninjau kemajuan, merancang alur kerja, dan menangani masalah keamanan teknologi di perusahaan. Di perusahaan menengah (belum terlalu besar), CTO juga kadang bertugas melakukan tugas terkait manajemen produk.

Menentukan visi dari penerapan

teknologi yang dibutuhkan dalam pengembangan produk menjadi tugas utama CTO. Perkembangan industri dan teknologi terkait juga harus menjadi hal dalam perhatian CTO untuk memastikan teknologi yang diterapkan adalah teknologi terbaik bagi produk *startup* tersebut.

### 4. *Expansion*

#### Lebih dari 100 *tech specialist* di divisi teknologi.

CTO harus dapat berpikir strategis dan membuat keputusan berdasarkan berbagai pertimbangan dari tim yang ia pimpin. Di tahapan ini, CTO tidak lagi mengerjakan pekerjaan teknis, tetapi lebih pada mengelola tim dan melakukan koordinasi bersama berbagai pihak.





# Kemampuan Dasar yang Harus Dimiliki CTO

## 1. Negosiasi

CTO memimpin divisi pengembangan teknologi, memilih kandidat baru untuk bergabung dalam tim, mendorong mereka untuk mempelajari teknologi baru dan mencoba cara baru. Selain itu, CTO juga harus berkoordinasi dengan divisi lainnya dalam perusahaan bahkan klien atau pengguna. Untuk itu, kemampuan negosiasi wajib dimiliki.

## 2. Kreativitas dalam Pemecahan Masalah

CTO menjadi sosok yang dapat menjadi panutan tim dalam berpikir kreatif saat melihat permasalahan dan mencari solusi. Hal tersebut penting karena dapat menginspirasi tim yang setiap harinya berkuat dengan proyek agar berani mencoba hal baru dalam menyelesaikan masalah dengan lebih efektif dan efisien.

## 3. Kepemimpinan dan Menjadi Mentor

Mampu memimpin dengan baik dan menginspirasi untuk tim terus berkembang adalah karakter sosok CTO yang diidam-idamkan banyak *developer* di *startup*. Sehingga, cakupan kerja CTO tidak hanya mengembangkan produk, tetapi juga mengembangkan kapasitas tim yang bekerja sama dengannya.

## 4. Kemauan Terus Belajar

Karena teknologi akan terus berkembang, CTO juga harus selalu menambah pengetahuan yang ia miliki. Sehingga, ia paham bagaimana cara terbaik untuk merealisasikan visi pengembangan produk yang ia punya menggunakan teknologi yang terbaru.

## 5. Kemampuan Teknologi Tingkat Lanjut

Untuk dapat memimpin tim yang terdiri dari sekumpulan orang dengan kemampuan teknologi yang berbeda-beda, CTO harus dapat memahami bagaimana teknologi yang diterapkan dalam startupnya bekerja. Hal itu akan membantu CTO dapat mengorkestrasikan tim dengan lebih baik.

## 6. Berpikir Strategis

Setiap keputusan CTO terkait produk akan dapat memengaruhi divisi-divisi lainnya. Dengan demikian, CTO harus memahami bagaimana gambaran besar dari dampak dari setiap keputusan yang akan diambil.

---

Setelah melihat betapa beragamnya peran dan tanggung jawab CTO, kamu mungkin melihat betapa CTO harus dapat menjalankan eksekusi teknis dan mengelola operasional tim teknologi.

Akhir kata, jika kamu ingin tahu dengan kriteria apa kamu harus memilih CTO untuk startupmu, coba evaluasi kebutuhan teknologi saat ini dan masa depan bisnis startupmu terlebih dahulu. Jika kamu berencana melamar posisi CTO di perusahaan mana pun, siapkan CV dengan fokus pada keahlian teknis dan keterampilan manajemen proses operasional.

# Hal yang *Founder* *Startup* Wajib Pahami Sebelum Membangun *Product Roadmap*



Ternyata, mengacu pada Laporan ProductPlan 2019, 32% dari *product manager* menyatakan bahwa tugas terkompleks mereka adalah membuat rencana kerja realistik yang mendorong pencapaian dari *goal strategist*.

Menentukan objektif jangka pendek dan jangka panjang, serta menurunkannya jadi rangkaian inisiatif adalah hal yang fundamental bagi proses pengembangan dan aplikasi. Kedua hal tersebut sama pentingnya seperti merumuskan USP (*unique selling proposition*) dan analisis pasar dalam upaya mencapai kesuksesan dari produk.

Sayangnya, pemahaman tentang *product roadmap* terkadang dipandang sesederhana menyiapkan presentasi yang menggambarkan evolusi dari produk saja. Untuk itu, artikel ini akan membahas tentang dasar-dasar dari *product roadmap* dan kepentingannya bagi audiens yang berbeda-beda.

*Product roadmap* adalah gambaran secara visual prioritas jangka pendek dan jangka panjang dari sebuah produk yang membantu kamu untuk membangun konsensus dan mengomunikasikannya pada pihak strategis. *Product roadmap* merupakan langkah yang sistematis dan logis untuk menerjemahkan visi dan strategi dari produk menjadi *project* dan tugas yang dapat dikerjakan oleh timmu secara harian.

*Product roadmap* bukanlah hal yang berdiri sendiri. Ada beberapa komponen yang kamu harus pastikan startupmu telah miliki sebelum mulai merancang *product roadmap*. Berikut adalah hal-hal yang membentuk strategi startupmu:

---

MISI                   »    Apa hal yang ingin kamu capai?

---

VISI                    »    Seperti apa perubahan yang kamu buat saat misimu tercapai?

---

STRATEGI       »    Bagaimana langkahmu untuk mewujudkan visimu?

---

GOAL                »    Bagaimana kamu menilai kemajuan dalam strategi yang kamu jalankan?

---

ROADMAP       »    Apa hal yang kamu perlu bangun untuk mencapai *goal*?

---

TASK                »    Apa hal yang bisa kamu lakukan hari ini untuk mencapai progres yang diharapkan?

---

# 4 Langkah Dasar Membangun *Product Roadmap*

Setelah memahami definisinya, sekarang kita coba buat *product roadmap*, yuk!

## 1. Tentukan “*Why*” dari produkmu — mengapa produkmu ada.

Ini adalah hal pertama yang harus kamu definisikan. “*Why*” di balik produkmu adalah pegangan yang kamu harus dapat pegang teguh baik di awal pengembangan produk baru atau pembaruan produk yang sudah ada. Jika kamu tidak dapat

menjawab pertanyaan tersebut dengan usaha pengembangan produk yang kamu lakukan ataupun data pencapaian startupmu, jangan-jangan sebetulnya kamu malah keluar jalur dan menjauh dari visi yang ingin kamu capai?

- 1. FEEDBACK PENGGUNA** » *User interview* adalah satu cara terbaik untuk menggali masalah pengguna. Namun, jika kamu tidak dapat berbicara langsung dengan pengguna atau menginginkan lebih banyak sumber data, carilah umpan balik dari penjualan, pemasaran, *customer support*, dan divisi lain yang berhubungan dengan pengguna.
- 2. PRODUCT BACKLOG** » Kamu mungkin telah menuliskan daftar tugas dan masalah yang harus dipecahkan. Saat kamu memetakan masalah, pastikan untuk merapikan dan memperbarui *user stories* di *backlog*-mu. Untuk masing-masing item dalam *backlog*-mu, cek lagi apakah mereka memberikan nilai pada pengguna dan masih cocok dengan strategi produkmu secara keseluruhan.
- 3. DATA PENGGUNAAN** » Pengguna tidak selalu memberi tahu kamu apa yang mereka inginkan. Untuk itu, coba kamu lihat bagaimana mereka benar-benar menggunakan produkmu. Dari situ, kamu dapat menemukan masalah dan hambatan yang mungkin kamu pecahkan.
- 4. ANALISIS PASAR & KOMPETITOR** » Setiap *startup* tentu ingin produknya menjadi leader, bukan follower. Untuk itu, kamu perlu senantiasa memperhatikan bagaimana preferensi pasar dan apa solusi yang ditawarkan oleh kompetitormu.

Jangan terburu-buru dan luangkan waktu untuk mendalami *problem discovery*. Menemukan masalah yang tepat untuk dipecahkan serta mencapai hasil yang diinginkan adalah fondasi dari *roadmap* yang baik.

## 2. Tentukan siapa audiens dari *Product Roadmap*-mu dan sesuaikan untuk dapat relevan dengan mereka.

*Product roadmap* adalah dokumen strategis. Untuk itu, tujuan strategis kamu dalam menceritakannya dengan *stakeholder* kamu akan sangat berbeda dengan tujuanmu dalam menunjukkannya kepada tim pemasaran atau pengembangan produk. Jangan gunakan dokumen *roadmap* yang sama untuk setiap audiens yang kamu temui karena mereka memiliki kepentingan informasi yang berbeda-beda.

Pemangku kepentingan (*stakeholder*) seperti investor atau *C-level* umumnya tidak ingin melihat detail tingkat dasar dari setiap fitur. Mereka lebih ingin melihat

bagaimana hal itu akan meningkatkan *market share* dan pengaruh pada angka pendapatan *startup*.

Jika kamu mengadakan rapat dengan tim pengembangan produk, tentu mereka ingin tahu detail dari setiap hal dalam *product roadmap* sehingga presentasi kamu perlu fokus pada bagaimana rencana tersebut akan memengaruhi pekerjaan dari tiap tim, seperti teknologi apa yang akan kamu minta mereka untuk gunakan, seperti apa jangka waktu apa yang mereka miliki untuk menyelesaikan pekerjaan, serta apa tugas dan prioritas mereka sehari-hari.

Karena *roadmap* dirancang untuk audiens yang berbeda yang memiliki kepentingan yang berbeda, ada baiknya untuk mendiversifikasi penyampaian *product roadmap*-mu.

### TIPE TIPE PRODUCT ROADMAP

	TIMELINE-BASED	WITHOUT DATE	KANBAN
GOAL	Mengomunikasikan rencana <i>high-level</i> .	Mengomunikasikan strategi produk.	Memberikan gambaran nyata tentang progres pengembangan produk saat ini.
AUDIENS	<i>C-level</i> atau <i>stakeholder</i> lainnya.	Divisi <i>marketing &amp; sales</i> .	Divisi teknologi.
BATASAN	Tanggal mulai dan selesai.	Berorientasi pada masa depan daripada progres saat ini.	Untuk dipresentasikan tim internal.

### 3. Prioritaskan tema strategis produkmu, lalu mulailah membangun *roadmap* yang sebenarnya.



Mulai dari level strategis terlebih dahulu, kemudian lengkapi dengan detail mengapa tema tersebut penting untuk masuk dalam *roadmap*, apa kaitannya dengan tema lain dan mengapa. Sebagai contoh, visi kamu adalah menciptakan peluang yang sama untuk setiap orang dapat berwirausaha, kamu dapat menurunkannya dalam tema seperti: akses ke wawasan dan jejaring bagi komunitas wirausaha.

Nah, untuk membentuk sebuah *product roadmap*, berikut adalah struktur yang umum digunakan.

## STRUKTUR PRODUCT ROADMAP

### GOAL (ATAU TEMA)

*Goal* harus memberikan jawaban untuk pertanyaan “Mengapa produk harus dikembangkan?”. Misalnya, kamu sedang membangun *platform crowdfunding* properti. *Goal* kamu dapat berupa sesuatu seperti “Membuat pengembang properti dapat mengumpulkan dana dari investor di *platform* dengan proses penggalangan dana dan investasi yang sepenuhnya otomatis”. Pastikan *goal* kamu jelas dan dapat dipahami.

### EPIC

*Goal* dapat dicapai melalui rangkaian *epic* yang menggambarkan fungsionalitas atau cerita pengguna yang diperlukan untuk diterapkan. Mari kita lanjutkan contoh yang disebutkan di poin sebelumnya: membuat *platform* untuk berinvestasi dalam proyek properti. *Epic*, dalam hal ini, bisa menjadi “Memungkinkan investor untuk mendaftar di *platform* dan mengelola portofolio investasi mereka” dan “Memungkinkan pengembang properti membuat kampanye penggalangan dana dan memantau proses pendanaan”.

### STORIES

*Stories* adalah rangkaian tugas pengembangan yang dilakukan di bawah *epic* tertentu. Misalnya, implementasi pendaftaran investor pada *platform crowdfunding* membutuhkan desain, pekerjaan *front-end* dan *back-end*. Semua item ini menjelaskan daftar fitur yang harus dan yang baik untuk dimiliki yang direncanakan dalam pengembangan produk.

### TIMELINES

*Timeline* akan membantumu menjadwalkan tanggal rilis produk dan menurunkannya jadi *deadline* pada rangkaian tugas yang harus diselesaikan untuk mewujudkannya.

#### 4. Penting untuk bersikap fleksibel karena *roadmap* pasti akan berubah.



Akhirnya, sembari kamu menuliskan detail di *roadmap*-mu, perlu diingat bahwa tidak ada detail yang permanen. Prioritas *startup* dapat berubah. Kesiapan sumber daya dapat berubah. Pergerakan oleh kompetitor juga dapat memaksa pengembangan produk atau jadwal rilis untuk berubah.

Semoga poin-poin di atas membantumu dalam membangun *product roadmap* yang lebih baik lagi ke depannya!

##### TAHUKAH KAMU

### MoSCoW Method

Untuk membantumu dalam menentukan prioritas dalam *product roadmap*, kamu bisa gunakan ini.

**M**

#### MUST HAVE

Fitur yang bersifat fundamental dan wajib dimiliki.

**S**

#### SHOULD HAVE

Fitur yang bersifat penting, tapi tanpa fitur ini pun produk tetap berfungsi dengan baik.

**C**

#### COULD HAVE

Fitur yang baik, namun hanya menambah sedikit nilai kegunaan.

**W**

#### WILL NOT HAVE

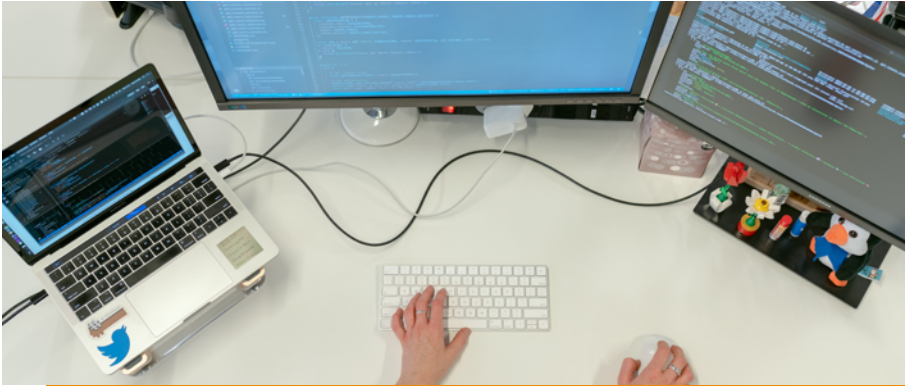
Fitur yang tidak kritis untuk dimiliki saat ini.



# Full Stack Developer atau Specialized Developer?

Seiring dengan perkembangan teknologi yang pesat, keterampilan dan standar kerja juga harus terus beradaptasi. Jika kamu adalah seorang *founder* yang tengah membentuk tim *developer*, pasti kamu mengalami dilema ini: haruskah merekrut *developer* yang bisa ini itu atau seorang spesialis

di satu bidang? Jika kamu adalah seorang *developer* yang sedang mencari kerja, kamu mungkin berada di persimpangan jalan, apakah aku harus setia mendalami satu bahasa pemrograman? Atau mencoba mengeksplor yang lain? Mau pilih *fullstack* atau *specialized*? Mari kita bahas di sini.



## APA ITU FULL-STACK DEVELOPER?



*Full-stack developer* adalah *developer* yang bisa mengerjakan *backend* dan *frontend* dari sebuah dan aplikasi. Dalam bahasa sederhananya, seorang *full-stack developer* itu harus dapat mengerjakan tampilan depan dari, mengelola *database*, dan juga berkoordinasi dengan tim lain saat melakukan perencanaan dari produk.

BERIKUT ADALAH GAMBARAN UMUM DARI KETERAMPILAN *FULL-STACK DEVELOPER*:

Frontend development (Vue.js, Angular, Node.js, React) • Backend development (PHP, Java, NET) • Desain UI/UX • Markup language (HTML and CSS) • Database (Firebase, MongoDB, PostgreSQL, MySQL) • Software testing (TestComplete, Katalon studio, Postman, Selenium, etc) • Berkoordinasi merancang *scalability*, arsitektur, dan *roadmap* produknya

Nah, yang tidak banyak orang tahu, seorang *full-stack developer* itu tidak serta merta menulis semua kode situs sendiri dari 0. Saat ini, sudah ada banyak *framework*, *tool*, dan *library* yang dapat disesuaikan dengan kebutuhan.

Inilah alasan mengapa memiliki pengetahuan yang memadai dalam mengimplementasikan *tool* dan *framework frontend* dan *backend* membuat *full-stack developer* memiliki prospek yang menarik bagi perusahaan.

```
'send_welcome' => false,  
  });  
  
  if (err) {  
    if (err instanceof Error) {  
      result = {  
        'response': 'error',  
        'message': err.message  
      };  
    } else {  
      result = {  
        'response': 'success',  
        'message': ''  
      };  
    }  
  }  
  res.json(result);  
}
```



## APA ITU SPECIALIZED DEVELOPER?

Seperti gambaran dari namanya, *specialized developer* adalah orang yang ahli dalam satu disiplin ilmu dan bekerja di sekitar bidang tersebut.

Namun, ini tidak berarti bahwa *specialized developer* tidak akan memiliki pemahaman yang sama yang diperlukan untuk membangun dan mengirimkan aplikasi seperti halnya *full-stack developer*. Ini hanya berarti bahwa *specialized developer* akan menjadi orang yang tepat untuk subjek tertentu saja.

Akhir kata, jika kamu adalah *founder* yang baru saja merintis *startup* dengan *bootstrapping*, berarti ada baiknya jika kamu mencari *full stack developer* terlebih dahulu. Kemudian, seiring dengan pertumbuhan *startup*, bentuklah tim dari *specialized developer*.

# Mana yang tepat untuk *startup* atau *project* saya?

Berikut ada beberapa poin yang bisa membantumu memilih mana yang cocok untuk kamu rekrut.



## SKALA DARI PEKERJAAN

Dalam proyek kecil, *full-stack developer* dapat menangani berbagai aspek secara bersamaan. Proyek besar dapat dikelola dengan baik dengan tim *specialized developer*, masing-masing menangani satu departemen.



## KOMPLEKSITAS DARI PEKERJAAN

Tugas yang tidak begitu kompleks cocok digarap oleh *full-stack developer*. Jika kompleksitas pengembangan lebih tinggi, berarti kamu membutuhkan persyaratan dedikasi yang lebih baik dan hal itu dapat dicapai dengan merekrut beberapa *specialized developer*.



## FLEKSIBILITAS DARI PEKERJAAN

Seorang *full stack developer* menawarkan fleksibilitas yang besar karena ia berpengalaman baik dengan sisi layanan dan sisi klien yang membuatnya mudah untuk berpindah di antara tugas-tugas. Kehadiran *full stack developer* mengurangi waktu pengembangan dengan menghilangkan kesalahpahaman antara sisi pengembangan karena bekerja sendirian.



## ANGGARAN BIAYA

Mempekerjakan satu *full-stack developer* dapat dengan mudah menggantikan dua atau terkadang lebih dari dua *specialized developer*. Oleh karena itu, mempekerjakan *full-stack developer* dapat mengurangi jumlah tenaga kerja yang dibutuhkan dan pada akhirnya menurunkan biaya perekrutan.



## KUALITAS DARI PEKERJAAN

Untuk menghasilkan kualitas yang tinggi, sebaiknya membentuk tim *specialized developer*.

# Mengenal MVP Tipe *Concierge* dan *The Wizard of Oz*

Cara terbaik untuk memvalidasi *value proposition* (manfaat yang ditawarkan perusahaan pada pengguna) adalah dengan menguji apakah solusi tersebut memberikan nilai yang dicari pengguna. Untuk menguji *value proposition*, kamu memerlukan cara untuk memberikan manfaat yang sama kepada pengguna seperti yang akan diberikan produkmu nantinya. Nah, untuk mencapai itu, kamu perlu membuat eksperimen yang mampu menyampaikan nilai produkmu itu. *Minimum Viable Product* (MVP) adalah produk minimum yang diperlukan untuk menguji hipotesis terpenting kamu. Tidak lebih, dan tidak kurang.

MVP biasanya merupakan penyederhanaan dari gambaran besar dari produk yang kamu bayangkan ketika kamu merintis startupmu. Faktanya, kamu akan merasa sedikit malu saat meluncurkan MVP. Jika kamu tidak merasa malu, berarti kamu telah membuat MVP yang lebih kompleks dari yang seharusnya.

Concierge MVP dan Wizard of Oz MVP adalah teknik cerdas yang menggantikan fungsi teknis sebuah produk yang rumit dengan menggunakan tenaga manusia. Dengan waktu pembuatan yang sebentar, kamu akan dapat menguji pertanyaan kunci startupmu: “apakah ada orang yang menginginkan produk yang sedang kamu bangun?”



## Wizard of Oz MVP

Wizard of Oz MVP membungkus sebuah solusi yang dijalankan oleh manusia dengan cangkang teknologi. Maksudnya, pengguna akan percaya bahwa ia berinteraksi dengan produk yang diotomatiskan, padahal kenyataannya manusialah yang menghasilkan layanan tersebut.

Sebagai contoh, kamu ingin membuat layanan travel yang memberikan perjalanan yang sempurna bagi pengguna. Produkmu akan dapat memfasilitasi permintaan tertulis seperti “saya ingin dibuatkan rencana perjalanan ke Bali, untuk 4 orang. Kami ingin datang ke tempat-tempat yang bersejarah sambil wisata kuliner tanpa ada makanan pedas. Tolong dicarikan juga alternatif

penginapan yang paling murah.”

Hal ini mungkin cukup sulit untuk diakomodasi oleh komputer karena mesin harus mengolah bahasa untuk dapat diolah menjadi perintah-perintah pada algoritma agar dapat menyusunnya menjadi rencana perjalanan yang baik. Nah, manusia dapat melakukan ini dengan mudah. Pada faktanya juga manusia merencanakan liburan mereka sendiri.

Untuk membuat Wizard of Oz MVP, pertama-tama kamu dapat membuat formulir isian sederhana di *web* untuk pengguna menuliskan perjalanan yang mereka inginkan dan juga email mereka. Lalu, setelah formulir dikirimkan, beri tahu

pengguna bahwa algoritma akan memproses permintaan mereka dengan menganalisis berbagai kemungkinan kemudian mereka akan mendapatkan jawaban dalam kurun waktu 6 jam. Di balik proses itu, kamu dan tim lah yang meriset semua permintaan dari pengguna tersebut.

Lalu, kamu mengemail pengguna tersebut dengan gaya seolah-olah merupakan sebuah email otomatis. Kemudian, kamu mengukur seperti apa respon yang diberikan pengguna pada layanan yang kamu berikan. Apakah mereka jadi melakukan pemesanan berdasarkan rekomendasimu? Seperti apa rating yang diberikan untuk layananmu?

Metode Wizard of Oz MVP dapat berjalan dengan baik karena sebenarnya pengguna tidak peduli apakah proses di baliknya

dijalankan oleh algoritma atau mesin.

---

**Yang dipedulikan adalah apakah masalah yang ia miliki dapat diselesaikan, dan dalam kasus yang kita bahas, adalah apakah ia mendapat rencana perjalanan sesuai yang ia impikan.**

---

Jika ternyata pengguna menyukai layananmu yang di baliknya dilakukan oleh manusia, kamu dapat lebih yakin bahwa produkmu yang nantinya terotomasi juga akan disukai oleh mereka.



## Concierge MVP

Saat kamu membayangkan produk ideal startupmu, kamu mungkin membayangkan aplikasi *web* canggih yang memberi pengguna pengalaman yang *seamless*, mampu memberi mereka rekomendasi tentang apa yang harus dilakukan, mampu membuat banyak hal jadi serba otomatis dan banyak lagi.

Nah, kamu jangan langsung membuatnya. Coba dulu sampaikan fungsi itu melalui tenaga manusia. Itulah yang disebut dengan Concierge MVP. Merupakan istilah dari bahasa Inggris, *concierge* adalah seorang karyawan hotel yang melayani bermacam-macam kebutuhan tamu.

Kelebihan dari *Concierge MVP* adalah untuk menyederhanakan produk dengan menggantikan komponen yang diotomatisasi menggunakan tenaga manusia. Setiap pengguna akan mendapatkan layanan produkmu langsung dari manusia, bukan mesin.

---

**Metode ini tidak hanya  
menghemat banyak waktu  
dalam pembuatan produk,  
tetapi juga membuatmu  
dapat bersentuhan  
langsung dengan pengguna  
awal dari produkmu.**

---

Dengan siap membantu mereka, kamu akan mulai memahami kebutuhan mereka secara mendalam dan hal itu akan membantumu menjawab pertanyaan penting: "apakah orang benar-benar menginginkan apa yang saya tawarkan?"

Mari kita bahas satu contoh. Misalnya, kamu sedang membangun aplikasi kencan yang mengukur kecocokan antar pengguna dengan memberikan serangkaian pertanyaan yang unik, seperti "apakah kamu lebih suka bubur diaduk atau tidak diaduk?" Tinder maupun Bumble tidak menanyakan pertanyaan tersebut pada penggunaanya. Hipotesis kamu adalah pertanyaan unik ini akan memungkinkan prediksi kecocokan yang lebih baik daripada layanan lain yang telah ada.

Dalam visi besarmu, kamu akan memiliki algoritma *machine learning* yang mengolah kumpulan data besar lalu menghitung kecocokan terbaik. Tetapi, tentu kamu belum memiliki semua ini sekarang.

Sebagai gantinya, kamu dapat mengganti hampir semua proses dalam aplikasimu dengan tenaga manusia. Alih-alih membuat algoritma, kamu secara manual mendaftarkan

pengguna baru dan mengirimkan kuesioner kepada mereka melalui email. Kamu meninjau semua jawaban mereka secara manual dan memasangkan pengguna secara manual berdasarkan kriteria yang kamu buat. Terlebih lagi, kamu memberi mereka layanan layaknya *concierge* yang baik, seperti kamu berulang kali menelepon mereka agar mereka berkomitmen untuk berkenan, kamu mencari tahu bagaimana preferensi orang yang mereka sukai, dan kamu membantu mencarikan orang yang lebih cocok berdasarkan masukan mereka.

Jelas yang kamu lakukan ini tidak mungkin dilakukan dalam jangka panjang dan pastinya tidak dapat melayani 100.000 pengguna sekaligus. Namun, saat ini kan kamu belum memiliki masalah ini. Masalahmu adalah menjawab apakah orang benar-benar membutuhkan solusi yang kamu tawarkan.

---

**Selain memungkinkan  
kamu untuk dapat menguji  
hipotesis produk kamu  
tanpa membuat produk  
sama sekali, *concierge*  
MVP juga membuatmu  
berinteraksi secara  
pribadi dengan pelanggan  
sehingga membantumu  
lebih memahami apa yang  
mereka hadapi.**

---

Ini akan memberi kamu dasar yang kuat untuk mengiterasi pada produk yang kamu tawarkan. Nah, mungkin dalam beberapa keadaan lain, kamu ingin penggunamu percaya bahwa mereka berinteraksi dengan mesin, bukan manusia. Tetapi di lain sisi, kamu tetap tidak ingin membangun produk yang rumit terlebih dahulu. Solusi untuk ini adalah *Wizard of Oz MVP*.

# Mana metode yang cocok untuk startupku?

## WIZARD OF OZ AKAN MEMBANTUMU DALAM MENJAWAB:

- ☑ Apakah *value proposition* saya memecahkan masalah pengguna?
- ☑ Interaksi apa yang perlu dilakukan oleh pelanggan kita dengan produk kita?
- ☑ Apakah fitur yang saya buat membantu menyelesaikan masalah pengguna?
- ☑ Apakah produk kita menciptakan nilai lebih bagi pengguna?
- ☑ Apakah produk kita harus dimiliki/ harus digunakan oleh pengguna?
- ☑ Apakah pengguna bersedia membayar untuk solusi yang saya tawarkan?



*Wizard of Oz* dan *Concierge Testing* adalah dua metode eksperimen untuk menguji solusi bagi masalah pengguna. Kedua metode tersebut melibatkan interaksi manusia yang membuatnya tampak serupa, padahal sebenarnya tidak. Perbedaannya adalah fakta bahwa dengan *Concierge Testing*, pengguna mengetahui bahwa manusialah yang memberikan solusi. Sementara dengan pembuatan prototipe *The Wizard of Oz*, pengguna menganggap teknologilah yang berfungsi dalam memberikan solusi.



## CONCIERGE TESTING AKAN MEMBANTUMU DALAM MENJAWAB:

- ☑ Apakah solusi yang ditawarkan memecahkan masalah pengguna yang sebenarnya?
- ☑ Bagaimana masalah bisa diselesaikan?
- ☑ Apa fitur dasar yang diperlukan untuk mengimplementasikan solusi?
- ☑ Apa masalah terbesar dalam memberikan solusi bagi pengguna?

Kedua metode eksperimen ini saling melengkapi. *Concierge MVP* akan membantumu mendapatkan wawasan yang lebih banyak terkait masalah pengguna dan bagaimana menyelesaikannya. Setelah kamu mendapatkan wawasan yang cukup dalam untuk merumuskan *value proposition*, barulah kamu dapat mengujinya dengan *Wizard of Oz MVP*.

Dengan membaca artikel ini, semoga kamu dapat pencerahan tentang MVP dan dapat menguji produkmu ke target pengguna dengan lebih cepat ya!



# LingoTalk: *Startup* untuk Belajar Bahasa

Kali ini Rintisan mewawancarai Andre Benito, *co-founder* dari LingoTalk. Sebuah *startup* yang menyediakan *platform* pembelajaran bahasa asing dengan personalisasi sesuai kebutuhan siswanya.

#### ELEVATOR PITCH

### *Personalized language learning solution*

#### VISION

*Democratizing access to applied knowledge through efficient and all-encompassing language learning journey.*

#### MISSION

*Powering the Indonesia's education agenda to uplift children's language learning pathway.*

*"We are committed to bridge the language skill gap with a solution that is efficient, affordable, and impactful since day-one."*

## PADA MULANYA, BAGAIMANA CERITA LATAR BELAKANG DALAM MEMBUAT LINGOTALK?

Kebetulan saya memang dari dulu senang belajar bahasa dan saat ini menguasai beberapa bahasa asing. Sewaktu masih kuliah di UGM, saya bergabung dengan komunitas yang fokusnya *language exchange*. Di sana saya bertemu dengan orang-orang yang bisa bahasa asing dan membutuhkan teman *sharing* dengan bicara bahasa asing. Misalnya ada yang menguasai bahasa Perancis tapi ingin belajar bahasa Mandarin, bisa mengobrol dengan teman yang menguasai bahasa Mandarin dan ingin belajar bahasa Perancis. Dari ikut komunitas itu, saya punya ide untuk menginspirasi orang lain untuk belajar bahasa dengan metode yang kami lakukan. Karena dengan metode ini, frekuensi penggunaan bahasa asing menjadi lebih sering dan orang akan lebih cepat menguasai bahasa asing dibandingkan belajar di sekolah atau kampus.

Di tahun 2020, saya bertemu dengan *co-founder* Lingotalk yang lain, yakni Hardi Wijaya. Dia punya ide untuk membuat *startup* di bidang pendidikan dan teknologi, akhirnya kami bawa ide ini jadi sebuah *platform* yang kita bangun di yang kami coba kembangkan sendiri bersama beberapa *inhouse*

#### FOUNDER

Andre Benito,  
Hardi Wijaya

#### DIDIRIKAN PADA

2020

#### INDUSTRI

*Education  
Technology  
(Edtech)*

#### JUMLAH TIM

20 orang

#### PARTNERS

±100 mitra,  
4-5 sekolah,  
10 korporasi

#### FASE PENDANAAN

*Seed round*

#### WEBSITE

lingotalk.co

engineer. Pada Agustus 2020, lingotalk.co beta version diluncurkan ke publik dan kami mengesahkan Lingotalk jadi sebuah PT. Yang awalnya kelas-kelas Lingotalk dilakukan offline di Yogyakarta saja dan menawarkan ke teman-teman sendiri yang mau berangkat pertukaran pelajar atau S2, karena pandemi kami fokus di pengembangan platform digital yang bisa mengakomodir 1-on-1 live tutoring.

Kami pilih industri education technology (Edtech) selain karena passion, juga karena kami lihat ada kesempatan besar yang bisa digarap. Tidak sampai 20% dari total siswa di Indonesia itu mendapatkan pendidikan yang optimal. Sebagian besar dari mereka pun tidak memiliki akses pendidikan yang berkualitas dan punya indeks pembelajaran bahasa yang rendah.

Dengan market sebesar ini saja, sebagian besar pengguna Lingotalk itu merupakan pengguna pertama platform edukasi berbasis teknologi. Jadi kami coba cari pain points yang belum diberikan oleh kompetitor kami, dan menonjolkan keunggulan “belajar bahasa

yang efisien” dimana materi pembelajaran disesuaikan dengan kebutuhan siswa sehingga lebih tepat sasaran. Harapannya dengan teknologi yang ditawarkan, para siswa akan lebih termotivasi belajar karena tahu kalau pembelajarannya sesuai dengan kebutuhan dan preferensi mereka.

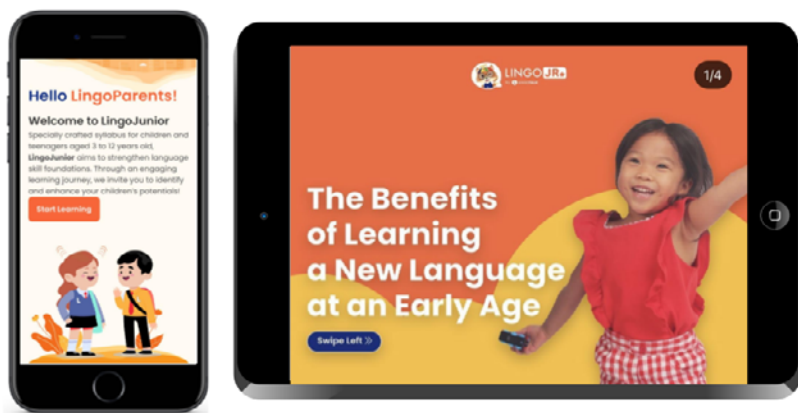


## APA SAJA, SIH PRODUK LINGOTALK?

Saat ini, Lingotalk menawarkan dua produk, yakni Bahasa Asing yang terdiri dari 10 bahasa, dan Lingo Junior. 10 Bahasa tersebut adalah bahasa Arab, Inggris, Prancis, Jerman, Indonesia, Korea, Mandarin, Spanyol, Thailand, Jepang. Namun kami fokus kepada 4 bahasa utama untuk dikembangkan menjadi lebih banyak produk, yakni bahasa Inggris, Mandarin, Korea, dan Jepang. Kursus ini punya empat tingkat kefasihan: pemula, dasar, menengah, dan menengah-atas.

Sedangkan Lingo Junior merupakan kursus *online* dengan materi yang dibuat khusus untuk anak-anak dan remaja berusia 3-12 tahun, tujuannya untuk memperkuat keterampilan bahasa dasar. Produk ini awalnya dibuat kami bertemu dengan klien yang merupakan sekolah inklusif khusus untuk anak-anak TK-SD. Klien ini secara spesifik minta untuk dibuatkan produk khusus untuk anak-anak dengan pengajar yang

berpengalaman mengajar anak-anak pula. Lalu setelah kami coba kembangkan, ternyata produk ini banyak diminati dan menyumbang *revenue* lumayan besar bagi Lingotalk. Darisitu kami melihat kesempatan bahwa sebetulnya *market leader* Indonesia dari industri *Edtech*, tepatnya di pembelajaran bahasa adalah anak-anak. Maka dari itu sejak 2021, kami berusaha mengembangkan Lingo Junior ini.



## APA TANTANGAN TERBESAR YANG DIHADAPI SAAT TAHUN PERTAMA MERINTIS LINGOTALK?

Pastinya sumber daya jadi tantangan terbesar, ya. Tahun pertama berdiri, kami beroperasi dengan *bootstrapping*. Saya dan Hardi termasuk *founders startup* yang masih muda dan belum punya banyak pengalaman kerja maupun koneksi. Jadi kami berusaha untuk terus persisten menjalaninya karena yakin kalau hal yang dikerjakan ini pasti bisa menjadi besar dan banyak peminatnya.

---

Dalam prosesnya, kami coba cari terus *pain points* pendidikan dan metode pembelajaran bahasa di Indonesia itu seperti apa.

---

Kompetitor Lingotalk pastinya pihak-pihak pengajar yang sudah lama berada di industri pendidikan. Jadi selain cari tahu cara bisa diterima di industri ini, kamu juga berusaha mencari inovasinya.

## APA KEUNIKAN UTAMA YANG DITAWARKAN LINGOTALK BAGI PENGGUNANYA?

Karena dari awal yang mau dibangun adalah metodenya, kami tidak fokus menawarkan gurunya siapa atau kualitas materinya sebagus apa. Kalau pengguna mau cari materi bagus, semua bertebaran aksesnya di internet dan bisa didapat dengan mudah secara *free*. Tapi ketika mempunyai metode yang bisa membantu penyampaian suatu pembelajaran jadi lebih efisien, itu adalah ide yang bisa Lingotalk eksekusi dengan baik. Dengan begitu orang-orang bisa punya kualitas pembelajaran yang lebih bagus sekaligus mudah diakses.

Metode yang dimaksud dimulai dari Lingo method atau *pre-assessment*. Siswa akan dinilai kemampuannya terlebih dahulu sebelum mendaftar kelasnya. Sistem penilaiannya semacam kuesioner dan VARK (*visual, auditory, reading/writing, kinesthetic*) *assessment* untuk mengukur gaya belajar siswa. Penilaian ini bisa melihat objektif dan kecepatan belajar para siswa. Dengan gaya belajar tertentu, level kemampuan tertentu, alokasi pembelajaran

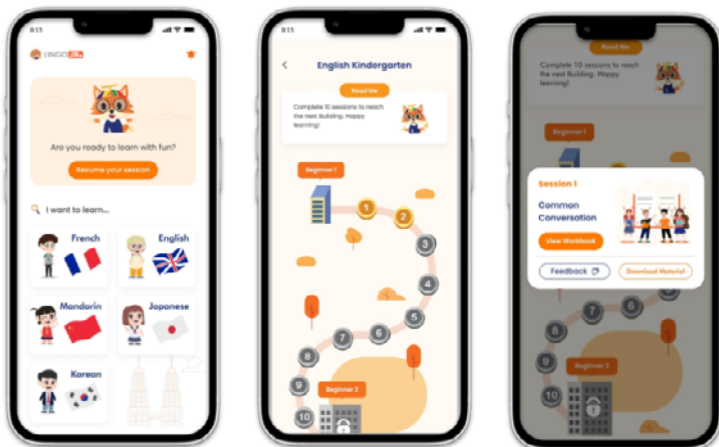
seperti apa, kemudian akan Lingotalk kombinasikan dengan *machine learning* sehingga siswa secara otomatis mendapatkan materi yang telah direkomendasikan oleh sistem. Kedepannya, kami akan fokus dan mengembangkan keakuratan rekomendasi dari *assessment* tersebut.

Maka dari itu, kursus yang diberikan juga tidak terlalu banyak pertemuannya karena kami ingin mengurangi waktu belajarnya. Dengan catatan, materi disesuaikan dengan tujuan siswa mempelajari bahasa tersebut. Misalnya seorang siswa punya target nilai 8 untuk IELTS, jika di tempat les konvensional ia harus belajar selama setahun, di Lingotalk ia bisa menyelesaikan dalam waktu sembilan bulan saja karena materi-materi berulang dan telah dikuasai sudah dieliminasi. Jika siswa tersebut butuh perbaikan di *listening* dan *reading*, maka tutor akan fokus ke bagian itu.

---

Dengan *learning pathway* yang lebih singkat, siswa bisa mendapatkan tujuan yang sama dengan orang lain.

---



## APA HAL YANG PALING BERPENGARUH BESAR PADA PERTUMBUHAN LINGOTALK?



Faktornya adalah hal yang *intangible* banget menurut saya, yaitu berani mimpi *gede* aja sebenarnya. Kayaknya itu, sih, yang bikin kita tetap lanjut ngembangin Lingotalk. Karena kalau ngikutin kata orang tua atau teman-teman yang meragukan karena kami masih muda dan pengalamannya sedikit, ya Lingotalk tidak akan jadi seperti sekarang.

Ada satu sifat yang disebut *Grit*, yakni sebuah *passion* dan ketekunan dalam menjalani sesuatu dalam waktu lama. Kami bisa menunjukkan kekuatan sifat itu, sehingga menjadi sebuah faktor besar dalam membangun Lingotalk. Saat bisa menunjukkan kita 100% *passionate* melakukan hal ini, pihak luar akan percaya sama kita tidak hanya untuk investasi tapi juga untuk kerja bareng.

## BIASANYA FOUNDER STARTUP PUNYA MENTOR, BAGAIMANA DENGAN KALIAN?

Kami *gak* punya mentor yang spesifik, karena awalnya juga kami *gak* meniatkan untuk bikin *startup*. Kami cuma ingin membuat *platform* belajar bahasa saja. Tapi di satu waktu kami mikir, tuh, gimana ya cara *scale-up* bisnisnya?

Lalu kami lihat banyak program akselerator *startup*, dan kami ikut program *Startup Studio* di tahun 2021. Di sana kami jadi banyak kenal dan bisa berjejaring ikut sesi *1-on-1* dengan para mentor yang perusahaannya sudah mendapat pendanaan *Series B* ke atas. Jadi meskipun tidak ada mentor spesifik, kami selalu coba mengumpulkan perspektif dari orang-orang yang sudah lama di dunia *startup*.

## APA PENCAPAIAN TERBESAR YANG PERNAH DIRAIH LINGOTALK?

Kalau dikuantifikasi tentunya mendapatkan pendanaan *pre-seed*. Kalau secara mitra dan pengguna, kami senang sekali bisa memperoleh mitra besar yang sudah lama ada di bidang pendidikan, seperti BPK Penabur misalnya. Karena mereka mempercayakan

Lingotalk untuk kerjasama bagi murid-muridnya yang ingin belajar bahasa asing, dan mau menerima inovasi yang kami tawarkan.

## APA PRIORITAS UTAMA DARI LINGOTALK DALAM 2 TAHUN MENDATANG?

Dalam dua tahun, rencananya Lingotalk akan mulai mengembangkan infrastruktur Lingo Junior sekaligus fokus ke B2B.

Karena model bisnis B2B ini salah satu strategi untuk mengumpulkan banyak calon *customer* dalam waktu yang relatif cepat. Salah satu caranya adalah bermitra dengan sekolah-sekolah. Karena sudah mendapat kepercayaan dari sekolah, akan lebih mudah berkomunikasi orang tua murid yang ingin anaknya diberikan kursus bahasa secara privat.

Jika dua hal itu sudah bisa dilakukan secara paralel selama setahun, baru di tahun berikutnya kami fokus pengembangan model bisnis B2C. Karena infrastrukturnya sudah terbentuk sebelumnya, maka harapannya kami bisa lebih mudah melakukan *scale-up* proses B2C ini.

**Startup  
Studio<sup>ID</sup>**

Lingotalk adalah salah satu alumni dari program Startup Studio Indonesia yang diselenggarakan oleh Kementerian Komunikasi dan Informatika Republik Indonesia. Untuk tahu lebih lanjut tentang program ini, silahkan mengunjungi [startupstudio.id](https://startupstudio.id)

KELAS DARING  
**SEKOLAH BETA**

H \_ C K \_ R

H \_ P \_ T E R

H U \_ T \_ E R

Ingin tahu lebih banyak tentang tiga peran penting dalam startup? Ikuti serial Kelas Daring Sekolah Beta dan belajar langsung dari ahlinya di industri startup.

Saksikan episode lengkapnya di:

[1000startupdigital.id/sekolahbeta](https://1000startupdigital.id/sekolahbeta)



# Cakap *Startup*

Pada episode ketiga dari *Cakap Startup*, Andreas Senjaya (CEO, iGrow) atau yang biasa disapa Mas Jay berbagi pengalaman tentang empati, visi, dan konsistensi.

i000 | Cakap  
Startup

Ep. 03

**Empati,  
Visi, dan  
Konsistensi**

**Andreas Senjaya**

Chief Executive Officer  
iGrow

Semasa mahasiswa, Mas Jay adalah pribadi yang aktif mengikuti berbagai organisasi dan juga memiliki prestasi kuliah yang cemerlang, seperti menjadi Mapres 2 Universitas Indonesia pada tahun 2020. Sebelum kuliah, Mas Jay bukanlah siswa yang aktif. Baru ketika di kampus ia bertemu banyak teman dan ikut banyak kegiatan, muncul dorongan untuk mengembangkan diri dan ia memutuskan untuk fokus pada *impact* yang bisa ia berikan selama hidup.

Dengan luasnya jejaring yang ia miliki dan tidak tergiur pada banyaknya tawaran pekerjaan padanya, ia memantapkan diri untuk merintis *startup* pertamanya pada tahun ke-3 kuliah, yakni Badr Interactive Inc. Di sana Mas Jay dan timnya memiliki visi untuk menciptakan 10 juta kebaikan harian

setiap harinya. Itu bukan hal yang tidak mungkin asal selalu berusaha untuk konsisten dan selalu berusaha mencari cara yang strategis dalam menciptakan dampak melalui karya yang diciptakan, dalam hal ini, dengan inovasi teknologi.

Memang Mas Jay belum pernah bekerja sebagai profesional sebelum

n  
ensi



membangun Badr Interactive Inc., sehingga muncul tantangan sebagai leader *startup*. Namun menurutnya, hal ini bisa diintervensi dengan membaca buku tentang *teamwork* dan kepemimpinan, serta berkunjung ke para pengusaha yang levelnya jauh lebih tinggi dari kita. Misalnya mereka yang sudah mengelola ratusan bahkan ribuan orang dalam timnya.

Kita bisa mendapatkan *insight* dari para pengusaha tersebut untuk mengintervensi *learning curve* yang sedang kita jalani, sehingga proses pembelajarannya lebih efektif dan efisien.

Saat ini, selain sibuk mengembangkan @igrowasia, Mas Jay pun aktif mendorong *startup founder* lainnya untuk berkembang bersama dalam berbagai komunitas dan juga melalui @founderplus.

Jika kamu adalah sahabat *startup* yang saat ini sedang berkuliah dan galau menentukan apakah harus terjun ke dunia *startup* sebagai pilihan karier, wajib banget kamu nonton Cakap *Startup* episode ini!

GERAKAN NASIONAL

**1000**

STARTUP DIGITAL

SEKOLAH BETA

KELAS DARING

Hacker

SERIAL HACKER

# Product Engineer

KELAS DARING  
SEKOLAH BETA

Hacker



1000

EPISODE 01:

## PRODUCT ENGINEER

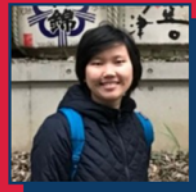
Apa saja peran seorang Product Engineer dalam sebuah perusahaan rintisan?



Selasa, 27 April 2021



20.00 - 21.00 WIB



**Janice Kartika**  
Product Engineer  
GOJEK



1000startupdigital.id



1000startupdigital



1000startupid



Dalam episode Sekolah Beta kali ini, Janice Kartika membagikan pengetahuan tentang role *Product Engineer*. Janice saat ini bekerja sebagai *Product Engineer* di Gojek, tepatnya mengurus bagian Transportation seperti GoRide, GoCar, GoBluebird, dan GoTransit untuk aplikasi di Android.

KELAS BERING  
SEKOLAH BETA

Hacker

Janice Kurnia

## Bekerja dalam Tim

- Berkomunikasi aktif dalam tim:
  - Bertanya tanpa sungkan
  - Mengemukakan pendapat
  - Memberikan saran/kritik
  - Mengajukan solusi dari masalah yang sedang didiskusikan.
  - Intinya ga boleh diam aja, ataupun terlalu galak.
- Mengerti bahwa tidak semua orang mengerti istilah-istilah di bidang kita, sesuaikan kalimat dengan bidang dari lawan bicara.
- Bahasa Inggris.

Dalam 1 tim, bisa ada:

- Product manager
- UI/UX designers
- Back-end devs
- Front-end devs
- Android devs
- iOS devs
- QAs
- Data Engineer/Analyst
- Engineering manager (EM)
- etc...

Silahkan untuk mengajukan pertanyaan melalui kolom chat ya!

Hal yang dikerjakan Janice di bagian *front-end development* seperti perancangan proses mulai dari pengguna bisa input alamat penjemputan, input alamat destinasi, *order*, hingga akhirnya pengguna mendapat *driver* dan sampai ke tempat tujuan. Singkatnya, pengguna dapat langsung merasakan apa yang Janice dan tim kerjakan.

Di sini Janice menjabarkan berbagai hal yang perlu dikuasai jika kamu ingin berkarir sebagai *Programmer* atau *Developer* suatu aplikasi. Meskipun seorang *programmer* memiliki kemampuan dan pengetahuan pemrograman yang mumpuni, belum tentu menjamin kelancaran dalam bekerja. Karena pasti

bekerja di dalam tim yang isinya banyak, kemampuan kerjasama dalam tim dan komunikasi haruslah bagus supaya tidak terjadi kesalahpahaman.

Sebagai *product engineer*, penting untuk punya pola pikir kalau produk akan berubah mengikuti kebutuhan pasar. Karena kode yang dirancang bergantung pada requirement dari tim produk maupun keinginan masyarakat, atau bahkan penyesuaian karena kompetisi dengan perusahaan lain. tentang pentingnya keberadaan *brand voice* dan *tone of voice*. Karena suatu *copy* atau tulisan di aplikasi sebaiknya punya “kepribadian” agar user tidak merasa dilayani oleh “mesin”.

Untuk diskusi yang lebih lengkap, jangan lupa menonton episode ini, ya! Videonya bisa diakses melalui **Youtube Gerakan Nasional 1000 Startup Digital**.

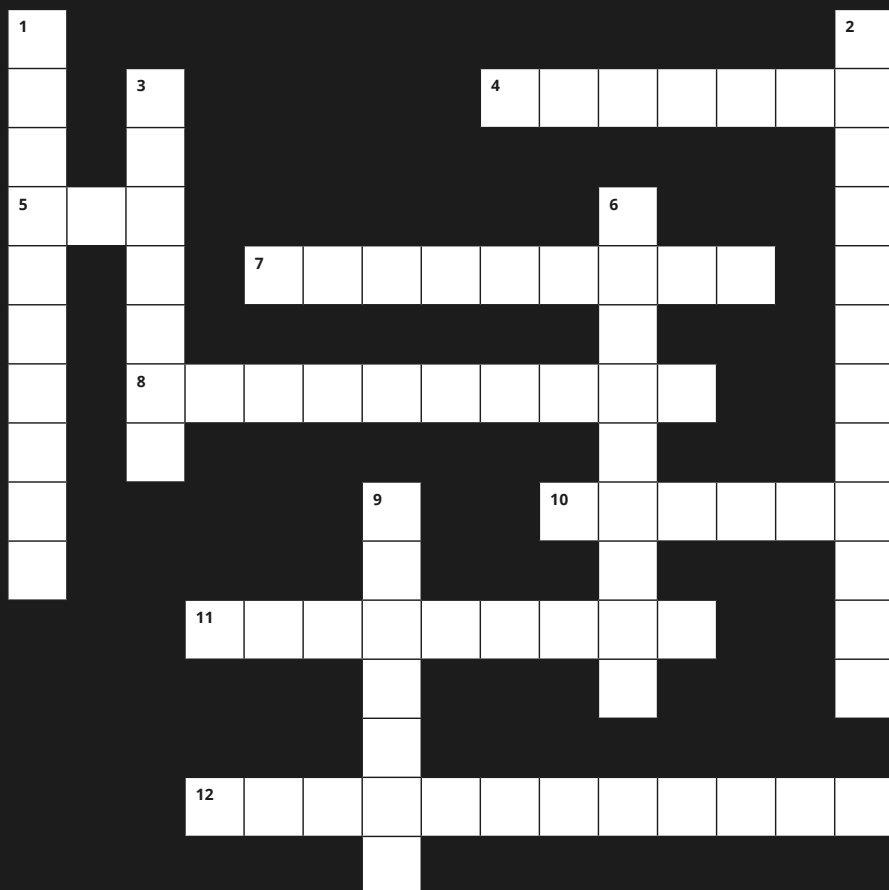
Saksikan dan ikuti diskusinya di:



# Teka Teki

## *Startup*

Uji pengetahuanmu tentang *startup*  
dengan mengerjakan teka-teki berikut!



#### MENDATAR

4. Kumpulan data sangat besar.
5. Perjanjian untuk menjaga informasi rahasia milik perusahaan.
7. Tingkat persentase customer berhenti berlangganan layanan perusahaan dalam setahun.
8. Proses pengenalan & pendalaman oleh karyawan baru saat masuk perusahaan.
10. Visualisasi sebuah konsep desain.
11. Teknik penjualan menawarkan produk sejenis yang nilai produknya lebih mahal dari produk yang ingin dibeli konsumen.
12. Teknik penjualan dengan menawarkan produk komplementer dari produk utama.

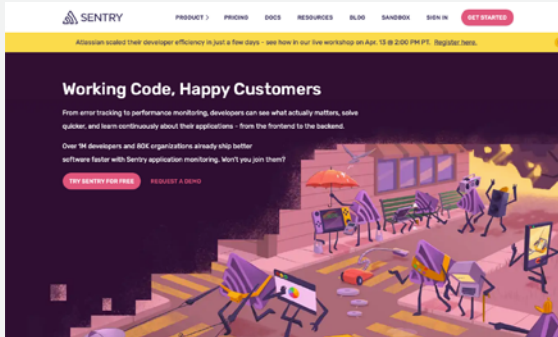
#### MENURUN

1. Kerangka rencana bisnis untuk membantu memecah ide menjadi beberapa asumsi agar mudah dipahami.
2. Pengguna produk, inovasi, atau teknologi baru sebelum orang lain dalam populasi yang luas.
3. Program komputer yang dirancang untuk simulasi percakapan intelektual dengan  $\geq 1$  manusia secara audio/teks.
6. Masalah berulang pada produk/layanan yang mengganggu kenyamanan pelanggan.
9. Proses pengulangan.



Cek jawabanmu  
dengan  
mengakses  
barcode ini.

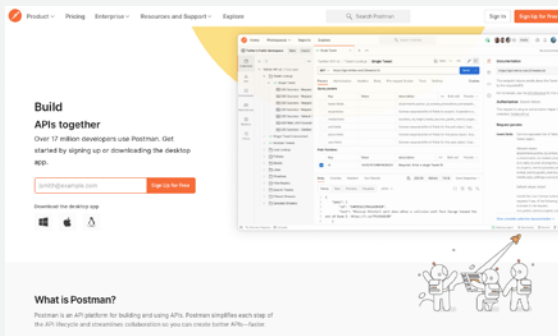
# Startup Hacker



## SENTRY CARA PRAKTIS MEMANTAU PERFORMA APLIKASI

Sentry adalah tools yang akan membantu memantau aplikasi, , dan game milikmu. Semua bugs akan dilaporkan secara *real time* agar *developer* dapat dengan sigap memperbaikinya. Terintegrasi dengan banyak tools pendukung, pengguna bisa bekerja secara efektif dan efisien.

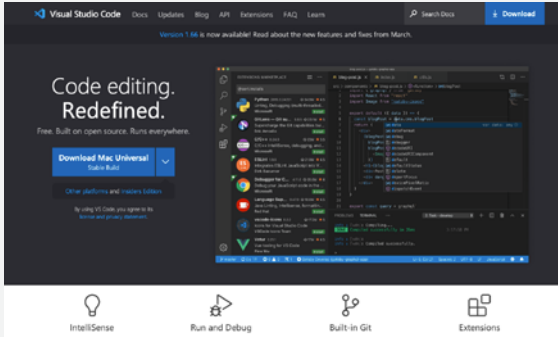
[HTTPS://SENTRY.IO](https://sentry.io)



## POSTMAN TOOLS WAJIB PARA API DEVELOPER

Sebuah *platform* yang bisa digunakan untuk membuat, menggunakan, dan menguji API (*Application Programming Interface*) yang telah dibuat. Postman juga dapat digunakan untuk mengumpulkan API menjadi sebuah dokumentasi utuh untuk satu proyek tertentu. Fitur unggulan lainnya yaitu *environment*, *response*, *mock server*, *script test*, *automated test*.

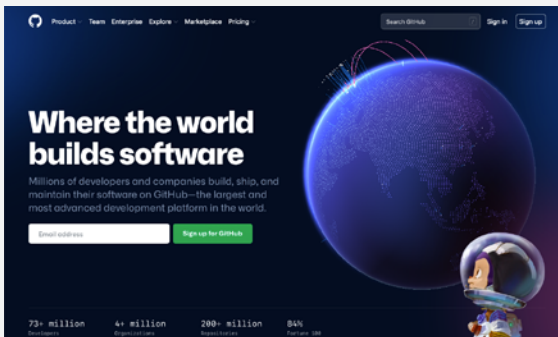
[HTTPS://WWW.POSTMAN.COM](https://www.postman.com)



## VISUAL STUDIO CODE TEXT EDITOR BERKONSEP ONE FOR ALL

Visual Studio Code (VS Code) merupakan *text editor* yang unggul karena dapat digunakan *cross platform* untuk Windows, Linux, dan Mac OS. Salah satu fitur andalannya adalah *Code Debugging*. VS Code menggunakan bahasa pemrograman Java, Javascript, GO, Node JS, Python, C++.

[HTTPS://CODE.VISUALSTUDIO.COM](https://code.visualstudio.com)



## GITHUB RUANG CODING & KOLABORASI NEXT LEVEL

Sebuah dan layanan berbasis *cloud* untuk menyimpan, mengelola, mendokumentasikan, dan mengontrol perubahan kode. Github memungkinkan pengguna berkolaborasi secara *online* saat bekerja dalam sebuah proyek. Di GitHub, *programmer* bisa langsung membuat portofolio dari hasil *coding* proyek kerjamu.

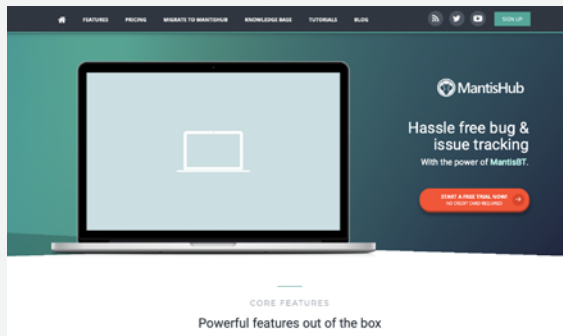
[HTTPS://GITHUB.COM](https://github.com)



## BLUESTACKS EMULATOR ANDALAN GAMERS

Dikenal sebagai aplikasi emulator untuk menjalankan game berbasis Android di perangkat Windows, BlueStacks memiliki fitur unggulan *Eco Mode*. Fitur ini dapat mengurangi penggunaan GPU (*Graphics Processing Unit*). Selain itu ada kemampuan *multitasking* serta *interface* yang intuitif.

[HTTPS://WWW.BLUESTACKS.COM](https://www.bluestacks.com)



## MANTISHUB KOMUNITAS ONLINE UNTUK PROGRAMMER

*Platform* untuk belajar, berbagi pengetahuan, berkolaborasi, dan membangun karier *developer* di seluruh dunia. Memiliki misi pemberdayaan dunia untuk pengembangan teknologi melalui pengetahuan kolektif, Stack Overflow memiliki 3 produk unggulan yaitu Stack Overflow for Teams, Stack Overflow Advertising, dan Stack Overflow for Talent and Jobs.

[HTTPS://MANTISHUB.COM](https://mantisHub.com)

1000

# Cakap Startup

Cari tahu beragam **cerita, pengalaman, tips,**  
dan **curhatan founder** hanya **di serial video**  
**podcast** dari **#1000StartupDigital**

Saksikan di:



# Glosarium



**AGILE:** metodologi pembuatan produk yang menggunakan siklus produksi pendek sehingga memungkinkan ide-ide diimplementasikan dan diuji dengan cepat untuk mengumpulkan umpan balik dan keputusan berbasis data dapat dibuat

**BACKEND:** sistem di balik layar yang mengolah *database* dan *server*

**BUG:** kecacatan atau kerusakan teknis dalam sebuah program yang dapat menyebabkan aplikasi/ tersebut tidak berfungsi sebagaimana mestinya

**CYBERSECURITY:** praktik melindungi berbagai sistem, jaringan, dan program dari serangan-serangan digital

**DEBUG:** proses mengidentifikasi dan menghapus *bug* atau *error* di dalam kode yang mengakibatkan sistem tidak berjalan baik atau bahkan mengalami *crash*

**DEEP LEARNING:** adalah bagian dari pembelajaran *Machine Learning* yang menggunakan *artificial neural networks* (ANN), yaitu metode pembelajaran oleh mesin dengan meniru cara kerja sistem saraf otak manusia

**FRONTEND:** hal yang pengguna lihat pada tampilan sebuah /aplikasi

**MACHINE LEARNING:** adalah ilmu yang berfokus pada proses pembelajaran komputer untuk bertindak seperti otak manusia dengan belajar secara mandiri dari waktu ke waktu. *Machine learning* merupakan salah satu cabang *artificial intelligence* (AI) yang paling banyak digunakan saat ini

**SPRINT:** dalam metode pengembangan produk *agile*, *sprint* adalah periode waktu tertentu di mana pekerjaan tertentu harus diselesaikan dan siap untuk ditinjau

**UNIQUE SELLING PROPOSITION:** faktor-faktor yang membuat kamu lebih unggul daripada kompetitor

**QUALITY ASSURANCE:** rangkaian proses sistematis untuk menentukan apakah suatu produk atau jasa memenuhi syarat yang ditentukan

**WATERFALL MODEL:** model pengembangan perangkat lunak yang menekankan fase-fase yang berurutan dan sistematis



# Kamu bisa mendapat Rintisan versi cetak!

Untuk menjangkau pembaca setia buku saku Rintisan lebih dekat, Kementerian Komunikasi dan Informatika membuka kesempatan khusus bagi para pembaca terpilih untuk kami kirimkan buku saku Rintisan versi cetak setiap kali edisi baru diterbitkan.



## Gimana caranya?

Ceritakan dengan detail mengapa kamu harus menjadi satu dari para pembaca terpilih yang akan kami kirimkan buku saku Rintisan dalam *form* ini:

<http://bit.ly/komunitas-rintisan>

# Bergabung Jadi Kontributor

Rintisan memiliki satu tujuan utama: menjadi sarana untuk membuka wawasan mengenai *startup*, talenta digital, ide, dan inovasi di ekosistem digital Indonesia.

Artikel di Rintisan memiliki topik yang beragam dan relevan untuk berbagai industri dan fungsi manajemen. Adapun beberapa area fokus yang dibahas adalah kepemimpinan, strategi, teknologi, operasional, *branding*, *marketing*, *legal*, keuangan, manajemen sumber daya manusia, produktivitas, dan kreativitas. Dalam memilih artikel yang diterbitkan, ini adalah 4 poin yang Rintisan cari:

## ORISINALITAS

Walaupun suatu topik mungkin sudah banyak yang membahas, selalu ada cara untuk melihatnya dari sudut pandang yang berbeda. Temukan itu dan bagikan pada pembaca.

## KEAHLIAN

Siapa pun dapat menjadi kontributor, asalkan ia benar-benar menguasai materi dalam tulisannya.

## KEGUNAAN

Utamakan gagasan yang praktis dan dapat dengan mudah dimengerti. Jika kamu bisa menjelaskan pemikiranmu sehingga pembaca mengerti bagaimana menerapkannya dalam situasi nyata, artikelmu dapat mengubah hidup seseorang!

## BUKTI

Menulis secara deskriptif itu bagus. Namun, lebih baik lagi jika tulisanmu didukung dengan data dan fakta.

## Ingin menjadi kontributor bagi Rintisan?

Kirim tulisanmu ke [gerak@1000startupdigital.id](mailto:gerak@1000startupdigital.id) dengan subjek “**Artikel untuk Rintisan**”. Kami akan memberikan *merchandise* bagi kontributor yang artikelnya terpilih untuk diterbitkan.

# Kritik & Saran



Kami ingin mendengar pendapatmu mengenai artikel dan topik yang kami sajikan. Yuk, sampaikan kritik dan saranmu di [bit.ly/ksrintisan](https://bit.ly/ksrintisan)

