

Konsep Produk

Silver - Chapter 0 - Topic 2

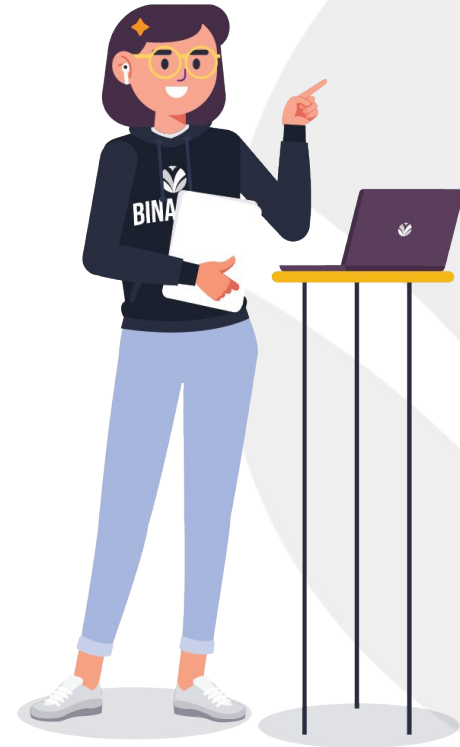
Selamat datang di **Chapter 0 Topic 2**
online course dari
Binar Academy!



Hai teman-teman! 🎉

Di topic sebelumnya, kamu udah belajar mengenai sejarah singkat aplikasi dan apa sih aplikasi itu. Di sini, kamu bakal belajar lebih dalam lagi tentang **isi aplikasi dan siapa aja orang-orang di balik layarnya**.

Kalau gitu, yuk langsung aja kita kepoin~



Apa yang bakal kamu pelajari di Topic 2 ini?

Pada kesempatan kali ini, kamu akan mengenal lebih dalam mengenai:

- 1) Cara-cara bisnis berbasis internet bikin revenue
- 2) Konsep sederhana frontend, backend, dan API
- 3) Perbedaan karakteristik project dan product



Kalau di topic sebelumnya kita udah belajar sejarah kenapa aplikasi bisa booming, sekarang kita masuk lebih dalam tentang salah satu pemain besarnya.

Pernah kepikiran gak, kenapa Facebook bisa tajir melintir padahal bikin akunnya gratis?



Sebelum diakuisisi, WhatsApp sempat berbayar, loh!

Harganya \$ 1 untuk setiap download lewat App Store. Whatsapp meraup \$ 0.70 karena Apple bikin skema pembagian 70:30 di App Store!

Artinya, tiap transaksi yang dilakukan oleh pengguna Apple di App Store, Apple bakal dapat 30%-nya.

Dalam kasus ini, tiap kali WhatsApp didownload, Apple bakal dapat \$ 0.30 dan WhatsApp \$ 0.70.



Sekarang kita ambil contoh di Facebook

Bisnis mana sih yang gak pengen ada pendapatan? Nah, walaupun bikin akunnya bisa gratis, tapi Facebook dapet duit dari para advertiser. Gimana ceritanya?



Kamu yang punya akun personal emang gak bayar pakai duit. Tapi **kamu "bayar" ke Facebook adalah dengan hal-hal sederhana yang kamu lakukan.** Kayak ngisi biodata, beraktivitas kayak ngelike atau ngeview, apa aktif di forum.

Yang bayar duit ke Facebook disebut advertiser yang pengen ngiklanin produknya ke kamu. Cerdas ya Facebook!



Intinya, baik Facebook atau bisnis berbasis internet lainnya perlu punya pendapatan (revenue) buat bayar operasional, kayak bayar gaji karyawan, nyewa kantor, dsb.

Cara mereka cari uang (revenue stream) juga unik, karena cara bekerja satu platform dan lainnya juga berbeda.



Nah, sekarang kita bahas sisi teknis tapi pakai startup lokal, yuk!

Di sesi ini, kita coba bongkar dapur engineering yang ada di Tokopedia.

Per April 2019, ada 4,200 nakama (sebutan pegawai Tokopedia) yang bekerja baik di dapur atau depan layar marketplace hijau ini. Banyak banget, ya? Tapi kayak gimana sih pembagiannya?



**Tapi sebelum masuk ke Tokopedia,
kita mulai pakai analogi yang
lumayan familiar dulu kali ya**

Anggep begini: Ceritanya, kalian lagi laper banget dan pengen makan burger nih.



Kamu (user) masuk ke restoran dan pilih meja makan.
Gak lama duduk, kamu didatengin sama pelayan.

👤 "Silakan menunya."

👤 "Makasih, Mbak. Saya pesen beef burger gak pakai acar, satu porsi ya."

👤 "Oke, Kak. Ada tambahan lain?"

👤 "Udah, itu aja."

👤 "Silakan tunggu sebentar. Terima kasih."

👤 "Sama-sama."

Sehabis milih menu, pesenanmu bakal dicatet di sebuah kertas orderan.

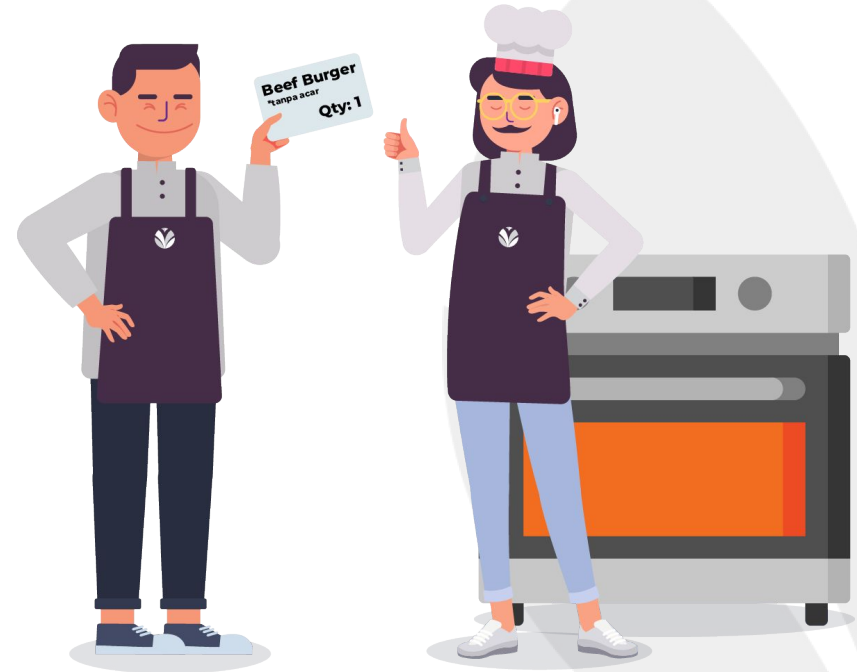


Habis itu, pelayan tadi bakal nerusin catetan pesenanmu ke koki di dapur.

👤 "Sist, meja no. 10 pesen beef burger tanpa acar sama kentang goreng, satu porsi."

👨‍🍳 "Oke siap!"

Koki yang ada di dapur masakin beef burger kamu di sana. Mulai dari menyiapkan roti bun, patty, tomat, kentang, dll.



Masakan udah jadi dan siap dihidangkan. Koki di dapur manggil pelayan lagi.

 "Udah jadi nih!"

 "Oke tengkyuu."



Pelayan bawain pesenanmu ke meja makan. Burger udah siap dimakan.

👤 "Silakan Kak pesanannya: beef burger tanpa acar."

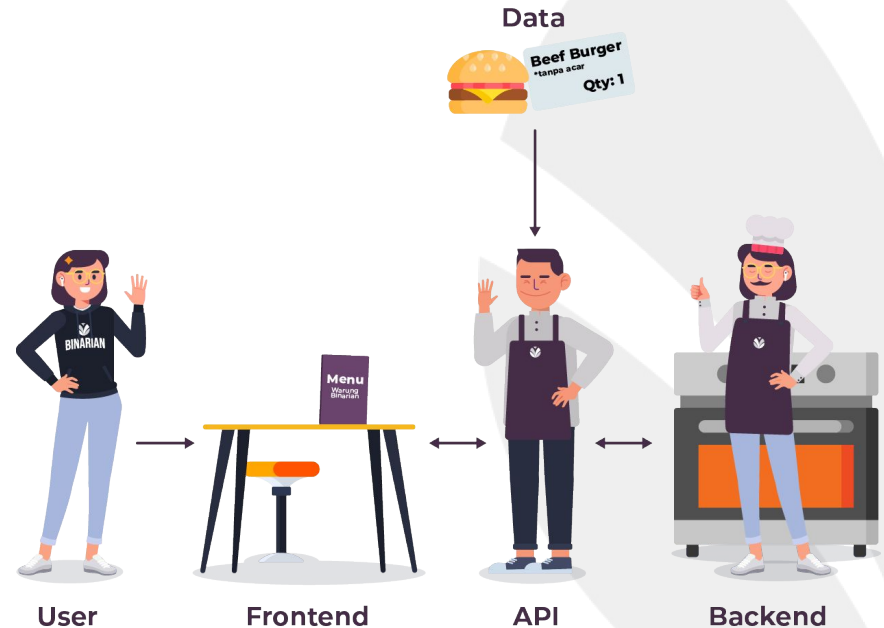
👩 "Oke. Terima kasih, Mbak."

Prosesnya simpel, kan?



Nah, cara kerja mobile app kurang lebih kayak proses di restoran tadi.

- Meja makan tempat kamu pesen tadi disebut **frontend**.
- Dapur tempat koki bikin burgermu disebut **backend**.
- Pelayan yang bolak-balik tadi adalah penghubung frontend dan backend yang disebut **API**.
- Kertas orderan dan beef burger disebut **data**.



Balik lagi ke Tokopedia ya. Kalau kamu buka app-nya dan search "sweater supermi" itu artinya kamu ngetik di frontend-nya Tokopedia.

Ssst.. **mau dari website apa mobile app, tempat di mana user bisa berinteraksi sama program, sebutannya frontend.**



Nah, trus pas kamu tap button Submit atau tekan Enter di kolom bar pencarian, ada sesuatu yang terjadi.

Frontend ngirim request ke backend: "Ada yang cari sweater supermi, nih!"

Request ini disampaikan secara otomatis oleh API dan ditujukan ke backend-nya Tokopedia.



Di sisi lain, backend terima request dari API kalau ada yang lagi cari "sweater supermi".

Nah, habis itu backend Tokopedia mulai langsung ngambilin "sweater supermi" di database, trus API nganterin barangmu ke frontend.

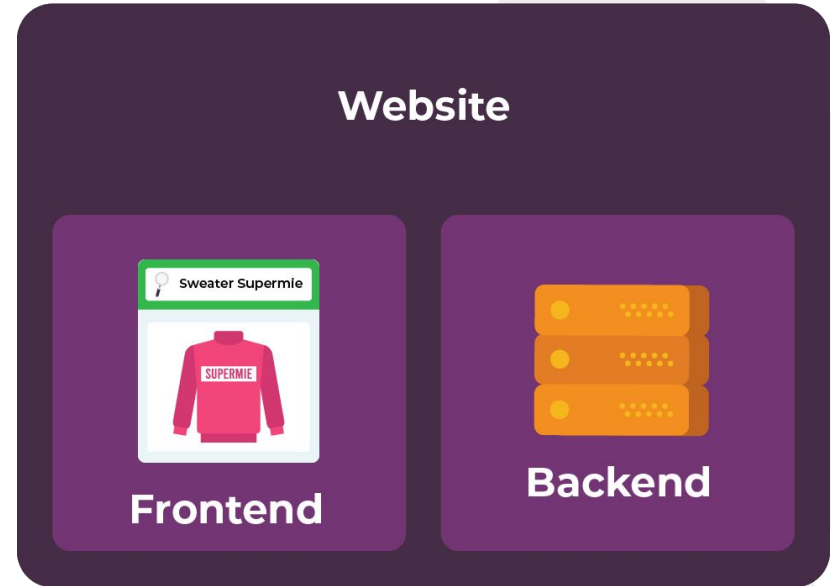


Terakhir, "sweater supermi" bakal ditampilkan di frontend dalam bentuk hasil pencarian yang ditunjukkan melalui visual di web atau app Tokopedia yang kamu buka.



Kenapa harus pakai API sih? Kenapa gak langsung aja dari frontend ke backend?

Nah, ini gambaran simpelnya kalau gak ada API.



Kalau kita mau bikin app android, iOS, dan website, kita jadi harus kerjain proses nganterin data ini 3 kali.

Kalau di restoran, yang terjadi kira-kira begini:

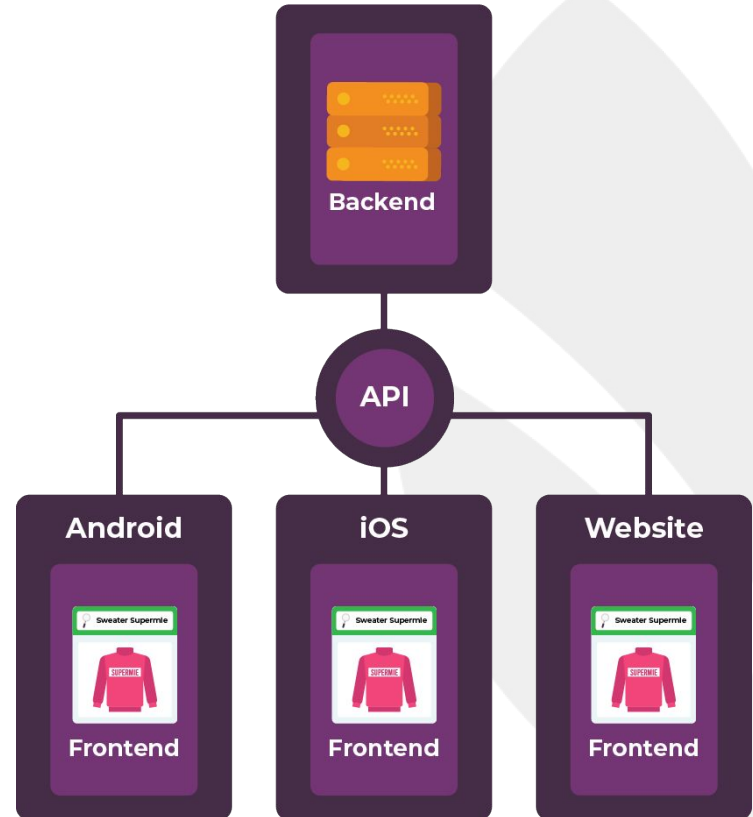
- 1) Meja makan (frontend) ada di 3 area terpisah
- 2) Dapur (backend) juga harus ada 3
- 3) Padahal lokasi restoran sama menunya (data) sama

Repot kan ya kalau gak pakai API?



Dengan adanya API, arsitekturnya jadi lebih simpel dan cara kerjanya juga jadi gampang

Daripada harus bikin 3 backend dan 3 frontend yang berbeda, **cukup bikin 1 backend yang ngomong ke 3 frontend yang berbeda, melalui API.**



FACEMASH

Were we let in for our looks? No. Will we be judged on them? Yes.

Who's Hotter? Click to Choose.



OR



Mumpung masih anget, kita lanjut cerita soal Traveloka, Tokopedia dan Facebook

Facebook, awalnya cuma project iseng bernama "**Facemash**" dan tujuan Mark Zuckerberg pas bikin project ini cuma buat ngebandingin mana yang lebih cakep antara 1 orang dan yang lain.

Nah, Facebook yang kita kenal sekarang akhirnya jadi sebuah product raksasa. Di dalamnya, banyak banget fitur kayak "Tag photo", "Create Group", dan lainnya.

Perbedaan paling keliatan dari project dan product yaitu soal rentang waktu

Project biasanya punya batas waktu dan target yang spesifik. Project biasanya dibikin buat nyelesein kondisi tertentu.

Contoh: Binar Academy ngajar di Batam dengan target sekian lulusan. Begitu target tercapai, kalau project-nya gak diperpanjang, berarti udah selesai.



PROJECT

vs.

PRODUCT

outputs

scope, time, budget

abstract (work management)

predictive

has a finish

investments promise ROI

outcomes

continuous value

concrete (an actual thing)

adaptive

ongoing

value drives investment

Sementara **product biasanya ada "barang konkret" yang dipakai buat menyelesaikan masalah seseorang.** Karena gak ada 'selesai'-nya, maka product bakal selalu berkembang.

Contoh: Binar Academy bikin aplikasi biar kamu bisa belajar dari mana aja. Awalnya appnya cuma bisa buat buka soal, lama-lama bisa buat buka video.

Berikut adalah bedanya project dan product.

Kita bisa ambil contoh Traveloka sebagai product

Awalnya, Traveloka dirilis cuma buat cari tiket pesawat. Tapi seiring waktu, Traveloka nambah fitur baru. Mulai dari booking hotel sampai rental mobil.

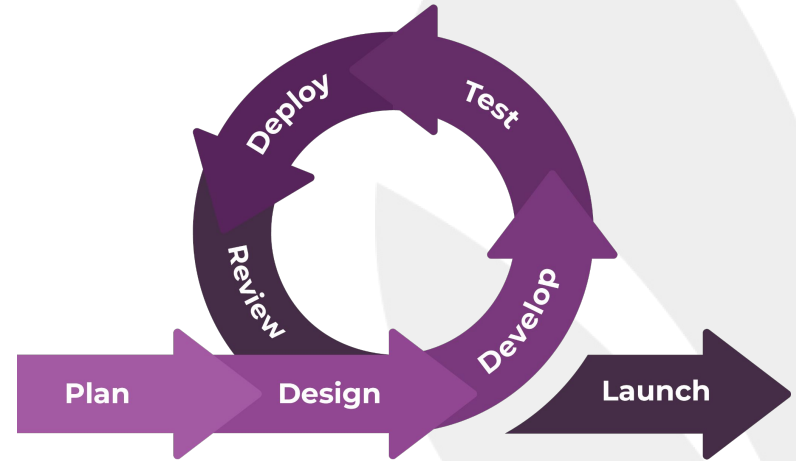
Bahkan sekarang, Traveloka juga jual tiket taman bermain yang bisa dipakai pengguna pas liburan.

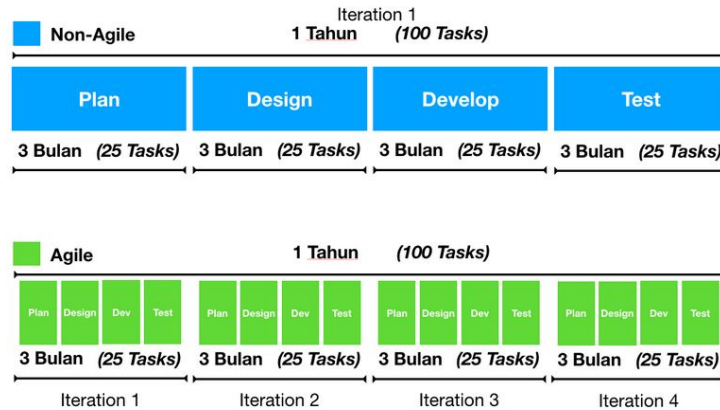


Dalam suatu product, yang terpenting adalah komponen tim dan metode kerjanya

Nah, buat perusahaan berbasis internet dan aplikasi, metode kerja yang umum dipakai adalah metode Agile.

Agile ini metode kerja yang berulang-ulang. Artinya, begitu selesai, bakal selalu ada evaluasi, perbaikan, riset, plan buat fitur baru, dan gitu terus...





Agile ini bukan berarti cepat, tapi artinya gesit atau lincah. Agile dibutuhkan biar bisa terus adaptasi sama perubahan.

Coba liat gambar di samping. Bayangin kalau kita cuma punya waktu 1 tahun untuk bikin suatu aplikasi. Nah, kalau gak pake Agile, kita cuma bisa bikin 1 kali.

Sementara kalau pakai agile, kita bisa bikin sampai 4 kali.

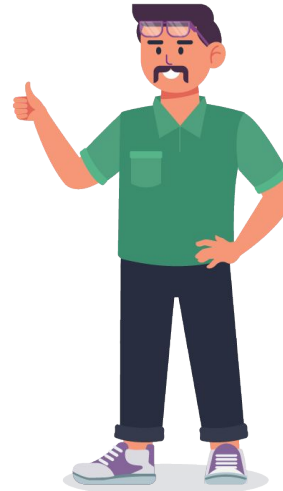
Nah, detail tentang metode pengerjaan Agile ini bakal dibahas di topic 3 ya!

Dalam Agile, ada yang disebut Scrum

Nah, kalau mau bikin product yang oke, metode kerja Scrum bakal mayan sering dipakai.

Secara garis besar, Scrum cuma ngebagi tim jadi 3 fungsi pekerjaan yang setara pentingnya:

- **Product owner**: role ini yang menentukan akan membuat product apa.
- **Scrum master**: servant leader, alias yang selalu bantuin tim dan mastiin masalah mereka selesai.
- **Team member**: mastiin produknya bisa sesuai dengan kriteria dari Product owner.



**Product
Owner**



**Scrum
Master**



**Team
Member**

Oke, selesai sudah pembahasan di Chapter 0 - Topic 2 ini. Tanggung ya? Penasaran apa yang dibahas besok?

Tapi sebelum lanjut, yuk kita rekap apa aja yang udah kamu pelajari di topic ini:

- 1) Cara-cara bisnis berbasis internet bikin revenue
- 2) Konsep sederhana frontend, backend, dan API
- 3) Perbedaan karakteristik project dan product



Yuhuuu! Kamu udah berhasil
menamatkan **Chapter 0 Topic 2** 🎉

Nanti, di topic 2 kita bakal ngebahas
tentang Agile dan SDLC lebih dalam!
Udah siap?

