ibimbing

# Modul Pembelajaran

## Product Management

# Problem Definition and Solutioning

## 1. *Introduction*

Sebagai PM penting untuk dapat menemukan masalah dan mencari solusi dari masalah yang ditemukan. Pentingnya *problem definition* dan *solutioning* adalah agar dapat melakukan pengelolaan produk dapat lebih terarah dan masalah yang ditemukan dapat ditentukan secara tepat dari akar hingga solusi yang komprehensif untuk diaplikasikan oleh tim.

Problem definition (pengartian masalah) dan solutioning (pembuatan solusi) adalah dua tahap yang sangat penting dalam pengembangan produk yang menjadi bagian integral dari pekerjaan seorang Product Manager (PM). Berikut adalah penjelasan mengapa kedua tahap ini sangat diperlukan:

1. Problem Definition (Pengartian Masalah):

- Memahami Prioritas: Tahap ini membantu PM untuk memahami dengan jelas masalah apa yang harus dipecahkan terlebih dahulu. Dalam pengembangan produk yang sukses, prioritas masalah yang akan dipecahkan harus ditentukan dengan baik agar sumber daya tidak terbuang percuma.

- Orientasi Pengembangan Produk: Dengan mendefinisikan masalah dengan baik, PM dapat menentukan arah pengembangan produk yang lebih tepat. Hal ini membantu dalam menghindari pengembangan produk yang "terlalu banyak fitur" atau kurang fokus pada apa yang benar-benar penting bagi pengguna.
- Menghindari Asumsi: Definisi masalah yang jelas membantu dalam menghindari asumsi tentang apa yang dianggap sebagai masalah. Ini memungkinkan PM untuk memastikan bahwa masalah yang akan dipecahkan didasarkan pada data dan pemahaman yang kuat.

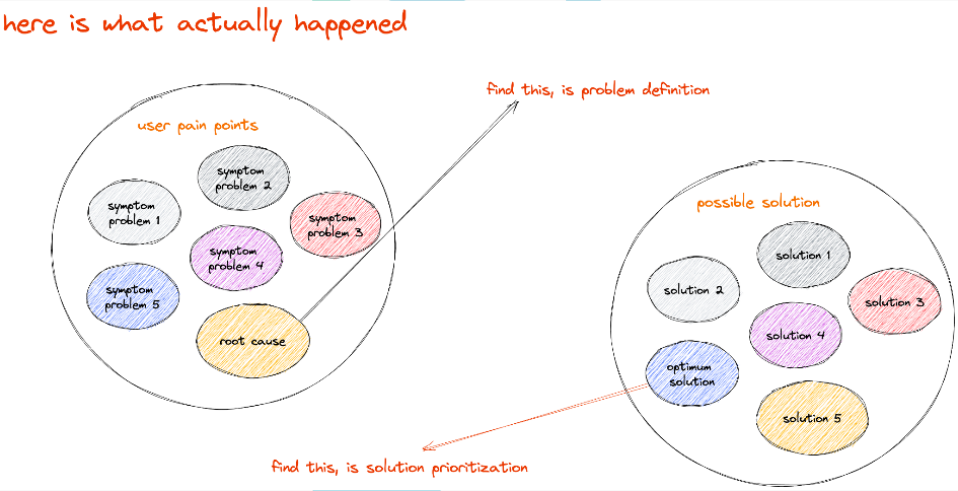
2. Solutioning (Pembuatan Solusi):

- Mengembangkan Solusi yang Efektif: Setelah masalah didefinisikan dengan baik, langkah selanjutnya adalah mencari solusi yang efektif. Ini melibatkan pemikiran kreatif dan strategis untuk mengembangkan produk atau fitur yang dapat mengatasi masalah tersebut.
- Meningkatkan Pengalaman Pengguna: Solusi yang baik dapat meningkatkan pengalaman pengguna dengan mengatasi masalah atau kebutuhan mereka. Ini penting untuk mempertahankan atau meningkatkan retensi pengguna dan kepuasan pelanggan.
- Mengukur Kesuksesan: Solusi yang baik harus dapat diukur keberhasilannya. PM dapat menentukan metrik kinerja yang sesuai untuk melihat apakah solusi tersebut efektif dalam mengatasi masalah yang telah didefinisikan.
- Mengarahkan Tim Pengembangan: Solusi yang jelas membantu dalam mengarahkan tim pengembangan dengan memahamkan mereka tentang apa yang harus dibangun dan mengapa itu penting. Ini memungkinkan tim untuk bekerja lebih fokus dan efisien.

Kombinasi antara pengartian masalah yang baik dan proses solutioning yang efektif membantu PM dalam menghasilkan produk yang relevan, efektif, dan berkualitas. Tanpa pengartian masalah yang tepat, solusi yang dibuat mungkin tidak akan memiliki dampak yang signifikan atau bahkan mungkin melewatkan masalah yang sebenarnya perlu dipecahkan. Sebaliknya, tanpa proses solutioning yang kuat, bahkan jika masalahnya

terdefinisikan dengan baik, produk mungkin tidak akan mampu mengatasi masalah tersebut. Oleh karena itu, kedua tahap ini adalah landasan penting dalam peran seorang PM dalam mengembangkan produk yang sukses.

Pada dasarnya pendefinisian masalah dan penentuan solusi seperti ditunjukan pada Gambar di bawah yang menggambarkan bahwa terdapat banyak masalah dan banyak pula solusi yang dapat ditemukan sekaligus.




Gambar 22. Ilustrasi Masalah dan Solusi yang dihadapi

## 2. *Problem Definition*

Sebagai PM, mendefinisikan suatu masalah menjadi hal yang penting dalam dunia *development product*. Dengan mendefinisikan masalah maka dapat ditemukan beberapa hal dan alasan mengapa masalah tersebut dapat terjadi serta memahami akar dari masalah tersebut.

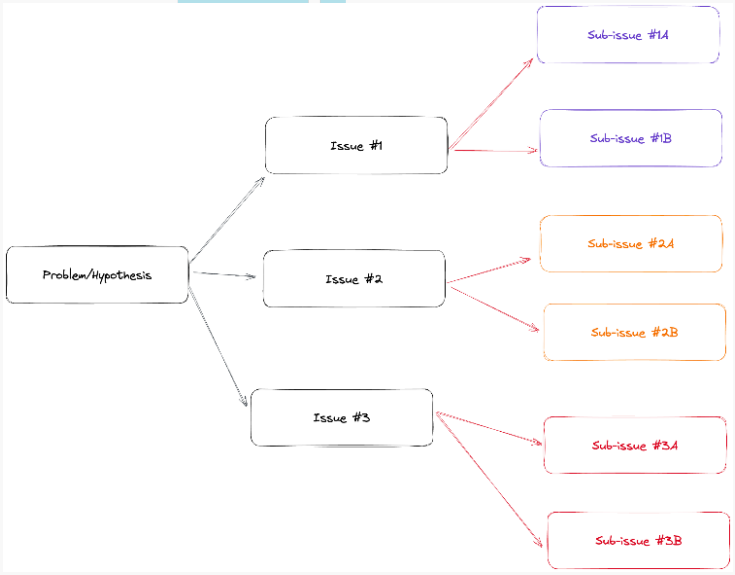
Dalam proses Product Management, mendefinisikan masalah dengan tepat adalah langkah awal yang esensial untuk merancang solusi yang efektif. Untuk mencapai hal ini, terdapat beberapa pendekatan yang dapat digunakan. Berikut adalah tiga cara yang sering digunakan untuk mendefinisikan masalah:

a. Issue Tree: Mengurai masalah menjadi masalah-masalah kecil

Issue Tree merupakan metode yang kuat untuk menguraikan masalah yang kompleks menjadi komponen yang lebih terukur dan terkelola. Pendekatan ini melibatkan pembuatan hierarki dari masalah utama hingga masalah-masalah yang lebih spesifik. Setiap masalah dipecah menjadi sub-masalah yang lebih kecil, menciptakan struktur yang terorganisir dan terstruktur.

**Contoh:**

Jika masalah utama adalah "Pengguna Mengalami Tingkat Penggunaan yang Rendah," maka sub-masalah bisa termasuk "Registrasi yang Rumit," "Navigasi yang Tidak Intuitif," dan "Fitur yang Kurang Menarik."



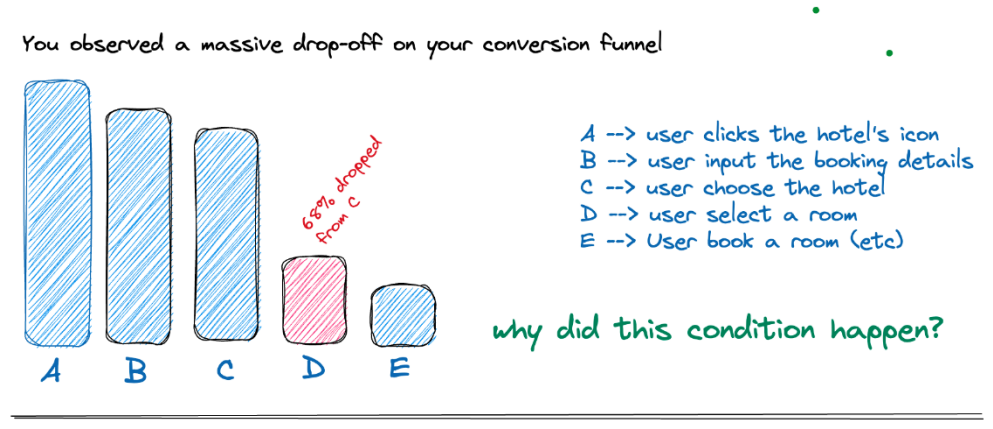

Gambar 23. Ilustrasi *Issue Tree*

b. Funnel/Conversion Analysis

Funnel Analysis atau Conversion Analysis digunakan untuk melacak perjalanan pengguna dalam bentuk pengukuran, seperti dari kunjungan ke situs web hingga pembelian produk. Pendekatan ini membantu mengidentifikasi titik-titik di *journey* tersebut di mana pengguna paling sering keluar atau tidak melanjutkan penggunaan aplikasi.

**Contoh:**

Dalam proses e-commerce, jika terdapat banyak pengguna yang meninggalkan keranjang belanja mereka sebelum pembayaran, ini bisa menjadi masalah yang perlu dipecahkan.



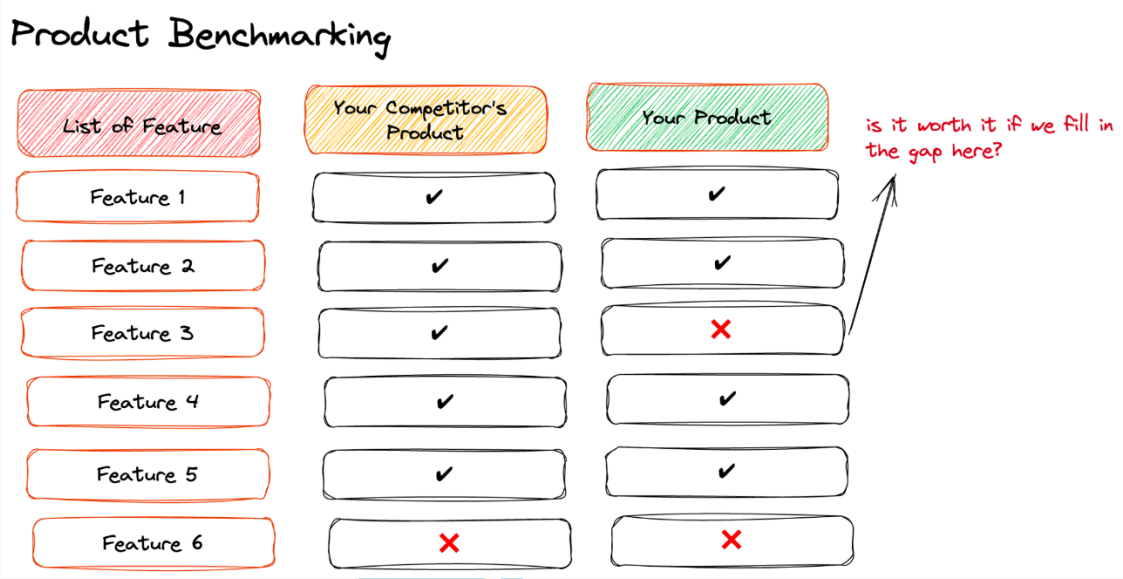

Gambar 24. Ilustrasi *Funnel/Conversion Analysis*

c. Benchmarking/Gap Analysis

Dalam pendekatan ini, bertujuan untuk membandingkan kinerja produk atau layanan *product* yang dimiliki dengan pesaing atau standar industri. Ini membantu mengidentifikasi kesenjangan (gap) antara kinerja aktual dan harapan pengguna atau standar pasar.

**Contoh:**

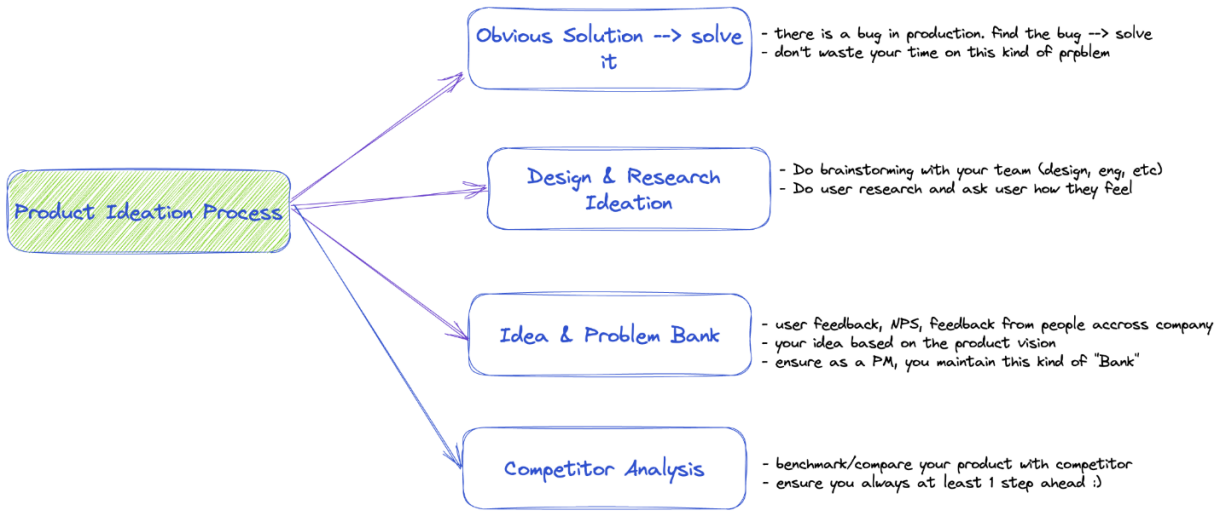
Jika produk Kita memiliki waktu respons yang lebih lambat dibandingkan dengan pesaing utama, ini bisa menjadi masalah yang memerlukan solusi.




Gambar 25. Ilustrasi *Product Benchmarking Analysis*

### 3. *Product Ideation*

Sebagai PM, pentingnya untuk selalu memiliki ide-ide yang *out of the box* yang selanjutnya agar dapat dilakukan *improvement* terhadap produk yang sedang dikembangkan. Dalam product ideation tersebut tidak serta merta dilakukan oleh seorang PM, melainkan dapat berkolaborasi dengan tim ataupun orang-orang yang berada di tempat yang sama. Tetapi bagaimana sebenarnya PM dapat menafsirkan ide-ide yang dimiliki agar dapat dilakukan *improvement* terhadap produknya? Hal tersebut dapat dilakukan dengan *Product Ideation Process*. *Product ideation process* merujuk pada proses generasi, seleksi, dan implementasi gagasan dalam tim produk sebelum memasuki tahap produksi. Hal ini adalah saat di mana tim berdiskusi tentang berbagai aspek produk sebelum produk tersebut benar-benar dibuat. Ideasi produk melibatkan pembicaraan yang mendalam untuk menentukan fitur-fitur, fungsionalitas, dan arah keseluruhan produk. Adapun Langkah *Product Ideation Process* dilakukan seperti pada Gambar di bawah.




Gambar 26. Ilustrasi *Product Ideation Process*

### 4. *Solution Prioritization*

Dalam dunia Product Management, tidak semua solusi diciptakan setara. Proses prioritasi solusi adalah langkah penting untuk menentukan solusi mana yang perlu diimplementasikan terlebih dahulu. Ada beberapa kerangka kerja dan teknik yang membantu dalam prioritas ini, termasuk ICE Framework, RICE Framework, dan Metode Moscow.

a. ICE Framework

ICE adalah singkatan dari Impact, Confidence, dan Effort. Ini adalah kerangka kerja yang digunakan untuk menilai proyek atau fitur potensial berdasarkan tiga faktor kunci:

- Impact (Dampak): Ini mengukur sejauh mana proyek atau fitur tersebut akan berdampak pada pengguna, bisnis, atau tujuan yang ingin dicapai. Dampak ini biasanya diukur dalam skala tertentu, misalnya, skala 1 hingga 10.
- Confidence (Keyakinan): Ini mengukur sejauh mana tim memiliki keyakinan bahwa proyek atau fitur tersebut akan berhasil. Keyakinan ini dapat berdasarkan bukti, data, atau pengetahuan yang ada. Ini juga dinilai dalam skala tertentu, misalnya, skala 1 hingga 10.

- Effort (Usaha): Ini mengukur perkiraan usaha yang dibutuhkan untuk menyelesaikan proyek atau fitur tersebut. Ini bisa berupa waktu, sumber daya, atau kerumitan teknis. Juga dinilai dalam skala tertentu, misalnya, skala 1 paling sulit hingga 10 paling mudah.

Setelah ketiga faktor ini dinilai, skor ICE dihitung dengan rumus sederhana: Impact × Confidence × Effort. Semakin tinggi skor ICE, semakin tinggi prioritasnya.


Gambar 27. ICE Framework Prioritization

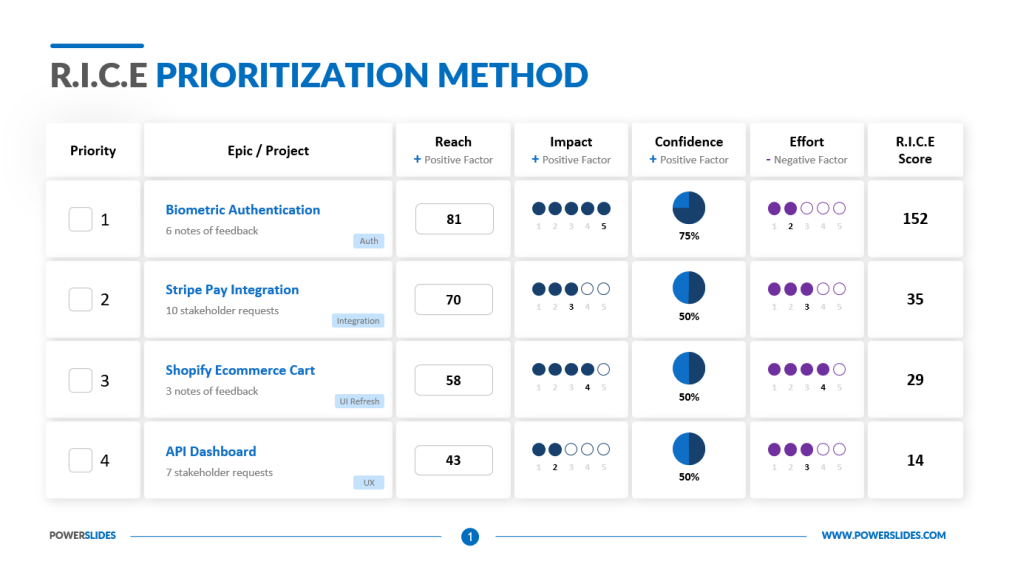
b. RICE Framework

RICE adalah singkatan dari Reach, Impact, Confidence, dan Effort. Ini adalah metode yang lebih lengkap dibandingkan dengan ICE dan mencakup empat faktor kunci:

- Reach (Jangkauan): Ini mengukur sejauh mana proyek atau fitur tersebut akan mencapai pengguna atau pemangku kepentingan. Ini sering diukur dalam jumlah pengguna yang akan terpengaruh atau dalam persentase dari total pengguna.

- Impact (Dampak): Seperti dalam ICE, ini mengukur sejauh mana proyek atau fitur tersebut akan berdampak pada pengguna, bisnis, atau tujuan yang ingin dicapai.
- Confidence (Keyakinan): Seperti dalam ICE, ini mengukur sejauh mana tim memiliki keyakinan bahwa proyek atau fitur tersebut akan berhasil.
- Effort (Usaha): Seperti dalam ICE, ini mengukur perkiraan usaha yang dibutuhkan untuk menyelesaikan proyek atau fitur tersebut.

Skor RICE dihitung dengan rumus: (Reach × Impact × Confidence) / Effort. Seperti dalam ICE, semakin tinggi skor RICE, semakin tinggi prioritasnya.




Gambar 28. RICE Framework Prioritization

c. Moscow Technique

Metode Moscow adalah pendekatan yang berguna untuk mengelompokkan solusi berdasarkan tingkat prioritas dan ketersediaan:

- Must Have (Harus Ada): Solusi yang mutlak diperlukan dan esensial bagi keseluruhan fungsi produk. Kehadirannya adalah faktor kunci dalam keberhasilan produk.

- Should Have (Sebaiknya Ada): Solusi yang penting, tetapi mungkin bukan elemen kritis. Tanpa solusi ini, produk masih dapat berfungsi, tetapi dengan solusi ini, pengalaman pengguna ditingkatkan.
- Could Have (Bisa Ada): Solusi tambahan yang mungkin akan meningkatkan produk, tetapi keberadaannya tidak kritis. Solusi ini bisa diimplementasikan jika waktu dan sumber daya memungkinkan.
- Won't Have (at this time) (Tidak Akan Ada): Solusi yang sengaja ditunda atau tidak akan diimplementasikan dalam waktu dekat. Keputusan ini bisa didasarkan pada keterbatasan sumber daya atau prioritas yang berbeda