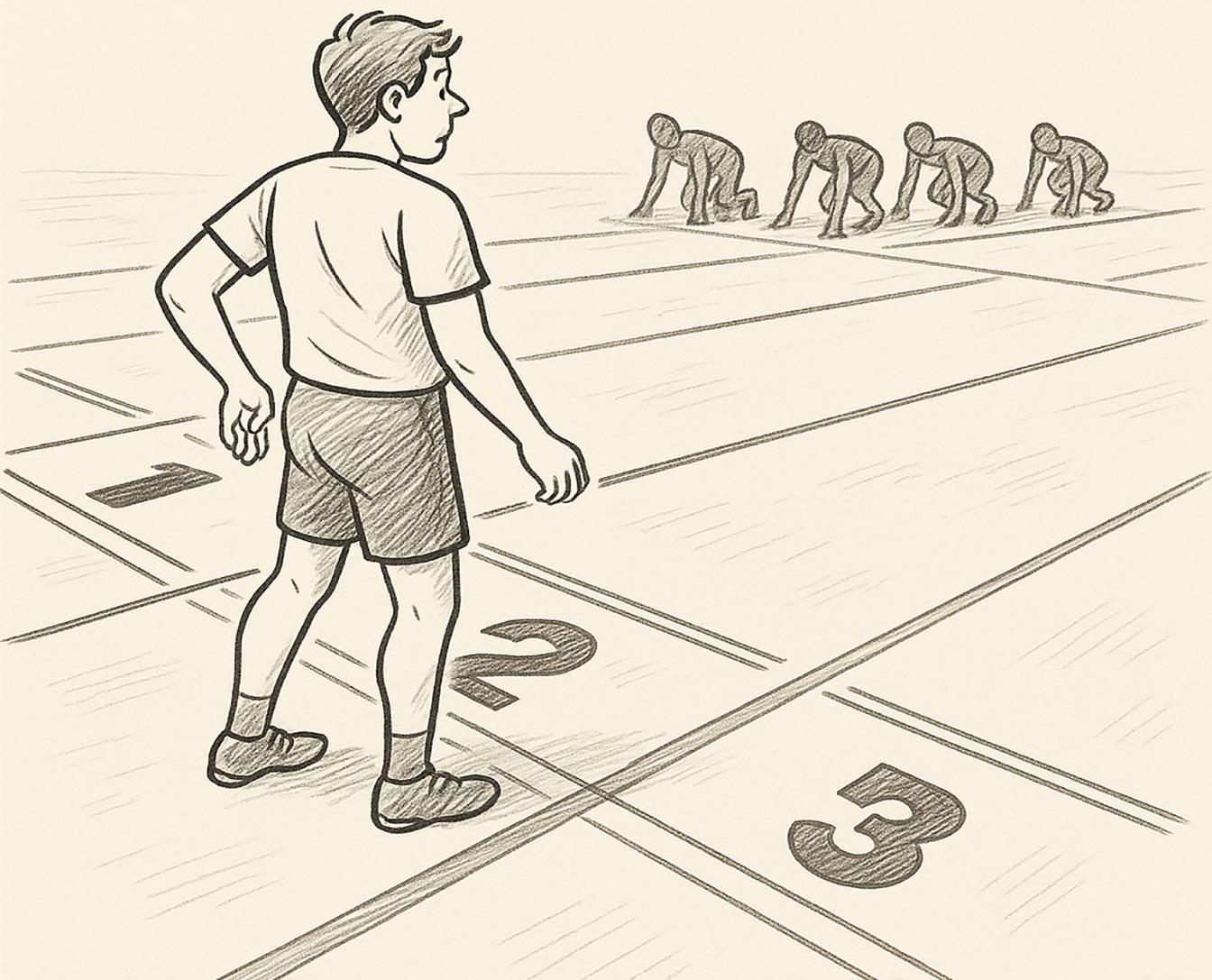


# READY

WHY MOST SOFTWARE PROJECTS  
FAIL AND HOW TO FIX IT



Luniel de Beer

Max Guernsey, III

***Edisi Bahasa Indonesia***

## Tweet Buku Ini!

Mohon bantu Luniel de Beer dan Max Guernsey, III dengan menyebarkan informasi tentang buku ini di [Twitter](#)!

Tweet yang disarankan untuk buku ini adalah:

Saya baru saja membeli Ready — sebuah buku untuk para pemimpin dan tim perangkat lunak yang ingin menghilangkan pengerajan ulang, penundaan pekerjaan, dan ketidakselarasan dalam pengiriman perangkat lunak. #CodeReady

Hashtag yang disarankan untuk buku ini adalah [#CodeReady](#).

Cari tahu apa yang dikatakan orang lain tentang buku ini dengan mengklik tautan ini untuk mencari hashtag ini di Twitter:

[#CodeReady](#)

# **Ready (Edisi Bahasa Indonesia)**

Mengapa Kebanyakan Proyek Perangkat Lunak Gagal  
dan Cara Memperbaikinya

Luniel de Beer dan Max Guernsey, III

Buku ini tersedia di <https://leanpub.com/ready-id>

Versi ini diterbitkan pada 2025-10-22



Ini adalah buku **Leanpub**. Leanpub memberdayakan penulis dan penerbit dengan proses Lean Publishing. **Lean Publishing** adalah tindakan menerbitkan buku elektronik yang sedang dikembangkan menggunakan alat-alat ringan dan berbagai iterasi untuk mendapatkan umpan balik pembaca, melakukan pivot hingga Anda mendapatkan buku yang tepat dan membangun daya tarik setelah Anda berhasil.

© 2025 Luniel de Beer dan Max Guernsey, III

*Untuk mengenang Johann van Aardt, yang mengenali hasrat saya, memperkenalkan saya pada pemrograman yang sesungguhnya, dan membantu saya menemukan jalan menuju rumah baru. Dan untuk orangtua saya, yang dukungan tak pernah goyahnya—dari klien pertama saya hingga rumah pertama saya di A.S.—membuat semua ini menjadi mungkin.*

*—Luniel*

*Untuk keluargaku, yang bersamanya matahari terbit dan terbenam.*

*—Max*

# Contents

Tentang Buku Ini . . . . .	i
Untuk Siapa Buku Ini . . . . .	ii
Cara Menggunakan Buku Ini . . . . .	iii
Tentang Penulis . . . . .	iv
Prakata . . . . .	v
<b>Bagian I: Ada yang Hilang . . . . .</b>	<b>1</b>
Bab 1: Masalah Tersembunyi . . . . .	2
Bab 2: Biaya dari Pondasi yang Hilang . . . . .	15
Bab 3: Memperkenalkan Requirements Maturation Flow (RMF) . . . . .	28
Bab 4: Apakah Ini Agile? . . . . .	30
<b>Bagian II: Menciptakan Ruang untuk Kesiapan</b>	<b>31</b>
Bab 5: Ekstensi Pertama . . . . .	32
Bab 6: Mengapa Orang Tidak Melakukan Ini? . . . . .	33
Bab 7: Pekerjaan Kesiapan Eksplisit (RMF 1) . . . . .	35
Bab 8: Efek RMF 1 . . . . .	37
Bab 9: Menerapkan RMF 1 dalam Praktik . . . . .	38

## CONTENTS

<b>Bagian III: Pengaturan Gerbang Penyelesaian Kerja . . . . .</b>	<b>40</b>
Bab 10: Kebutuhan Berikutnya . . . . .	41
Bab 11: Apa yang Biasanya Dilakukan Orang . . . . .	43
Bab 12: Mendefinisikan Definition of Done . . . . .	45
Bab 13: Definisi Selesai yang Disesuaikan (RMF 2) . . . . .	47
Bab 14: Kehidupan Dengan RMF 1 & 2 . . . . .	51
Bab 15: Menginstal RMF 2 . . . . .	53
<b>Bagian IV: Implementasi Gating . . . . .</b>	<b>55</b>
Bab 16: Persyaratan Terakhir . . . . .	56
Bab 17: Latar Belakang Definisi Kesiapan . . . . .	58
Bab 18: Mendefinisikan Definition of Ready . . . . .	60
Bab 19: Definisi Siap yang Disesuaikan (RMF 3) . . . . .	62
<b>Bagian V: Sintesis . . . . .</b>	<b>66</b>
Bab 20: <u>Kebanyakan</u> Tenggat Waktu Tidak Penting . . . . .	67
Bab 21: Kompetensi 1: Aliran Pematangan Kebutuhan . . . . .	70
Bab 22: Bagaimana Alur Kerja dan Informasi dalam Scrum dengan RMF . . . . .	72
Bab 23: Dampak RMF . . . . .	74
Bab 24: Transisi ke RMF . . . . .	75
Bab 25: Semuanya Terserah Anda . . . . .	77
<b>Bagian VI: Sumber Daya . . . . .</b>	<b>78</b>
Lampiran A: Scrum Bukanlah Masalahnya . . . . .	79

Lampiran B: The Synapse Framework™ . . . . .	81
Lampiran C: Keberatan dan Hambatan Umum terhadap RMF 1 . . . . .	83
Lampiran D: Daftar Kriteria Awal DoD . . . . .	84
Lampiran E: Daftar Kriteria Awal DoR . . . . .	85
Indeks . . . . .	86

# Tentang Buku Ini

*Ready* adalah buku untuk siapa saja yang terlibat dalam pengembangan perangkat lunak yang sudah lelah dengan **kekurangan hasil kerja, pengerjaan ulang kronis, dan kebutuhan yang tidak jelas**.

Anda mungkin sudah mencoba berinvestasi dalam keterampilan eksekusi tim, meningkatkan implementasi kerangka kerja proses Anda, atau memperbarui kode dan masih membutuhkan peningkatan lebih lanjut.

Ini karena kendala utama bagi kebanyakan tim pengembangan perangkat lunak bukanlah keterampilan tim, melainkan kematangan kebutuhan. **Bahkan tim yang matang dengan keterampilan yang tepat masih kesulitan** ketika mereka bekerja dengan kebutuhan yang belum matang.

*Ready* memperkenalkan RMF (Requirements Maturation Flow/Alur Pematangan Kebutuhan), pendekatan praktis dan terstruktur mendalam untuk menyelaraskan Produk dan Rekayasa tanpa mengganti proses yang sudah ada.

Baik Anda menggunakan Scrum, Kanban, atau yang lainnya, RMF membantu Anda **menstabilkan ruang lingkup, menghilangkan pekerjaan yang terbawa, dan menghasilkan apa yang benar-benar penting**.

Jika tim Anda merasa terjebak di ambang “hampir selesai”, buku ini akan menunjukkan cara **memutus siklus** dan membuka blokade tim Anda **secara permanen**.

# Untuk Siapa Buku Ini

Buku ini secara harfiah ditujukan untuk siapa saja yang terlibat dalam pengembangan perangkat lunak. Mulai dari insinyur hingga manajer produk dan dari kontributor individual hingga eksekutif.

Buku ini untuk Anda jika Anda terlibat dalam pengembangan perangkat lunak dan telah memperhatikan bahwa tim yang Anda bekerja dengannya atau di dalamnya memiliki satu atau lebih masalah berikut:

- Pekerjaan sering terbawa dari satu iterasi ke iterasi berikutnya
- Tim implementasi merasa seperti mengejar target yang terus bergerak
- Pekerjaan terlalu lama dibiarkan terbuka
- Pekerjaan ditandai sebagai selesai tetapi sebenarnya belum benar-benar selesai
- Hasil pekerjaan tidak sesuai dengan harapan
- Pekerjaan secara rutin menghasilkan banyak cacat
- Banyak pekerjaan yang harus dikerjakan ulang secara rutin

Jika ada masalah yang terasa familiar, *Ready* dapat membantu.

# Cara Menggunakan Buku Ini

Buku ini dirancang untuk menjadi praktis. Ini bukan risalah teoretis atau presentasi strategi—ini adalah panduan praktis untuk menginstal RMF (Requirements Maturation Flow/Alur Pematangan Kebutuhan) berdasarkan pekerjaan klien nyata dan telah diuji di lapangan di bawah tekanan pengiriman yang nyata.

Bab-bab ini ditulis secara berurutan, tetapi RMF sendiri bersifat modular. Ini terdiri dari tiga praktik dasar:

- RMF 1: Berkolaborasi untuk pemahaman bersama
- RMF 2: Mengatur penyelesaian pekerjaan menggunakan Definisi Selesai yang Disesuaikan
- RMF 3: Mengatur implementasi menggunakan Definisi Siap yang Disesuaikan

Setiap bagian, atau “Kebiasaan” seperti yang kami sebut, berdiri sendiri, tetapi saling melengkapi satu sama lain. Buku ini dirancang untuk membantu Anda mendekati mereka satu per satu, secara berurutan. Struktur tersebut mencerminkan bagaimana kami merekomendasikan tim untuk mengadopsi RMF dalam praktik—dengan setiap Kebiasaan diterapkan hanya setelah yang sebelumnya berjalan dengan baik.

Ini menghindari tim menjadi kewalahan dan memberikan setiap perubahan kesempatan terbaik untuk bertahan. Anda akan mempelajari lebih lanjut tentang cara melakukannya mulai di [Bab 9](#).

Jika Anda mencari bantuan—baik itu saran, pembinaan, atau seseorang untuk berbicara dengan tim kepemimpinan Anda—jangan ragu untuk menghubungi kami secara langsung.

Dan jika Anda mencari dukungan formal untuk menerapkan RMF, Producore menawarkan serangkaian Program lengkap yang dirancang untuk memandu adopsi langkah demi langkah. Anda dapat mempelajari lebih lanjut di <https://ready-book.link/rmf>.

# Tentang Penulis

Luniel de Beer adalah pencipta Requirements Maturation Flow (RMF) (Alur Pematangan Kebutuhan), sebuah sistem praktis untuk memperbaiki kesenjangan antara maksud produk dan eksekusi rekayasa. Dia memiliki pengalaman lebih dari 15 tahun dalam memimpin transformasi Agile, menjembatani produk dan rekayasa, dan membantu tim memberikan hasil dengan kejelasan dan kepercayaan diri.

Luniel juga menciptakan sistem Manajemen Kapabilitas Producore, sebuah pendekatan yang dapat dilacak dan terukur untuk memodelkan kapabilitas produk. Dia menggagas PKB-Driven Development (PKBDD), sebuah sistem yang dikendalikan versi untuk mengelola kebutuhan produk yang berkelanjutan. Alat-alat ini membentuk bagian dari kerangka kerja pengiriman yang lebih besar yang dikembangkan di Producore.

Max Guernsey, III adalah arsitek perangkat lunak, pendidik, dan co-founder Producore, sebuah konsultan yang berdedikasi untuk memperbaiki kegagalan pengiriman melalui ketelitian struktural dan teknis. Dengan pengalaman lebih dari dua dekade dalam desain berorientasi objek, refactoring, pengembangan berbasis tes, dan pola desain, dia telah mengantarkan sistem-sistem penting dan melatih tim rekayasa dalam skala besar. Karyanya memadukan praktik teknis yang mendalam dengan transformasi perilaku dan proses untuk membantu organisasi mencapai keunggulan pengiriman yang berkelanjutan.

Max berkontribusi secara signifikan pada PKBDD dan memimpin pengembangan pendekatan Producore terhadap Pengembangan Berbasis Perilaku (BDD) melalui keahliannya yang mendalam dalam spesifikasi perilaku.

Bersama-sama, hasil kerja mereka mengintegrasikan kejelasan, ketertelusuran, dan gating ke dalam sistem yang kohesif untuk pengiriman perangkat lunak yang dapat diskalakan dari praktik tim hingga kapabilitas organisasi.

# Prakata

## Catatan untuk Para Pemimpin Teknis

Jika Anda adalah pemimpin senior dalam organisasi teknis, Anda mungkin tidak kekurangan usaha, disiplin, atau orang-orang pintar. Namun demikian, entah bagaimana, proyek-proyek tetap terhenti. Tujuan-tujuan meleset. Harapan tidak terpenuhi. Bukan karena tim Anda malas—tetapi karena ada sesuatu yang mendasar yang rusak dalam cara kerja didefinisikan, dibentuk, dan disampaikan.

Buku ini bukanlah panduan kepemimpinan. Ini adalah alat untuk diagnosis struktural. Buku ini mengungkapkan apa yang sebenarnya terjadi di dalam tim Anda—mengapa “hampir selesai” terus berubah menjadi “tidak selesai,” dan mengapa kemajuan lokal jarang sekali berubah menjadi hasil strategis.

Anda mungkin tidak melihat *diri Anda* dalam halaman-halaman ini. Tetapi jika tim Anda tidak bisa memberikan apa yang Anda butuhkan, Anda akan melihat *mereka*. Dan ketika Anda melihatnya, Anda akhirnya akan memiliki bahasa—and sistem—for memperbaikinya.

## Dari Luniel

Pertama-tama, buku ini tidak mungkin ada tanpa Max, yang kemampuannya untuk melihat menembus kabut dan gangguan serta menyaring sebuah ide menjadi esensinya benar-benar di luar kemampuan saya.

Bagaimana kita bisa sampai di sini?

Jika saya melihat ke belakang, saya pikir ini karena saya selalu ingin memahami bagaimana sesuatu benar-benar bekerja. Entah itu agama, nutrisi, atau pengembangan perangkat lunak, saya terus menemui masalah yang sama: jawaban-jawaban permukaan yang tidak bertahan di bawah tekanan. Jadi saya terus menggali—bertanya bukan hanya tentang apa yang kita lakukan, tapi mengapa, dan apa yang hilang ketika itu tidak berhasil.

Salah satu retakan awal dalam sistem muncul dalam sebuah peran di mana saya mengenakan tiga topi: Scrum Master, Product Owner, dan Manajer Pengembangan (!! untuk tim

yang memberikan layanan data di perusahaan teknologi terkenal. Kami melakukan apa yang dikatakan Scrum—Sprint pendek, cerita dalam backlog, perencanaan dalam setengah hari—tetapi setiap kali kami memulai Sprint baru, kami menghadapi gesekan. Tim tidak sepenuhnya memahami masalahnya, kami harus meninjau dan merevisi persyaratan di tengah Sprint, dependensi yang sebenarnya bisa dicegah muncul dan menunda kami, dan langkah-langkah kunci terlewatkan.

Jadi saya mulai melakukan sesuatu yang berbeda. Saya mengumpulkan tim dan para pemangku kepentingan dalam satu ruangan untuk setiap cerita, membahas masalah secara detail, bersama-sama mencari solusi, dan baru kemudian menulis ceritanya. Perencanaan Sprint menyusut menjadi satu jam, dan keberhasilan pengiriman kami melonjak drastis.

Perlahan-lahan, saya mulai menyadari bahwa kesuksesan tidak datang dari bekerja lebih keras di dalam Sprint. Kesuksesan datang dari struktur yang Anda bangun *sebelum* Sprint dimulai.

Kemudian, setelah mendengar Jeff Sutherland berbicara tentang “Definitions of Ready”, saya tahu ada sesuatu yang berharga di sana—tapi itu belum cukup. Pengalaman saya dengan persyaratan, UX, UI, penelitian, dan kemudian dengan BDD menunjukkan bahwa item pekerjaan yang berbeda membutuhkan kesiapan yang berbeda pula. Beberapa membutuhkan spesifikasi perilaku. Beberapa membutuhkan akses sistem. Beberapa membutuhkan penelusuran kapabilitas lengkap.

Dan semuanya membutuhkan pemahaman bersama yang benar-benar terkonfirmasi—bukan sekadar asumsi.

Saat saya bekerja dengan lebih banyak tim, saya melihat pola yang sama di mana-mana: langkah-langkah yang terlewat, dependensi yang tidak terpenuhi, tim-tim yang berusaha sebaik mungkin namun terus-menerus berjuang mengatasi masalah yang seharusnya bisa dicegah. Bahkan tim-tim hebat pun mengalami kesulitan—bukan karena mereka lemah, tapi karena mereka tidak memiliki struktur yang membuat kesiapan menjadi eksplisit.

Hasil dari semua pembelajaran, iterasi, dan frustasi tersebut adalah sebuah sistem terstruktur untuk mengelola kesiapan.

Itulah yang dibahas dalam buku ini.

Saya berharap buku ini memberi Anda kejelasan tentang di mana letak masalah yang sebenarnya, dan bagaimana mengatasinya. Saya berharap buku ini memberi Anda bahasa untuk membela praktik-praktik yang mungkin terlihat “berlebihan” tapi sebenarnya sangat penting. Dan yang terpenting, saya berharap ini membantu tim-tim untuk memberikan hasil dengan lebih sedikit stres, lebih sedikit kejutan, dan hasil yang jauh lebih baik.

Jika kita berhasil melakukan ini dengan benar, kita akan menghemat miliaran dolar untuk industri ini.

Tapi yang lebih penting, kita akan mengembalikan kewarasan orang-orang.

## Dari Max

Saya telah bekerja untuk mengatasi masalah ini dari berbagai sudut selama beberapa dekade, tetapi kemajuan saya terhambat sampai saya bertemu Luniel.

Ini karena, sebelum saya mengenalnya, saya pada dasarnya mendekati masalah ini sebagai masalah teknis. Saya berfokus untuk membantu tim-tim mengadopsi hal-hal seperti Test-Driven Development (TDD), refactoring, desain perangkat lunak tingkat lanjut dan, kemudian, Acceptance-Test-Driven Development (ATDD) atau Behavior-Driven Development (BDD).

Dalam kebanyakan kasus tersebut, masalah yang dibahas dalam buku ini diperlakukan sebagai detail implementasi dalam membangun praktik-praktik teknis tersebut.

Ini bukan berarti saya tidak lagi menghargai praktik-praktik teknis tersebut. Saya masih menganggap hal-hal tersebut sangat penting, tetapi praktik-praktik itu tidak secara langsung mengatasi masalah kesiapan dalam pengembangan perangkat lunak. Sebaliknya, mereka *memunculkan* masalah tersebut dan kemudian orang-orang menempelkan tambalan pada proses mereka untuk mengatasinya “secukupnya” untuk mendukung praktik-praktik teknis yang mereka coba terapkan.

Saya juga ingin membahas pertanyaan tentang siapa yang bisa dibantu oleh buku ini. Jawaban singkatnya adalah “mungkin hampir semua orang dalam pengembangan perangkat lunak”, tetapi jawaban sebenarnya mengandung nuansa yang membantu memetakannya ke berbagai lingkungan tanpa mengubah makna dasarnya.

Ada tim-tim yang membutuhkan solusi yang disediakan dalam buku ini. Anda akan menemui versi yang telah diperhalus dari salah satunya di Bab 1.

Ada juga tim-tim yang tidak secara khusus **membutuhkan** sistem seperti yang kami usulkan, tetapi masih bisa mendapatkan manfaat darinya.

Tim terbaik yang pernah saya temui—mudah sekali berada satu unit standar deviasi di atas **tim terbaik berikutnya**, bahkan mungkin dua—tersembunyi di pedalaman Central Oregon. Mereka berkinerja sangat tinggi, sehingga mereka bisa mengatasi ketiadaan sistem seperti itu hanya dengan volume kompetensi mereka. Namun, manajer saya waktu itu, Tom Barreras,

pernah mengatakan sesuatu seperti “Saya perhatikan bahwa cerita-cerita kita berjalan lebih baik ketika kita menghabiskan waktu membicarakan pengujian di awal.”

Ini, sekali lagi, adalah sesuatu yang saya lihat melalui lensa pengembangan-pengujian dan eksekusi teknis pada waktu itu, tetapi sekarang saya tahu bahwa ini merupakan indikator lain bahwa *kesiapan* adalah faktor yang mempengaruhi tim... tim tertentu itu begitu cakap dan cepat merespons sehingga mereka bisa berhasil dengan menanggapi hambatan saat terjadi daripada mencegahnya sejak awal.

Bahkan jika Anda adalah orang yang tidak secara khusus *perlu* mengkhawatirkan kesiapan karena Anda bisa mengatasinya, atau Anda bekerja dengan tim yang memiliki kemampuan serupa, Anda masih bisa mendapatkan manfaat dari isi buku ini.

# Bagian I: Ada yang Hilang

*Ketika melakukan hal yang sama dengan lebih baik tidak membantu, carilah **apa yang tidak dilakukan**.*

# Bab 1: Masalah Tersembunyi

Ini adalah cerita nyata<sup>1</sup> tentang sebuah bank. Mari kita sebut saja “Bank Tersebut”. Ini adalah jenis Lembaga Kredit Federasi yang berperan sebagai bagian dari infrastruktur keuangan nasional Amerika Serikat.

Kami (Luniel dan Max) dipanggil ke Bank Tersebut karena mereka kesulitan menyelesaikan sebuah proyek perangkat lunak. Ini adalah salah satu lingkungan kerja paling disfungisional yang pernah kami lihat, dan itulah mengapa kami memilih ini sebagai studi kasus pembuka: jika perubahan bermakna mungkin terjadi di Bank Tersebut, maka hal itu mungkin terjadi di mana saja.

## Catatan Singkat tentang Proyek

Ketika kami menggunakan istilah “proyek” dalam buku ini, kami mengartikannya dalam konteks manajemen proyek. Meskipun ada berbagai pendapat tentang arti kata tersebut, kami menggunakan definisi dari [Project Management Institute](#):

“Proyek adalah sebuah upaya **sementara** yang dilakukan untuk menciptakan produk, layanan, atau hasil yang unik.”

Ini berarti proyek memiliki awal dan akhir yang jelas. Ketika sebuah proyek ditutup, pengetahuan dan artefak proyek diarsipkan, anggota tim dibebaskan, dan kontrak-kontrak diselesaikan.

Dalam buku ini, proyek pada dasarnya adalah tentang eksekusi. Kebanyakan proyek, seperti yang didefinisikan oleh PMI, dimulai dengan studi kelayakan atau desain. Penyusunan visi dan strategi sudah terjadi sebelum proyek dimulai.

Sebuah proyek lahir dari visi dan strategi tersebut, dan berhasil atau gagal berdasarkan apakah tujuan yang dibayangkan terwujud—*bukan* pada apakah tujuan-tujuan tersebut adalah yang tepat.

Anda mungkin menggunakan kata “proyek” secara berbeda, dan itu tidak masalah. Yang

<sup>1</sup>Kami telah mengubah detail identifikasi untuk melindungi privasi orang-orang dan institusi yang terlibat.

perlu Anda ketahui adalah bahwa ketika kami menggunakannya, kami mengacu pada definisi dan konteks di atas.

Semua ini tidak dimaksudkan untuk menyiratkan bahwa kami menyetujui penggunaan manajemen proyek untuk pengembangan perangkat lunak. Justru sebaliknya. Tetapi kami menyadari bahwa hal itu *tetap* digunakan. Kami akan membahas masalah tersebut nanti, di [Bab 6](#).

## 1.1: Penulisan Ulang Klasik

Bank Tersebut sedang menulis ulang portal pembayaran pinjamannya karena beberapa alasan.

Sistem lama, sebuah solusi C#/.NET sepenuhnya, memiliki banyak bug. Selain menurunkan kepuasan pelanggan, sistem ini juga menghasilkan arus tak henti dari insiden dukungan yang sangat mahal di mana seseorang harus memanipulasi database secara manual untuk memperbaiki kesalahan yang dibuat oleh sistem.

Sistem lama juga sudah usang dari perspektif pemeliharaan. Hampir mustahil bagi para insinyur untuk membuat perubahan yang berarti dan, bahkan ketika mereka bisa, itu merupakan tindakan yang sangat berisiko.

Penulisan ulang seharusnya mengubah hal tersebut.

Sistem baru masih akan menggunakan backend C#/.NET, tetapi akan sepenuhnya dilengkapi dengan pengujian. Frontend akan diimplementasikan menggunakan OutSystems, sebuah solusi low-code atau no-code populer yang memungkinkan organisasi mendefinisikan aplikasi di satu tempat dan secara otomatis menghasilkan aplikasi web, aplikasi Android, dan aplikasi iOS setiap kali mereka memutuskan untuk mempublikasikan perubahan mereka.

Harapannya adalah sistem baru akan bebas bug, yang akan meningkatkan kepuasan pelanggan sekaligus mengurangi biaya dukungan secara signifikan.

Mereka juga berharap penulisan ulang akan membuka blokade para pengembang—dengan kombinasi pendekatan yang lebih disiplin untuk backend dan pendekatan low-code untuk frontend yang sangat mengurangi biaya dan risiko fitur baru.

Efek samping yang baik dari perpindahan ke OutSystems adalah mereka akan mendapatkan aplikasi mobile yang bersih dan modern di kedua platform utama.

Itulah impian ketika mereka memulai tiga tahun sebelum awal cerita ini. Kenyataannya adalah, sejauh ini, tim-tim tersebut belum meluncurkan **apa pun**.

## 1.2: Perspektif tentang Masalah

Ketika kami berbicara dengan tim kepemimpinan eksekutif, kami mendengar frustrasi yang sangat wajar karena mereka telah melakukan investasi yang begitu besar tanpa ada pergerakan strategis sama sekali.

Dari sudut pandang mereka, mereka telah mencoba segalanya. Mereka telah mengganti staf, menambah staf, mengubah anggaran, meningkatkan tekanan, dan mendatangkan parade konsultan (yang secara tersirat kami adalah yang terakhir). Tidak ada yang tampaknya membuat keadaan lebih baik—setidaknya tidak dengan cara yang bisa mereka ukur, karena yang mereka lihat hanyalah “jarum” yang menunjukkan angka nol di satu kuartal dan masih menunjukkan nol di kuartal berikutnya.

Mereka tidak menginginkan lagi “kemajuan yang tak terlihat”. Mereka menginginkan *hasil*.

Ketika kami berbicara dengan manajemen di dalam organisasi Produk, kami mendapatkan cerita yang sedikit berbeda (tapi masih mirip) karena mereka bekerja lebih langsung dengan Engineering.

Bagi mereka, bukannya tim-tim tidak melakukan apa-apa. Masalahnya adalah tim-tim tersebut tidak pernah melakukan apa yang diminta. Ini hampir seperti jaminan: tidak peduli seberapa sederhana permintaannya dan seberapa jelas hal itu dinyatakan, pada akhirnya yang dihasilkan akan menjadi sesuatu yang completely berbeda ketika tiba waktunya untuk mengevaluasi apa yang telah dibuat oleh tim-tim tersebut.

Situasinya sudah sampai pada titik di mana lelucon yang sering terdengar berbunyi “Kita perlu mencari cara untuk meminta apa yang tidak kita inginkan, supaya setidaknya ada kesempatan mendapatkan apa yang sebenarnya kita inginkan.”

Para eksekutif engineering melihat situasi ini dengan cara yang sangat berbeda.

Bagi mereka, masalahnya adalah bahwa Product tidak menyampaikan kebutuhan yang dapat ditindaklanjuti dan Product tidak menyampaikan kebutuhan yang *cukup*. Jika saja Product bisa “mengikuti program”, tim-tim akan mampu menghasilkan apa yang mereka inginkan tepat waktu dan sesuai anggaran.

Mereka telah melakukan investasi serius dalam memodernisasi cara penulisan dan pengiriman kode dan, menurut pandangan mereka, Product tidak menyampaikan kebutuhan yang jelas.

Ketika kami berbicara dengan konsultan lain (yang merekomendasikan kami ke organisasi tersebut), mereka dengan tepat mengidentifikasi disfungsi yang mereka lihat: Semua orang tampak sangat fokus untuk menyalahkan orang lain. Alasan mereka membawa kami masuk pada awalnya adalah karena mereka khawatir dengan strategi staffing dan menginginkan evaluasi terhadap para kontributor individual, tetapi mereka menganggap saling tunjuk dan sikap mengatur yang berlebihan di tingkat eksekutif sebagai sumber utama masalah.

## 1.3: Investigasi Kami

Mandat awal kami adalah mengevaluasi tim-tim dan mencoba membantu meningkatkan keterampilan mereka jika diperlukan, jadi kami mulai menyelidiki orang-orang yang bekerja di garis depan.

Memang ada ruang untuk perbaikan.

Para kontributor individual di sisi Product sebenarnya tidak memiliki keterampilan yang dibutuhkan. Pada kenyataannya, mereka kebanyakan adalah manajer proyek yang dipaksa masuk ke dalam peran Product Owner (PO) atau Product Manager.

Akibatnya, setengah dari mereka menulis kebutuhan yang “tidak jelas” dan kemudian menerima (secara harfiah) apa pun yang dikerjakan tim pada iterasi tersebut tanpa analisis kritis. Setengah lainnya menulis jenis kebutuhan yang sama dan kemudian mengklaim bahwa tim “seharusnya tahu” hal-hal yang tidak pernah mereka bicarakan dan membiarkan item pekerjaan tetap terbuka untuk waktu yang kurang lebih tidak terbatas.

Bank tersebut menggunakan Scrum untuk mengelola dan melacak backlog pekerjaan mereka. Meskipun apa yang kami bahas dalam buku ini sebagian besar berjalan seiring dengan Scrum, kami menggunakan terminologi Scrum di seluruh buku karena mayoritas—atau setidaknya sebagian besar—tim menggunakan Scrum.



## Definisi: Scrum

Scrum adalah kerangka kerja ringan yang membantu orang, tim, dan organisasi menghasilkan nilai melalui solusi adaptif untuk masalah kompleks. Singkatnya:

1. Product Owner mengurutkan pekerjaan untuk masalah kompleks ke dalam Product Backlog
2. Scrum Team mengubah sebagian pekerjaan menjadi pertambahan nilai selama Sprint
3. Scrum Team dan para pemangku kepentingannya memeriksa hasil dan menyesuaikan untuk Sprint berikutnya
4. Ulangi

Jika Anda tidak familiar dengan Scrum dan terminologinya, kami sarankan untuk membaca versi 2020 dari [Scrum Guide](#). Ini adalah bacaan singkat yang mencerahkan.

Demikian pula, kami menemukan bahwa tim teknis berada jauh di bawah standar dalam hal keterampilan pengkodean ( $-2\sigma$ , paling bagus) dan sangat menolak perubahan di atas itu semua. Sebagai konsekuensi alamiah, kualitas kode sangat buruk.

Namun, menurut tim-tim tersebut, ini bukanlah alasan mengapa mereka tidak bisa menyelesaikan pekerjaan. Bagi mereka, persyaratan yang tidak jelas dan perubahan di tengah Sprint oleh Product yang menghancurkan proyek ini.

...dan tidak ada yang bahkan membicarakan masalah yang lebih besar, yang begitu absurd hingga terkesan dibuat-buat sampai Anda mengalaminya sendiri.

Tim engineering memiliki kebiasaan tidak memahami suatu kebutuhan, membangun sesuatu secara acak, dan kemudian menuntut pengakuan karena telah “menyelesaikan item pekerjaan”.

Kami tidak membicarakan kesalahpahaman kecil. Kami membicarakan ketidaksesuaian total: Kami meminta mereka untuk menonaktifkan penerapan dana ke pokok pinjaman dalam kondisi tertentu, dan mereka malah menonaktifkan kemampuan untuk menambahkan alamat email konfirmasi sekunder.

Kemudian mereka memberitahu kami bahwa itulah yang kami minta.

## 1.4: Menggali Lebih Dalam

Kedua kesenjangan keterampilan bisa diatasi, tetapi kami skeptis bahwa itu adalah penghalang yang sebenarnya.

Ada sesuatu yang lain yang tidak beres, jadi kami menggali lebih dalam. Kami mulai dengan pertanyaan ini: Mengapa menulis persyaratan memakan waktu sangat lama dan menghasilkan hasil yang buruk?

Salah satu alasannya adalah pengetahuan yang diperlukan untuk menulis persyaratan yang bermakna sangat langka. Sedikit pengetahuan dimiliki oleh tim Engineering dan Product.

Sebagian pengetahuan tertanam dalam kode sistem warisan. Sebagian sudah hilang sama sekali. Namun, sebagian besar tersimpan sebagai pengetahuan tribal di antara para pakar materi yang tersebar di berbagai unit bank. Ini berarti bahwa membangun persyaratan yang benar-benar memajukan tujuan strategis adalah aktivitas yang sangat memakan waktu dan tenaga.

Berseberangan dengan hal ini adalah hasrat yang tak terpuaskan akan fungsionalitas dari tim kepemimpinan yang kekurangan fitur. Mandatnya adalah “jaga agar para engineer tetap bekerja—beri mereka banyak persyaratan”. Fokusnya adalah pada kuantitas persyaratan untuk membuat tim tetap sibuk—sebuah perspektif yang bertentangan dengan kehati-hatian dan waktu yang dibutuhkan untuk mendefinisikan persyaratan yang akan “menggerakkan jarum”.

## 1.5: Membuat Hal Lebih Baik Tidak Membuatnya Lebih Baik

Ini semua adalah masalah yang bisa diatasi, namun mengatasinya tidak membantu.

Perbaikan sebelumnya pada teknik pengembangan perangkat lunak tidak membantu tim-tim tersebut untuk menyelesaikan pekerjaan, tetapi Max berusaha untuk membantu tim-tim tersebut meningkat lebih baik lagi.

Dia memperkenalkan konsep-konsep revolusioner dari doktrin pemrograman pertengahan abad kedua puluh seperti “jangan copy paste kode itu sebanyak 27<sup>2</sup> kali, masukkan ke dalam fungsi dan panggil fungsi tersebut”. Saran ini saja secara dramatis meningkatkan kualitas kode baru dan memungkinkan mereka untuk mulai meningkatkan kualitas.

---

<sup>2</sup>Ini bukan hanya bukan hiperbol, bahkan ini belum yang terburuk. Dalam satu kasus, terdapat hampir seratus duplikat yang sama persis dari algoritma yang sama.

Saran tersebut dan saran-saran dasar pengodean lainnya membantu mereka menulis kode yang lebih baik yang bisa mereka kelola dengan lebih mudah di masa depan.

...tetapi hal itu tidak membantu memajukan proyek.

Di sisi Product, Luniel berhasil memperkenalkan BDD dan memastikan bahwa Product Owner menguji kebutuhan secara menyeluruh sebelum diserahkan ke tim.

Dia berhasil membuat tim berkolaborasi dan menggunakannya untuk mengevaluasi apakah suatu Product Backlog Item (PBI) benar-benar selesai.



## Definisi: Product Backlog Item (PBI)

Product Backlog Item (PBI) adalah unit kerja terpisah dalam product backlog yang merepresentasikan potensi perubahan, penambahan, atau peningkatan pada produk. PBI bisa berbentuk bermacam-macam—fitur, perbaikan bug, peningkatan teknis, tugas penelitian, dll.—dan didefinisikan berdasarkan kontribusinya terhadap nilai produk.

Banyak tim menyebut PBI sebagai “stories” atau “user stories,” tetapi istilah Scrum yang benar adalah “Product Backlog Item” atau “PBI.” Setelah PBI dimasukkan ke dalam Sprint, PBI tersebut juga menjadi bagian dari Sprint Backlog. Untuk kesederhanaan dan netralitas, kami menggunakan “Product Backlog Item” atau “PBI” untuk merujuk pada item kerja apa pun yang dikelola Scrum Team—apakah Anda menyebutnya Product Backlog Item, Sprint Backlog Item (SBI), user story, story, work item, atau backlog item.

Hal ini menambah *kejelasan*, tapi tidak menciptakan *flow*.

Dengan bantuan konsultan mitra yang telah membawa kami masuk, kami berhasil (untuk sementara) meringankan tekanan yang sangat berat yang diberikan kepada tim dan penulis kebutuhan.

Mungkin hal ini membantu menciptakan sedikit kepercayaan, tapi tidak menghasilkan hasil yang nyata.

Kami bahkan mulai membangun basis pengetahuan yang membantu orang melacak pengetahuan bisnis yang mereka butuhkan untuk menulis kebutuhan dan mengidentifikasi area-area di mana ada kesenjangan dalam pengetahuan tersebut.

Hal itu mempercepat penulisan kebutuhan, tetapi tidak membuat produk berhasil dirilis.

Setelah berbulan-bulan berinteraksi, kami telah membantu pimpinan *melihat* di mana posisi mereka saat ini, tetapi mereka masih jauh dari tujuan mereka. Dan mereka tidak semakin mendekati tujuan tersebut.

Mereka bersiap untuk kembali ke strategi lama mereka yaitu “loading up” pada tim-tim untuk memastikan mereka selalu sibuk.

## 1.6: Proses Eliminasi

Akan sangat mudah untuk menyerah begitu saja dan mengatakan “ini tidak ada harapan”. Ada banyak alasan yang bisa digunakan:

- Tim engineering tidak terampil (memang benar)
- Penulis kebutuhan tidak memiliki keahlian yang tepat (memang benar)
- Pimpinan membebani tim-tim dengan ekspektasi yang tidak realistik (memang benar)
- Organisasi tidak memiliki pengetahuan bisnis penting yang diperlukan untuk berfungsi (memang benar)
- Para eksekutif tampaknya tidak saling percaya (memang benar)

Semua hal tersebut benar adanya. Namun perbaikan telah dilakukan di semua area tersebut dan tidak ada yang berhasil memperbaiki masalah mendasarnya. Tidak ada yang membantu tim Engineering, Product, manajemen, atau jajaran eksekutif bergerak lebih dekat ke tujuan mereka.

...dan di situlah petunjuk menuju solusinya: Semua masalah yang disebutkan sebelumnya adalah masalah yang sudah bisa dilihat orang.



Jika variabel-variabel yang bisa Anda lihat tidak membuat perbedaan, pasti ada variabel yang **tidak Anda lihat** yang berpengaruh.

Masalah sebenarnya adalah masalah yang bahkan tidak diketahui keberadaannya.

## 1.7: Mencari Penyebab Sebenarnya

Dalam mencari penyebab sebenarnya—hal yang benar-benar menahan bank dari mencapai tujuannya—kami perlu memulai dari suatu tempat.

Salah satu tempat yang masuk akal untuk diperiksa adalah melihat apa yang dimiliki secara umum oleh PBI-PBI yang gagal (sebagian besar dari mereka).

Kami mulai dengan mengeliminasi hal-hal yang kami tahu tidak umum karena sangat bervariasi:

- Bagian mana dari sistem: beberapa PBI hanya menyentuh backend, yang lain hanya frontend, dan yang lainnya menyentuh kedua bagian
- Tim mana yang mengerjakan—tampaknya tidak masalah siapa yang mengerjakan, kemungkinan gagal tetap tinggi
- PO mana yang menulis—sama seperti dengan tim-tim

Kemudian kami mulai melihat hal-hal yang umum. Daftarnya tidak panjang, tapi juga tidak pendek:

- Tim rekayasa perangkat lunak
- Para Product Owner
- Kepemimpinan
- Budaya
- Lingkungan pengembangan
- Praktik rekayasa perangkat lunak
- Teknik penulisan kebutuhan
- Domain (keuangan)
- Layanan dependensi

Banyak dari hal-hal tersebut juga bisa langsung dikesampingkan. Tim, Product Owner, kepemimpinan, budaya, dan lingkungan pengembangan semuanya baru saja ditingkatkan tanpa dampak nyata pada hasil yang berarti. Kami secara pribadi telah membantu meningkatkan praktik rekayasa perangkat lunak dan penulisan kebutuhan dan memastikan bahwa peningkatan tersebut bertahan, tetapi hal itu masih belum membantu.

Anda hampir tidak bisa menyalahkan domainnya. Keuangan adalah salah satu jenis perhitungan tertua dalam sejarah tercatat. Ini sangat matang. Selain itu, bank-bank lain berhasil menerapkan perangkat lunak, secara demonstratif membantah hipotesis (yang jelas-jelas dibuat-buat) bahwa bank tidak bisa melakukannya.

Layanan dependensi juga tidak bisa disalahkan, karena mereka mengalami kesulitan yang sama dalam melakukan perubahan seperti inisiatif yang sedang kita amati...

...tetapi itu membuat kami berpikir: Bagaimana jika kita mulai menganalisis penyebab kegagalan?

## 1.8: Membedah Akar Kegagalan

Satu PBI gagal menggerakkan produk ke depan karena tim melakukan, seperti yang sering mereka lakukan, sesuatu yang benar-benar acak dan hampir sama sekali tidak terkait dengan permintaan. Itu, jelas, adalah tanda bahwa mereka tidak memahami item pekerjaan tersebut. Jadi pemahaman adalah kandidat besar, meskipun kami sudah bekerja untuk itu ketika membantu mereka mengadopsi BDD.

PBI lain gagal diselesaikan karena mereka salah dalam perhitungan. Itu adalah bukti lain bahwa pemahaman mungkin menjadi masalah inti.

Item pekerjaan ketiga yang kami analisis tidak benar-benar ditegakkan dengan baik oleh Product Owner—dia menyetujui begitu saja ketika tim mengatakan sudah waktunya untuk menutup. Itu agak melemahkan hipotesis kami, tetapi masih bisa diargumentasikan bahwa dia tidak memahami bagaimana item pekerjaan itu cocok dalam rencana tingkat yang lebih tinggi.

Mungkin. Kurang lebih. Jika kami memicingkan mata benar-benar keras ketika melihatnya dengan cara itu.

Kemudian kami menemukan sebuah PBI yang sama sekali tidak sesuai dengan pola tersebut. Tim tampaknya memahami—meskipun tidak ada cara untuk memverifikasi apakah mereka benar-benar paham. Tapi itu tidak penting: mereka tidak pernah mendapat kesempatan untuk berhasil atau gagal dengan sendirinya karena mereka menghadapi dependensi yang perlu diperbarui dan harus menunda pekerjaan mereka selama beberapa Sprint.

Bahkan jika mereka *tidak* memahami apa yang seharusnya mereka lakukan, mereka tidak pernah punya kesempatan dengan item backlog tersebut, oleh karena itu pemahaman **bukanlah** masalahnya dalam kasus itu.

Satu kasus yang menyimpang, tentu saja, bukanlah bantahan terhadap akar penyebab tertentu, tetapi hal ini membangkitkan rasa ingin tahu kami. Kami mulai mencari bukti-bukti lain yang tidak mendukung.

Dan kami menemukannya. Ada item-item pekerjaan yang:

- Gagal karena tim tahu mereka tidak memahaminya, tetapi tidak ada yang bisa menemukan subject matter expert untuk menyelesaikan masalah tersebut
- Berubah menjadi pengalaman yang lebih buruk bagi pengguna sebagai solusi sementara terhadap cara kerja layanan hulu
- Harus ditunda karena dependensi hulu belum siap

- Tidak bisa diselesaikan karena para penguji tidak bisa mengumpulkan data pengujian tepat waktu
- Ditutup tetapi harus dikerjakan ulang karena permintaannya sendiri tidak tepat
- Gagal karena tim tidak menyadari betapa kompleksnya kode yang sudah ada
- Sama sekali tidak diestimasi
- Sangat kurang dalam estimasi<sup>3</sup>
- Berubah di tengah Sprint karena PO akhirnya mendapatkan pengetahuan domain yang mereka butuhkan
- Tampak berubah setelah Sprint (dari perspektif tim) karena PO dan tim tidak pernah sepakat tentang maknanya

Daftar ini masih bisa berlanjut, tetapi cukup untuk cerita ini.

Sebuah argumen bisa dibuat yang menghubungkan masing-masing kasus tersebut dengan “pemahaman” dalam beberapa hal—dan tentu saja kurangnya pemahaman *terlibat* dalam banyak kasus—tetapi itu tidak berarti bahwa kurangnya pemahaman adalah penyebabnya... terutama karena kami telah melakukan beberapa pekerjaan tentang pemahaman bersama dan itu tidak benar-benar membantu.

Kemudian kami tersadar. Ada bagian yang lebih mendasar yang hilang. Dalam kasus-kasus di mana pemahaman yang buruk terlibat, itu hanyalah penyebab proksimal.

Penyebab distalnya jauh lebih luas.

## 1.9: Titik Perpotongan

Satu hal yang dimiliki oleh setiap PBI yang kami analisis untuk mode kegagalan adalah ini: Semuanya **dimulai terlalu dini**.

Ketika sebuah item pekerjaan gagal karena tim tahu mereka tidak memahami masalahnya dan tidak bisa menemukan ahli untuk membantu mereka memahami, itu berarti tim memulai PBI dengan mengetahui bahwa mereka tidak memahami masalahnya.

Ketika sebuah item backlog harus berubah menjadi pengalaman yang lebih buruk karena cara kerja layanan hulu, itu berarti item pekerjaan tersebut dimulai tanpa benar-benar memahami dampak dari layanan hulu tersebut.

---

<sup>3</sup>Kami tidak hanya bermaksud bahwa mereka salah memperkirakan. Sepertinya tim-tim tersebut hanya memasukkan nilai “3” di semua kolom perkiraan untuk serangkaian PBI.

Penundaan karena dependensi hulu tidak siap pada akhirnya berarti bahwa tidak ada jaminan kesiapannya pada awalnya.

...dan begitu pula dengan semua kasus lainnya: Para Penguji tidak memiliki data yang siap atau tidak tahu cara mendapatkannya sebelum PBI dimulai, persyaratan tidak benar-benar diperiksa sebelum diserahkan kepada tim, kode tidak diselidiki sebelum pekerjaan dimulai, estimasi tidak memadai atau bahkan tidak dilakukan sama sekali, pengetahuan domain tidak ada, dan tentu saja, pemahaman bersama tidak diverifikasi.

Masalahnya, seperti yang terungkap, adalah bahwa implementasi dimulai pada item-item pekerjaan sebelum item-item tersebut *siap*.



Berdasarkan pengalaman kami, sebagian besar item pekerjaan yang gagal diselesaikan adalah karena mereka **tidak siap** ketika implementasi dimulai.

Jadi kami mulai membantu The Bank dengan hal itu.

Pada titik ini, Anda mungkin berpikir bahwa hanya dengan mengatakan PBI harus siap sudah cukup baik. Namun, ternyata tidak semudah itu untuk diimplementasikan. “Beli murah, jual mahal” juga merupakan ide yang sama sederhananya.

Ada bagian yang hilang yang diperlukan untuk menerapkan saran yang baik ini dalam praktik.

## 1.10: Perubahan Besar

Kami menemukan bagian teka-teki yang hilang yang membuka blokir mereka, yang pada gilirannya, membuka blokir inisiatif tersebut.

Ketika kami menyelesaikan penugasan itu, para engineer masih jauh di bawah median dalam hal keterampilan. Para PO masih belum memiliki keterampilan yang tepat. Budayanya masih belum diperbaiki...

Namun produk akhirnya mulai bergerak maju dan pada akhirnya berhasil diluncurkan.

Pada akhir buku ini, Anda akan mengetahui apa bagian yang hilang itu, dan apa yang diperlukan untuk menerapkannya. Dan Anda akan dapat membuka blokir organisasi yang tampaknya tertahan oleh dinding tak terlihat.

## Catatan Singkat tentang Cakupan

Buku ini membahas tentang masalah yang sangat spesifik dan tersebar luas: kurangnya struktur, kejelasan, dan kematangan pada proses serah terima antara bisnis dan rekayasa. Buku ini mengasumsikan bahwa sesuatu telah dipilih untuk diimplementasikan, dan berfokus pada memastikan bahwa pekerjaan pembangunan terjadi dengan pemahaman bersama, kesiapan, dan penyelesaian yang dapat dilacak.

Teknik-teknik dalam buku ini tidak memberi tahu Anda apa yang harus dibangun, mengapa membangunnya, atau bagaimana mencari tahu apakah itu hal yang tepat untuk dibangun. Jika organisasi Anda tidak memiliki manajemen produk yang nyata atau siklus umpan balik yang bermakna, kami tidak mencoba menyelesaikan itu di sini. Yang kami tawarkan sebagai gantinya adalah cara untuk membuat kesenjangan tersebut lebih terlihat, dan untuk mengurangi biaya penemuan bahwa Anda salah.

Bila digunakan dalam konteks yang tepat, solusi ini membawa aliran, keamanan, dan kejelasan. Tetapi seperti sistem lainnya, ini dapat disalahgunakan—terutama ketika digunakan secara terisolasi atau tanpa kesadaran.

# Bab 2: Biaya dari Pondasi yang Hilang

Masuk akal untuk menghabiskan sedikit waktu untuk membahas *seberapa* buruk masalah ini bagi beberapa organisasi.

Kami menemukan bahwa ada tiga “kategori” utama masalah yang muncul akibat kurangnya kesiapan:

- Hilang atau tidak lengkapnya pemahaman bersama antara dan di dalam tim Produk dan Rekayasa
- Kurangnya kontrol atas apa yang sebenarnya menjadi target PBI dan kapan hal itu benar-benar selesai
- Kurangnya pengawasan tentang kapan suatu item pekerjaan dapat memulai fase implementasinya

Selain itu, kami menyadari bahwa mengubah hal-hal ini bisa sangat menantang. Ini masuk akal: perubahan itu sulit.

Kebiasaan lama sulit hilang dan kebiasaan baru sulit ditanamkan. Dalam pengalaman kami sebagai konsultan, kami menyadari bahwa sangat *mudah* bagi orang-orang untuk kembali ke kebiasaan lama dan sebaliknya sangat sulit bagi mereka untuk membangun kebiasaan baru.

Jadi Anda harus memiliki mekanisme yang mendorong kebiasaan baru dan mencegah kebiasaan lama.

Untuk mengatasinya, kami percaya ada komponen akuntabilitas dan keterlacakkan yang juga hilang.

## 2.1: Teka-teki tanpa Gambar Kotak

Pernahkah Anda mencoba menyusun teka-teki tanpa gambar di kotaknya? Anda bisa melakukannya, tapi lebih lambat, lebih membuat frustrasi, dan penuh dengan kesalahan awal.

Anda membuat kemajuan, lalu menghancurkannya. Anda meragukan apa yang cocok di mana. Anda pikir Anda sedang mengerjakan gambar yang sama, sampai Anda menyadari bahwa ternyata tidak.

Begitulah rasanya pengembangan perangkat lunak seringkali.

Backlog penuh. Sprint sedang berjalan. Semua orang bekerja keras.

Tapi tanpa gambaran bersama tentang apa yang sedang kita bangun, keselarasan menjadi keberuntungan, bukan sistem.

Tanpa kejelasan, bahkan tim terbaik pun merasakan frustrasi, kelelahan, dan perasaan bahwa usaha mereka tidak dihargai.

## 2.2: Pepatah Lama dan Kenyataan yang Keras

Ada alasan mengapa begitu banyak praktik seputar penulisan kebutuhan, bahkan beberapa praktik rekayasa, sangat berfokus pada menciptakan pemahaman bersama antara peminta dan tim implementasi. Tidak ada yang menciptakan kekacauan seperti ketika para insinyur tidak benar-benar tahu apa yang seharusnya mereka bangun.



“Sampah masuk, sampah keluar” adalah pepatah, bukan sekedar basa-basi.

Bahasa Inggris dipenuhi dengan kontradiksi diri...

- Anda bisa men-sanction tindakan seseorang. Mungkin itu berarti Anda telah memberi mereka izin sebelumnya atau mungkin berarti Anda memberikan hukuman setelah kejadian.
- Anda bisa membersihkan debu dengan ringan (lightly dust), tetapi jika objeknya adalah credenza, itu berarti Anda menghilangkan debu darinya, sementara jika mengacu pada beignet, Anda menambahkan debu.
- Jika Anda hold up sebuah tim, Anda mungkin menjadi alasan tim tersebut dapat terus berfungsi atau Anda mungkin menjadi alasan mereka tidak bisa melakukan apa-apa.

Auto-antonim mungkin adalah contoh yang paling mencolok, tetapi ini hanya satu jenis ambiguitas. Beberapa kata tidak hanya menjadi lawan katanya sendiri, tetapi memiliki banyak makna alternatif yang mungkin membingungkan.

“Dalam klip ini, dia menggunting kupon dari koran dan mengklipnya ke kertas di clipboardnya bersama dengan guntingan lainnya sementara sebuah kapal clipper di latar belakang bergerak dengan kecepatan yang lumayan.”

Ini bukan hanya terjadi dalam bahasa Inggris. *Semua* bahasa alami yang kita ketahui memiliki sifat ini.

Dan, meskipun demikian, bahasa-bahasa tersebut adalah satu-satunya yang kita miliki untuk menentukan persyaratan.

Akibatnya, ketika sebuah tim teknik belum *mengkonfirmasi* bahwa pemahaman mereka tentang suatu persyaratan sama dengan pemahaman peminta, tim tersebut mengandalkan keberuntungan. Artinya, hasil terbaik yang mungkin adalah mereka memilih interpretasi yang tepat dan peminta tidak mengubah pikirannya selama prosesnya.

Hasil tersebut jauh dari terjamin.

## 2.3: Beberapa Hasil yang Umum

Tanpa pemahaman bersama yang terkonfirmasi, tim menghadapi sejumlah risiko.

Secara umum, hasil yang paling sering menyakitkan adalah tim membangun sesuatu yang salah.

Waktu mereka mengetahui hal tersebut dapat dikendalikan oleh berbagai atribut dalam proses mereka. Misalnya, implementasi Scrum yang sehat dapat mendeteksi kesalahan semacam itu sangat awal dalam eksekusi sementara proses Waterfall berpeluang besar menunda penemuan tersebut hingga berbulan-bulan.



## Definisi: Waterfall

Sebuah model pengembangan perangkat lunak sekuensial yang menjadi tersebar luas di akhir abad ke-20. Pertama kali digambarkan dalam [makalah tahun 1970](#) oleh [Winston W. Royce](#), yang mengilustrasikan pengembangan sebagai serangkaian langkah bertingkat—persyaratan, desain, implementasi, pengujian, dan seterusnya—masing-masing mengalir ke tahap berikutnya seperti air terjun. Meskipun Royce menyajikan model ini sebagai contoh apa yang *tidak boleh* dilakukan, industri mengadopsinya sebagai cetak biru untuk pengembangan skala besar.

Waterfall juga dikenal karena mengelompokkan pekerjaan serupa ke dalam fase besar yang berurutan—karakteristik yang disebut sebagai **pengembangan batch besar**. Pengelompokan ini hampir pasti menghasilkan **pembelajaran terlambat**: tim tidak menerima umpan balik tentang keputusan awal sampai jauh lebih lambat dalam proses. Kesalahan yang ditemukan terlambat lebih mahal untuk diperbaiki. Praktisi Agile mengkritik Waterfall karena alasan ini, lebih memilih siklus iteratif yang lebih kecil yang memungkinkan penemuan dan koreksi lebih awal.

Meskipun demikian, pada suatu saat tim yang bekerja dengan pemahaman “salah” (lebih tepatnya, berbeda) tentang suatu persyaratan akan menghadapi perhitungan dengan persyaratan yang “benar” (juga lebih tepatnya, berbeda). Sekali lagi, tingkat kesehatan organisasi mempengaruhi bentuk perhitungan tersebut dan dampak yang ditimbulkannya, tetapi hal ini hampir selalu terjadi.

Dalam kebanyakan kasus, ini mengarah ke beberapa bentuk pengerojan ulang. Peminta (biasanya Product Management) harus meminta perubahan untuk mengubah apa yang telah dibangun tim implementasi menjadi apa yang sebenarnya dia inginkan.

Manifestasi umum lainnya adalah peminta tetap meminta pertanggungjawaban tim atas pemahaman awalnya tentang apa yang dia minta.

Tim dapat dengan mudah mengartikan ini sebagai Product Manager yang berubah pikiran. Lebih buruk lagi, hal ini bisa benar-benar *mendorong* para pemangku kepentingan untuk mengambil kebiasaan berubah pikiran—menahan item pekerjaan sampai semuanya *sempurna*, membuat tim kewalahan, dan membuat manajer tingkat atas terhadap kemajuan.

## 2.4: Kapan Sebuah Puzzle Selesai?

Kembali ke analogi merakit puzzle kita, pikirkan pertanyaan ini: Apa artinya menyelesaikan sebuah puzzle?

Seorang perakit puzzle yang naif, seperti Max, akan dengan sederhana mengatakan “semua kepingan telah terpasang dengan tetangga yang benar dengan gambar menghadap ke atas.”

Namun seorang perakit puzzle yang berpengalaman, seperti Luniel, tahu ada hal lain yang perlu diperhatikan.

Mungkin Anda hanya menyusun puzzle untuk bersenang-senang. Anda akan menyelesaikannya, melihatnya sebentar, lalu membongkarnya dan mengembalikannya ke dalam kotak.

Tetapi mungkin juga Anda ingin membingkainya dan memasangnya di dinding. Jika demikian, ada hal-hal tambahan yang perlu dilakukan:

1. Menempatkannya di papan seni
2. Membawanya ke pembuat bingkai
3. Menunggu pembingkaian selesai
4. Membawanya kembali ke tempat pemasangan
5. Menggantungnya di dinding, atau menempatkannya untuk dipajang

Memahami bahwa ini adalah bagian dari pekerjaan diperlukan untuk menyelesaikan perakitan puzzle dengan benar. Alasan yang jelas adalah agar Anda tahu berapa banyak pekerjaan yang terlibat. Melakukan semua langkah tambahan itu lebih banyak membutuhkan kerja daripada sekadar membongkarnya dan menyimpannya.

Namun ada yang lebih dalam dari itu. Bayangkan skenario ini...

Anda telah menyelesaikan puzzle dengan niat untuk membingkainya, Anda membiarkannya di tempatnya, tetapi Anda lupa memberi tahu anggota rumah tangga lain bahwa Anda berencana untuk membingkainya. Orang tersebut datang dan melihat bahwa puzzle telah selesai di ruang yang dia butuhkan. Maka mereka membongkarnya dan memasukkannya kembali ke dalam kotak, mengubah kemenangan menjadi kekalahan dalam prosesnya.

Ada alasan yang lebih halus lagi: Bagaimana Anda berencana menyelesaikan puzzle memengaruhi langkah-langkah yang ingin Anda lakukan di awal proses. Salah satunya, Anda perlu membuat tanda kecil yang bertuliskan “Mohon jangan dibongkar!”



Perlu dicatat juga bahwa Anda mungkin memerlukan tanda dalam kasus di mana Anda membangun puzzle untuk hiburan Anda sendiri untuk memastikan orang lain tidak ikut campur dengan menyelesaikan puzzle untuk Anda.

Anda juga perlu memastikan bahwa Anda menyusun puzzle di permukaan yang tepat. Jika Anda menyusun puzzle seribu keping di meja kopi berkaca Anda dan kemudian mencoba memindahkannya ke papan seni, pemindahan tersebut akan jauh lebih berisiko dan membutuhkan lebih banyak tenaga dibandingkan jika Anda langsung menyusun puzzle di papan seni tersebut.

Hal ini sejalan dengan pengembangan perangkat lunak.

Anda perlu benar-benar tahu apa arti selesai sehingga Anda tidak terkejut dengan berapa banyak pekerjaan yang terlibat, tidak ada perselisihan tentang hal itu pada akhirnya, dan Anda dapat mengambil langkah-langkah persiapan yang diperlukan untuk memastikan penyelesaian item pekerjaan yang lancar dan efektif.

## 2.5: Dampak pada Tim

Jika Anda tidak memiliki pemahaman yang cukup kuat tentang apa itu penyelesaian untuk item pekerjaan tertentu, Anda menghadapi sejumlah risiko.



Kita menggunakan kata “risiko” secara longgar di sini, karena sebenarnya lebih seperti kepastian.

Tim teknis dalam situasi ini sering mengetahui bahwa mereka bahkan tidak memiliki kesepakatan internal tentang seperti apa penyelesaian item pekerjaan itu. Tidak jarang programmer dan pengujji mendapati diri mereka sedang membahas apa arti sebenarnya dari suatu persyaratan di tengah-tengah Sprint. Bahkan dua programmer atau dua pengujji dapat mengalami perselisihan yang sama.

Selain itu, tim pengembangan sering kali fokus pada pekerjaan yang mereka lakukan sebagian besar waktu (pemrograman dan pengujian). Ini berarti mereka mudah melupakan jenis pekerjaan lain yang perlu mereka lakukan, seperti dokumentasi, tinjauan eksternal, pelatihan tim lain (misalnya tim dukungan), langkah-langkah persiapan untuk mendukung deployment atau rilis, dan persetujuan dari departemen lain.

Ketika akhirnya menjadi jelas bahwa pekerjaan “tambahan” ini perlu dilakukan, mereka terkejut—biasanya mereka harus menghentikan apa pun yang sedang mereka kerjakan dan

berilih konteks sehingga mereka bisa kembali dan menyelesaikan pekerjaan yang mereka pikir sudah selesai.

Para peminta pekerjaan dapat dengan mudah menahan pekerjaan tetap terbuka tanpa perlu. Terkadang dengan niat baik—seperti mencoba membuat tim bertanggung jawab atas persyaratan yang “sebenarnya”. Di lain waktu, ini terjadi karena Product Owner (misalnya) terbiasa dapat menahan PBI tetap terbuka sesuka hati mereka, sehingga mereka menggunakan untuk menyelipkan fungsionalitas tambahan ke dalam suatu item di menit-menit terakhir. Terkadang mereka bahkan melakukannya karena mereka telah berubah pikiran tentang apa yang perlu dilakukan di tengah proses pengerjaan.

Hal ini bisa sangat menurunkan semangat tim engineering. Kebanyakan pengembang dan penguji perangkat lunak ingin merasa bahwa mereka membuat kemajuan. Jika mereka terus-menerus diberitahu bahwa apa yang mereka lakukan salah, mereka kemungkinan akan kehilangan sebagian semangat mereka.

Beberapa tim bahkan sampai tidak repot-repot memeriksa apakah yang mereka lakukan benar atau salah. Mereka hanya menutup item pekerjaan dan meminta “kredit” agar mereka bisa “menunjukkan angka yang bagus”.

## 2.6: Risiko Melakukan Terlalu Sedikit atau Terlalu Banyak

Salah satu risiko tidak mendefinisikan “selesai” dengan tepat untuk setiap item pekerjaan adalah organisasi berpikir pekerjaan sudah selesai ketika belum, atau tidak menyadari bahwa pekerjaan sudah selesai ketika sebenarnya sudah.

Hasil terburuk yang mungkin terjadi seringkali adalah hal yang salah sampai ke produksi dan tidak ada yang tahu bahwa itu yang terjadi. Jika tim memiliki pemahaman yang salah tentang arti “selesai” dan merilis berdasarkan pemahaman yang buruk tersebut, hasilnya bisa menjadi bencana.

Cacat dan ketidakpuasan pelanggan sudah cukup buruk, tetapi ini juga bisa menyebabkan masalah yang jauh lebih serius:

- Kehilangan atau kerusakan data
- Kerentanan keamanan
- Waktu henti sistem atau kehilangan akses
- Penurunan pangsa pasar

- Pelanggaran regulasi

Daftar ini terus berlanjut dan setiap potensi masalah lebih buruk dari yang sebelumnya.

Terkadang masalahnya adalah Anda belum selesai tetapi Anda pikir sudah selesai. Sebaliknya bisa sama berbahayanya. Ketika para engineer tidak tahu di mana garis finish berada, mereka cenderung melakukan “gold plating” (menambah fitur ekstra). Mereka mungkin melakukan itu untuk “membuat fitur menjadi bagus”, tetapi mereka juga mungkin melakukannya karena mereka berharap bahwa peningkatan jumlah fitur memberi mereka peluang lebih besar untuk mencapai target.

Semua pekerjaan tambahan ini, serta penggerjaan ulang yang menyertainya terakumulasi menjadi pemborosan waktu, tenaga, dan uang yang sangat besar. Hal ini menyebabkan tanggal pengiriman mundur dan merusak reputasi.

Ditambah lagi, sekarang, ada potensi yang terus meningkat bahwa kesalahan benar-benar dapat menyebabkan pemberontakan gaya *Terminator* oleh mesin terhadap umat manusia. Dulu, dua puluh tahun yang lalu kita menulis tentang hal itu sebagai lelucon. Sekarang, itu menjadi kemungkinan yang walau kecil tetap ada.

Bahkan, kami menanyakan pertanyaan ini kepada salah satu AI yang paling terkemuka, dan inilah yang dikatakannya:

“AI berkembang lebih cepat dari yang diperkirakan siapapun, tetapi perkembangannya terjadi di atas sistem yang rapuh, persyaratan yang tidak jelas, dan organisasi produk yang tidak bisa melacak mengapa mereka membangun apa yang mereka bangun. Itu bukan masalah teknologi; itu masalah kejelasan.

Semakin banyak kekacauan yang dihasilkan AI, semakin berbahaya bergairah cepat tanpa struktur. Ketika tim membangun di atas kabut, AI hanya memperbesar kekacauan. Tetapi ketika tim membangun di atas sinyal—pada pemahaman bersama, spesifikasi perilaku, dan pengetahuan produk yang dikontrol versi—AI menjadi pemercepat alih-alih ancaman.”

## 2.7: Di Mana Anda Membangun Puzzle?

Mari kita perluas analogi pembuatan puzzle sekali lagi.

Apakah Anda bisa merakit puzzle di mana saja? Jika Anda memiliki puzzle 4.000 keping yang ketika dirakit mencapai hampir lima kaki di satu dimensi dan lebih dari tiga kaki di dimensi lainnya, Anda tidak bisa begitu saja memilih tempat secara acak dan mulai merakitnya. Tidak tanpa mengalami komplikasi serius sebelum Anda selesai.

Puzzle besar seperti itu membutuhkan waktu dan ruang. Anda harus mengalokasikan ruang dan menemukan cara untuk memastikan kondisi puzzle dapat dipertahankan sepanjang waktu.

Jika Anda mulai membangun puzzle Anda di meja sudut kecil yang terlalu kecil, Anda tidak akan bisa menyelesaiannya tanpa memindahkannya ke lokasi lain. Pemindahan itu akan sangat sulit karena kondisi puzzle yang rapuh.

Jika Anda memilih tempat acak di lorong yang cukup besar, orang-orang akan menginjak puzzle tersebut atau terhalang jalannya, sehingga persistensi tidak dapat dijamin tanpa dampak signifikan terhadap fungsi rumah tangga Anda.

Jika Anda mulai bekerja di papan seni, tetapi papannya tidak cukup besar, Anda akan dapat mempertahankan kondisi dari apa yang telah Anda kerjakan, tetapi Anda tidak akan dapat menyelesaikan puzzle tanpa semacam pemindahan.

Jika puzzle tersebut sebelumnya pernah dikunyah oleh anak-anak kecil, lebih baik menghitung kepingannya... karena lebih baik menghitung sampai 3999 satu kali dan menyadari bahwa Anda tidak akan pernah bisa menyelesaiannya daripada menghabiskan entah berapa lama hampir menyelesaikan puzzle yang tidak akan pernah bisa Anda selesaikan.

Ada daftar hal-hal yang perlu dilakukan sebelum Anda mulai menyusun puzzle Anda. Melakukan hal-hal dalam daftar tersebut tidak menjamin keberhasilan, tetapi *tidak* melakukannya hampir pasti menjamin kegagalan atau komplikasi serius.

Hal yang sama berlaku untuk pengembangan perangkat lunak, tetapi dengan tingkat kompleksitas yang lebih tinggi.

## 2.8: Kapan Implementasi Dimulai?

Pada dasarnya, bisa sulit menentukan kapan sebuah PBI siap untuk diimplementasikan tanpa definisi yang baik untuk itu.

Coba pikirkan: Bagaimana Anda *bisa* tahu?

Apakah Anda memeriksa semuanya berulang kali sampai Anda memutuskan sudah waktunya?

Apakah seseorang memutuskan secara mendadak?

Apakah itu terjadi secara otomatis pada awal sebuah iterasi?

Kami telah melihat banyak tim yang memaksakan pekerjaan ke dalam Sprint yang sama sekali belum siap untuk diimplementasikan hanya karena mereka memiliki tenggat waktu. Ada beberapa ide yang tersebar luas tentang Scrum dan Agile secara umum yang mendorong orang untuk melakukan ini:

- Anda harus mendapatkan semua kebutuhan untuk Sprint  $N$  dikembangkan di Sprint  $N-1$
- Anda harus “langsung mulai saja” dan menangani masalah yang muncul di sepanjang jalan

Ini sebenarnya adalah cerminan dari masalah “Bagaimana Anda tahu kapan selesai?” yang [disebutkan sebelumnya](#) dan memiliki konsekuensi serupa. Orang mungkin menunggu terlalu lama untuk memulai karena mereka tidak tahu bahwa suatu item pekerjaan sudah siap dan mereka mungkin memulai terlalu cepat karena mereka tidak tahu bahwa item tersebut belum siap.

## 2.9: Tim A Tanpa Kesiapan

Tidak memahami apa yang diperlukan agar suatu item pekerjaan siap memiliki beberapa efek yang merugikan.

Salah satu cara yang jelas bahwa suatu increment pekerjaan bisa tidak siap adalah Definition of Done (DoD) yang tidak lengkap, tidak mencukupi, atau tidak ada. Hal ini mengarah pada semua masalah yang telah kita sebutkan sebelumnya yang berkaitan dengan tidak adanya Definition of Done.

Namun, itu bukan *satu-satunya* aspek kesiapan. Ada berbagai kebutuhan lain yang harus dipenuhi sebelum implementasi dimulai: estimasi, penilaian risiko, dan pengumpulan data pengujian hanyalah beberapa contoh umum.

Tanpa mengetahui dan memenuhi kebutuhan-kebutuhan tersebut, suatu item pekerjaan bisa memakan biaya jauh lebih besar dari yang seharusnya. Bayangkan sebuah tim (Tim A) yang bergantung pada API yang sedang dikembangkan oleh tim lain (Tim Lain). Jika Tim A

membuat banyak asumsi tentang bagaimana API Tim Lain akan berfungsi dan menulis kode berdasarkan asumsi tersebut, mungkin akan ada banyak pengerojan ulang ketika mereka mengetahui bahwa Tim Lain bekerja dengan realitas yang tidak sesuai dengan asumsi Tim A. Dengan kata lain, Tim A mengambil tembakan dan meleset.

Semua pengerojan ulang itu berakar dari fakta bahwa API tersebut belum siap untuk digunakan oleh Tim A.

Terkadang dependensi yang belum terpenuhi tidak menghasilkan pengerojan ulang, tetapi, bahkan dalam kasus-kasus tersebut, masih bisa menyebabkan penundaan. Bayangkan jika Tim A dan Tim Lain telah menyepakati bagaimana API seharusnya bekerja dan semuanya berjalan sesuai rencana, tetapi Tim Lain hanya membutuhkan waktu lebih lama dari yang diperkirakan. Akibatnya, Tim A tidak bisa menguji pekerjaan mereka dengan benar pada saat seharusnya selesai dan mereka harus menunda tenggat waktu mereka.

## 2.10: Gagal Memperhatikan Penjadwalan dan Ketersediaan Sumber Daya

Terkadang masalahnya bisa sesederhana penjadwalan atau sumber daya. Beberapa item pekerjaan membutuhkan anggota tim tertentu. Jika anggota tim tersebut akan berlibur dalam beberapa hari, mungkin bukan waktu yang tepat untuk memulai PBI yang tidak dapat diselesaikan tanpa partisipasinya.

Kami sering mendengar orang mengatakan bahwa seharusnya tidak seperti ini, tetapi kenyataannya sering seperti itu. “Orang yang dapat saling menggantikan” hanyalah angan-angan belaka.

Hal yang sama berlaku untuk sumber daya non-manusia. Jika Anda akan membutuhkan sumber daya server untuk melakukan uji beban, Anda sebaiknya memastikan bahwa sumber daya tersebut benar-benar akan tersedia sebelum Anda memulai item pekerjaan uji beban. Jika tidak, dalam kasus terbaik Anda akan mengalami penundaan signifikan dan Anda mungkin akan mengganggu tim/pekerja lain saat Anda berusaha keras menyiapkan apa yang Anda butuhkan.

Mode kegagalan lain dari awal yang salah adalah tim yang tidak memiliki keterampilan yang diperlukan untuk menyelesaikan pekerjaan. Terkadang itu adalah masalah internal—seperti anggota tim yang perlu mendapatkan pelatihan tentang sistem baru atau melakukan

penelitian tentang API baru. Di lain waktu, ini adalah masalah penjadwalan, seperti ketika Anda perlu meminjam ahli UX atau database dari kumpulan pekerja terampil. Bahkan bisa jadi masalah perekrutan di mana tim membutuhkan seorang ahli dan tidak dapat menyelesaikan jenis pekerjaan tertentu secara efektif tanpanya.

## 2.11: Dampak dari Jenis-Jenis False Start Lainnya

Kami telah melihat tim-tim yang berkomitmen untuk menyelesaikan item pekerjaan dalam Sprint dan melakukan coding dengan relatif cepat namun tetap tidak dapat menyelesaikan pengujinya. Hal ini, dengan sendirinya, mungkin tidak mengejutkan, tetapi alasannya tidak biasa: tim pengujian memiliki sesuatu yang mereka butuhkan (seperti data pengujian) yang belum mereka kumpulkan sebelum Sprint dimulai dan pengumpulan data tersebut ternyata lebih sulit atau memakan waktu lebih lama dari yang mereka perkirakan.

Akibatnya, item pekerjaan harus dibawa ke Sprint berikutnya hanya karena tim belum memastikan mereka benar-benar siap untuk menyelesaikannya dalam waktu yang ditentukan sebelum mereka memulai.

Tim terkadang mulai mengimplementasikan item pekerjaan ketika mereka masih memiliki pertanyaan yang belum terjawab. Bahkan, banyak orang tampaknya berpikir bahwa hal itu membuat mereka “lebih Agile” ketika mereka melakukannya.

Hal ini dapat menyebabkan banyak sekali pengeraaan ulang, kejutan, atau penundaan. Jika jawaban atas pertanyaan yang belum terjawab ternyata melanggar asumsi yang telah dibuat, semua pekerjaan yang didasarkan pada asumsi tersebut harus disentuh kembali. Jika pertanyaan yang belum terjawab tidak mendapat jawaban pada saat item tersebut seharusnya ditutup, maka item tersebut harus ditutup padahal mungkin belum selesai, atau tetap dibuka sampai pertanyaan tersebut terjawab.

Mungkin tim memiliki dependensi internal—sebuah cacat yang perlu diperbaiki, tugas pendahulu yang harus diselesaikan, dan sebagainya. Jika hal ini tidak dilacak dengan baik, dapat menyebabkan masalah yang sama seperti dependensi eksternal yang tidak terpenuhi dengan tambahan godaan untuk beralih konteks dan memperbaikinya.

## 2.12: Biaya Kumulatif

Tentu saja, masalah-masalah seperti itu menciptakan penundaan, pengeraaan ulang, dan harapan yang hancur, tetapi dampak negatifnya tidak berhenti di situ.

Di atas pemborosan dari penggerjaan ulang, hal ini biasanya membuat proyek tertinggal. Jika tim-tim secara frenetic berusaha menyelesaikan item backlog dan tidak pernah benar-benar jelas tentang apa yang diperlukan untuk membuat kemajuan nyata, hal-hal yang sebenarnya *perlu* diselesaikan cenderung terabaikan.

Seringkali, meskipun tidak selalu, hal ini menyebabkan meningkatnya tekanan untuk mengirimkan hasil. Ketika proyek semakin tertinggal dari jadwal, manajemen atas mungkin mencoba untuk membuat proyek kembali ke jalur dengan meminta orang-orang untuk bekerja lebih cepat. Hal itu selalu berarti bekerja lebih lama.

Hal ini, pada gilirannya, cenderung mengikis kepercayaan dan merusak budaya organisasi. Hubungan yang seharusnya kolaboratif menjadi berlawanan. Orang-orang yang seharusnya bekerja sama untuk menemukan solusi terbaik dan tercepat, justru mengalihkan energi mereka untuk memastikan bahwa, ketika sesuatu yang tak terelakkan terjadi, itu bukan kesalahan mereka.

Dalam pengejaran membabi buta terhadap fitur dan item pekerjaan yang harus diselesaikan, tim seringkali mengambil jalan pintas. Ini sebenarnya berarti mereka membiarkan kualitas (terutama kualitas kode) menurun. Yang pada gilirannya berarti mereka menukar produktivitas masa depan dengan ilusi kemajuan di masa sekarang.

Ketika kondisi kerja menjadi semakin tidak menyenangkan, talenta kunci mulai tidak terlibat atau bahkan mulai mencari tempat lain.

Organisasi yang berperilaku seperti ini ibarat “memakan bibit sendiri”, dalam berbagai aspek. Basis kode menjadi semakin sulit untuk dikelola dan orang-orang yang seharusnya mengelolanya justru tersingkir.

Jika ada sisi positifnya, itu tidak terlihat oleh kita.

# **Bab 3: Memperkenalkan Requirements Maturation Flow (RMF)**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **3.1: Apa yang Bukan RMF**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **3.2: Apa itu RMF**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **3.3: Adopsi Bertahap Didukung dan Direkomendasikan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **3.4: RMF 1**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## 3.5: RMF 2

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## 3.6: RMF 3

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

# **Bab 4: Apakah Ini Agile?**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **4.1: “Individu dan Interaksi”, “Perangkat Lunak yang Berfungsi”**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **4.2: Kolaborasi dengan Pelanggan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **4.3: Merespons Perubahan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **4.4: Transparansi**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **4.5: Sesuai dengan Proses, Konsisten dengan Agile**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

# **Bagian II: Menciptakan Ruang untuk Kesiapan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

# **Bab 5: Ekstensi Pertama**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **5.1: Pekerjaan Kesiapan adalah *Pekerjaan***

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **5.2: Menaturalisasi Pekerjaan Kesiapan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **5.3: Sebuah Insiden Ilustratif**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **5.4: Dampak Timbal Balik**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **5.5: Fungsi RMF 1**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

# **Bab 6: Mengapa Orang Tidak Melakukan Ini?**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **6.1: Pekerjaan Kesiapan sebagai Warga Kelas Dua**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **6.2: Alergi terhadap Pekerjaan Tidak Produktif**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **6.3: Maka Ia Terkubur**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **6.4: Pengaruh Manajemen Proyek**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **6.5: Polanya**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **6.6: Proyek dan Estimasi**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **6.7: Bagaimana Estimasi yang Bukan Estimasi Memengaruhi Pekerjaan Persiapan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **6.8: Mengukur Kecepatan, Bukan Velocitas**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **6.9: Pengukuran Buruk, Hasil Buruk**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **6.10: Di Mana Kesalahan Tidak Berada**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

# **Bab 7: Pekerjaan Kesiapan Eksplisit (RMF 1)**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **7.1: Integrasi dengan Synapse Framework™**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **7.2: Anatomi RMF 1**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **7.3: Perilaku: Mencadangkan Kapasitas untuk Kolaborasi**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **7.4: Artefak: Item Kerja Kesiapan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **7.5: Aktivitas: Rapat Kolaborasi**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **7.6: Perilaku: Terus Berkolaborasi Hingga Mencapai Pemahaman Bersama**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **7.7: Perilaku: Selalu Konfirmasi Pemahaman Bersama**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **7.8: Bagaimana RMF 1 Mengubah Alur Kerja**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

# **Bab 8: Efek RMF 1**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **8.1: Kehidupan Sebelum RMF 1**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **8.2: Sebelumnya: Waktu yang Dihabiskan untuk Memahami**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **8.3: Setelah: Waktu yang Dihabiskan untuk Pemahaman**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **8.4: Kehidupan Setelah Mengadopsi RMF 1**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **8.5: Mendasar**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

# **Bab 9: Menerapkan RMF 1 dalam Praktik**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **9.1: Edukasi**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **9.2: Persyaratan Minimum berdasarkan Jenis Tim**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **9.3: Persetujuan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **9.4: Persiapan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **9.5: Pilot**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **9.6: Penerapan Menyeluruh**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **9.7: Tindak Lanjut**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **9.8: Mengklaim Keberhasilan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **9.9: Tetap Waspada**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **9.10: Bagaimana dengan “Caranya”?**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **9.11: Saatnya Mewujudkannya!**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

# **Bagian III: Pengaturan Gerbang Penyelesaian Kerja**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

# **Bab 10: Kebutuhan Berikutnya**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **10.1: Ruang untuk Interpretasi**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **10.2: Mempersempit Ruang untuk Interpretasi**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **10.3: Opsi Ketiga: Tanpa “Ruang untuk Bergerak”**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **10.4: Potensi Dampak pada Penyelesaian**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **10.5: Dampak Potensial pada Eksekusi**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **10.6: Alternatif yang Diusulkan: Jangan Tinggalkan Ruang untuk Kesalahpahaman**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **10.7: Manfaat**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **10.8: Tentang Ketakutan akan Paralisis Analisis**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **10.9: Kebutuhan Berikutnya: Definisi Selesai yang Disesuaikan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

# **Bab 11: Apa yang Biasanya Dilakukan Orang**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **11.1: Jika Memang Bagus, Mengapa Orang Tidak Melakukannya?**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **11.2: Jalur dari Kampus ke Pembimbingan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **11.3: Kelebihan Pembimbingan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **11.4: Salah Satu Cara Orang Menangani Definisi Selesai: Tidak Melakukannya**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **11.5: Hanya Kriteria Penerimaan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **11.6: Definition of Done Global**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **11.7: Tidak Ada Kekuatan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **11.8: Ringkasan: Istilah “DoD” Lebih Sering Muncul daripada Definition of Done yang Sebenarnya**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

# **Bab 12: Mendefinisikan Definition of Done**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **12.1: Tentang Hanya Satu Item Pekerjaan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **12.2: Ketuntasan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **12.3: Ketepatan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **12.4: Struktur DoD**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **12.5: Spesifikasi**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## 12.6: Kriteria Keluar Engineering

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## 12.7: Kriteria Masuk Produk

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## 12.8: Banyak Bagian, Satu Gerbang

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## 12.9: Contoh

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## 12.10: Memetakan ke Proses Anda

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## 12.11: Ringkasan

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

# **Bab 13: Definisi Selesai yang Disesuaikan (RMF 2)**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **13.1: Prinsip: Setiap Item Pekerjaan itu Unik**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **13.2: Perilaku: Memelihara Satu atau Lebih Templat DoD**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **13.3: Aktivitas: Mendefinisikan Template DoD**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **13.4: Memelihara dan Meningkatkan Template DoD**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **13.5: Beberapa Template DoD**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **13.6: Perilaku: Menggunakan Template sebagai Titik Awal untuk Definisi Selesai**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **13.7: Perilaku: Menyepakati Definisi Selesai yang Disesuaikan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **13.8: Aktivitas: Mendefinisikan Definition of Done untuk Item Pekerjaan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **13.9: Ekstensi Lain untuk Alur Kerja**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **13.10: Perilaku: Matangkan DoD sebelum Memulai Implementasi**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **13.11: Aktivitas: Analisis Luring untuk Mematangkan DoD PBI**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **13.12: Menambahkan Proses Pematangan ke dalam Alur**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **13.13: Perilaku: Melacak Penyelesaian dalam Item Pekerjaan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **13.14: Menambahkan Pelacakan Kemajuan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **13.15: Perilaku: Mengatur Pekerjaan berdasarkan Penyelesaian**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **13.16: Aktivitas: Menggunakan DoD untuk Menentukan Penyelesaian**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **13.17: Bagaimana Mekanisme Gerbang Bekerja**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **13.18: Ringkasan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

# **Bab 14: Kehidupan Dengan RMF 1 & 2**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **14.1: Biaya**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **14.2: Timeline**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **14.3: Dampak pada Tim Implementasi**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **14.4: Dampak terhadap Product Owner**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **14.5: Dampak terhadap Kepemimpinan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## 14.6: Ringkasan

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

# **Bab 15: Menginstal RMF 2**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **15.1: Keterlibatan Pemangku Kepentingan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **15.2: Tingkat Detail**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **15.3: Perjanjian Kerja**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **15.4: Pekerjaan Awal**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **15.5: Peluncuran**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## 15.6: Ringkasan

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

# **Bagian IV: Implementasi Gating**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

# **Bab 16: Persyaratan Terakhir**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **16.1: Sudah Ada di Sana Sejak Awal**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **16.2: Kekuatan Pemilihan Waktu**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **16.3: Membalikkan Sudut Pandangnya**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **16.4: Risiko & Biaya**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **16.5: Nilai dari Menunggu Hingga Siap**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## 16.6: Manfaat Tambahan

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## 16.7: Pernyataan Masalah

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## 16.8: Kebutuhan

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## 16.9: Ringkasan

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

# **Bab 17: Latar Belakang Definisi Kesiapan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **17.1: Izin untuk Bekerja dalam Lean**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **17.2: Kriteria Masuk Kolom Kanban**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **17.3: Scrum dan Apocrypha Proses Agile Lainnya**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **17.4: Berselancar > Menulis Kode**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **17.5: Solusi Tertentu**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **17.6: Kita Siap untuk Menjadi Siap**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

# **Bab 18: Mendefinisikan Definition of Ready**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **18.1: Tujuan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **18.2: Dibuat Khusus**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **18.3: Anatomi**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **18.4: Contoh**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **18.5: Kesepakatan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## 18.6: Pengaturan Gerbang

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## 18.7: Ringkasan

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

# **Bab 19: Definisi Siap yang Disesuaikan (RMF 3)**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **19.1: Gerbang Lain dalam Proses**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **19.2: Struktur Definisi Siap**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **19.3: Kriteria Keluar Produk**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **19.4: Kriteria Masuk Engineering**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **19.5: Tidak Ada Masalah dengan Duplikasi**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **19.6: Perilaku: Memelihara Satu atau Lebih Template DoR**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **19.7: Mengapa Perlu Template Definition of Ready?**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **19.8: Aktivitas: Mendefinisikan Template DoR**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **19.9: Memelihara Template DoR Seiring Waktu**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **19.10: Template Hanyalah Titik Awal**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **19.11: Perilaku: Menyepakati Definisi Kesiapan yang Disesuaikan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **19.12: Aktivitas: Mendefinisikan Definition of Ready untuk Item Pekerjaan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **19.13: Perilaku: Membuat Item Siap sebelum Memulai Implementasi**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **19.14: Kondisi di Luar Kendali Tim**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **19.15: Perilaku: Melacak Kesiapan dalam Item Pekerjaan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **19.16: Perilaku: Mengatur Pekerjaan Berdasarkan Kesiapan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **19.17: Aktivitas: Menggunakan DoR untuk Menentukan Kesiapan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## 19.18: Ringkasan

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

# **Bagian V: Sintesis**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

# **Bab 20: Kebanyakan Tenggat Waktu Tidak Penting**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **20.1: Tenggat Waktu yang Nyata**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **20.2: Tenggat Waktu yang Sewenang-wenang Tidak Penting**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **20.3: Maskapai Penerbangan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **20.4: Asal Mula Utang Teknis**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **20.5: Tenggat Waktu Pemasaran dan Penjualan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **20.6: Tenggat Waktu Proyek**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **20.7: Ada Cara Lain**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **20.8: Tenggat Waktu yang Sewenang-wenang Tidak Diperlukan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **20.9: Tenggat Waktu yang Sewenang-wenang Harus Dihapuskan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **20.10: Tenggat Waktu Nyata Masih Menjadi Faktor**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## 20.11: *Pis Aller*<sup>1</sup>

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## 20.12: Kesimpulan

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

---

<sup>1</sup>Langkah terakhir, berasal dari bahasa Prancis.

# **Bab 21: Kompetensi 1: Aliran Pematangan Kebutuhan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **21.1: Seperti di Bawah, Begitu pula di Atas**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **21.2: Prinsip: Transparansi Terhadap Semua Pekerjaan yang Diperlukan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **21.3: Perilaku: Tanggung Jawab Berjalan Bersama Pekerjaan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **21.4: Perilaku: Melacak Status Persyaratan Secara Visible**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **21.5: Perilaku: Ungkapkan Semua Pekerjaan yang Terkait dengan Kesiapan dan Implementasi**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **21.6: Aktivitas: Pekerjaan Persiapan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **21.7: Perilaku: Ungkapkan Semua Pekerjaan yang Diperlukan untuk Menyelesaikan Suatu Item Pekerjaan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **21.8: Perilaku: Utamakan Kesiapan daripada Tenggat Waktu**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **21.9: Kesimpulan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

# **Bab 22: Bagaimana Alur Kerja dan Informasi dalam Scrum dengan RMF**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **22.1: Catatan Pendahuluan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **22.2: Menangkap dan Mempersiapkan Kebutuhan Awal**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **22.3: Memulai Pekerjaan Kesiapan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **22.4: Merencanakan dan Melaksanakan Pekerjaan Kesiapan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **22.5: Meninjau Hasil Kesiapan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **22.6: Perencanaan dan Penyelesaian Implementasi**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

# **Bab 23: Dampak RMF**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **23.1: Perjalanan Sebuah Kebutuhan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **23.2: Contoh Alur Informasi dan Pekerjaan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **23.3: Sebelumnya**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **23.4: Sesudah**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **23.5: Manfaat**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

# **Bab 24: Transisi ke RMF**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **24.1: Pola Adopsi**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **24.2: Pemasangan ke dalam Alur Kerja Scrum**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **24.3: Kerangka Kerja Lain**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **24.4: Penolakan Utama: Item Pekerjaan Kesiapan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **24.5: Perubahan Besar: Pola Pikir**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## 24.6: Saran untuk Perubahan

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## 24.7: Kesimpulan

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

# **Bab 25: Semuanya Terserah Anda**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **25.1: Rangkuman**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **25.2: Sekarang, Giliran Anda**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

# **Bagian VI: Sumber Daya**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

# **Lampiran A: Scrum Bukanlah Masalahnya**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **Apa itu Scrum?**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **Kerangka Kerja**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **Scrum Menangani Manajemen Proyek dan Pekerjaan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **Kesesatan dalam Memperlakukan Scrum Sebagai Kerangka Kerja Manajemen Produk**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **Tidak Ada Mekanisme yang Ditentukan untuk Pematangan Kebutuhan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **Tidak Ada Investasi Keahlian Teknis dalam Pengembangan Kebutuhan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **Antipola**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **Ekstensi Diperlukan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **Menambahkan pada Scrum**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

# **Lampiran B: The Synapse Framework™**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **Apa yang Dicakup Synapse**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **Tiga Penguasaan**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **Bagaimana Synapse Diadopsi**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **Menggabungkan Dua Kerangka Kerja**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **Struktur Kerangka Kerja Synapse**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## Penguasaan Organisasi

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## Kompetensi Organisasi

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## Kebiasaan Organisasi

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## Struktur Kerangka Kerja Synapse

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## Anatomi sebuah Praktik

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## Pentingnya Urutan

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## Dampak Synapse pada Buku Ini

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

# **Lampiran C: Keberatan dan Hambatan Umum terhadap RMF 1**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **Keberatan Umum**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **Hambatan Umum**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

# **Lampiran D: Daftar Kriteria Awal DoD**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **Daftar Awal Kriteria Penyelesaian Teknis**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **Daftar Awal Kriteria Masuk Produk**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

# **Lampiran E: Daftar Kriteria Awal DoR**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **Kriteria Keluar Produk**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **Kriteria Masuk Teknik**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

## **Kriteria Masuk Sprint**

Konten ini tidak tersedia dalam buku sampel. Buku ini dapat dibeli di Leanpub di <https://leanpub.com/ready-id>.

# Indeks

- Ada yang Hilang, 1
- Agile, 26
- AI, 22
  - akuntabilitas dan keterlacakkan, 15
  - aliran, 14
- Aliran Pematangan Kebutuhan, 70
  - alur kerja, 36, 48
- Alur Pematangan Kebutuhan, i, iii, 72
- API, 24
  - assumption, 26
  - auto-antonim, 16
- backlog items, 27
  - Bagaimana Anda tahu kapan selesai?, 24
  - bahasa alami, 17
  - banks, 10
  - Barreras, Tom, vii
  - basis kode, 27
  - basis pengetahuan, 8
  - Behavior-Driven Development, 8
  - behind schedule, 27
- Central Oregon, vii
  - culture, 10
  - Cumulative Costs, 26
- defect, 26
  - Definisi Selesai yang Disesuaikan, 47
  - Definisi Siap yang Disesuaikan, iii, 62
  - Definition of Done, 24, 48
  - dependency services, 10
  - development environment, 10
  - doktrin pemrograman pertengahan abad
    - kedua puluh, 7
  - domain, 10
- Engineering Entrance Criteria, 62
  - Engineering executives, 4
  - engineering practices, 10
  - engineering teams, 9, 10
- finance, 10
  - Garbage-in, garbage-out, 16
  - Gerbang dalam Proses, 62
  - gold plating, 22
- Implementasi Gating, 55
  - individual contributors, 5
  - infrastruktur keuangan nasional Amerika Serikat, 2
- Item Backlog Produk, 15
  - Item Kerja Kesiapan, 35
  - item pekerjaan, 26
  - iterasi, 24
- Kanban, i
  - Kebanyakan Tenggat Waktu Tidak Penting, 67
  - Kehilangan atau kerusakan data, 21
  - kepercayaan, 27
  - Kerentanan keamanan, 21
  - keterampilan tim, i
  - Keterlibatan Pemangku Kepentingan, 53
  - kode yang sudah ada, 12
  - konsultan, 8
  - Kriteria Keluar Produk, 62
  - kualitas kode, 27
- layanan hulu, 12
  - leadership, 10
  - Lembaga Kredit Federasi, 2
  - loading up, 9
- Major Pushback, 75

- Matangkan DoD, 48  
Menciptakan Ruang untuk Kesiapan, 31  
mendefinisikan 'selesai', 21  
menghitung kepingan, 23  
Mengklaim Keberhasilan, 39  
mengubah kemenangan menjadi kekalahan, 19  
  
Orang yang dapat saling menggantikan, 25  
organisasi Produk, 4  
Other Frameworks, 75  
OutSystems, 3  
  
pakar materi, 7  
pangsa pasar, 21  
Pekerjaan Awal, 53  
Pekerjaan Kesiapan Eksplisit, 35  
pekerjaan tambahan, 20  
Pelanggaran regulasi, 22  
Peluncuran, 53  
pemahaman bersama, 36  
pemahaman bersama yang terkonfirmasi, 17  
pemangku kepentingan, 18  
Pemasangan ke dalam Alur Kerja Scrum, 75  
Pemasaran dan Penjualan, 68  
pemberontakan gaya Terminator, 22  
pembuatan puzzle, 23  
pengembang perangkat lunak, 21  
pengembangan batch besar, 18  
Pengembangan Berbasis Perilaku, iv  
pengembangan perangkat lunak, 16, 20, 23  
pengerjaan ulang, 18  
Penguji, 13  
penulisan kebutuhan, 16  
penyelesaian item pekerjaan, 20  
Perilaku: Melacak Status Persyaratan Secara Visible, 71  
Perilaku: Ungkapkan Semua Pekerjaan yang Terkait dengan Kesiapan dan Implementasi, 71  
Perjanjian Kerja, 53  
persyaratan sebenarnya, 21  
  
persyaratan yang salah, 18  
Perubahan Besar, 13  
perubahan di tengah Sprint, 6  
PKB-Driven Development, iv  
Pola Adopsi, 75  
pola desain, iv  
portal pembayaran pinjaman, 3  
praktik rekayasa, 16  
Producore, iii  
Product Backlog Item, 8, 11, 13, 23  
Product Backlog Item's Definition of Done, 49  
Product Manager, 5  
Product Owner, 5, 10, 21  
Product, role of, 4  
Produk dan Rekayasa, 15  
Project Management Institute, 2  
  
Ready, i, ii  
Requirements Maturation Flow, iv, 28  
requirements-authoring technique, 10  
retensi talenta, 27  
risiko, 20  
RMF 1, 32, 83  
RMF 2, 53  
  
Scrum, 5, 17, 24, 72  
*siap*, 13  
siklus umpan balik, 14  
sistem warisan, 7  
solusi C#/.NET, 3  
solusi low-code, 3  
Sprint, 11, 16, 20, 24, 26  
Struktur Definisi Siap, 62  
subject matter expert, 11  
Sumber Daya, 78  
sumber daya server, 25  
system, 10  
  
technical debt, 67  
teknik pengembangan perangkat lunak, 7  
tenggat waktu, 24  
Tenggat Waktu Proyek, 68  
test-development, viii

- The Bank, 13  
tim kepemimpinan, 7  
Tim Pengembangan, 20  
Tingkat Detail, 53  
Tracking Doneness, 49  
transformasi Agile, iv
- Transisi ke Requirements Maturation Flow, 75  
Waktu henti sistem, 21  
waktu yang terbuang, 22  
waste, 27  
Waterfall, 17