code basicstyle=, frame=single, rulecolor=, breaklines=true, showstringspaces=false, columns=fullflexible, keepspaces=true, framesep=6pt, aboveskip=1em, belowskip=1em, xleftmargin=0pt

out basicstyle=, frame=tb, rulecolor=, breaklines=true, showstringspaces=false, columns=fullflexible, keepspaces=true, framesep=4pt, aboveskip=1em, belowskip=1em

Requirement Engineering
Panduan Praktis untuk Minggu Pertama

Muhammad Riduwan Khafidi
Program Studi Teknik Rekayasa Perangkat Lunak
Politeknik Negeri Batam
4342611034@students.polibatam.ac.id

November 5, 2026

Abstrak

Makalah ini adalah panduan praktis untuk minggu pertama mata kuliah Analisis dan Spesifikasi Kebutuhan Perangkat Lunak. Ia menjelaskan apa itu requirement engineering, mengapa ia penting, masalah umum yang muncul bila diabaikan, serta dua proses requirement yang lazim dipakai di industri. Tujuh visualisasi dan tiga tabel menyertai penjelasan agar konsep abstrak menjadi konkret. Sebuah studi kasus berisi sepuluh langkah menunjukkan artefak yang harus dihasilkan pada setiap tahap. Lembar rujukan cepat di akhir merangkum istilah yang wajib dikuasai sebelum pertemuan berikutnya.

1 Pendahuluan

1.1 Analogi Pembuka

Bayangkan seorang tukang bangunan yang hendak mendirikan rumah. Ia tidak mulai dengan memasang batu bata. Ia duduk bersama pemilik rumah dulu: berapa kamar, di mana dapur, apakah perlu garasi, berapa anggarannya. Percakapan itu bukan gambar rumah. Tapi tanpa percakapan itu, tukang akan membangun kamar mandi di ruang tamu.

Requirement engineering adalah percakapan itu. Ia adalah seni menggali apa yang sebenarnya diinginkan pemilik rumah sebelum satu batu pun diletakkan.

1.2 Mengapa Topik Ini Penting

Perangkat lunak dibuat untuk memenuhi tujuan dan kebutuhan pengguna. Ia tidak boleh dibuat berdasarkan asumsi developer. Kesalahan yang ditemukan pada tahap requirement jauh lebih murah diperbaiki daripada kesalahan yang baru ditemukan setelah sistem diluncurkan. Inilah sebabnya requirement engineering diletakkan di awal seluruh siklus pengembangan.

1.3 Peta Pembahasan

Makalah ini disusun sebagai berikut. Bagian kedua menjelaskan apa itu requirement dan level-levelnya. Bagian ketiga menjelaskan mengapa ia penting dengan kurva biaya perubahan. Bagian keempat memetakan enam masalah umum. Bagian kelima membedakan product requirement dan project requirement. Bagian keenam membedah aktivitas requirement engineering. Bagian ketujuh membandingkan proses Volere dan Agile. Bagian kedelapan menyajikan studi kasus sepuluh langkah. Bagian kesembilan sampai kesebelas merangkum kesalahan umum, lembar rujukan cepat, serta tiga prinsip penutup.

2 Apa Itu Requirement

Requirement adalah spesifikasi tentang apa yang harus diimplementasikan. Ia mendeskripsikan bagaimana sistem harus berperilaku, properti atau atribut apa yang harus dimiliki, dan batasan apa yang membatasi proses pengembangannya. Requirement mencakup dua perspektif sekaligus: perspektif pengguna tentang perilaku sistem, dan perspektif developer tentang karakteristik internal sistem.

Perhatikan Gambar **??**. Sebuah requirement yang baik menyentuh empat lapis: perilaku, properti, atribut, dan batasan.

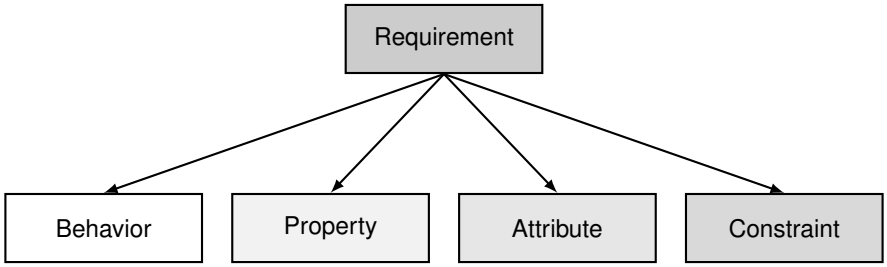


Figure 1. Anatomi requirement yang terdiri dari empat lapis.

Contohnya, pada sistem pemesanan tiket kereta: perilaku adalah “sistem menampilkan jadwal kereta berdasarkan tanggal yang dipilih”; properti adalah “sistem menyimpan riwayat pemesanan selama tiga tahun”; atribut adalah “halaman jadwal memuat paling banyak dua puluh baris per halaman”; dan batasan adalah “sistem harus mematuhi UU Perlindungan Data Pribadi”.

Requirement juga hidup di beberapa level. Perhatikan Gambar **??**. Di level paling atas ada business requirement yang berbicara tentang tujuan sponsor. Di tengah ada user requirement yang berbicara tentang tugas pengguna. Di bawah ada system requirement yang dipecah menjadi fungsional dan non-fungsional. Semakin ke bawah, semakin detail dan semakin teknis.

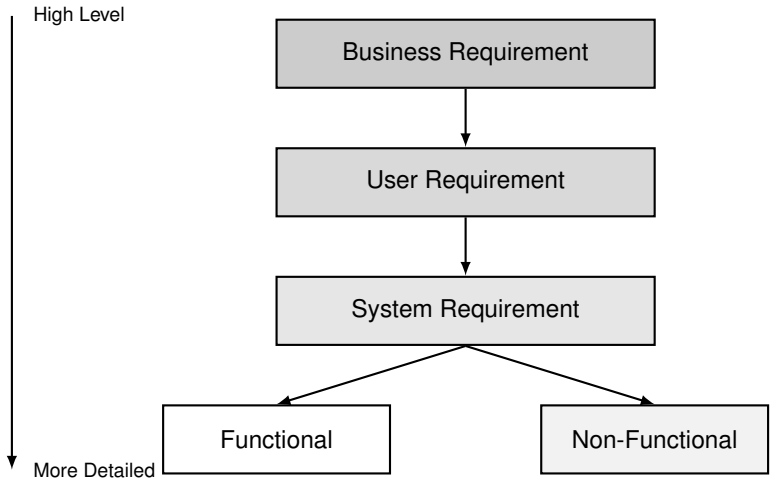


Figure 2. Hierarki level requirement dari high level menuju detail.

Pemetaan level ini penting karena requirement yang hanya menyebut perilaku, tanpa menyebut properti, atribut, dan batasan, hampir pasti akan menghasilkan perdebatan di tengah pengembangan. Ia juga menjelaskan mengapa requirement yang ditulis dengan bahasa sponsor berbeda dari requirement yang ditulis dengan bahasa developer.

3 Mengapa Requirement Engineering Penting

Aturan industri: semakin awal masalah ditemukan, semakin murah biaya memperbaikinya. Perhatikan Gambar **??**. Biaya perubahan naik perlahan di tahap requirement, lalu melonjak tajam di tahap produksi.

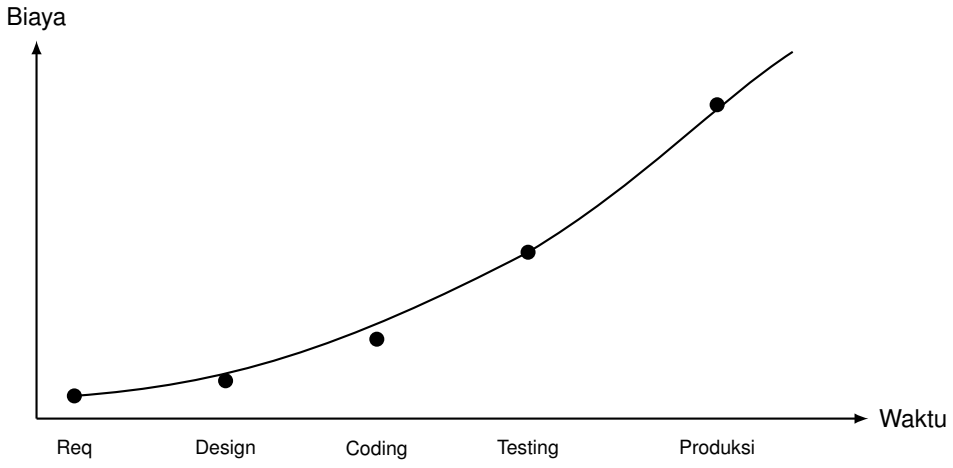


Figure 3. Kurva biaya perubahan sepanjang siklus proyek.

Mengubah kode sumber pada sistem yang sudah diluncurkan jauh lebih sulit dan berisiko daripada mengubah desain sistem di tahap awal. Karena itu, setiap jam yang diinvestasikan untuk menggali requirement di awal akan menghemat puluhan jam di akhir.

4 Masalah Umum pada Proyek Perangkat Lunak

Enam masalah berikut adalah yang paling sering muncul. Perhatikan Gambar **??**.

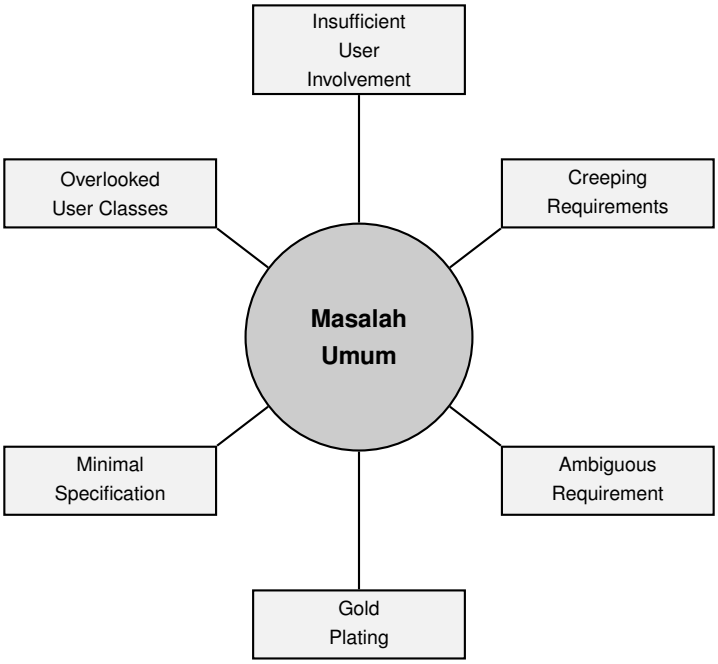


Figure 4. Enam masalah umum dalam proyek perangkat lunak.

Pertama, insufficient user involvement. Kurangnya keterlibatan pengguna mengakibatkan miskomunikasi, yang pada akhirnya menunda jadwal proyek.

Kedua, creeping user requirements. Klien terus menambah dan mengubah requirement bahkan setelah pengembangan dimulai, tanpa memahami kompleksitas dan dampaknya.

Ketiga, ambiguous requirement. Requirement yang ambigu membuat tiap pihak punya tafsir berbeda, sehingga developer bisa membangun solusi untuk masalah yang salah.

Keempat, gold plating. Developer menambahkan fitur ekstra yang tidak diminta, dengan asumsi pengguna akan memakainya. Jika ternyata tidak, seluruh usaha itu terbuang.

Kelima, minimal specification dan perencanaan tidak akurat. Spesifikasi minim memicu banyak perubahan di tengah jalan, dan perencanaan yang tidak akurat menyebabkan estimasi anggaran dan jadwal meleset.

Keenam, overlooked user classes. Satu produk biasanya dipakai berbagai kelas pengguna. Jika ada kelas yang terlupakan, kebutuhan mereka tidak akan terpenuhi.

5 Product Requirement dan Project Requirement

Dua istilah ini sering tertukar. Perhatikan Gambar **??**.

Product Requirement	**Project Requirement**
Requirement dari sistem, produk, atau perangkat lunak itu sendiri. Contoh: sistem harus mendukung login dengan email dan kata sandi.	Requirement dari proyek di luar produk. Contoh: infrastruktur cloud, pelatihan staf, dokumen legal, prosedur rilis.

Figure 5. Perbedaan product requirement dan project requirement.

Product requirement berbicara tentang software-nya: fitur apa, alur apa, batasan apa. **Project requirement** berbicara tentang proyek yang mengembangkan software itu: sumber daya, infrastruktur, pelatihan, legal, dan prosedur rilis. Seorang analis requirement wajib bisa membedakan keduanya, karena solusi untuk masing-masing ada di ranah yang berbeda.

6 Aktivitas Requirement Engineering

Requirement engineering terdiri dari dua aktivitas besar. Perhatikan Gambar **??**.

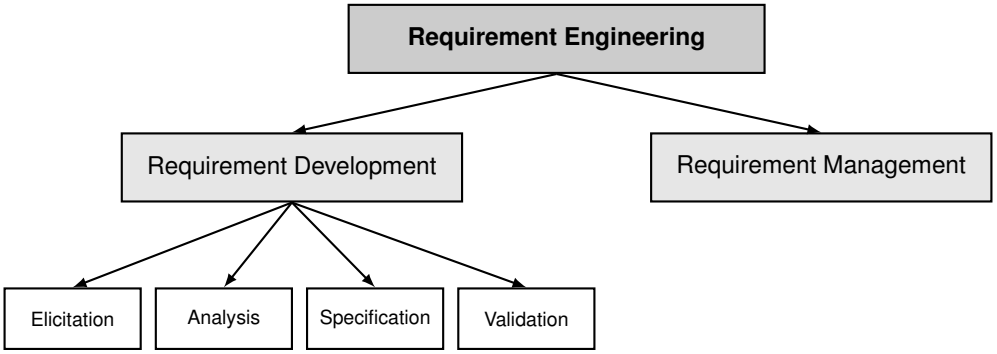


Figure 6. Aktivitas requirement engineering dan pembagiannya.

Requirement Development mencakup seluruh aktivitas yang berhubungan dengan eksplorasi, evaluasi, dokumentasi, dan konfirmasi requirement. Ia terdiri dari empat sub-aktivitas: elicitation (menggali), analysis (menganalisis), specification (menuliskan), dan validation (memvalidasi).

Requirement Management mencakup aktivitas menjaga rencana proyek tetap selaras dengan requirement, melacak status dan perubahan requirement sepanjang proyek, serta meninjau usulan perubahan dan mengevaluasi dampaknya sebelum disetujui.

Perbedaan keduanya sederhana. Development menjawab pertanyaan *apa kebutuhannya*. Management menjawab pertanyaan *bagaimana perubahannya dikendalikan*.

7 Dua Proses Requirement yang Lazim

Di industri, ada dua pendekatan besar. Perhatikan Gambar **??**.

Well Established (Volere)	**Agile Requirements**
Volere adalah bahasa untuk discovery, communication, dan management requirement, serta desain solusi. Alur: blastoff, travel for requirements, prototype, review, design, quality gateway, spesifikasi. Tantangan: delivery terlalu lama, kolaborasi sulit, prototyping sukar dibuat.	Alur: scope of work, business events, BUC scenario, stories, conversation, implement, iterasi. Fokus pada iterasi pendek dan kolaborasi intens dengan stakeholder. Tantangan: dokumentasi mudah tertinggal, requirement mudah menguap bila tidak dijaga.

Figure 7. Perbandingan proses requirement Volere dan Agile.

Volere cocok untuk proyek dengan requirement yang relatif stabil dan tim yang tersebar. Agile cocok untuk proyek yang requirement-nya cepat berubah dan tim berkumpul di satu tempat. Tidak ada yang lebih unggul secara mutlak. Yang penting adalah memilih satu, dan konsisten.

8 Perbandingan Ringkasan

Tabel **??** merangkum perbedaan kunci antar konsep yang telah dibahas.

Table 1. Ringkasan konsep utama minggu pertama.

Aspek	Konsep Pertama	Konsep Kedua
Jenis requirement	Product requirement	Project requirement
Aktivitas RE	Requirement development	Requirement management
Sub-aktivitas dev	Elicitation, analysis	Specification, validation
Fokus aktivitas	Menggali dan mendokumentasikan	Mengendalikan perubahan
Proses industri	Volere (terstruktur)	Agile (iteratif)
Tantangan	Delivery lama, prototyping sukar	Dokumentasi tertinggal, requirement menguap

9 Studi Kasus: Sepuluh Langkah yang Harus Kamu Hasilkan

Bagian ini memandu langkah demi langkah untuk memproduksi deliverable minggu pertama. Contoh kasus: Sistem Informasi Perpustakaan Kampus. Setiap langkah menghasilkan satu artefak yang bisa dikumpulkan.

9.1 Langkah 1: Baca Kasus dan Identifikasi Masalah

Sebelum menulis apa pun, baca deskripsi kasus sampai habis. Tulis satu paragraf masalah dalam bahasa sehari-hari.

[style=out] Masalah: Proses peminjaman buku di perpustakaan kampus masih manual. Mahasiswa mengisi formulir kertas, pustakawan memeriksa satu per satu, dan pengembalian sering

terlambat tanpa pemberitahuan. Akibatnya antrean panjang di jam sibuk dan denda tidak konsisten.

9.2 Langkah 2: Daftar Stakeholder

Tulis siapa saja yang akan terpengaruh keputusan proyek.

[style=out] Stakeholder: 1. Kepala Perpustakaan (project sponsor) 2. Pustakawan (end user aktif) 3. Mahasiswa (end user pasif) 4. Dosen (peminjam jangka panjang) 5. Tim IT kampus (pengelola infrastruktur)

9.3 Langkah 3: Daftar User Class

Kelompokkan pengguna berdasarkan frekuensi, kemampuan, dan hak akses.

[style=out] User Class: 1. Pustakawan (favored) - memakai tiap hari 2. Mahasiswa - memakai tiap minggu 3. Dosen - memakai tiap bulan 4. Kepala Perpustakaan - memakai untuk laporan

9.4 Langkah 4: Tulis Business Requirement

Jelaskan mengapa proyek ini ada.

[style=out] Business Requirement: Mengotomasi proses peminjaman dan pengembalian buku agar antrean di jam sibuk berkurang, denda terhitung konsisten, dan kepala perpustakaan mendapat laporan bulanan tanpa harus menghitung manual.

9.5 Langkah 5: Tulis Vision Statement

Rumuskan arah produk dalam satu kalimat.

[style=out] Vision Statement: Sistem Informasi Perpustakaan adalah aplikasi berbasis web yang memungkinkan mahasiswa meminjam buku tanpa antre, pustakawan mengelola stok dengan cepat, dan kepala perpustakaan memantau peminjaman secara real-time.

9.6 Langkah 6: Tentukan Scope

Pisahkan apa yang masuk dan apa yang tidak masuk.

[style=out] In Scope: - Pencarian katalog buku - Peminjaman dan pengembalian - Perhitungan denda otomatis - Laporan bulanan

Out of Scope: - Rekomendasi buku berbasis AI - Pembayaran denda via payment gateway - Aplikasi mobile native

9.7 Langkah 7: Tulis User Story

Gunakan format role-goal-benefit.

[style=out] User Story US-01: Sebagai mahasiswa, saya ingin mencari buku berdasarkan judul atau penulis, agar saya tidak perlu berkeliling rak.

User Story US-02: Sebagai pustakawan, saya ingin memindai kartu mahasiswa untuk peminjaman, agar proses pinjam selesai di bawah satu menit.

9.8 Langkah 8: Tulis Functional Requirement

Rumuskan perilaku sistem dengan input dan output.

[style=out] FR-01: Sistem menampilkan daftar buku yang cocok dengan kata kunci pencarian. Input: kata kunci (string), maksimal 50 karakter. Output: daftar maksimal 20 baris berisi judul, penulis, dan status ketersediaan.

FR-02: Sistem mencatat peminjaman saat pustakawan memindai kartu mahasiswa. Input: nomor kartu (string), kode buku (string). Output: transaksi peminjaman dengan tanggal jatuh tempo 14 hari setelah tanggal pinjam.

9.9 Langkah 9: Tulis Non-Functional Requirement

Rumuskan kualitas sistem.

[style=out] NFR-01 (Performance): Pencarian katalog selesai dalam waktu kurang dari 2 detik untuk 10.000 judul.

NFR-02 (Availability): Sistem tersedia 99,5 persen waktu dalam satu bulan, di luar jadwal pemeliharaan.

NFR-03 (Security): Password pengguna disimpan dengan hash bcrypt. Sesi berakhir setelah 30 menit idle.

9.10 Langkah 10: Validasi dan Kelola

Periksa apakah semua requirement konsisten, lalu tetapkan proses perubahan.

[style=out] Checklist Validasi: - Semua stakeholder sudah membaca dan setuju - Tidak ada requirement yang saling bertentangan - Setiap requirement bisa diuji (testable) - Setiap perubahan setelah ini harus melalui change request dengan persetujuan project sponsor

Sepuluh langkah di atas menghasilkan tiga artefak siap kumpul: dokumen vision and scope, dokumen requirement, dan matriks validasi. Inilah yang dimaksud dengan deliverable minggu pertama.

10 Kesalahan Umum

Tabel **??** merangkum kesalahan yang paling sering muncul saat mengerjakan tugas minggu pertama.

Table 2. Kesalahan umum pada pekerjaan requirement minggu pertama.

Kesalahan	Akibat	Perbaikan
Menulis solusi, bukan requirement	Dokumen jadi desain, bukan kebutuhan	Tulis apa yang dibutuhkan, bukan bagaimana mewujudkannya
Mencampur product dan project requirement	Scope melebar tak terkendali	Pisahkan dua daftar dan beri label jelas
Melewatkan satu user class	Kebutuhan kelas itu tidak terpenuhi	Periksa ulang daftar stakeholder sebelum menutup dokumen
Requirement ambigu seperti “cepat” tanpa angka	Developer menebak-nebak	Ganti dengan angka terukur: “kurang dari 2 detik”
Menulis requirement yang tidak bisa diuji	Validasi tidak bisa dilakukan	Ubah jadi kalimat dengan input, proses, dan output yang jelas
Scope tidak pernah ditulis eksplisit	Fitur baru terus masuk tanpa kendali	Selalu cantumkan in scope dan out of scope
Menganggap elicitation selesai setelah satu wawancara	Kebutuhan tidak lengkap	Lakukan minimal tiga sumber: wawancara, observasi, dokumen
Lupa menetapkan proses perubahan	Requirement berubah tanpa jejak	Tetapkan change request dan persetujuan sponsor

Dua kesalahan yang paling sering muncul adalah **mencampur solusi dengan requirement** dan **tidak menulis scope secara eksplisit**. Keduanya sama-sama berakar pada satu hal: ketergesaan. Sempatkan satu putaran penuh untuk membaca ulang sebelum menyerahkan.

11 Lembar Rujukan Cepat

Tabel **??** adalah ringkasan istilah dan artefak yang wajib dikuasai sebelum pertemuan kedua.

Table 3. Lembar rujukan cepat untuk minggu pertama.

Kebutuhan atau Istilah	Definisi atau Artefak
Requirement	Spesifikasi apa yang harus diimplementasikan oleh sistem
Requirement engineering	Disiplin menggali, menganalisis, mendokumentasikan, dan memvalidasi kebutuhan
Stakeholder	Siapa pun yang terpengaruh aktivitas dan keputusan proyek
User class	Kelompok pengguna berdasarkan frekuensi, kemampuan, dan hak akses
Product champion	Perwakilan user class yang menjadi jembatan ke analis
Business requirement	Alasan mengapa proyek dilakukan, biasanya di level sponsor
User requirement	Tugas yang bisa dilakukan pengguna dengan produk
System requirement	Requirement di level sistem: fungsional dan non-fungsional
Functional requirement	Perilaku yang harus dilakukan sistem
Non-functional requirement	Kualitas yang harus dimiliki sistem
Product requirement	Requirement dari sistem/produk/software
Project requirement	Requirement dari proyek di luar produk
Requirement development	Elicitation, analysis, specification, validation
Requirement management	Tracking status, change control, impact evaluation
Elicitation	Aktivitas menggali kebutuhan dari stakeholder
Vision	Arah jangka panjang produk
Scope	Bagian yang dikerjakan pada rilis atau iterasi saat ini
Volere	Proses requirement terstruktur untuk discovery hingga design
Agile requirements	Proses requirement iteratif berbasis cerita pengguna

12 Kesimpulan

Requirement engineering adalah fondasi seluruh siklus pengembangan perangkat lunak. Ia terdiri dari dua aktivitas besar, yaitu requirement development dan requirement management, dengan empat sub-aktivitas pengembangan: elicitation, analysis, specification, dan validation. Tiga prinsip penutup:

1. **Gali dulu, jangan menebak.** Perangkat lunak tidak boleh dibangun berdasarkan asumsi developer. Setiap requirement wajib punya jejak ke stakeholder nyata.
2. **Pisahkan produk dari proyek.** Product requirement dan project requirement harus ditulis sebagai dua daftar terpisah agar scope tidak melebar tanpa kendali.
3. **Uji sebelum menyerahkan.** Requirement yang tidak bisa diuji bukan requirement, melainkan harapan. Ubah setiap kalimat menjadi sesuatu yang bisa diverifikasi dengan input, proses, dan output yang jelas.

Pada pertemuan berikutnya, prinsip-prinsip ini akan dipakai untuk mempelajari stakeholder dan aktivitas requirement engineering secara lebih mendalam. Lembar rujukan cepat di atas adalah bekal minimum yang harus dibawa.

Referensi

References

[1] Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson.

[2] Wiegers, K., & Beatty, J. (2013). *Software Requirements* (3rd ed.). Microsoft Press.

[3] Robertson, S., & Robertson, J. (2012). *Mastering the Requirements Process: Getting Requirements Right* (3rd ed.). Addison-Wesley.

[4] International Institute of Business Analysis. (2015). *A Guide to the Business Analysis Body of Knowledge (BABOK Guide)* (v3.0). IIBA.

[5] Pressman, R. S., & Maxim, B. R. (2020). *Software Engineering: A Practitioner's Approach* (9th ed.). McGraw-Hill Education.

[6] IEEE. (1998). *IEEE Recommended Practice for Software Requirements Specifications (IEEE Std 830-1998)*. Institute of Electrical and Electronics Engineers.