Vision dan Scope
Kompas dan Rute Proyek Perangkat Lunak

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

November 5, 2026

Abstrak

Makalah ini membahas dua konsep yang sering dicampur padahal berbeda secara fundamental: product vision dan product scope. Ia memberi tiga tes diagnosis untuk membedakan keduanya, menganalisis ulang kuis "apakah ini perubahan vision" dari slide kelas dengan lebih cermat, dan memperkenalkan ide bahwa scope creep bukan persoalan klien, melainkan persoalan disiplin analyst. Delapan visualisasi dan tiga tabel menyertai pembahasan, termasuk template Vision and Scope Document empat bagian serta lima proses Project Scope Management. Sebuah studi kasus sepuluh langkah dengan sistem reservasi lapangan futsal menunjukkan bagaimana vision dipecah menjadi scope per rilis, bagaimana WBS dipakai sebagai alat negosiasi, dan bagaimana setiap permintaan baru dijawab dengan tepat satu dari tiga kemungkinan: ya, tidak, atau rilis berikutnya.

1 Pendahuluan

1.1 Analogi Pembuka

Bayangkan seorang pendaki gunung. Ia membuka peta, menemukan puncak yang ingin ia tuju, dan menandainya dengan lingkaran merah. Puncak itu adalah **vision**. Ia tidak berubah meskipun ia salah jalan, kehabisan air, atau berpapasan dengan jalur alternatif.

Lalu ia menetapkan: hari ini saya akan mendaki dari basecamp ke shelter satu. Besok dari shelter satu ke shelter dua. Lusa dari shelter dua ke puncak. Rute-rute harian ini adalah **scope**. Mereka bisa berubah kapan saja, tergantung cuaca, kondisi fisik, atau informasi baru dari pendaki lain yang baru turun.

Pendaki yang tidak punya puncak akan berjalan berputar-putar. Pendaki yang tidak punya rute harian akan berjalan tanpa rencana dan kehabisan bekal di tengah jalan. Vision tanpa scope adalah mimpi. Scope tanpa vision adalah kesibukan tanpa arah.

1.2 Mengapa Topik Ini Penting

Slide kelas membuka dengan kalimat tajam: “A project without a clearly defined vision and scope will be an unfocused and undirected project.” Tapi kalimat itu belum menjelaskan *mengapa* dua hal ini sering disalahpahami sebagai satu hal yang sama.

Di lapangan, banyak tim yang menyebut “visi kami adalah membuat aplikasi kasir dengan fitur A, B, dan C”. Itu bukan vision. Itu scope yang sedang menyamar jadi vision. Ciri khasnya: begitu salah satu fitur dibatalkan, seluruh proyek terasa kehilangan arah. Tim yang benar-benar punya vision akan tetap tahu ke mana harus pergi meskipun satu fitur dibuang.

1.3 Peta Pembahasan

Makalah ini disusun berurutan. Bagian kedua membahas vision, termasuk tiga tes diagnosis. Bagian ketiga membahas scope dengan tiga tes yang setara. Bagian keempat menyatukan keduanya dalam model hierarki rilis. Bagian kelima memberi diagnosis untuk pola paling sering muncul: fitur yang masuk, keluar, lalu masuk lagi. Bagian keenam membahas cara mengukur scope. Bagian ketujuh membahas siapa yang berwenang menyelesaikan konflik requirement. Bagian kedelapan menyajikan template Vision and Scope Document. Bagian kesembilan membedah lima proses scope management. Bagian kesepuluh sampai kedua belas membahas “is this in scope”, scope creep, dan kuis vision change dari slide. Bagian ketiga belas sampai kelima belas merangkum perbandingan, kesalahan umum, dan lembar rujukan cepat. Bagian keenam belas memandu studi kasus sepuluh langkah, dan bagian ketujuh belas menutup dengan tiga prinsip.

2 Apa Itu Product Vision

Product vision adalah arah jangka panjang produk. Fungsinya menyelaraskan seluruh stakeholder pada arah yang sama. Ia berlaku untuk produk secara keseluruhan, bukan untuk satu rilis tertentu. Biasanya ia tidak berubah untuk waktu yang lama.

Perhatikan Gambar 1. Vision itu seperti kompas, bukan peta jalan.

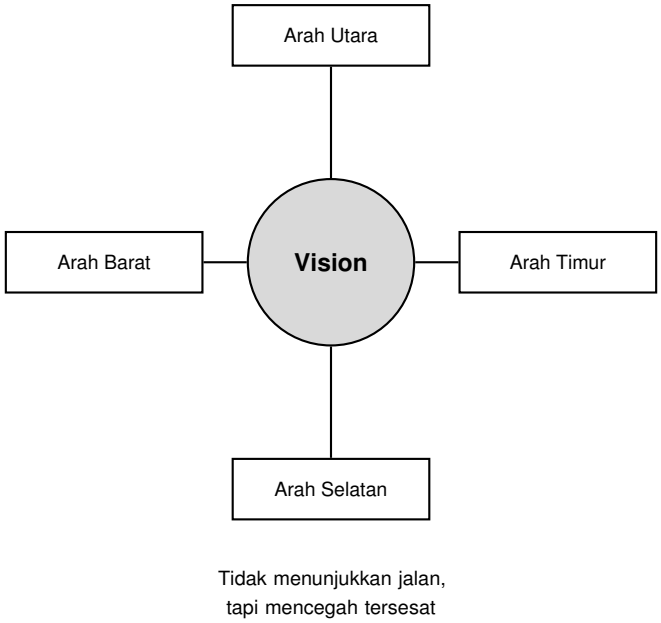


Figure 1. Vision sebagai kompas, bukan peta jalan.

Kompas tidak memberi tahu Anda harus belok kiri di pertigaan mana. Ia memberi tahu Anda sedang menghadap ke mana. Begitu juga vision: ia tidak menentukan fitur apa yang dibangun minggu depan, tapi ia memberi tahu apakah fitur itu membawa Anda lebih dekat atau lebih jauh dari tujuan.

2.1 Tiga Tes Diagnosis

Bagaimana membedakan vision yang asli dari scope yang menyamar? Perhatikan Gambar 2. Ada tiga pertanyaan yang bisa dipakai.

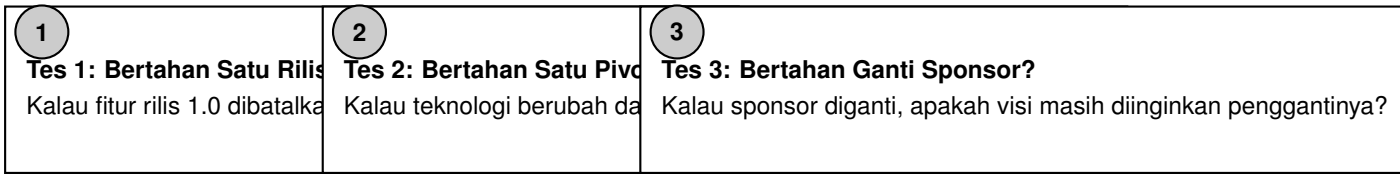


Figure 2. Tiga tes untuk memastikan sesuatu itu vision asli, bukan scope yang menyamar.

Kalau kalimat yang Anda tulis sebagai “visi” gagal lulus salah satu tes di atas, ia bukan vision. Ia scope yang ditulis terlalu jauh ke depan. Contoh visi yang gagal lulus Tes 1: “Visi kami adalah membangun aplikasi kasir dengan modul inventori dan laporan penjualan.” Buang modul inventori, dan visi itu sudah tidak berarti apa-apa. Visi yang lulus Tes 1 bisa jadi seperti ini: “Visi kami adalah membuat pemilik toko kecil bisa mengelola keuangannya tanpa perlu belajar akuntansi.” Ini tetap berarti meskipun modul inventori dihapus.

2.2 Vision Conflict Diselesaikan oleh Siapa

Business requirement yang dikumpulkan dari berbagai sumber bisa saling bertentangan. Contoh: pemilik toko ingin laporan yang detail, sementara kasir ingin tampilan yang sesederhana mungkin. Siapa yang memutuskan?

Jawaban di slide jelas: **project sponsor**, bukan developer. Developer yang memutuskan adalah developer yang akan disalahkan ketika hasilnya tidak sesuai harapan siapa pun. Sponsor adalah pihak yang paling tahu apa yang mereka butuhkan, karena merekalah yang menanggung konsekuensi bisnisnya.

Analyst dalam situasi ini bukan pengambil keputusan, tapi fasilitator. Ia memetakan konflik, menyajikan opsi, dan meminta sponsor memilih. Ini paralel dengan apa yang sudah dibahas di minggu ketiga: analyst memfasilitasi, bukan mengambil alih.

3 Apa Itu Product Scope

Product scope mengidentifikasi bagian dari proyek saat ini yang akan dikerjakan. Ia berlaku untuk iterasi spesifik yang akan diimplementasikan pada increment berikutnya. Scope lebih dinamis dari vision.

Kalau vision adalah kompas, scope adalah rute hari ini. Ia punya titik awal yang jelas (sumber daya saat ini), titik akhir yang jelas (apa yang akan dirilis), dan batas waktu yang jelas (kapan rilis itu terjadi).

Perhatikan Gambar 3. Scope itu seperti jendela, bukan dinding.

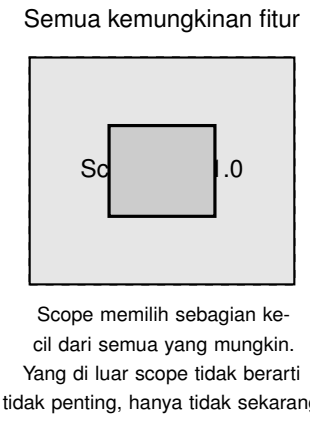


Figure 3. Scope sebagai jendela yang memilih sebagian dari semua kemungkinan.

Yang di luar scope tidak sama dengan yang tidak penting. Ia hanya tidak dikerjakan pada iterasi saat ini. Perbedaan ini kecil tapi menentukan: kalau Anda mencampur keduanya, Anda akan menolak fitur bagus karena mengira fitur itu tidak penting, padahal hanya belum waktunya.

3.1 Tiga Tes Diagnosis untuk Scope

Sebagaimana vision punya tiga tes, scope juga punya tiga tes yang setara. Perhatikan Gambar 4.

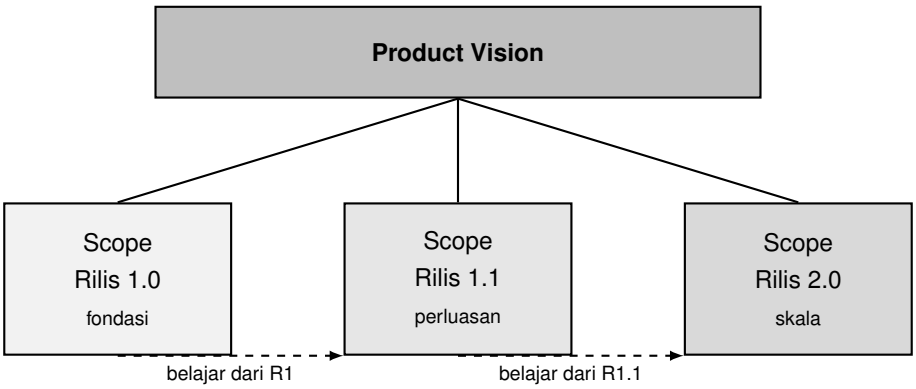
Tes 1: Bisa Selesai dalam	**Tes 2: Bisa Dipecah ke W**	**Tes 3: Punya Batas Waktu Jelas?**
Apakah semua item bisa di	Apakah setiap item cukup k	Apakah ada tanggal atau milestone yang menandai akhir?

Figure 4. Tiga tes untuk memastikan scope-nya realistis dan terukur.

Kalau sesuatu gagal lulus Tes 1, ia bukan scope rilis ini, melainkan scope beberapa rilis ke depan. Kalau gagal lulus Tes 2, ia terlalu besar dan perlu dipecah. Kalau gagal lulus Tes 3, ia belum scope, masih harapan.

4 Hierarki Vision dan Scope Rilis

Satu product vision melingkupi banyak product scope. Setiap rilis punya scope-nya sendiri. Perhatikan Gambar 5.



Vision tetap. Scope tiap rilis menyesuaikan diri berdasarkan apa yang dipelajari dari rilis sebelumnya.

Figure 5. Product vision melingkupi scope setiap rilis. Scope rilis berikutnya disesuaikan berdasarkan pembelajaran dari rilis sebelumnya.

Ada satu konsekuensi penting dari model ini yang sering terlewat: **scope rilis berikut tidak ditentukan di awal**. Ia ditentukan setelah rilis sebelumnya berjalan, ketika tim sudah punya data nyata tentang apa yang dipakai dan apa yang tidak. Ini yang disebut *release treadmill*: setiap rilis memberi umpan balik yang mengubah scope rilis berikutnya, tapi tidak mengubah vision.

Tim yang menetapkan scope untuk semua rilis di awal adalah tim yang akan memaksa scope rilis 2.0 tetap sama meskipun data dari rilis 1.0 menunjukkan asumsi awalnya salah. Itu bukan disiplin, itu kekakuan.

5 Tanda Vision Belum Cukup

Slide kelas menyebut satu tanda yang sangat tajam: **business requirement belum cukup terdefinisi jika ada fitur yang awalnya dimasukkan, lalu dihapus, lalu dimasukkan lagi**. Perhatikan Gambar 6.

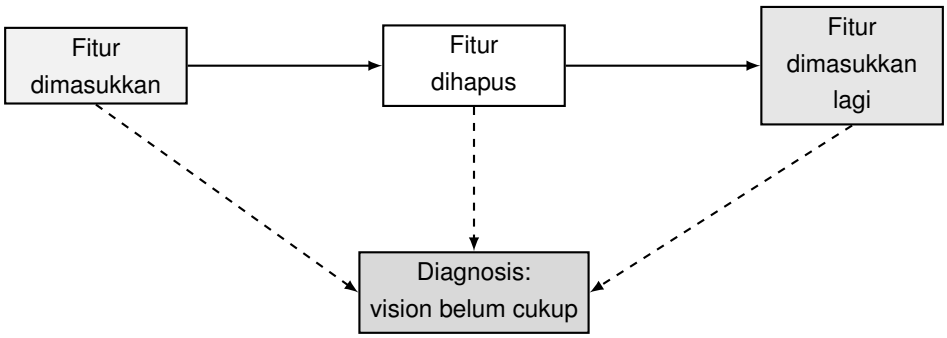


Figure 6. Pola osilasi fitur yang menandakan vision belum cukup dalam.

Analisis yang lebih dalam: osilasi ini bukan tanda scope yang salah. Ia tanda vision yang belum cukup dalam untuk memutuskan apakah fitur itu bagian dari arah atau bukan. Tanpa vision yang jelas, setiap fitur diuji dengan selera hari itu. Hari Senin terasa penting, hari Rabu terasa tidak, hari Jumat terasa penting lagi. Yang berubah bukan fiturnya, tapi selera yang menguji.

Solusinya bukan menulis scope lebih detail. Solusinya memperdalam vision sampai ia bisa menjawab satu pertanyaan sederhana untuk setiap fitur: fitur ini membawa kita lebih dekat ke arah yang kita tuju, atau tidak. Kalau vision cukup dalam, jawaban itu keluar dalam hitungan menit, bukan minggu.

6 Mengukur Scope

Slide menyebut empat cara mengukur scope. Perhatikan Gambar 7.

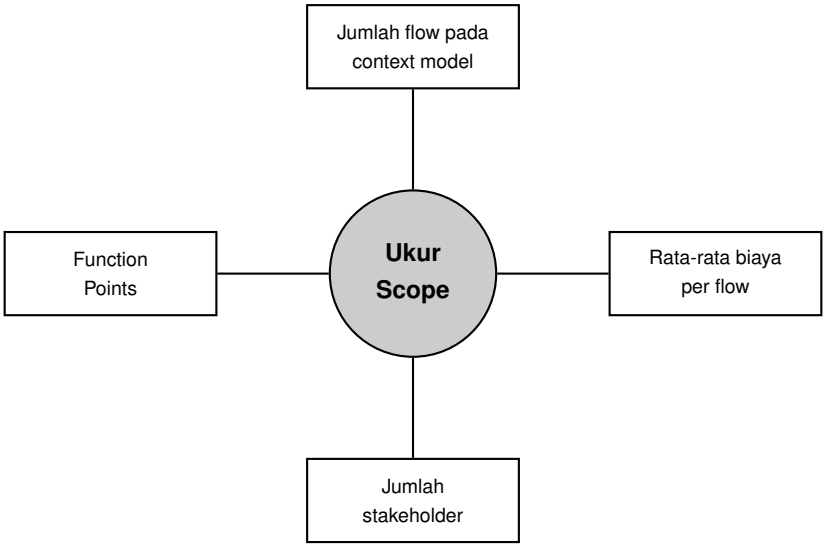


Figure 7. Empat cara mengukur scope.

Pertama, jumlah flow pada context model. Context model adalah diagram yang menampilkan sistem sebagai satu kotak pusat, dengan panah masuk (input) dan panah keluar (output). Setiap panah adalah satu flow. Semakin banyak flow, semakin besar scope.

Kedua, rata-rata biaya per flow. Kalau satu flow rata-rata memakan 3 hari kerja, dan ada 20 flow, maka scope adalah sekitar 60 hari kerja. Ini estimasi kasar, tapi berguna untuk menolak scope yang tidak realistis di tahap awal.

Ketiga, jumlah stakeholder. Semakin banyak stakeholder, semakin banyak sudut pandang yang harus dipuaskan, semakin besar scope-nya.

Keempat, Function Points. Metode pengukuran yang menghitung jumlah input, output, query, file internal, dan file eksternal dari suatu sistem. Metode ini lebih formal dan sering dipakai di proyek enterprise.

Keempat cara ini tidak harus dipakai semua. Pilih satu atau dua yang paling cocok dengan konteks proyek, dan gunakan secara konsisten.

7 Template Vision and Scope Document

Vision and Scope Document mengumpulkan business requirement ke dalam satu dokumen yang menjadi fondasi pengembangan produk. Pemilik dokumen ini biasanya project sponsor. Project sponsor bekerja bersama development team atau analyst untuk membuatnya.

Dokumen ini punya empat bagian dengan empat belas subbagian. Perhatikan Tabel 1.

Table 1. Template Vision and Scope Document empat bagian.

Bagian	Subbagian dan Isi
1. Business Requirements	1.1 Background, 1.2 Business Opportunity, 1.3 Business Objectives and Success Criteria, 1.4 Customer or Market Needs, 1.5 Business Risks
2. Vision of the Solution	2.1 Vision Statement, 2.2 Major Features, 2.3 Assumptions and Dependencies
3. Scope and Limitations	3.1 Scope of Initial Release, 3.2 Scope of Subsequent Releases, 3.3 Limitations and Exclusions
4. Business Context	4.1 Stakeholder Profiles, 4.2 Project Priorities, 4.3 Operating Environment

Bagian 1 dan 2 menjelaskan **mengapa dan apa**. Bagian 3 dan 4 menjelaskan **seberapa banyak dan dengan siapa**. Dokumen ini bukan technical specification. Ia tidak menjelaskan bagaimana sistem dibangun. Ia hanya menjelaskan apa yang akan dicapai, untuk siapa, dalam batas apa.

8 Lima Proses Project Scope Management

Project scope management punya lima proses yang membentuk satu siklus. Perhatikan Gambar 8.

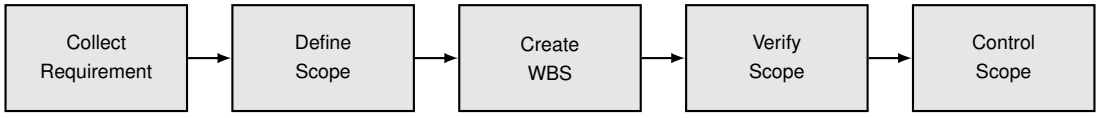


Figure 8. Lima proses project scope management.

Collect Requirement. Mendefinisikan dan mendokumentasikan kebutuhan stakeholder untuk memenuhi tujuan proyek.

Define Scope. Mengembangkan deskripsi detail dari proyek dan produk. Hasilnya adalah scope statement yang jelas.

Create WBS. Memecah deliverable dan pekerjaan proyek menjadi komponen yang lebih kecil dan bisa dikelola. WBS adalah singkatan dari Work Breakdown Structure.

Verify Scope. Memformalkan penerimaan deliverable yang sudah selesai. Ini bukan uji coba teknis, tapi persetujuan formal dari pihak yang berwenang.

Control Scope. Memantau status dan mengelola perubahan pada baseline scope, baik untuk produk maupun proyek.

Yang sering terlewat adalah **Create WBS**. Banyak tim menganggap WBS sebagai dokumen teknis yang hanya dibutuhkan project manager. Padahal WBS adalah alat negosiasi paling ampuh dalam scope management. Ketika sponsor meminta tambahan fitur, analyst bisa membuka WBS dan berkata: "Ini yang sudah ada di dalam 3 bulan. Kalau fitur ini ditambahkan, salah satu dari tiga item ini harus dikeluarkan, atau timeline bertambah."

WBS memaksa percakapan tentang trade-off menjadi konkret. Tanpa WBS, percakapan selalu berakhir dengan "kita coba dulu" — yang artinya "kita akan menyelesaikannya dengan lembur,

meskipun tidak akan selesai tepat waktu."

9 Keeping Scope in Focus

Setiap kali ada permintaan requirement baru, analyst wajib bertanya satu pertanyaan: **"Is this in scope?"** Perhatikan Gambar 9.

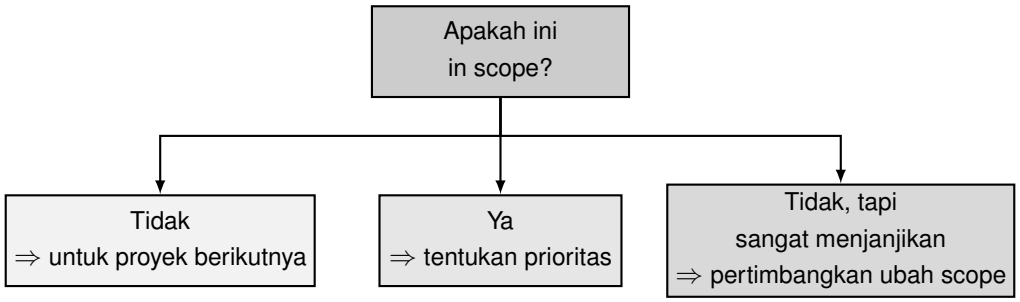


Figure 9. Tiga kemungkinan jawaban atas pertanyaan "is this in scope?".

Jika di luar scope tapi menarik: catat untuk proyek berikutnya. Jangan buang, jangan kerjakan sekarang.

Jika di dalam scope: tentukan prioritasnya dibanding requirement lain. Bisa jadi ia masuk rilis ini, bisa jadi ia menggeser requirement lain yang lebih rendah prioritas.

Jika di luar scope tapi sangat menjanjikan: pertimbangkan untuk mengubah scope proyek. Ini keputusan berat, dan hanya sponsor yang bisa mengambilnya. Analyst memfasilitasi.

Perhatikan bahwa ada tiga jawaban, bukan dua. Sebagian besar analyst pemula hanya punya dua jawaban: ya atau tidak. Padahal jawaban ketiga — "sangat menjanjikan tapi tidak untuk sekarang" — adalah jawaban yang menyelamatkan hubungan dengan sponsor. Ia menunjukkan bahwa Anda serius mendengarkan, tapi juga serius menjaga scope.

10 Scope Creep dan Cara Menanganinya

Scope creep adalah pertumbuhan scope yang tidak terkendali. Ia terjadi ketika permintaan baru diterima tanpa perubahan timeline, anggaran, atau kualitas yang sepadan. Perhatikan Gambar 10.

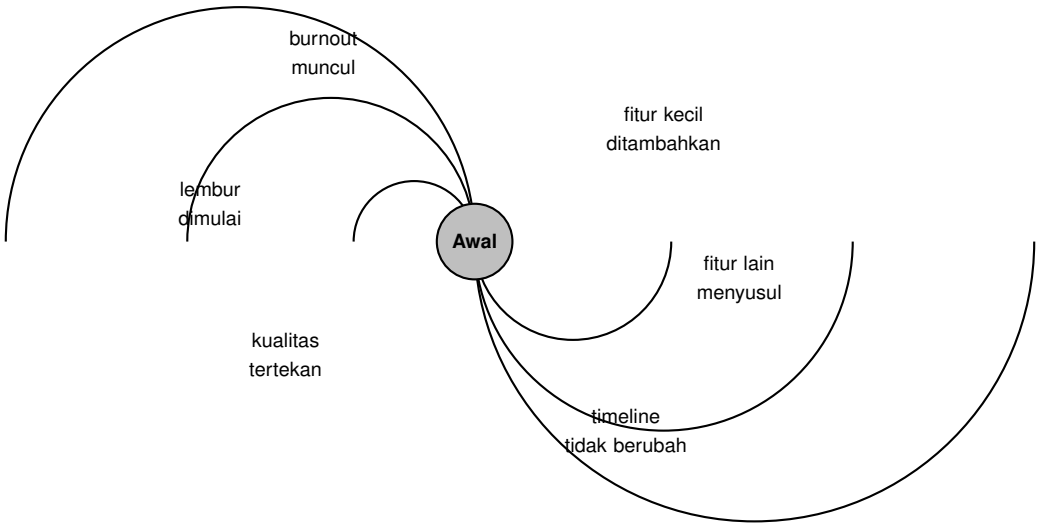


Figure 10. Spiral scope creep yang mulai dari fitur kecil dan berakhir pada kelelahan tim.

Slide menyebut enam respons terhadap scope creep:

1. Minta klien memprioritaskan requirement.
2. Buat ekspektasi jelas.
3. Gambarkan scope bersama klien.
4. Selalu tanya “is this in scope?”.
5. Perpanjang jadwal proyek.
6. Tambah biaya proyek.

Tapi ini belum menjawab akar masalahnya. Akar scope creep bukan klien yang terus meminta. Akar scope creep adalah analyst yang tidak punya sistem untuk menolak dengan elegan. Kalau setiap permintaan baru dijawab dengan “kita coba dulu”, maka setiap permintaan pada akhirnya akan dikerjakan, dan spiral pada Gambar 10 dimulai.

Analyst yang punya sistem menjawab setiap permintaan dengan tepat satu dari tiga kata: **ya**, **tidak**, **rilis berikutnya**. Tidak ada kata keempat seperti “mungkin” atau “kita lihat nanti”. Kata-kata itu adalah cara paling halus untuk memulai scope creep.

Perhatikan pula satu jebakan yang sering muncul: *sunk cost fallacy*. Tim kadang menolak membuang fitur yang sudah setengah jadi meskipun fitur itu di luar scope. Alasan yang muncul: “sudah terlanjur dikerjakan, sayang kalau dibuang.” Padahal kode yang tidak dipakai adalah technical debt yang terus menua. Membuangnya sekarang lebih murah daripada memeliharanya selama tiga tahun.

11 Analisis Kuis Vision Change

Slide memberi kuis dengan empat pilihan. Perhatikan Tabel 2.

Table 2. Analisis kuis vision change dari slide kelas.

Pilihan	Analisis	Kesimpulan
A. Klien pindah dari mobile gaming ke virtual reality	Tergantung bagaimana vision ditulis. Kalau visi "game imersif", VR adalah scope. Kalau visi "game mobile", VR adalah vision change.	Ambigu, tergantung rumusan
B. Tim mengubah detail implementasi produk	Detail implementasi adalah ranah scope, bukan vision.	Bukan vision change
C. Proyek kelebihan anggaran, dipaksa dihentikan sementara	Ini masalah constraint, bukan vision.	Bukan vision change
D. Klien mengubah produk dari game project management menjadi aplikasi desain furnitur	Domain produk berubah total. Visi lama tidak lagi relevan.	Vision change

Kunci jawaban yang biasa diberikan adalah **A dan D**. Tapi analisis di Tabel 2 menunjukkan sesuatu yang lebih menarik: pilihan A sebenarnya ambigu. Ia hanya menjadi vision change jika vision statement-nya secara eksplisit menyebut “mobile”. Kalau vision statement-nya menyebut “game imersif untuk profesional muda”, maka VR adalah salah satu rute, bukan perubahan tujuan.

Pelajaran dari analisis ini: **jawaban atas pertanyaan “apakah ini vision change” bergantung pada bagaimana vision ditulis**. Ini bukan kelemahan kuis, ini kekuatan kuis. Ia menunjukkan bahwa vision statement yang ditulis dengan buruk akan menghasilkan kebingungan bahkan pada pertanyaan sesederhana ini.

Pilihan D tidak ambigu. Nama domain berubah dari game ke furnitur. Tidak ada rumusan vision yang bisa menutupi fakta bahwa dua hal ini adalah dua produk yang berbeda.

12 Perbandingan Ringkasan

Tabel 3 merangkum perbedaan kunci vision dan scope.

Table 3. Perbandingan vision dan scope.

Aspek	Vision	Scope
Fungsi	Arah jangka panjang	Bagian yang dikerjakan sekarang
Berlaku untuk	Produk secara keseluruhan	Satu iterasi atau rilis
Dinamika	Relatif tetap	Lebih dinamis
Pemilik	Project sponsor	Analyst bersama project manager
Tes 1	Bertahan jika satu rilis dibatalkan	Bisa selesai dalam satu rilis
Tes 2	Bertahan jika teknologi berubah	Bisa dipecah ke WBS
Tes 3	Bertahan jika sponsor diganti	Punya batas waktu yang jelas
Pertanyaan uji	Membawa lebih dekat ke tujuan?	Masuk dalam baseline yang disepakati?

Perhatikan bahwa kolom Tes 1 sampai 3 diisi dengan pertanyaan, bukan jawaban. Ini disengaja. Tes itu tidak memberi jawaban otomatis. Tes itu hanya memberi pertanyaan yang wajib dijawab analyst sebelum ia menulis sesuatu sebagai vision atau scope.

13 Studi Kasus: Sistem Reservasi Lapangan Futsal

Bagian ini memandu sepuluh langkah untuk menghasilkan deliverable minggu kelima. Kasus yang dipakai: sistem reservasi lapangan futsal. Tujuannya memudahkan pelanggan memesan lapangan tanpa harus menelepon pengelola.

13.1 Langkah 1: Tulis Vision Statement

Tulis visi dalam satu kalimat yang lulus tiga tes.

```
Vision Statement:
```

```
Memungkinkan pengelola lapangan futsal kecil mengelola
reservasi tanpa perlu menelepon pelanggan satu per
satu, sehingga lapangan lebih sering terpakai dan
pendapatan lebih stabil.

Tes 1 (rilis dibatalkan): tetap berarti karena masalah
utamanya (telepon manual) masih ada.

Tes 2 (teknologi berubah dari web ke WhatsApp bot):
tetap berarti karena masalahnya bukan di platform,
tapi di cara reservasi.

Tes 3 (sponsor diganti): tetap berarti karena
pengelola baru juga ingin pendapatan stabil.
```

13.2 Langkah 2: Tulis Business Opportunity

Kaitkan visi dengan peluang bisnis yang konkret.

```
Business Opportunity:

Pengelola lapangan futsal saat ini mencatat reservasi
di buku tulis. Sering terjadi double booking karena
dua pelanggan menelepon di saat bersamaan. Selain
itu, pengelola tidak punya data historis untuk
menentukan jam sewa yang paling menguntungkan.

Dengan sistem reservasi, pengelola bisa menghindari
double booking dan melihat pola jam sibuk untuk
menetapkan harga dinamis.
```

13.3 Langkah 3: Rumuskan Tiga Tes untuk Setiap Kandidat Vision

Sebelum menetapkan satu vision, uji beberapa kandidat.

```
Kandidat A: "Aplikasi reservasi futsal berbasis web."
Tes 1: gagal, karena kalau aplikasi web dibatalkan,
        tidak ada visi alternatif.
Tes 2: gagal, karena kalau platform berubah ke
        mobile, visi ini tidak lagi berlaku.
Keputusan: ini scope, bukan vision.

Kandidat B: "Pemilik lapangan futsal bisa mengelola
reservasi tanpa double booking."
Tes 1: lulus.
```

```
Tes 2: lulus.
Tes 3: lulus.
Keputusan: ini vision.
```

13.4 Langkah 4: Tulis Scope of Initial Release

Dari visi, turunkan apa yang dikerjakan di rilis pertama.

```
Scope Rilis 1.0:

In scope:
- Daftar lapangan (nama, harga per jam)
- Formulir reservasi pelanggan
- Kalender reservasi (view per lapangan)
- Notifikasi WhatsApp otomatis setelah bayar
- Dashboard pengelola (lihat reservasi hari ini)

Out of scope untuk rilis 1.0:
- Harga dinamis (rilis 1.1)
- Pembayaran online (rilis 2.0)
- Aplikasi mobile native (rilis 2.0)
- Integrasi dengan kasir (rilis 3.0)

Batas waktu: rilis 1.0 dalam 8 minggu
```

13.5 Langkah 5: Petakan Flow di Context Model

Ukur scope dengan menghitung flow.

```
Context Model Rilis 1.0:

Flow masuk (input):
1. Pelanggan memilih lapangan
2. Pelanggan memilih tanggal dan jam
3. Pelanggan mengisi nama dan nomor telepon
4. Pelanggan konfirmasi pembayaran manual
5. Pengelola login

Flow keluar (output):
1. Konfirmasi reservasi ke pelanggan
2. Notifikasi ke pengelola
3. Daftar reservasi hari ini
4. Kalender reservasi

Total: 9 flow
Estimasi: 9 flow x 3 hari per flow = 27 hari kerja
```

```
Buffer 30 persen: 35 hari kerja = 7 minggu
Batas waktu 8 minggu: realistis
```

13.6 Langkah 6: Buat WBS Rilis 1.0

Pecah scope menjadi work package.

```
WBS Rilis 1.0:

1. Modul Lapangan
   1.1 Skema database lapangan
   1.2 API daftar lapangan
   1.3 Halaman daftar lapangan

2. Modul Reservasi
   2.1 Skema database reservasi
   2.2 API cek ketersediaan
   2.3 API buat reservasi
   2.4 Halaman formulir reservasi
   2.5 Kalender reservasi

3. Modul Notifikasi
   3.1 Integrasi WhatsApp API
   3.2 Template pesan konfirmasi

4. Modul Dashboard Pengelola
   4.1 Autentikasi pengelola
   4.2 Halaman daftar reservasi hari ini

Total work package: 11
```

13.7 Langkah 7: Siapkan Jawaban untuk Setiap Permintaan Baru

Setiap permintaan baru harus dijawab dengan satu dari tiga kemungkinan.

```
Permintaan 1: "Bisa tambahkan fitur harga dinamis
untuk jam malam?"
Jawaban: Rilis berikutnya (1.1).
Alasan: tidak masuk baseline rilis 1.0.

Permintaan 2: "Bisa tambahkan fitur booking untuk
turnamen antar kampung?"
Jawaban: Tidak.
Alasan: membutuhkan modul baru (bracket turnamen)
yang bukan bagian dari visi reservasi.
```

```
Permintaan 3: "Bisa tampilkan peta lokasi lapangan
di halaman reservasi?"
Jawaban: Ya, jika masuk kerja sama dengan
prioritas lebih rendah, seperti penggantian
warna tombol.
Alasan: kecil, masuk scope, tapi harus geser
prioritas.

Setiap jawaban dicatat di change log.
```

13.8 Langkah 8: Susun Rencana Verifikasi Scope

Setiap work package harus bisa diverifikasi.

```
Kriteria Verifikasi per Work Package:

1. Modul Lapangan
   Kriteria: admin bisa tambah, ubah, hapus lapangan
   dan pelanggan melihat daftar terbaru

2. Modul Reservasi
   Kriteria: pelanggan bisa reservasi lapangan yang
   masih kosong di tanggal dan jam yang dipilih,
   dan sistem menolak reservasi yang bentrok

3. Modul Notifikasi
   Kriteria: pelanggan menerima pesan WhatsApp
   dalam 30 detik setelah pembayaran manual
   dikonfirmasi pengelola

4. Modul Dashboard Pengelola
   Kriteria: pengelola login dengan email dan
   password, dan melihat daftar reservasi hari ini

Verifikasi dilakukan dengan skenario uji dan
persetujuan tertulis dari sponsor.
```

13.9 Langkah 9: Kontrol Scope dengan Change Log

Setiap perubahan scope dicatat dan dievaluasi.

```
Format Change Log:

No | Tanggal | Deskripsi Permintaan | Keputusan |
Dampak ke Timeline | Dampak ke Anggaran
```

```
CL-01 | 2026-11-01 | Tambah harga dinamis |
Ditunda ke rilis 1.1 | Tidak ada | Tidak ada

CL-02 | 2026-11-03 | Tambah fitur turnamen |
Ditolak | Tidak ada | Tidak ada

CL-03 | 2026-11-05 | Tambah peta lokasi |
Diterima, geser prioritas | +2 hari |
+5 persen anggaran

Semua perubahan di atas baseline harus melalui
persetujuan sponsor.
```

13.10 Langkah 10: Evaluasi Vision vs Scope Setelah Rilis

Setelah rilis 1.0, evaluasi apa yang berubah.

```
Evaluasi Setelah Rilis 1.0:

Vision: tidak berubah.
Masalah double booking teratasi.
Pendapatan lebih stabil.

Scope rilis 1.1 disesuaikan berdasarkan data:
- Harga dinamis ternyata lebih penting dari
  perkiraan awal (60 persen pelanggan bertanya
  soal harga malam)
- Fitur peta lokasi tidak dipakai (kurang dari
  5 persen pelanggan membuka)
- Fitur baru masuk radar: reminder otomatis
  H-1 sebelum jadwal

Keputusan rilis 1.1:
- Harga dinamis: naik prioritas
- Peta lokasi: turun prioritas
- Reminder otomatis: masuk scope baru

Vision tidak berubah. Scope berubah berdasarkan data.
```

Sepuluh langkah di atas menghasilkan empat artefak siap kumpul: dokumen vision and scope empat bagian, context model dengan flow terhitung, WBS rilis 1.0, dan change log rilis 1.0. Inilah deliverable minggu kelima.

14 Kesalahan Umum

Tabel 4 merangkum kesalahan yang paling sering muncul saat menulis vision dan scope.

Table 4. Kesalahan umum pada pekerjaan vision dan scope.

Kesalahan	Akibat	Perbaikan
Menulis daftar fitur sebagai vision	Visi hilang saat satu fitur dibatalkan	Tulis vision dalam bahasa masalah, bukan fitur
Menganggap vision sebagai scope	Scope melebar tak terkendali	Pisahkan dua dokumen, dua pemilik
Scope tanpa batas waktu	Scope creep tidak terdeteksi	Setiap scope harus punya milestone akhir
Menulis scope untuk semua rilis di awal	Kaku terhadap pembelajaran rilis sebelumnya	Scope rilis n ditentukan setelah rilis n-1 berjalan
Menganggap “di luar scope” berarti “tidak penting”	Fitur bagus ditolak mentah-mentah	Jawab “rilis berikutnya”, bukan “tidak akan pernah”
Tidak menjawab dengan tiga kata (ya/tidak/rilis berikutnya)	Semua permintaan akhirnya dikerjakan	Gunakan tiga jawaban, catat di change log
Membiarkan developer memutuskan konflik vision	Developer disalahkan ketika hasilnya salah	Sponsor yang memutuskan, analyst memfasilitasi
Tidak membuat WBS sebagai alat negosiasi	Tidak ada dasar untuk menolak penambahan	Buat WBS sebelum menerima permintaan baru
Sunk cost fallacy: mempertahankan fitur out-of-scope karena sudah setengah jadi	Technical debt menua	Buang fitur di luar scope meskipun sudah setengah jadi
Melewatkan evaluasi setelah rilis	Scope rilis berikut tidak berbasis data	Selalu evaluasi usage setelah rilis

Tiga kesalahan yang paling sering muncul adalah **menulis daftar fitur sebagai vision**, **tidak membuat WBS sebagai alat negosiasi**, dan **tidak punya tiga jawaban untuk permintaan scope baru**. Ketiganya sama-sama membuat analyst kehilangan alat untuk menolak dengan elegan. Tanpa alat menolak, analyst hanya bisa berkata "ya". Dan kalau setiap permintaan dijawab "ya", scope berhenti menjadi konsep dan berubah menjadi daftar keinginan.

15 Lembar Rujukan Cepat

Tabel 5 merangkum istilah dan template yang wajib dikuasai sebelum pertemuan keenam.

Table 5. Lembar rujukan cepat untuk minggu kelima.

Kebutuhan atau Istilah	Definisi atau Template
Product vision	Arah jangka panjang produk, tidak berubah untuk waktu lama
Product scope	Bagian yang dikerjakan pada iterasi atau rilis tertentu
Tiga tes vision	Bertahan jika rilis dibatalkan, teknologi berubah, sponsor diganti
Tiga tes scope	Bisa selesai dalam satu rilis, bisa dipecah ke WBS, punya batas waktu
Oscillation test	Fitur masuk-keluar-masuk = vision belum cukup, bukan scope salah
Release treadmill	Vision tetap, scope tiap rilis disesuaikan dengan data dari rilis sebelumnya
Mengukur scope	Jumlah flow, rata-rata biaya per flow, jumlah stakeholder, Function Points
Vision change test	Apakah domain produk berubah, atau hanya platform dan detail?
Project Scope Management	Collect, Define, Create WBS, Verify, Control
WBS	Work Breakdown Structure, alat negosiasi scope
Is this in scope	Pertanyaan wajib setiap ada permintaan baru
Tiga jawaban scope	Ya, tidak, rilis berikutnya. Tidak ada "mungkin"
Scope creep	Pertumbuhan scope tanpa penyesuaian timeline atau anggaran
Sunk cost fallacy	Buang fitur out-of-scope meskipun sudah setengah jadi
Vision and Scope Document	Empat bagian: Business Requirement, Vision, Scope, Business Context
Conflict resolution	Sponsor yang memutuskan, analyst yang memfasilitasi
MoSCoW	Must, Should, Could, Won't untuk memetakan prioritas scope

16 Kesimpulan

Vision dan scope adalah dua konsep yang berbeda secara fundamental. Vision adalah kompas, scope adalah rute hari ini. Vision tetap, scope berubah setiap rilis berdasarkan pembelajaran. Tiga prinsip penutup:

1. **Vision yang baik lulus tiga tes.** Bertahan jika satu rilis dibatalkan, bertahan jika teknologi berubah, bertahan jika sponsor diganti. Kalau satu tes gagal, yang Anda tulis bukan vision, melainkan scope yang menyamar. Bedakan keduanya sebelum menulis lebih jauh.
2. **Scope yang baik punya tiga jawaban, bukan dua.** Setiap permintaan baru dijawab dengan "ya", "tidak", atau "rilis berikutnya". Tidak ada kata keempat. Kata "mungkin" dan "kita lihat nanti" adalah cara paling halus memulai scope creep. Analyst yang tidak punya tiga jawaban akan selalu berakhir dengan satu.

3. **WBS adalah alat negosiasi, bukan dokumen teknis.** Ketika sponsor meminta tambahan fitur, WBS memberi Anda angka konkret: "ini yang ada dalam 3 bulan, kalau tambah fitur ini, salah satu dari tiga item ini harus keluar, atau timeline bertambah". Tanpa WBS, percakapan selalu berakhir dengan "kita coba dulu", yang artinya lembur tanpa bayaran.

Pada pertemuan berikutnya, prinsip-prinsip ini akan dipakai untuk mempelajari dokumentasi SRS 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] Project Management Institute. (2021). *A Guide to the Project Management Body of Knowledge (PMBOK Guide)* (7th ed.). PMI.

[7] Cockburn, A. (2000). *Writing Effective Use Cases*. Addison-Wesley.