Memahami Stakeholder dan Aktivitas Requirement Engineering

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 minggu kedua mata kuliah Analisis dan Spesifikasi Kebutuhan Perangkat Lunak. Ia menjelaskan mengapa keterlibatan klien menentukan berhasil atau gagalnya sebuah perangkat lunak, siapa saja yang disebut stakeholder, masalah umum yang muncul dalam relasi klien dan developer, serta dua aktivitas utama requirement engineering beserta empat sub-aktivitasnya. Tujuh visualisasi dan tiga tabel menyertai penjelasan agar pembaca dapat memahami konsep tanpa menghafal. Sebuah studi kasus berisi sepuluh langkah menunjukkan artefak konkret yang harus dihasilkan pada setiap tahap, dengan contoh kasus sistem penjadwalan kuliah. Lembar rujukan cepat di akhir merangkum istilah yang wajib dikuasai sebelum pertemuan berikutnya.

1 Pendahuluan

1.1 Analogi Pembuka

Bayangkan seorang pasien masuk ke ruang dokter. Ia mengeluh demam dan pusing. Dokter yang baik akan bertanya: sudah berapa hari, apakah disertai mual, apakah ada riwayat alergi obat. Dokter yang buruk langsung menulis resep tanpa bertanya. Kalau resepnya kebetulan cocok, pasien sembuh. Kalau tidak, penyakitnya tambah parah.

Developer perangkat lunak adalah dokter itu. Klien adalah pasiennya. Requirement engineering adalah sesi tanya-jawab sebelum resep ditulis.

1.2 Mengapa Topik Ini Penting

Perangkat lunak dibuat untuk memenuhi kebutuhan pengguna. Kalau developer bekerja hanya berdasarkan asumsi, ia berisiko membangun solusi untuk masalah yang salah. Keterlibatan klien sejak awal bukan formalitas. Ia adalah satu-satunya cara memastikan perangkat lunak yang dibangun benar-benar dibutuhkan.

1.3 Peta Pembahasan

Makalah ini disusun berurutan. Bagian kedua menjelaskan mengapa keterlibatan klien tidak bisa ditawar. Bagian ketiga mendefinisikan stakeholder dan memetakannya. Bagian keempat membeberkan masalah umum relasi klien dan developer. Bagian kelima membedah dua aktivitas requirement engineering. Bagian keenam memperdalam empat sub-aktivitas development. Bagian ketujuh menyajikan studi kasus sepuluh langkah dengan sistem penjadwalan kuliah. Bagian kedelapan sampai kesepuluh merangkum perbandingan, kesalahan umum, dan lembar rujukan cepat. Bagian kesebelas menutup dengan tiga prinsip.

2 Mengapa Keterlibatan Klien Tidak Bisa Ditawar

Empat kenyataan berikut menjelaskan mengapa developer tidak boleh bekerja sendirian.

Pertama, requirement tidak boleh dibuat dari asumsi developer. Developer tidak tahu apa yang klien butuhkan kecuali klien mengatakannya.

Kedua, tim software biasanya tidak memahami proses bisnis klien. Mereka menguasai teknologi, tapi belum tentu paham bagaimana faktur dibuat, bagaimana stok dihitung, atau bagaimana jadwal kuliah disusun.

Ketiga, klien paham kebutuhannya tapi belum tentu bisa menerjemahkannya ke dokumen teknis. Klien tahu ia ingin “laporan yang cepat”, tapi tidak tahu bahwa “cepat” baginya berarti tiga detik dan bagi developer berarti dua puluh detik.

Keempat, komunikasi yang minim menghasilkan perangkat lunak yang tidak sesuai kebutuhan. Bug sering bukan karena kode salah, tapi karena kesepakatan awal tidak pernah tercapai.

Perhatikan Gambar **??**. Ini siklus yang terus berulang ketika komunikasi tidak dibangun dengan benar.

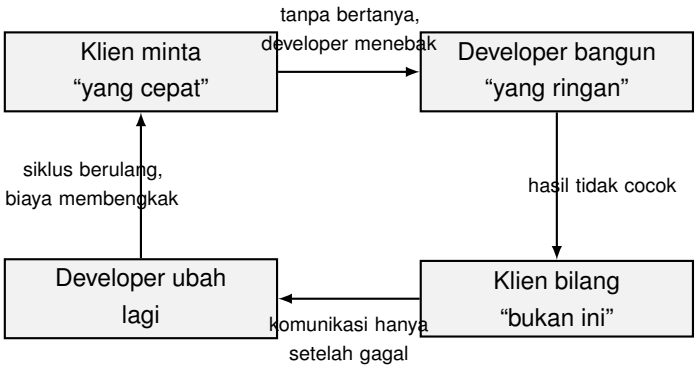


Figure 1. Siklus miskomunikasi yang terjadi ketika klien tidak dilibatkan sejak awal.

Satu-satunya cara keluar dari siklus ini adalah memindahkan percakapan ke awal. Sebelum satu baris kode ditulis, klien dan developer harus sepakat tentang apa yang akan dibangun, mengapa dibangun, dan bagaimana mengukur keberhasilannya.

3 Siapa Itu Stakeholder

Stakeholder adalah individu, kelompok, organisasi, atau siapa pun yang akan terpengaruh oleh aktivitas dan keputusan dalam sebuah proyek. Kata kuncinya adalah **terpengaruh**, bukan **terlibat**. Seorang mahasiswa yang tidak pernah menyentuh sistem penjadwalan tetap stakeholder, karena jadwal kuliahnya dihasilkan oleh sistem itu.

Perhatikan Gambar **??**. Peta ini memperlihatkan bahwa stakeholder sebuah proyek biasanya lebih banyak daripada yang diperkirakan.

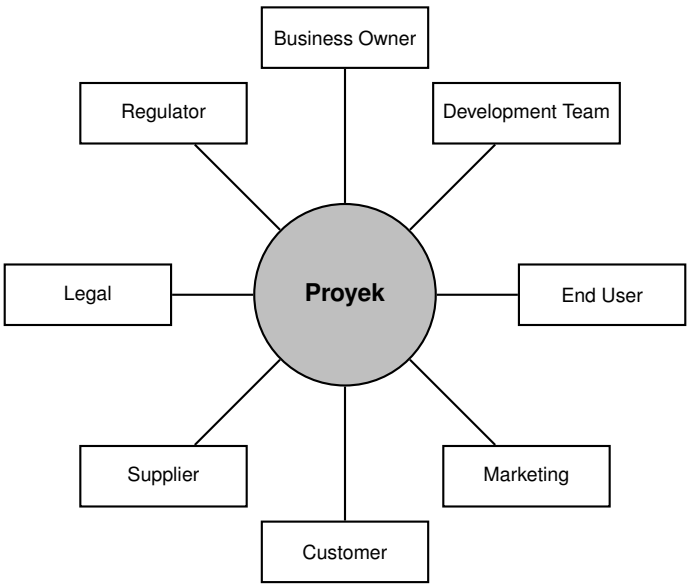


Figure 2. Peta stakeholder yang mengelilingi sebuah proyek.

Tidak semua stakeholder punya bobot yang sama. Sebagian hanya perlu diberi tahu, sebagian lain perlu dilibatkan langsung. Memetakan stakeholder berarti menentukan siapa yang masuk ke kelompok mana, dan seberapa besar pengaruhnya terhadap keputusan.

4 Masalah Umum pada Relasi Klien dan Developer

Tiga masalah berikut paling sering muncul. Perhatikan Gambar **??**.

Klien menganggap developer tahu sendiri. Tanpa banyak diskusi, klien berharap developer bisa membaca pikirannya.	**Developer tidak merasa perlu bertanya.** Ia menganggap dirinya sudah tahu apa yang user inginkan.	**Tidak ada dokumen kesepakatan.** Semua percakapan menguap begitu saja. Tidak ada jejak yang bisa diuji.

Figure 3. Tiga masalah umum dalam relasi klien dan developer.

Perangkat lunak yang sangat baik adalah hasil dari desain yang dieksekusi dengan baik

berdasarkan requirement yang sangat baik. Requirement yang sangat baik lahir dari komunikasi dan kolaborasi yang efektif antara developer dan customer. Tiga masalah di atas memutus rantai itu di titik awalnya.

5 Dua Aktivitas Requirement Engineering

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

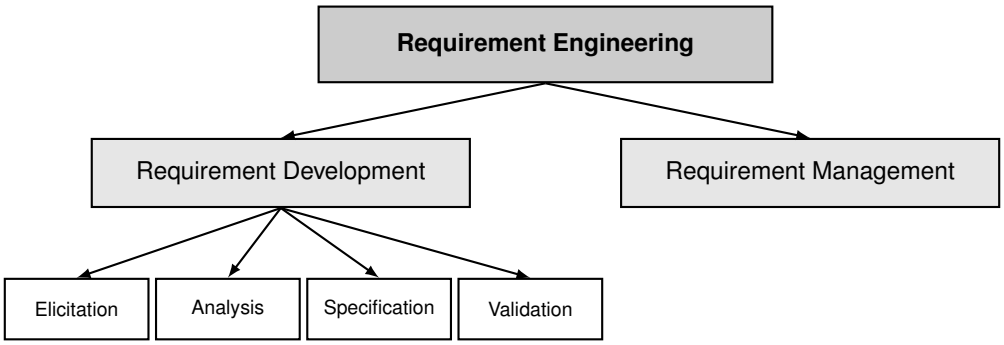


Figure 4. Dua aktivitas utama requirement engineering dan empat sub-aktivitas development.

Requirement Development mencakup semua aktivitas eksplorasi, evaluasi, dokumentasi, dan konfirmasi requirement. Ia menjawab pertanyaan: apa yang dibutuhkan.

Requirement Management mencakup semua aktivitas menjaga rencana proyek tetap selaras dengan requirement, melacak status dan perubahan, serta mengevaluasi dampak sebelum perubahan disetujui. Ia menjawab pertanyaan: bagaimana perubahannya dikendalikan.

Analogi sederhana: development adalah kegiatan memasak. Management adalah kegiatan mencatat bahan apa yang masuk, kapan kadaluwarsa, dan siapa yang menghabiskan. Keduanya diperlukan. Tanpa development, tidak ada makanan. Tanpa management, dapur berantakan dan bahan terbuang.

6 Empat Sub-Aktivitas Development

Requirement Development terdiri dari empat tahap yang berjalan berurutan. Perhatikan Gambar **??**.

$$\text{Elicitation}_{\text{bertanya}} \rightarrow \text{Analysis}_{\text{memeriksa}} \rightarrow \text{Specification}_{\text{menuliskan}} \rightarrow \text{Validation}_{\text{mengonfirmasi}}$$

Figure 5. Empat sub-aktivitas development yang berjalan berurutan.

Elicitation adalah kegiatan menggali kebutuhan. Tekniknya bisa wawancara, observasi langsung, workshop, kuesioner, atau analisis dokumen. Intinya satu: bertanya kepada stakeholder, bukan menebak.

Analysis adalah kegiatan memeriksa hasil elicitation. Apakah ada yang bertentangan? Apakah

ada yang belum jelas? Apakah ada kebutuhan yang saling meniadakan? Analisis mencari benang merah dan konflik.

Specification adalah kegiatan menuliskan hasil analisis ke dalam dokumen. Requirement yang masih berupa percakapan diubah menjadi kalimat yang bisa diuji, dengan format role-goal-benefit atau input-proses-output.

Validation adalah kegiatan mengonfirmasi dokumen kepada stakeholder. Apakah dokumen ini benar mencerminkan kebutuhan? Apakah ada yang kurang, lebih, atau salah tafsir?

Hubungan keempatnya seperti alur memasak: belanja bahan (elicitation), periksa bahan (analysis), tulis resep (specification), sajikan untuk dicicip (validation). Kalau salah satu dilewatkan, hasilnya bisa hambar atau basi.

7 Studi Kasus: Sistem Penjadwalan Kuliah

Bagian ini memandu langkah demi langkah untuk menghasilkan deliverable minggu kedua. Kasus yang dipakai: sistem penjadwalan kuliah kampus. Tujuannya mengotomasi proses penyusunan jadwal yang sebelumnya dilakukan manual oleh staf akademik.

7.1 Langkah 1: Tulis Deskripsi Singkat Sistem

Sebelum apa pun, tulis satu paragraf tentang apa yang sistem akan lakukan.

```
Deskripsi Sistem:
Sistem Penjadwalan Kuliah adalah aplikasi yang
mengotomasi proses penyusunan jadwal mata kuliah
berdasarkan ketersediaan dosen, ketersediaan ruang,
dan preferensi waktu dosen. Output-nya berupa jadwal
lengkap per semester yang bisa dilihat mahasiswa,
dosen, dan staf akademik.
```

7.2 Langkah 2: Identifikasi Stakeholder Internal

Stakeholder internal adalah mereka yang berada di dalam tim pengembangan atau organisasi yang membangun.

```
Stakeholder Internal:
1. Project Manager   - mengatur waktu dan sumber daya
2. Developer Team    - membangun dan menguji sistem
3. Business Analyst  - menggali dan mendokumentasikan
4. Project Owner     - pemilik anggaran dan keputusan
```

7.3 Langkah 3: Identifikasi Stakeholder External

Stakeholder external adalah mereka yang berada di luar tim pengembangan tapi terpengaruh hasilnya.

```
Stakeholder External:
1. Staffs (staf akademik) - pengguna harian sistem
2. Lecturers (dosen)      - pemasok data preferensi
3. Students (mahasiswa)   - penerima hasil jadwal
```

7.4 Langkah 4: Petakan Stakeholder pada Kuadran

Tidak semua stakeholder punya bobot sama. Gunakan dua sumbu: seberapa besar pengaruh mereka terhadap keputusan, dan seberapa besar dampak keputusan pada mereka.

```
Kuadran Stakeholder:

Power tinggi + Interest tinggi (Manage Closely):
- Project Owner
- Staffs (staf akademik)

Power tinggi + Interest rendah (Keep Satisfied):
- Project Manager

Power rendah + Interest tinggi (Keep Informed):
- Lecturers
- Students

Power rendah + Interest rendah (Monitor):
- Tim IT pendukung
```

7.5 Langkah 5: Tandai User Class yang Favored

Dari daftar stakeholder external, mana yang paling kritis bagi keberhasilan produk?

```
User Class (Favored):
- Staffs (favored) - memakai sistem tiap hari saat
  menyusun jadwal, jika sistem gagal mereka yang
  pertama kena imbas

User Class (Supporting):
- Lecturers - mengisi data preferensi tiap awal
  semester
- Students - hanya melihat hasil jadwal
```

7.6 Langkah 6: Rencanakan Elicitation

Buat rencana konkret: siapa yang akan ditanya, kapan, dan dengan teknik apa.

```
Rencana Elicitation:

1. Wawancara dengan Project Owner (1 jam)
   Topik: visi, batasan anggaran, target waktu rilis

2. Wawancara dengan Kepala Staf Akademik (2 jam)
   Topik: alur kerja penjadwalan saat ini, kendala,
   keluhan yang sering muncul

3. Observasi langsung di ruang staf akademik (1 hari)
   Topik: melihat langsung bagaimana jadwal disusun
   manual, dokumen apa saja yang dipakai

4. Kuesioner singkat ke 30 dosen
   Topik: preferensi waktu mengajar, konflik jadwal
   yang paling sering dialami
```

7.7 Langkah 7: Analisis Hasil Elicitation

Setelah data terkumpul, periksa apakah ada konflik atau kesenjangan.

```
Temuan Analisis:

- Kebutuhan staf akademik: jadwal tidak boleh bentrok
  ruang maupun dosen
- Kebutuhan dosen: bisa memilih preferensi waktu,
  minimal dua slot per mata kuliah
- Kebutuhan mahasiswa: jadwal bisa diakses online
  sebelum semester dimulai

Konflik yang ditemukan:
- Staf ingin jadwal selesai cepat, dosen ingin
  preferensi dihormati. Sistem harus menyeimbangkan
  dua kepentingan ini.

Kesenjangan:
- Belum ada keputusan tentang siapa yang berwenang
  mengubah jadwal setelah dipublikasikan.
```

7.8 Langkah 8: Tulis Spesifikasi Requirement

Ubah temuan analisis menjadi requirement yang bisa diuji.

```
Functional Requirement:

FR-01: Sistem menghasilkan jadwal tanpa bentrok
ruang dan dosen.
Input: data mata kuliah, data dosen, data ruang.
Output: jadwal per semester yang bebas bentrok.

FR-02: Sistem menerima preferensi waktu dari dosen.
Input: slot waktu yang dipilih dosen.
Output: jadwal dengan preferensi dosen dipenuhi
sebisa mungkin.

Non-Functional Requirement:

NFR-01: Sistem menghasilkan jadwal lengkap untuk
satu semester dalam waktu kurang dari 5 menit.

NFR-02: Jadwal bisa diakses mahasiswa lewat browser
tanpa login.
```

7.9 Langkah 9: Validasi ke Stakeholder

Sebelum masuk pengembangan, konfirmasi dokumen ke stakeholder.

```
Agenda Validasi:
1. Presentasi dokumen requirement ke Project Owner
2. Review alur kerja dengan Kepala Staf Akademik
3. Uji coba prototipe jadwal dengan 5 dosen
4. Catat semua masukan, putuskan mana yang masuk
   iterasi pertama, mana yang ditunda

Kriteria Validasi:
- Semua stakeholder favored setuju isi dokumen
- Tidak ada requirement yang saling bertentangan
- Setiap requirement bisa diuji dengan data nyata
```

7.10 Langkah 10: Tetapkan Proses Perubahan

Setelah dokumen disahkan, tetapkan bagaimana perubahan dikelola.

```
Proses Change Management:

1. Setiap usulan perubahan ditulis di form change
   request berisi: deskripsi, alasan, dampak ke
   jadwal dan anggaran
2. Business Analyst mengevaluasi dampak
```

```
3. Project Owner memutuskan terima atau tolak
4. Jika diterima, dokumen requirement diperbarui
   dan versi baru diterbitkan
5. Semua perubahan dicatat di change log

Aturan:
- Tidak ada perubahan yang dikerjakan tanpa
  persetujuan Project Owner
- Setiap perubahan harus tercatat, tidak boleh lisan
```

Sepuluh langkah di atas menghasilkan tiga artefak siap kumpul: peta stakeholder, dokumen requirement hasil elicitation dan analisis, serta catatan validasi dan proses perubahan. Inilah deliverable minggu kedua.

8 Perbandingan Ringkasan

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

Table 1. Ringkasan konsep utama minggu kedua.

Aspek	Konsep Pertama	Konsep Kedua
Lokasi stakeholder	Internal (dalam tim/organisasi)	External (di luar tim)
Bobot stakeholder	Pengaruh terhadap keputusan	Dampak keputusan pada mereka
Aktivitas RE	Development	Management
Sub-aktivitas dev	Elicitation dan analysis	Specification dan validation
Fokus aktivitas	Menggali dan memahami	Mengendalikan perubahan
User class	Favored (kritis)	Supporting (pendukung)

9 Kesalahan Umum

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

Table 2. Kesalahan umum pada pekerjaan stakeholder minggu kedua.

Kesalahan	Akibat	Perbaikan
Hanya mendaftar stakeholder tanpa memetakan bobot	Semua dianggap sama penting	Buat kuadran power-interest
Menganggap semua end user identik	Kebutuhan satu kelas terabaikan	Definisikan user class berdasarkan frekuensi dan kemampuan
Elicitation hanya lewat satu wawancara	Gambaran kebutuhan tidak lengkap	Gabungkan minimal tiga teknik: wawancara, observasi, kuesioner
Tidak mencatat hasil elicitation	Insight hilang, tidak bisa diaudit	Selalu tulis catatan lapangan dan transkrip singkat
Langsung menulis spesifikasi tanpa analisis	Requirement bertentangan lolos ke dokumen	Lakukan analisis konflik dan kesenjangan sebelum menulis
Menganggap validasi selesai setelah presentasi	Stakeholder belum benar-benar setuju	Minta tanda tangan atau persetujuan tertulis
Tidak menetapkan proses change management	Requirement berubah tanpa jejak	Tetapkan form change request dan persetujuan sponsor
Menunda dokumentasi sampai akhir	Semua percakapan menguap	Dokumentasikan sambil elicitation berlangsung

Dua kesalahan yang paling sering muncul adalah **hanya mendaftar stakeholder tanpa memetakan bobot** dan **menganggap elicitation selesai setelah satu wawancara**. Keduanya berakar pada ketergesaan yang sama: ingin cepat sampai ke pengembangan, padahal fondasinya belum kokoh.

10 Lembar Rujukan Cepat

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

Table 3. Lembar rujukan cepat untuk minggu kedua.

Kebutuhan atau Istilah	Definisi atau Artefak
Stakeholder	Siapa pun yang terpengaruh aktivitas dan keputusan proyek
Stakeholder internal	Berada di dalam tim pengembangan atau organisasi
Stakeholder external	Berada di luar tim, tapi menerima dampak keputusan
Stakeholder map	Diagram yang memetakan stakeholder dan hubungannya dengan proyek
Kuadran power-interest	Alat bantu menentukan seberapa sering stakeholder perlu dilibatkan
User class	Kelompok pengguna berdasarkan frekuensi, kemampuan, dan hak akses
User class favored	Kelas pengguna yang paling kritis bagi keberhasilan produk
User class supporting	Kelas pengguna pendukung yang tetap perlu dipenuhi kebutuhannya
Requirement development	Aktivitas eksplorasi, evaluasi, dokumentasi, konfirmasi requirement
Requirement management	Aktivitas tracking status, change control, evaluasi dampak perubahan
Elicitation	Menggali kebutuhan lewat wawancara, observasi, kuesioner, atau analisis dokumen
Analysis	Memeriksa hasil elicitation: konflik, kesenjangan, konsistensi
Specification	Menuliskan requirement dalam bentuk yang bisa diuji
Validation	Mengonfirmasi dokumen requirement ke stakeholder
Change request	Formulir usulan perubahan requirement
Change log	Catatan semua perubahan yang terjadi pada dokumen requirement

11 Kesimpulan

Stakeholder adalah siapa pun yang terpengaruh keputusan proyek, bukan hanya yang terlibat langsung. Memahami stakeholder berarti memetakan siapa mereka, seberapa besar pengaruh mereka, dan bagaimana mereka perlu dilibatkan. Requirement engineering terdiri dari dua aktivitas besar, development dan management, dengan empat sub-aktivitas development: elicitation, analysis, specification, dan validation. Tiga prinsip penutup:

1. **Libatkan, jangan asumsikan.** Setiap stakeholder yang terpengaruh keputusan berhak menyuarakan kebutuhannya. Developer tidak boleh mengisi kekosongan itu dengan tebakan.
2. **Petakan sebelum bertanya.** Sebelum wawancara dimulai, tentukan dulu siapa yang masuk kuadran mana. Waktu wawancara terbatas, jadi utamakan mereka yang pengaruh dan dampaknya paling besar.
3. **Dokumentasikan sambil berjalan.** Catatan lapangan yang ditulis saat wawancara berlangsung lebih jujur daripada rangkuman yang ditulis seminggu kemudian. Ingatan menguap, dokumen tetap.

Pada pertemuan berikutnya, prinsip-prinsip ini akan dipakai untuk mempelajari peran Business Analyst 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] Cockburn, A. (2000). *Writing Effective Use Cases*. Addison-Wesley.