

# REKAYASA PERANGKAT LUNAK KEBUTUHAN PERANGKAT LUNAK

Dr. R. Teduh Dirgahayu

[teduh.dirgahayu@uii.ac.id](mailto:teduh.dirgahayu@uii.ac.id) | [teduh.dirgahayu@gmail.com](mailto:teduh.dirgahayu@gmail.com)



**TEKNIK INFORMATIKA**  
UNIVERSITAS ISLAM INDONESIA

Di tahap awal pengembangan sistem, pengembang harus

- **Memahami masalah** yang dihadapi calon pengguna.
- **Menetapkan karakteristik sistem** yang dapat menyelesaikan masalah tersebut.

Di tahap awal pengembangan sistem, pengembang harus

- **Memahami masalah** yang dihadapi calon pengguna.
- **Menetapkan karakteristik sistem** yang dapat menyelesaikan masalah tersebut.

Selama pengembangan, pemahaman masalah dan penetapan karakteristik tersebut dapat/harus tetap berlanjut.

# Kebutuhan

Kebutuhan perangkat lunak (*software requirements*):

**Kemampuan atau karakteristik yang harus dimiliki perangkat lunak** untuk menyelesaikan masalah pengguna.

## **Standar IEEE 610.12-1990 :**

- (1) A condition or capability needed by a user to solve a problem or achieve an objective,*
  
- (2) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents,*

## Standar IEEE 610.12-1990 :

(1) *A condition or capability needed by a user to solve a problem or achieve an objective,*

(2) *A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents,*

*A documented representation of a condition or capability as in (1) and (2).*

# Macam kebutuhan

- Produk vs. proses
- Fungsional vs. non-fungsional

# Kebutuhan produk

Kemampuan/karakteristik yang harus tersedia pada sistem perangkat lunak.

Dibedakan menjadi

- Kebutuhan fungsional
- Kebutuhan non-fungsional

# Kebutuhan fungsional

Fungsionalitas atau layanan yang diberikan oleh sistem perangkat lunak untuk menyelesaikan masalah pengguna.

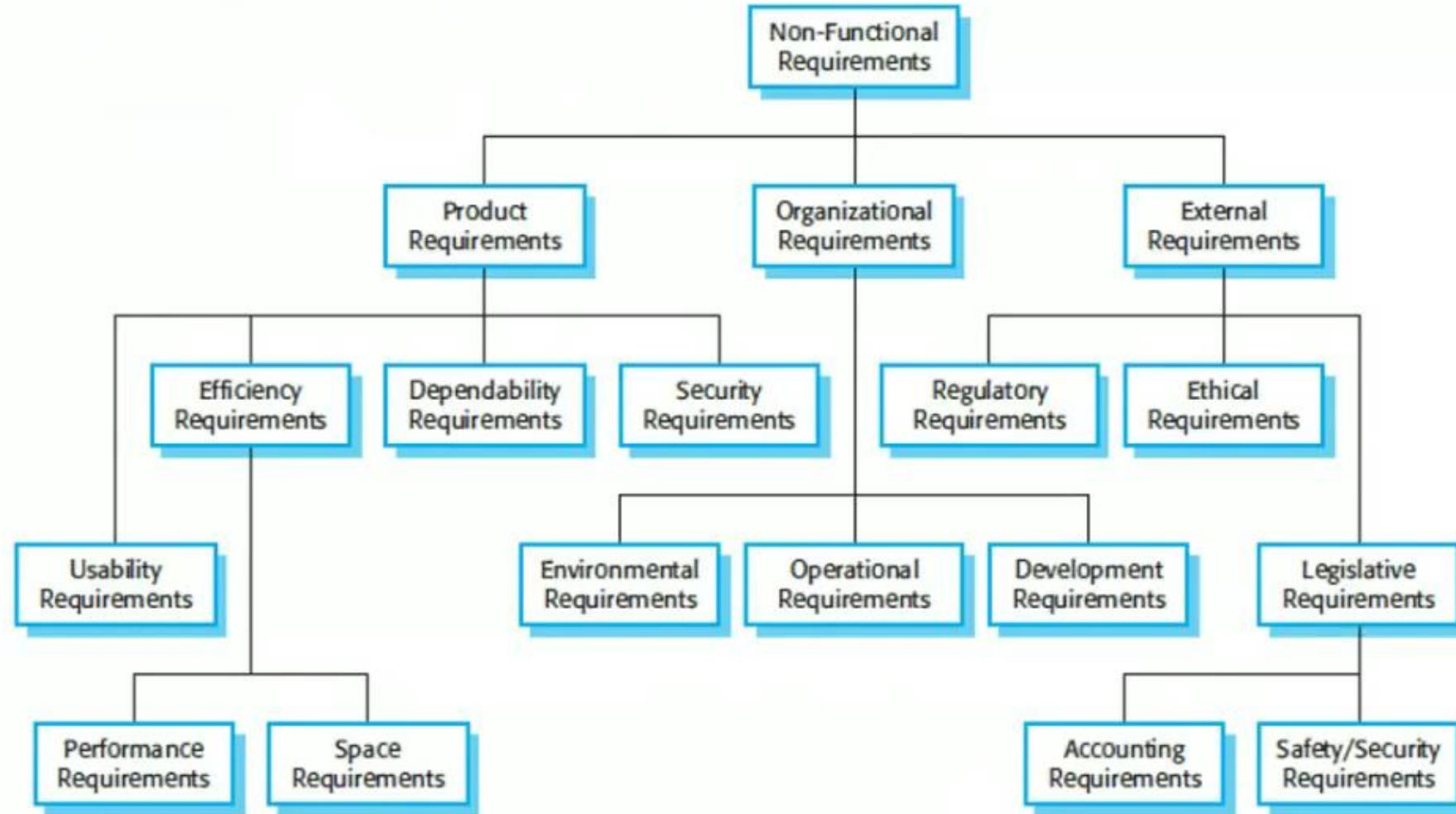
- *Sistem harus mampu **menampilkan ketersediaan dagangan**.*
- *Sistem harus mampu **membuat laporan penjualan bulanan**.*

# Kebutuhan non-fungsional

Kualitas atau kekangan sistem perangkat lunak dalam memberikan fungsionalitasnya, misal: kinerja, keamanan, dsb.

- *Sistem harus mampu menangani 300 penjualan per menit.*
- *Sistem harus menjaga bahwa transaksi dilakukan secara aman.*

Kegagalan dalam memenuhi kualitas dapat menyebabkan sistem tidak digunakan sama sekali.



# Kebutuhan proses

Hal-hal yang harus dipenuhi dalam proses pengembangan sistem perangkat lunak.

- *Pengembangan sistem harus menerapkan **metode agile**.*
- *Sistem harus dikembangkan dalam **bahasa pemrograman PHP**.*

# Analisis kebutuhan

Aktivitas dalam RPL untuk menetapkan daftar kebutuhan yang harus disediakan oleh sistem perangkat lunak.

- Tahap awal dalam RPL
- Dilakukan oleh analis sistem (*system analyst*)
- Disebut juga **rekayasa kebutuhan** (*requirements engineering*)

## Aktivitas dalam rekayasa kebutuhan:

- Elisitasi
- Prioritisasi
- Spesifikasi
- Validasi

# Referensi

Aurum, A. and Wohlin, C. (eds). **Engineering and Managing Software Requirements**. Springer, 2005.

Pressman, R.S. **Software Engineering: A Practitioner's Approach**, 7<sup>th</sup> edition. McGraw-Hill, 2010.

Sommerville, I. **Software Engineering**, 9<sup>th</sup> edition. Addison-Wesley, 2011.

# REKAYASA PERANGKAT LUNAK ELISITASI KEBUTUHAN

Dr. R. Teduh Dirgahayu

[teduh.dirgahayu@uii.ac.id](mailto:teduh.dirgahayu@uii.ac.id) | [teduh.dirgahayu@gmail.com](mailto:teduh.dirgahayu@gmail.com)



**TEKNIK INFORMATIKA**  
UNIVERSITAS ISLAM INDONESIA

# Pengantar



Manajer



Pegawai



Pemerintah



Pelanggan



Masyarakat



Sistem e-commerce

Sebelum sistem dikembangkan, **harapan/keinginan semua pihak yang berkepentingan (stakeholder) harus diidentifikasi** terlebih dahulu.

Bagaimana cara mengidentifikasi harapan stakeholder?

# Elisitasi kebutuhan

Elisitasi kebutuhan (*requirements elicitation*)

Aktivitas untuk **mencari, mengungkapkan, mendapatkan, dan merinci kebutuhan.**

- Menuntut komitmen dan kerjasama semua pihak yang berkepentingan (*stakeholder*)
- Memerlukan kemampuan komunikasi analis sistem.

## Langkah elisitasi kebutuhan

1. Memahami **ranah aplikasi** (*application domain*)
2. Mengidentifikasi **sumber kebutuhan**
3. Memilih **teknik elisitasi kebutuhan**
  
4. Melakukan elisitasi kebutuhan dari sumber yang teridentifikasi menggunakan teknik yang tepat.

## Ranah aplikasi

Mempelajari **situasi dunia nyata** di mana sistem akan diterapkan, termasuk dari aspek organisasi, politis, dan sosial.

Penyelesaian masalah harus

- mendukung tujuan bisnis organisasi
- mempertimbangkan isu politis yang berkembang
- menghormati nilai sosial yang berlaku.

# Sumber kebutuhan

Sumber kebutuhan bisa berupa

- *Stakeholder*
- Proses bisnis dan sistem yang berjalan
- Dokumentasi (manual, form, laporan, dsb.)
- Perkembangan eksternal: politik, pasar, teknologi, dsb.

# Teknik elisitasi kebutuhan

Ada banyak teknik, a.l.

- Wawancara
- Analisis ranah
- Kerja kelompok
- Observasi
- Prototyping

Teknik dapat digabungkan agar efektif.

# Wawancara

Bisa dilakukan secara

- Tak terstruktur
- Terstruktur
- Semi terstruktur

Dapat diawali terlebih dulu dengan **kuisisioner** untuk mendapatkan informasi secara cepat dari banyak sumber.

## Analisis ranah (*domain analysis*)

Mempelajari pengetahuan yang terkait permasalahan dari bermacam sumber, misal: dokumentasi proses bisnis, form, laporan, dsb.

Termasuk di dalamnya adalah

- **analisis pekerjaan** (*task analysis*): merinci pekerjaan menjadi sub-pekerjaan beserta kebutuhan masukan dan keluarannya.

# Kerja kelompok

Diskusi bersama dengan sejumlah *stakeholder*.

Dapat juga berupa

- **Brainstorming** : menyampaikan sebanyak mungkin ide kebutuhan secara cepat, tanpa mengkritisinya.
- **Joint application development (JAD)** : mendiskusikan masalah dan kemungkinan solusi, serta memutuskan solusi yang diambil.

# Observasi

Mengamati bagaimana pekerjaan biasanya dilakukan.

Dapat juga dilakukan dengan

- **Apprenticing** : analis sistem melakukan pekerjaan di bawah instruksi/supervisi.
- **Analisis protokol** : partisipan melakukan pekerjaannya sembari mengatakan apa yang sedang dikerjakan, beserta alasannya.

# Prototyping

Membuat prototipe sistem dari sejumlah kebutuhan awal.

Menunjukkan prototipe untuk

- mendapatkan konfirmasi apakah identifikasi kebutuhan awal sudah benar
- **menggali kebutuhan lain yang belum terungkap.**

# Referensi

Aurum, A. and Wohlin, C. (eds). **Engineering and Managing Software Requirements**. Springer, 2005.

Pressman, R.S. **Software Engineering: A Practitioner's Approach**, 7<sup>th</sup> edition. McGraw-Hill, 2010.

Sommerville, I. **Software Engineering**, 9<sup>th</sup> edition. Addison-Wesley, 2011.

# REKAYASA PERANGKAT LUNAK PRIORITISASI KEBUTUHAN

Dr. R. Teduh Dirgahayu

[teduh.dirgahayu@uii.ac.id](mailto:teduh.dirgahayu@uii.ac.id) | [teduh.dirgahayu@gmail.com](mailto:teduh.dirgahayu@gmail.com)



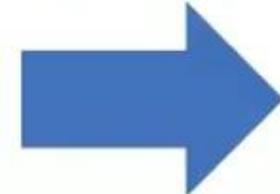
**TEKNIK INFORMATIKA**  
UNIVERSITAS ISLAM INDONESIA

# Pengantar

Hasil elisitasi kebutuhan

- Kebutuhan K1
- Kebutuhan K2
- Kebutuhan K3
- Kebutuhan K4
- Kebutuhan K5
- Kebutuhan K6
- Kebutuhan K7
- ...
- Kebutuhan K20

Kekangan  
biaya, waktu, dsb



**Prioritasi  
kebutuhan**

Hasil prioritisasi kebutuhan

1. Kebutuhan K11
2. Kebutuhan K12
3. Kebutuhan K13
4. Kebutuhan K6
5. Kebutuhan K5
6. Kebutuhan K4
7. Kebutuhan K3

# Prioritisasi kebutuhan

Prioritisasi kebutuhan (*requirements prioritization*)

Aktivitas untuk **memilih sehimpunan kebutuhan yang tepat untuk disediakan dalam kekangan yang diberikan.**

Semakin banyak aspek dan *stakeholder* yang harus dipertimbangkan, prioritisasi semakin sulit.

Bagaimana cara melakukan prioritisasi kebutuhan?

## Kegunaan prioritisasi, a.l.

- Menentukan **kebutuhan inti sistem**
- Menyeimbangkan **lingkup sistem dan kekangan proyek**
- Mengatur **harapan pengguna**
- Merencanakan **implementasi dalam rilis berurut**

# Aspek prioritisasi

Beberapa aspek umum yang biasa dipertimbangkan, a.l.

- Kepentingan tiap *stakeholder*
- Biaya, termasuk denda
- Waktu

# Skala prioritas

Prioritas kebutuhan dinyatakan dalam **skala prioritas**, yang dapat berupa

- Ordinal (urutan)
- Rasio

## Skala ordinal

Kebutuhan diurutkan mulai dari prioritas tertinggi hingga terendah.

Tidak menyatakan seberapa penting suatu kebutuhan jika dibandingkan kebutuhan yang lain

- Prioritas = K<sub>1</sub>, K<sub>2</sub>, K<sub>3</sub>, ...

## Skala rasio

Prioritas kebutuhan dinyatakan dalam nilai kepentingannya.

Menyatakan seberapa penting suatu kebutuhan jika dibandingkan kebutuhan yang lain.

- Prioritas = K1 (80), K2 (60), K3 (55), ...

# Teknik prioritisasi

Banyak teknik, a.l.

- Cummulative voting (100-dollar test)
- Numerical assignment
- Top-ten requirements

## Cummulative voting (rasio)

Jika tiap *stakeholder* diberi uang 100 dollar, bagaimana mereka akan mengalokasikannya?

	K1	K2	K3	K4
Stakeholder 1	40	30	10	20
Stakeholder 2	30	20	20	30
Stakeholder 3	30	30	10	30
<b>TOTAL</b>	<b>100</b>	<b>80</b>	<b>40</b>	<b>80</b>

- Prioritas = K1 (100), K2 dan K4 (80), K3 (40)

# Numerical assignment (ordinal)

Tiap *stakeholder* memberi nilai untuk tiap kebutuhan.

Misal: 1 = sangat penting, 2 = penting, 3 = opsional

	K1	K2	K3	K4
Stakeholder 1	2	2	1	3
Stakeholder 2	2	3	1	2
Stakeholder 3	1	1	2	3
RERATA (PEMBULATAN)	1,67 (2)	2	1,33 (1)	2,67 (3)

- Prioritas = 1: K3, 2: K1 dan K2, 3 : K4

# Top-ten requirements (rasio)

Tiap *stakeholder* memilih  $n$  kebutuhan yang paling penting.

- Misal: pilih 2 dari 4 kebutuhan

	K1	K2	K3	K4
Stakeholder 1	✓	✓		
Stakeholder 2		✓		✓
Stakeholder 3		✓		✓
DIPILIH	1	3	0	2

- Prioritas = K2, K4, K1, K3

# REKAYASA PERANGKAT LUNAK SPESIFIKASI KEBUTUHAN

Dr. R. Teduh Dirgahayu

[teduh.dirgahayu@uii.ac.id](mailto:teduh.dirgahayu@uii.ac.id) | [teduh.dirgahayu@gmail.com](mailto:teduh.dirgahayu@gmail.com)



**TEKNIK INFORMATIKA**  
UNIVERSITAS ISLAM INDONESIA

# Pengantar

Kebutuhan K1  
Kebutuhan K2  
Kebutuhan K3  
Kebutuhan K4  
Kebutuhan K5  
Kebutuhan K6

Daftar kebutuhan

Spesifikasi  
kebutuhan



Masalah, asumsi,  
model, ...



Dokumen spesifikasi

# Spesifikasi kebutuhan

Spesifikasi kebutuhan (*requirements specification*)

Aktivitas menuliskan kebutuhan dalam dokumen secara **sistematis, jelas, tak ambigu, mudah dipahami, lengkap, dan konsisten.**

Hasil berupa dokumen **spesifikasi kebutuhan perangkat lunak** (*software requirements specification, SRS*)

# Bahasa spesifikasi

Dokumen spesifikasi bisa dinyatakan dalam

- Bahasa alami
- Bahasa rancangan (*pseudo code*)
- Bahasa pemodelan (notasi grafis)
- Bahasa formal (matematis)

Dapat dikombinasikan untuk mendapatkan spesifikasi yang efektif (jelas, tak ambigu, dan mudah dipahami).

Dokumen spesifikasi dibaca oleh

- **Klien** : kesepakatan antara klien dan pengembang
- **Perancang** : dasar perancangan perangkat lunak

Bahasa harus dapat **dipahami oleh klien** (non teknis), tetapi **cukup rinci bagi perancang** untuk melakukan perancangan teknis.

# Sistematika spesifikasi

Tersedia panduan penyusunan dokumen SRS, a.l.  
**rekomendasi IEEE 830-1998**

1. Pengantar
2. Gambaran umum sistem
3. Fungsionalitas sistem
4. Kebutuhan antarmuka eksternal
5. Kebutuhan lain

## 1. Pengantar

- 1.1. Kegunaan SRS
- 1.2. Lingkup (*masalah, tujuan dan manfaat produk*)
- 1.3. Definisi dan singkatan
- 1.4. Referensi
- 1.5. Sistematika

## 2. Gambaran umum

- 2.1. Perspektif produk (*posisi produk dalam sistem besar*)
- 2.2. Fungsi produk (*secara umum*)
- 2.3. Karakteristik pengguna
- 2.4. Kekangan/batasan (*dan dampaknya pada pengembangan*)
- 2.5. Asumsi dan ketergantungan

### **3. Fungsionalitas sistem**

#### **3.1. Fungsionalitas 1**

*Deskripsi*

*Urutan langkah*

*Kebutuhan sistem*

#### **3.2. Fungsionalitas 2**

*Deskripsi*

*Urutan langkah*

*Kebutuhan sistem*

dst.

## **4. Kebutuhan antarmuka eksternal**

- 4.1. Antarmuka pengguna
- 4.2. Antarmuka perangkat keras
- 4.3. Antarmuka perangkat lunak
- 4.4. Antarmuka komunikasi

## **5. Kebutuhan lain**

- 5.1. Kinerja
- 5.2. Keamanan
- 5.3. Kehandalan
- dsb.

# Penulisan kebutuhan

Ditulis **dari sisi sistem** (penyediaan fungsionalitas)

- *Pelanggan harus dapat melihat katalog buku*
- *Sistem harus dapat menampilkan katalog produk*
  
- *Pelanggan harus mendapatkan biaya total pembelian*
- *Sistem harus dapat menghitung biaya total pembelian*

- Level kebutuhan

- **Wajib** : harus (*must, shall, required*)
- **Disarankan** : sebaiknya (*should*)
- **Opsional** : boleh (*may, optional*)

- *Sistem **harus** bisa menampilkan daftar pelatihan.*
- *Daftar pelatihan **sebaiknya** diurutkan berdasar nama pelatihan atau tanggal pelatihan.*
- *Pelatihan yang ditawarkan **boleh** menyertakan nama instruktur.*

# Referensi

Aurum, A. and Wohlin, C. (eds). **Engineering and Managing Software Requirements**. Springer, 2005.

Pressman, R.S. **Software Engineering: A Practitioner's Approach**, 7<sup>th</sup> edition. McGraw-Hill, 2010.

Sommerville, I. **Software Engineering**, 9<sup>th</sup> edition. Addison-Wesley, 2011.

# REKAYASA PERANGKAT LUNAK VALIDASI KEBUTUHAN

Dr. R. Teduh Dirgahayu

[teduh.dirgahayu@uii.ac.id](mailto:teduh.dirgahayu@uii.ac.id) | [teduh.dirgahayu@gmail.com](mailto:teduh.dirgahayu@gmail.com)



**TEKNIK INFORMATIKA**  
UNIVERSITAS ISLAM INDONESIA

# Pengantar



Manajer



Pegawai



Spesifikasi



# Validasi kebutuhan

Validasi kebutuhan (*requirements validation*)

Aktivitas untuk mengecek **apakah spesifikasi kebutuhan sesuai harapan stakeholder.**

Meminimalkan upaya koreksi pada saat tahap pengembangan berikutnya.

Apa dan bagaimana pengecekan dilakukan?

# Jenis pengecekan

- **Correctness check**

Apakah semua kebutuhan benar sesuai harapan stakeholder?

- **Completeness check**

Apakah spesifikasi kebutuhan sudah lengkap?

- **Consistency check**

Apakah spesifikasi kebutuhan dituliskan secara konsisten?

- **Feasibility check**

Apakah semua kebutuhan dapat diimplementasikan, baik dari sisi teknologi, anggaran, dan jadwal?

- **Verifiability check**

Apakah semua kebutuhan dapat diverifikasi (diuji) kebenaran implementasinya?

# Teknik validasi

Ada beberapa teknik, a.l.

- Review
- Prototyping
- Pembuatan test case

# Review

Tim reviewer membaca dan memeriksa spesifikasi.

Membaca dapat menggunakan pendekatan

- **Ad hoc** : tanpa panduan, berdasar pengalaman reviewer
- **Checklist** : disertai daftar hal yang harus diperiksa
- **Scenario** : menerapkan skenario tertentu

# Prototyping

Membuat prototipe sistem dari sejumlah kebutuhan awal.

Menunjukkan prototipe untuk

- mendapatkan konfirmasi apakah identifikasi kebutuhan awal sudah benar
- menggali kebutuhan lain yang belum terungkap.

## Pembuatan test case

Kebutuhan harus dapat diuji setelah dikonstruksi.

Jika kasus uji (*test case*) tidak dapat dibuat, kemungkinan kebutuhan tak tepat, tak jelas, tak lengkap atau tak konsisten.

- **Kebutuhan:** *Sistem harus mampu menyarankan produk belanja.*
- *Kapan saran ditampilkan? Apa dasar pemberian saran?*
- **Kebutuhan:** *Sistem harus mampu menyarankan produk belanja setelah memilih produk, berdasarkan riwayat belanja pelanggan.*

## Referensi

Aurum, A. and Wohlin, C. (eds). **Engineering and Managing Software Requirements**. Springer, 2005.

Pressman, R.S. **Software Engineering: A Practitioner's Approach**, 7<sup>th</sup> edition. McGraw-Hill, 2010.

Sommerville, I. **Software Engineering**, 9<sup>th</sup> edition. Addison-Wesley, 2011.