

# Functional Requirements (FR) dan Non-Functional Requirements (NFR)

Aryo Pinandito  
Program Vokasi Universitas Brawijaya 2012

# Functional Requirements (FR)

---

- ▶ Menggambarkan fungsionalitas sistem atau layanan-layanan sistem
- ▶ Sangat bergantung dari jenis perangkat lunak, pengguna sistem, dan jenis sistem dimana perangkat lunak tersebut digunakan
- ▶ Kebutuhan fungsional dapat berupa pernyataan-pernyataan tingkat tinggi dari:
  - ▶ Apa yang sistem harus lakukan
  - ▶ Harus dapat menggambarkan layanan-layanan yang dapat diberikan oleh sistem kepada pengguna secara mendetail



# Contoh Sistem Perpustakaan

---

- ▶ Sistem perpustakaan menyediakan antarmuka tunggal untuk mengakses artikel-artikel dalam *database* perpustakaan yang berbeda-beda
- ▶ Pengguna dapat mencari, men-*download*, dan mencetak artikel yang ditampilkan.
- ▶ Contoh FR sistem perpustakaan:
  - ▶ Pengguna harus dapat mencari di seluruh database yang ada, atau mencari di sebagian database yang disediakan.
  - ▶ Sistem harus menyediakan program yang memungkinkan penggunanya membaca artikel atau dokumen dalam perpustakaan
  - ▶ Sistem harus dapat mencatat buku-buku atau dokumen yang dipinjam oleh pengguna.



# Non-functional Requirements (NFR)

---

- ▶ Non-functional requirements atau kebutuhan non-fungsional menentukan atribut atau kualitas secara keseluruhan dari suatu sistem.
- ▶ Kebutuhan non-fungsional menempatkan batasan pada produk yang sedang dikembangkan, proses pengembangannya, dan menentukan batasan-batasan eksternal yang harus dipenuhi oleh produk tersebut.



# Beberapa Kategori NFR

---

- ▶ Keamanan (*safety* dan *security*)
- ▶ Ketergunaan (*usability*)
- ▶ Reliabilitas
- ▶ Performansi



# Contoh NFR

---

- ▶ “Sistem harus dapat memastikan bahwa data yang digunakan dalam sistem harus terlindung dari akses yang tidak berwenang.”
  - ▶ Dengan kata lain, data tidak dapat diakses oleh pengguna yang tidak berhak.
- ▶ Secara konvensional kebutuhan ini termasuk kebutuhan non-fungsional, karena tidak menyebutkan secara spesifik kebutuhan fungsional yang harus disediakan oleh sistem.
- ▶ Namun dapat dispesifikasikan lebih lanjut dengan:
  - ▶ “Sistem harus menyertakan sebuah prosedur otorisasi dimana penggunaanya harus mengidentifikasi diri dengan sebuah username dan password. Hanya pengguna yang memiliki wewenang melalui prosedur ini yang dapat mengakses data dalam sistem.”



# Jenis-jenis NFR (IEEE-Std 830-1993)

---

13 NFR yang harus disertakan dalam sebuah dokumen spesifikasi kebutuhan perangkat lunak (SRS)

- ▶ **Performance**
  - ▶ **Interface/Usability**
  - ▶ Operational
  - ▶ Resource
  - ▶ Verification
  - ▶ Acceptance
  - ▶ *\*critical system NFR*
  - ▶ Documentation
  - ▶ **Security**
  - ▶ Portability
  - ▶ Quality
  - ▶ **Reliability**
  - ▶ Maintainability
  - ▶ **Safety**
- 



## Contoh NFR

---

- ▶ Sistem layanan X harus memiliki tingkat ketersediaan 999/1000 atau 99%.
  - ▶ Hal ini merupakan kebutuhan akan reliabilitas yang berarti setiap 1000 permintaan layanan, 999 permintaan harus dapat terpenuhi.
- ▶ Sistem Y harus mampu mengolah transaksi sekurang-kurangnya 8 transaksi dalam setiap detik.
  - ▶ Hal ini merupakan sebuah kebutuhan akan performansi.
- ▶ Besarnya program sistem Z dibatasi sebesar 512 Kbytes.
  - ▶ Hal ini merupakan kebutuhan sumber daya (resource) yang menentukan ukuran *memory* maksimum sistem tersebut.





# Mengidentifikasi NFR

---

- ▶ Umumnya NFR dapat dikenali dari “keinginan”, “kebutuhan”, atau “kepentingan” dari para *stakeholders*.
- ▶ keinginan, kebutuhan, atau kepentingan tersebut umumnya bersifat non-fungsional
  - ▶ Tujuan bisnis
  - ▶ Karakteristik sistem
  - ▶ Keamanan, performa, fungsionalitas, dan perawatan sistem



# Hubungan antara kebutuhan user, kepentingan, dan NFR

---

Kebutuhan	Kepentingan	NFR
Fungsi	<ol style="list-style-type: none"><li>1. Kemudahan penggunaan</li><li>2. Otoritas Akses</li><li>3. Kemungkinan kegagalan sistem</li></ol>	<ol style="list-style-type: none"><li>1. Usability</li><li>2. Security</li><li>3. Reliability</li></ol>
Performansi	<ol style="list-style-type: none"><li>1. Penggunaan sumber daya</li><li>2. Verifikasi performa</li><li>3. Kemudahan antarmuka</li></ol>	<ol style="list-style-type: none"><li>1. Efficiency</li><li>2. Verifiability</li><li>3. Interoperability</li></ol>
Perubahan	<ol style="list-style-type: none"><li>1. Kemudahan perbaikan</li><li>2. Kemudahan perubahan</li><li>3. Kemudahan transport/lokasi penggunaan</li><li>4. Kemudahan dalam meningkatkan kapasitas dan performa</li></ol>	<ol style="list-style-type: none"><li>1. Maintainability</li><li>2. Flexibility</li><li>3. Portability</li><li>4. Expandability</li></ol>



# Reliability

---

- ▶ Memberikan batasan perilaku sistem pada saat beroperasi
  - ▶ Availability – Ketersediaan sistem dalam memberikan layanan ketika diperlukan oleh pengguna
  - ▶ Tingkat kegagalan – Seberapa sering sistem gagal untuk dapat memberikan layanan yang diharapkan oleh pengguna



# Performance

---

- ▶ Memberikan batasan mengenai kecepatan operasional sebuah sistem
  - ▶ Kebutuhan akan respon
  - ▶ Kebutuhan *throughput* (keluaran)
  - ▶ Kebutuhan akan pewaktuan (*timing*)



# Security

---

- ▶ Kebutuhan akan keamanan disertakan dalam sistem untuk menjamin:
  - ▶ Akses ke dalam sistem tanpa otorisasi tidak dimungkinkan
  - ▶ Memastikan integritas sistem dari kecelakaan atau kerusakan
- ▶ Contoh:
  - ▶ Data hanya dapat diubah oleh administrator sistem
  - ▶ Seluruh data harus di-*backup* setiap 24 jam, dan hasil *backup*-nya disimpan di lokasi yang berbeda dengan sistem
  - ▶ Seluruh komunikasi antara *client-server* harus dienkrripsi



# Usability

---

- ▶ Terkait dengan penentuan antarmuka dan interaksi pengguna dengan sistem
- ▶ User manual yang terstruktur, pesan kesalahan yang informatif, fasilitas pertolongan, dan antarmuka yang konsisten dapat meningkatkan kebutuhan *usability* ini.



# Safety

---

- ▶ Umumnya diasumsikan sebagai seluruh kebutuhan yang terkait dengan keamanan sistem
- ▶ Umumnya sangat berhubungan dengan kebutuhan untuk memastikan keamanan dalam operasional sistem
  - ▶ Kebutuhan untuk melindungi sistem
  - ▶ Kebutuhan untuk menghindari kecelakaan dalam sistem maupun dalam menggunakan sistem.
- ▶ Penggunaannya seringkali bergantung dengan budaya dan perilaku dalam organisasi



## Contoh Kebutuhan *Safety*

---

- ▶ Sistem tidak mengizinkan pengoperasian alat/perangkat kecuali terdapat petugas di lokasi
- ▶ Sistem tidak boleh memberikan obat kepada pasien dengan dosis yang lebih dari yang diizinkan oleh dokter pasien yang bersangkutan
- ▶ Sistem tidak boleh beroperasi jika suhu di luar ruangan berada di bawah 4 derajat Celcius.
- ▶ Petugas yang melaksanakan proses perawatan sistem harus menggunakan gelang anti-statik selama berada di dalam ruangan.





# Petunjuk Penulisan Requirements

---

- ▶ Gunakan format yang standar dan gunakan pada seluruh kebutuhan yang dituliskan
- ▶ Gunakan bahasa yang konsisten dan tidak membingungkan atau ambigu.
- ▶ Gunakan kata “harus” pada kondisi kebutuhan yang memang harus dipenuhi oleh sistem, dan gunakan kata “seharusnya/sebaiknya” untuk menuliskan kondisi kebutuhan yang diinginkan.
- ▶ Hindari penggunaan bahasa yang terlalu bersifat teknis dalam penulisan *requirements*.



# Be S.M.A.R.T

---

- ▶ **S**pecific
  - ▶ Spesifik dan detail
- ▶ **M**easurable
  - ▶ Dapat diukur
- ▶ **A**ttainable
  - ▶ Dapat dicapai
- ▶ **R**ealistic
  - ▶ Realistis
- ▶ **T**angible
  - ▶ Berwujud atau dapat diwujudkan



# Struktur Dokumen SRS

---

- ▶ **Pendahuluan**
  - ▶ Pengantar
  - ▶ Daftar Istilah
  - ▶ Daftar Revisi
- ▶ **Deskripsi Umum**
  - ▶ Penjelasan sistem secara umum
- ▶ **Spesifikasi Kebutuhan**
  - ▶ Arsitektur sistem
  - ▶ Spesifikasi kebutuhan sistem
    - ▶ FR dan NFR
  - ▶ Model-model sistem
  - ▶ Evolusi sistem
- ▶ **Lampiran**
- ▶ **Index**

