# UNIVERSITAS INDONESIA

**ANALISIS FAKTOR-FAKTOR YANG MENYEBABKAN KEGAGALAN APLIKASI DENGAN METODE *ANALYTIC HIERARCHY PROCESS* (AHP): STUDI KASUS PT TELKOMSEL**

# KARYA AKHIR

**ROSA RESKA RISKIANA 1506706710**

**FAKULTAS ILMU KOMPUTER**

**PROGRAM STUDI MAGISTER TEKNOLOGI INFORMASI JAKARTA**

**JANUARI 2017**

**UNIVERSITAS INDONESIA**

**ANALISIS FAKTOR-FAKTOR YANG MENYEBABKAN KEGAGALAN APLIKASI DENGAN METODE *ANALYTIC HIERARCHY PROCESS* (AHP): STUDI KASUS PT TELKOMSEL**

# KARYA AKHIR

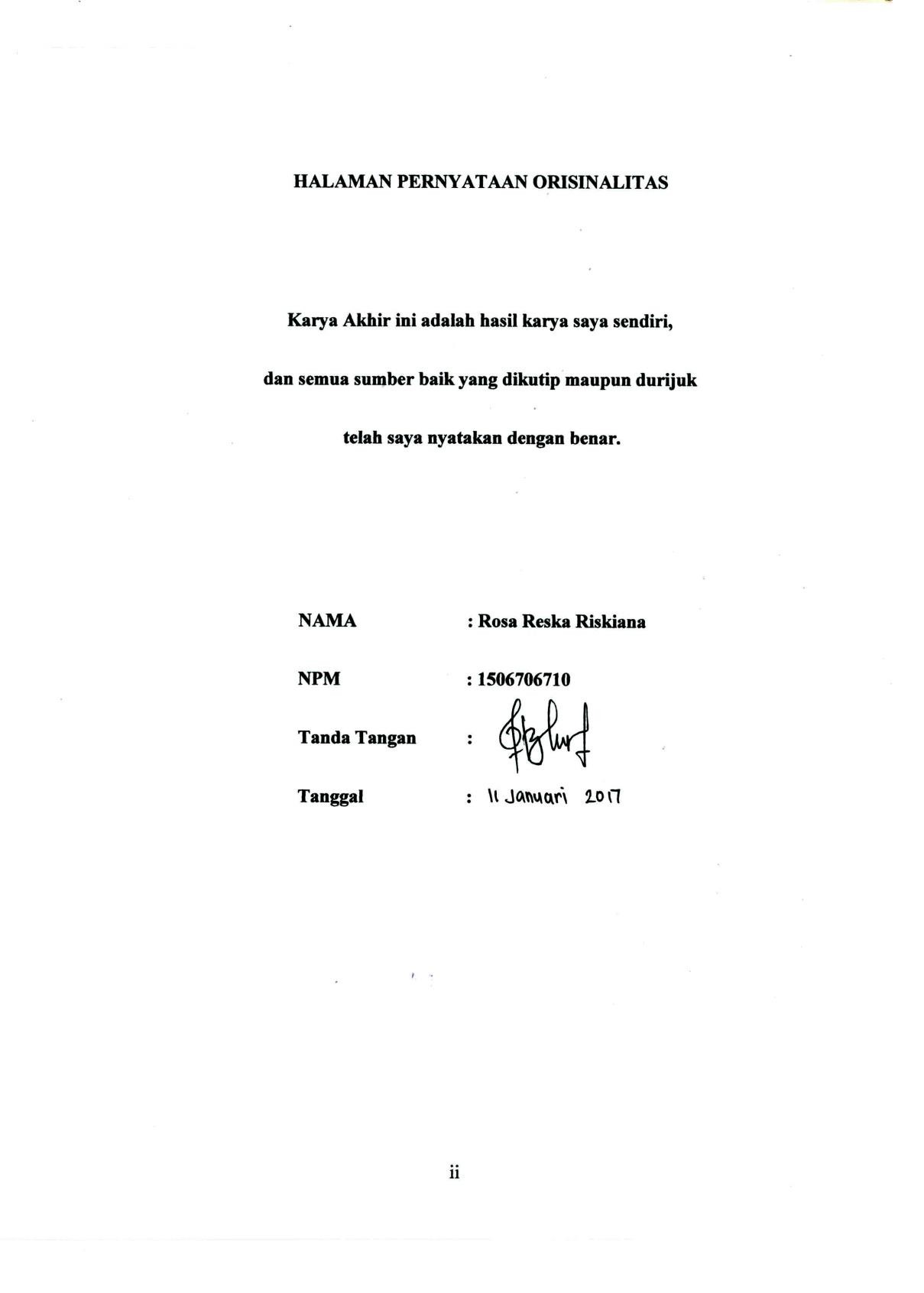
## Diajukan sebagai salah satu syarat untuk memperoleh gelar Magister Teknologi Informasi

**ROSA RESKA RISKIANA 1506706710**

**FAKULTAS ILMU KOMPUTER**

**PROGRAM STUDI MAGISTER TEKNOLOGI INFORMASI JAKARTA**

**JANUARI 2017**



## HALAMAN PERNYATAAN ORISINALITAS

**Karya Akhir ini adalah hasil karya saya sendiri, dan semua sumber baik yang dikutip maupun durijuk**

**telah saya nyatakan dengan benar.**

**NAMA : Rosa Reska Riskiana**

**NPM 1506706710**

**Tanda Tangan :**

**Tanggal :**

ii

## HALAMAN PENGESAHAN

Karya Akhir ini diajukan oleh:

Nama : Rosa Reska Riskiana

NPM : 1506706710

Program Studi : Magister Teknologi Informasi

Judul Karya Akhir : Analisis Faktor-Faktor yang Menyebabkan

Kegagalan Aplikasi dengan Metode *Analytic Hierarchy Process* (AHP): Studi Kasus PT Telkomsel

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Magister Teknologi Informasi pada Program Studi Magister Teknologi Informasi, Fakultas Ilmu Komputer, Universitas Indonesia

## DEWAN PENGUJI

Pembimbing 1 : Betty Purwandari, Ph.D ( )

Pembimbing : Riri Satria, M.M ( )

Penguji : Dana Indra Sensuse, Ph.D ( )

Penguji : Dr. Achmad Nizar Hidayanto S.Kom.,M.Kom ( )

Ditetapkan di : Jakarta

Tanggal : Januari 2017

iii

## KATA PENGANTAR/UCAPAN TERIMA KASIH

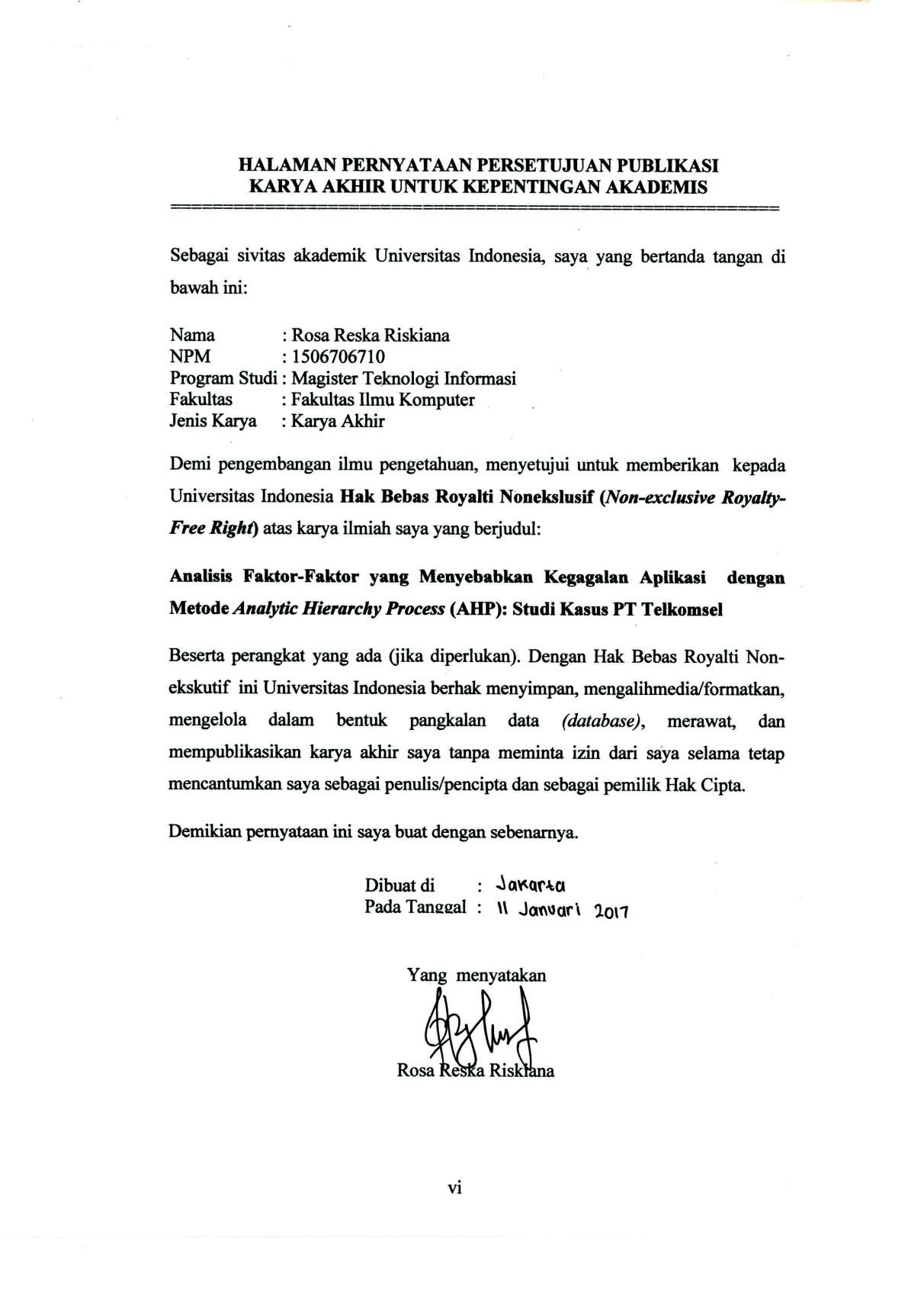
Puji syukur saya panjatkan kepada Allah Yang Maha Esa, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan Karya Akhir ini. Penulisan Karya Akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Magister Teknologi Informasi pada Program Studi Magister Teknologi Informasi, Fakultas Ilmu Komputer – Universitas Indonesia. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan karya akhir ini, sangatlah sulit bagi saya untuk menyelesaikannya. Oleh karena itu saya mengucapkan terima kasih kepada:

1. Betty Purwandari, Ph.D dan Riri Satria, M.M selaku pembimbing yang telah menyediakan banyak waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan karya akhir ini.
2. Para pakar dari PT Telkomsel yang telah banyak membantu dalam usaha memperoleh data yang saya perlukan, serta menyediakan banyak waktu untuk memberi informasi sesuai kebutuhan penelitian.
3. Orang tua saya, Fatimah Zuhro, SH,CN,MH, yang selalu memberikan bantuan dukungan material dan moral selama pengerjaan karya akhir ini.
4. Teman-teman di PT Telkomsel, M. Gentur Witcaksono, Kukuh Hernadi, Gervasius L. Kuma, Fauzi Fuadillah, Ratih R. Hutami, Norlela Handayani, Willias P Sihutang, Novia Liestarini, Nur Baithya, Ludfi Eka Lesmana, dan Wahyu Setiawan yang sudah membantu memberikan masukan pada kekurangan-kekurangan yang ada pada karya akhir ini serta membantu menjembatani kepada pakar yang akan menjadi narasumber penelitian.
5. Teman-teman di MTI UI 2015SB, Yahya, Askarudin Tamam, M. Khanifan, dan Chandra, yang telah menemani perjuangan menempuh perkuliahan selama 3 semester, yang juga sangat banyak membantu memberikan dukungan serta meringankan beban dalam pengerjaan tugas perkuliahan.
6. Sahabat saya, Prayoga Seto Anungputro yang telah banyak membantu saya dalam mencari jurnal-jurnal yang saya butuhkan untuk karya akhir ini.
7. Hidayatulloh, yang selalu memberikan dukungan moral selama pengerjaan karya akhir ini.

Akhir kata, saya berharap Allah Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga karya akhir ini membawa manfaat bagi pengembangan ilmu.

Jakarta, Januari 2017

Penulis



## HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI KARYA AKHIR UNTUK KEPENTINGAN AKADEMIS

==========================================================

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Rosa Reska Riskiana

NPM 1506706710

Program Studi : Magister Teknologi Informasi Fakultas : Fakultas Ilmu Komputer Jenis Karya : Karya Akhir

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Nonekslusif (*Non-exclusive Royalty- Free Right*)** atas karya ilmiah saya yang berjudul:

**Analisis Faktor-Faktor yang Menyebabkan Kegagalan Aplikasi dengan Metode *Analytic Hierarchy Process* (AHP): Studi Kasus PT Telkomsel**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non- ekskutif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data *(database)*, merawat, dan mempublikasikan karya akhir saya tanpa meminta izin dari saya selama tetap mencantumkan saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Pada Tanggal :

Yang menyatakan

Rosa Reska Riskiana

vi

## ABSTRAK

Nama :Rosa Reska Riskiana Program Studi :Magister Teknologi Informasi

Judul : Analisis Faktor-Faktor yang Menyebabkan Kegagalan Aplikasi dengan Metode *Analytic Hierarchy Process* (AHP): Studi Kasus PT Telkomsel

Saat ini, hampir semua kegiatan bisnis mempunyai ketergantungan yang cukup tinggi terhadap Teknologi Informasi (TI) khususnya untuk PT Telkomsel yang merupakan salah satu perusahaan telekomunikasi terdepan di Indonesia. Namun, data dari IT Service Performance Management (IT SPM) menunjukkan pada tahun 2015 masih terdapat insiden pada aplikasi yang mengindikasikan bahwa aplikasi tersebut belum berkualitas yang mengakibatkan tidak tercapainya target *success rate* PT Telkomsel.

Untuk mengatasi hal tersebut, hal yang perlu dilakukan adalah mengetahui faktor penyebab dari insiden aplikasi tersebut dari perspektif kegagalan aplikasi. Karya Akhir ini memperingkatkan penyebab kegagalan aplikasi dengan menggunakan AHP *(Analytic Hierarchy Process)*. Dari penelitian ini, didapatkan tujuh faktor kegagalan aplikasi yang paling berpotensi atau sering terjadi di PT Telkomsel, yaitu (1) *Computational/Logic Error,* (2) *Insider Threat,* (3) *Improper Testing of Change,* (4) *Configuration Error*, (5) *Network Hardware Failure,* (6) *Password Disclosure,* (7) *Denial of Service Attack.*

Kata Kunci : *Software Quality,* Insiden, Kegagalan Aplikasi, AHP

## ABSTRACT

Name :Rosa Reska Riskiana

Study Program :Magister Teknologi Informasi

Title :Analysis of Factors Contributing to Software Failure using *Analytic Hierarchy Process (*AHP): A case study of PT Telkomsel

It is common for modern businesses to be highly dependent on Information Technology (IT). This includes PT Telkomsel, which is one of the leading telecommunication companies in Indonesia. However, the 2015 data from the PT Telkomsel IT Service Performance Management (IT SPM) showed frequent application incidents. This indicates that PT Telkomsel applications were not good enough to achieve the company’s target of success rate.

In order to address this problem, it is crucial to identify the causes of these application incidents from software failure perspectives. This study aims to analyze the causes and rank them using Analytic Hierarchy Process (AHP). The investigation results that there are seven most potential or frequent factors causing the software failures. They are: (1) Computational/Logic Error, (2) Insider Threat,

(3) Improper Testing of Change, (4) Configuration Error, (5) Network Hardware Failure (6) Password Disclosure, (7) Denial of Services Attack.

Keyword : *Software Quality, Incident, Software Failure*, AHP

## DAFTAR ISI

HALAMAN JUDUL i

[HALAMAN PERNYATAAN ORISINALITAS ii](#_bookmark0)

[HALAMAN PENGESAHAN iii](#_bookmark1)

[KATA PENGANTAR/UCAPAN TERIMA KASIH iv](#_bookmark2)

[HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI KARYA AKHIR](#_bookmark3) [UNTUK KEPENTINGAN AKADEMIS vi](#_bookmark3)

[ABSTRAK vii](#_bookmark4)

[ABSTRACT viii](#_bookmark5)

[DAFTAR ISI ix](#_bookmark6)

[DAFTAR TABEL xi](#_bookmark7)

[DAFTAR GAMBAR xiii](#_bookmark8)

[DAFTAR LAMPIRAN xvi](#_bookmark9)

[BAB 1 PENDAHULUAN 1](#_bookmark10)

* 1. [Latar Belakang 1](#_bookmark11)
  2. [Perumusan Masalah 2](#_bookmark13)
  3. [Tujuan Penelitian 9](#_bookmark20)
  4. [Batasan Masalah 9](#_bookmark21)
  5. [Manfaat Penelitian 9](#_bookmark22)

[BAB 2 STUDI LITERATUR 11](#_bookmark23)

* 1. [Profil Perusahaan 11](#_bookmark25)
     1. [Struktur Organisasi PT Telkomsel 12](#_bookmark26)
     2. [Sejarah Perkembangan Telkomsel 14](#_bookmark29)
     3. [Strategi Bisnis Perusahaan 17](#_bookmark30)
     4. [Strategi Teknologi Informasi 21](#_bookmark31)
  2. [Landasan Teori 23](#_bookmark32)
     1. [*Bugs, faults, errors, failures, incidents* 23](#_bookmark33)
     2. [Aplikasi 25](#_bookmark34)
     3. [*Software Quality* 26](#_bookmark35)
     4. [Perumusan Kriteria Kegagalan Aplikasi 33](#_bookmark42)
     5. [Faktor-Faktor Penyebab Kegagalan Aplikasi 34](#_bookmark44)
     6. [Perumusan Faktor-Faktor Penyebab Kegagalan Aplikasi 36](#_bookmark45)
  3. [Metode Pemeringkatan Faktor-Faktor 38](#_bookmark47)
     1. [*Multi Criteria Decision Making Method* 38](#_bookmark48)
     2. [Perumusan Metode Pemeringkatan Faktor-Faktor Penyebab Kegagalan Aplikasi 40](#_bookmark49)
     3. [AHP *(Analytic Hierarchical Process)* 42](#_bookmark51)
  4. [Penelitian Sebelumnya 47](#_bookmark56)
     1. [*Causes of Failure in Web Applications* (Pertet dan Narasimhan, 2005)47](#_bookmark57)
     2. [*Identifying Software Quality Factors for Tellecomunication Industri in* Malaysia (Hamid dan Hasan, 2011) 48](#_bookmark58)
     3. [*Modelling Software Quality Evaluation Using The Analytical Hierarchy* Process (Yoon et al, 2011) 49](#_bookmark60)
     4. [*Case Studies of Most Common and Severe Types of Software System* Failure (Dalal dan Chhillar, 2012) 51](#_bookmark62)
     5. [*Insider Threat Behaviour Factors: A Comparison of Theory with* Reported Incidents (Munshi, Dell, dan Armstrong, 2012) 53](#_bookmark64)
     6. [*Comparative Study of Software Quality Model* (Youness, Abdelaziz, dan Habib, 2013) 54](#_bookmark65)
     7. [*Quality and Human Errors in IT Service Infrastructure* (Saaralainen dan Jantti, 2015) 56](#_bookmark67)
     8. [Perbandingan Antara Penelitian Terdahulu dan Penelitian Saat Ini 57](#_bookmark68)
  5. [*Theoretical Framework* 61](#_bookmark70)

[BAB 3 METODOLOGI PENELITIAN 65](#_bookmark72)

* 1. [Rancangan Penelitian 65](#_bookmark73)
  2. [Tahapan Penelitian 67](#_bookmark75)
  3. [Instrumen Penelitian 71](#_bookmark77)
  4. [Metode dan Proses Olah Data 76](#_bookmark81)

[BAB 4 ANALISIS 79](#_bookmark82)

* 1. [Analisis Variabel dalam *Theoretical Framework* 79](#_bookmark83)
     1. [Analisis Kriteria Kegagalan Aplikasi 80](#_bookmark84)
     2. [Analisis Faktor Penyebab Kegagalan Aplikasi 82](#_bookmark86)
     3. [Analisis dan Perancangan Struktur Hirarki AHP 84](#_bookmark88)
  2. [Analisis Kuisioner Perbandingan Berpasangan 89](#_bookmark92)
  3. [Analisis Hasil Kuisioner 90](#_bookmark93)
  4. [Analisis Perhitungan Matriks Perbandingan Berpasangan 92](#_bookmark96)
     1. [Bobot Kriteria Aplikasi Penyebab Kegagalan Aplikasi 93](#_bookmark97)
     2. [Bobot Kategori Faktor Penyebab Kegagalan Aplikasi 94](#_bookmark102)
     3. [Bobot Kriteria Faktor 106](#_bookmark131)
  5. [Analisis Prioritisasi Faktor Penyebab Kegagalan Aplikasi Aplikasi 117](#_bookmark162)
     1. [Hasil Urutan Berdasarkan Nilai Eigen 117](#_bookmark163)
  6. [Implikasi 133](#_bookmark169)
     1. [Implikasi Teori 133](#_bookmark170)
     2. [Implikasi Praktikal 134](#_bookmark171)

[BAB 5 KESIMPULAN dan SARAN 136](#_TOC_250004)

* 1. [Kesimpulan 136](#_TOC_250003)
  2. [Saran 137](#_TOC_250002)

[DAFTAR PUSTAKA 138](#_TOC_250001)

[LAMPIRAN 143](#_TOC_250000)

## DAFTAR TABEL

[Tabel 1.1 Tabel target *Success Rate* SLA PT. Telkomsel tahun 2015 3](#_bookmark14)

[Tabel 1.2 Tabel *Success Rate* Aplikasi Saat Terjadi Insiden (2015) 6](#_bookmark17)

[Tabel 1.3 Dampak dari Insiden Aplikasi yang Termasuk Kuadran *Key*](#_bookmark19)[*Operational* 7](#_bookmark19)

Tabel 1.3 Dampak dari Insiden Aplikasi yang Termasuk Kuadran *Key Operational* (sambungan) 8

[Tabel 2.1 Tabel Perbandingan Antar *Quality Model* 32](#_bookmark43)

[Tabel 2.2 Pemetaan faktor-faktor penyebab insiden 36](#_bookmark46)

[Tabel 2.3 Tabel perbandingan metode Multi Criteria Decision Making 41](#_bookmark50)

[Tabel 2.4 Matriks Perbandingan Berpasangan 44](#_bookmark53)

[Tabel 2.5 Tabel Skala Matriks Perbandingan Berpasangan 44](#_bookmark54)

[Tabel 2.6Indeks Acak (Random Index) 46](#_bookmark55)

[Tabel 2.7 Kesesuaian antara ISO 9126 dan AHP 49](#_bookmark61)

[Tabel 2.8 Daftar beberapa kegagalan *aplikasi* dan tingkat severity dari kegagalan 52](#_bookmark63)

[Tabel 2.9 Perbandingan model kualitas 55](#_bookmark66)

[Tabel 2.10 Tabel Perbandingan antara penelitian sebelumnya dan penelitian saat](#_bookmark69)

[ini 58](#_bookmark69)

[Tabel 3.1 Tabel *research design* 66](#_bookmark74)

[Tabel 3.2 Tabel skala penilaian Saaty 73](#_bookmark78)

[Tabel 3.3 Indikator Pengukuran Variabel Penelitian 74](#_bookmark79)

[Tabel 3.4 Indikator Pengukuran Variabel Pemeringkatan Faktor 76](#_bookmark80)

[Tabel 4.1 Tabel rangkuman evaluasi kriteria *software quality* 81](#_bookmark85)

[Tabel 4.2 Tabel rangkuman evaluasi faktor-faktor penyebab insiden aplikasi 82](#_bookmark87)

[Tabel 4.3 Hasil analisis faktor penyebab insiden aplikasi 85](#_bookmark89)

[Tabel 4.4 Hasil analisis kriteria software yang dapat menyebabkan insiden 88](#_bookmark90)

[Tabel 4.5 Perbandingan Skala Prioritas untuk Kriteria 91](#_bookmark94)

[Tabel 4.6 Perhitungan Nilai Rata-Rata Geometris - Kriteria 92](#_bookmark95)

[Tabel 4.7 Hasil Pembulatan Rata-Rata Geometris untuk Kriteria Aplikasi](#_bookmark98) [Penyebab Insiden 93](#_bookmark98)

[Tabel 4.8 Pembulatan Rata-Rata Geometris Kategori Faktor Terhadap Kriteria](#_bookmark103)

[*Unextendibility* 95](#_bookmark103)

[Tabel 4.9 Pembulatan Rata-Rata Geometris Kategori Faktor Terhadap Kriteria](#_bookmark107)

[*Inflexibility* 96](#_bookmark107)

[Tabel 4.10 Pembulatan Rata-Rata Geometris Kategori Faktor Terhadap Kriteria](#_bookmark111)

[*Unfunctionality* 98](#_bookmark111)

[Tabel 4.11 Pembulatan Rata-Rata Geometris Kategori Faktor Terhadap Kriteria](#_bookmark115)

[*Unusability* 99](#_bookmark115)

[Tabel 4.12 Pembulatan Rata-Rata Geometris Kategori Faktor Terhadap Kriteria](#_bookmark119)

[*Untestability* 101](#_bookmark119)

[Tabel 4.13 Pembulatan Rata-Rata Geometris Kategori Faktor Terhadap Kriteria](#_bookmark123)

[*Unreliability* 103](#_bookmark123)

[Tabel 4.14 Pembulatan Rata-Rata Geometris Kategori Faktor Terhadap Kriteria](#_bookmark127)

[*Unintegrity* 105](#_bookmark127)

[Tabel 4.15 Pembulatan Rata-Rata Geometris Faktor Penyebab Insiden pada](#_bookmark132) [Kategori *Software Failure* 106](#_bookmark132)

[Tabel 4.16 Pembulatan Rata-Rata Geometris Faktor Penyebab Insiden pada](#_bookmark136) [Kategori *Human/Operator Error* 108](#_bookmark136)

[Tabel 4.17 Pembulatan Rata-Rata Geometris Faktor Penyebab Insiden pada](#_bookmark140) [Kategori *Hardware/Environmental Error* 109](#_bookmark140)

[Tabel 4.18 Pembulatan Rata-Rata Geometris Faktor Penyebab Insiden pada](#_bookmark144) [Kategori *Fault Recovery* 111](#_bookmark144)

[Tabel 4.19 Pembulatan Rata-Rata Geometris Faktor Penyebab Insiden pada](#_bookmark147) [Kategori *Security Violations* 112](#_bookmark147)

[Tabel 4.20 Pembulatan Rata-Rata Geometris Faktor Penyebab Insiden pada](#_bookmark151) [Kategori *Change Management* 113](#_bookmark151)

[Tabel 4.21 Pembulatan Rata-Rata Geometris Faktor Penyebab Insiden pada](#_bookmark155) [Kategori *Communication Management* 115](#_bookmark155)

[Tabel 4.22 Pembulatan Rata-Rata Geometris Faktor Penyebab Insiden pada](#_bookmark158) [Kategori *Institusional Pressure* 116](#_bookmark158)

[Tabel 4.23 Urutan Kriteria Aplikasi Penyebab Insiden 118](#_bookmark164)

[Tabel 4.24 Urutan Kategori Faktor terhadap setiap Kriteria 119](#_bookmark165)

[Tabel 4.25 Urutan Faktor Penyebab Insiden pada Setiap Kategori 121](#_bookmark166)

[Tabel 4.26 Urutan Faktor Penyebab Insiden secara Global 124](#_bookmark168)

## DAFTAR GAMBAR

[Gambar 1.1 Persentase *Modern Resolution for All Projects (On Time, On Budget,*](#_bookmark12)[*with a satisfaktory result)* 2](#_bookmark12)

[Gambar 1.2 Jumlah Insiden per Aplikasi Tahun 2015 4](#_bookmark15)

[Gambar 1.3 Portofolio Aplikasi McFarlan PT. Telkomsel yang Terkena Insiden 4](#_bookmark16)

[Gambar 1.4 Target SLA dan realita *success rate* PT Telkomsel (2015*)* 6](#_bookmark18)

[Gambar 2.1 *Mind Map* Penelitian 11](#_bookmark24)

[Gambar 2.2 Struktur Organisasi PT Telkomsel 12](#_bookmark27)

[Gambar 2.3 Struktur Organisasi Direktorat TI Tahun 2015 14](#_bookmark28)

[Gambar 2.4 McCall’s faktor model 27](#_bookmark36)

[Gambar 2.5 Evans & Marciniak Model 28](#_bookmark37)

[Gambar 2.6 Deutsch & Willis *quality* model 29](#_bookmark38)

[Gambar 2.7 ISO 9126 *quality* model 30](#_bookmark39)

[Gambar 2.8 ISO 25000 (SQuaRE) model 30](#_bookmark40)

[Gambar 2.9 Hamid & Hasan quality model for Telecommunication Industri 31](#_bookmark41)

[Gambar 2.10 Struktur Hierarki AHP 43](#_bookmark52)

[Gambar 2.11 Proses penentuan faktor kualitas 49](#_bookmark59)

[Gambar 2.12 *Theoretical Framework* 61](#_bookmark71)

[Gambar 3.1 Tahapan Penelitian 67](#_bookmark76)

[Gambar 4.1 Struktur AHP penelitian 89](#_bookmark91)

[Gambar 4.2 Tabel Penilaian Kriteria Aplikasi Penyebab Insiden pada Expert](#_bookmark99) [Choice 11 93](#_bookmark99)

[Gambar 4.3 Nilai Eigen dan Konsistensi Kriteria Aplikasi Penyebab Insiden 94](#_bookmark100)

[Gambar 4.4 Nilai Eigen Kriteria Penyebab Insiden (*Sort by priority*) 94](#_bookmark101)

[Gambar 4.5 Tabel Penilaian Kategori Faktor Penyebab Insiden Terhadap Kriteria](#_bookmark104)

[*Unextendibility* pada Expert Choice 11 95](#_bookmark104)

[Gambar 4.6 Nilai Eigen dan Konsistensi Kategori Faktor Penyebab Insiden](#_bookmark105) [Terhadap Kriteria *Unextendibility* 95](#_bookmark105)

[Gambar 4.7 Nilai Eigen Kategori Faktor terhadap Kriteria *Unextendibility* (*Sort by*](#_bookmark106)[*priority*) 96](#_bookmark106)

[Gambar 4.8 Tabel Penilaian Kategori Faktor Penyebab Insiden Terhadap Kriteria](#_bookmark108)

[*Inflexibility* pada Expert Choice 11 97](#_bookmark108)

[Gambar 4.9 Nilai Eigen dan Konsistensi Kategori Faktor Penyebab Insiden](#_bookmark109) [Terhadap Kriteria *Inflexibility* 97](#_bookmark109)

[Gambar 4.10 Nilai Eigen Kategori Faktor terhadap Kriteria *Inflexibility* (*Sort by*](#_bookmark110)[*priority*) 97](#_bookmark110)

[Gambar 4.11 Tabel Penilaian Kategori Faktor Penyebab Insiden Terhadap Kriteria](#_bookmark112)

[*Unfunctionality* pada Expert Choice 11 98](#_bookmark112)

[Gambar 4.12 Nilai Eigen dan Konsistensi Kategori Faktor Penyebab Insiden](#_bookmark113) [Terhadap Kriteria *Unfunctionality* 99](#_bookmark113)

[Gambar 4.13 Nilai Eigen Kategori Faktor terhadap Kriteria *Unfunctionality* (*Sort*](#_bookmark114)

[*by priority*) 99](#_bookmark114)

[Gambar 4.14Tabel Penilaian Kategori Faktor Penyebab Insiden Terhadap Kriteria](#_bookmark116)

[*Unusability* pada Expert Choice 11 100](#_bookmark116)

[Gambar 4.15 Nilai Eigen dan Konsistensi Kategori Faktor Penyebab Insiden](#_bookmark117) [Terhadap Kriteria *Unusability* 100](#_bookmark117)

[Gambar 4.16 Nilai Eigen Kategori Faktor terhadap Kriteria *Unusability* (*Sort by*](#_bookmark118)[*priority*) 101](#_bookmark118)

[Gambar 4.17Tabel Penilaian Kategori Faktor Penyebab Insiden Terhadap Kriteria](#_bookmark120)

[*Untestability* pada Expert Choice 11 102](#_bookmark120)

[Gambar 4.18 Nilai Eigen dan Konsistensi Kategori Faktor Penyebab Insiden](#_bookmark121) [Terhadap Kriteria *Untestability* 102](#_bookmark121)

[Gambar 4.19 Nilai Eigen Kategori Faktor terhadap Kriteria *Untestability* (*Sort by*](#_bookmark122)[*priority*) 102](#_bookmark122)

[Gambar 4.20Tabel Penilaian Kategori Faktor Penyebab Insiden Terhadap Kriteria](#_bookmark124)

[*Unreliability* pada Expert Choice 11 103](#_bookmark124)

[Gambar 4.21 Nilai Eigen dan Konsistensi Kategori Faktor Penyebab Insiden](#_bookmark125) [Terhadap Kriteria *Unreliability* 104](#_bookmark125)

[Gambar 4.22 Nilai Eigen Kategori Faktor terhadap Kriteria *Unreliability* (*Sort by*](#_bookmark126)[*priority*) 104](#_bookmark126)

[Gambar 4.23Tabel Penilaian Kategori Faktor Penyebab Insiden Terhadap Kriteria](#_bookmark128)

[*Unintegrity* pada Expert Choice 11 105](#_bookmark128)

[Gambar 4.24 Nilai Eigen dan Konsistensi Kategori Faktor Penyebab Insiden](#_bookmark129) [Terhadap Kriteria *Unintegrity* 105](#_bookmark129)

[Gambar 4.25 Nilai Eigen Kategori Faktor terhadap Kriteria *Unintegrity* (*Sort by*](#_bookmark130)[*priority*) 106](#_bookmark130)

[Gambar 4.26Tabel Penilaian Faktor Penyebab Insiden pada Kategori *Software*](#_bookmark133)[*Failure* di Expert Choice 11 107](#_bookmark133)

[Gambar 4.27 Nilai Eigen dan Konsistensi Faktor Penyebab Insiden pada Kategori](#_bookmark134)

[*Software Failure* 107](#_bookmark134)

[Gambar 4.28 Nilai Eigen Faktor Penyebab Insiden pada Kategori *Software*](#_bookmark135)[*Failure* (*Sort by priority*) 107](#_bookmark135)

[Gambar 4.29Tabel Penilaian Faktor Penyebab Insiden pada Kategori](#_bookmark137)

[*Human/Operator Error* di Expert Choice 11 108](#_bookmark137)

[Gambar 4.30 Nilai Eigen dan Konsistensi Faktor Penyebab Insiden pada Kategori](#_bookmark138)

[*Human/Operator Error* 108](#_bookmark138)

[Gambar 4.31 Nilai Eigen Faktor Penyebab Insiden pada Kategori](#_bookmark139)

[*Human/Operator Error* (*Sort by priority*) 109](#_bookmark139)

[Gambar 4.32Tabel Penilaian Faktor Penyebab Insiden pada Kategori](#_bookmark141)

[*Hardware/Environmental Error* di Expert Choice 11 110](#_bookmark141)

[Gambar 4.33 Nilai Eigen dan Konsistensi Faktor Penyebab Insiden pada Kategori](#_bookmark142)

[*Hardware/Environmental Error* 110](#_bookmark142)

[Gambar 4.34 Nilai Eigen Faktor Penyebab Insiden pada Kategori](#_bookmark143)

[*Hardware/Environmental Error* (*Sort by priority*) 110](#_bookmark143)

[Gambar 4.35Tabel Penilaian Faktor Penyebab Insiden pada Kategori *Fault*](#_bookmark145)[*Recovery* 111](#_bookmark145)

[Gambar 4.36 Nilai Eigen dan Konsistensi Faktor Penyebab Insiden pada Kategori](#_bookmark146)

[*Fault Recovery* 111](#_bookmark146)

[Gambar 4.37Tabel Penilaian Faktor Penyebab Insiden pada Kategori *Seurity*](#_bookmark148)[*Violations* di Expert Choice 11 112](#_bookmark148)

[Gambar 4.38 Nilai Eigen dan Konsistensi Faktor Penyebab Insiden pada Kategori](#_bookmark149)

[*Security Violations* 113](#_bookmark149)

[Gambar 4.39 Nilai Eigen Faktor Penyebab Insiden pada Kategori *Security*](#_bookmark150)[*Violations* (*Sort by priority*) 113](#_bookmark150)

[Gambar 4.40Tabel Penilaian Faktor Penyebab Insiden pada Kategori *Change*](#_bookmark152)[*Management* di Expert Choice 11 114](#_bookmark152)

[Gambar 4.41 Nilai Eigen dan Konsistensi Faktor Penyebab Insiden pada Kategori](#_bookmark153)

[*Change Management* 114](#_bookmark153)

[Gambar 4.42 Nilai Eigen Faktor Penyebab Insiden pada Kategori *Change*](#_bookmark154)[*Management* (*Sort by priority*) 114](#_bookmark154)

[Gambar 4.43Tabel Penilaian Faktor Penyebab Insiden pada Kategori](#_bookmark156)

[*Communication Management* di Expert Choice 11 115](#_bookmark156)

[Gambar 4.44 Nilai Eigen dan Konsistensi Faktor Penyebab Insiden pada Kategori](#_bookmark157)

[*Software Failure* 115](#_bookmark157)

[Gambar 4.45Tabel Penilaian Faktor Penyebab Insiden pada Kategori *Institusional*](#_bookmark159)[*Pressure* di Expert Choice 11 116](#_bookmark159)

[Gambar 4.46 Nilai Eigen dan Konsistensi Faktor Penyebab Insiden pada Kategori](#_bookmark160)

[*Institusional Pressure* 116](#_bookmark160)

[Gambar 4.47 Nilai Eigen Faktor Penyebab Insiden pada Kategori *Institusional*](#_bookmark161)[*Pressure* (*Sort by priority*) 117](#_bookmark161)

[Gambar 4.48 Peringkat Faktor Penyebab Insiden Secara Global 123](#_bookmark167)

## DAFTAR LAMPIRAN

Lampiran 1 Service Level Agreement (SLA) PT Telkomsel 143

Lampiran 2 Daftar Insiden Tahun 2015 144

Lampiran 3 Transkrip Wawancara Tahap 1 (Pakar 1) 145

Lampiran 4 Transkrip Wawancara Tahap 1 (Pakar 2) 149

Lampiran 5 Transkrip Wawancara Tahap 1 (Pakar 3) 150

Lampiran 6 Kerangka AHP pada Expert Choice 11 159

Lampiran 7 Kuisioner Pairwise Comparison 168

Lampiran 8 Hasil Kuisioner Pairwise Comparison 203

Lampiran 9 Perhitungan Rata-Rata Geometris 214

Lampiran 10 Transkrip Wawancara Tahap 2 (Pakar 4) 226

Lampiran 11 Transkrip Wawancara Tahap 2 (Pakar 5) 231

## BAB 1 PENDAHULUAN

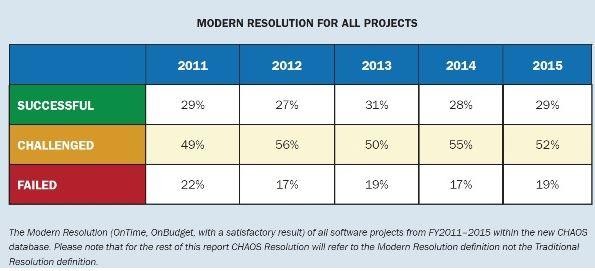
Bab ini membahas tentang latar belakang penelitian, perumusan masalah, tujuan, manfaat dan batasan penelitian.

## Latar Belakang

Sebuah aplikasi dikatakan berkualitas bila aplikasi tersebut sesuai dengan kebutuhan konsumen, sesuai dengan kegunaan produk, dan jumlah *defect* yang ditemukan (Olaperi dan Misra, 2015). Hal ini menunjukkan bahwa aplikasi yang sesuai dengan kebutuhan pengguna adalah aplikasi yang sesuai dengan spesifikasi yang sudah dinyatakan dalam spesifikasi kebutuhan, dan aplikasi yang sesuai kegunaannya adalah aplikasi yang kegunaannya persis sama seperti keinginan pengguna (Olaperi dan Misra, 2015).

*Software Quality Management* adalah suatu disiplin yang memastikan bahwa aplikasi yang digunakan memiliki kualitas yang baik, baik dari segi kesesuaian dengan ekspektasi konsumen maupun proses bisnisnya. Saat ini aplikasi dapat ditemukan dimana-mana, baik dalam sektor pemerintahan, pendidikan, serta bisnis. Melihat dari definisi tersebut maka tujuan dari aktivitas ini adalah mendefinisikan dan menjaga agar *software process* dan *software plan* tetap terkontrol.

Berdasarkan 2015 CHAOS Report yang dipublikasikan oleh The Standish Group, pada tahun 2015 terdapat 19% proyek dari 50000 proyek yang diteliti dari seluruh penjuru dunia dari perangkat kecil sampai implementasi *system re-engineering* besar yang mengalami kegagalan dalam *software development*. Persentase tersebut naik 2% dari tahun 2014 dimana proyek yang dinilai *failed* sekitar 17%. Gambar 1.1 merangkum hasil proyek selama lima tahun terakhir menggunakan definisi baru dari faktor keberhasilan *(on time, on budget, with a satisfactory result).*



**Gambar 1.1 Persentase *Modern Resolution for All Projects (On Time, On Budget, with a satisfaktory result)***

The Standish Group, 2015

Hasil dari CHAOS Report tahun 2015, mengindikasikan bahwa masih banyak yang harus diperbaiki dalam *System Development Life Cycle*. Terkait laporan tersebut, menjadi tantangan tersendiri bagi perusahaan yang bergerak di bidang *Information and Communication Technology* (ICT) untuk tetap menjaga kualitas produk yang diberikan kepada pengguna.

Kegagalan proyek TI tersebut dapat menyebabkan kehilangan secara finansial bahkan dapat membahayakan kelangsungan hidup perusahaan (Subiyakto dan Ahlan, 2013). Mishra, et al (2011) menyatakan adanya kebutuhan untuk mengidentifikasi faktor yang paling kritikal terhadap kesuksesan proyek yang relevan terhadap waktu saat ini dan kondisi suatu lokasi. Namun tidak hanya faktor yang mempengaruhi kesuksesan sebuah proyek saja yang harus diketahui, sebaliknya faktor yang paling kritikal dalam membuat kegagalan proyek juga harus diketahui. Oleh karena itu, diperlukan telaah dalam memahami faktor-faktor yang mempengaruhi kegagalan dalam proyek.

## Perumusan Masalah

Saat ini hampir semua kegiatan bisnis mempunyai ketergantungan yang cukup tinggi terhadap Teknologi Informasi (TI) (Johnson, et al, 2007). PT. Telkomsel yang merupakan salah satu perusahaan telekomunikasi terdepan di Indonesia, memiliki visi untuk menjadi penyedia layanan dan solusi *mobile digital lifestyle* kelas dunia yang terpercaya dengan memberikan layanan dan solusi *mobile digital*

yang melebihi ekspektasi pelanggan, memberikan nilai tambah kepada para

*stakeholder*, dan mendukung pertumbuhan ekonomi bangsa.

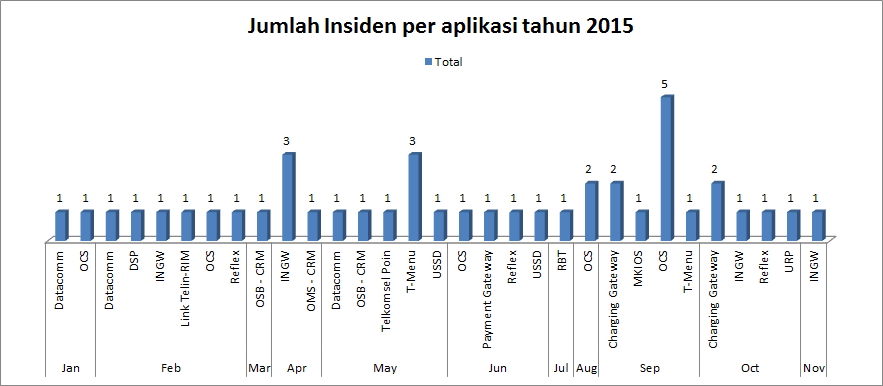
Boehm dan Basili (2001) menyatakan bahwa menemukan dan memperbaiki *software problem* setelah *delivery* seringkali 100 kali lebih mahal daripada menemukan dan memperbaikinya selama fase *requirement and design*. Dokumen IT Service Catalog tahun 2015 mendefinisikan terdapat 4 kategori *Service Level Agreement* (SLA) di PT. Telkomsel, yaitu *Mission Critical*, *Business Critical*, *Business Improvement*, dan *Business Support* yang masing-masing memiliki target *success rate* (SR) untuk setiap servis. Detail mengenai aplikasi terkait servis untuk setiap kategori SLA dapat dilihat pada tabel Lampiran 1. Untuk merangkum target *success rate* setiap kategori, dapat dilihat pada Tabel 1.1.

**Tabel 1.1 Tabel target *Success Rate* SLA PT. Telkomsel tahun 2015**

|  |  |
| --- | --- |
| **Kategori SLA** | **Target Pencapaian (*Success rate*)** |
| *Mission Critical* | 99,99 % |
| *Business Critical* | 99,90 % |
| *Business Improvement* | 99% |
| *Business Support* | 90% |

IT Service Catalog PT. Telkomsel – telah diolah kembali

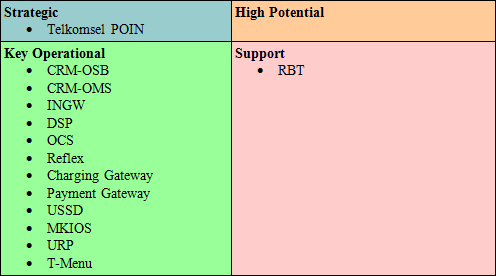
Setiap servis memiliki berbagai aplikasi yang mendukung jalannya servis tersebut. Masing-masing servis tersebut memiliki target seperti yang telah disebutkan pada Tabel 1.1. Namun, data dari IT Service Performance Management (IT SPM) menunjukkan pada tahun 2015 masih terdapat insiden. Gambar 1. 2 menunjukkan jumlah insiden yang terjadi pada tahun 2015.



## Gambar 1.2 Jumlah Insiden per Aplikasi Tahun 2015

IT Service Performance Management, 2015 – telah diolah kembali

Dari Gambar 1.2 tersebut terlihat bahwa insiden masih terjadi di sepanjang tahun 2015. *Systems and Software Engineering Vocabulary* (2010) mendefinisikan insiden merupakan kejadian yang bukan merupakan standar operasi layanan yang akan atau mungkin menyebabkan interupsi, atau pengurangan kualitas layanan. Pada tahun 2015, terdapat 16 aplikasi yang mengalami insiden. Masing-masing aplikasi tersebut memiliki perannya tersendiri dalam mendukung jalannya bisnis perusahaan. Untuk dapat mengetahui aplikasi terkait servis mana yang paling kritikal dalam mendukung bisnis, maka penulis melakukan pemetaan portfolio aplikasi dengan analisis McFarlan (1984) yang dapat dilihat pada Gambar 1.3:



## Gambar 1.3 Portofolio Aplikasi McFarlan PT. Telkomsel yang Terkena Insiden

Berdasarkan portofolio aplikasi PT. Telkomsel tersebut, 12 aplikasi yang termasuk dalam kuadran *Key Operational* mengalami insiden pada tahun 2015, yang terdiri dari CRM-OSB, CRM-OMS, INGW, OCS, DSP, Reflex, Charging Gateway, Payment Gateway, USSD, Mkios, URP, dan T-Menu. Kemudian, 1 aplikasi yang termasuk kuadran *Strategic* mengalami insiden, yaitu Telkomsel poin. Sedangkan untuk kuadran *High Potential* tidak terdapat aplikasi yang mengalami insiden, dan untuk kuadran *Support* memiliki 1 aplikasi, yaitu RBT. Menurut Ward dan Peppard (2002), portofolio aplikasi McFarlan dapat digolongkan menjadi 4 kategori, yaitu:

1. *Support*, sistem informasi yang berfungsi sebagai pendukung fungsi organisasi, artinya organisasi tidak tergantung pada peranan sisitem informasi

/ teknologi informasi. Sistem informasi / teknologi informasi ini tidak memiliki keunggulan kompetitif organisasi.

1. *Key Operational, s*istem informasi tidak secara langsung memberikan keunggulan kompetitif kepada organisasi, tetapi keberadaannya mutlak diperlukan.
2. *High Potential,* sistem informasi yang secara langsung memberikan keunggulan kompetitif kepada organisasi dan memberikan nilai potensi untuk organisasi untuk masa yang akan datang.
3. *Strategic, s*istem informasi secara signifikan memberikan nilai strategis kepada organisasi, dengan tidak adanya sistem ini organisasi tidak hanya kehilangan keunggulan kompetitif tetapi juga beresiko untuk mengalami kerugian.

Aplikasi yang masuk ke kuadran *Key Operational* merupakan aplikasi yang sangat penting untuk menjaga kelangsungan proses bisnis perusahaan. Adanya insiden pada aplikasi yang termasuk dalam kuadran ini akan menyebabkan pengurangan kualitas dan mengakibatkan kerugian bagi perusahaan. Pengurangan kualitas layanan tersebut dapat diukur dengan *success rate.* Berikut merupakan persentase *success rate* dari aplikasi-aplikasi tersebut saat terjadi insiden.

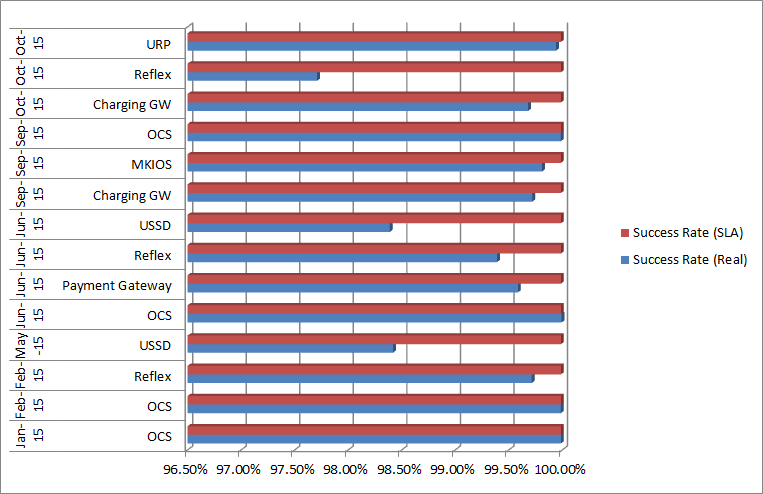
## Tabel 1.2 Tabel *Success Rate* Aplikasi Saat Terjadi Insiden (2015)

|  |  |  |
| --- | --- | --- |
| **Month** | **Application** | **Success**  **Rate** |
| Jan-15 | OCS | 99.99% |
| Feb-15 | OCS | 99.99% |
| Feb-15 | Reflex | 99.71% |
| May-15 | USSD | 98.42% |
| Jun-15 | OCS | 99.99% |
| Jun-15 | Payment Gateway | 99.59% |
| Jun-15 | Reflex | 99.39% |
| Jun-15 | USSD | 98.39% |
| Sep-15 | Charging GW | 99.72% |
| Sep-15 | MKIOS | 99.81% |
| Sep-15 | OCS | 99.99% |
| Oct-15 | Charging GW | 99.68% |
| Oct-15 | Reflex | 97.71% |
| Oct-15 | URP | 99.94% |

IT Service Performance Management, 2015

Kesenjangan yang terjadi antara realita dan target SLA Telkomsel untuk aplikasi yang termasuk dalam kuadran *Key Operational* dapat digambarkan pada Gambar

1.4 berikut:



## Gambar 1.4 Target SLA dan realita *success rate* PT Telkomsel (2015)

Gambar 1.4 di atas menampilkan kesenjangan antara target dan realita *success rate* aplikasi saat terjadi insiden untuk aplikasi-aplikasi yang termasuk dalam kuadran *Key Operaitonal*. Dampak dari masing-masing insiden tersebut akan dijelaskan dalam Tabel 1.3:

## Tabel 1.3 Dampak dari Insiden Aplikasi yang Termasuk Kuadran *Key Operational*

|  |  |  |
| --- | --- | --- |
| **Bulan** | **Aplikasi** | **Dampak pada Konsumen** |
| Januari | OCS | Konsumen tidak dapat melakukan registrasi Flash  Small Deno, SMS Mania, SMS Gift |
| Februari | DSP | Konsumen tidak dapat melakukan registrasi Flash  dan BIS Prepaid |
| Februari | Link Telin  RIM | Konsumen tidak dapat melakukan registrasi BIS  Prepaid |
| Februari | OCS | Konsumen tidak dapat melakukan registrasi AS Jagoan Serbu, Talk Mania, Dual Mode, Flash Small  Deno, dan Transfer Pulsa |
| Februari | Reflex | Konsumen tidak dapat melakukan registrasi BIS  Prepaid, Flash Small Deno, dan Opera Mini |
| Mei | USSD | Beberapa konsumen tidak dapat mengakses USSD  ADN (888,100,363,999,567) |
| Juni | OCS | Recharge ( Mkios Regional 1= 88%, Mkios Regional  10=91% ) |
| Juni | Payment Gateway | Finnet postpaid (SR = 0.06%), SMS Banking, Payment Gateway, HRIS dan Aplikasi Call Center  Outbound (Link komunikasi mengalami kegagalan) |
| Juni | Reflex | Konsumen tidak dapat melakukan aktivasi paket  VAS (AS Jagoan Serbu, Big TM, Talk Mania, SMS Mania, Wow SMS, Dual Mode dan Flash MDS |
| Juni | USSD | Beberapa konsumen di area Sumatera tidak dapat  mengakses AND (888,100,363,999,567) |
| September | Charging  Gateway | Konsumen tidak bisa melakukan registrasi Flash  Volume, Talk Mania, dan As Jagoan SMS Double |

**Tabel 1.3 Dampak dari Insiden Aplikasi yang Termasuk Kuadran *Key Operational* (sambungan)**

|  |  |  |
| --- | --- | --- |
| **Bulan** | **Aplikasi** | **Dampak pada Konsumen** |
| September | MKIOS | Konsumen tidak dapat melakukan isi ulang melalui  MKIOS |
| September | OCS | * Konsumen tidak dapat melakukan panggilan ke luar, tidak dapat menerima panggilan dan SMS * Beberapa konsumen di Pekanbaru (Regional 1   dan Regional 10) tidak dapat melakukan isi ulang melalui MKIOS |
| Oktober | Charging Gateway | * Beberapa konsumen tidak dapat melakukan registrasi dan aktivasi Nelpon Jagoan Serbu, Talk Mania, Jagoan Serbu SMS, Flash Volume, dan BIS Prepaid * Konsumen tidak dapat melakukan aktivasi   Broadband, Voice, dan SMS Package |
| Oktober | Reflex | Konsumen tidak dapat melakukan registrasi Talk  Mania, Nelpon Jagoan Serbu Malam, Jagoan Serbu Mala Double, dan FSD |
| Oktober | URP | Konsumen tidak dapat melakukan isi ulang dari  Merchant dan ATM |

Sesuai dengan program transformasi Telkomsel (Telkomsel 2.0) dan fokus tahun 2015 yaitu “*WOW! Customer Experience*” dengan tekanan pada kepentingan untuk menyediakan pengalaman yang hebat pada konsumen saat menggunakan layanan dan produk dari Telkomsel serta strategi bisnis dan strategi TI yang ingin membuat transformasi dari Telco ke Digico tersebut tercapai, sudah seharusnya aplikasi-aplikasi yang berada pada kuadran *Key Operational* ini tidak mengalami insiden. Bila aplikasi yang termasuk kuadran ini mengalami kegagalan, maka dampak pada konsumen akan benar-benar terasa dan tidak dipungkiri Telkomsel dapat mengalami kerugian. Adapun dengan adanya insiden ini mengindikasikan bahwa aplikasi tersebut belum sepenuhnya dapat memenuhi definisi aplikasi berkualitas yaitu sesuai dengan kebutuhan konsumen, sesuai dengan kegunaan produk, dan jumlah *defect* yang ditemukan seidkit (Olaperi dan Misra, 2015).

Dari papaparan di atas, sangatlah penting bagi pihak manajemen untuk mengetahui faktor-faktor apa yang menyebabkan kegagalan aplikasi yang

mengindikasikan bahwa aplikasi tersebut belum berkualitas, serta prioritas faktor tersebut khususnya pada aplikasi-aplikasi yang paling kritikal dalam proses bisnis PT Telkomsel sehingga manajemen dapat mengambil keputusan mengenai strategi yang akan diambil untuk mencapai target SLA tersebut. Untuk itu penulis menentukan *Research Question* yang relevan dengan topik Karya Akhir ini, yaitu **“Bagaimanakah prioritas dari faktor-faktor yang menyebabkan kegagalan aplikasi terkait kualitas aplikasi di PT Telkomsel?”**

## Tujuan Penelitian

Terkait dengan permasalahan yang telah dikemukakan di atas, maka tujuan dari Karya Akhir ini adalah mengetahui dan mengidentifikasi prioritas dari faktor- faktor yang menyebabkan kegagalan aplikasi yang mengindikasikan bahwa aplikasi tersebut belum berkualitas yang mengakibatkan insiden di PT Telkomsel yang diharapkan dapat membantu manajemen mengambil keputusan terkait strategi untuk mencapai target *success rate* layanan sehingga dapat membantu PT. Telkomsel mempertahankan posisinya di industri telekomunikasi.

## Batasan Masalah

Penelitian ini dibatasi dalam ruang lingkup :

* 1. Penelitian yang dilakukan dibatasi hanya pada insiden yang terjadi di PT Telkomsel pada tahun 2015
  2. Penelitian yang dilakukan dibatasi hanya pada 7 Aplikasi pada kuadran *Key Operational* yang mengalami insiden pada 2015, yaitu OCS, Reflex, USSD, Payment Gateway, Charging Gateway, MKIOS, URP.
  3. Penelitian yang dilakukan dibatasi hanya untuk kualitas aplikasi di perusahaan telekomunikasi

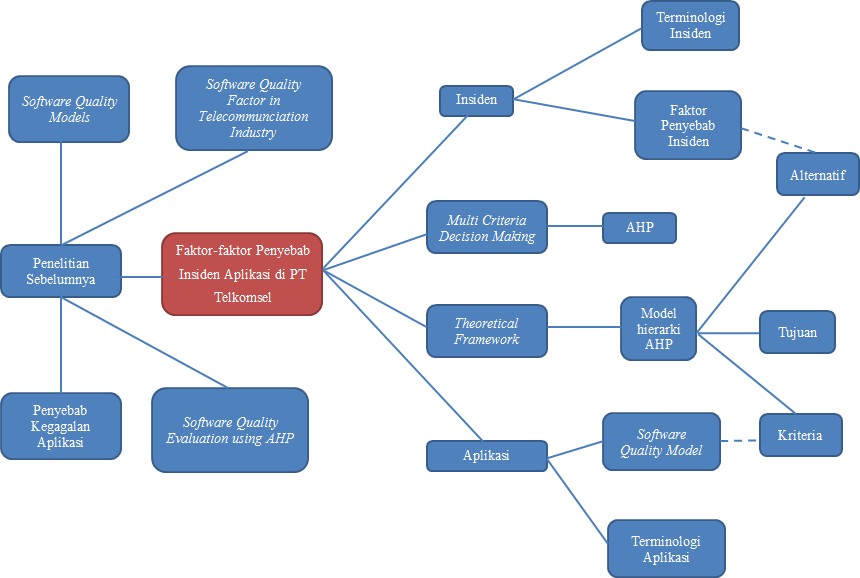
## Manfaat Penelitian

Manfaat dari penelitian ini adalah :

1. Bagi akademisi, diharapkan penelitian ini dapat dijadikan referensi untuk penelitian selanjutnya di bidang *Software Quality Management* dengan memaparkan tentang faktor-faktor yang menyebabkan suatu aplikasi tidak memenuhi kriteria berkualitas (*Critical Failure Factor*)
2. Bagi Direktorat TI khusunya Departemen IT Service Management di PT. Telkomsel, diharapkan penelitian ini dapat membantu pihak manajemen dalam mengambil keputusan terkait strategi yang akan dilakukan untuk mengantisipasi, meminimalisir bahkan menghilangkan insiden yang menyebabkan penurunan kualitas layanan yang nantinya berdampak pada tercapainya Service Level Agreement (SLA) PT. Telkomsel.

## BAB 2 STUDI LITERATUR

Bab 2 membahas tentang teori yang relevan terhadap *Research Question*, penelitian-penelitian sebelumnya, metode yang digunakan, dan kerangka teoritis yang akan menjadi dasar penelitian. Struktur pembahasan pada bab ini akan dipetakan dalam sebuah *Mind Map* berikut :



**Gambar 2.1 *Mind Map* Penelitian**

## Profil Perusahaan

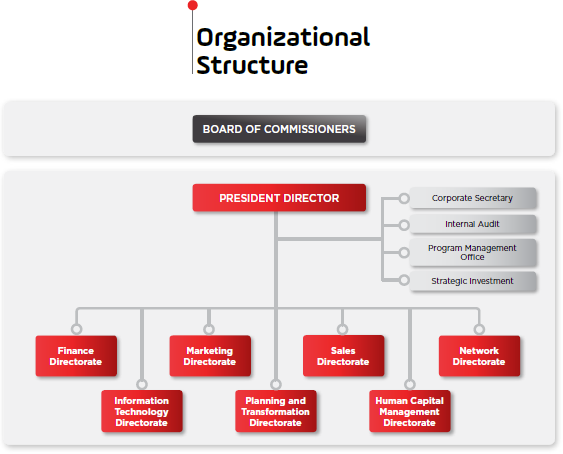
Telkomsel didirikan pada tahun 1995 sebagai wujud semangat inovasi untuk mengembangkan telekomunikasi Indonesia yang terdepan. Saat ini Telkomsel merupakan operator selular terbesar di Indonesia dan memiliki jaringan terluas yang mampu menjangkau lebih dari 95% populasi Indonesia di seluruh penjuru nusantara untuk melayani kebutuhan komunikasi berbagai lapisan masyarakat. Telkomsel menjadi pelopor untuk berbagai teknologi telekomunikasi selular di Indonesia, termasuk yang pertama meluncurkan layanan *roaming* internasional dan layanan 3G di Indonesia. Telkomsel merupakan operator yang pertama kali melakukan ujicoba teknologi jaringan pita lebar LTE. Di kawasan Asia,

Telkomsel menjadi pelopor penggunaan energi terbarukan untuk menara-menara *Base Transceiver Station* (BTS). Keunggulan produk dan layanannya menjadikan Telkomsel sebagai pilihan utama pelanggan di seluruh Indonesia.

Telkomsel memiliki 4 produk dan servis utama, yaitu kartuHalo,simPATI, Kartu As, dan LOOP. KartuHALO diposisikan sebagai merek pascabayar pilihan di segmen pelanggan professional dan korporat. KartuHALO memberikan layanan *full mobile* dan hak istimewa. Telkomsel memperluas cakupan segmen kartuHalo untuk merangkul segmen kaum muda dan memposisikan kartuHalo sebagai kartu pascabayar yang terbaik untuk professional muda. SimPATI merupakan kartu prabayar untuk segmen kelas menengah. Sedangkan kartuAS diposisikan sebagai kartu prabayar yang terjangkau, dan merupakan merk prabayar paling terkenal yang memberikan nilai terbaik untuk konsumen. LOOP diluncurkan pertama kali pada 9 Maret 2014 yang diposisikan sebagai kartu prabayar yang dikhususkan untuk segmen kalangan muda.

## Struktur Organisasi PT Telkomsel

Direksi dan Dewan Komisaris Telkomsel ditetapkan oleh pemegang saham PT Telekomunikasi Selular (Telkomsel) yaitu PT Telekomunikasi Indonesia (Persero) Tbk. (Telkom) dan Singapore Telecommunications Ltd. (SingTel). Struktur Organisasi PT Telkomsel dapat dilihat pada Gambar 2.2:



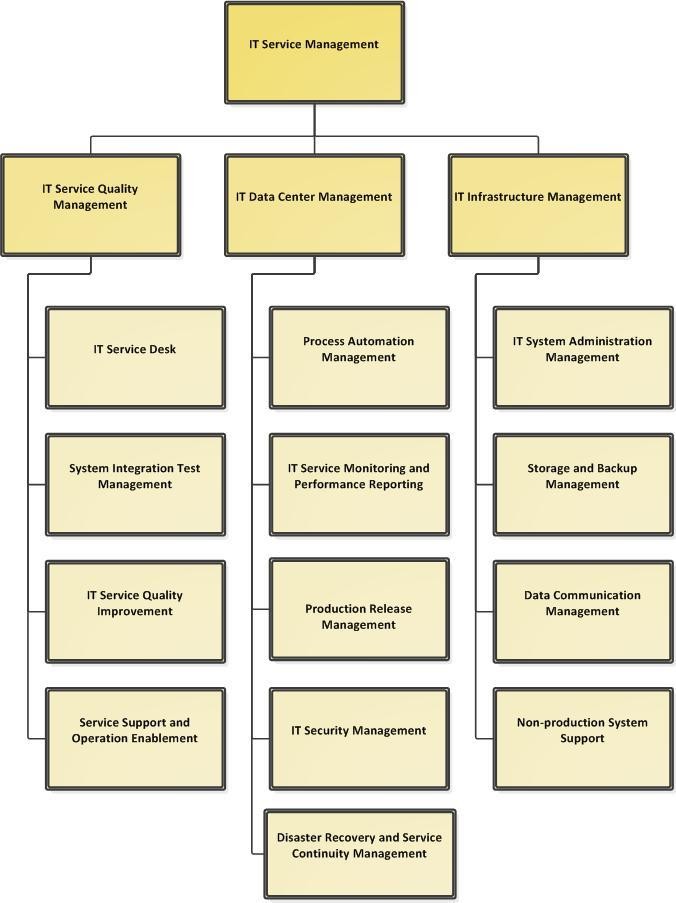
## Gambar 2.2 Struktur Organisasi PT Telkomsel

Annual Report 2015 PT Telkomsel

Susunan direktur Telkomsel adalah sebagai berikut :

1. Presiden Direktur/*President Director* : Ririek Adriansyah
2. Direktur/*Director of Sales* : Mas’ud Khamid
3. Direktur /*Director of Finance* : Heri Supriadi
4. Direktur/*Director of Human Capital Management* : Priyantono Rudito
5. Direktur/*Director of Network* : Sukardi Silalahi
6. Direktur/*Director of Planning & Transformation* :Edward Ying Siew Heng
7. Direktur/*Director of IT* : Ng Soo Kee
8. Direktur/*Director of Marketing* : Alistair Johnston Sedangkan susunan dewan komisaris Telkomsel adalah sebagai berikut :
9. Presiden Komisaris/*President Commissioner* : Alex Janangkih Sinaga
10. Komisaris/*Commissioner* : Heri Sunaryadi
11. Komisaris/*Commissioner* : Diaz Hendropriono
12. Komisaris/*Commissioner* : Mukhlis Moechtar
13. Komisaris/*Commissioner* :Paul Dominic O’Sullivan
14. Komisaris/*Commissioner* : Yuen Kuan Moon

Adapun Strutur organisasi Direktorat TI untuk Departemen IT Service Management PT Telkomsel adalah sebagai berikut:



## Gambar 2.3 Struktur Organisasi Direktorat TI Tahun 2015

## Sejarah Perkembangan Telkomsel

Sejak didirikan pada tahun 1995, Telkomsel terus menerus berevolusi menjadi penyedia jasa telekomunikasi terbaik di Indonesia. Telkomsel secara konsisten melayani negeri, menghadirkan akses telekomunikasi kepada masyarakat Indonesia yang tersebar dari Sabang sampai Merauke.

1. 1995

1. Telkom dan PT Indosat mendirikan Telkomsel
2. Peluncuran kartuHALO pascabayar.

2. 1997

1. Memberikan layanan telekomunikasi *mobile* ke semua provinsi di Indonesia
2. Meluncurkan simPATI Nusantara sebagai layanan prabayar pertama di Asia.

3. 2002

a. Meluncurkan layanan *mobile data* termasuk layanan WAP melalui GPRS *(General Packet Radio Services)*, MMS *(Multimedia Messaging Services)*, dan *Content Services*

4. 2003

a. Pertama di Indonesia memperkenalkan layanan *roaming* internasional prabayar.

5. 2004

1. Meluncurkan Kartu As prabayar.
2. Menerapkan teknologi EDGE, sebagai teknologi *roadmap* berikutnya setelah GPRS.
3. Bergabung dengan *Bridge Alliance*, aliansi regional telekomunikasi selular, untuk memberi manfaat lebih bagi pelanggan.
4. Meluncurkan layanan Nada Sambung Pribadi (NSP). 6. 2005
5. *Call Center* meraih sertifikasi ISO 9001:2000.
6. Meluncurkan layanan *mobile banking* ATM 7. 2006

a. Pertama di Indonesia meluncurkan layanan 3G. 8. 2007

1. Mengenalkan layanan *push email*
2. Pertama di Indonesia meluncurkan layanan Telkomsel Flash HSDPA.
3. Pertama di Indonesia meluncurkan TCASH, layanan uang digital melalui telepon selular.

9. 2008

1. Pertama di Asia menggunakan energi terbarukan untuk BTS.
2. Pertama di dunia menyediakan layanan suara dan data *mobile* di atas kapal PELNI yang memungkinkan pelanggan dapat berkomunikasi di tengah laut.
3. Meluncurkan program Telkomsel Merah Putih dalam rangka memberikan layanan telekomunikasi bagi pulau-pulau, desa-desa terpencil dan daerah perbatasan.

10. 2009

1. Meningkatkan jaringan Telkomsel menjadi HSUPA (*High Speed Uplink Packet Access*), dengan kecepatan akses data mencapai 21 Mbps guna memberikan layanan *mobile broadband* yang lebih baik.
2. Meluncurkan Desa Dering Program untuk menyediakan akses telekomunikasi ke lebih dari 25000 desa

11. 2010

1. Pertama di Indonesia meluncurkan aplikasi *Mobile Newspaper* yang memungkinkan pelanggan membaca berita melalui telepon selular.
2. Pertama di Indonesia melakukan ujicoba teknologi jaringan pita lebar LTE.

12. 2011

1. Pertama di Indonesia mencapai 100 juta pelanggan yang memposisikan Telkomsel sebagai 7 operator telekomunikasi terbesar dari sisi jumlah konsumen
2. Membuka fasilitas riset dan pengembangan untuk teknologi selular pertama di Indonesia
3. Mengenalkan Tap Izy, sistem pembayaran *contactless* pertama di Indonesia yang membuat telepon genggam menjadi dompet elektronik

13. 2012

1. Mengenalkan *seamless mobile wi-fi* pertama di Indonesia untuk transfer otomatis dari jaringan koneksi 2G/3G ke jaringan Wi-Fi untuk akses kecepatan tinggi
2. Mencapai 125 juta pelanggan dan memperkuat posisi Telkomsel sebagai penyedia jasa telekomunikasi terbesar di Indonesia dari sisi konsumen

14. 2013

1. Meluncurkan *Masterpiece*, program yang terdiri dari 268 unit *mobile* GraPARI*,* membentuk 268 *Mobile Broadband Cities*, 68 aplikasi dari pengembang lokal untuk mendukung pengembangan industri digital kreatif
2. Meluncurkan 450 unit *Compact Mobile Base Station* (COMBAT)
3. Percobaan LTE selama APEC 2013 berjalan sukses
4. Mencapai Rp 60 T dalam pendapatan 15. 2014
5. Meluncurkan program *Masterpiece* : CRM, M2M dan *Winning The Youth*
6. Telkomsel dan Telkom mencapai kesepakatan dalam *Conditional Business Transfer Agreement* (CBTA) untuk menggunakan spektrum Flexi
7. Telkomsel menjadi yang pertama meluncurkan layanan 4G LTE di Indonesia
8. Mencapai pertumbuhan *triple double digit* dalam pendapatan, EBITDA dan pendapatan bersih selama 3 tahun berturut-turut

16. 2015

1. Meluncurkan program *Masterpiece* : *Digital World, Great Payment Experience* dan TrueBEx (*True Broadband Experience*)
2. Menyelesaikan perencanaan ulang spektrum di 1800MHz, dan meluncurkan LTE di 14 kota dengan 2.2 juta pengguna LTE
3. Tahun ke-4 mencapai pertumbuhan *triple double digit* dalam pendapatan (+14.8% YoY), EBITDA (+14.4% YoY) dan pendapatan bersih (+15.4% YoY). Pertumbuhan pendapatan YoY tertinggi sejak 2007.
4. Memiliki lebih dari 100 ribu BTS dan 150 juta konsumen
5. Mencapai *Triple 3 Strategy*, dengan nilai perusahaan > Rp 300 Tn, pendapatan kumulatif selama 3 tahun 33%, dan kontribusi bisnis digital 33% dari pendapatan yang dicapai pada Q4-15

## Strategi Bisnis Perusahaan

Telkomsel memiliki visi untuk menjadi penyedia solusi dan layanan *digital lifestyle* kelas dunia dan terpercaya dengan cara menghadirkan jaringan dunia

dengan konektivitas tanpa batas, ragam aplikasi untuk memfasilitasi gaya hidup modern serta rangkaian produk untuk meningkatkan kualitas hidup manusia serta layanan multimedia dengan teknologi terdepan membuka peluang kreatif yang memperkaya sendi-sendi kehidupan bangsa Indonesia. Misi dari Telkomsel adalah menyediakan solusi dan layanan *mobile digital* yang melebihi ekspektasi konsumen, memberikan nilai kepada *stakeholder,* dan mendukung perkembangan ekonomi bangsa.

## Strategi Korporat

Pencapaian dari strategi jangka menengah (*Triple 3*) PT Telkomsel menjaga Telkomsel untuk tetap ada di jalur transformasi ke Telkomsel 2.0, dari Telco ke Digico, dan juga meraih rencana 5 tahun strategi korporat untuk memastikan pertumbuhan berkelanjutan dalam jangka panjang. Telkomsel memiliki 5 strategi pertumbuhan :

* Eksploitasi warisan bisnis
* Memperkuat bisnis utama (data)
* Membangkitkan ekosistem layanan digital
* Penggerak investasi dalam inovasi
* Mengakuisisi secara selektif sebagai bagian dari strategi grup induk

Telkomsel 2.0 merupakan program transformasi Telkomsel yang diluncurkan tahun 2012 untuk memastikan eksekusi kesuksesan pertumbuhan strategi jangka panjang. Telkomsel berkomitmen untuk mentransformasikan bisnis, organisasi, karyawan dan budaya untuk mencapai tujuan strategis perusahaan dan mengantisipasi pergeseran dinamis dalam pasar.

Dari segi transformasi bisnis, Telkomsel fokus pada DNA (*Device-Network- Application*). DNA merupakan transformasi utama dalam bisnis data dan digital, menambah inovasi dan kualitas dalam memberikan layanan pada konsumen. Untuk *device,* Telkomsel menciptakan kemitraan strategis dengan produsen perangkat untuk mendorong penetrasi telepon pintar dan menurunkan biaya *handset* dengan program *bundling* eksklusif. Untuk *network*, Telkomsel meluncurkan LTE di 14 kota dan menyelesaikan penyusunan kembali spektrum di

1800 MHz saat mempertahankan posisi dalam pasokan jaringan dengan hampir 1800 BTS baru pada tahun 2015. Untuk *applications*, Telkomsel melanjutkan kompetisi tahunan Digital Creative Indonesia, The NextDev 2015, pada tema WOW Smart City, dengan tujuan utama untuk menstimulasi inovasi digital di Indonesia.

Dari segi transformasi organisasi, Telkomsel fokus dalam membangun *Digital Ready Organization* melalui 3 program kunci: *Transformation Jumpstart, People’s Digital Built in Capabilities Development,* dan *Living The Telkomsel Way & Nurture Digital Ready Leadership.* Dari segi transformasi karyawan, Telkomsel fokus dalam meningkatkan kapabilitas dalam bisnis digital dan dengan pola pikir digital. Program pengembangan karyawan terdahulu fokus meningkatkan kompetensi teknis dan bisnis untuk bisnis digital, khususnya dalam area *machine to machine*, *mobile finance services*, *big data analytics*, *application programming interface*, *mobile advertising*, dan *digital lifestyle*. Kompetensi utama telah diolah menjadi lebih berorientasi pelanggan, memelihara orang dan fokus eksekusi dengan tujuan memastikan bahwa semua orang di Telkomsel terus menerus berjuang untuk lebih berorientasi pelanggan dan tepat dalam mengeksekusi tujuan dan objektif perusahaan, serta mampu menjadi pemimpin perubahan. Dengan cara ini, Telkomsel memastikan bahwa kualitas bakat tersedia dan siap untuk dikembangkan dalam layanan digital masa depan.

## Telkomsel Masterpiece 2015

Telkomsel memiliki 3 program *Masterpiece* pada tahun 2015, yaitu *MyTelkomsel Digital World, Great Payment Experience*, dan *True Broadband Experience* (TrueBEx).

1. *MyTelkomsel Digital World*

Untuk menguatkan posisi Telkomsel sebagai perusahaan digital terdepan di Indonesia, Telkomsel menginisiasi program *Telkomsel Digital World* (TDW) pada tahun 2015. Program ini merupakan satu dari program *Masterpiece* 2015 yang menyorot perbaikan aplikasi yang sudah ada sebelumnya, yaitu MyTelkomsel dan situs korporat [www.telkomsel.com](http://www.telkomsel.com/). TDW didesain untuk

memfasilitasi promosi Telkomsel agar lebih mudah dipersonalisasi, tepat waktu, dan relevan, serta penawaran *Digital Lifestyle* dan pelayanan pelanggan melalui *mobile* dan *web*.

Melalui aplikasi ini, pelanggan akan mendapatkan informasi tentang pulsa prabayar dan pemakaian pascabayar melalui dasbor pribadi penguna ‘*My Account*’, menghubungi *Customer Care*, isi ulang menggunakan berbagai metode pembayaran termasuk TCASH dan kartu kredit (VISA & *Mastercard*), menukar Telkomsel POIN, membeli konten digital, menikmati penawaran kontekstual dari penjual Telco dan non-Telco dan mencari GraPARI terdekat menggunakan teknologi berbasis lokasi. Melalui aplikasi ini, Telkomsel dapat mengumpulkan data kebiasaan pelanggan dan menggunakannya untuk membuat segmen profil dari pelanggan Telkomsel.

1. *Great Payment Experience*

Untuk program *Great Payment Experience*, Telkomsel telah fokus membuat ekosistem keuangan digital untuk memudahkan transaksi di pasar *online*. Pada tahun 2015, beberapa inovasi dikenalkan untuk mengembangkan layanan pembayaran digital seperti peluncuran TCASH baru dan TCASH *Tap* dengan teknologi NFC. TCASH yang baru memungkinkan konsumen dan kios untuk menggunakan metode pembayaran baru untuk transaksi *mobile* apa saja. TCASH *Tap* sudah diterima di 2500 kios dalam area Jabodetabek dan area lainnya yang masih dalam rencana ekspansi. TCASH *Tap* mengutamakan layanan uang *mobile*, seperti pembayaran cepat menggunakan *tap*, belanja *online*, membeli dan membayar dengan telepon genggam, dan transfer uang. Usaha Telkomsel untuk melanjutkan pengembangan layanan pembayaran digital diharapkan dapat mempercepat program pemasukan keuangan dengan mempromosikan kemitraan dengan organisasi terkait dan badan pemerintahan.

1. *True Broadband Experience* (TrueBEx)

Untuk tetap bertahan di dalam industri IT, sangat penting bagi Telkomsel untuk melanjutkan peningkatan bisnis digital melalui strategi transformasi utama Telkomsel dari DNA (*Device-Network-Application*). Pengalaman konsumen merupakan kunci utama untuk memenangkan bisnis digital dan

pada tahun 2015. Telkomsel fokus pada implementasi *True Broadband Experience* (TrueBEx).

Ada beberapa inisiatif yang diimplementasikan untuk memastikan konsumen dapat menikmati *True Broadband Experience* :

* 1. Peningkatan kualitas jaringan untuk menyediakan cakupan, kapasitas, dan kualitas yang besar melalui optimisasi jaringan, instalasi COMBAT (*Compact Mobile Base Station*), *Customer Experience Test* (CET), DC- HDSPA dan HSPA+ pada 30 kota *broadband* untuk memastikan kecepatan *throughput* mencapai 42Mbps, penambahan BTS dan kapasitas jaringan, pengawasan kinerja dan penanganan keluhan.
  2. Peningkatan *customer interface* untuk meningkatkan pengalaman konsumen melalui info peningkatan harga di telkomsel.com, menu *broadband*, pencegahan kejutan untuk biaya PAYU, kemudahan akses UMB \*363# dengan info harga yang dinamis, klasifikasi layanan yang jelas, dan penawaran produk, serta mempersingkat waktu aktivasi.
  3. Pengalaman akses yang mulus di jaringan 2G-3G-4G dan WiFI, dengan bekerjasama dengan Telkom untuk menyediakan solusi WiFi untuk konsumen yang berada di area jaringan yang padat dengan membebankan penggunaan data ke jaringan WiFi menggunakan Aplikasi Telkomsel WIFI.
  4. Strategi harga yang lebih baik dengan menggunakan basis zona untuk mempertahankan daya saing.
  5. Membuat jaringan 4G yang lebih luas untuk menyediakan LTE yang terbaik dan tercepat di 14 kota di Indonesia dan untuk mempertahankan kecepatan akses sampai 75Mbps pada spektrum 1800 MHz

## Strategi Teknologi Informasi

Teknologi informasi di Telkomsel fokus untuk memungkinkan transformasi dari Telco ke Digico dalam rangka menopang pertumbuhan melalui pendapatan bisnis digital. Fokus tahun 2015 adalah “*WOW! Customer Experience*” dengan tekanan pada kepentingan untuk menyediakan pengalaman yang hebat pada konsumen

saat menggunakan layanan dan produk dari Telkomsel. Oleh karena itu, Telkomsel memulai beberapa inisiatif strategi IT dalam rangka mengenalkan teknologi terbaru dan paling maju, dan juga membangun kompetensi sumber daya internal untuk reliabilitas dan efisiensi yang lebih tinggi.

1. Membuat Perusahaan Digital (Digico)

Telkomsel fokus dalam mengekploitasi sumber daya pendapatan digital, sejalan dengan transformasi ke perusahaan digital. Pada tahun 2015, teknologi informasi memungkinkan platform pembayaran digital, yang membuka kesempatan untuk metode pembayaran baru dengan adanya solusi NFC untuk TCASH Tap. Telkomsel meluncurkan ulang TCASH Tap sebagai usaha untuk meningkatkan pengalaman transaksi digital yang menghasilkan transaksi *mobile* yang lebih mudah dan lebih cepat menggunakan teknologi *Near Field Communication* (NFC).

Teknologi informasi Telkomsel juga mengimplementasikan penagihan terintegrasi dengan OTT seperti Google Play Store dan Microsoft Store untuk memungkinkan pembelian aplikasi secara langsung menggunakan pulsa Telkomsel. Integrasi ini sangat sukses dan telah memberikan kontribusi dalam pencapaian target pendapatan bisnis digital.

1. Membenahi *Customer Experience* melalui data analisis

Dalam rangka untuk memahami konsumen, Telkomsel menyadari kebutuhan kapabilitas data analisis yang kuat yang dapat menyediakan pengetahuan tentang profil, serta pola penggunaan dan konsumsi konsumen. Telkomsel IT telah mengenalkan profil pelanggan multi dimensi yang memungkinkan Telkomsel menyediakan penawaran yang sesuai dengan kebutuhan dari segmen konsumen tertentu. Segmentasi konsumen akan memainkan peran yang sangat penting dalam meningkatkan efektivitas kampanye pemasaran.

Kapabilitas data analisis paling canggih yang dikembangkan oleh Telkomsel IT dapat dijalankan dengan adanya peluncuran platform *big data*, terutama *Operational and Strategic Decision Support System* (OSDSS), ditambah

dengan investasi dalam analisis canggih di atas *Secured Hadoop Cluster*. Solusi *big data* ini telah meningkatkan tanggapan ke kampanye pemasaran secara signifikan.

1. Menuju IT yang dapat diandalkan dan efisien

Sebagai tambahan usaha untuk memungkinkan jalannya bisnis, Telkomsel IT juga berusaha untuk meningkatkan keunggulan teknologi informasi internal dari perspektif teknologi, proses, dan manusia. Pada tahun 2015, Telkomsel fokus dalam memberdayakan sumber daya internalnya untuk memastikan kualitas dari hasil kerja. Telkomsel meninjau kembali dan memperbarui model operasi IT dalam rangka menemukan strategi operasional internal yang ‘sesuai tujuan’. Telkomsel juga terus menerus berusaha untuk meningkatkan ketahanan dari infrastruktur keamanan yang merupakan salah satu perhatian penting.

Direktorat IT mendapatkan sertifikat ISO/IEC 20000-1:2011, sertifikat internasional yang dikenal sebagai standar dari manajemen kualitas layanan IT. Pencapaian ini benar-benar menunjukkan niat untuk terus meningkatkan manajemen layanan IT di Telkomsel. Sebagai tambahan, Direktorat IT terus menerus mendorong daya saing biaya dengan cara memaksimalkan pemanfaatan dari sumber daya internal dalam domain pengisian dan penagihan, serta mengoptimasi infrastruktur IT seperti *server* dan *storage* untuk mengelola penambahan biaya pengeluaran operasi.

## Landasan Teori

Pada bagian ini akan dijelaskan tentang teori-teori yang relevan dengan *Research Question*, yaitu mengenai terminologi insiden dan kegagalan aplikasi, terminologi aplikasi, *software quality*, dan faktor penyebab kegagalan aplikasi.

### *Bugs, faults, errors, failures, incidents*

Perlu dicatat bahwa terminologi mengenai masalah dalam aplikasi tidak sepenuhnya konsisten (Asadollah, et al, 2015). Bourque (2014) menjelaskan bahwa *faults* merupakan sumber dari malfungsi, sedangkan Eldh et al (2007)

mendefinisikan *fault* sebagai sumber statis masalah dalam *source code*. Adapun yang dimaksud dengan *bugs* menurut Asadollah, et al (2015) merupakan malfungsi yang dapat diobservasi dalam program yang sedang dites. Sedangkan *error* merupakan masalah yang terdeteksi selama eksekusi atau saat *run time* yang menyebabkan program berjalan tidak sesuai fungsinya (Eldh, et al, 2007).

Terminologi *Failure* dan *Incident* seringkali digunakan bergantian karena dekatnya pengertian antara *failure* dan *incident*. *Systems and software engineering vocabulary* (2010) mendefinisikan kegagalan (*failure)* akan terjadi ketika sistem atau komponen tidak dapat melakukan fungsinya sesuai dengan kebutuhan, sedangkan insiden (*incident)* merupakan kejadian yang bukan merupakan standar operasi layanan yang akan atau mungkin menyebabkan interupsi, atau pengurangan kualitas layanan. Dari pengertian ini, didapat bahwa terminology insiden lebih condong ke arah pengurangan kualitas layanan, sedangkan kegagalan hanya kegagalan dari komponen itu sendiri. Namun, dalam terminologi ITIL, gangguan yang telah atau belum berdampak pada layanan keduanya tetap disebut sebagai insiden.

Selaras dengan pengertian insiden di atas, Guo dan Wang (2009) mendefinisikan insiden sebagai gangguan layanan yang dapat menimbulkan dampak yang cukup besar untuk kapabilitas bisnis organisasi IT, sehingga membutuhkan implementasi *incident management* dan proses restorasi layanan. Adapun definisi lainnya yang sedikit berbeda dari pendapat Guo dan Wang dari segi dampak insiden, yaitu insiden sebagai gangguan tak terencana pada sebuah layanan, pengurangan kualitas layanan atau kejadian yang belum berdampak pada pelanggan (IEEE Standard - Adoption of ISO/IEC 20000-1:2011, Information Technology - Service Management- Part 1: Service Management System Requirements, 2013).

Dari paparan teori di atas, maka dapat disimpulkan bahwa insiden merupakan kejadian yang menyebabkan pengurangan kualitas layananan yang dapat berdampak pada kapabilitas organisasi, yang mana dalam penelitian ini erat kaitannya dengan pencapaian *service level agreement* perusahaan.

## Aplikasi

Aplikasi atau perangkat lunak merupakan seperangkat program komputer, prosedur, dan dokumentasi data terkait (Systems and software engineering vocabulary, 2010). Terdapat dua jenis utama dari perangkat lunak yaitu sistem perangkat lunak dan aplikasi perangkat lunak (Turban, Rainer, & Potter, 2004). Berbeda dengan aplikasi, *software system* adalah komponen yang saling berinteraksi yang membentuk sebuah perangkat lunak yang merupakan bagian dari sistem komputer (kombinasi *hardware* dan *software*) (Dalal dan Chhillar, 2012). Adapun sejalan dengan pendapat tersebut, menurut Turban, et al (2004) sistem perangkat lunak adalah sekumpulan instruksi yang memiliki fungsi utama sebagai perantara antara perangkat keras komputer dan program aplikasi dimana pengguna yang memiliki pengetahuan tersebut dapat secara langsung melakukan manipulasi.

Dari dua pendapat mengenai perngertian sistem perangkat lunak tersebut, didapat bahwa aplikasi terbentuk dari sistem yang saling berkomunikasi satu sama lain agar dapat digunakan oleh pengguna. Pernyataan ini juga didukung oleh Turban, et al (2004) dimana mereka mendefinisikan aplikasi sebagai sekumpulan instruksi komputer yang menyediakan fungsionalitas yang spesifik ke pengguna. Sedangkan menurut Bryan maizlish dan Robert Handler mendefinisikan aplikasi sebagai sebuah kumpulan kode perangkat lunak yang menyertakan logika dan aturan-aturan bisnis yang mengubah pengguna atau masukan sistem menjadi keluaran data, dengan tujuan untuk mengotomisasi dan mengoptimalkan fungsi bisnis, proses, tugas, dan kegiatan di dalamnya (Maizlish dan Handler, 2005).

Dari berbagai pengertian mengenai aplikasi di atas, maka dapat disimpulkan bahwa aplikasi merupakan seperangkat kode program yang menyediakan fungsionalitas spesifik untuk mendukung proses bisnis tertentu. Oleh karena itu aplikasi yang dimaksudkan dalam penelitian ini adalah aplikasi perangkat lunak, bukan sistem perangkat lunak.

### *Software Quality*

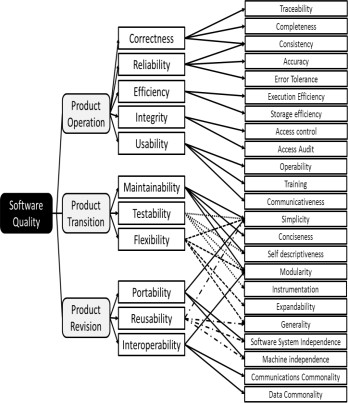
Kualitas dalam lingkup rekayasa perangkat lunak berarti ‘memenuhi persyaratan’ dan ‘sesuai dengan kegunaan’ (Lewis, 2004). Sejalan dengan pengertian tersebut dalam buku *Software Quality Assurance From theory to implementation* oleh Galin (2004) menyatakan bahwa *software quality* adalah : (1) Derajat dimana sebuah sistem, komponen, atau proses memenuhi persyaratan yang telah ditentukan dan (2) Derajat dimana sebuah sistem, komponen, atau proses, memenuhi kebutuhan dan harapan pengguna atau pelanggan.

Olaperi dan Misra (2015) menambahkan bahwa sebuah produk aplikasi dikatakan berkualitas bila produk tersebut sesuai dengan kebutuhan konsumen, sesuai dengan kegunaan produk, dan jumlah *defect* yang ditemukan. Hal ini menunjukkan bahwa aplikasi yang sesuai dengan kebutuhan pengguna adalah aplikasi yang sesuai dengan spesifikasi yang sudah dinyatakan dalam *requirement specification*, dan aplikasi yang sesuai kegunaannya adalah aplikasi yang kegunaannya persis sama seperti keinginan pengguna.

Maka dapat disimpulkan bahwa sebuah aplikasi dikatakan berkualitas bila aplikasi tersebut memenuhi *software requirement*, memenuhi harapan dan kebutuhan pengguna, serta bebas dari *defect*. Luasnya pembahasan mengenai hal-hal yang berkaitan dengan berbagai atribut aplikasi penggunaan, dan pemliharaannya, butuh definisi komprehensif dari *requirement* yang dapat mencakup semua atribut dari aplikasi dan aspek kegunaan aplikasi, termasuk *usability, reusability*, dan *maintainability* yang dapat menjamin kepuasan pengguna yang dapat diklasifikasikan sebagai faktor kualitas (Galin, 2004).

### *McCall’s quality model*

McCall mengklasifikasikan semua *software requirement* ke dalam 11 faktor kualitas dan 23 kriteria *aplikasi* yang dapat dilihat pada Gambar 2.4 . 11 faktor tersebut kemudian dikelompokkan menjadi 3 kategori, terdiri dari *product operation* (*Correctness, Reliability, Efficiency, Integrity*,*Usability*) , *product revision* (*Maintainability*, Flexibility, Testability) , dan *product transition (Portability, Reusability, Interoperability).*

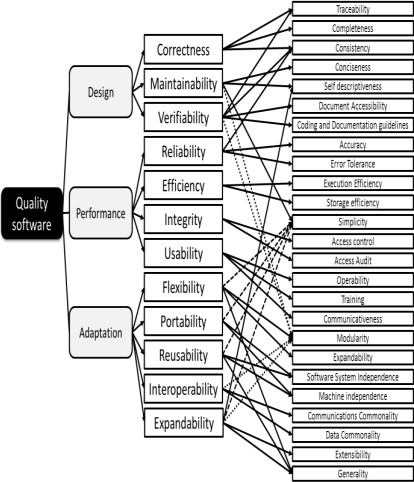


## Gambar 2.4 McCall’s faktor model

Youness et all, 2013

* + - 1. **Evans & Marciniak dan Deutsch & Willis *quality models***

Model Evans dan Marciniak merupakan model alternatif yang muncul setelah model McCall, ia mendefinisikan dua belas faktor yang dikelompokkan ke dalam tiga kategori: *design, performance,* dan *adaptation* (Gambar 2.5)

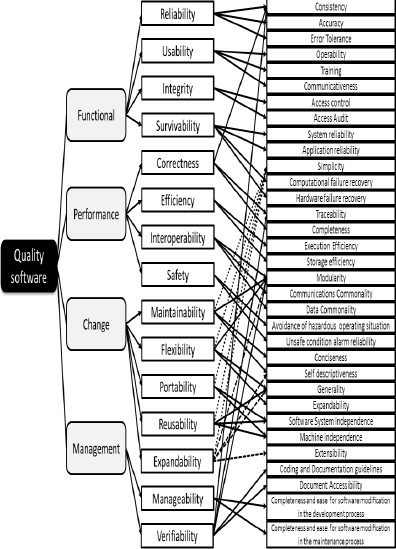


## Gambar 2.5 Evans & Marciniak Model

Youness et all, 2013

Model Deutsch and Willis juga merupakan alternatif model yang muncul setelah McCall. Model ini memiliki 15 faktor yang dikategorikan menjadi 4 kategori, yaitu *operating, performance, change* dan *management.* (Gambar 2.5).

Perbandingan formal terhadap dua model ini menunjukkan bahwa kedua model alternatif ini hanya mengeluarkan satu dari 11 faktor pada model McCall, yaitu *Testability* (Galin, 2004). Selain itu berdasarkan Gambar 2.4 dan Gambar 2.5 terlihat bahwa terdapat 5 faktor tambahan yang disarankan oleh dua alternatif model ini, yaitu *verifiability* (kedua model), *expandability* (kedua model), *safety* (Deutsch and Willis), *manageability* (Deutsch and Willis), dan *survivability* (Deutsch and Willis).

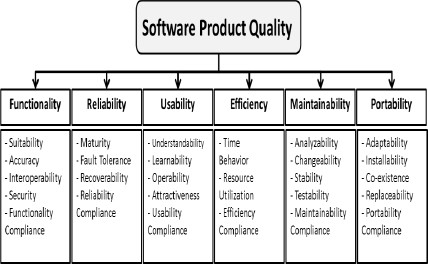


## Gambar 2.6 Deutsch & Willis *quality* model

Youness, 2013

## ISO 9126’s Quality Model (1991)

ISO 9126 *quality model* diajukan sebagai standar internasional untuk *software quality measurement*. Model ini diturunkan dari model McCall (1977) dan memiliki 2 bagian utama yang terdiri dari *Internal and External Quality Attributes* dan *Quality in Use Attributes*. Internal atribut merujuk pada properti sistem yang dapat dievaluasi tanpa mengeksekusinya, sedangkan eksternal atribut merujuk pada properti sistem yang dapat dievaluasi saat proses eksekusi. *Quality* in use atribut merujuk pada properti sistem yang dialami oleh pengguna saat sistem dalam kondisi beroperasi dan saat *maintenance*. Karakteristik dari model ini (Internal dan External) terdiri dari *Efficiency, Functionality, Maintainability, Portability, Reliability dan Usability* (Gambar 2.7).

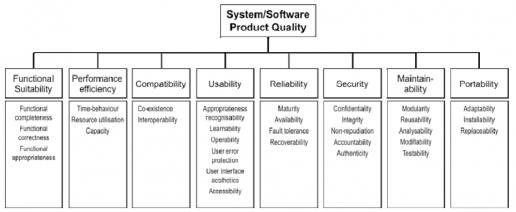


**Gambar 2.7 ISO 9126 *quality* model**

Youness, 2013

## 2.2.3.4 ISO 25000 (SQuaRE) (2011)

ISO 25000 atau *Systems and aplikasi Quality Requirements and Evaluation* (SQuaRE) merupakan standar internasional yang diturunkan dari ISO/IEC 9126:1991. Model ini mendefinisikan : (1) *Quality in use model* dan (2) *Product Quality Model. Product Quality Model* terdiri dari 8 kriteria, yang kemudian terbagi menjadi 30 sub-kriteria yang dapat dilihat pada Gambar 2.8:

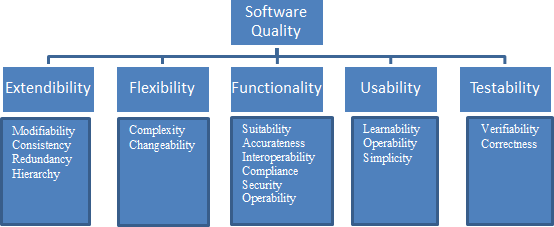


## Gambar 2.8 ISO 25000 (SQuaRE) model

Youness, 2013

**2.2.3.5 Hamid & Hasan *quality model* (2011)**

Model ini merepresentasikan kualitas aplikasi khusus untuk industri telekomunikasi dengan mempertimbangkan tiga sudut pandang, yaitu sudut pandang manajer untuk merepresentasikan dimensi ekonomi, kemudian sudut pandang pengguna untuk merepresentasikan dimensi social, dan sudut pandang pengembang untuk merepresentasikan dimensi teknis (Hamid dan Hasan, 2011). Hamid dan Hasan (2011) mengklasifikasikan *software quality factor* menjadi 5 kategori, yaitu *Extendibility, Flexibility, Functionality, Usability,* dan *Testability.*



## Gambar 2.9 Hamid & Hasan quality model for Telecommunication Industri

Hasan & Hamid, 2011 – telah diolah kembali

Dari Gambar 2.9 tersebut, maka terdapat 5 karakteristik aplikasi yang menjadi faktor kualitas aplikasi, dan 14 sub karakteristik.

1. ***Extendibility*** merupakan usaha yang dibutuhkan untuk mengembangkan kapabilitas dan performance dengan mengembangkan fungsi saat ini atau dengan menambahkan fungsionalitas baru. Subkarakteristik dari *extendibility* adalah *modifiability, consistency, redundancy*, dan *hierarchy.*
   1. *Modifiability* untuk membuat dasar atau perubahan fundamental yang seringnya untuk memberikan orientasi baru
   2. *Consistency* berarti sistem memiliki notasi, symbol, dan terminologi yang seragam
   3. *Redundancy* adalah duplikasi dari komponen kritikal sistem dengan maksud meningkatkan reliabilitas sistem
   4. *Hierarchy* adalah pengaturan item dimana item akan direpresentasikan di atas, di bawah, atau pada level yang sama dengan item lainnya
2. ***Flexibility*** merupakan kemudahan dalam mengubah tujuan, fungsi, atau data aplikasi untuk memenuhi kebutuhan perubahan. Subkarakteristik dari *flexibility* adalah *Complexity,* dan *Changeability.*
   1. *Complexity* fokus pada mengklasifikasikan masalah komputasional berdasarkan kesulitan turunan
   2. *Changeability* merupakan seberapa mudah aplikasi dapat menjalani berbagai perubahan
3. ***Functionality*** berarti kapabilitas aplikasi untuk menyediakan fungsi yang sesuai dengan kebutuhan yang tersurat maupun tersirat ketika aplikasi berada pada kondisi tertentu. *Functionality* memiliki 6 subkarakteristik, yaitu *suitability, accurateness, interoperability, compliance, security*, dan *operability.*
   1. *Suitability* merupakan atribut aplikasi yang menanggung kehadiran dan ketepatan satu set fungsi untuk tugas tertentu.
   2. *Accurateness* merupakan atribut aplikasi yang menanggung kemampuan aplikasi untuk berinteraksi dengan sistem tertentu
   3. *Interoperability* merupakan usaha relatif yang dibutuhkan untuk memasangkan aplikasi pada suatu platform ke platform lainnya
   4. *Compliance* untuk membuat aplikasi mengikuti standar dari aplikasi terkait atau ketentuan atau regulasi hokum dan keputusan serupa
   5. *Security* merupakan kemampuan aplikasi untuk mencegah akses pihak yang tidak sah
   6. *Operability* adalah kemudahan operasi program
4. ***Usability*** merupakan kemudahan belajar dan operasi aplikasi. *Usability*

memiliki 3 subkarakteristik, yaitu *learnability, operability,* dan *simplicity.*

* 1. *Learnability* merupakan usaha pengguna untuk mengenali konsep logika dan penerapannya
  2. *Operability* merupakan usaha pengguna untuk operasi dan kontrol operasi
  3. *Simplicity* merupakan derajat sebuah program dapat dimengerti tanpa kesulitan

1. ***Testability*** berarti kemudahan dalam menguji program untuk memastikan bahwa aplikasi melakukan fungsi spesifiknya. Subkarakteristik dari *testability* adalah *veriviability* dan *correctness*
   1. *Verifiability* kemudahan untuk memastikan fitur aplikasi dan kinerja aplikasi berdasarkan tujuan yang sudah dinyatakan
   2. *Correctness* merupakan tingkat sebuah program memenuhi spesifikasi dan tujuan klien

## Perumusan Kriteria Kegagalan Aplikasi

Berikut merupakan perbandingan antar *quality model* dari 6 *quality model* yang telah dijelaskan di atas :

**Tabel 2.1 Tabel Perbandingan Antar *Quality Model***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***Software*** | **McCall** | **Evans &** | **Deutsch** | **ISO** | **SQuaRE** | **Hamid** |
| ***Quality*** | **(1977)** | **Marciniak** | **&** | **9126** | **(2011)** | **&** |
| ***Criteria*** |  | **(1988)** | **Willis** | **(1991)** |  | **Hasan** |
|  |  |  | **(1988)** |  |  | **(2011)** |
| *Maintainability* | X | X | X | X | X |  |
| *Flexibility* | X | X | X |  |  | X |
| *Testability* | X |  |  |  |  | X |
| *Correctness* | X | X | X |  |  |  |
| *Efficiency* | X | X | X | X | X |  |
| *Reliablity* | X | X | X | X | X |  |
| *Integrity* | X | X | X |  |  |  |
| *Usability* | X | X | X | X | X | X |
| *Portability* | X | X | X | X | X |  |
| *Reusability* | X | X | X |  |  |  |
| *Interoperability* | X | X | X |  |  |  |
| *Functionality* |  |  |  | X | X | X |
| *Performance* |  |  |  |  |  |  |
| *Verifiability* |  | X | X |  |  |  |
| *Expandability* |  | X | X |  |  | X |
| *Survivability* |  |  | X |  |  |  |
| *Safety* |  |  | X |  |  |  |
| *Manageability* |  |  | X |  |  |  |
| *Security* |  |  |  |  | X |  |
| *Compatibility* |  |  |  |  | X |  |
|  |  |  |  |  |  |  |

Dari hasil pemetaan pada Tabel 2.1 di atas, kriteria yang akan digunakan pada penelitian ini adalah 5 kriteria dari model Hamid dan Hasan (2011) dimana kriteria *software quality* yang dihasilkan adalah khusus untuk aplikasi yang digunakan pada industri telekomunikasi, hal ini sesuai dengan penelitian yang dilakukan oleh penulis dimana studi kasusnya adalah perusahaan telekomunikasi. 5 kriteria yang dihasilkan oleh Hamid dan Hasan (2011) adalah *Extendibility, Flexibility, Functionality, Usability,* dan *Testability.*

Menurut Hamid dan Hasan (2011), *Extendability* dan subkarakteristiknya merupakan faktor yang penting dan sesuai dengan industri telekomunikasi karena cepatnya pertumbuhan industri telekomunikasi yang mengharuskan sistem dapat dikembangkan untuk memenuhi kebutuhan di masa depan. *Flexiblity* merupakan faktor yang penting karena merupakan sebuah keharusan bagi aplikasi pada industri telekomunikasi memiliki sifat fleksibel untuk perubahan untuk memenuhi peningkatan masa depan. *Functionality* juga merupakan faktor yang penting karena semua sistem dan aplikasi dalam industri telekomunikasi harus berfungsi sesuai dengan kebutuhannya. *Usability* sangat penting karena pengguna aplikasi di industri telekomunikasi membutuhkan sistem ergonomis yang mudah digunakan. *Testability* menjadi kriteria yang penting karena pada industri ini pengujian aplikasi untuk memastikan aplikasi sesuai dengan spesifikasi aplikasi adalah proses yang sangat penting.

## Faktor-Faktor Penyebab Kegagalan Aplikasi

Setiap organisasi memulai proyek dengan maksud untuk menyelesaikannya dengan sukses agar dapat bekerja sesuai dengan spesifikasi fungsi yang sudah didefinisikan oleh klien atau sesuai kebutuhan bisnis, namun banyak terdapat penyebab tidak tercapainya tujuan organisasi tersebut karena adanya beberapa kesalahan yang kemudian menyebabkan insiden (Dalal dan Chhillar, 2012). Menurut (Saaralainen dan Jantti, 2015), terdapat tiga jalur kesalahan yang dapat menyebabkan insiden, yaitu :

1. Proses definisi atau instruksi yang tidak memadai, yang menyebabkan *error*. Akar penyebab utamanya adalah kurangnya persyaratan untuk menguji konfigurasi baru setelah perubahan
2. Operasi pengujian tidak dilakukan sesuai rencana. Hal ini dapat terjadi karena hilangnya *requirement* dalam proses definisi.
3. Instruksi pengoperasian yang tidak diikuti karena kurangnya pengawasan

Dari hasil penelitian tersebut terlihat bahwa penelitian yang dilakukan Saaralainen dan Jantti (2015) hanya berfokus pada faktor kesalahan manusia. Padahal kenyataannya insiden dapat terjadi oleh faktor teknis, lingkungan, dan faktor lainnya, seperti temuan yang dikemukakan oleh (Pertet dan Narasimhan, 2005) bahwa :

1. Sebanyak 80% insiden terjadi karena *software error* dan *human error*
2. Insiden terjadi kebanyakan saat perawatan rutin, *software upgrade*, dan sistem integrasi. Namun tidak jelas apakah insiden ini terjadi karena kompleksitas sistem, pengujian tidak memadai, atau pemahaman yang kurang tentang dependensi sistem.
3. Penyebab penting lainnya dari insiden perangkat lunak adalah *system overload, resource exhaustion*, dan kompleksnya *fault recovery*
4. Pemadaman aplikasi selama beberapa menit, atau beberapa minggu
5. *Downtime* berkepanjangan sering terjadi ketika beberapa komponen terkena

*error*

1. *Backup* menjadi lebih lama dari yang direncanakan

Dari temuan tersebut, Pertet dan Narasimhan (2005) menyimpulkan bahwa terdapat 4 penyebab umum insiden, yaitu *software failure, human/operator error, hardware and environmental failure*, dan *security violations* serta insiden kebanyakan terjadi saat *change management*. Ancaman internal dikenal sebagai masalah keamanan dari tahun 1980an (Chinchani et al., 2005), oleh karena itu pada penelitian yang dilakukan oleh Munshi, Dell, dan Armstrong (2012), hal ini diteliti secara mendalam karena masalah keamanan insiden internal dapat menyebabkan lebih dari sekadar kehilangan secara finansial, namun juga kehilangan konsumen, serta rusaknya reputasi perusahaan. Ancaman dari orang

dalam merupakan isu utama yang dihadapi oleh perusahaan dari berbagai sektor. Sedangkan menurut (Dalal dan Chhillar,2012) menyimpulkan penyebab umum dari kegagalan aplikasi dari berbagai *case studies* bahwa terdapat 6 penyebab umum, antara lain :

1. Kurang jelasnya definisi tujuan dan spesifikasi aplikasi
2. Manajemen yang buruk, dan komunikasi yang buruk antara konsumen, desainer, dan *programmer*
3. Langkah yang salah dalam mencatat insiden (*improper fault assignment*)
4. Adanya tekanan institusional (*low budget, fast delivery*)
5. Penggunaan teknologi baru dengan ketidakpastian reliabilitas dan masalah
6. Penolakan pengakuan bahwa proyek dalam masalah

## Perumusan Faktor-Faktor Penyebab Kegagalan Aplikasi

Berikut merupakan perbandingan antara faktor-faktor penyebab kegagalan pada aplikasi berdasarkan penelitian yang dilakukan oleh Saaralainen dan Jantti (2015), Dalal dan Chhillar (2012), Pertet dan Narasimhan (2005), dan Munshi, Dell, dan Armstrong (2012):

## Tabel 2.2 Pemetaan Faktor-Faktor Kegagalan Aplikasi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Faktor Kegagalan Aplikasi** | **(Pertet dan Narasimhan, 2005)** | **(Dalal dan Chhillar, 2012)** | **(Munshi, Dell, dan Armstrong, 2012),** | **(Saaralainen dan Jantti, 2015),** |
| *Software Error* | X | X |  |  |
| *Human/Operator*  *Error* | X | X |  | X |
| *Hardware and Environtmental*  *Error* | X | X |  |  |
| *Security Violations* | X |  | X |  |
| *Fault Recovery*  *Error* | X (tersirat) | X |  |  |
| *Change*  *Management Issue* |  |  |  | X |
| *Communication*  *Management Issue* |  | X |  |  |
| *Institutional*  *pressure* |  | X |  |  |

Dari Tabel 2.2 di atas, faktor-faktor yang akan digunakan dalam penelitian ini adalah faktor yang terdapat pada empat penelitian tersebut. Maka faktor-faktor yang digunakan sebagai faktor-faktor penyebab kegagalan aplikasi, yaitu *software error, human/operator error, hardware and environtmental error, security violations, fault recovery error, communication management issue*, *institusional pressure* dan *change management issue.* Berikut merupakan penjelasan dari masing-masing kategori faktor dan faktornya.

1. ***Software Error*** biasa terjadi saat proses *maintenance, upgrade*, dan integrasi. Penyebab signifikan dari *software error* adalah *system overload, computational /logic error, resource exhaustion, failed software upgrade*, dan *fault recovery routine* (Pertet dan Narasimhan, 2005).
2. ***Human/Operator Error*** diklasifikasikan menjadi 3 kategori *configuration error, procedural error, dan miscellaneous accidents* (Pertet & Narasimhan, 2005). 3 jalur yang menyebabkan human error adalah *organizational process, inadequate supervision*, dan *planned inappropriate operations* (Saaralainen dan Jantti, 2015)
3. ***Hardware and Environmental Error*** dapat terjadi karena beberapa alasan, seperti penggunaan dan pelepasan bagian mekanikal, kesalahan desain, dll. *Hardware failure* dapat terjadi juga karena penggunaan teknologi baru yang tidak diketahui reliabilitasnya (Dalal dan Chhillar, 2012). *Hardware failure* terdiri dari *device driver failure, I/O error, memory parity error*, dan *network hardware failure*. Sedangkan *environmental failure* dapat terjadi karena *power outages, overheating, high humidity,* dan *natural disaster* (Pertet dan Narasimhan, 2005)
4. ***Security Violations*** dapat terjadi karena beberapa kejadian seperti *Password disclosures, Denial of Service (DOS) attacks, Worms and viruses, Browser vulnerabilities, Authentication failures*, dan *Theft* (Pertet dan Narasimhan, 2005)
5. ***Fault Recovery Error*** dapat terjadi karena langkah yang salah dalam mencatat insiden (*improper fault assignment*) (Dalal dan Chhillar, 2012) dan kompleksnya sistem *fault recovery* (Pertet dan Narasimhan, 2005)

### *Communication Management Issue*

1. ***Institusional Pressure*** dapat terjadi karena tawaran yang sangat rendah, permintaan anggaran yang tidak realistis (sangat rendah)*,* dan meremehkan kebutuhan waktu
2. ***Change Management Issue*** dapat terjadi pada saat perencanaan perubahan dan pengujian perubahan, dimana definisi proses atau instruksi tidak memadai. Penyebab utama dari kegagalan ini adalah kurangnya requirement untuk menguji konfigurasi baru setelah perubahan (Saaralainen dan Jantti, 2015)

## Metode Pemeringkatan Faktor-Faktor

Menurut Zhong et all (2008) kualitas merupakan nyawa dari sebuah produk dan keputusan adalah manajemen utama. Saat ini, banyak perusahaan bergegas ke pasar tertentu untuk mendapatkan keuntungan bisnis. Namun, hal ini membawa masalah dalam *quality control* dan pengambilan keputusan dan tidak ada dasar dari kualitas itu sendiri.

Melihat hal tersebut, maka pada bagian ini akan dijelaskan mengenai metode- metode yang umumnya digunakan untuk melakukan pengambilan keputusan terkait pemeringkatan faktor-faktor, yang kemudian akan dibandingkan untuk melihat kesesuaian dengan penelitian yang saat ini dilakukan.

### *Multi Criteria Decision Making Method*

Menurut Velasquez dan Hester (2013) metode *Multi Criteria Decision Making* (MCDM) telah berevolusi untuk mengakomodasi berbagai jenis penerapan. Jumlah penggunaan MCDM sangat luar biasa besar dalam beberapa dekade terakhir, perannya pada area penerapan yang berbeda meningkat secara signifikan, seiring dengan pengembangan metode baru dan peningkatan dari metode lama. Berikut merupakan 3 metode yang umumnya digunakan dalam pengambilan keputusan:

### SAW (*Simple Additive Weighting)*

SAW merupakan fungsi nilai yang dibentuk berdasarkan penambahan skor sederhana yang merepresentasikan pencapaian tujuan di bawah setiap kriteria, yang dikalikan dengan bobot tertentu (Qin et all., 2008). Metode ini memiliki kemampuan untuk mengimbangi antar kriteria, intuitif ke pembuat keputusan, perhitungan yang sederhana, dan dapat dilakukan tanpa bantuan program komputer yang kompleks (Velasquez dan Hester, 2013).

Menurut Podvezko (2011) metode ini memiliki beberapa kekurangan, antara lain

(1) semua nilai-nilai kriteria Ri (i=1,m) harus maksimal. Meminimalkan kriteria harus ditransformasikan ke maksimal dengan formula sebelum digunakan untuk analisis (2) semua nilai kriteria Ri (i=1,m) harus positif. Hasil evaluasi tergantung pada tipe transformasi ke nilai positif. (3) Perkiraan yang dihasilkan oleh SAW tidak selalu mencerminkan keadaan yang sebenarnya. Hasil yang didapat mungkin tidak logis, dengan nilai dari kriteria tertentu berbeda jauh dengan kriteria lain.

SAW sering diaplikasikan pada *water management, business, and financial management*. Metode ini sangat mudah digunakan dan pengguna mengaplikasikannya pada ruang lingkup yang terbatas (Velasquez dan Hester, 2013).

### AHP *(Analytic Hierarchy Process)*

Metode ini dikembangkan oleh Dr. Thomas Saaty untuk menyelesaikan permasalahan baik teknikal maupun manajerial yang membutuhkan pengambilan keputusan (Pinto, 2010). AHP melibatkan perilaku dan proses pemikiran manusia dalam pengambilan keputusannya dengan cara menyederhanakan permasalahan tersebut dan mengubah situasi yang tidak terstruktur menjadi bagian-bagian yang lebih kecil serta tersusun dalam urutan tertentu, memberikan nilai subjektif terhadap variabel, dan kemudian menyatukan penilaian tersebut untuk menentukan variabel mana yang memiliki prioritas tertinggi (Saaty T. L., 2012). AHP digunakan untuk perbandingan, pembobotan, dan pemeringkatan. (Velasquez dan Hester, 2013)

Menurut Velasquez dan Hester (2013) salah satu manfaat dari AHP adalah kemudahan dalam penggunaanya, penggunaan matriks perbandingan dapat membuat pengambil keputusan memberikan bobot pada koefisien dan membandingkan alternatif dengan mudah. Selain itu, AHP memiliki skalabilitas yang tinggi, dapat dengan mudah menyesuaikan ukurannya untuk mengakomodasi pengambilan keputusan (tidak data *intensive*). Salah satu kritik untuk AHP adalah rentan terhadap permeringkatan *reversal* karena sifat dari perbandingan dalam pemeringkatan, penambahan alternatif dapat membuat pemeringkatan menjadi terbalik. AHP sering digunakan pada masalah performansi, *resource management, corporate policy and strategy, public policy, political strategy*, dan perencanaan.

### TOPSIS *(Technique for Order Preference by Similarity to the Ideal* Solution)

TOPSIS adalah pendekatan untuk mengidentifikasi alternatif yang mendekati solusi ideal dan terjauh dari solusi negatif dalam *multi-dimensional computing space* (Qin et al., 2008). Metode ini memiliki banyak manfaat, antara lain mudah digunakan dan diprogram, serta jumlah langkah tetap sama terlepas dari jumlah atribut (Ic, 2012).

Menurut (Velasquez dan Hester, 2013) kerugian dai TOPSIS adalah penggunaan *Euclidean Distance* yang tidak mempertimbangkan korelasi antar atribut. Sulit untuk memberi bobot pada atribut dan menjaga konsistensi dari keputusan, khususnya dengan tambahan atribut. Metode ini banyak digunakan dalam *supply chain management* dan logistik, desain, teknik dan sistem manufaktur, bisnis dan *marketing management, environmental management, human resource management*, dan *water resource management*.

## Perumusan Metode Pemeringkatan Faktor-Faktor Penyebab Kegagalan Aplikasi

Berikut merupakan perbandingan antar 3 metode yang telah diuraikan sebelumnya, yaitu SAW, AHP, dan TOPSIS:

**Tabel 2.3 Tabel Perbandingan Metode *Multi Criteria Decision Making***

|  |  |  |  |
| --- | --- | --- | --- |
|  | **SAW** | **AHP** | **TOPSIS** |
| Konsistensi | Tidak | Ya | Tidak |
| Proses Utama | *Weighted average*  *principle* | *Hieararchy*  *principle* | *Distance principle* |
| Struktur Masalah | Banyak kriteria  dan alternatif | Sedikit kriteria  dan alternative | Banyak kriteria dan  alternative |
| Konsep | *Scoring Model* | *Scoring Model* | *Compromising*  *Model* |
| Hasil akhir | *Global, net ordering* | *Global, net ordering* | *Global, net ordering* |
| Manfaat | Kemampuan untuk mengimbangi antar kriteria ; intuitif untuk pembuat keputusan; perhitungan sederhana | Mudah digunakan; *scalable*; struktur hirarki dapat  sesuai dengan berbagai ukuran masalah; tidak data intensif | Proses sederhana; mudah digunakan dan deprogram; jumlah langkah tidak berubah tanpa memperhatikan jumlah atribut |
| Kekurangan | Hasil estimasi tidak selalu mencerminkan keadaan yang sebenarnya; hasil mungkin tidak logis | Masalah ketergantungan antara kriteria dan alternatif dapat menyebabkan inkonsistensi antara penilaian dan kriteria perangkingan; *rank reversal* | Penggunaan *Euclidean Distance* tidak mempertimbangkan korelasi antar atribut; susah untuk memberikan pembobotan dan menjaga konsistensi  keputusan |
| Area | *Water management, business, and financial management.* | *Performance-type problems, resource management, corporate policy and*  *strategy, public policy, political strategy,*  *and planning.* | *Supply chain management and logistics, engineering, manufacturing systems, business and marketing, environmental, human resources, and water*  *resources management* |

Velasquez dan Hester (2013) , Thor, Ding, dan Kamaruddin (2013) – telah diolah kembali

Dari Tabel 2.3 di atas terlihat bahwa metode yang sesuai dengan penelitian saat ini adalah AHP, karena penelitian ini menggunakan jumlah kriteria dan alternatif yang sedikit dimana metode SAW dan TOPSIS mensyaratkan kriteria dan alternatif yang banyak. Selain itu, hanya metode AHP yang dapat menjaga konsistensi keputusan.

### AHP *(Analytic Hierarchical Process)*

Metode ini dikembangkan oleh Dr. Thomas Saaty untuk menyelesaikan permasalahan baik teknikal maupun manajerial yang membutuhkan pengambilan keputusan (Pinto, 2010). AHP melibatkan perilaku dan proses pemikiran manusia dalam pengambilan keputusannya dengan cara menyederhanakan permasalahan tersebut dan mengubah situasi yang tidak terstruktur menjadi bagian-bagian yang lebih kecil serta tersusun dalam urutan tertentu, memberikan nilai subjektif terhadap variabel, dan kemudian menyatukan penilaian tersebut untuk menentukan variabel mana yang memiliki prioritas tertinggi (Saaty, 2012)

Berdasarkan pengertian dan karakteristik pengambilan keputusan AHP tersebut, maka AHP merupakan metode yang tepat untuk menyelesaikan permasalahan pada penelitian ini. Pada penerapan metode AHP yang diutamakan adalah kualitas data dari responden, dan tidak tergantung pada kuantitasnya (Saaty, 1993). Oleh karena itu pengambilan keputusan dalam metode ini membutuhkan pendapat para pakar. Pakar merupakan orang-orang yang berkompeten dalam pengambilan kebijakan atau benar-benar mengetahui informasi yang dibutuhkan.

## Tahapan dalam AHP

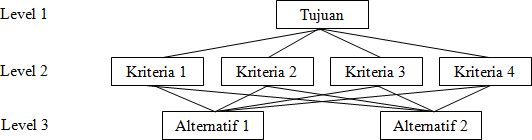
Berikut merupakan langkah-langkah AHP dalam melakukan proses pengambilan keputusan (Saaty, 2012):

1. Membangun struktur hierarki

Tahapan ini merupakan tawahapan awal dari AHP, yaitu memecah realita yang ada menjadi beberapa bagian, kemudian bagian tersebut dipecah lagi menjadi kelompok yang lebih kecil lagi. Berikut merupakan tahapan untuk merancang struktur hierarki AHP :

* 1. Menentukan tujuan utama
  2. Menentukan sub tujuan dari tujuan utama
  3. Menentukan kriteria yang harus dipenuhi dari sub tujuan tersebut
  4. Menentukan sub kriteria dari setiap kriteria yang ada
  5. Menentukan aktor yang terlibat
  6. Menentukan tujuan dari aktor tersebut
  7. Menentukan kebijakan dari aktor tersebut
  8. Menentukan hasil keluaran (alternatif)

Berikut merupakan contoh hierarki AHP sederhana :



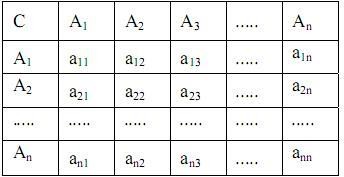
## Gambar 2.10 Struktur Hierarki AHP

Dari Gambar 2.10 tersebut, Level 1 merupakan tingkat paling atas dari struktur AHP dimana Level 1 tersebut merupakan tujuan yang ingin dicapai, sedangkan pada Level 2 merupakan kriteria, dan Level 3 adalah alternatif. Alternatif-alternatif tersebut akan berkontribusi terhadap tujuan melalui dampaknya terhadap kriteria.

1. Menentukan prioritas

Langkah pertama dalam menentukan prioritas adalah menyusun perbandingan berpasangan (*pairwise comparison).* Perbandingan tersebut akan ditransformasikan dalam bentuk matriks n x n untuk melakukan analisis numerik. Matriks perbandingan berpasangan ini berfungsi untuk mengetahui kontribusi dan pengaruh setiap elemen yang relevan atas setiap kriteria yang mempengaruhi elemen di atasnya.

## Tabel 2.4 Matriks Perbandingan Berpasangan



Dari Tabel 2.4 di atas, dapat diartikan bahwa nilai a11 adalah nilai perbandingan elemen A1 terhadap A1. Perbandingan dapat berupa seberapa kuat satu elemen terhadap elemen lain, pengaruh, atau keuntungan dari variabel yang dipertimbangkan. Saat melakukan perbandingan antara satu elemen dengan elemen yang sama missal A1 dan A1 maka diberikan nilai 1, yang berarti kolom diagonal pada matriks tersebut semuanya akan bernilai 1, sedangkan untuk perbandingan elemen yang berbeda, nilai yang digunakan adalah dari 1 sampai 9 yang akan dijelaskan pada Tabel 2.5 di bawah ini.

## Tabel 2.5 Tabel Skala Matriks Perbandingan Berpasangan

|  |  |  |
| --- | --- | --- |
| **Nilai** | **Definisi** | **Keterangan** |
| 1 | Sama penting | Dua aktivitas memiliki kontribusi yang sama terhadap tujuan tertentu |
| 3 | Sedikit lebih penting | Satu aktivitas sedikit lebih penting dari aktivitas lainnya |
| 5 | Jelas lebih penting | Satu ativitas jelas lebih penting dari aktivtas lainnya |
| 7 | Sangat jelas penting | Satu aktivitas mendominasi atau sangat penting terhadap aktivitas lainnya |
| 9 | Mutlak lebih penting | Dapat dibuktikan satu aktivitas memiliki tingkat kepentingan tertinggi terhadap aktivitas lain |
| 2,4,6,8 | Apabila ragu- ragu antara 2 nilai yang  berdekatan | Digunakan jika penilaian tidak dapat dilakukan dengan menggunakan pernyataan di atas |

Saaty, 2012 – telah diolah kembali

Karena sifat AHP bahwa penilaian dapat menyimpang dari konsistensi logis, maka bila sebuah elemen A adalah 7 kali lebih penting dari elemen D, B adalah 5 kali lebih penting dari D, dan C adalah 3 kali lebih penting dari B maka tidak akan dengan mudah menentukan bahwa secara numerik C adalah 15/7 kali lebih penting dari A.

1. Melakukan perhitungan matriks

Perhitungan matriks dilakukan untuk mendapatkan nilai dari setiap kriteria dan elemen. Perhitungan ini berdasarkan prinsip eigenvector. Untuk mendapatkan nilai eigen dapat dilakukan tahapan sebagai berikut (Marimin, 2004) :

* 1. Ubah matriks dengan nilai dalam bilangan decimal
  2. Kuadratkan matriks tersebut
  3. Jumlahkan nilai dari setiap baris matriks
  4. Normalisasikan matriks tersebut
  5. Lanjutkan iterasi berikutnya
  6. Iterasi berhenti jika perbedaan jumlah nilai dari dua perhitungan berturut- turut tidak lebih besar dari suatu nilai batas tertentu.

1. Menghitung nilai rasio konsistensi (CR)

Pengukuran konsistensi ini diperlukan untuk mengetahui konsistensi jawaban yang berpengaruh terhadap kebenaran dari hasil yang didapat. Nilai rasio CR harus sama atau lebih kecil dari 10% (Saaty, 2012). Jika lebih dari 10% maka terdapat kesalahan pada penilaian tersebut dan perlu diperbaiki. Adapun rumus untuk uji konsistensi adalah sebagai berikut :

**CI (*Consistency Index*) :**

CI = 𝛾 max − 𝑛

𝑛−1

Keterangan :

..................................................................................................(1)

CI : Indeks Konsistensi

𝛾𝑚𝑎𝑥 : eigen value maksimum

n : jumlah elemen yang diperbandingkan

dimana

𝛾 max = ∑ 𝑉𝐵 (2)

𝑛

**VB (Nilai Eigen)** = 𝑉𝐴 (3)

𝑉𝑃

**VA (Vektor Antara) =** aij x VP (4)

## VP (Vektor Prioritas) = 𝑽𝑬

∑ 𝒏 ∏𝒏

𝒂𝒊𝒋

...............................................................(5)

√ 𝒊=𝟏

**VE (Vektor Eigen) =** 𝒏√∏𝒏 𝑎𝑖𝑗 (6)

Untuk mengetahui apakah CI dengan besaran cukup baik atau tidak, maka perlu diketahui rasio konsistensinya (CR) yaitu :

𝒊=𝟏

**CR (*Consistency Ratio*) :**

CR = 𝐶𝐼 (7)

𝑅𝐼

RI merupakan indeks acak (*Random index*) yang dikeluarkan oleh Oak Ridge Laboratory. Berikut merupakan tabel indeks acak:

## Tabel 2.6 Indeks Acak (Random Index)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **N** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| RI | 0,00 | 0,00 | 0,58 | 0,90 | 1,12 | 1,24 | 1,32 | 1,41 |
| **N** | **9** | **10** | **11** | **12** | **13** | **14** | **15** |  |
| RI | 1,45 | 1,49 | 1,51 | 1,48 | 1,56 | 1,57 | 1,59 |  |

Dengan membandingkan CI dan RI maka akan didapatkan patokan untuk menentukan tingkat konsistensi suatu matriks. Apabila rasio konsistensi semakin mendekati ke angka nol berarti semakin baik nilainya dan menunjukkan kekonsistenan matriks perbandingan tersebut.

## Penelitian Sebelumnya

Pada bagian ini akan dijelaskan mengenai penelitian-penelitian yang sudah pernah dilakukan terkait faktor-faktor penyebab kegagalan aplikasi, serta analisis persamaan dan perbedaan antara penelitian sebelumnya dan penelitian yang sedang dilakukan.

* + 1. ***Causes of Failure in Web Applications* (Pertet dan Narasimhan, 2005)** Penelitian ini berisi laporan tentang penyelidikan mengenai kegagalan dalam aplikasi berbasis web, dan melihat prevalensi dari kegagalan tersebut. Menurut laporan yang dikumpulkan oleh Pertet dan Narasimhan (2005) dari 40 insiden yang terdeteksi dari berbagai sumber, penyebab insiden pada aplikasi berbasis web terdiri dari :

1. Sebanyak 80% insiden terjadi karena *software failure* dan *human error*
2. Insiden terjadi kebanyakan saat perawatan rutin*, software upgrade*, dan integrasi sistem. Namun tidak jelas apakah insiden ini terjadi karena kompleksitas sistem, pengujian tidak memadai, atau pemahaman yang kurang tentang dependensi sistem.
3. Penyebab pentingnya lainnya dari insiden perangkat lunak adalah *system overload, resource exhaustion*, dan kompleksnya *fault recovery*
4. Pemadaman aplikasi selama beberapa menit, atau beberapa minggu. *Downtime* berkepanjangan sering terjadi ketika beberapa komponen terkena error
5. *Backup* menjadi lebih lama dari yang direncanakan

Dari laporan-laporan tersebut Pertet dan Narasimhan mengklasifikasikan insiden pada aplikasi berbasis web menjadi 4 kategori, yaitu *software failure, operator error, hardware and environmental failure*, dan *security violations*. Dari penelitian ini, ditemukan bahwa insiden terjadi 80% karena *software failure* dan *human error*. Kemudian, sejumlah besar insiden terjadi selama *change management*, namun belum jelas apakah insiden ini terjadi karena kompleksnya sistem, pengujian tidak memadai, atau pemahaman yang buruk mengenai dependensi sistem. Penyebab insiden lainnya adalah *system overload, resource exhaustion, dan fault recovery routine*. Masukan dari penelitian ini dapat

digunakan untuk mencari faktor-faktor penyebab insiden aplikasi perangkat lunak, walau penelitian ini fokus pada aplikasi berbasis web saja, namun pada dasarnya faktor-faktor insiden yang didefinisikan dalam penelitian ini dapat digunakan juga untuk aplikasi berbasis desktop.

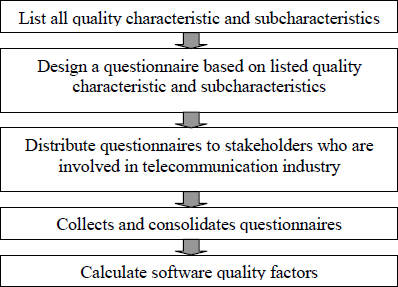
* + 1. ***Identifying Software Quality Factors for Tellecomunication Industri in Malaysia* (Hamid dan Hasan, 2011)**

Latar belakang dari penelitian ini adalah adanya pertumbuhan yang sangat cepat dari industri telekomunikasi pada beberapa tahun ini yang membawa permintaan yang tinggi terkait servis dan aplikasi terkait industri telekomunikasi dari berbagai level pengguna. Hal ini terjadi di Malaysia dengan adanya *Multimedia Super Corridor* (MSC) yang menunjukkan pertumbuhan dinamis dari status perusahaan 2192 MSC pada tahun 2009. Dari 2192 perusahaan MSC, 75% dari mereka bergerak pada industri telekomunikasi.

Terkait cepatnya pertumbuhan industri telekomunikasi tersebut, maka sudah saatnya industri telekomunikasi memiliki *software quality model* sendiri untuk memastikan sektor telekomunikasi dapat memberikan produknya sesuai jadwal, sesuai biaya, dan dengan kualitas tinggi. Oleh karena itu, sebelum membuat *software quality model*, penelitian ini membahas *software quality factor* untuk industri telekomunikasi untuk memberikan kualitas sebuah nilai yang relevan dengan tiga sudut pandang, yaitu sudut pandang manager untuk merepresentasikan dimensi ekonomi, sudut pandang pengguna untuk merepresentasikan dimensi sosial, dan sudut pandang pengembang untuk merepresentasikan dimensi teknik.

Faktor kualitas akan diukur menggunakan metodologi *open weight scale*. Gambar

2.11 akan menunjukkan aliran prosedur untuk mengidentifikasi faktor kualitas.



## Gambar 2.11 Proses penentuan faktor kualitas

(Hamid& Hasan, 2011)

Hasil dari penelitian yang dilakukan oleh Hamid dan Hasan (2011) menunjukkan bahwa dari 12 karakteristik kualitas aplikasi yang diidentifikasi dari *questionnaire*

– *Efficiency, Extendibility, Flexibility, Functionality, Reusability, Understandibility, Reliability, Maintainability, Portability, Usability, Completeness*, dan *Testability,* terdapat 5 karakteristik yang sesuai dengan industri telekomunikasi, yaitu *Extendibility, Flexibility, Functionality, Usability,* dan *Testability.*

### *Modelling Software Quality Evaluation Using The Analytical Hierarchy* Process (Yoon et al, 2011)

Penelitian ini mengajukan metode evaluasi kualitas aplikasi berdasarkan perbandingan produk yang dinilai oleh kepuasan pengguna dari segi kualitas. Dalam mengevaluasi aplikasi dari sudut pandang pengguna, terdapat dua masalah utama. Pertama, tidak semua pengguna sudah memiliki pengalaman dalam meenggunakan semua produk. Kedua, evaluasi produk membutuhkan justifikasi yang melibatkan beberapa evaluator.

Terdapat berbagai macam karakteristik dan kriteria kualitas aplikasi yang diajukan oleh para peneliti di bidang aplikasi engineering, oleh sebab itu dihadirkan ISO 9126 sebagai standar model untuk kualitas aplikasi dan ISO 12119 untuk paket aplikasi. ISO 9126-1 mendefinisikan 6 karakteristik, yaitu *functionality, reliability, usability, efficiency, maintainability,* dan *portability.* Semua model kualitas aplikasi harus dikaji terlebih dulu sebelum digunakan dalam evaluasi untuk menyesuaikan dengan sudut pandang evaluator atau level aplikasi. Untuk mendapatkan bobot kualitas, penelitian ini mengadopsi ISO 9126 sebagai kriteria evaluasi, dan AHP ssebagai metode evaluasi.

AHP/ANP dianggap sebagai metode yang tepat ketika evaluasi tergantung pada penilaian subjektif. Kesesuaian antara AHP dan ISO 9126 dapat dilihat pada Tabel berikut:

## Tabel 2.7 Kesesuaian antara ISO 9126 dan AHP

|  |  |
| --- | --- |
| **Kebutuhan ISO 9126** | **AHP** |
| Relasi hirarki dari karakteristik kualitas  Mencakup semua aspek dari kualitas S/W  Deskripsi dengan kemungkinan overlap yang kecil  Membentuk suatu set yang tidak lebih dari 6 sampai 8  Menggunakan terminologi yang sudah ada | Evaluasi diuraikan dalam bentuk hirarki  Struktur hirarki yang lengkap Kriteria independen dari alternatif  Elemen dalam satu set untuk dibandingkan  Metode ilmiah untuk penilaian subjektif |

Yoon et all, 2011– telah diolah kembali

Salah satu masalah dalam metode yang diajukan adalah mengatasi nilai yang hilang pada matriks. Matriks perbandingan berpasangan membutuhkan n(n-1)/2 kali perbandingan. Namun, ketika penelitian ini tidak memenuhi jumlah yang dipersyaratkan, maka nilai yang hilang tersebut harus ditaksir oleh nilai perbandingan lain. Penelitian ini memberikan perbandingan antara indeks kualitas dengan 2 nilai yang hilang dan 4 nilai yang hilang.

Hasil kombinasi dari perbandingan tersebut, terlihat bahwa hasil dari dua vector tersebut tidak berbeda dari segi SI (Indeks kompatibilitas) = 1.005. Maka

penelitian ini menyarankan untuk menggabungkan perbandingan berpasangan pada AHP yang dapat memberikan hasil yang lebih valid daripada evaluasi alternatif individual, dan menutupi bahwa jumlah evaluator untuk beberapa produk tertentu tidak cukup serta evaluator tidak memiliki pengalaman atau menilai semua fokus alternatif pada kategori yang sama. Untuk ke depannya, diharapkan penelitian ini dapat menjadi basis untuk penelitian empiris untuk mengklarifikasi hubungan antara persepsi kualitas dan indicator objektif dari kualitas eksternal.

* + 1. ***Case Studies of Most Common and Severe Types of Software System Failure* (Dalal dan Chhillar, 2012)**

Pada penelitian ini Dalal dan Chhillar (2012) mendeskripsikan studi kasus tentang *software system failure* yang sering terjadi dan tingkat *severity* dari kegagalan tersebut dalam indusri perangkat lunak. Pada intinya penelitian ini akan mempertimbangkan studi kasus yang menunjukkan ancaman, resiko, dan kegagalan sistem perangkat lunak yang mendukung keamanan bangsa, perbankan, penerbangan, kesehatan, dan jejaring social yang digunakan secara global.

Kriteria *severity* dari kegagalan aplikasi menurut Dalal dan Chhillar (2012) adalah yang dianggap berdampak paling parah pada kehidupan orang-orang, dan menyebabkan kerugian modal yang besar. *Severity* ditentukan atas dasar beberapa parameter sebagai berikut :

1. Jumlah pengguna aplikasi
2. Keterlibatan transaksi moneter
3. Jenis penggunaan aplikasi
4. Dapatkah aplikasi berdampak pada kehidupan orang-orang jika gagal Berikut merupakan studi kasus yang diteliti oleh Dalal dan Chhillar (2012):

## Tabel 2.8 Daftar Beberapa Kegagalan Aplikasi dan Tingkat *Severity*nya

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Aplikasi** | **Deksripsi insiden** | **Kerugian** |
| 1 | ERP project *failure* di Jordan | Adanya kesenjangan yang cukup besar antara asumsi dan persyaratan yang dibangun ke dalam desain sistem ERP, dan realitas yang sebenarnya dari organisasi | Mengalami kerugian modal  dalam jumlah  besar, dan  ketidakpuasan klien |
| 2 | Ariane 5 | Ariane 5 merupakan roket tak berawak terbaru dari Eropa sengaja dihancurkan beberapa detik setelah diluncurkan pada penerbangan pertamanya bersama dengan muatannya yaitu empat satelit ilmiah untuk mempelajari bagaimana medan magnet bumi berinteraksi dengan  angin surya | Kerugian 10 tahun kerja keras dan  $100 million. Reputasi ESA (Europian Space Agency) memburuk |
| 3 | Therac – 25 | Mesin terapi radiasi Therac-25 milik Kanada mengalami malfungsi dan mengirimkan dosis radiasi  mematikan pada pasien | Banyak orang meninggal dunia dan terluka parah |
| 4 | STS – 126 | Perubahan perangat lunak telah secara tidak sengaja mengubah data pada kode perangkat lunak  penerbangan pesawat ulang alik | Anomali software in flight terjadi dan beberapa fungsi  otomasi gagal |
| 5 | Automated airport baggage  handling (DIA) | Kegagalan mengantisipasi jumlah cart mengakibatkan keterlambatan dalam mengambil tas yang akan melemahkan kinerja sistem | Biaya pemeliharaan bulanan melebihi  investasi pengguna bulanan |

Dari penelitian terkait 5 studi kasus di atas Dalal dan Chhillar (2012) menyimpulkan bahwa terdapat 6 penyebab kegagalan aplikasi, yaitu :

1. Kurang jelasnya definisi *goals* dan spesifikasi aplikasi
2. Manajemen yang buruk, dan komunikasi yang buruk antara konsumen, desainer, dan *programmer*
3. Langkah yang salah dalam mencatat insiden (*improper fault assignment*)
4. Adanya tekanan institusional (*low budget, fast delivery*)
5. Penggunaan teknologi baru dengan ketidakpastian reliabilitas dan masalah
6. Penolakan pengakuan bahwa proyek dalam masalah
   * 1. ***Insider Threat Behaviour Factors: A Comparison of Theory with Reported Incidents* (Munshi, Dell, dan Armstrong, 2012)**

Insiden internal dapat menyebabkan lebih dari sekedar kehilangan finansial, yaitu kehilangan konsumen dan rusaknya nama baik perusahaan. Penelitian ini memeriksa beberapa model teoritis yang ditarik dari literatur akademis untuk mengidentifikasi seperangkat faktor yang dianggap sebagai faktor perilaku terkait dengan insiden internal. Faktor-faktor tersebut kemudian akan dikritik menggunakan bukti empiris dari laporan insiden yang akan menghasilkan pengetahuan mendalam ke area di mana perspektif teoritis dari literatur akademis didukung atau tidak oleh bukti kasus nyata.

Ancaman yang ditimbulkan oleh orang dalam di organisasi terus menerus menjadi salah satu masalah utama bagi organisasi dan institusi pemerintahan pada sektor infrastruktur kritis. Penelitian ini melibatkan tiga aktivitas utama. Pertama, literatur akademik ditinjau dan dipelajari secara mendalam untuk mengidentifikasi faktor yang dianggap mempengaruhi perilaku orang dalam. Kedua, laporan insiden yang dipublikasikan dianalisis untuk menentukan seperangkat faktor yang telah diobservasi dalam kasus nyata terkait perilaku nyata dari *insider threat*. Terakhir, informasi empiris dari langkah ke dua digunakan untuk mengkritik faktor teoritis dari literatur yang akan menghasilkan beberapa pengetahuan ke dalam area di mana literatur teoritis kurang baik, dan di mana dibutuhkan investigasi lebih jauh. Pengetahuan inilah yang akan menjadi dasar rekomendasi yang dibuat pada akhir penelitian.

Terdapat total 13 laporan yang diambil dari 500 kasus insiden internal dari CERT yang diinvestigasi pada penelitian ini. Berdasarkan CERT, *insider threat* dibagi menjadi 3 kategori utama, yaitu *IT Sabotage, fraud,* dan *Theft*. Beberapa faktor yang paling banyak berkontribusi pada perilaku *insider threat* yang muncul dari investigasi literature penelitian terdahulu, antara lain akses dan tingkat kepercayaan, posisi dan keahlian teknikal orang dalam, motivasi, *outsourcing*, faktor budaya, kebijakan keamanan informasi, faktor psikologi, *remote access*, dan jenis kelamin.

Perbandingan dengan bukti laporan insiden membuktikan bahwa penelitian akademik teoretis telah mengabaikan jenis kelamin sebagai faktor yang penting. Bukti laporan insiden menunjukkan bahwa jenis kelamin laki-laki merupakan faktor penyebab paling banyak dalam kasus CERT. Sama halnya dengan *remote access*, penelitian akademis mengabaikan faktor ini pada perilaku *insider threat*. Kedua, dimungkinkan untuk literatur teoretis yang menekankan pada pentingnya posisi teknikal yang dipegang dalam organisasi benar-benar mengemukakan pentingnya akses tersebut; mereka yang memiliki posisi teknikal secara definisi memiliki akses sedangkan karyawan yang memiliki posisi teknikal rata-rata memiliki keahlian teknis, keahlian tersebut lah yang secara tidak sengaja menjadi penyebab banyaknya kasus insiden internal.

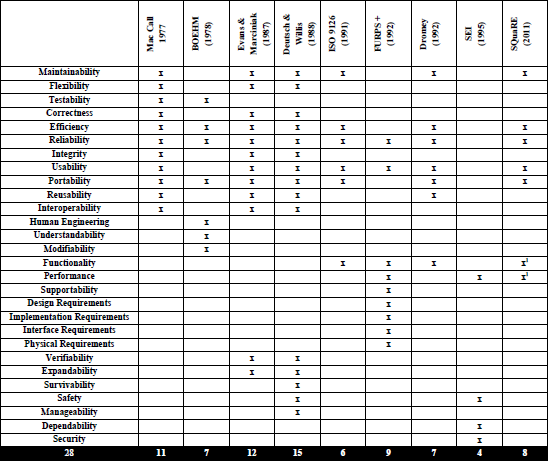
Ketiga, walaupun banyak dari literatur akademis menyarankan bahwa *outsourcing* dan faktor budaya merupakan hal yang penting dalam perilaku *insider threat*, ternyata sangat sedikit bukti empiris yang mendukung hal ini. Bukti empiris yang mendukung pernyataan akademis bahwa kebijakan keamanan dan budaya kebijkan merupakan faktor dalam perilaku *insider threat* sangat bervariasi. Hal ini tidak terlalu umum, namun terdapat beberapa laporan yang mempertimbangkan kebijakan merupakan faktor yang berkontribusi dalam membantu serangan orang dalam. Literatur akademik tidak sepakat terkait pentingnya faktor psikologi, namun bukti empiris dari laporan insiden mengusulkan bahwa kecenderungan pribadi dan perilaku merupakan faktor yang umum dalam insiden internal. Terakhir, bukti dari laporan insiden mendukung perspektif teoritis bahwa motivasi orang dalam dapat menjadi faktor *insider threat*.

* + 1. ***Comparative Study of Software Quality Model* (Youness, Abdelaziz, dan Habib, 2013)**

Penelitian ini menyajikan tentang konten dan perbandingan dari berbagai *software quality model* : *McCall’s Quality Model, Boehm’s Quality Model, Evans & Marciniak Model, Deutsch & Willis Quality Model, ISO 9126’s Quality Model, Dromey's Quality Model, FURPS+ Quality Model, SEI Model* dan *ISO 25000 (SQuaRE) Model.* Tujuan dari penelitian ini adalah untuk menghitung model kualitas dan serta karakteristiknya dan untuk menyajikan kepada pengguna dan

pengembang beberapa solusi dari perhitungan kualitas sehingga dapat memilih model-model kualitas aplikasi serta faktor yang sesuai dengan kebutuhan mereka. Berikut merupakan hasil dari perbandingan 9 *software quality model* :

## Tabel 2.9 Perbandingan Model Kualitas



Dari 28 karakteristik, hanya 1 karakteristik yang dimiliki oleh 8 model kualitas kecuali model SEI, yaitu *reliability*. Kemudian terdapat 3 karakteristik (*efficiency, usability* dan *portability)* yang dimiliki oleh 7 model. *Maintainability* hanya dimiliki oleh 6 model kualitas. Sedangkan *Functionality* dan *Reusability* dimiliki oleh 4 model. 5 karakteristik (*Performance, Interoperability, Integrity, Correctness and Flexibility characteristics*.) dimiliki oleh 3 model, dan 4 karakteristik dimiliki oleh 2 model (*Safety, Expandability, Verifiability,* dan *Testability*). Dan terakhir, 12 karakteristik hanya dimiliki oleh 1 model, yaitu *Manageability, Dependability, Security, Survivability, Supportability, Design Requirements, Implementation Requirements, Interface Requirements* dan *Physical Requirements*

* + 1. ***Quality and Human Errors in IT Service Infrastructure* (Saaralainen dan Jantti, 2015)**

Masalah pada penelitian ini adalah bagaimana kualitas servis dapat ditingkatkan dengan menganalisis insiden dengan *root cause* terkait *human error. Human error* diklasifikasikan dengan *Human Factors Analysis and Classification System* (HFACS) dan juga berdasarkan *IT Service Management Processes*. Penelitian dilakukan terhadap 18 laporan insiden.

Kontribusi utama dari penelitian ini adalah untuk menunjukkan bagaimana taksonomi HFACS dapat diaplikasikan pada ITSM analisis trend insiden. Hasil penelitian yang dilakukan Saaralainen dan Jantti (2015) menunnjukkan terdapat 3 jalur error penyebab insiden.

**Jalur 1**: Proses definisi atau instruksi yang tidak memadai, yang menyebabkan *error.* Akar penyebab utamanya adalah kurangnya persyaratan untuk menguji konfigurasi baru setelah perubahan (insiden dan efeknya pada QoS : *Network outage, service break pada SAP router, message server crash, storage unavailability*)

**Jalur 2**: Operasi pengujian tidak dilakukan sesuai rencana. Hal ini dapat terjadi karena hilangnya *requirement* dalam proses definisi. (insiden dan efeknya pada QoS : *Network outage, slowness in Internet services, unavailability of web services, problem at faktory production line, problem in storage*)

**Jalur 3**: Instruksi pengoperasian yang tidak diikuti karena kurangnya pengawasan (insiden dan efeknya pada QoS : *Network outage, unavailability of web site, unavailability of faktory production line, slowness in Internet services, problem in storage.)*

Temuan lainnya adalah saat proses sebagai *root cause* ditambahkan pada analisis sebagai dimensi ekstra, ditemukan 86% insiden terkait *change management*. Dari temuan yang dihasilkan dari penelitian ini maka variabel faktor penyebab insiden yang akan digunakan adalah *human error* dan *change management.*

## Perbandingan Antara Penelitian Terdahulu dan Penelitian Saat Ini

Bagian ini akan menjelaskan tentang persamaan dan perbedaan antara 6 penelitian sebelumnya yang telah dijelaskan di atas dengan penelitian yang saat ini sedang dilakukan yang dapat dilihat pada Tabel 2.10 berikut :

## Tabel 2.10 Tabel Perbandingan antara penelitian sebelumnya dan penelitian saat ini

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Penulis** | **(Pertet & Narasimhan, 2005)** | **(Hamid & Hasan, 2011)** | **(Yoon, Lee, & Whang, 2011)** | **(Dalal & Chhillar, 2012)** | **(Munshi, Dell, & Armstrong, 2012)** | **(Youness, Abdelaziz, & Habib, 2013)** | **(Saaralainen & Jantti, 2015)** | **Penelitian Penulis (2016)** |
| **Judul** | *Causes of Failure in Web Applications* | *Identifying Software quality factors for Tellecomunication*  *Industri in Malaysia* | *Modeling Software Quality Evaluation Using The Analytic Hierarchy Process* | *Case Studies of Most Common and Severe Types of Software System Failure* | *Insider Threat Behaviour Factors: A Comparison of theory with reported incidents* | *Comparative Study of Software Quality Models* | *Quality and Human Errors in IT Service Infrastruxcture* | Analisis Faktor-Faktor yang menyebabkan insiden pada Aplikasi: Studi Kasus PT. Telkomsel |
| **Pertanyaan Penelitian** | Apakah penyebab dari kegagalan dan kegagalan yang sering terjadi pada aplikasi berbasis web  ? (Implisit) | Apa saja faktor- faktor yang mempengaruhi kualitas perangkat lunak di industri telekomunikasi ? (Implisit) | Bagaimana metode evaluasi persepsi pengguna terhadap kualitas perangkat lunak mewakili karakteristik penggunaanserta apa saja poin kelemahan dan kelebihan yang berkontribusi pada preferensi pasar? (Implisit) | Apakah penyebab umum dari kegagalan aplikasi berdasarkan studi kasus terbaru? (Implisit) | Bagaimana perbandingan antara faktor perilaku *insider threat* dari studi akademis dengan bukti empiris dari laporan insiden? (Implisit) | Apa sajakah *software quality model* yang ada kemudian bagaimana perbandingannya? (Implisit) | 1. Bagaimana taksonomi HFACS dapat diaplikasikan ke analisis tren insiden ITSM? 2. Apakah urutan kejadian tipikal, jalur kegagalan pada kesalahan manusia terkait insiden servis TI? 3. Bagaimana perspektif organisasi dari HFACS dapat dikombinasikan dengan proses terkait framework ITIL pada   analisis insiden? | Bagaimanakah prioritas dari faktor-faktor penyebab Insiden Aplikasi di PT. Telkomsel?” |
| **Ruang Lingkup** | Faktor-faktor penyebab insiden pada Aplikasi berbasis web | Faktor kualitas pada perangkat lunak di industri telekomunikasi | Evaluasi kualitas aplikasi berdasarkan perbandingan produk, dinilai dari persepsi kepuasan pengguna dalam hal kualitas | Faktor-faktor penyebab insiden pada *Software System* | Faktor penyebab *Insider Threat* dalam Insiden | Perbandingan antar model kualitas perangkat lunak | Faktor kesalahan manusia pada insiden servis TI | Faktor-faktor penyebab insiden pada aplikasi di PT. Telkomsel |
| **Jenis Penelitian** | Studi Kasus (kualitatif) | Studi Kasus (kuantitatif) | Studi Kasus  *(Mixed Method)* | Studi Kasus (kualitatif) | Studi Kasus (Kualitatif) | Studi Literatur (kualitatif) | Studi Kasus (kuantitatif) | Studi Kasus *(Mixed Method)* |
| **Metode Pengumpulan Data** | Survey dan memeriksa insiden website yang tercantum di website CNET.com dan  eweek.com | Kuesioner | Kuesioner dan Wawancara | *-* | - | **-** | Analisis Laporan Insiden dan Wawancara | Kuesioner dan Wawancara |
| **Metode Olah data** | *Fault Chain* | *Open weight scale* | AHP | - | **-** | **-** | *Fisher’s Exact test HFACS*  *ITSM Process* | AHP |
| **Objek Penelitian** | CNET.com; Eweek.com | Industri Telekomunikasi di Malaysia | *Software Quality Model* | ERP *project failure* in Jordan;  Ariane 5; Therac-25; STS-126;  Automated airport baggage handling(DIA) | 13 laporan internal insiden dari CERT | 9 model kualitas perangkat lunak : McCall, Boehm, Evans & Marciniak, Deutsch & Willis, ISO 9126, Dromey, FURPS+, SEI, ISO 25000 | 18 Laporan Insiden | Insiden aplikasi *(key operatioal)*di PT Telkomsel |

**Tabel 2.10 Tabel Perbandingan antara penelitian sebelumnya dan penelitian saat ini (sambungan)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Penulis** | **(Pertet & Narasimhan, 2005)** | **(Hamid & Hasan, 2011)** | **(Yoon, Lee, & Whang, 2011)** | **(Dalal & Chhillar, 2012)** | **(Munshi, Dell, & Armstrong, 2012)** | **(Youness, Abdelaziz, & Habib, 2013)** | **(Saaralainen & Jantti, 2015)** | **Penelitian Penulis (2016)** |
| **Hasil** | Didapatkan 4 faktor penyebab insiden: ***Software failure; Human & operator error;***  ***Hardware and environmental failure;***  ***Security violations;***dan satu faktor yang implisit yaitu,  ***Fault recovery*** | Didapatkan 5 *Software quality factor* untuk industri telekomunikasi: ***Extendibility, Flexibility, Functionality, Usability, Testability*** | Modifikasi metode evaluasi kualitas aplikasi dinilai dari persepsi kepuasan pengguna | Secara implisit didapatkan 6 faktor- faktor penyebab kegagalan aplikasi: ***Software failure; Human & operator error;***  ***Hardware and environmental failure; Fault recovery; Communication management ; Institusional pressure*** | 1. Penelitian akademik mengabaikan jenis kelamin dan *remote access* sebagai faktor penyebab *insider threat* 2. Penelitian akademik yang menekankan posisi dan keahlian teknikal orang dalam sebagai faktor penyebab insider threat benar-benar menggambarkan pentingnya hal tersebut 3. Walau penelitian akademik banyak menekankan pada *outsourcing* dan faktor kebudayaan, namun sedikit sekali bukti empiris dari laporan insiden 4. Terdapat beberapa laporan yang mempertimbangkan kebijakan merupakan faktor yang berkontribusi dalam membantu serangan orang dalam 5. Literatur akademik tidak sepakat terkait pentingnya faktor psikologi, namun bukti empiris dari laporan insiden mengusulkan bahwa kecenderungan pribadi dan perilaku merupakan faktor yang umum dalam insiden internal 6. Bukti dari laporan insiden mendukung perspektif teoritis bahwa motivasi orang dalam dapat menjadi faktor insider threat. | Perhitungan model kualitas dan karakteristik masing-masing model | 1. Hampir semua laporan insiden (71%) ditemukan kurangnya analisis kondisi yang menyebabkan kegagalan. HFACS harusnya sudah digunakan pada investigasi insiden 2. Terdapat 3 jalur kegagalan 3. Mengkombinasikan HFACS dan ITIL dalam matriks dua dimensi sangat berguna untuk menunjukkan defect | Hasil yang diharapkan adalah pemeringkatan dari faktor-faktor penyebab insiden aplikasi di PT Telkomsel |
| ***Limitations*** | - | - | Evaluator tidak memiliki pengalaman secara empiris atau menilai semua fokus alternatif pada kategori yang sama | - |  | - | Jumlah laporan insiden yang ditulis dengan baik sangat sedikit yang menyebabkan kredibilitas statistic yang rendah dan sebagian dari taksonomi HFACS tidak kompatibel dengan konsep ITSM |  |
| ***Future Research*** |  | - | Penelitian empiris untuk mengklarifikasi hubungan antara persepsi terhadap kualitas dan indikator tujuan dari kualitas eksternal |  | 1. Penelitian akademik terkait isu *insider threat* dengan faktor jenis kelamin khususnya laki-laki serta *remote access* 2. Investigasi keahlian teknis yang mana yang merupakan faktor penyebab *insider threat* 3. Penelitian terkait pentingnya *outsourcing* dan faktor budaya sebagai penyebab *insder threat* 4. Penelitian mengenai pentingnya kebijakan keamanan dan budaya kebijkan sebagai faktor yang berkontribusi dalam membantu serangan orang dalam | - | Adaptasi TI untuk taksonomi HFACS serta terdapat model yang dapat mengintegrasikan semua aspek kegagalan yang dapat menjelaskan bahkan memprediksi insiden |  |

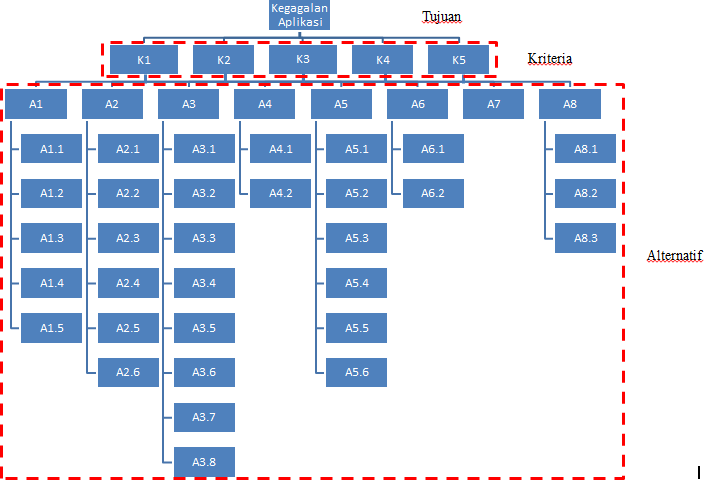
Dari Tabel 2.10 di atas, dapat dilihat persamaan dan perbedaan antar penelitian sebelumnya serta antara penelitian-penelitian sebelumnya dengan penelitian yang saat ini dilakukan serta kelemahan dan kelebihan dari setiap penelitian. Kelemahan dari penelitian yang dilakukan oleh Pertet dan Narasimhan (2005) dan Dalal dan Chhillar (2012) adalah pertanyaan penelitian tidak didefinisikan secara eksplisit, kemudian tidak terdapat limitasi dan saran penelitian selanjutnya

Kekurangan dari penelitian yang dilakukan oleh Dalal dan Chhillar (2012) adalah selain pertanyaan penelitian yang tidak didefinisikan secara eksplisit, peneliti tidak mencantumkan metode pengumpulan data serta metode olah data yang digunakan. Penelitian ini juga tidak menunjukkan batasan dan saran penelitian selanjutnya. Sedangkan untuk penelitian yang dilakukan oleh Saaralainen dan Jantti, 2015 sudah cukup lengkap. Pertanyaan penelitian didefinisikan secara eksplisit, limitasi dan saran penelitian selanjutnya juga sudah didefinisikan.

Pada umumnya, kekurangan dari penelitian-penelitian tersebut adalah tidak mencantumkan pertanyaan penelitian secara eksplisit, seperti penelitian yang dilakukan Hamid dan Hasan (2011), Youness, et al (2013), dan Yoon, et al (2011). Kekurangan lainnya dari penelitian Hamid dan Hasan, 2011 adalah tidak mencantumkan limitasi dan saran penelitian selanjutnya, sedangkan untuk penelitian Youness et all, 2013 kekurangannya cukup banyak, yaitu tidak mencantumkan metode pengolahan data, metode olah data, limitasi, dan saran penelitian selanjutnya.

### 2.5 *Theoretical Framework*

Dari berbagai teori, *model/framework*, dan metode yang telah dibahas, berikut merupakan kerangka teoritis penelitian:



**Gambar 2.12 *Theoretical Framework***

Kerangka teoritis di atas mengadaptasi model hierarki dari AHP yang menggambarkan hubungan antara tujuan, kriteria, dan alternatif. Level tertinggi dari hierarki merupakan tujuan yaitu kegagalan aplikasi. Level kedua merupakan kriteria aplikasi berkualitas untuk industri telekomunikasi yaitu *Extendibility*, *Flexibility*, *Functionality*, *Usability*, dan *Testability.* Untuk menyesuaikan dengan tujuan maka kriteria ini akan digunakan sebagai kriteria aplikasi yang tidak berkualitas yang menyebabkan kegagalan aplikasi. Kriteria yang digunakan menjadi *Unextendibility, Inflexibility, Unfunctionality, Unusability,* dan *Untestability.* Pemilihan kriteria kualitas aplikasi sudah dilakukan pada subbab

2.2.4. Kriteria-kriteria ini akan menjadi indikator pembanding dari alternatif.

Level paling akhir adalah alternatif yang merupakan faktor penyebab kegagalan pada aplikasi yang telah dibahas pada ssub bab 2.2.6. Level 3 terdiri dari 2 bagian, yaitu kategori faktor dan faktornya itu sendiri. Faktor-faktor ini akan disusun dan

diperingkatkan untuk memudahkan pengambilan keputusan terkait strategi mengatasi dan menghilangkan insiden.

Berikut ini adalah penjelasan dari Kriteria dan Alternatif pada Gambar 2.12 : K1 : *Extendibility*

K2 : *Flexibility* K3 : *Functionality* K4 : *Usability*

K5 : *Testability*

A1 : ***Software Error***

* + 1. : *System overload*
    2. : *Computational /logic error*
    3. : *Resource exhaustion*
    4. : *Failed software upgrade*
    5. : *Fault recovery routine*

### A2 : *Human/Operator Error*

* + 1. : *Configuration error*
    2. : *Procedural error*
    3. : *Miscellaneous accidents*
    4. : *Organizational process*
    5. : *Inadequate supervision*
    6. : *Planned inappropriate operations*

### A3 : *Hardware and Environmental Error*

* + 1. : *Device driver failure*
    2. : *I/O error*
    3. : *Memory parity error*
    4. : *Network hardware failure*
    5. : *Power outages*
    6. : *Overheating*
    7. : *High humidity*
    8. : *Natural disaster*

### A4 : *Fault Recovery Error*

* + 1. : *Improper fault assignment*
    2. : *Complex fault recovery* **A5 *: Security Violations*** A5.1 : *Password disclosures*

A5.2 : *Denial of Service (DOS) attacks*

A5.3 : *Worms and viruses* A5.4 : *Browser vulnerabilities* A5.5 : *Authentication failures* A5.6 : *Insider Threat*

### A6 *: Change Management Issue*

A6.1 : *Planning the change*

A6.2 : *Testing the change*

### A7 *: Communication Management Issue*

**A8 *: Institusional Pressure***

A8.1 : *Unrealisticcaly Low bids*

A8.2 : *Unrealistically low budget requests*

A8.3 : *Underestimates of time requirements*

## BAB 3 METODOLOGI PENELITIAN

Bab ini akan membahas tentang rancangan penelitian, tahapan penelitian, metode dan proses olah data, serta instrument penelitian.

## Rancangan Penelitian

Rancangan penelitian merupakan rencana tentang cara mengumpulkan dan mengolah data agar dapat dilaksanakan untuk mencapai tujuan penelitian. Dilihat dari ruang lingkup penelitian ini yang terbatas pada insiden yang terjadi di PT. Telkomsel saja, maka penelitian ini tergolong penelitian Studi kasus, dimana ruang lingkup penelitian dibatasi pada organisasi, komunitas, atau negara tertentu. Berdasarkan tujuan dan manfaat dari penelitian ini, maka penelitian ini termasuk pada *applied research,* yaitu penelitian yang dirancang untuk menyelesaikan masalah praktis, daripada untuk mendapatkan pengetahuan (akademis). Berdasarkan tujuan dari penelitian ini, maka penelitian ini termasuk dalam penelitian kausal yang bertujuan untuk membuktikan hubungan sebab akibat dari variabel yang diteliti, yaitu kriteria kualitas aplikasi (*dependent variabel*) dan faktor penyebab kegagalan aplikasi (*independent variabel*).

Sumber data dari penelitian ini adalah data primer dan data sekunder. Data primer diperoleh dari perusahaan yang dipilih sebagai objek penelitian melalui wawancara dan kuesioner kepada para pakar di Departemen TI PT Telkomsel. Data sekunder didapat dari berbagai jurnal internasional, dan buku referesi yang digunakan untuk pembentukan *theoretical framework*.

Berdasarkan jenis data dan metode olah data yang digunakan, yaitu AHP maka penelitian ini menggunakan metode gabungan, yaitu penggabungan antara pendekatan kuantitatif dan pendekatan kualitatif. Menurut Cresswell (2002), pendekatan kuantitatif merupakan pendekatan yang dilakukan melalui eksperimen atau survey dengan menggunakan pernyataan postpositivist untuk menguji suatu

teori. Sedangkan pendekatan kualitatif dapat dikatakan sebagai suatu proses pemahaman akan suatu permasalahan manusia atau sosial berdasarkan gambaran holistik lengkap yang dibentuk dengan kata-kata, melaporkan pandangan

informan secara terperinci, dan disusun dalam sebuah latar alamiah. Konsep metode gabungan ini mungkin pertama kali muncul pada tahun 1959, ketika Campbell dan Fiske menggunakan beberapa metode untuk menguji validitas dari sifat-sifat psikologis.

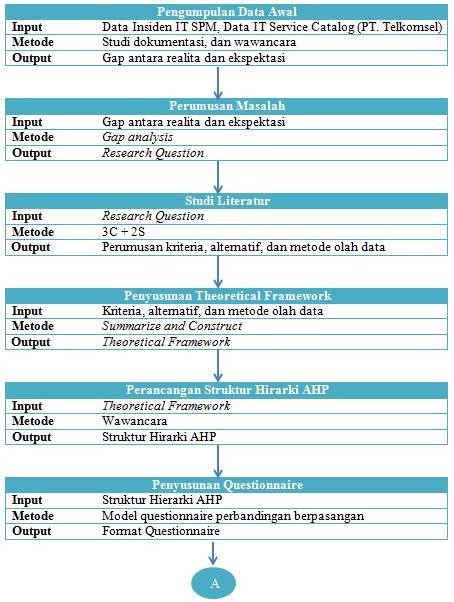
Berikut merupakan ringkasan dari desain penelitian yang telah dijabarkan di atas :

**Tabel 3.1 Tabel *research design***

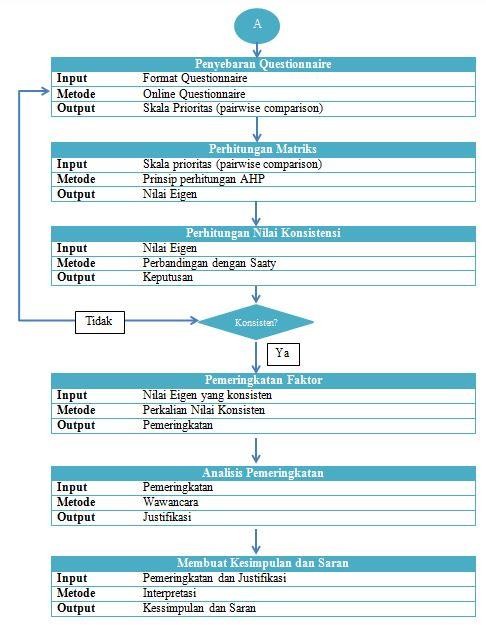
|  |  |
| --- | --- |
| **Elemen** | **Jenis Penelitian** |
| Klasifikasi | Penelitian Studi Kasus |
| Manfaat | *Applied Research* |
| Sumber data | Data Primer dan Data Sekunder |
| Jenis Data | *Mixed Method* (Kuantitatif dan Kualitatif) |
| Tujuan dan  manfaat | *Causal Research* |
| **Keterangan** | |
| Tujuan | Mengetahui dan mengidentifikasi prioritas dari faktor-faktor  yang menyebabkan kegagalan aplikasi di PT. Telkomsel |
| Metode  Pengumpulan Data | Studi literatur, wawancara, dan kuisioner |
| Responden | Para pakar di Departemen IT Service Management PT  Telkomsel |
| Tools | Microsoft Excell, Expert Choice 11 |
| Metode Olah  Data | *Analytic Hierarchy Process* (AHP) |
| Instrumen  Penelitian | 5 *Independent Variabel* (kriteria), 5 *Dependent Variabel*  (alternatif) |

## Tahapan Penelitian

Tahapan penelitian ini dapat dilihat pada Gambar 3.1 berikut :



## Gambar 3.1 Tahapan Penelitian



**Gambar 3.1 Tahapan Penelitian (sambungan)**

Berikut merupakan penjelasan dari setiap tahap yang ada pada Gambar 3.1 :

1. Pengumpulan data awal

Pada tahap ini, penulis melakukan pengumpulan data awal dari objek penelitian yang berupa data insiden dan *Service Level Agreement* (SLA) PT. Telkomsel. Dari data tersebut kemudian dilakukan studi dokumentasi dan wawancara dengan Manager System Integration Test Management PT. Telkomsel untuk mendapatkan realita dan ekspektasi dari perusahaan.

1. Perumusan masalah

Dari realita dan ekspektasi tersebut, dilakukan *gap analysis* untuk mencari akar masalah yang terjadi di perusahaan. Masalah ini akan diangkat menjadi *Research Question* (RQ) yang harus dijawab dalam penelitian ini, yaitu **“Bagaimanakah prioritas dari faktor-faktor yang menyebabkan kegagalan aplikasi terkait kualitas aplikasi di PT Telkomsel?”**

1. Studi literatur

Pada tahap ini, penulis melakukan studi literatur terhadap teori, model, metode olah data, dan penelitian-penelitian sebelumnya untuk mencari persamaan dan perbedaan antar penelitian mengenai variabel dan metode yang relevan terhadap *Research Question*. Penulis melakukan 3C + 2 S (*compare, contrast, criticize, synthesize,* dan *summarize)* antar penelitian sebelumnya untuk merumuskan variabel-variabel dan metode olah data yang dapat memecahkan RQ. Dari proses 3C + 2S tersebut, didapatkan kriteria sebagai *dependent variabel* (variabel akibat), dan alternatif (faktor-faktor) sebagai *independent variabel* (variabel sebab), serta AHP sebagai metode olah data yang sesuai dengan kondisi penelitian saat ini.

1. Penyusunan *theoretical framework*

Berdasarkan variabel-variabel penelitian dari tahap studi literatur, dilakukan penyusunan *theoretical framework* yang akan menjadi acuan untuk pemecahan RQ. Kerangka ini akan menggambarkan hubungan antara kriteria dan alternatif dengan mengadopsi struktur hirarki dari metode olah data yang digunakan, yaitu AHP.

1. Perancangan struktur hirarki AHP

Untuk menguji variabel-variabel yang digunakan dalam *Theoretical Framework* yang telah dibangun pada tahap sebelumnya, dilakukan wawancara pada para pakar di Departemen TI PT. Telkomsel untuk mengetahui apakah perlu penambahan atau pengurangan terhadap variabel- variabel tersebut untuk menyesuaikan dengan kondisi perusahaan. Dari hasil wawancara tersebut, akan dibangun sebuah struktur hirarki AHP yang terdiri dari 3 level, yaitu tujuan, kriteria, dan alternatif. Pada penelitian ini, yang akan menjadi tujuan adalah kegagalan aplikasi, kemudian yang menjadi kriteria

adalah kriteria *software quality factor*. Sementara itu, yang akan menjadi alternatif adalah faktor-faktor penyebab kegagalan / insiden pada aplikasi.

1. Penyusunan kuisioner

Format kuisioner disusun berdasarkan kriteria dan alternatif yang ada pada struktur hirarki AHP yang dapat mengakomodasi *pairwise comparison* dengan skala penilaian yang direkomendasikan oleh Saaty. Perbandingan akan dilakukan pada antar elemen kriteria, antar kategori alterntif pada setiap kriteria, dan antara antar alternatif pada setiap kategori alternatif.

1. Penyebaran kuisioner

Pada tahap ini, penulis akan menyebarkan kuisioner yang telah disusun pada tahap sebelumnya kepada pakar dari Departemen IT Service Management di PT Telkomsel. Hasil dari kuisioner ini berupa skala prioritas antar elemen sesuai dengan skala penilaian yang direkomendasikan oleh Saaty.

1. Perhitungan matriks

Dari skala penilaian tersebut, akan dilakukan perhitungan matriks. Perhitungan matriks dilakukan untuk mendapatkan nilai dari setiap kriteria dan alternatif. Perhitungan ini berdasarkan prinsip *eigenvector*. Untuk mendapatkan nilai eigen dapat dilakukan tahapan sebagai berikut (Marimin, 2004) :

* 1. Ubah matriks dengan nilai dalam bilangan decimal
  2. Kuadratkan matriks tersebut
  3. Jumlahkan nilai dari setiap baris matriks
  4. Normalisasikan matriks tersebut
  5. Lanjutkan iterasi berikutnya
  6. Iterasi berhenti jika perbedaan jumlah nilai dari dua perhitungan berturut- turut tidak lebih besar dari suatu nilai batas tertentu.

Untuk memudahkan perhitungan eigenvector tersebut, maka penulis akan menggunakan aplikasi Expert Choice 11 untuk mendapatkan nilai eigen dari setiap kriteria dan alternatif.

1. Perhitungan nilai konsistensi

Perhitungan konsistensi ini diperlukan untuk mengetahui konsistensi jawaban yang berpengaruh terhadap kebenaran dari hasil yang didapat. Nilai rasio CR

harus sama atau lebih kecil dari 10% (Saaty, 2012). Jika lebih dari 10% maka terdapat kesalahan pada penilaian tersebut dan perlu diperbaiki dengan kembali ke tahap penyebaran kuisioner.

1. Pemeringkatan faktor

Setelah mendapatkan nilai eigen yang telah diuji kekonsistenannya, maka untuk mendapatkan pemeringkatan akan dilakukan perkalian nilai eigen dari setiap alternatif dan bobot kriteria. Nilai eigen yang paling besar akan menduduki peringkat pertama.

1. Analisis pemeringkatan

Untuk menguji hasil pemeringkatan tersebut, maka dilakukan kembali wawancara kepada para pakar untuk mendapatkan opini dan justifikasi mengenai hasil pemeringkatan yang telah didapatkan dengan menggunakan AHP.

1. Membuat kesimpulan dan saran

Dari hasil pemeringkatan fakto-faktor penyebab kegagalan aplikasi yang telah dijustifikasi oleh para pakar pada tahap sebelumnya dapat diketahui faktor mana yang paling kritikal. Setelah mengetahui hasil pemeringkatan faktor tersebut, maka dirumuskan kesimpulan untuk menjawab RQ dan saran untuk pengembangan penelitian selanjutnya.

## Instrumen Penelitian

Sumber data dari penelitian ini adalah data primer dan data sekunder. Data primer diperoleh dari perusahaan yang dipilih sebagai objek penelitian melalui wawancara dan kuisioner kepada para pakar di Departemen IT Service Management PT Telkomsel. Data sekunder didapat dari berbagai jurnal internasional, dan buku referesi sebagai pendukung dari data primer dalam penelitian ini.

Terdapat dua metode pengumpulan data primer dalam penelitian ini, yaitu kuisioner dan wawancara.

## Kuisioner

Penyebaran kuisioner dilakukan untuk mendapatkan skala prioritas (*pairwise comparison*) antar elemen sesuai dengan skala penilaian yang

direkomendasikan oleh Saaty. Responden dari kuisioner ini adalah para pakar di Departemen IT Service Management PT. Telkomsel dengan kriteria sebagai berikut :

* + 1. Memiliki pengalaman bekerja di bidang IT Service Management selama minimal 5 tahun di PT Telkomsel
    2. Memiliki latar belakang pendidikan di bidang Teknologi Informasi

Format kuisioner disusun berdasarkan kriteria dan alternatif yang ada pada struktur hirarki AHP yang dapat mengakomodasi *pairwise comparison* dengan skala penilaian yang direkomendasikan oleh Saaty. Kuisioner terbagi menjadi 2 bagian, bagian 1 merupakan isian data responden, dan bagian 2 merupakan perbandingan berpasangan. Perbandingan akan dilakukan pada antar elemen kriteria, antar elemen alternatif, dan antara elemen kriteria dan alternatif. Berikut merupakan contoh format kuisioner :

BAGIAN 1 : PROFIL RESPONDEN

* + Nama
  + Jenis Kelamin
  + Usia
  + Latar Belakang Pendidikan
  + Jabatan
  + Pengalaman (tahun)

BAGIAN 2 : PERBANDINGAN BERPASANGAN

* + Pengenalan dan penjelasan umum mengenai kuisioner
  + Penjelasan kriteria kualitas aplikasi yang diteliti
  + Penjelasan alternatif faktor-faktor penyebab kegagalan aplikasi
  + Penjelasan skala penilaian

## Tabel 3.2 Tabel skala penilaian Saaty

|  |  |
| --- | --- |
| **Nilai** | **Keterangan** |
| 1 | A dan B memiliki tingkat kepentingan yang sama |
| 3 | A sedikit lebih penting dari B |
| 5 | A jelas lebih penting dari B |
| 7 | A sangat jelas penting mendominasi B |
| 9 | A mutlak lebih penting dari B |
| 2,4,6,8 | Ragu-ragu |

* + Perbandingan berpasangan antara faktor dan kriteria

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Faktor | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Kriteria |
| A |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | B |

* + Perbandingan berpasangan antar faktor

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Faktor | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Faktor |
| A |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | B |

* + Perbandingan berpasangan antar kriteria

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Kriteria | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Kriteria |
| A |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | B |

## Wawancara

Wawancara pada penelitian ini terdiri dari dua tahap yaitu, wawancara untuk mengulas *theoretical framework*, terkait kriteria dan faktor-faktor alternatif yang telah dipilih dari proses studi lliteratur, dan wawancara untuk mendapatkan justifikasi pemeringkatan faktor-faktor tersebut.

1. Wawancara untuk mengulas kriteria dan alternatif kegagalan aplikasi Wawancara ini dilakukan pada 3 pakar di Departemen IT Service Management dengan kriteria sebagai berikut :
   * Memiliki pengalaman bekerja di Deaprtemen IT Service Management selama minimal 10 tahun
   * Memiliki latar belakang pendidikan di bidang Teknologi Informasi

Penulis memilih 3 pakar untuk meminimalisir ambiguitas bila terjadi perbedaan pendapat antar narasumber yang dapat terjadi bila jumlah narasumber berjumlah genap. Dengan jumlah narasumber ganjil maka dapat memudahkan pengambilan keputusan terhadap evaluasi kriteria dan alternatif dengan mengambil jumlah pendapat terbanyak.

Wawancara ini dilakukan untuk menguji variabel-variabel yang digunakan dalam *Theoretical Framework*, apakah perlu penambahan atau pengurangan terhadap variabel-variabel tersebut untuk menyesuaikan dengan kondisi perusahaan. Dari hasil wawancara tersebut, akan dibangun sebuah struktur hirarki AHP yang terdiri dari 3 level, yaitu tujuan, kriteria, dan alternatif. Pada penelitian ini, yang akan menjadi tujuan adalah kegagalan aplikasi, kemudian yang menjadi kriteria adalah kriteria *software quality factor*. Sementara itu, yang akan menjadi alternatif adalah faktor-faktor penyebab kegagalan / insiden pada aplikasi.

Berikut merupakan pointer pertanyaan wawancara dalam penelitian ini :

## Tabel 3.3 Indikator Pengukuran Variabel Penelitian

|  |  |  |
| --- | --- | --- |
| **Variabel Kriteria** | **Sumber** | **Indikator Pertanyaan** |
| *Extendibility* | (Hamid & Hasan, 2011) | Pengaruh ketidakmampuan aplikasi untuk  dikembangkan kapabilitas dan performancenya terhadap kegagalan aplikasi |
| *Flexibility* | Pengaruh tingkat kesulitan dalam mengubah tujuan, fungsi, atau data aplikasi untuk memenuhi kebutuhan perubahan terhadap  kegagalan aplikasi |
| *Functionality* | Pengaruh ketidakmampuan aplikasi untuk menyediakan fungsi yang sesuai dengan kebutuhan yang tersurat maupun tersirat ketika aplikasi berada pada kondisi tertentu  terhadap kegagalan aplikasi |
| *Usability* | Pengaruh kesulitan dalam mempelajari dan mengoperasikan aplikasi terhadap kegagalan  aplikasi |
| *Testability* | Pengaruh kesulitan dalam pengujian program untuk memastikan bahwa aplikasi melakukan fungsi spesifiknya terhadap kegagalan aplikasi |

**Tabel 3.3 Indikator Pengukuran Variabel Penelitian (sambungan)**

|  |  |  |
| --- | --- | --- |
| **Variabel Kriteria** | **Sumber** | **Indikator Pertanyaan** |
| Kriteria lain | (Saaty T. L.,  2012) | Pendapat dari pakar mengenai kriteria  kualitas aplikasi |
| *Software error* | (Dalal & Chhillar, 2012)  (Pertet & Narasimhan,  2005) | 1. Pengkategorian penyebab *software error* 2. Pengaruh aplikasi failure terhadap kegagalan aplikasi di instansi |
| *Human/operator error* | (Saaralainen & Jantti, 2015)  (Dalal & Chhillar, 2012)  (Pertet & Narasimhan, 2005) | 1. Pengategorian penyebab human/operator error 2. Pengaruh human/operator error terhadap kegagalan aplikasi di instansi |
| *Hardware and environtmental failure* | (Dalal & Chhillar, 2012)  (Pertet & Narasimhan, 2005) | 1. Pengkategorian penyebab *Hardware and environmental failure* 2. Pengaruh *HW/Environmental* failure terhadap kegagalan aplikasi pada instansi |
| *Security Violations* | (Pertet & Narasimhan, 2005) | 1. Pengkategorian penyebab *security violations* 2. Pengaruh *security violations* pada kegagalan aplikasi di instansi |
| *Fault Recovery* | (Dalal & Chhillar, 2012)  (Pertet & Narasimhan, 2005) | 1. Pengkategorian penyebab *fault recovery* 2. Pengaruh *Fault Recovery* pada kegagalan aplikasi di instansi |
| *Communication*  *Management* | (Dalal dan  Chhillar, 2012) | Pengaruh *Communication Management* pada  kegagalan aplikasi di instansi |
| *Change Management* | (Saaralainen & Jantti, 2015) | 1. Pengkategorian penyebab *change management* 2. Pengaruh *change management* pada kegagalan aplikasi di instansi |

**Tabel 3.3 Indikator Pengukuran Variabel Penelitian (sambungan)**

|  |  |  |
| --- | --- | --- |
| **Variabel Kriteria** | **Sumber** | **Indikator Pertanyaan** |
| *Institusional Pressure* | (Dalal dan Chhillar, 2012) | 1. Pengkategorian penyebab   *institusional pressure*   1. Pengaruh *institusional pressure* pada kegagalan aplikasi di instansi |
| Faktor Lain | (Saaty, 2012) | 1. Faktor kegagalan aplikasi 2. Subfaktor kegagalan aplikasi |

1. Wawancara untuk mendapatkan justifikasi pemeringkatan faktor penyebab kegagalan aplikasi

Wawancara ini dilakukan pada 2 pakar di bidang terkait faktor yang menyebabkan kegagalan aplikasi di PT Telkomsel untuk mengkonfirmasi pemeringkatan faktor yang telah didapatkan dari proses olah data menggunakan AHP.. Berikut merupakan pointer pertanyaan wawancara dalam penelitian ini :

## Tabel 3.4 Indikator Pengukuran Variabel Pemeringkatan Faktor

|  |  |  |
| --- | --- | --- |
| **Variabel** | **Sumber** | **Indikator Pertanyaan** |
| Pemeringkatan  faktor | (Saaty, 2012) | Kesesusaian hasil pemeringkatan  dengan kondisi instansi penelitian |

## Metode dan Proses Olah Data

Dalam penelitian ini, peneliti akan meneliti tentang faktor kegagalan aplikasi yang paling berpotensi menjadi insiden di salah satu perusahaan telekomunikasi di Indonesia, yaitu PT Telkomsel dengan menggunakan model *software quality factor* Hamid dan Hasan (2011) yang merupakan model khusus untuk kriteria aplikasi di industri telekomunikasi. Karena dalam penelitian ini faktor-faktor (alternatif) tersebut akan diperingkatkan prioritasnya berdasarkan kriteria *quality factor* dimana kriteria independen terhadap alternatif, dan dalam pemeringkatannya tidak hanya membutuhkan perhitungan dari data *questionnaire* namun juga subjektivitas dari pihak instansi terkait, serta dari penelitian sebelumnya terkait evaluasi kualitas aplikasi, maka penelitian ini akan menggunakan metode AHP sebagai metode olah data.

Metode AHP dikembangkan oleh Dr. Thomas Saaty untuk menyelesaikan permasalahan baik teknikal maupun manajerial yang membutuhkan pengambilan keputusan (Pinto, 2010). AHP melibatkan perilaku dan proses pemikiran manusia dalam pengambilan keputusannya dengan cara menyederhanakan permasalahan tersebut dan mengubah situasi yang tidak terstruktur menjadi bagian-bagian yang lebih kecil serta tersusun dalam urutan tertentu, memberikan nilai subjektif terhadap variabel, dan kemudian menyatukan penilaian tersebut untuk menentukan variabel mana yang memiliki prioritas tertinggi (Saaty, 2012)

Data yang telah dikumpulkan dari hasil kuisioner akan diolah menggunakan tools Microsoft Excell 2010. Untuk melakukan analisis perhitungan matriks AHP, penulis menggunakan aplikasi Expert Choice 11. Berikut merupakan proses olah data dengan metode AHP menurut Saaty (2012) :

1. Membangun struktur hierarki

Tahapan ini merupakan tawahapan awal dari AHP, yaitu memecah realita yang ada menjadi beberapa bagian, kemudian bagian tersebut dipecah lagi menjadi kelompok yang lebih kecil lagi. Dari hasil wawancara untuk mengulas *theoretical framework*, akan dibangun sebuah struktur hirarki AHP yang terdiri dari 3 level, yaitu tujuan, kriteria, dan alternatif. Pada penelitian ini, yang akan menjadi tujuan adalah kegagalan aplikasi, kemudian yang menjadi kriteria adalah kriteria *software quality factor*. Sementara itu, yang akan menjadi alternatif adalah faktor-faktor penyebab kegagalan / insiden pada aplikasi.

1. Menentukan skala prioritas

Kuisioner yang telah disusun dengan esurveycreator.com akan mengakomodasi perbandingan berpasangan *(pairwise comparison)* dengan skala penilaian yang direkomendasikan oleh Saaty ( dapat dilihat pada sub bab 2.3.3). Perbandingan tersebut akan ditransformasikan dalam bentuk matriks n x n untuk melakukan analisis numerik. Perbandingan akan dilakukan pada antar elemen kriteria, antar elemen alternatif, dan antara elemen kriteria dan alternatif.

1. Melakukan perhitungan matriks

Perhitungan matriks dilakukan untuk mendapatkan nilai dari setiap kriteria dan alternatif. Perhitungan ini berdasarkan prinsip *eigenvector.* Hasil dari penilaian seluruh responden disusun menjadi matriks n x n yang berisikan rata-rata geometris untuk setiap pasangan penilaian. Nilai tersebut dibulatkan ke nilai Saaty terdekat dan akan diproses menggunakan aplikasi Expert Choice 11 yang akan menghasilkan nilai eigen dari setiap kriteria dan alternatif.

1. Menghitung nilai rasio konsistensi (CR)

Pengukuran konsistensi ini diperlukan untuk mengetahui konsistensi jawaban yang berpengaruh terhadap kebenaran dari yang didapat. Nilai eigen setiap kriteria dan alternatif dihitung nilai rasio konsistensinya. Nilai ini akan dibandingkan dengan Nilai rasio CR, yaitu harus sama atau lebih kecil dari 10% (Saaty, 2012). Jika lebih dari 10% maka terdapat kesalahan pada penilaian tersebut dan perlu diperbaiki dengan kembali pada tahap penyebaran kuisioner. Adapun rumus untuk uji konsistensi dapat dilihat pada sub bab 2.3.3.

## BAB 4 ANALISIS

Bab ini menjelaskan proses pengolahan data primer pada penelitian ini. Tahap pertama yang dilakukan adalah menguji variabel-variabel yang digunakan dalam *Theoretical Framework* yang telah dibangun pada para pakar di Departemen IT Service Managemenr di perusahaan telekomunikasi untuk mengetahui apakah perlu penambahan atau pengurangan terhadap variabel-variabel tersebut untuk kemudian dianalisis dengan AHP sampai mendapatkan peringkat yang kemudian akan dievaluasi lagi pada para pakar.

* 1. **Analisis Variabel dalam *Theoretical Framework***

Struktur hirarki AHP didapatkan dari pengujian variabel-variabel yang digunakan dalam *Theoretical Framework* yang telah dibangun pada subbab 2.5. Evaluasi variabel ini dilakukan dengan mewawancarai 3 pakar Departemen IT Service Management di perusahaan telekomunikasi untuk mengetahui apakah perlu penambahan atau pengurangan terhadap variabel-variabel tersebut agar sesuai dengan kondisi perusahaan. Hal yang ditekankan pada evaluasi ini adalah relevansi setiap variabel terhadap penelitian yang dilakukan, penambahan variabel yang dianggap relevan, serta pengurangan variabel yang tidak relevan dengan penelitian. Adapun variabel-variabel yang dievaluasi oleh para pakar adalah :

1. 5 Kriteria Kegagalan Aplikasi, yaitu: *unextendibility, inflexibility, unfunctionality, unusability,* dan *untestability*
2. 8 Kategori Alternatif penyebab kegagalan aplikasi, yaitu: *software error, human/operator error, hardware and environmental error, fault recovery error, security violations, change management issue, communication management issue, institusional pressure* dan pengkategorian 32 alternatif dari masing-masing kategori alternatif penyebab kegagalan aplikasi, yaitu: *system overload, computational /logic error, resource exhaustion, failed software upgrade, fault recovery routine, configuration error, procedural error, miscellaneous accidents, organizational process, inadequate supervision, planned inappropriate operations, device driver failure, i/o error, memory*

*parity error, network hardware failure, power outages, overheating, high humidity, natural disaster, improper fault assignment*, *complex fault recovery, password disclosures, denial of service (dos) attacks, worms and viruses, browser vulnerabilities, authentication failures, insider threat, improper planning of change, improper testing of change, unrealisticcaly low bids, unrealistically low budget requests, underestimates of time requirements.*

Agar struktur AHP yang dihasilkan sesuai dengan kondisi perusahaan, maka telah didefinisikan kriteria pemilihan pakar pada subbab 3.3. Berdasarkan kriteria tersebut, maka dipilih tiga pakar, sebagai berikut:

1. Pakar 1

Nama : Sjuhud Al Amien

Jabatan : Manager IT System Integration Testing Management Pengalaman : 19 Tahun

Pendidikan : S1 Teknik Informatika

1. Pakar 2

Nama : Irawan Koesmardianto

Jabatan : General Manager IT Service Quality Management Pengalaman : 13 Tahun

Pendidikan : S1 Manajemen Informatika

1. Pakar 3

Jabatan : Division Head of IT Service QA & Compliance dan IT Service Management

Pengalaman : 22 Tahun

Pendidikan : D3 Elektronika (Politeknik Institut Teknologi Bandung)

## Analisis Kriteria Kegagalan Aplikasi

Kegagalan aplikasi disebabkan oleh aplikasi tersebut belum memenuhi definisi aplikasi berkualitas. Maka pada penelitian ini, terdapat 5 kriteria aplikasi yang tidak berkualitas khusus di bidang telekomunikasi yang akan dianalisis, antara lain *unextendibility, inflexibility, unfunctionality, unusability*, dan un*testability*. Pada transkrip wawancara pada Lampiran 3, Lampiran 4, dan Lampiran 5 yang

telah dilakukan dengan 3 pakar Deaprtemen IT Service Management, diberikan kode untuk menandakan kriteria mana yang tetap ada, dihapus, ditambah, atau diubah. Berikut merupakan rangkuman evaluasi kriteria aplikasi tidak berkualitas untuk penyusunan struktur hirarki AHP:

**Tabel 4.1 Tabel rangkuman evaluasi kriteria *software quality***

|  |  |  |  |
| --- | --- | --- | --- |
| **Kriteria** | **Pakar 1** | **Pakar 2** | **Pakar 3** |
| *UnReliability* |  | *Add* 5 | *Add*12 |
| *UnIntegrity* |  | *Add* 5 | *Add*13 |
| *UnResiliancy* |  |  | *Add*12 |

Dari tabel 4.1 terlihat bahwa pakar pertama tidak memberikan masukan terkait kriteria, karena kriteria-kriteria yang disajikan sudah dianggap sesuai untuk kualitas aplikasi di perusahaan telekomunikasi. Pakar kedua menambahkan dua kriteria, yaitu *unreliability* dan *unintegrity.* Sedangkan Pakar ketiga menambahkan tiga kriteria, yaitu *unreliability*, *unintegrity*, dan *unresiliency*. Menurut pakar ketiga, kriteria *unintegrity* khusus ke *data integrity*. Untuk kriteria lainnya, yaitu *unextendibility, inflexibility, unfunctionality, unusability*, dan *untestability*, pakar kedua dan ketiga sepakat bahwa kriteria-kriteria tersebut sudah sesuai untuk aplikasi di perusahaan telekomunikasi. Berikut merupakan hasil dari analisis kriteria kualitas yang menyebabkan kegagalan aplikasi berdasarkan pendapat para pakar:

1. Kriteria *unextendibility, inflexibility, unfunctionality, unusability*, dan *untestability* sudah dianggap sesuai dengan aplikasi di perusahaan telekomunikasi
2. Penambahan kriteria *unreliability* akan diakomodasi, karena kriteria tersebut diusulkan oleh dua pakar, yaitu pakar kedua dan pakar ketiga
3. Penambahan kriteria *unintegrity* akan diakomodasi, karena terdapat dua pakar yang mengusulkan kriteria ini, yaitu pakar kedua dan pakar ketiga. Meskipun pakar ketiga mengkhusukan kriteria tersebut ke *data integrity*, namun yang akan digunakan adalah terminologi *unintegrity* secara umum, tidak khusus ke *data integrity*.
4. Penambahan kriteria *unresiliency* yang diusulkan oleh pakar ketiga tidak akan diakomodasi karena hanya satu pakar yang mengusulkan kriteria tersebut.

## Analisis Faktor Penyebab Kegagalan Aplikasi

Dari wawancara yang telah dilakukan pada tiga pakar, terdapat beberapa penambahan faktor yang dilakukan para pakar pada beberapa kategori faktor. Hal tersebut dapat diketahui dengan melakukan hal yang sama dengan analisis kriteria kualitas aplikasi, yaitu dengan teknik kodifikasi pada transkrip wawancara untuk menandakan faktor-faktor mana yang tetap ada, dihapus, ditambah, atau diubah. Berikut merupakan rangkuman evaluasi faktor-faktor penyebab kegagalan pada aplikasi beserta untuk penyusunan struktur hirarki AHP:

## Tabel 4.2 Tabel Rangkuman Evaluasi Faktor-Faktor Penyebab Kegagalan Aplikasi

|  |  |  |  |
| --- | --- | --- | --- |
| **Alternatif** | **Pakar 1** | **Pakar 2** | **Pakar 3** |
| *Security*  *Violations* |  |  | subfaktor *fraud*11 |
| *Change Management Issue* | subfaktor *request changes*3*;*  subfaktor *monitoring*2; subfaktor  *infrastructure changes*4 | subfaktor *deployment changes*6 | detail subfaktor (contohnya : *unproper planning*)8 |
| *Communication Management Issue* |  | subfaktor :  i. Internal (sesama IT)  j. Eksternal *(third party)*7 | subfaktor : *Internal communication*9 |
| *Institusional*  *Pressure* | *emergency changes*1 |  | subfaktor *time to*  *market*14 |
| Faktor Lain |  |  | *vendor*  *Management*10 |

Dari tabel 4.2 dapat dilihat bahwa pakar pertama memberi masukan pada satu kategori faktor, yaitu *change management*. Menurut pakar pertama, faktor dari *change management* harus ditambahkan *request changes, monitoring changes*, dan *infrastructure changes*. Sementara pakar kedua memberi masukan pada dua kategori faktor yaitu, *change management* dan *communication management*. Untuk kategori faktor *change management*, pakar kedua memberi masukan bahwa faktor dari *change management* ditambahkan *deployement changes*, sedangkan

untuk kategori faktor *communication management* pakar kedua memberikan dua tambahan faktor, yaitu *internal communication* dan *external communication*.

Pakar ketiga memberikan masukan terhadap empat kategori faktor, antara lain *security violations, change management, communication management*, dan *institusional pressure.* Untuk kategori faktor *security violations*, pakar ketiga menambahkan satu faktor yaitu *fraud.* Sedangkan masukan untuk kategori faktor *change management*, pakar ketiga lebih menyarankan agar faktor dari *change management* lebih didetailkan, contohnya *planning the change* menjadi *improper planning of changes*. Berbeda dengan pakar kedua yang membagi kategori faktor *communication management* menjadi 2 faktor, yaitu *internal communication* dan *external communication*, pakar ketiga beranggapan bahwa kategori faktor *communication management* lebih ke internal, sementara untuk komunikasi pada pihak ketiga menjadi kategori faktor tambahan baru, yaitu *vendor management* termasuk di sana kualifikasi vendor yang akan dipakai oleh perusahaan. Sementara untuk kategori faktor *institusional pressure*, pakar ketiga menambahkan faktor *time to market.* Pendapat pakar ketiga ini didukung oleh pernyataan dari pakar pertama terkait *emergency changes* di mana terdapat suatu aplikasi yang di-*release* karena tuntutan pasar atau kompetitor sehingga mengabaikan beberapa tahapan dan dokumentasi dalam pengujian yang menyebabkan insiden. Faktor-faktor lainnya dan pengkategorian faktor yang tidak ditampilkan di dalam tabel dianggap sudah tepat oleh ketiga pakar.

Berikut merupakan hasil dari analisis faktor penyebab kegagalan pada aplikasi berdasarkan pendapat para pakar:

1. Penambahan faktor *fraud* pada kategori faktor *security violations* oleh pakar ketiga tidak diakomodasi karena hanya satu pakar yang memberi masukan tersebut
2. Penambahan faktor *request changes, monitoring changes,* dan *infrastructure changes* pada kategori faktor *change management* oleh pakar pertama tidak akan diakomodasi karena hanya satu pakar yang mengusulkan subfaktor tersebut
3. Penambahan faktor *deployment changes* pada kategori faktor *change management* tidak akan diakomodasi karena hanya pakar kedua yang mengusulkan faktor tersebut
4. Masukan pakar ketiga terkait penamaan faktor pada *change management* akan diakomodasi agar faktor penyebab kegagalan aplikasi menjadi lebih jelas
5. Penambahan faktor *internal communication* dan *external communication* dari pakar kedua akan diakomodasi karena pakar ketiga juga berpendapat bahwa komunikasi di dalam sebuah perusahaan memang terdiri dari dua kategori, yaitu internal dan eksternal. Hanya saja pakar ketiga memisahkan faktor *external communication* menjadi sebuah kategori faktor baru, yaitu *vendor management*. Namun, karena untuk mendukung *pairwise comparison* yang mangharuskan minimal dua faktor yang dapat dibandingkan maka faktor *vendor management* tersebut dianggap sebagai faktor *external communication.*
6. Penambahan faktor *time to market* pada kategori faktor *institusional pressure* akan diakomodasi karena terdapat pernyataan dari pakar pertama yang mendukung subfaktor ini
7. Penambahan kategori faktor *vendor management* oleh pakar ketiga tidak akan diakomodasi karena hanya pakar ketiga yang menambahkan faktor tersebut

## Analisis dan Perancangan Struktur Hirarki AHP

Setelah dilakukan analisis terhadap kriteria, kategori faktor, serta faktor penyebab kegagalan aplikasi yang telah dilakukan dengan cara kodifikasi transkrip wawancara pada Lampiran 3, Lampiran 4, dan Lampiran 5 dengan 3 pakar di Departemen IT Service Management, maka berdasarkan analisis tersebut, kategori faktor dan faktor penyebab kegagalan aplikasi di perusahaan telekomunikasi yang akan digunakan pada penelitian ini dapat dilihat pada Tabel 4.3. Fakor yang akan diteliti lebih lanjut menjadi 35 faktor dengan 8 kategori. Kategori faktor *software error* memiliki 5 faktor, kategori faktor *human/operator error* memiliki 6 faktor, kategori faktor *hardware/environmental error* memiliki 8 faktor, kategori faktor *fault recovery* memiliki 2 faktor, kategori faktor *security violations* memiliki 6 faktor, kategori faktor *change management* memiliki 2 faktor, kategori faktor

*communication management* memiliki 2 fak tor, kategori faktor *Institusional Pressure*

## Tabel 4.3 Hasil Analisis Faktor Penyebab Kegagalan Aplikasi

|  |  |  |
| --- | --- | --- |
| **Kategori Faktor** | **Subfaktor** | **Definisi** |
| *Software Error* | *System overload* | Ketidakmampuan untuk mengatasi pesan dan ekspektasi dari sejumlah sumber  dalam batas waktu tertentu |
| *Computational/Logic error* | Bug dalam program yang menyebabkan aplikasi beroperasi secara tidak benar,  tetapi tidak menyebabkan crash |
| *Resource exhaustion* | Keamanan komputer mengeksploitasi crash, hang, atau menganggu program  atau sistem yang menjadi target |
| *Failed software*  *upgrade* | Kegagalan yang terjadi saat mengupgrade  aplikasi |
| *Fault recovery routine* | Mekanisme dan teknik yang dilakukan untuk mengembalikan level operasi yang diinginkan dari sebuah komponen atau sistem setelah terjadi kegagalan |
| *Human/Operato r Error* | *Configuration error* | Kegagalan yang terjadi karena kesalahan  dalam pengkonfigurasian aplikasi |
| *Procedural error* | Kegagalan yang terjadi karena prosedur  pengoperasian aplikasi tidak diikuti oleh operator |
| *Miscellaneous*  *accidents* | Berbagai kesalahan yang tidak disengaja |
| *Organizational process* | Penentuan proses dan tugas-tugas apa yang harus dilakukan untuk mencapai tujuan dan pengambilan keputusan tidak dilaksanakan dengan baik saat  pengembangan / pengoperasian aplikasi |
| *Inadequate*  *supervision* | Pengawasan yang tidak memadai saat  pengembangan / pengoperasian aplikasi |
| *Planned inappropriate operations* | Merencanakan langkah operasi aplikasi dengan tidak tepat |

**Tabel 4.3 Hasil Analisis Faktor Penyebab Kegagalan Aplikasi (sambungan)**

|  |  |  |
| --- | --- | --- |
| **Kategori Faktor** | **Subfaktor** | **Definisi** |
| *Hardware and Environmental Error* | *Device driver failure* | Kegagalan yang disebabkan oleh device driver mengalami error. Salah satu sebabnya adalah serangan *malware*. Kegagalan suatu device driver biasa  ditandai dengan adanya pesan error |
| *I/O error* | Komputer tidak dapat melakukan tindakan input/output ketika mencoba  mengakses drive atau hard disk |
| *Memory parity error* | Satu atau lebih data yang disimpan membawa nilai yang berbeda ketika dipanggil kembali dengan nilai saat data tersebut disimpan. Parity error merupakan  jenis kerusakan data. |
| *Network hardware failure* | Kegagalan seluruh atau sebagian dari komponen atau komponen jaringan  karena sebuah kerusakan |
| *Power outages* | Hilangnya daya listrik dalam jangka  waktu yang pendek atau panjang |
| *Overheating* | Suhu hardware melebihi Batas normal sehingga menghasilkan panas yang berlebihan |
| *Hardware and Environmental*  *Error* | *High Humidity* | Hardware berada di lingkungan yang  tingkat kelembabannya tinggi |
| *Natural disaster* | Bencana alam |
| *Fault Recovery* | *Improper fault*  *assignment* | Proses menanggapi munculnya kesalahan  tidak diserahkan kepada pihak yang tepat |
| *Complex fault recovery* | Prosedur penanganan fault susah dipahami |

**Tabel 4.3 Hasil Analisis Faktor Penyebab Kegagalan Aplikasi (sambungan)**

|  |  |  |
| --- | --- | --- |
| **Kategori Faktor** | **Subfaktor** | **Definisi** |
| *Security Violations* | *Password disclosures* | Pengungkapan password |
| *Denial of services (DOS) attacks* | Serangan terhadap sebuah komputer atau server di dalam jaringan internet dengan cara menghabiskan resource yang  dimiliki oleh komputer tersebut |
| *Worms and viruses* | Worms : Sebuah program komputer yang dapat menggandakan dirinya secara  sendiri dalam sistem komputer. |
| *Browser vulnerabilities* | Kerentanan dalam sistem operasi atau software dengan maksud untuk melanggar keamanan browser untuk  mengubah pengaturan browser pengguna tanpa sepengetahuan mereka |
| *Authentication*  *failures* | Kegagalan otentikasi |
| *Insider threat* | Adanya orang yang memiliki informasi orang dalam mengenai praktek keamanan  organisasi, data dan sistem komputer. |
| *Change Management* | *Improper planning of*  *change* | Perubahan dari sebuah aplikasi tidak  direncanakan dengan baik |
| *Improper testing of*  *change* | Perubahan dari sebuah aplikasi tidak diuji  dengan baik |
| *Communication Management* | *Improper internal*  *communication* | Komunikasi internal IT tidak dilakukan  dengan baik |
| *Improper external communication* | Komunikasi ke pihak ketiga tidak dilakukan dengan baik |
| *Institusional Pressure* | *Unrealistically low*  *bids* | Tawaran yang terlalu rendah untuk  sebuah pengembangan aplikasi |
| *Unrealistically low*  *budget requests* | Permintaan biaya yang terlalu rendah  untuk sebuah pengembangan aplikasi |
| *Underestimates of time requests* | Tidak memperhitungkan secara tepat lama waktu pengembangan ssebuah  aplikasi |
| *Time to market* | Desakan pasar/kompetitor agar aplikasi  segera di-release |

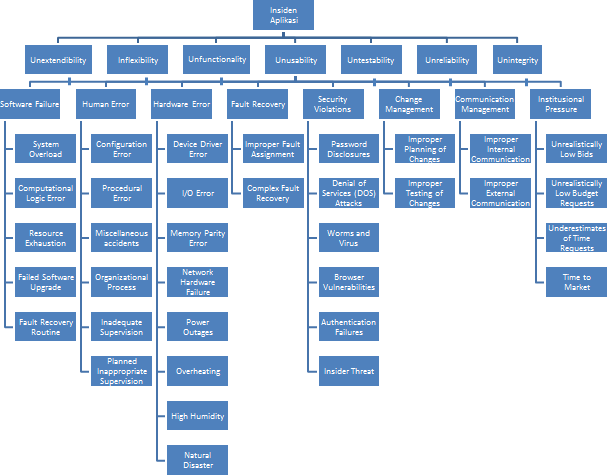
Adapun kriteria yang akan diteliti lebih lanjut berdasarkan hasil evaluasi dengan wawancara pakar pada subbab 4.1.1 menjadi 7 kriteria, hasil evaluasi tersebut dapat dilihat pada tabel 4.4.

## Tabel 4.4 Hasil Analisis Kriteria Kegagalan Aplikasi

|  |  |  |
| --- | --- | --- |
| **Kriteria Aplikasi Berkualitas** | **Kriteria Aplikasi Tidak Berkualitas** | **Definisi** |
| *Extendibility* | ***Unextendability*** | Ketidakmampuan aplikasi untuk dapat dikembangkan kapabilitas dan performancenya dengan mengembangkan fungsi saat ini atau  dengan menambahkan fungsionalitas baru |
| *Flexibility* | ***Inflexibility*** | Ketidakmudahan dalam mengubah tujuan, fungsi, atau data aplikasi untuk  memenuhi kebutuhan perubahan |
| *Functionality* | ***Unfunctionality*** | Ketidakmampuan aplikasi untuk menyediakan fungsi yang sesuai dengan kebutuhan yang tersurat maupun tersirat ketika aplikasi berada pada kondisi  tertentu |
| *Usability* | ***Unusability*** | Ketidakmudahan belajar dan operasi  aplikasi |
| *Testability* | ***Untestability*** | Ketidakmudahan dalam menguji  program untuk memastikan bahwa aplikasi melakukan fungsi spesifiknya |
| *Reliability* | ***Unreliability*** | Ketidakmampuan sebuah aplikasi untuk bebas dari kesalahan/kegagalan dalam waktu yang ditentukan dalam  lingkungan tertentu |
| *Integrity* | ***Unintegrity*** | Ketidakmampuan sebuah aplikasi menjaga data tidak diubah oleh pihak  yang tidak berwenang |

Struktur hierarki AHP terdiri 3 level, yaitu tujuan, kriteria, dan alternative. Dari paparan di atas, level 1 merupakan tujuan yang ingin dicapai yaitu kegagalan aplikasi. Level 2 yaitu kriteria kualitas aplikasi yang dapat menyebabkan kegagalan aplikasi yang terdiri dari 7 kriteria. Kriteria-kriteria ini diambil dari definisi software yang berkualitas (Tabel 4.4). Untuk level 3 yaitu alternatif digambarkan dalam dua tingkat, yang terdiri dari 8 kategori faktor dan 35 subfaktor.

Berikut merupakan struktur hierarki AHP yang dihasilkan dari hasil analisis yang sudah dilakukan pada subbab 4.1.1 dan 4.1.2:



## Gambar 4.1 Struktur AHP penelitian

Struktur AHP tersebut kemudian akan dipetakan dan diolah dengan *software* Expert Choice 11. Expert Choice merupakan aplikasi pengambilan keputusan yang mendukung *collaborative decision* terkait *Analytical Hierarchy Process* (AHP) yang dikembangkan oleh Thomas L. Saaty. Setelah struktur tersebut dipetakan ke dalam Expert Choice, maka akan dilakukan perhitungan *pairwise comparison* yang juga akan dilakukan dengan aplikasi ini. Struktur AHP yang telah disusun di aplikasi Expert Choice 11 dapat dilihat pada Lampiran 6.

## Analisis Kuisioner Perbandingan Berpasangan

Sub bab ini menjelaskan tentang analisis perbandingan berpasangan yang dilakukan dengan teknik AHP. Format kuisioner disusun berdasarkan kriteria dan alternatif yang ada pada struktur hirarki AHP yang dapat mengakomodasi perbandingan berpasangan dengan skala penilaian yang direkomendasikan oleh

Saaty. Perbandingan akan dilakukan pada antar elemen kriteria, antar elemen kategori alternatif terkait kriteria, dan antara setiap alternatif untuk setiap kategori alternatif.

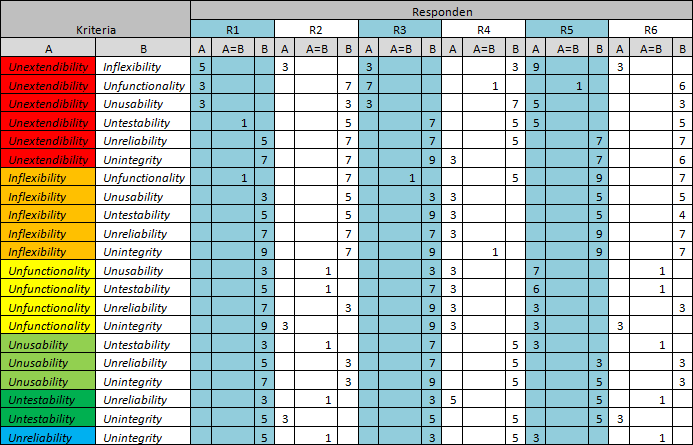
Kuisioner yang telah disusun diberikan pada 6 responden yang telah memenuhi kriteria pakar untuk mengisi kuisioner serta memegang peran penting di Departemen IT Service Management PT Telkomsel (Level General Manager dan Manager). Hasil dari kuisioner ini akan dipetakan menjadi matriks perbandingan berpasangan yang akan dianalisis nilai eigennya menggunakan software Expert Choice 11 untuk menentukan prioritas dari masing-masing level pada AHP.

Sebelum diberikan pada responden, penulis memberikan kuisioner pada 2 orang yang bukan merupakan calon responden untuk melakukan uji keterbacaan. Dari hasil uji keterbacaan tersebut, didapatkan lama waktu pengisian kuisioner, yaitu sekitar 2 jam bila diisi sendiri, dan 1 jam 30 menit bila didampingi serta kesulitan- kesulitan yang dihadapi saat melakukan pengisian kuisioner seperti bagaimana cara menentukan skala prioritas dan pengertian dari masing-masing terminologi yang digunakan dalam kuisioner. Dari masukan-masukan tersebut, penulis memperbaiki metode pengisian dan beberapa elemen kuisioner yang dianggap kurang jelas.

## Analisis Hasil Kuisioner

Setelah didapatkan hasil kuisioner dari 6 responden dari Departemen IT Service Management PT Telkomsel, kuisioner tersebut dianalisis dengan cara membuat tabel perbandingan hasil kuisioner tiap responden yang berisikan skala prioritas setiap kriteria, kategori alternatif, dan alternatif. Berikut merupakan hasil skala prioritas dari kuisioner untuk kriteria aplikasi yang dapat menyebabkan kegagalan aplikasi:

## Tabel 4.5 Perbandingan Skala Prioritas untuk Kriteria



Dari tabel di atas, kemudian dianalisis rata-rata geometrisnya untuk mendapatkan skala prioritas yang akan dijadikan matriks perbandingan berpasangan. Contoh perhitungan untuk mendapatkan rata-rata geometris untuk tingkat potensi antara K1 dan K2 (*Unextendibility* dan *Inflexibility*) dan K1 dan K3 (*Unextendibility* dan *Unfunctionality*) adalah sebagai berikut :

Rata-rata Geometris K1,K2 = 6√5 + 3 + 3 + 1/3 + 9 + 3

= 2.715491012

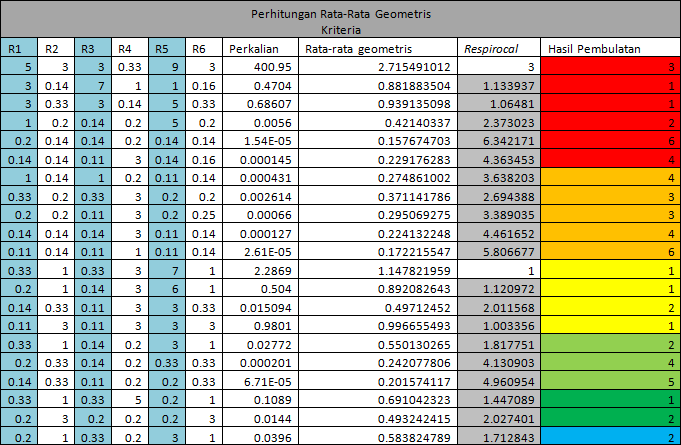
Rata-rata Geometris K2,K3 = 6√1 + 1/7 + 1 + 1/5 + 1/9 + 1/7

= 0.274861002

Nilai rata-rata geometris tersebut kemudian dibulatkan ke nilai terdekat. Jika nilai desimal dari rata-rata geometris tersebut lebih besar atau sama dengan 0.5 maka akan dibulatkan ke atas, contohnya nilai rata-rata geometris K1,K2 2.71 akan dibulatkan menjadi 3. Sedangkan untuk yang nilai desimalnya lebih kecil dari 0.5 maka akan dibulatkan ke bawah. Sementara untuk nilai rata-rata geometris yang lebih kecil dari 1 seperti nilai rata-rata geometris K2,K3 yaitu 0.27 akan dilakukan

perhitungan *reciprocal* yaitu = 1/0.27 = 3.63 dikarenakan skala penilaian Saaty hanya bernilai 1 sampai 9. Nilai tersebut kemudian akan dibulatkan ke nilai terdekat, karena nilai decimalnya lebih besar dari 0.5 maka akan dibulatkan ke atas menjadi 4. Maka nilai rata-rata geometrisnya menjadi ¼. Berikut merupakan hasil perhitungan rata-rata geometris untuk kriteria aplikasi yang berpotensi menyebabkan kegagalan pada aplikasi:

## Tabel 4.6 Perhitungan Nilai Rata-Rata Geometris - Kriteria



Untuk hasil kuisioner perbandingan skala prioritas dan perhitungan nilai rata-rata geometris kategori faktor untuk setiap kriteria serta perhitungan nilai rata-rata geometris untuk setiap faktor dapat dilihat pada Lampiran 8 dan Lampiran 9.

## Analisis Perhitungan Matriks Perbandingan Berpasangan

Setelah melakukan perhitungan rata-rata geometris untuk setiap pasang kriteria, setiap pasang kategori faktor, dan setiap pasang faktor, disusun matriks perbandingan berpasangan yang berisikan nilai rata-rata geometris. Nilai tersebut kemudian akan dimasukkan dan diolah dengan software Expert Choice 11.

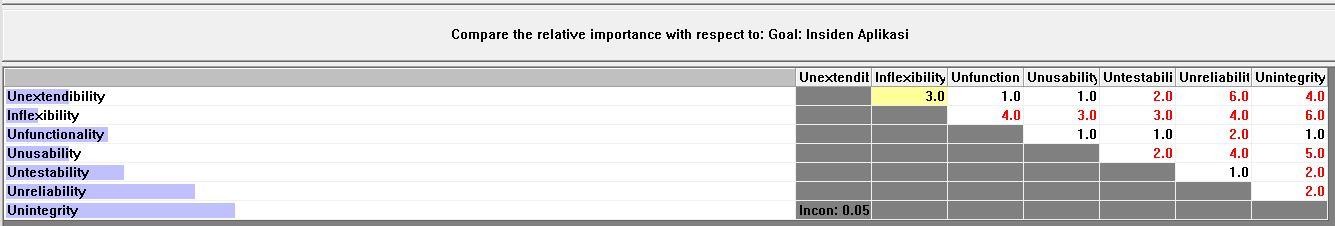
## Bobot Kriteria Aplikasi Penyebab Kegagalan Aplikasi

Nilai rata-rata geometris yang telah dibulatkan kemudian dimasukkan ke matriks perbandingan berpasangan. Bila nilai rata-rata geometris tersebut dibulatkan dari nilai *reciprocal*, maka di matriks perbandingan berpasangan nilainya menjadi 1/hasil pembulatan. Matriks perbandingan berpasangan untuk kriteria aplikasi penyebab kegagalan aplikasi dapat dilihat pada Tabel 4.7

## Tabel 4.7 Hasil Pembulatan Rata-Rata Geometris untuk Kriteria Aplikasi Penyebab Kegagalan Aplikasi

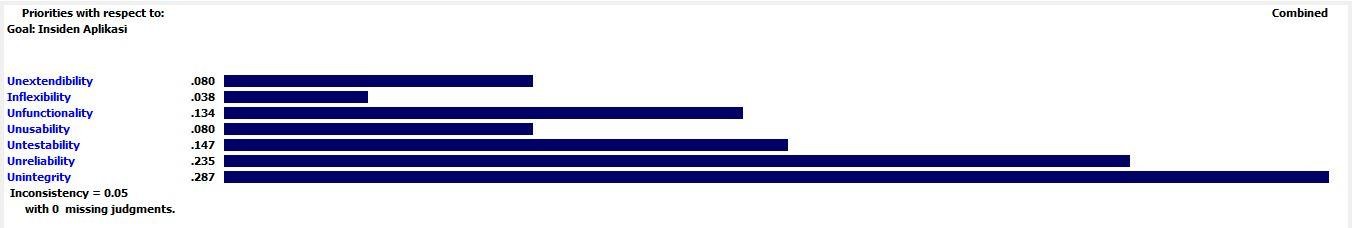
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | K1 | K2 | K3 | K4 | K5 | K6 | K7 |
| K1 |  | 3 | 1 | 1 | 1/2 | 1/6 | 1/4 |
| K2 |  |  | 1/4 | 1/3 | 1/3 | 1/4 | 1/6 |
| K3 |  |  |  | 1 | 1 | 1/2 | 1 |
| K4 |  |  |  |  | 1/2 | 1/4 | 1/5 |
| K5 |  |  |  |  |  | 1 | 1/2 |
| K6 |  |  |  |  |  |  | 1/2 |
| K7 |  |  |  |  |  |  |  |

Nilai rata-rata geometris tersebut kemudian dimasukkan ke *software* Expert Choice 11 untuk kemudian dapat diolah. Tampilan matriks nilai rata-rata geometris pada Expert Choice 11 dapat dilihat pada Gambar 4.2



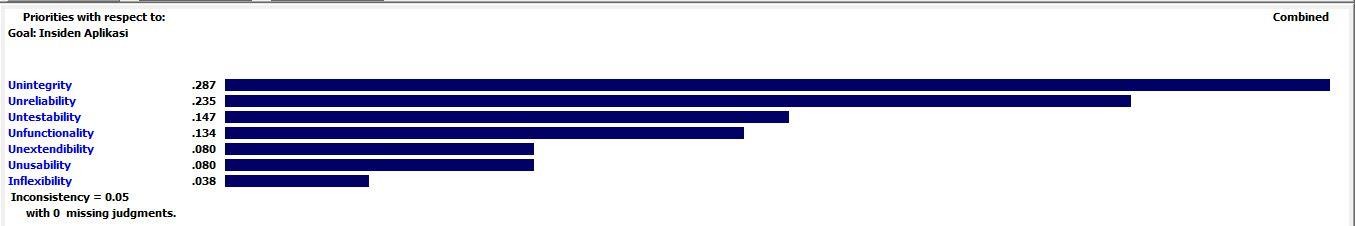
## Gambar 4.2 Tabel Penilaian Kriteria Aplikasi Penyebab Kegagalan Aplikasi pada Expert Choice 11

Setelah nilai rata-rata geometris tersebut dimasukkan ke dalam *software* Expert Choice 11, didapatkan nilai eigen untuk setiap kriteria sebagai berikut:



## Gambar 4.3 Nilai Eigen dan Konsistensi Kriteria Penyebab Kegagalan Aplikasi

Untuk mendapatkan hasil sesuai dengan urutan potensinya, maka dipilih *Sort by priority* pada software Expert Choice 11 tersebut, sehingga menampilkan hasil sebagai berikut:



**Gambar 4.4 Nilai Eigen Kriteria Penyebab Kegagalan Aplikasi (*Sort by priority*)**

Gambar 4.4 di atas menampilkan nilai eigen untuk setiap kriteria aplikasi penyebab kegagalan aplikasi dengan nilai konsistensi sebesar 0.05, maka hasil ini sudah sesuai dengan aturan AHP di mana CR (*Consistency Ratio*) harus lebih kecil dari 10% atau 0.1.

## Bobot Kategori Faktor Penyebab Kegagalan Aplikasi

Matriks nilai rata-rata geometris dihitung untuk setiap kategori faktor penyebab kegagalan aplikasi terkait dengan masing-masing kriteria.

* + - 1. **Kriteria *Unextendibility***

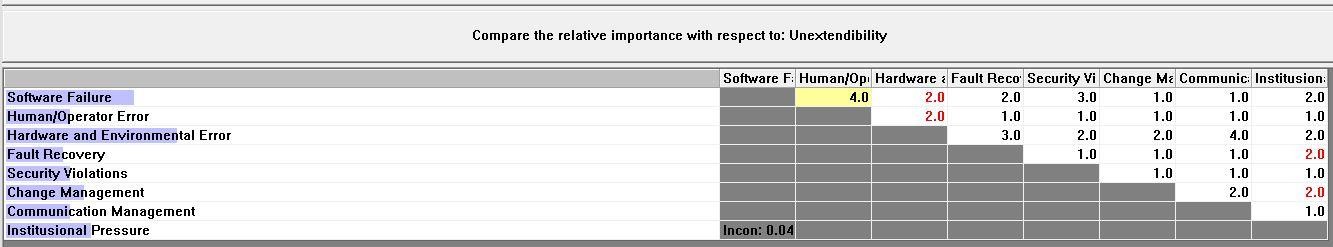
Nilai rata-rata geometris yang telah dibulatkan kemudian dimasukkan ke matriks perbandingan berpasangan. Bila nilai rata-rata geometris tersebut dibulatkan dari nilai *reciprocal*, maka di matriks perbandingan berpasangan nilainya menjadi

1/hasil pembulatan. Matriks perbandingan berpasangan untuk setiap kategori faktor terkait kriteria *Unextendibility* dapat dilihat pada Tabel 4.8.

## Tabel 4.8 Pembulatan Rata-Rata Geometris Kategori Faktor Terhadap Kriteria *Unextendibility*

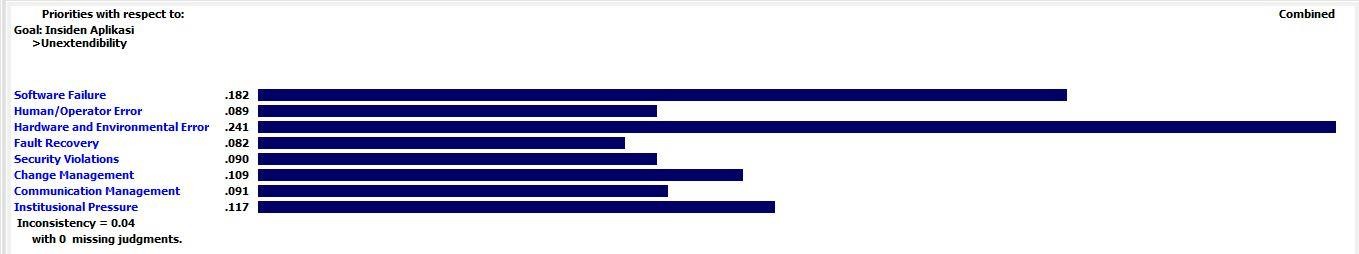
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 |
| A1 |  | 4 | 1/2 | 2 | 3 | 1 | 1 | 1/2 |
| A2 |  |  | 1/2 | 1 | 1 | 1 | 1 | 1 |
| A3 |  |  |  | 3 | 2 | 2 | 4 | 2 |
| A4 |  |  |  |  | 1 | 1 | 1 | 1/2 |
| A5 |  |  |  |  |  | 1 | 1 | 1 |
| A6 |  |  |  |  |  |  | 2 | 1/2 |
| A7 |  |  |  |  |  |  |  | 1 |
| A8 |  |  |  |  |  |  |  |  |

Nilai rata-rata geometris tersebut kemudian dimasukkan ke *software* Expert Choice 11 untuk kemudian dapat diolah. Tampilan matriks nilai rata-rata geometris pada Expert Choice 11 dapat dilihat pada Gambar 4.5



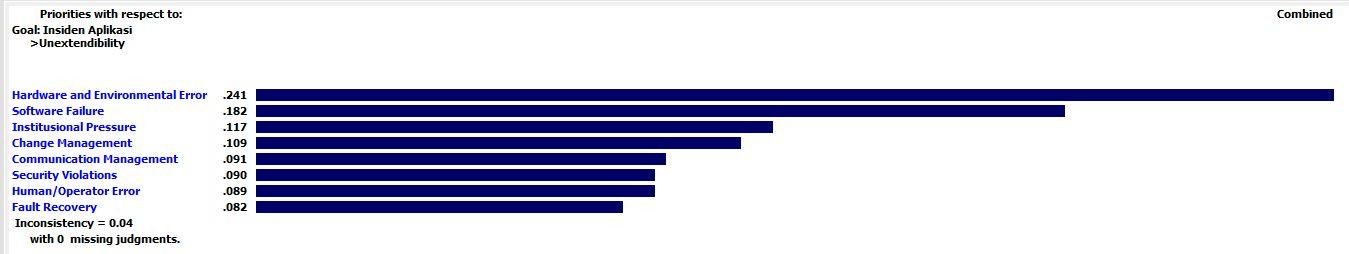
## Gambar 4.5 Tabel Penilaian Kategori Faktor Penyebab Kegagalan Aplikasi Terhadap Kriteria *Unextendibility* pada Expert Choice 11

Setelah nilai rata-rata geometris tersebut dimasukkan ke dalam *software* Expert Choice 11, didapatkan nilai eigen untuk setiap kategori faktor terhadap kriteria *Unextendibility* sebagai berikut:



## Gambar 4.6 Nilai Eigen dan Konsistensi Kategori Faktor Penyebab Kegagalan Aplikasi Terhadap Kriteria *Unextendibility*

Untuk mendapatkan hasil sesuai dengan urutan potensinya, maka dipilih *Sort by priority* pada software Expert Choice 11 tersebut, sehingga menampilkan hasil sebagai berikut:



**Gambar 4.7 Nilai Eigen Kategori Faktor terhadap Kriteria *Unextendibility***

### (*Sort by priority*)

Gambar 4.7 di atas menampilkan nilai eigen untuk setiap kategori faktor penyebab kegagalan aplikasi terhadap kriteria *Unextendibility* dengan nilai konsistensi sebesar 0.04, maka hasil ini sudah sesuai dengan aturan AHP di mana CR (*Consistency Ratio*) harus lebih kecil dari 10% atau 0.1.

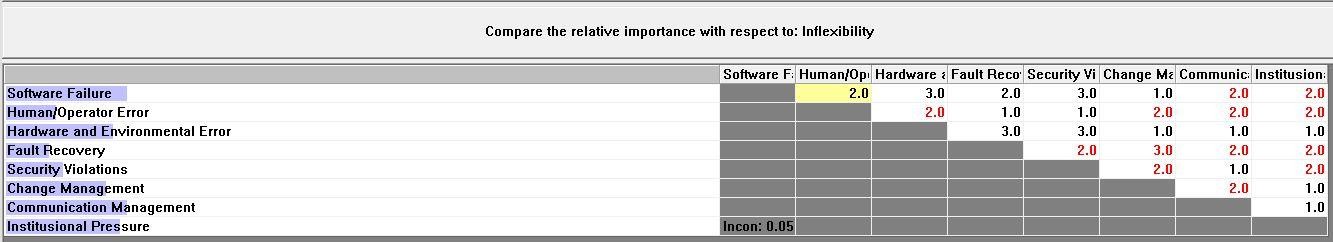
* + - 1. **Kriteria *Inflexibility***

Nilai rata-rata geometris yang telah dibulatkan kemudian dimasukkan ke matriks perbandingan berpasangan. Bila nilai rata-rata geometris tersebut dibulatkan dari nilai *reciprocal*, maka di matriks perbandingan berpasangan nilainya menjadi 1/hasil pembulatan. Matriks perbandingan berpasangan untuk setiap kategori faktor terkait kriteria *Inflexibility* dapat dilihat pada Tabel 4.9.

## Tabel 4.9 Pembulatan Rata-Rata Geometris Kategori Faktor Terhadap Kriteria *Inflexibility*

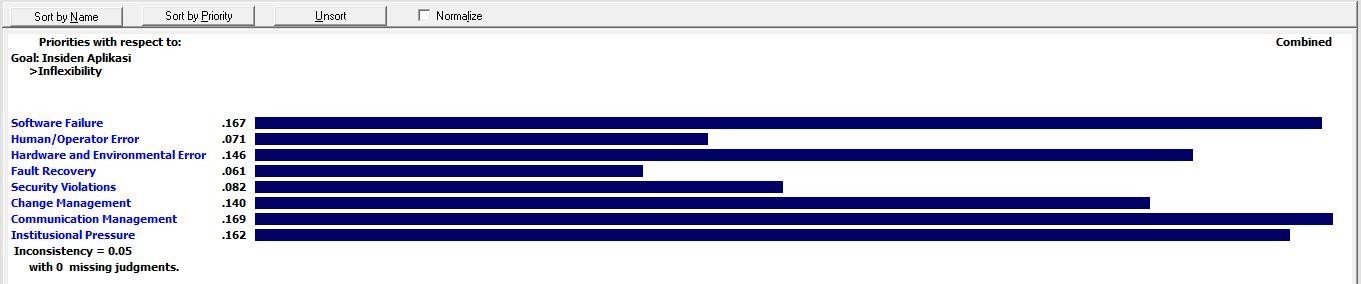
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 |
| A1 |  | 2 | 3 | 2 | 3 | 1 | 1/2 | 1/2 |
| A2 |  |  | 1/2 | 1 | 1 | 1/2 | 1/2 | 1/2 |
| A3 |  |  |  | 3 | 3 | 1 | 1 | 1 |
| A4 |  |  |  |  | 1/2 | 1/3 | 1/2 | 1/2 |
| A5 |  |  |  |  |  | 1/2 | 1 | 1/2 |
| A6 |  |  |  |  |  |  | 1/2 | 1 |
| A7 |  |  |  |  |  |  |  | 1 |
| A8 |  |  |  |  |  |  |  |  |

Nilai rata-rata geometris tersebut kemudian dimasukkan ke *software* Expert Choice 11 untuk kemudian dapat diolah. Tampilan matriks nilai rata-rata geometris pada Expert Choice 11 dapat dilihat pada Gambar 4.8



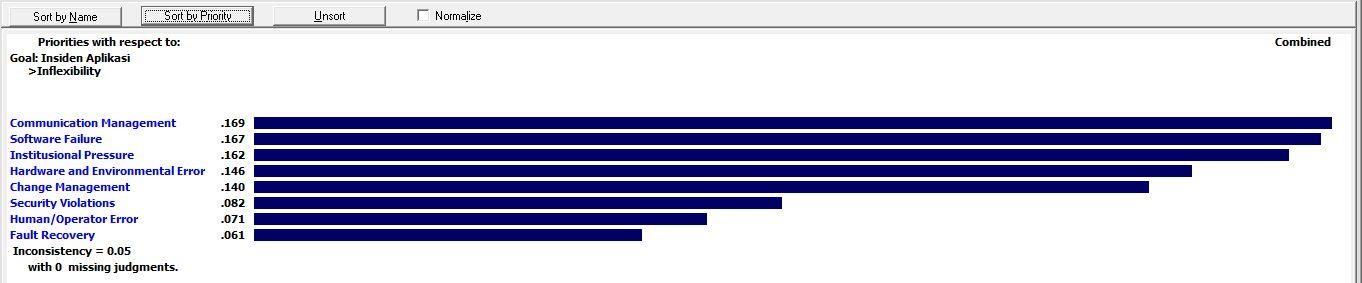
## Gambar 4.8 Tabel Penilaian Kategori Faktor Penyebab Kegagalan Aplikasi Terhadap Kriteria *Inflexibility* pada Expert Choice 11

Setelah nilai rata-rata geometris tersebut dimasukkan ke dalam *software* Expert Choice 11, didapatkan nilai eigen untuk setiap kategori faktor terhadap kriteria *Inflexibility* sebagai berikut:



## Gambar 4.9 Nilai Eigen dan Konsistensi Kategori Faktor Penyebab Kegagalan Aplikasi Terhadap Kriteria *Inflexibility*

Untuk mendapatkan hasil sesuai dengan urutan potensinya, maka dipilih *Sort by priority* pada software Expert Choice 11 tersebut, sehingga menampilkan hasil sebagai berikut:



## Gambar 4.10 Nilai Eigen Kategori Faktor terhadap Kriteria *Inflexibility*

### (*Sort by priority*)

Gambar 4.10 di atas menampilkan nilai eigen untuk setiap kategori faktor penyebab kegagalan aplikasi terhadap kriteria *Inflexibility* dengan nilai konsistensi

sebesar 0.05, maka hasil ini sudah sesuai dengan aturan AHP di mana CR (*Consistency Ratio*) harus lebih kecil dari 10% atau 0.1.

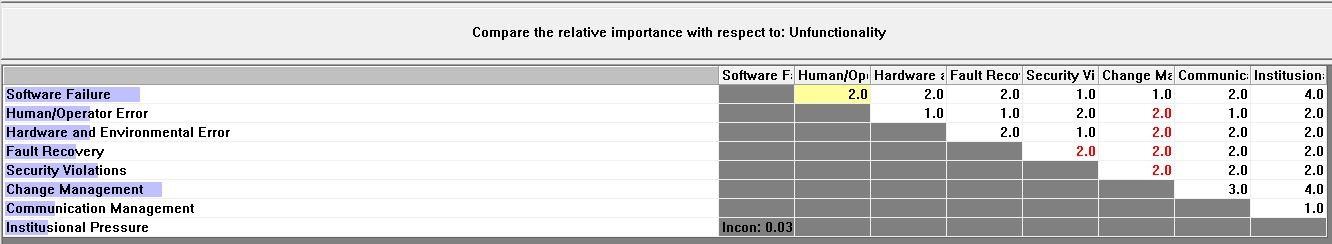
* + - 1. **Kriteria *Unfunctionality***

Nilai rata-rata geometris yang telah dibulatkan kemudian dimasukkan ke matriks perbandingan berpasangan. Bila nilai rata-rata geometris tersebut dibulatkan dari nilai *reciprocal*, maka di matriks perbandingan berpasangan nilainya menjadi 1/hasil pembulatan. Matriks perbandingan berpasangan untuk setiap kategori faktor terkait kriteria *Unfunctionality* dapat dilihat pada Tabel 4. 10.

## Tabel 4.10 Pembulatan Rata-Rata Geometris Kategori Faktor Terhadap Kriteria *Unfunctionality*

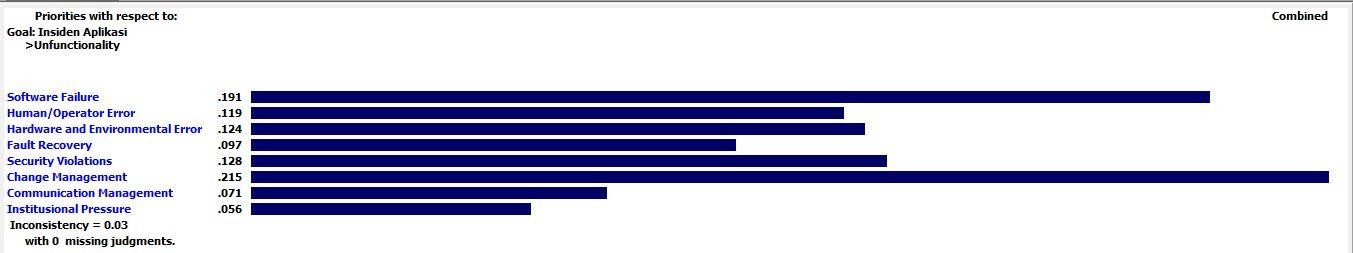
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 |
| A1 |  | 2 | 2 | 2 | 1 | 1 | 2 | 4 |
| A2 |  |  | 1 | 1 | 2 | 1/2 | 1 | 2 |
| A3 |  |  |  | 2 | 1 | 1/2 | 2 | 2 |
| A4 |  |  |  |  | 1/2 | 1/2 | 2 | 2 |
| A5 |  |  |  |  |  | 1/2 | 2 | 2 |
| A6 |  |  |  |  |  |  | 3 | 4 |
| A7 |  |  |  |  |  |  |  | 1 |
| A8 |  |  |  |  |  |  |  |  |

Nilai rata-rata geometris tersebut kemudian dimasukkan ke *software* Expert Choice 11 untuk kemudian dapat diolah. Tampilan matriks nilai rata-rata geometris pada Expert Choice 11 dapat dilihat pada Gambar 4.11.



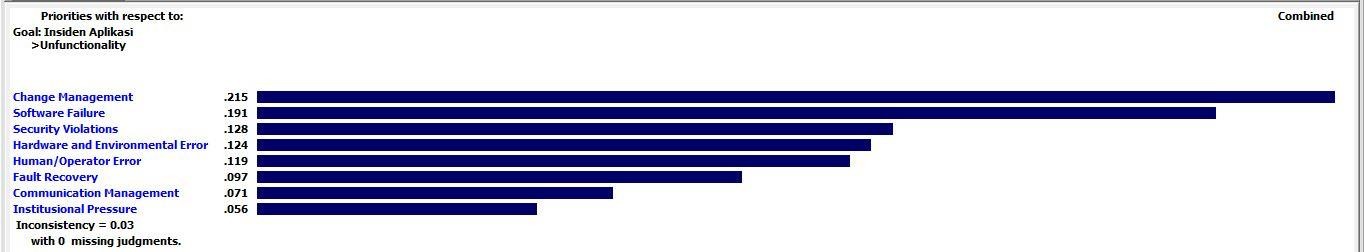
## Gambar 4.11 Tabel Penilaian Kategori Faktor Penyebab Kegagalan Aplikasi Terhadap Kriteria *Unfunctionality* pada Expert Choice 11

Setelah nilai rata-rata geometris tersebut dimasukkan ke dalam s*oftware* Expert Choice 11, didapatkan nilai eigen untuk setiap kategori faktor terhadap kriteria *Unfunctionality* sebagai berikut:



## Gambar 4.12 Nilai Eigen dan Konsistensi Kategori Faktor Penyebab Kegagalan Aplikasi Terhadap Kriteria *Unfunctionality*

Untuk mendapatkan hasil sesuai dengan urutan potensinya, maka dipilih *Sort by priority* pada software Expert Choice 11 tersebut, sehingga menampilkan hasil sebagai berikut:



**Gambar 4.13 Nilai Eigen Kategori Faktor terhadap Kriteria *Unfunctionality***

### (*Sort by priority*)

Gambar 4.13 di atas menampilkan nilai eigen untuk setiap kategori faktor penyebab kegagalan aplikasi terhadap kriteria *Unfunctionality* dengan nilai konsistensi sebesar 0.03, maka hasil ini sudah sesuai dengan aturan AHP di mana CR (*Consistency Ratio*) harus lebih kecil dari 10% atau 0.1.

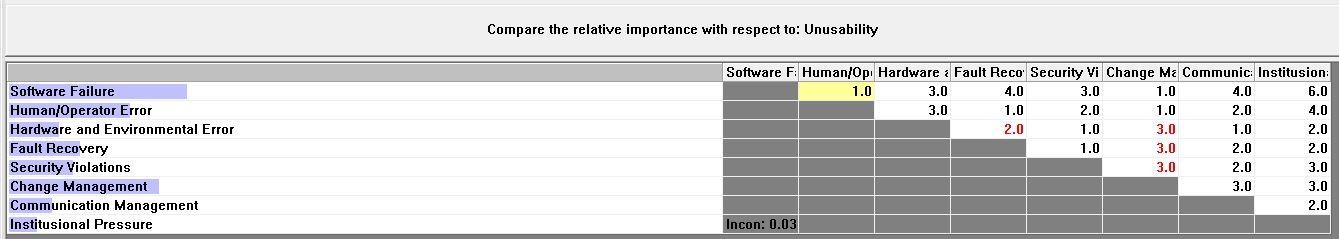
* + - 1. **Kriteria *Unusability***

Nilai rata-rata geometris yang telah dibulatkan kemudian dimasukkan ke matriks perbandingan berpasangan. Bila nilai rata-rata geometris tersebut dibulatkan dari nilai *reciprocal*, maka di matriks perbandingan berpasangan nilainya menjadi 1/hasil pembulatan. Matriks perbandingan berpasangan untuk setiap kategori faktor terkait kriteria *Unusability* dapat dilihat pada Tabel 4. 11.

## Tabel 4.11 Pembulatan Rata-Rata Geometris Kategori Faktor Terhadap Kriteria *Unusability*

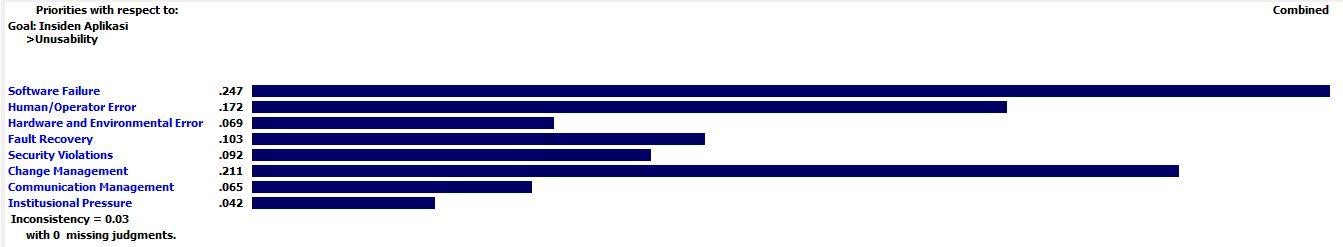
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 |
| A1 |  | 1 | 3 | 4 | 3 | 1 | 4 | 6 |
| A2 |  |  | 3 | 1 | 2 | 1 | 2 | 4 |
| A3 |  |  |  | 1/2 | 1 | 1/3 | 1 | 2 |
| A4 |  |  |  |  | 1 | 1/3 | 2 | 2 |
| A5 |  |  |  |  |  | 1/3 | 2 | 3 |
| A6 |  |  |  |  |  |  | 3 | 3 |
| A7 |  |  |  |  |  |  |  | 2 |
| A8 |  |  |  |  |  |  |  |  |

Nilai rata-rata geometris tersebut kemudian dimasukkan ke *software* Expert Choice 11 untuk kemudian dapat diolah. Tampilan matriks nilai rata-rata geometris pada Expert Choice 11 dapat dilihat pada Gambar 4.14



## Gambar 4.14Tabel Penilaian Kategori Faktor Penyebab Kegagalan Aplikasi Terhadap Kriteria *Unusability* pada Expert Choice 11

Setelah nilai rata-rata geometris tersebut dimasukkan ke dalam *software* Expert Choice 11, didapatkan nilai eigen untuk setiap kategori faktor terhadap kriteria *Unusability* sebagai berikut:



## Gambar 4.15 Nilai Eigen dan Konsistensi Kategori Faktor Penyebab Kegagalan Aplikasi Terhadap Kriteria *Unusability*

Untuk mendapatkan hasil sesuai dengan urutan potensinya, maka dipilih *Sort by priority* pada software Expert Choice 11 tersebut, sehingga menampilkan hasil sebagai berikut:

**Gambar 4.16 Nilai Eigen Kategori Faktor terhadap Kriteria *Unusability* (*Sort by priority*)**

Gambar 4.16 di atas menampilkan nilai eigen untuk setiap kategori faktor penyebab kegagalan aplikasi terhadap kriteria *Unfunctionality* dengan nilai konsistensi sebesar 0.03, maka hasil ini sudah sesuai dengan aturan AHP di mana CR (*Consistency Ratio*) harus lebih kecil dari 10% atau 0.1.

* + - 1. **Kriteria *Untestability***

Nilai rata-rata geometris yang telah dibulatkan kemudian dimasukkan ke matriks perbandingan berpasangan. Bila nilai rata-rata geometris tersebut dibulatkan dari nilai *reciprocal*, maka di matriks perbandingan berpasangan nilainya menjadi 1/hasil pembulatan. Matriks perbandingan berpasangan untuk setiap kategori faktor terkait kriteria *Untestability* dapat dilihat pada Tabel 4. 12.

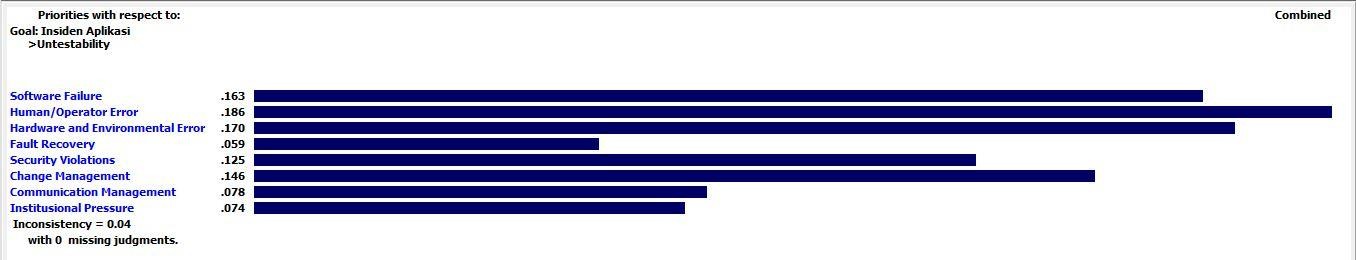
## Tabel 4.12 Pembulatan Rata-Rata Geometris Kategori Faktor Terhadap Kriteria *Untestability*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 |
| A1 |  | 1 | 2 | 2 | 1 | 1 | 1 | 3 |
| A2 |  |  | 1 | 3 | 1 | 2 | 3 | 3 |
| A3 |  |  |  | 5 | 2 | 1 | 2 | 2 |
| A4 |  |  |  |  | 1/2 | 1/3 | 1 | 1 |
| A5 |  |  |  |  |  | 1/2 | 2 | 2 |
| A6 |  |  |  |  |  |  | 2 | 1 |
| A7 |  |  |  |  |  |  |  | 1 |
| A8 |  |  |  |  |  |  |  |  |

Nilai rata-rata geometris tersebut kemudian dimasukkan ke *software* Expert Choice 11 untuk kemudian dapat diolah. Tampilan matriks nilai rata-rata geometris pada Expert Choice 11 dapat dilihat pada Gambar 4.17.

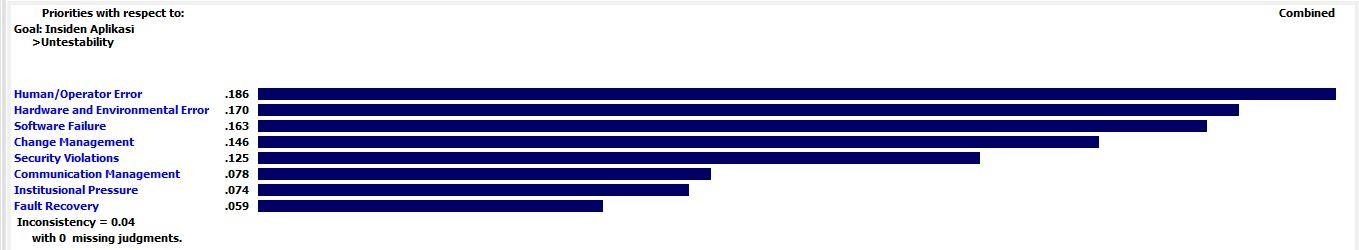
## Gambar 4.17Tabel Penilaian Kategori Faktor Penyebab Kegagalan Aplikasi Terhadap Kriteria *Untestability* pada Expert Choice 11

Setelah nilai rata-rata geometris tersebut dimasukkan ke dalam *software* Expert Choice 11, didapatkan nilai eigen untuk setiap kategori faktor terhadap kriteria *Untestability* sebagai berikut:



## Gambar 4.18 Nilai Eigen dan Konsistensi Kategori Faktor Penyebab Kegagalan Aplikasi Terhadap Kriteria *Untestability*

Untuk mendapatkan hasil sesuai dengan urutan potensinya, maka dipilih *Sort by priority* pada software Expert Choice 11 tersebut, sehingga menampilkan hasil sebagai berikut:



## Gambar 4.19 Nilai Eigen Kategori Faktor terhadap Kriteria *Untestability*

### (*Sort by priority*)

Gambar 4.19 di atas menampilkan nilai eigen untuk setiap kategori faktor penyebab kegagalan aplikasi terhadap kriteria *Unfunctionality* dengan nilai konsistensi sebesar 0.04, maka hasil ini sudah sesuai dengan aturan AHP di mana CR (*Consistency Ratio*) harus lebih kecil dari 10% atau 0.1.

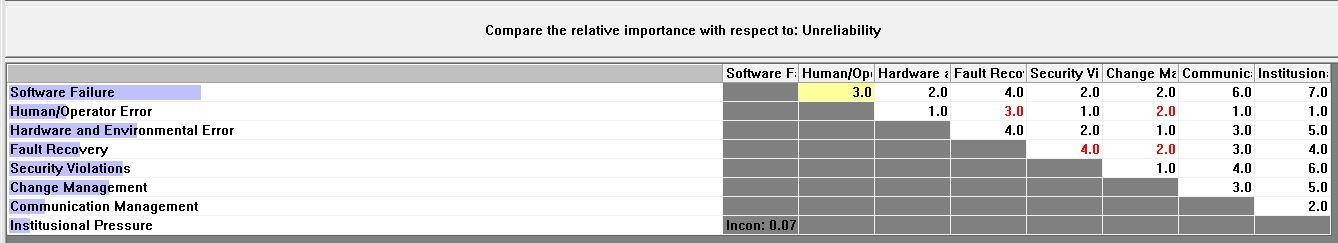
* + - 1. **Kriteria *Unreliability***

Nilai rata-rata geometris yang telah dibulatkan kemudian dimasukkan ke matriks perbandingan berpasangan. Bila nilai rata-rata geometris tersebut dibulatkan dari nilai *reciprocal*, maka di matriks perbandingan berpasangan nilainya menjadi 1/hasil pembulatan. Matriks perbandingan berpasangan untuk setiap kategori faktor terkait kriteria *Unreliability* dapat dilihat pada Tabel 4. 13.

## Tabel 4.13 Pembulatan Rata-Rata Geometris Kategori Faktor Terhadap Kriteria *Unreliability*

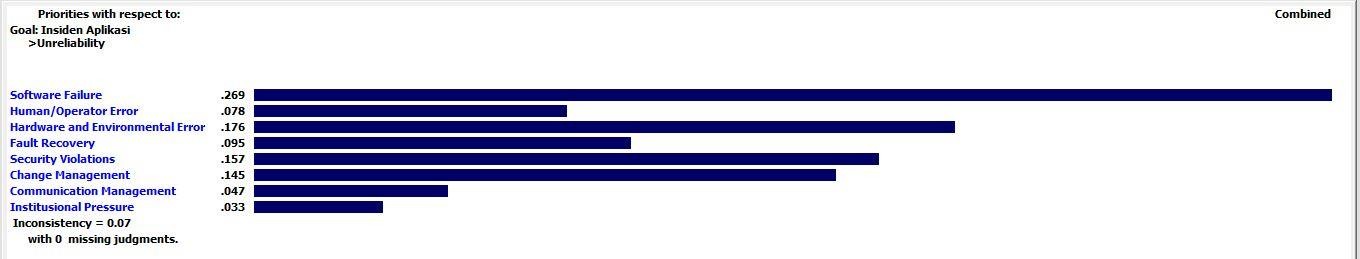
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 |
| A1 |  | 3 | 2 | 4 | 2 | 2 | 6 | 7 |
| A2 |  |  | 1 | 1/3 | 1 | 1/2 | 1 | 1 |
| A3 |  |  |  | 4 | 2 | 1 | 3 | 5 |
| A4 |  |  |  |  | 1/4 | 1/2 | 3 | 4 |
| A5 |  |  |  |  |  | 1 | 4 | 6 |
| A6 |  |  |  |  |  |  | 3 | 5 |
| A7 |  |  |  |  |  |  |  | 2 |
| A8 |  |  |  |  |  |  |  |  |

Nilai rata-rata geometris tersebut kemudian dimasukkan ke *software* Expert Choice 11 untuk kemudian dapat diolah. Tampilan matriks nilai rata-rata geometris pada Expert Choice 11 dapat dilihat pada Gambar 4.20



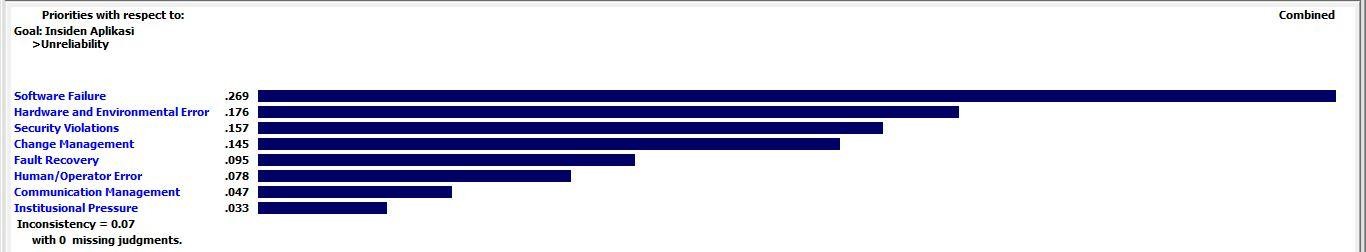
## Gambar 4.20Tabel Penilaian Kategori Faktor Penyebab Kegagalan Aplikasi Terhadap Kriteria *Unreliability* pada Expert Choice 11

Setelah nilai rata-rata geometris tersebut dimasukkan ke dalam *software* Expert Choice 11, didapatkan nilai eigen untuk setiap kategori faktor terhadap kriteria *Unreliability* sebagai berikut:



## Gambar 4.21 Nilai Eigen dan Konsistensi Kategori Faktor Penyebab Kegagalan Aplikasi Terhadap Kriteria *Unreliability*

Untuk mendapatkan hasil sesuai dengan urutan potensinya, maka dipilih *Sort by priority* pada software Expert Choice 11 tersebut, sehingga menampilkan hasil sebagai berikut:



## Gambar 4.22 Nilai Eigen Kategori Faktor terhadap Kriteria *Unreliability*

### (*Sort by priority*)

Gambar 4.22 di atas menampilkan nilai eigen untuk setiap kategori faktor penyebab kegagalan aplikasi terhadap kriteria *Unfunctionality* dengan nilai konsistensi sebesar 0.07, maka hasil ini sudah sesuai dengan aturan AHP di mana CR (*Consistency Ratio*) harus lebih kecil dari 10% atau 0.1.

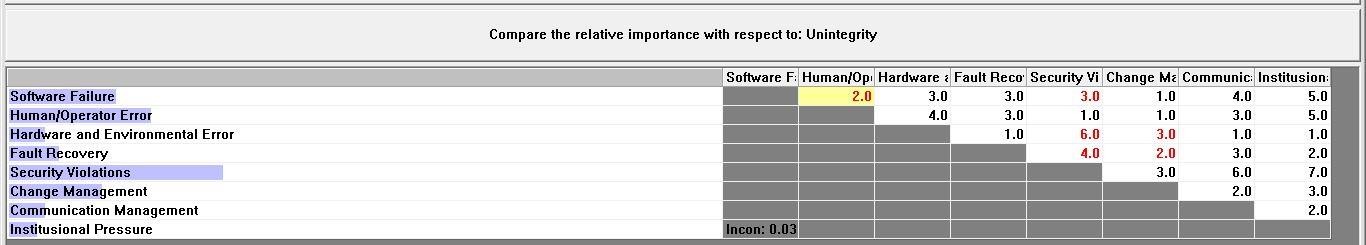
* + - 1. **Kriteria *Unintegrity***

Nilai rata-rata geometris yang telah dibulatkan kemudian dimasukkan ke matriks perbandingan berpasangan. Bila nilai rata-rata geometris tersebut dibulatkan dari nilai *reciprocal*, maka di matriks perbandingan berpasangan nilainya menjadi 1/hasil pembulatan. Matriks perbandingan berpasangan untuk setiap kategori faktor terkait kriteria *Unintegrity* dapat dilihat pada Tabel 4. 14.

## Tabel 4.14 Pembulatan Rata-Rata Geometris Kategori Faktor Terhadap Kriteria *Unintegrity*

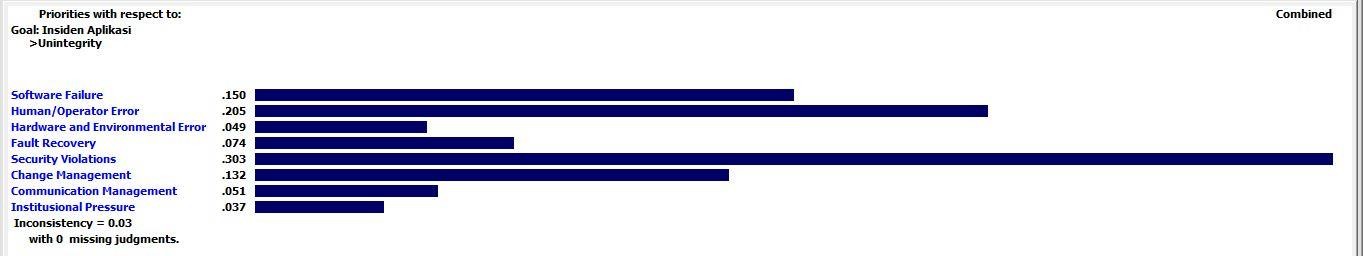
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 |
| A1 |  | 1/2 | 3 | 3 | 1/3 | 1 | 4 | 5 |
| A2 |  |  | 4 | 3 | 1 | 1 | 3 | 5 |
| A3 |  |  |  | 1 | 1/6 | 1/3 | 1 | 1 |
| A4 |  |  |  |  | 1/4 | ½ | 3 | 2 |
| A5 |  |  |  |  |  | 3 | 6 | 7 |
| A6 |  |  |  |  |  |  | 2 | 3 |
| A7 |  |  |  |  |  |  |  | 3 |
| 3 |  |  |  |  |  |  |  |  |

Nilai rata-rata geometris tersebut kemudian dimasukkan ke *software* Expert Choice 11 untuk kemudian dapat diolah. Tampilan matriks nilai rata-rata geometris pada Expert Choice 11 dapat dilihat pada Gambar 4.23



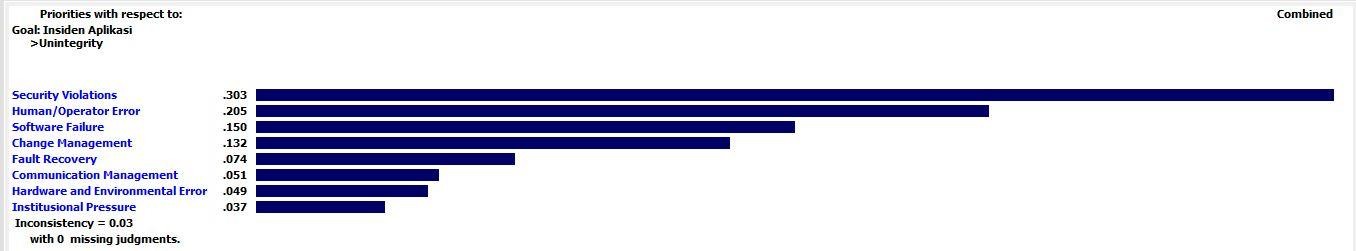
## Gambar 4.23Tabel Penilaian Kategori Faktor Penyebab Kegagalan Aplikasi Terhadap Kriteria *Unintegrity* pada Expert Choice 11

Setelah nilai rata-rata geometris tersebut dimasukkan ke dalam *software* Expert Choice 11, didapatkan nilai eigen untuk setiap kategori faktor terhadap kriteria *Unusability* sebagai berikut:



## Gambar 4.24 Nilai Eigen dan Konsistensi Kategori Faktor Penyebab Kegagalan Aplikasi Terhadap Kriteria *Unintegrity*

Untuk mendapatkan hasil sesuai dengan urutan potensinya, maka dipilih *Sort by priority* pada software Expert Choice 11 tersebut, sehingga menampilkan hasil sebagai berikut:



**Gambar 4.25 Nilai Eigen Kategori Faktor terhadap Kriteria *Unintegrity* (*Sort by priority*)**

Gambar 4.25 di atas menampilkan nilai eigen untuk setiap kategori faktor penyebab kegagalan aplikasi terhadap kriteria *Unfunctionality* dengan nilai konsistensi sebesar 0.03, maka hasil ini sudah sesuai dengan aturan AHP di mana CR (*Consistency Ratio*) harus lebih kecil dari 10% atau 0.1.

## Bobot Kriteria Faktor

Matriks nilai rata-rata geometris dihitung untuk setiap pasang faktor pada masing- masing kategori faktor yang berkontribusi terhadap kriteria kegagalan aplikasi.

* + - 1. **Kategori *Software Error***

Nilai rata-rata geometris yang telah dibulatkan kemudian dimasukkan ke matriks perbandingan berpasangan. Bila nilai rata-rata geometris tersebut dibulatkan dari nilai *reciprocal*, maka di matriks perbandingan berpasangan nilainya menjadi 1/hasil pembulatan. Matriks perbandingan berpasangan untuk setiap faktor pada kategori *Software Error* dapat dilihat pada Tabel 4. 15.

## Tabel 4.15 Pembulatan Rata-Rata Geometris Faktor Penyebab Kegagalan Aplikasi pada Kategori *Software Error*

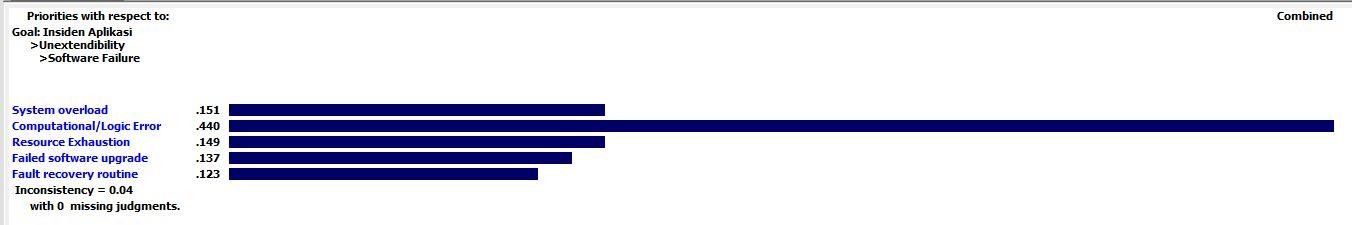
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A1.1 | A1.2 | A1.3 | A1.4 | A1.5 |
| A1.1 |  | 1/2 | 1 | 1 | 1 |
| A1.2 |  |  | 5 | 3 | 3 |
| A1.3 |  |  |  | 1 | 2 |
| A1.4 |  |  |  |  | 1 |
| A1.5 |  |  |  |  |  |

Nilai rata-rata geometris tersebut kemudian dimasukkan ke *software* Expert Choice 11 untuk kemudian dapat diolah. Tampilan matriks nilai rata-rata geometris pada Expert Choice 11 dapat dilihat pada Gambar 4.26.



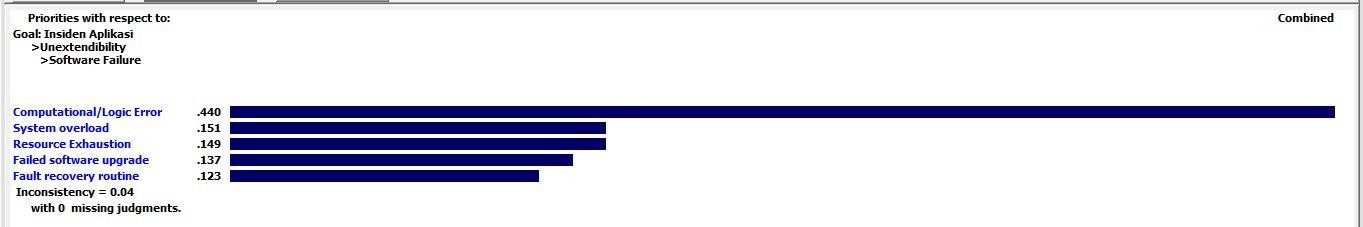
## Gambar 4.26Tabel Penilaian Faktor Penyebab Kegagalan Aplikasi pada Kategori *Software Error* di Expert Choice 11

Setelah nilai rata-rata geometris tersebut dimasukkan ke dalam *software* Expert Choice 11, didapatkan nilai eigen untuk setiap faktor pada kategori faktor *Software Error* sebagai berikut:



## Gambar 4.27 Nilai Eigen dan Konsistensi Faktor Penyebab Kegagalan Aplikasi pada Kategori *Software Error*

Untuk mendapatkan hasil sesuai dengan urutan potensinya, maka dipilih *Sort by priority* pada software Expert Choice 11 tersebut, sehingga menampilkan hasil sebagai berikut:



**Gambar 4.28 Nilai Eigen Faktor Penyebab Kegagalan Aplikasi pada Kategori *Software Failure* (*Sort by priority*)**

Gambar 4.28 di atas menampilkan nilai eigen untuk setiap faktor penyebab kegagalan aplikasi terhadap kategori *Software Failure* dengan nilai konsistensi sebesar 0.04, maka hasil ini sudah sesuai dengan aturan AHP di mana CR (*Consistency Ratio*) harus lebih kecil dari 10% atau 0.1.

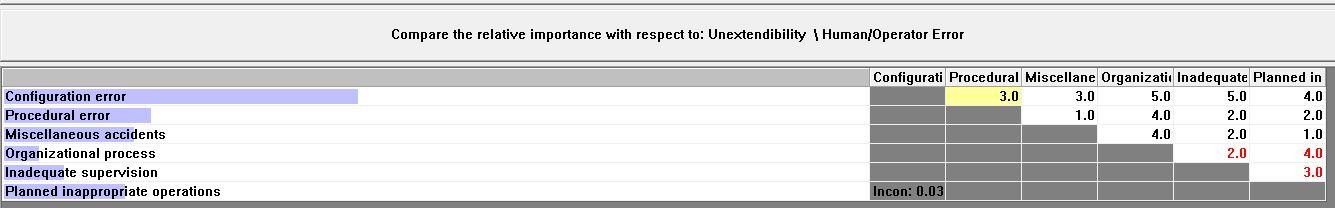
* + - 1. **Kategori *Human/Operator Error***

Nilai rata-rata geometris yang telah dibulatkan kemudian dimasukkan ke matriks perbandingan berpasangan. Bila nilai rata-rata geometris tersebut dibulatkan dari nilai *reciprocal*, maka di matriks perbandingan berpasangan nilainya menjadi 1/hasil pembulatan. Matriks perbandingan berpasangan untuk setiap faktor pada kategori *Human/Operator Error* dapat dilihat pada Tabel 4. 16.

## Tabel 4.16 Pembulatan Rata-Rata Geometris Faktor Penyebab Kegagalan Aplikasi pada Kategori *Human/Operator Error*

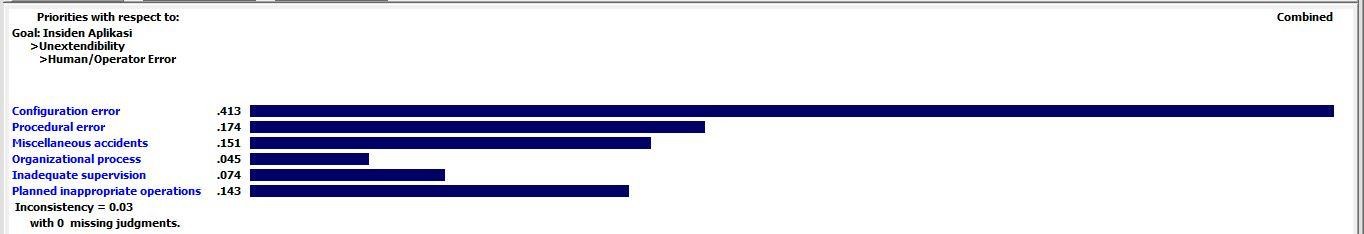
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | A2.1 | A2.2 | A2.3 | A2.4 | A2.5 | A2.6 |
| A2.1 |  | 3 | 3 | 5 | 5 | 4 |
| A2.2 |  |  | 1 | 4 | 2 | 2 |
| A2.3 |  |  |  | 4 | 2 | 1 |
| A2.4 |  |  |  |  | 1/2 | 1/4 |
| A2.5 |  |  |  |  |  | 1/3 |
| A2.6 |  |  |  |  |  |  |

Nilai rata-rata geometris tersebut kemudian dimasukkan ke *software* Expert Choice 11 untuk kemudian dapat diolah. Tampilan matriks nilai rata-rata geometris pada Expert Choice 11 dapat dilihat pada Gambar 4.29.



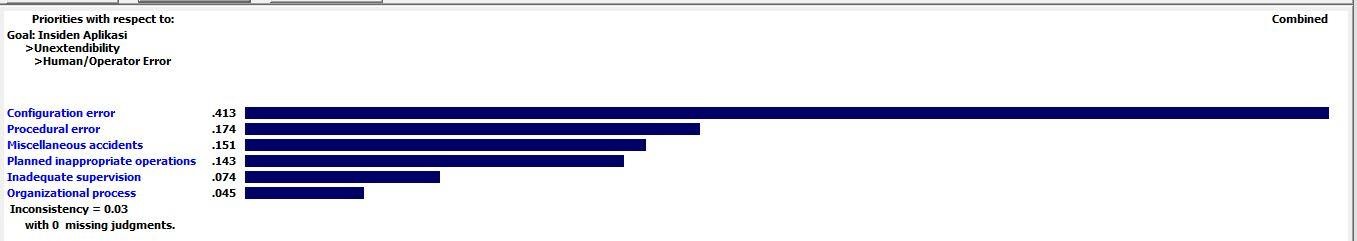
## Gambar 4.29Tabel Penilaian Faktor Penyebab Kegagalan Aplikasi pada Kategori *Human/Operator Error* di Expert Choice 11

Setelah nilai rata-rata geometris tersebut dimasukkan ke dalam *software* Expert Choice 11, didapatkan nilai eigen untuk setiap faktor pada kategori faktor *Human/Operator Error* sebagai berikut:



## Gambar 4.30 Nilai Eigen dan Konsistensi Faktor Penyebab Kegagalan Aplikasi pada Kategori *Human/Operator Error*

Untuk mendapatkan hasil sesuai dengan urutan potensinya, maka dipilih *Sort by priority* pada software Expert Choice 11 tersebut, sehingga menampilkan hasil sebagai berikut:



**Gambar 4.31 Nilai Eigen Faktor Penyebab Kegagalan Aplikasi pada Kategori *Human/Operator Error* (*Sort by priority*)**

Gambar 4.31 di atas menampilkan nilai eigen untuk setiap faktor penyebab kegagalan aplikasi terhadap kategori *Human/Operator Error* dengan nilai konsistensi sebesar 0.03, maka hasil ini sudah sesuai dengan aturan AHP di mana CR (*Consistency Ratio*) harus lebih kecil dari 10% atau 0.1.

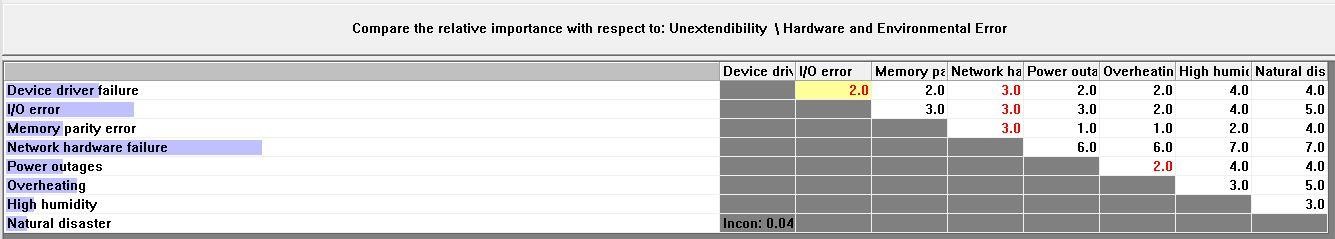
* + - 1. **Kategori *Hardware/Environmental Error***

Nilai rata-rata geometris yang telah dibulatkan kemudian dimasukkan ke matriks perbandingan berpasangan. Bila nilai rata-rata geometris tersebut dibulatkan dari nilai *reciprocal*, maka di matriks perbandingan berpasangan nilainya menjadi 1/hasil pembulatan. Matriks perbandingan berpasangan untuk setiap faktor pada kategori *Hardware/Environmental Error* dapat dilihat pada Tabel 4. 17.

**Tabel 4.17 Pembulatan Rata-Rata Geometris Faktor Penyebab Kegagalan Aplikasi pada Kategori *Hardware/Environmental Error***

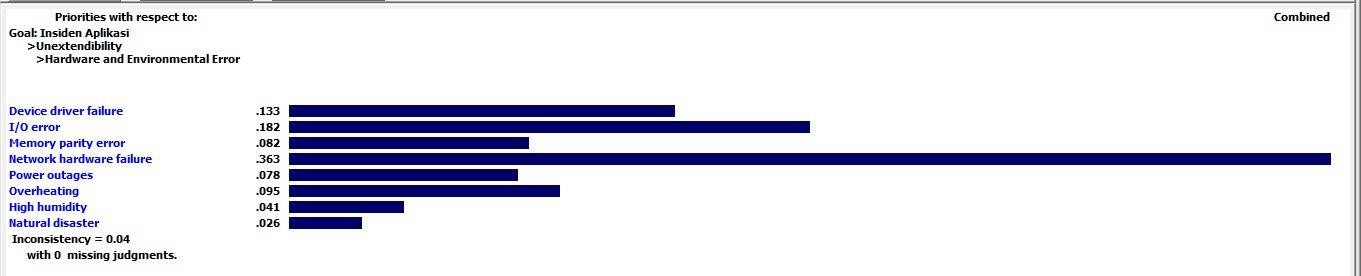
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A3.1 | A3.2 | A3.3 | A3.4 | A3.5 | A3.6 | A3.7 | A3.8 |
| A3.1 |  | 1/2 | 2 | 1/3 | 2 | 2 | 4 | 4 |
| A3.2 |  |  | 3 | 1/3 | 3 | 2 | 4 | 5 |
| A3.3 |  |  |  | 1/3 | 1 | 1 | 2 | 4 |
| A3.4 |  |  |  |  | 6 | 6 | 7 | 7 |
| A3.5 |  |  |  |  |  | 1/2 | 4 | 4 |
| A3.6 |  |  |  |  |  |  | 3 | 5 |
| A3.7 |  |  |  |  |  |  |  | 3 |
| A3.6 |  |  |  |  |  |  |  |  |

Nilai rata-rata geometris tersebut kemudian dimasukkan ke *software* Expert Choice 11 untuk kemudian dapat diolah. Tampilan matriks nilai rata-rata geometris pada Expert Choice 11 dapat dilihat pada Gambar 4.26.



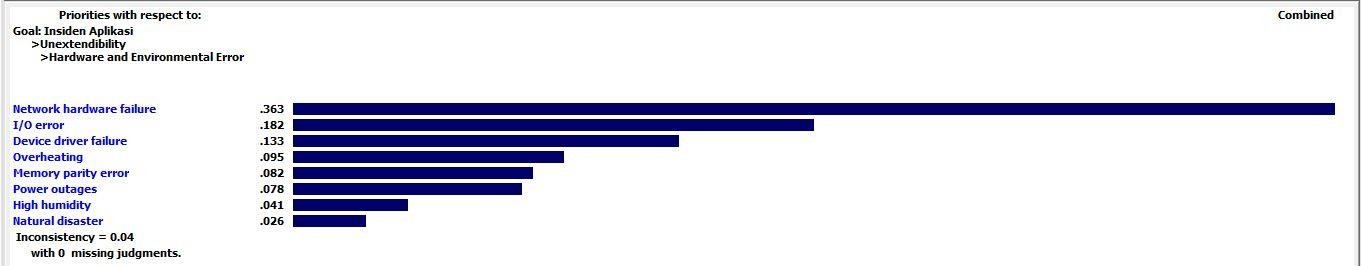
**Gambar 4.32Tabel Penilaian Faktor Penyebab Kegagalan Aplikasi pada Kategori *Hardware/Environmental Error* di Expert Choice 11**

Setelah nilai rata-rata geometris tersebut dimasukkan ke dalam *software* Expert Choice 11, didapatkan nilai eigen untuk setiap faktor pada kategori *Hardware/Environmental Error* sebagai berikut:



**Gambar 4.33 Nilai Eigen dan Konsistensi Faktor Penyebab Kegagalan Aplikasi pada Kategori *Hardware/Environmental Error***

Untuk mendapatkan hasil sesuai dengan urutan potensinya, maka dipilih *Sort by priority* pada software Expert Choice 11 tersebut, sehingga menampilkan hasil sebagai berikut:



**Gambar 4.34 Nilai Eigen Faktor Penyebab Kegagalan Aplikasi pada Kategori *Hardware/Environmental Error* (*Sort by priority*)**

Gambar 4.34 di atas menampilkan nilai eigen untuk setiap faktor penyebab kegagalan aplikasi terhadap kategori *Hardware/Environmental Error* dengan

nilai konsistensi sebesar 0.04, maka hasil ini sudah sesuai dengan aturan AHP di mana CR (*Consistency Ratio*) harus lebih kecil dari 10% atau 0.1.

* + - 1. **Kategori *Fault Recovery Error***

Nilai rata-rata geometris yang telah dibulatkan kemudian dimasukkan ke matriks perbandingan berpasangan. Bila nilai rata-rata geometris tersebut dibulatkan dari nilai *reciprocal*, maka di matriks perbandingan berpasangan nilainya menjadi 1/hasil pembulatan. Matriks perbandingan berpasangan untuk setiap faktor pada kategori *Fault Recovery* dapat dilihat pada Tabel 4. 18.

## Tabel 4.18 Pembulatan Rata-Rata Geometris Faktor Penyebab Kegagalan Aplikasi pada Kategori *Fault Recovery*

|  |  |  |
| --- | --- | --- |
|  | A4.1 | A4.2 |
| A4.1 |  | 2 |
| A4.2 |  |  |

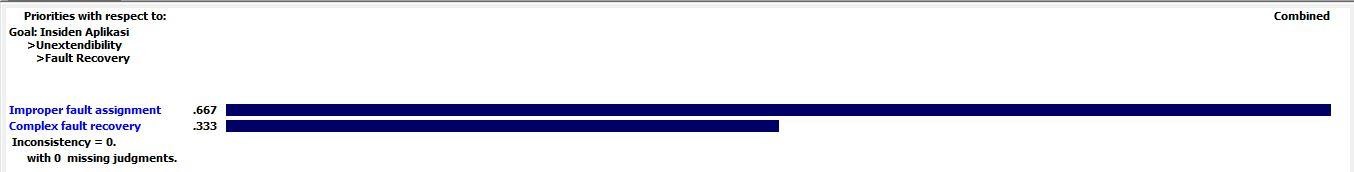
Nilai rata-rata geometris tersebut kemudian dimasukkan ke *software* Expert Choice 11 untuk kemudian dapat diolah. Tampilan matriks nilai rata-rata geometris pada Expert Choice 11 dapat dilihat pada Gambar 4.35.



## Gambar 4.35Tabel Penilaian Faktor Penyebab Kegagalan Aplikasi pada Kategori *Fault Recovery*

**di Expert Choice 11**

Setelah nilai rata-rata geometris tersebut dimasukkan ke dalam *software* Expert Choice 11, didapatkan nilai eigen untuk setiap faktor pada kategori faktor *Fault Recovery* sebagai berikut:



## Gambar 4.36 Nilai Eigen dan Konsistensi Faktor Penyebab Kegagalan Aplikasi pada Kategori *Fault Recovery*

Gambar 4.36 di atas menampilkan nilai eigen untuk setiap faktor penyebab kegagalan aplikasi terhadap kategori *Fault Recovery* dengan nilai konsistensi sebesar 0.00, maka hasil ini sudah sesuai dengan aturan AHP di mana CR (*Consistency Ratio*) harus lebih kecil dari 10% atau 0.1.

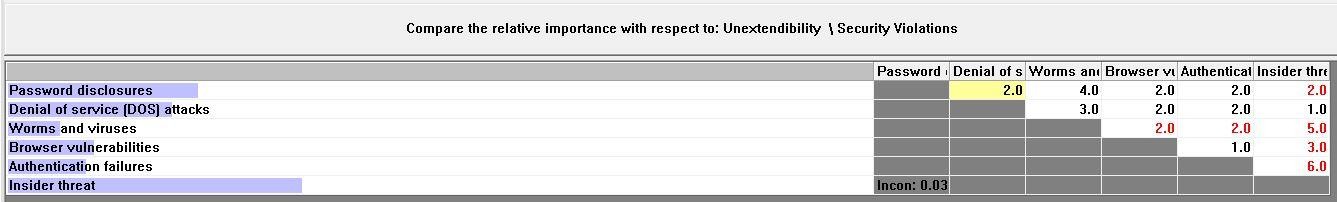
* + - 1. **Kategori *Security Violations***

Nilai rata-rata geometris yang telah dibulatkan kemudian dimasukkan ke matriks perbandingan berpasangan. Bila nilai rata-rata geometris tersebut dibulatkan dari nilai *reciprocal*, maka di matriks perbandingan berpasangan nilainya menjadi 1/hasil pembulatan. Matriks perbandingan berpasangan untuk setiap faktor pada kategori *Security Violations* dapat dilihat pada Tabel 4. 19.

## Tabel 4.19 Pembulatan Rata-Rata Geometris Faktor Penyebab Kegagalan Aplikasi pada Kategori *Security Violations*

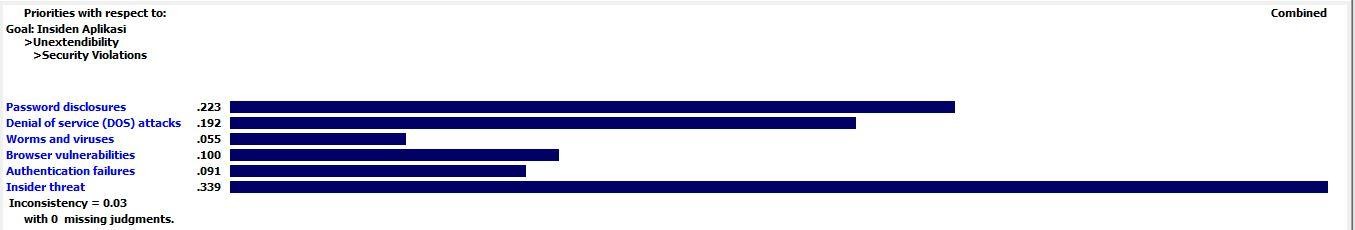
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | A5.1 | A5.2 | A5.3 | A5.4 | A5.5 | A5.6 |
| A5.1 |  | 2 | 4 | 2 | 2 | 1/2 |
| A5.2 |  |  | 3 | 2 | 2 | 1 |
| A5.3 |  |  |  | 1/2 | 1/2 | 1/5 |
| A5.4 |  |  |  |  | 1 | 1/3 |
| A5.5 |  |  |  |  |  | 1/6 |
| A5.6 |  |  |  |  |  |  |

Nilai rata-rata geometris tersebut kemudian dimasukkan ke *software* Expert Choice 11 untuk kemudian dapat diolah. Tampilan matriks nilai rata-rata geometris pada Expert Choice 11 dapat dilihat pada Gambar 4.37.



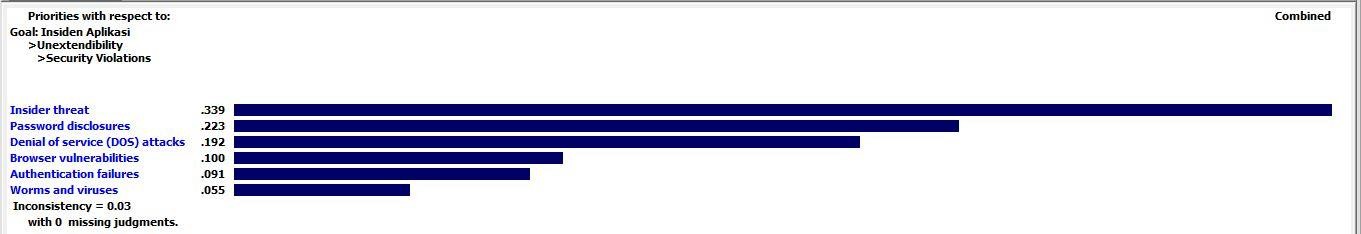
## Gambar 4.37Tabel Penilaian Faktor Penyebab Kegagalan Aplikasi pada Kategori *Seurity Violations* di Expert Choice 11

Setelah nilai rata-rata geometris tersebut dimasukkan ke dalam *software* Expert Choice 11, didapatkan nilai eigen untuk setiap faktor pada kategori faktor *Security Violations* sebagai berikut:



## Gambar 4.38 Nilai Eigen dan Konsistensi Faktor Penyebab Kegagalan Aplikasi pada Kategori *Security Violations*

Untuk mendapatkan hasil sesuai dengan urutan potensinya, maka dipilih *Sort by priority* pada software Expert Choice 11 tersebut, sehingga menampilkan hasil sebagai berikut:



**Gambar 4.39 Nilai Eigen Faktor Penyebab Kegagalan Aplikasi pada Kategori *Security Violations* (*Sort by priority*)**

Gambar 4.39 di atas menampilkan nilai eigen untuk setiap faktor penyebab kegagalan aplikasi terhadap kategori *Security Violations* dengan nilai konsistensi sebesar 0.03, maka hasil ini sudah sesuai dengan aturan AHP di mana CR (*Consistency Ratio*) harus lebih kecil dari 10% atau 0.1.

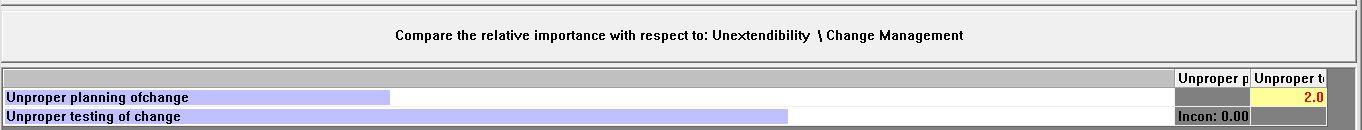
* + - 1. **Kategori *Change Management Issue***

Nilai rata-rata geometris yang telah dibulatkan kemudian dimasukkan ke matriks perbandingan berpasangan. Bila nilai rata-rata geometris tersebut dibulatkan dari nilai *reciprocal*, maka di matriks perbandingan berpasangan nilainya menjadi 1/hasil pembulatan. Matriks perbandingan berpasangan untuk setiap faktor pada kategori *Change Management* dapat dilihat pada Tabel 4. 20.

## Tabel 4.20 Pembulatan Rata-Rata Geometris Faktor Penyebab Kegagalan Aplikasi pada Kategori *Change Management*

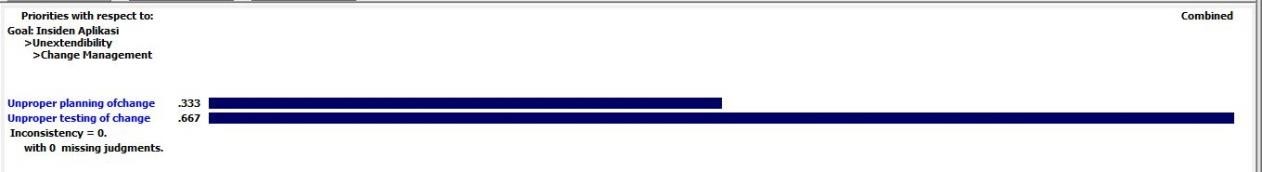
|  |  |  |
| --- | --- | --- |
|  | A6.1 | A6.2 |
| A6.1 |  | 1/2 |
| A6.2 |  |  |

Nilai rata-rata geometris tersebut kemudian dimasukkan ke *software* Expert Choice 11 untuk kemudian dapat diolah. Tampilan matriks nilai rata-rata geometris pada Expert Choice 11 dapat dilihat pada Gambar 4.40.



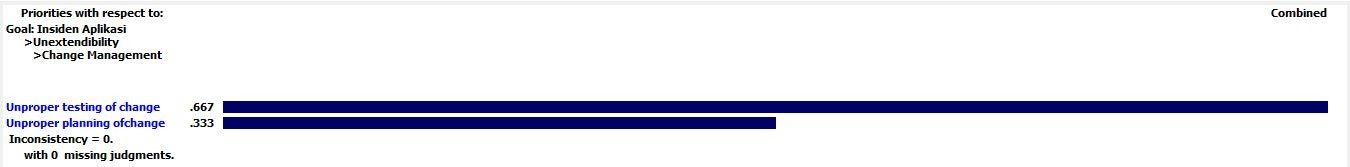
## Gambar 4.40Tabel Penilaian Faktor Penyebab Kegagalan Aplikasi pada Kategori *Change Management* di Expert Choice 11

Setelah nilai rata-rata geometris tersebut dimasukkan ke dalam *software* Expert Choice 11, didapatkan nilai eigen untuk setiap faktor pada kategori faktor *Change Management* sebagai berikut:



## Gambar 4.41 Nilai Eigen dan Konsistensi Faktor Penyebab Kegagalan Aplikasi pada Kategori *Change Management*

Untuk mendapatkan hasil sesuai dengan urutan potensinya, maka dipilih *Sort by priority* pada software Expert Choice 11 tersebut, sehingga menampilkan hasil sebagai berikut:



**Gambar 4.42 Nilai Eigen Faktor Penyebab Kegagalan Aplikasi pada Kategori *Change Management* (*Sort by priority*)**

Gambar 4.42 di atas menampilkan nilai eigen untuk setiap faktor penyebab kegagalan aplikasi terhadap kategori *Change Management* dengan nilai konsistensi sebesar 0.00, maka hasil ini sudah sesuai dengan aturan AHP di mana CR (*Consistency Ratio*) harus lebih kecil dari 10% atau 0.1.

* + - 1. **Kategori *Communication Management Issue***

Nilai rata-rata geometris yang telah dibulatkan kemudian dimasukkan ke matriks perbandingan berpasangan. Bila nilai rata-rata geometris tersebut dibulatkan dari nilai *reciprocal*, maka di matriks perbandingan berpasangan nilainya menjadi 1/hasil pembulatan. Matriks perbandingan berpasangan untuk setiap faktor pada kategori *Communication Management* dapat dilihat pada Tabel 4. 21.

**Tabel 4.21 Pembulatan Rata-Rata Geometris Faktor Penyebab Kegagalan Aplikasi pada Kategori *Communication Management***

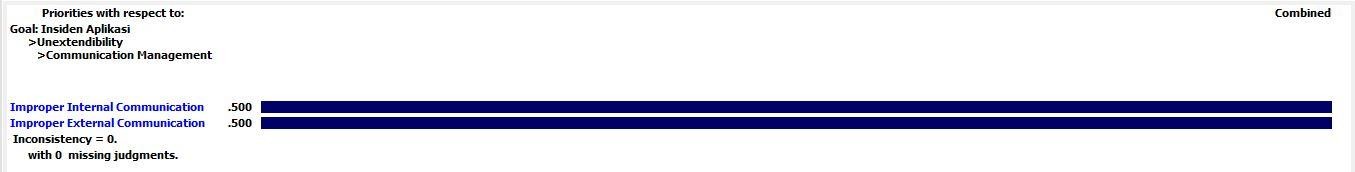
|  |  |  |
| --- | --- | --- |
|  | A7.1 | A7.2 |
| A7.1 |  | 1 |
| A7.2 |  |  |

Nilai rata-rata geometris tersebut kemudian dimasukkan ke *software* Expert Choice 11 untuk kemudian dapat diolah. Tampilan matriks nilai rata-rata geometris pada Expert Choice 11 dapat dilihat pada Gambar 4.43.



**Gambar 4.43Tabel Penilaian Faktor Penyebab Kegagalan Aplikasi pada Kategori *Communication Management* di Expert Choice 11**

Setelah nilai rata-rata geometris tersebut dimasukkan ke dalam *software* Expert Choice 11, didapatkan nilai eigen untuk setiap faktor pada kategori faktor *Software Error* sebagai berikut:



## Gambar 4.44 Nilai Eigen dan Konsistensi Faktor Penyebab Kegagalan Aplikasi pada Kategori *Software Error*

Gambar 4.44 di atas menampilkan nilai eigen untuk setiap kategori faktor penyebab kegagalan aplikasi terhadap kriteria *Unfunctionality* dengan nilai konsistensi sebesar 0.00, maka hasil ini sudah sesuai dengan aturan AHP di mana CR (*Consistency Ratio*) harus lebih kecil dari 10% atau 0.1.

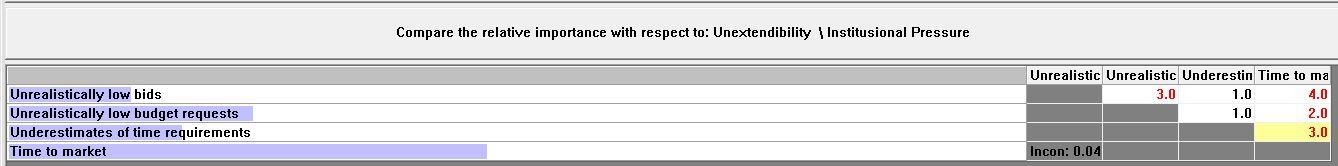
* + - 1. **Kategori *Institusional Pressure***

Nilai rata-rata geometris yang telah dibulatkan kemudian dimasukkan ke matriks perbandingan berpasangan. Bila nilai rata-rata geometris tersebut dibulatkan dari nilai *reciprocal*, maka di matriks perbandingan berpasangan nilainya menjadi 1/hasil pembulatan. Matriks perbandingan berpasangan untuk setiap faktor pada kategori *Institusional Pressure* dapat dilihat pada Tabel 4. 22.

## Tabel 4.22 Pembulatan Rata-Rata Geometris Faktor Penyebab Kegagalan Aplikasi pada Kategori *Institusional Pressure*

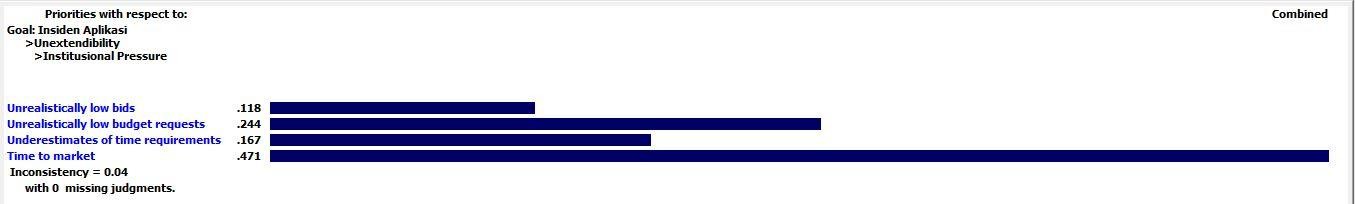
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A8.1 | A8.2 | A8.3 | A8.4 |
| A8.1 |  | 1/3 | 1 | 1/4 |
| A8.2 |  |  | 1 | 1/2 |
| A8.3 |  |  |  | 1/3 |
| A8.4 |  |  |  |  |

Nilai rata-rata geometris tersebut kemudian dimasukkan ke *software* Expert Choice 11 untuk kemudian dapat diolah. Tampilan matriks nilai rata-rata geometris pada Expert Choice 11 dapat dilihat pada Gambar 4.45.



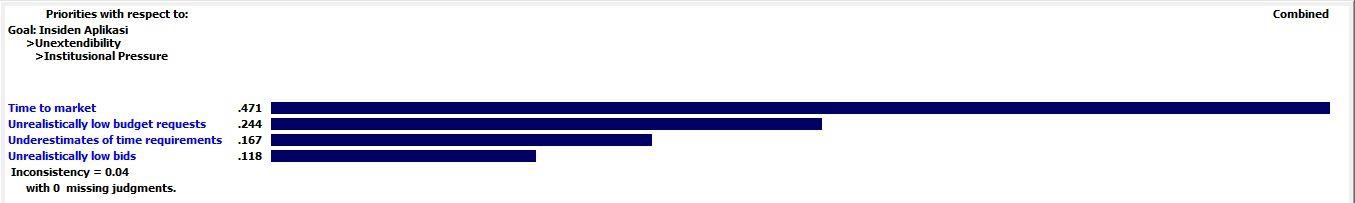
## Gambar 4.45Tabel Penilaian Faktor Penyebab Kegagalan Aplikasi pada Kategori *Institusional Pressure* di Expert Choice 11

Setelah nilai rata-rata geometris tersebut dimasukkan ke dalam *software* Expert Choice 11, didapatkan nilai eigen untuk setiap faktor pada kategori faktor *Institusional Pressure* sebagai berikut:



## Gambar 4.46 Nilai Eigen dan Konsistensi Faktor Penyebab Kegagalan Aplikasi pada Kategori *Institusional Pressure*

Untuk mendapatkan hasil sesuai dengan urutan potensinya, maka dipilih *Sort by priority* pada software Expert Choice 11 tersebut, sehingga menampilkan hasil sebagai berikut:



**Gambar 4.47 Nilai Eigen Faktor Penyebab Kegagalan Aplikasi pada Kategori *Institusional Pressure* (*Sort by priority*)**

Gambar 4.49 di atas menampilkan nilai eigen untuk setiap faktor penyebab kegagalan aplikasi terhadap kategori *Institusional Pressure* dengan nilai konsistensi sebesar 0.04, maka hasil ini sudah sesuai dengan aturan AHP di mana CR (*Consistency Ratio*) harus lebih kecil dari 10% atau 0.1.

* 1. **Analisis Prioritisasi Faktor Penyebab Kegagalan Aplikasi Aplikasi** Berdasarkan hasil analisis perhitungan masing-masing nilai eigen pada setiap level AHP yang telah dilakukan, didapatkan urutan prioritas untuk setiap level dilihat dari potensi atau frekuensi terjadinya di PT Telkomsel. Setelah mendapatkan hasil prioritisasi faktor penyebab kegagalan aplikasi, hasil tersebut akan dibandingkan dengan hasil penelitian lain. Hasil ini kemudian akan diperkuat juga dengan pendapat dari pakar.

## Hasil Urutan Berdasarkan Nilai Eigen

Hasil urutan berdasarkan nilai eigen ini akan dikelompokkan menjadi 4 jenis, yaitu urutan kriteria, urutan kategori faktor terhadap masing-masing kriteria, urutan faktor pada setiap kategori, dan prioritas faktor terhadap kegagalan aplikasi secara keseluruhan.

## Prioritas Kriteria

Berikut merupakan hasil urutan kriteria aplikasi yang berpotensi atau sering terjadi di PT Telkomsel berdasarkan hasil pengolahan data kuisioner dari 6 responden yang telah diolah dengan software Expert Choice 11

## Tabel 4.23 Urutan Kriteria Kegagalan Aplikasi

|  |  |  |
| --- | --- | --- |
| **Kriteria** | **Nilai eigen** | **Peringkat** |
| *Unextendibility* | 0.080 | 5.5 |
| *Inflexibility* | 0.038 | 7 |
| *Unfunctionality* | 0.134 | 4 |
| *Unusability* | 0.080 | 5.5 |
| *Untestability* | 0.147 | 3 |
| *Unreliability* | 0.235 | 2 |
| *Unintegrity* | 0.287 | 1 |

Dari Tabel 4.23 di atas terlihat bahwa kriteria aplikasi yang dapat menyebabkan kegagalan aplikasi adalah saat data pada aplikasi tersebut tidak dijaga dan dapat diubah oleh pihak yang tidak berwenang (*Unintegrity*) dengan nilai eigen sebesar

0.287. Hal ini didukung oleh pendapat pakar ketiga yang mengatakan, “Karena *integrity* tidak hanya orang, tapi data juga butuh *integrity*, karena nanti potensinya ada *fraud*, *misssleading information*. Proses jalan tapi masukin data algoritmanya beda antara satu sama lainnya” (Lampiran 5).

Peringkat kedua diduduki oleh kriteria *Unreliability* dengan nilai eigen sebesar

0.235 di mana nilai ini tidak jauh berbeda dengan *Unintegrity*. Kriteria *reliability* merupakan kriteria yang digunakan oleh banyak *software quality model.* Berdasarkan penelitian yang dilakukan oleh Youness, Abdelaziz, dan Habib (2013), dari 9 *software quality model* yang dibandingkan, kriteria *reliablity* digunakan oleh 8 model, yaitu *McCall’s Quality Model, Boehm’s Quality Model, Evans & Marciniak Model, Deutsch & Willis Quality Model, ISO 9126’s Quality Model, Dromey's Quality Model, FURPS+ Quality Model,* dan *ISO 25000 (SQuaRE) Model.* Dari penelitian tersebut dapat disimpulkan bahwa kriteria *reliability* merupakan kriteria yang sangat penting bagi kualitas aplikasi. Bila suatu aplikasi tidak memenuhi kriteria *reliability*, maka bisa saja aplikasi berpotensi mengalami kegagalan aplikasi.

Kriteria yang menduduki peringkat ketiga dan keempat memiliki nilai eigen yang tidak jauh berbeda, yaitu 0.147 dan 0.134. Peringkat ketiga diduduki oleh kriteria *untestability*, sedangkan peringkat keempat diduduki oleh kriteria *unfunctionality*. Kemudian peringkat kelima dan keenam memiliki nilai eigen yang sama yaitu

0.080, yang diduduki oleh *unusability* dan *unextendibility*. Sementara peringkat terakhir adalah kriteria *inflexibility* dengan nilai eigen sebesar 0.038.

## Prioritas Kategori Faktor pada Setiap Kriteria

Prioritas kategori faktor dihitung untuk menentukan tingkat potensi kategori faktor penyebab kegagalan aplikasi terhadap masing-masing kriteria. Urutan kategori faktor penyebab kegagalan aplikasi pada masing-masing kriteria dapat dilihat pada Tabel berikut:

## Tabel 4.24 Urutan Kategori Faktor terhadap setiap Kriteria

|  |  |  |  |
| --- | --- | --- | --- |
| **Kriteria** | **Kategori Faktor** | **Nilai Eigen** | **Peringkat** |
| *Unextendibility* | *Software* | 0.182 | 2 |
| *Human/Operator* | 0.089 | 7 |
| *Hardware/Environmental* | 0.241 | 1 |
| *Fault Recovery* | 0.082 | 8 |
| *Security* | 0.090 | 6 |
| *Change Management* | 0.109 | 4 |
| *Communication*  *Management* | 0.091 | 5 |
| *Institusional Pressure* | 0.117 | 3 |
| *Inflexibility* | *Software* | 0.167 | 2 |
| *Human/Operator* | 0.071 | 7 |
| *Hardware/Environmental* | 0.146 | 4 |
| *Fault Recovery* | 0.061 | 8 |
| *Security* | 0.082 | 6 |
| *Change Management* | 0.140 | 5 |
| *Communication*  *Management* | 0.169 | 1 |
| *Institusional Pressure* | 0.162 | 3 |
| *Unfunctionality* | *Software* | 0.191 | 2 |
| *Human/Operator* | 0.119 | 5 |
| *Hardware/Environmental* | 0.124 | 4 |
| *Fault Recovery* | 0.097 | 6 |
| *Security* | 0.128 | 3 |
| *Change Management* | 0.215 | 1 |
| *Communication*  *Management* | 0.071 | 7 |
| *Institusional Pressure* | 0.056 | 8 |

**Tabel 4.24 Urutan Kategori Faktor terhadap setiap Kriteria (sambungan)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Kriteria** | **Kategori Faktor** | **Nilai Eigen** | **Peringkat** |
| *Unusability* | *Software* | 0.247 | 1 |
| *Human/Operator* | 0.172 | 3 |
| *Hardware/Environmental* | 0.069 | 6 |
| *Fault Recovery* | 0.103 | 4 |
| *Security* | 0.092 | 5 |
| *Change Management* | 0.211 | 2 |
| *Communication*  *Management* | 0.065 | 7 |
| *Institusional Pressure* | 0.042 | 8 |
| *Untestability* | *Software* | 0.163 | 3 |
| *Human/Operator* | 0.186 | 1 |
| *Hardware/Environmental* | 0.170 | 2 |
| *Fault Recovery* | 0.059 | 8 |
| *Security* | 0.125 | 5 |
| *Change Management* | 0.146 | 4 |
| *Communication*  *Management* | 0.078 | 6 |
| *Institusional Pressure* | 0.074 | 7 |
| *Unreliability* | *Software* | 0.269 | 1 |
| *Human/Operator* | 0.078 | 6 |
| *Hardware/Environmental* | 0.176 | 2 |
| *Fault Recovery* | 0.095 | 5 |
| *Security* | 0.157 | 3 |
| *Change Management* | 0.145 | 4 |
| *Communication*  *Management* | 0.047 | 7 |
| *Institusional Pressure* | 0.033 | 8 |
| *Unintegrity* | *Software* | 0.150 | 3 |
| *Human/Operator* | 0.205 | 2 |
| *Hardware/Environmental* | 0.049 | 7 |
| *Fault Recovery* | 0.074 | 5 |
| *Security* | 0.303 | 1 |
| *Change Management* | 0.132 | 4 |
| *Communication*  *Management* | 0.051 | 6 |
| *Institusional Pressure* | 0.037 | 8 |

Untuk kriteria *unextendibility*, *hardware/environmental* menjadi kategori faktor yang paling berpotensi menyebabkan kegagalan aplikasi dengan nilai eigen sebesar 0.241. Untuk kriteria *inflexibility*, *communication management* menjadi kategori faktor paling berpotensi menyebabkan kegagalan aplikasi dengan nilai eigen sebesar 0.169. Sementara untuk kriteria *unfunctionality*, kategori faktor

*change management* menduduki peringkat pertama dengan nilai eigen 0.215. Kemudian untuk kriteria *unusability* dan *unreliability*, peringkat pertama diduduki oleh kategori faktor *software* dengan nilai eigen masing-masing adalah 0.186 dan

0.269. Untuk kriteria *untestability*, kategori *human/operator* menjadi peringkat pertama dengan nilai eigen sebesar 0.186. Terakhir, untuk kriteria *uintegrity*, kategori *security* menjadi kategori paling berpotensi menyebabkan kegagalan aplikasi dengan nilai eigen sebesar 0.303.

## Prioritas Faktor Penyebab Kegagalan Aplikasi Pada Setiap Kategori

Urutan tingkat potensi setiap faktor pada masing-masing kategori faktor penyebab kegagalan aplikasi dapat dilihat pada Tabel 4.25

## Tabel 4.25 Urutan Faktor Penyebab Kegagalan Aplikasi pada Setiap Kategori

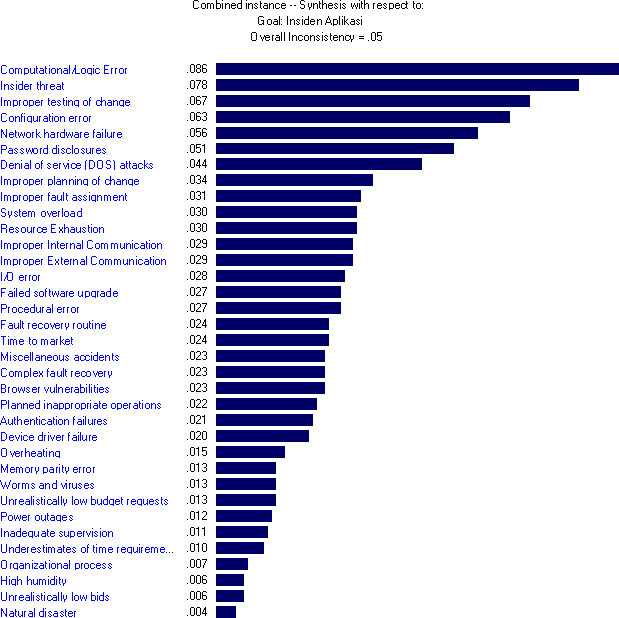
|  |  |  |  |
| --- | --- | --- | --- |
| **Kategori Faktor** | **Faktor** | **Nilai Eigen** | **Peringkat** |
| *Software Error* | *System Overload* | 0.151 | 2 |
| *Computational/Logic*  *Error* | 0.440 | 1 |
| *Resource Exhaustion* | 0.149 | 3 |
| *Failed Software*  *Upgrade* | 0.137 | 4 |
| *Fault Recovery*  *Routine* | 0.123 | 5 |
| *Human/Operator* | *Configuration Error* | 0.413 | 1 |
| *Procedural Error* | 0.174 | 2 |
| *Miscellaneous*  *Accidents* | 0.151 | 3 |
| *Organizational*  *Process* | 0.045 | 6 |
| *Inadequate*  *Supervision* | 0.074 | 5 |
| *Planned Inappropriate*  *Operations* | 0.143 | 4 |

**Tabel 4.25 Urutan Faktor Penyebab Kegagalan Aplikasi pada Setiap Kategori (sambungan)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Kategori Faktor** | Faktor | Nilai Eigen | Peringkat |
| *Hardware/Environmental* | *Device Driver*  *Failure* | 0.133 | 3 |
| *I/O Error* | 0.182 | 2 |
| *Memory Parity*  *Error* | 0.082 | 5 |
| *Network Hardware*  *Failure* | 0.363 | 1 |
| *Power Outages* | 0.078 | 6 |
| *Overheating* | 0.095 | 4 |
| *High Humidity* | 0.041 | 7 |
| *Natural Disaster* | 0.026 | 8 |
| *Fault Recovery* | *Improper Fault*  *Assignment* | 0.667 | 1 |
| *Complex Fault*  *Recovery* | 0.333 | 2 |
| *Security Violations* | *Password Disclosure* | 0.223 | 2 |
| *Denial of Services*  *(DOS) Attacks* | 0.192 | 3 |
| *Worms and Viruses* | 0.055 | 6 |
| *Browser*  *Vulnerabilities* | 0.100 | 4 |
| *Authentication*  *Failures* | 0.091 | 5 |
| *Insider Threat* | 0.339 | 1 |
| *Change Management* | *Improper Planning*  *of Changes* | 0.333 | 2 |
| *Improper Testing of*  *Changes* | 0.667 | 1 |
| *Communication Management* | *Improper Internal*  *Communication* | 0.500 | 1.5 |
| *Improper External*  *Communciations* | 0.500 | 1.5 |
| *Institusional Pressure* | *Unrealistically Low*  *Bids* | 0.118 | 4 |
| *Unrealistically Low*  *Budget Requests* | 0.244 | 2 |
| *Underestimates of*  *Time Requirement* | 0.167 | 3 |
| *Time to Market* | 0.471 | 1 |

* + - 1. **Analisis Prioritas Faktor Penyebab Kegagalan Aplikasi**

Prioritas potensi faktor penyebab kegagalan aplikasi secara global dapat dilihat pada gambar beirkut:



## Gambar 4.48 Peringkat Faktor Penyebab Kegagalan Aplikasi Secara Global

Berdasarkan Gambar 4.48 di atas, berikut merupakan urutan peringkat faktor secara global:

## Tabel 4.26 Urutan Faktor Penyebab Kegagalan Aplikasi secara Global

|  |  |  |  |
| --- | --- | --- | --- |
| **Faktor** | **Nilai Eigen** | **Kategori** | **Peringkat** |
| ***Computational /Logic***  ***Error*** | 0.086 | *Software Error* | 1 |
| ***Insider Threat*** | 0.078 | *Security Violations* | 2 |
| ***Improper Testing of Change*** | 0.067 | *Change Management* | 3 |
| ***Configuration Error*** | 0.063 | *Human/Operator Error* | 4 |
| ***Network Hardware***  ***Failure*** | 0.056 | *Hardware/Environmental*  *Error* | 5 |
| ***Password Disclosure*** | 0.051 | *Security Violations* | 6 |
| ***Denial of Services Attack*** | 0.044 | *Security Violations* | 7 |
| ***Improper Planning of Change*** | 0.034 | *Change Management* | 8 |
| ***Improper Fault Assignment*** | 0.031 | *Fault Recovery* | 9 |
| ***System Overload*** | 0.030 | *Software Error* | 10 |
| ***Resource Exhaustion*** | 0.030 | *Software Error* | 11 |
| ***Improper Internal Communication*** | 0.029 | *Communciation*  *Management* | 12 |
| ***Improper External***  ***Communication*** | 0.029 | *Communciation*  *Management* | 13 |
| ***I/O Error*** | 0.028 | *Hardware/Environmental*  *Error* | 14 |
| ***Failed Software Upgrade*** | 0.027 | *Software Error* | 15 |
| ***Procedural Error*** | 0.027 | *Human/Operator Error* | 16 |
| ***Fault Recovery Routine*** | 0.024 | *Software Error* | 17 |
| ***Time to Market*** | 0.024 | *Institusional Presssure* | 18 |
| ***Miscellaneous Accidents*** | 0.023 | *Human/Operator Error* | 19 |
| ***Complex Fault Recovery*** | 0.023 | *Fault Recovery* | 20 |
| ***Browser Vulnerabilities*** | 0.023 | *Security Violations* | 21 |
| ***Planned Inappropriate Operations*** | 0.022 | *Human/Operator Error* | 22 |
| ***Authentication Failure*** | 0.021 | *Security Violations* | 23 |
| ***Device Driver Failure*** | 0.020 | *Hardware/Environmental*  *Error* | 24 |
| ***Overheating*** | 0.015 | *Hardware/Environmental*  *Error* | 25 |

**Tabel 4.26 Urutan Faktor Penyebab Kegagalan Aplikasi secara Global (sambungan)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Faktor** | **Nilai Eigen** | **Kategori** | **Peringkat** |
| ***Memory Parity Error*** | 0.013 | *Hardware/Environmental*  *Error* | 26 |
| ***Worms and Viruses*** | 0.013 | *Security Violations* | 27 |
| ***Unrealistically low budget requests*** | 0.013 | *Institusional Pressure* | 28 |
| ***Power Outages*** | 0.012 | *Hardware/Environmental*  *Error* | 29 |
| ***Inadequate Supervision*** | 0.011 | *Human/Operator Error* | 30 |
| ***Underestimates of time***  ***requirement*** | 0.010 | *Institusional Pressure* | 31 |
| ***Organizational Process*** | 0.007 | *Human/Operator Error* | 32 |
| ***High Humidity*** | 0.006 | *Hardware/Environmental*  *Error* | 33 |
| ***Unrealistically low bids*** | 0.006 | *Institusional Pressure* | 34 |
| ***Natural Disaster*** | 0.004 | *Hardware/Environmental*  *Error* | 35 |

Berdasarkan pola yang didapatkan pada Gambar 4.48 di atas, terlihat ada 5 kluster dari faktor penyebab kegagalan aplikasi tersebut.

1. Kluster 1 : Peringkat 1 sampai peringkat 7
2. Kluster 2 : Peringkat 8 sampai peringkat 16
3. Kluster 3 : Peringkat 17 sampai peringkat 24
4. Kluster 4 : Peringkat 25 sampai peringkat 31
5. Kluster 5 : Peringkat 32 sampai pperingkat 35

Dari kluster-kluster tersebut, akan dianalisis kluster pertama, yaitu faktor-faktor yang menduduki peringkat 1 sampai peringkat 7 yang nantinya akan menjadi fokus pertama bagi PT Telkomsel. Dari 7 faktor tersebut, 3 diantaranya termasuk kategori faktor *Security Violations,* yaitu *Insider Threat, Password Disclosure,* dan *Denial of Services Attack..* Kemudian, 1 faktor termasuk kategori *Software* Error, yaitu *Computaional/Logic Error*, 1 faktor termasuk kategori *Change Management*, yaitu *Improper Testing of Change,* 1 faktor masuk kategori *Human/Operator Error,* yaitu *Configuration error*, dan 1 faktor termasuk kategori *Hardware/Environmental Error*, yaitu *Network Hardware Failure*.

*Computational/Logic Error* yang merupakan faktor dari kategori *Software Error* menduduki peringkat pertama dengan nilai eigen sebesar 0.086, sementara *Software Error* sendiri menjadi kategori faktor yang memiliki nilai eigen tertinggi untuk dua kriteria yaitu *Unusability* dan *Unreliability* yang merupakan peringkat ke-2 dan ke-5.5. Setelah dikonfirmasi kepada pakar mengenai hasil dari pemeringkatan ini, ternyata hal ini sejalan dengan pendapat Pakar 5 yang mengatakan:

“Karena kita di IT kan memang banyak *application based* ya servisnya di IT. Seiring dengan kompleksitas produk kita yang bertambah terus kompleksitas integrasi antara satu aplikasi dengan aplikasi lainnya makin ditambah pasti bakal banyak *bugs* yang ditemukan. Karena lebih ke implementasinya sih,karena misalkan satu produk kan kita banyaknya produk kita beli ya. Pasti kita akan kustomisasi sesuai dengan kebutuhan Telkomsel, dan kebutuhan Telkomsel itu mengikuti kebutuhan bisnis, nah itu sih sebenernya yang sering menyebabkan insiden. Misal bisnis maunya A, kita punya produk X, sementara untuk memenuhi kebutuhan bisnis, produk ini harus dikustomisasi. Nah itu yang sering menyebabkan *bugs*, terutama yang terkait integrasi antar aplikasi” (Lampiran 11).

*Computational/Logic Error* yang terjadi di PT Telkomsel rata-rata 1 kali dalam 1 minggu (Lampiran 5). Kasus yang pernah terjadi di PT Telkomsel salah satunya adalah adanya *instance* dari salah satu aplikasi di PT Telkomsel yaitu CRM mengalami *hang* dan menyebabkan *stuck order*, namun menurut Pakar 5 hal ini masih menjadi *issue* di mana *issue* merupakan insiden yang minor sesuai dengan pernyataan Pakar 5, “Sebenernya di kita itu *issue* insiden juga sih, tapi minor insiden. Kalau insiden yang bener-bener di-*broadcast* itu *major incident*” (Lampiran 11). Adapun hal ini sejalan dengan penelitian yang dilakukan oleh Pertet dan Narasimhan (2005) yang menyatakan bahwa *Software Error* dan *Human Error* menjadi penyebab 80% kegagalan.

Peringkat kedua diduduki oleh *Insider Threat* dengan nilai 0.078, kemudian peringkat keenam diduduki oleh *Password Disclosure* dengan nilai 0.051, dan peringkat ketujuh diduduki oleh *Denial of Services Attcak* dengan nilai 0.044. Ke

tiga faktor tersebut merupakan faktor dari kategori *Security Violation* yang memiliki nilai eigen terbesar dari seluruh kategori faktor untuk setiap kriteria, yaitu 0.303. Adapun kategori *Security Violation* ini menjadi kategori paling krusial untuk kriteria *Unintegrity* yang merupakan kriteria aplikasi yang paling berpotensi atau paling sering menyebabkan kegagalan aplikasi dengan nilai eigen sebesar 0.287.

Menurut penelitian yang dilakukan oleh Munshi, Dell, dan Armstrong (2012), ancaman yang timbul dari orang dalam organisasi terus menjadi salah satu isu utama yang harus dihadapi oleh organisasi dan institusi pemerintah di semua sektor kritikal infrastruktur. Dampak dari sumber daya internal merupakan salah satu masalah serius yang dihadapi oleh organisasi, dan sangat sulit diatasi (Chinchani, Iyer, Ngo, dan Upadhyaya, 2005). Hal ini sejalan dengan pendapat Pakar 5 yang mengatakan bahwa:

“Memang banyak terjadi kasus *fraud*, baik dari luar atau pun dari dalam, ya memang *security* di kita juga gak bisa dibilang, ya maksudnya gini lah semakin canggih *security* di kita semakin banyak juga cara membobolnya. Memang *security threat* ini banyak banget sih. Lumayan banyak lah kasus di Telkomsel semenjak aku ada di Telkomsel. gitu, aku nemuin waktu aku di *Problem*. Beberapa kayak ada orang yang masuk ke server kita lah atau yang ngubah- ngubah *config* kita” (Lampiran 11).

Menurut Pakar 4, *Security Violations* sendiri terjadi minimal 1 bulan sekali di PT Telkomsel (Lampiran 10). Selanjutnya bahwa dampak yang terjadi dari *insider threat* dan DOS memang sangat besar didukung pula oleh pernyataan Pakar 5,

“Kalau *security violations* kayak *insider threat*, atau DOS ini iya sebulan paling sekali tapi dampak nya besar, even dia cuma sekali. Mungkin dalam waktu 6 bulan cuman dua kali misalkan tapi impact nya besar. Yang banyak kecil-kecil itu masalah config.” (Lampiran 11).

Salah satu contoh kasus *Security Violations* yang pernah terjadi di PT Telkomsel adalah kasus Mkios. Berdasarkan pernyataan Pakar 4, akun seorang *reseller* PT

Telkomsel tiba-tiba bisa melakukan transaksi sampai 1M. seperti yang diungkapkan oleh Pakar 4 berikut:

“Oh..Misalnya kayak kemaren, kasus dari MKios lah ya. Tiba-tiba akunnya tinggi banget, dia bisa transaksi sampai 1 M. Dia bisa beli pulsa sampe 1 M. Ternyata yang ngelakuin salah satu vendor di sini. Nah itu sebelumnya kita selidiki, ini bener gak nih transaksinya, terus kita cek 3 bulan ke belakang ritmenya dia. Ritme pembeliannya dia. Kalau misalkan ini ter-compromise misalkan dia ini ada kecurigaan dari hasil penyelidikan ya, kita udah kuat gitu dari segi ininya kita blok nomor itu.” (Lampiran 10).

IT Security PT Telkomsel sendiri telah melakukan tindakan preventif, forensik, dan detektif untuk mengatasi masalah keamanan di PT Telkomsel (Lampiran 10). Pengawasan yang dilakukan oleh tim IT Security adalah 24 jam dengan *alert conditioning* (Lampiran 10). Tindakan yang dilakukan saat tim IT Security mendapatkan *alert* terdiri dari 2 level, sebagaimana dijelaskan oleh Pakar 4:

“Nah, biasanya ketika ada *alert* di tempat kita, nah nanti kita punya tim semacam FBI, ya tim investigasinya, itu ada di level 1, Level 1 *investigation*. Nah itu mereka yang akan turun, misalkan ada perubahan di aplikasi apa, misalnya Reflex atau apa. Biasanya terjun ke orang Reflexnya, didatengin” (Lampiran10)

“Jadi kalau di IT Security ada 24x7 *monitoring* itu level 1, tugasnya memberikan analisa *alert-alert* yg masuk, kemudian *raise* ke *custody*, setelah *raise* ke *custody* dia dapet konfirmasi terus dia melakukan *summary* juga analisa di tingkat awal ini insiden apa bukan. Kalau itu insiden, maka itu nanti Level 2 yang akan turun. Level 2 itu misalkan ada brute force gitu di dalam aplikasi kita, biasanya mereka itu akan melakukan *reset password*-nya” (Lampiran 10)

Adapun konfigurasi dari IT Security memiliki 4 domain servis yang mendukung 24x7 monitoring, yaitu Incident *Monitoring Services, Security Assessment, Security Development*, dan *User Access Management*. Berikut merupakan pernyataan dari Pakar 4 terkait hal tersebut:

“Kalo konfigurasi dari IT Security itu kan dari 24x7 itu kan ada 4 domain servis lagi yang men-*support* 24x7 itu. Yang pertama itu ada *Incident Monitoring Services,* yang kedua itu adalah *Security Assessment*, itu yang melakukan VA dan *hardening*. Gunannya itu kalau VA mencari celah aplikasi, terus yang *hardening* itu ibaratnya menutup segala kemungkinan akses ke suatu aplikasi. Terus yang ketiga itu di dalam itu ada *Security Development*, biasanya ini biro namanya itu biro *developer* untuk mengembangkan *alert- alert* baru dari kalo serangan itu kan dia itu ter*compromise*-nya datanya tuh beda-beda ya, kalau ada serangan-serangan baru, nah *developer* ini tugasnya nangkep, untuk meng-*create* lagi *alert-alert* baru. Nah yang ke empat ada *User Access Management*, kita melakukan yang namanya *mapping user access* ke semuanya ke sistem telkomsel itu harus dari *User Access Management*. Itu aja kalau contoh kasusnya banyak, tapi gak bisa di*share*.” (Lampiran 10)

Peringkat ketiga untuk faktor yang paling berpotensi atau sering terjadi di PT Telkomsel adalah *Improper testing of change*. Faktor ini masuk ke kategori *change management* dengan nilai eigen sebesar 0.067. *Change management* sendiri merupakan kategori faktor yang paling krusial untuk kriteria *Unfunctionality* yang menduduki peringkat ke empat sebagai kriteria aplikasi yang dapat menyebabkan insiden.

Berdasarkan hasil dari penelitian yang dilakukan oleh Pertet dan Naramsimhan (2005), insiden yang disebabkan oleh *software failure* pada aplikasi web mengindikasikan jumlah yang besar terhadap banyak kegagalan terjadi selama *change management.* Hal ini didukung juga oleh hasil dari penelitian yang dilakukan Saaralainen dan Jantti (2015) yang menemukan bahwa 86% (18 dari 21 *root cause*) dari insiden terkait dengan *change management.*

*Improper testing of change* yang terjadi di PT Telkomsel kebanyakan disebabkan karena *Standard Changes* yang dilakukan oleh pihak internal tanpa melibatkan tim *testing* (Lampiran 11). Untuk *Normal Changes*, faktor yang menyebabkan sering terjadinya *improper testing* adalah dari segi *environment testing* seperti pernyataan Pakar 5 berikut:

“Kalau *changes* itu yang pasti kalau misalkan ada perubahan, OK di aplikasi A dilakukan perubahan, nah aplikasi A ini berhubungan dengan aplikasi B nah kita suka *miss* bahwa ternyata *impact* juga ke aplikasi B gitu. Bahwa *impact-* nya ke *surrounding-*nya juga. Nah mungkin di situ, kalau di sisi *testing*-nya kadang-kadang bisa jadi *miss*. Jadi misalnya kita *scope testing-*nya harus jelas kan kalo ada perubahan itu, misalkan aplikasi *surrounding*-nya apa aja, ketika *surrounding-*nya banyak itu akan lebih memunculkan kemungkinan bahwa akan ada *test case* yang *miss* di situ, belum lagi kalau untuk *point testing-*nya kalau yang aku lihat sih pengadaan *resource testing-*nya kayak *environment* itu nggak ideal lah, jauh dari ideal lah kalau di Telkomsel. Kayak kita gak punya *testing environment* yang sesuai dengan *production*, kita gak semua aplikasi punya gitu. Jangankan yang sesuai dengan *production*, yang aplikasi yang punya *testing environment* aja gak semuanya kan. Nah itu sih sebenernya. Salah satu poin yang besar juga terkait environment.” (Lampiran 11)

Adapun kategori *testing* di PT Telkomsel terdiri dari 3 kategori, yaitu *Standard Changes*, *Normal Changes*, dan *Emergency Changes* seperti yang dikemukakan oleh Pakar 1:

“Kalau di telkomsel ada *Standard Changes*, *Normal Changes*, dan *Emergency Changes*. *Standard Changes* merupakan *testing* yang dilakukan oleh tower. *Standard Changes* ini otorisasinya tower karena tidak melalui *rule testing* kita. Kalau *Normal Changes* ini lah yang menjadi tanggung jawab kita. Untuk yang *Emergency Changes* tidak dilakukan tes pun bisa *release* asal siapa yg bertanggung jawab atas resiko yg terjadi, dalam hal ini *Management*. Jadi ini sebenarnya merupakan resiko, resiko ini di-*handle* oleh *managemet*. *Emergency Changes* bisa mem-*bypass* seluruh proses atau prosedur yang terjadi contohnya dokumennya nanti saja, testing tidak perlu karena ada unsur yang mengikuti biasanya unsur kecepatan produksinya atau berhubungan dengan bisnis tertentu yang notabene jangan sampai kalah sama kompetitor jadi *management* bilang *release* aja dulu, atau berhubungan dengan insiden, kalau ada insiden kan kita benerin dulu bisa jadi *changes* juga kalau *changes*-

nya ada di infra, berhubungan dengan *changes* yang 3 kategori tadi” (Lampiran 11).

*Configuration Error* menduduki peringkat keempat dengan nilai eigen tidak terlalu jauh berbeda dengan *improper testing*, yaitu 0.063. *Configuration error* merupakan faktor dari kategori *Human/Operator Error* yang menjadi kategori paling krusial untuk kriteria *Untestability*. Terlihat bahwa proses testing di kategori *change management* berkaitan dengan *Human/Operator Error*. Adapun penelitian dari Pertet dan Naramsimhan (20005) menyatakan bahwa *Software Failure* dan *Human Error* menyebabkan 80% kegagalan.

*Configuration Error* yang terjadi di Telkomsel baru-baru ini adalah kasus konfigurasi *session* pada aplikasi Reflex, seperti pernyataan dari Pakar 5 berikut:

“Kalau yang *config* itu kecil-kecil sih biasanya, cuman contohnya ada sih cuman yang ini gak kecil waktu kemarin. Ada konfigurasi *timeout*, err bukan *timeout* juga sih, *session* gitu lah antara *load balancer* sama reflex aplikasi. Na itu efeknya ya banyak transaksi yang gagal karena *session*-nya udah *full*. Jadi karena konfigurasi jumlah *session*-nya yang gak *match* gitu kan” (Lampiran 11)

Pakar 5 menyatakan bahwa penyebab dari seringnya *Configuration Error* ini terjadi adalah karena sering terjadi perubahan setelah mengimplementasikan suatu analisis konfigurasi:

“Kalau *configuration* itu misalkan gini, kenapa itu sering terjadi soalnya yang tadi misal ada integrasi A dan B. Nah itu pasti konfigurasinya harus *match* kan antara A dan B, misal yang *simple* aja, konfigurasi *timeout*. *Timeout* A dan B kalau dia saling terhubung kan. Terus ada lagi aplikasi di belakangnya ada C,

D. Konfigurasinya harus *match* kan, gak boleh misalkan aplikasi depannya ini *timeout-*nya lebih kecil daripada yang di belakang itu gak bisa atau dan kita harus tau *behavior* tiap aplikasi untuk *set up* konfugurasi yang pas. Nah di Telkomsel ini masih sering apa ya bukan *trial* dan *error* tapi lebih sering, kita punya analisis terus kita *apply*, ternyata ini kurang pas nih, terus kita ubah lagi. Kalau konfigurasi itu gitu sih” (Lampiran 11)

Peringkat kelima yaitu *Network Hardware Failure* dengan nilai eigen sebesar

0.056 merupakan faktor di kategori *Hardware/Environmental Error* yang menjadi faktor paling krusial untuk kriteria *Unextendibility*. Kriteria *Unextendibility* sendiri menduduki peringkat ke 5.5 sebagai kriteria aplikasi yang dapat menyebabkan kegagalan aplikasi. Contoh kegagalan *hardware* yang sering terjadi di PT Telkomsel yaitu FO Cut dengan berbagai penyebab dan beberapa tidak diketahui penyebabnya, seperti pernyataan Pakar 5, “Contohnya kalau *network* itu misalkan FO cut. Fiber Optik. Fisiknya putus. Dulu sering jaman 2014-2015. 2015 paling sering. Sekarang udah jarang”, dan “Macem-macem, kadang-kadang, karena putus sendiri, karena kan ada yang di dalem laut kan, ada yang di dalem tanah atau kecangkul. Dulu itu sering banget” (Lampiran 11).

Adapula contoh kasus lain terkait *network hardware* di PT Telkomsel, yaitu kerusakan *Port* yang penyebabnya juga belum diketahui, sesuai pernyataan Pakar 5 berikut, “Ada lagi masalah *port*, misalkan *port* di *router* atau di *switch* itu kan ada *port*-nya, *port*-nya bermasalah, *port*-nya rusak akhirnya kita ganti *port* aja. Nah itu kan kita juga gak tau penyebabnya apa, ya mungkin karena, ya *hardware problem* lah itu” (Lampiran 11).

Perangkat *Network Hardware* di PT Telkomsel merupakan milik internal Telkomsel, selain Fiber Optik yang merupakan milik PT Telkom. Perangkat tersebut dikelola oleh vendor di bawah pengawasan pegawai PT Telkomsel (Lampiran 11). Terkait dengan hal ini, Pakar 5 memberikan pandangannya terhadap faktor *managed service* untuk insiden yang sering terjadi terkait *Network Hardware*, “Kalau menurut ku sih nggak ya, karena si vendor pun *under* supervisinya orang Telkomsel juga sih. Kalau menurut ku memang ke perangkatnya. Mungkin ada masalah apa gitu, ya kerusakan perangkat” (Lampiran 11).

Menurut Pakar 5 kegagalan tersebut tidak ada kaitannya dengan adanya pihak vendor yang mengelola perangkat tersebut. Hal itu murni terjadi karena kerusakan perangkat network. Adapun untuk mencegah kerusakan perangkat network yang ada di Telkomsel, Pakar 5 menambahkan bahwa ada mekanisme *hardware upgrade* tergantung kebutuhan, rekomendasi, dan teknologi:

“Iya, biasanya kalau umurnya udah berapa tahun gitu ya bisa di-*upgrade*. Persisnya berapa tahun tergantung produknya, ada setahun, ada yang lima tahun. Tergantung kebutuhan, tergantung rekomendasi dari yang punya produknya, tergantung teknologi juga, kalau misalkan ada teknologi baru yang Telkomsel rasa perlu, diganti.” (Lampiran 11).

## Implikasi

Penelitian ini memiliki implikasi pada dua area, yaitu secara teori dan praktikal. Implikasi secara praktikal berdampak pada 3 Divisi utama di bawah Departemen IT Service Management PT Telkomsel, yaitu IT Service Quality Management (IT SQM), IT Data Center Management (IT DCM), dan IT Infrastructure Management (IT IM).

## Implikasi Teori

Berdasarkan hasil analisis kriteria dan faktor penyebab kegagalan aplikasi yang telah dijabarkan sebelumnya, faktor yang paling berpotensi menyebabkan kegagalan aplikasi, yaitu *Computational/Logic Error* yang termasuk pada kategori *Software Error.* Hal ini sejalan dengan hasil penelitian yang dilakukan oleh Pertet dan Naramsimhan (2005) yang menyebutkan bahwa 80% kegagalan aplikasi dikarenakan oleh kerusakan pada aplikasi itu sendiri. Terkait hal ini, dapat dilakukan penelitian lebih lanjut dengan investigasi empiris terhadap frekuensi terjadinya *bugs*, serta apa yang dapat dilakukan untuk mencegah dan memperbaiki *bugs* tersebut. Hal ini sejalan dengan *future work* dari penelitian yang dilakukan oleh Asadollah, Hansson, dan Sundmark (2015).

Sedangkan untuk kriteria yang paling berpotensi menyebabkan kegagala pada aplikasi adalah kriteria *Unintegrity*, yaitu ketidakmampuan sebuah aplikasi dalam menjaga datanya untuk tidak diubah oleh pihak yang tidak berwenang. Hal ini sejalan dengan hasil pemeringkatan faktor penyebab kegagalan aplikasi di mana dari 7 faktor teratas pada kluster 1 terdiri dari faktor-faktor yang berasal dari kategori *Security Violations*, yaitu *insider threat, password disclosure,* dan *denial of services attack.* Terkait faktor *Insider* Threat yang menjadi faktor kedua penyebab kegagalan aplikasi, hal ini sejalan dengan penelitian yang dilakukan

oleh Chinchani, Iyer, Ngo, dan Upadhyaya (2005) yang menyatakan bahwa dampak dari sumber daya internal merupakan salah satu masalah serius yang dihadapi oleh organisasi, dan sangat sulit diatasi.

Penelitian ini juga berimplikasi pada *software quality model* di industri telekomunikasi. Sebelumnya, Hamid dan Hasan (2011) telah mendefinisikan *software quality factor* untuk industri telekomunikasi di Malaysia. Model ini dapat digunakan baik untuk mengukur kualitas aplikasi secara kualitatif maupun kuantitatif.

## Implikasi Praktikal

Untuk Divisi IT SQM, hasil penelitian ini dapat berkontribusi bagi strategi untuk tetap menjaga kualitas layanan yang diberikan pada konsumen. Dari hasil urutan faktor penyebab kegagalan aplikasi secara global, faktor *Computational/Logic Error* merupakan faktor yang paling berpotensi menyebabkan kegagalan aplikasi, kemudian pada peringkat ketiga terdapat *Improper Testing of Change*, dan disusul oleh *Configuration Error*. Tiga faktor ini erat kaitannya dengan masing-masing bagian di bawah IT SQM. Berkaca dari hal tersebut Manager dari System Integration Test Management (SITM) dapat merumuskan strategi untuk menghadapi faktor-faktor tersebut. Karena kebanyakan kegagalan aplikasi terjadi saat melakukan integrasi antar aplikasi, maka sebaiknya proses *testing* dari aplikasi tersebut benar-benar dapat mengakomodasi perubahan tersebut. Adapun terkait *Standard Change* yang tidak masuk ke aturan *testing* di SITM, sebaiknya tetap terdapat pengawasan dari pihak SITM selama *testing,* karena risiko tetap dipegang oleh SITM.

Dari hasil pemeringkatan faktor penyebab kegagalan aplikasi secara global, terlihat bahwa *Insider Threat* dari kategori *Security Violations* menduduki peringkat kedua dengan dampak kerugian paling besar (Lampiran 11). Untuk itu, tim IT Security Management yang berada di bawah divisi IT Data Center Management harus tetap terus menjaga dan meningkatkan pengawasan dan responsif terhadap alarm yang masuk. Adapun strategi agar tidak terjadi penyalahgunaan otoritas, disarankan untuk mengadakan *mandatory vacation* yang

dapat mengurangi risiko tersebut. Dengan strategi tersebut, diharapkan sumber daya internal tidak melakukan hal yang tidak seharusnya dilakukan karena secara periodik akan digantikan oleh orang lain.

Terkait faktor Network Hardware Failure yang menjadi peringkat kelima penyebab kegagalan aplikasi di PT Telkomsel, hal ini berdampak pada tim Data Communication Management yang mengelola network hardware di PT Telkomsel. Karena ada banyaknya kegagalan yang tidak diketahui sebabnya, hal ini dapat menjadi masukan bagi Divisi IT Infrastructure Management untuk melakukan analisa root cause terkait hal tersebut.

## BAB 5 KESIMPULAN dan SARAN

Bab ini terdiri dari dua bagian, yaitu kesimpulan dan saran. Kesimpulan akan menjawab *Research Question* yang telah didefinisikan pada Bab 1, selanjutnya saran akan diberikan untuk pengembangan lebih lanjut dari penelitian ini.

## Kesimpulan

Berdasarkan analisis yang telah dilakukan pada penelitian ini, maka kesimpulannya adalah:

1. Lima faktor yang paling berpotensi atau sering terjadi di PT Telkomsel adalah:
   1. *Computational/Logic Error*, (2) *Insider Threat*, (3) *Improper Testing of Change*, (4) *Configuration Error*, (5) *Network Hardware Failure. Computational/Logic Error* sering terjadi di PT Telkomsel dikarenakan kompleksitas produk yang terus bertambah, maka integrasi antar aplikasi sering dilakukan. Di situlah banyak terjadi *bugs* yang menjadi minor insiden bagi PT Telkomsel.
2. Kategori faktor yang paling berpotensi di setiap kriteria aplikasi berbeda-beda. Untuk kriteria *Unextendibility*, kategori faktor yang paling krusial adalah yang terkait dengan *Hardware and Environmental Error*. Untuk kriteria *Inflexibility*, kategori faktor *Communication Management* menjadi kategori yang paling berpengaruh. Untuk kriteria *Unfunctionality*, kategori faktor *Change Management* menjadi kategori paling krusial. Sementara untuk kriteria *Unusability* dan *Unreliability*, kategori *Software Error* menduduki peringkat pertama. Selanjutnya untuk kriteria *Untestability,* kategori faktor *Human/Operator Error* menjadi kategori yang paling berpengaruh. Yang terakhir untuk kriteria *Unintegrity*, peringkat pertama diduduki oleh kategori *Security Violations*
3. Kriteria aplikasi yang paling berpotensi menyebabkan kegagalan aplikasi adalah kriteria *Unintegrity*

## Saran

Adapun beberapa saran dari penelitian ini yang dapat digunakan untuk pengembangan lebih lanjut yaitu:

1. Penelitian selanjutnya dapat menggunakan metode selain wawancara, misalnya FGD agar dapat mencapai kesepakatan dan tingkat kepuasan yang sama dengan semua pakar yang terlibat. Hal ini juga disarankan oleh Kasayu (2015).
2. Terkait dengan faktor yang paling berpotensi menyebabkan kegagalan aplikasi, yaitu *Computational/Logic Error*, penelitian lebih lanjut dapat dilakukan investigasi empiris terhadap frekuensi terjadinya *bugs*, serta apa yang dapat dilakukan untuk mencegah dan memperbaiki *bugs* tersebut. Hal ini sejalan dengan *future work* dari penelitian yang dilakukan oleh Asadollah, Hansson, dan Sundmark (2015).
3. Terkait kriteria yang paling berpotensi menyebabkan kegagalan aplikasi yaitu *Unintegrity* yang kaitannya erat dengan faktor-faktor dari kategori *Security Violations*, khususnya untuk faktor *Insider Threat* yang menjadi faktor kedua yang berpotensi menyebabkan kegagalan aplikasi, dan memiliki dampak kehilangan paling besar bagi perusahaan, disarankan untuk dilakukan penelitian selanjutnya untuk mengembangkan model konseptual holistik yang dapat mengenkapsulasi perspektif faktor yang lebih luas terkait situasi sumber daya internal yang lebih mencerminkan pengalaman secara empiris, seperti faktor karakter, sosial, teknikal, dan faktor organisasi. Penelitian selanjutnya ini juga disarankan oleh Munshi, Dell, dan Armstrong (2012).
4. Penelitian selanjutnya terkait penentuan karakteristik *Software Quality Model* khusus di sektor industri tertentu juga dapat dilakukan dengan metode AHP, karena memilih salah satu model kualitas merupakan tantangan bagi suatu organisasi, serta peran *user* juga memiliki dampak langsung ke *Software Quality* (Youness, Abdelaziz, & Habib, 2013)

## DAFTAR PUSTAKA

Asadollah, S. A., Hansson, H., & Sundmark, D. (2015). Towards classification of concurrency bugs based on observable properties. *2015 IEEE/ACM 1st International Workshop on Complex faUlts and Failures in LargE Software Systems*, 41-47.

Boehm, B., & Basili, V. R. (2001). Software defect reduction top 10 list. *Software Management*, 135-137.

Chinchani, R., Iyer, A., Ngo, H. Q., & Upadhyaya, S. (2005). Towards a theory of insider threat assessment. *IEEE International Conference*.

Creswell, J. W. (2002). *Research design : qualitative, quantitative, and mixed methods approaches.* SAGE Publications.

Dalal, S., & Chhillar, D. S. (2012). Case studies of most common and severe types of software system failure. *International Journal of Advanced Research in Computer Science and Software Engineering*, 341-347.

Eldh, S., Sasikumar, P., & Hansson, H. (2007). Component testing is not enough - a study of ssoftware faults in telecom middleware. *Testing of Software and Comunication Systems, Springer*, 74-89.

Galin, D. (2004). *Software quality assurance from theory to implementation.*

England: Addison Wesley.

Guo, W., & Wang, Y. (2009). An incident management model for SaaS application in the IT organization. *International Conference on Research Challenges in Computer Science*, 137-140.

Hamid, N. F., & Hasan, M. K. (2011). Identifying software quality factors for telecommunication industry in Malaysia. *International Conference on Electrical Engineering and Informatics*.

Ic, Y. (2012). An experimental design approach using TOPSIS method for the selection of computer-integrated. *Robotics and Computer-Integrated Manufacturing,*, 245-256.

IEEE Standard - Adoption of ISO/IEC 20000-1:2011, Information Technology - Service Management- Part 1: Service Management System Requirements. (2013). *IEEE*, 1-24.

Jamwal, D. D. (2010). Analysis of Software Quality Models for Organizations.

*International Journal of Latest Trends in Computing*, 19-23.

Johnson, M., Hately, A., Miller, B. A., & Orr, R. (2007). Evolving standards for IT service management. *IBM Systems Journal*, 583-597.

Kasayu, R. (2015). *Analisis faktor-faktor yang paling berpengaruh terhadap kesuksesan proyek pengembangan perangkat lunak di Indonesia dengan metode analytic hierarchy process (AHP).* Jakarta: Magister Teknologi Informasi Universitas Indonesia.

Lewis, W. E. (2004). *Software Testing and Continuous Quality Improvement.*

Auerbach Publications.

Maizlish, B., & Handler, R. (2005). *IT (information technology) portfolio management step-by-step: unlocking the business value of technology.* Wiley.

Marimin. (2004). *Teknik dan aplikasi pengambilan keputusan kriteria majemuk.*

Jakarta.

McMilan, J., & Schumacher, S. (2003). *Research in education.* New York: Longman.

Mishra, P., Dangayach, G. S., & Mittal, M. L. (2011). An empirical study on identification of critical success factors in project based organizations.

*Global Business and Management Research; An International Journal*, 356-368.

Munshi, A., Dell, P., & Armstrong, H. (2012). Insider threat behavior factors: a comparison of theory with reported. *Hawaii International Conference on System Sciences*, 2402-2411.

Olaperi, Y. S., & Misra, S. (2015). An empirical evaluation of software quality assurance practices and challenges in a developing country. *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications;Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, 867-871.

P, B., R, F., & eds. (2014). *Guide to the software engineering body of knowledge, version 3.0.*

Pertet, S., & Narasimhan, P. (2005). Causes of failure in web application.

*Carnegie Mellon University Research Showcase*.

Pinto, J. (2010). *Project management achieving competitive advantage second edition.* Pearson.

Podvezko, V. (2011). The comparative analysis of MCDA methods SAW and COPRAS. *Inzinerine-Ekonomika-Engineering Economics*, 134-146.

Qin, X. S., Huang, G. H., & Chakma, A. (2008). A MCDM-based expert system for climate-change impact assessment and adaptation planning – A case study for the Georgia Basin, Canada. *Expert Systems with Applications*, 2164-2179.

Saaralainen, K., & Jantti, M. (2015). Quality and human errors in IT service infrastructure - human error based root cause of incidents and their organization. *International Conference on Innovations in Information technology (IIT)*, 207-212.

Saaty, T. L. (1993). *Pengambilan keputusan bagi para pemimpin, proses hirarki analitik untuk pengambilan keputusan dalam situasi yang kompleks.* Pustaka Binama Pressindo.

Saaty, T. L. (2012). *Decision making for leaders.* Pittsburgh: RWS Publications.

Subiyakto, A., & Ahlan, b. A. (2013). A coherent framework for understanding critical success factors of ICT project environment. *3rd International Conference on Research and Innovation in Information Systems*, 342-347.

Systems and software engineering vocabulary. (2010). *ISO/IEC/IEEE 24765:2010(E)*, 1-418.

The Standish Group. (2015). *Chaos Report.*

Thor, J., Ding, S. H., & Kamaruddin, S. (2013). Comparison of multi criteria decision making methods from the maintenance alternative selection perspective. *The International Journal Of Engineering And Science (IJES)*, 27-34.

Turban, E., Rainer, K. R., & Potter, R. E. (2004). *Introduction to information technology.* Wiley.

Velasquez, M., & Hester, P. T. (2013). An Analysis of Multi Criteria Decision Making Methods. *International Journal of Operations Research* , 56-66.

Ward, J., & Peppard, J. (2002). *Strategic Planning for Information Systems.*

England: John Wiley & Sons,Ltd.

Yoon, M. s., Lee, B., & Whang, K. S. (2011). Modeling software quality evaluation using the analytical hierarchy process. *Proceedings of the international symposium on the analytical hierarchy process*.

Youness, B., Abdelaziz, M., & Habib, B. (2013). Comparative Study of Software Quality Models . *IJCSI International Journal of Computer Science Issues* , 309-314.

Zhong, L., Bin, L. C., & Cong, Z. S. (2008). Using analytic hierarchy process to evaluate software quality characteristics of smartphone. *International Conference on Advanced Information Technologies (AIT)*.

## LAMPIRAN

**Lampiran 1 Service Level Agreement (SLA) PT. Telkomsel**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kategori SLA** | **Business Service** | **Application** | **Target** | **Downtime Target**  **(minutes)** |
|  |  | - OCS |  |  |
|  | Customer | - Cookies |  |  |
|  | Invoice | - SPR |  |  |
|  |  | - OSDSS |  |  |
| Mission Critical | Payment | * OCS * CRM | 99,9 % | 4,32 minutes |
|  | -SASN |
|  | Data Package Usage | * SAPC * UPCC * DSP |  |  |
|  | Customer | - Telkomsel | 99,9 % | 43,2 minutes |
|  | Loyalty & | Poin |
|  | Community |  |
|  | Customer | -Cookies |
|  | Interface | - CRM-CIM |
|  | Management | - CIS |
| Business |  | - SMSC |
| Critical |  | * SMS Gateway * USSD |
|  | Mobile Banking Management | Gateway  - USSD Service Gateway |
|  |  | - VAS MG - |
|  |  | UMB |
|  |  | - I-Charming |
|  |  | - DSP |
|  |  | - OCS |
|  |  | - NBP |  |  |
|  | Digital Service | - VAS MG - |  |  |
| Business Improvement | Catalog | UMB  - Customer | 99% | 432 minutes |
| Application | - Selena |
|  | Development | - Remedy |  |  |
|  | Management |  |  |  |
|  | Compliance |  |  |  |
|  | Management | - AMS |  |  |
| Business  Support | Knowledge | * Remedy * Knowledge | 90% | 4320 minutes |
|  | Management | Management |  |  |
|  |  | System |  |  |

**Lampiran 2 Daftar Insiden Tahun 2015**

**te**

**h**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **application** | **incident\_da** | **Start Time** | **End Time** | **Mont** | **Year** | **disruption\_time** | **disruption\_time(in hours)** |
| OCS | 3-Jan-15 | 9:30 | 12:30 | Jan | 2015 | 180 | 3.00 |
| Datacomm | 14-Jan-15 | 15:45 | 17:25 | Jan | 2015 | 100 | 1.40 |
| INGW | 2-Feb-15 | 10:15 | 10:55 | Feb | 2015 | 40 | 0.40 |
| Reflex | 11-Feb-15 | 14:30 | 15:23 | Feb | 2015 | 53 | 0.53 |
| DSP | 23-Feb-15 | 15:30 | 16:10 | Feb | 2015 | 40 | 0.40 |
| Link Telin-RIM | 26-Feb-15 | 0:00 | 4:00 | Feb | 2015 | 235 | 3.55 |
| OCS | 26-Feb-15 | 19:30 | 20:20 | Feb | 2015 | 50 | 0.50 |
| Datacomm | 27-Feb-15 | 8:40 | 9:10 | Feb | 2015 | 30 | 0.30 |
| OSB - CRM | 28-Mar-15 | 20:00 | 1:00 | Mar | 2015 | 300 | 5.00 |
| INGW | 2-Apr-15 | 20:00 | 21:30 | Apr | 2015 | 90 | 1.30 |
| INGW | 3-Apr-15 | 19:30 | 22:25 | Apr | 2015 | 175 | 2.55 |
| INGW | 4-Apr-15 | 20:15 | 20:55 | Apr | 2015 | 40 | 0.40 |
| OMS - CRM | 23-Apr-15 | 12:30 | 13:45 | Apr | 2015 | 75 | 1.15 |
| T-Menu | 11-May-15 | 19:30 | 21:35 | May | 2015 | 125 | 2.05 |
| Datacomm | 13-May-15 | 10:20 | 10:50 | May | 2015 | 30 | 0.30 |
| USSD | 13-May-15 | 20:50 | 21:30 | May | 2015 | 40 | 0.40 |
| T-Menu | 16-May-15 | 19:00 | 19:30 | May | 2015 | 30 | 0.30 |
| Telkomsel Poin | 17-May-15 | 14:20 | 18:20 | May | 2015 | 240 | 4.00 |
| OSB - CRM | 28-May-15 | 4:15 | 8:20 | May | 2015 | 245 | 4.05 |
| T-Menu | 30-May-15 | 12:30 | 14:10 | May | 2015 | 100 | 1.40 |
| Payment Gateway | 10-Jun-15 | 10:45 | 13:00 | Jun | 2015 | 135 | 2.15 |
| OCS | 12-Jun-15 | 22:00 | 22:35 | Jun | 2015 | 35 | 0.35 |
| Reflex | 18-Jun-15 | 4:20 | 5:00 | Jun | 2015 | 40 | 0.40 |
| USSD | 26-Jun-15 | 22:30 | 23:52 | Jun | 2015 | 82 | 1.22 |
| RBT | 15-Jul-15 | 21:30 | 0:30 | Jul | 2015 | 180 | 3.00 |
| OCS | 21-Aug-15 | 4:10 | 7:15 | Aug | 2015 | 185 | 3.05 |
| OCS | 25-Aug-15 | 22:00 | 23:15 | Aug | 2015 | 75 | 1.15 |
| Charging Gateway | 2-Sep-15 | 20:30 | 22:30 | Sep | 2015 | 120 | 2.00 |
| Charging Gateway | 3-Sep-15 | 19:30 | 23:30 | Sep | 2015 | 120 | 4.00 |
| OCS | 12-Sep-15 | 0:15 | 1:00 | Sep | 2015 | 45 | 0.45 |
| OCS | 12-Sep-15 | 20:40 | 21:20 | Sep | 2015 | 40 | 0.40 |
| T-Menu | 15-Sep-15 | 8:00 | 11:25 | Sep | 2015 | 205 | 3.25 |
| OCS | 17-Sep-15 | 19:10 | 19:55 | Sep | 2015 | 45 | 0.45 |
| MKIOS | 20-Sep-15 | 18:30 | 19:00 | Sep | 2015 | 30 | 0.30 |
| OCS | 23-Sep-15 | 2:00 | 2:20 | Sep | 2015 | 20 | 0.20 |
| OCS | 29-Sep-15 | 19:00 | 20:15 | Sep | 2015 | 156 | 1.15 |
| Charging Gateway | 10-Oct-15 | 20:00 | 23:10 | Oct | 2015 | 190 | 3.10 |
| URP | 12-Oct-15 | 11:40 | 13:00 | Oct | 2015 | 80 | 1.20 |
| Reflex | 19-Oct-15 | 20:00 | 21:15 | Oct | 2015 | 75 | 1.15 |
| INGW | 21-Oct-15 | 9:45 | 10:45 | Oct | 2015 | 60 | 1.00 |
| Charging Gateway | 21-Oct-15 | 18:45 | 19:40 | Oct | 2015 | 55 | 0.55 |
| INGW | 25-Nov-15 | 17:45 | 20:10 | Nov | 2015 | 145 | 2.25 |

**Lampiran 3 Transkrip Wawancara Tahap 1 (Pakar 1)**

Nama : Sjuhud Al Amien Tanggal Wawancara : 22 September 2016 Keterangan

P : Peneliti

R : Responden

P : Menurut Bapak, dari 5 kriteria, 8 faktor dan 32 subfaktor penyebab insiden yang sudah saya lampirkan yang didapat dari studi literatur. Apakah ada yang perlu ditambahkan atau dikurangi? Atau sudah sesuai dengan kondisi perusahaan telekomunikasi?

R : Sebenarnya kita itu dalam melakukan *testing* itu untuk menjaga *service quality*. Karena memang men-*develop* apapun harus ada *project control*. Hasil dari *testing* kita juga melihat bahwa target dan *achievement* itu apa, yang dilihat dari hasil *testing* bukan hanya dari jumlah *quantity* dari jumlah apa yang kita kerjakan (CR yang kita kerjakan) tapi dari *quality* apa yang kita dapat dari *changes* yang kita kerjakan tersebut. Untuk yang bisa kita lihat sebagai hasil dari kita itu sebenarnya kita harus meminumumkan *incident after release*. *Incident after release* ada banyak kriterianya . Insiden itu terjadi karena *availability* nya terganggu. *Availability* terganggu karena apa? Faktornya udah disebut di dokumen seperti *application* (*software failure*), *infrastructure (hardware/environment)*, dan *network*.

P : Berarti daftar faktor-faktor penyebab insiden yang di sini sudah sesuai ya Pak?

R : Iya, Iya sama. Terus dari sisi *human error* yang berhubungan dengan

*release*. Jadi *release* juga harus sesuai dengan prosedur P : Ini berhubungan sama vendor ya Pak?

## (lanjutan)

R : Bisa, vendor. Pokoknya *human error*. Ada lagi *Other*. *Others* itu di luar dari kita , contoh *network* tapi bukan punya kita, bisa masuk sebagai faktor. Atau yang tidak termasuk dalam *activity* ini, yaitu 4 yang saya sebutkan tadi, *application, infrastructure, network, human error*. Ada lagi bugs, configuration bisa masuk ke application. Satu lagi adalah *incident* disebabkan oleh *changes*. Sistem kalau tidak dilakukan perubahan, resiko *down* nya hanya karena dari 4 itu, tapi kalau contohnya ada hubungan nya dgn perubahan, akan terhubung dengan *release* yang berkorelasi dengan proses *testing*. Nanti di *testing* ini dibagi lagi mana yang *under* kita (SQM), mana yang *non under* kita, nanti disesuaikan dengan instansi tersebut. Kalau di telkomsel ada *Standard Changes, Normal Changes*, dan *Emergency Changes*. *Standard Changes* merupakan *testing* yang dilakukan oleh *tower*. *Standard Changes* ini otorisasi nya *tower* karena tidak melalui *rule testing* kita. Kalau *Normal Changes* ini lah yang menjadi tanggung jawab kita. Untuk yang *Emergency Changes* tidak dilakukan tes pun bisa *release* asal siapa yg bertanggung jawab atas resiko yg terjadi, dalam hal ini *Management*. Jadi ini sebenarnya merupakan resiko, resiko ini di-*handle* oleh *managemet*. ***Emergency Changes*1** bisa mem-*bypass* seluruh proses atau prosedur yang terjadi contohnya dokumen nya nanti saja, *testing* tidak perlu karena ada unsur yang mengikuti biasanya unsur kecepatan produksi nya atau berhubungan dengan bisnis tertentu yang notabene jangan sampai kalah sama *competitor* jadi *management* bilang *release* aja dulu, atau berhubungan dengan insiden, kalau ada insiden kan kita benerin dulu bisa jadi *changes* juga kalau *changes* nya ada di infra, berhubungan dengan changes yang 3 kategori tadi. Jadi apapun yang menyebabkan insiden berhubungan dengan *availability* dan yg berhubungan dengan kita adalah yang *release*. Kalau yang 4 itu, nggak diapa apain tetap terjadi insiden. Inilah yang menjadi *concern* kenapa *Software Quality Assurance* harus ada.

## (lanjutan)

P : Berarti ini masuk ke kriteria ya Pak? Availablity itu? R : Iya

P : Tadi belum ada, nanti ditambahin di kriteria ya Pak.

R : Jadi gini, sebenarnya insiden ini ada dua, ada yang berhubungan dgn *capacity* dan ada yang *availability* nya terganggu. Contohnya nggak sama antara *requirement* yang dibutuhkan dan yang di-*suppo rt* oleh kita. Akhirnya ujung-ujungnya apa? Akan ada resiko insiden. Insiden nya apa? Contoh kita gak bisa bicara kalau mesin nya mati, tetapi kalau ada kerjaan misal di modul *registarion* tiba-tiba kita tidak bisa *entry* dan nge-*save* ini berhubungan dgn *capacity*. Jadi dua ini bisa jadi penyebab insiden. Insiden itu terdiri dari dua kategori, *major* dan *minor*. Apa *major* apa *minor*? Semuanya ada resiko terhadap konsumen sewaktu kita melakukan *recovery*. Kenapa dianggap major, biasanya di atas 30 menit kita tidak bisa melakukan *recovery*. Tapi contohnya minor 0-30 menit bisa *recovery* dan ini ada hubungan nya dengan CMDB (*Configuration Management Database*), konfigurasinya apa saja. Kenapa saya bilang? Karena kalau kita bicara insiden kan terkait servis (*Hardware/Software/Brwainware*). Ini yang harus dilihat, kalau terganggu *availability* berarti *capacity* juga. berikutnya lebih dalam lagi biar besok nggak terjadi nsiden yg sama. Kalau bagian SQM berhubungan dengan *changes*.

P : Ini berarti masuk ke *changes management* ya Pak.

R : **Iya, ini *changes, testing* dan *release* nya sendiri masuk ke *Changes Management****,* **tambah *monitoring*2**. *Changes* itu perubahan terhadap CMDB. Selain *release management*, kita bicara tentang insiden ini ada kaitannya sama *problem*. *Problem* itu adalah insiden berulang, insiden yan tidak bisa *restore* . Contohnya CMDB nya banyak yang berubah.

## (lanjutan)

P : Ini penyebab insiden?

R :Bukan, ini problem bisa berbulan-bulan. Ini tim nya beda. Ini berhubungan dengan *Change Request*. Membuat *Business Request* kemudian *Change Request*. Ini apa? CMDB ini. Kalo insiden gak bisa bikin BR dan CR. Ini hanya bisa terkena *impact* nya. Tapi kalau *problem* bisa bikin BR dan CR.

P : Berarti secara umum ini sudah sesuai semua ya pak? Tambahin

*availability* ke kriteria ya Pak.

R : Semuanya sebenarnya masuk tapi ini secara servis bukan aplikasi.

Insiden itu terjadi kalau *availability* nya terganggu.

P : Berarti ini perlu ditambahin gak pak?

R : Ini semua tergabung ke *availability.* Pas ini gak usah nambah.

P : Untuk faktor *testing,* dan *release* masuk ke *change management*? S : Iya bener, *testing* udah bener masuk ke *change management*.

P : Kalau *network* yang punya vendor? Kira-kira masuk ke subfaktor mana Pak? Perlu dibedain gak pak?

R : *Network* vendor sama kayak *network* internal. Paling *request for change*.

**Tambahin *request for change* di *change management*3**

P : Kalau *request* bisa menyebabkan insiden dimana?

R : Misal saya minta perubahan 1 konten, permintaan nya tidak tepat. *Request* nya dia gak bener. Ada satu lagi ***infrastructure changes*, mempengaruhi juga4**. Minta pindah data aja, bisa menyebabkan insiden.

## Lampiran 4 Transkrip Wawancara Tahap 1 (Pakar 2)

Nama : Irawan Koesmardianto Tanggal Wawancara : 27 September 2016 Keterangan

P : Peneliti

R : Responden

P : Menurut Bapak, dari 5 kriteria, 8 faktor dan 32 subfaktor penyebab insiden yang sudah saya lampirkan yang didapat dari studi literatur. Apakah ada yang perlu ditambahkan atau dikurangi? Atau sudah sesuai dengan kondisi perusahaan telekomunikasi?

R : Bisa, bisa. Ini didapat dari mana kriterianya?

P : Dari penelitian nya Hamid dan Hasan Pak. Sesuai sama studi kasusnya, biar fokus ke telko.

R : Kenapa Hamid hasan nggak nampilin *efficiency, reliability, integrity* ya? P : Kalau perlu ditambahkan tidak apa-apa Pak.

R : **Ini perlu ditambahin ini *reliability, integrity*5** . Ini maksudnya

*manageability* gimana?

P : Kemampuan aplikasi itu untuk dikelola. R : Ada lagi yang mau ditanya?

P : Konfirmasi kesesuaian kriteria sama faktor itu aja Pak

R : **Untuk faktor *change management,* perlu ditambahin *deployment changes* nya6. Ini untuk *communication management*, ada komunikasi internal sama eksternal.7** .Internal itu ke sesama IT, kalau eksternal ke *third party*.

## Lampiran 5 Transkrip Wawancara Tahap 1 (Pakar 3)

Nama : Subagia

Tanggal Wawancara : 4 Oktober 2016 Keterangan

P : Peneliti

R : Responden

P : Menurut Bapak, dari 5 kriteria, 8 faktor dan 32 subfaktor penyebab insiden yang sudah saya lampirkan yang didapat dari studi literatur. Apakah ada yang perlu ditambahkan atau dikurangi? Atau sudah sesuai dengan kondisi perusahaan telekomunikasi? 22 tahun di Indosat. Operation

R : Secara garis besar sama ya. Untuk semua aplikasi itu sama kriteria nya. Software development itu kan ada standar. Jadi biasanya sama mau perusahaan nya IT atau tidak. Kalau saya melihat kerangka ini kan Software Development ya. Software itu sudah jadi. Kalau kita melihat dari perspektif insiden problem meet dengan kriteria mana, pengelompokkan insiden nya kan, kriteria insiden itu terjadi karena lack atau dari salah satu komponen. Biasanya kalau kita menggunakan standar Software Development Life Cycle (SDLC), memang harus ada definisinya, objektifnya, didefinisikan di Business Requirement sampai testing performance, harusnya kalau lewat semua itu harusnya minor lah isu insidennya. Kebanyakan problem yang muncul itu karena tidak mengikuti SDLC. Sebenarnya di dalam lifecycle itu di-address functionality nya, kapasitas performance nya, maksimal berapa begitu. Kemudian flexibility, usability, testability itu lifecycle sebelum dirilis itu ada itu semua.

P : Berarti ini masuk ke faktor ya, Pak?

R : Iya, kalau di dalam proses SDLC itu ada yang kurang atau tidak maksimal di salah satu yang di-bypass pasti muncul insiden nantinya. Paling banyak kejadian insiden software itu kalau gak ngikutin SDLC, kalau ngikutin

## (lanjutan)

lifecycle mostly minor lah tapi kalau nggak ngikutin ya tergantung lifecycle bagian mana yang gak diikutin, contohnya biasanya kalau yang dikejar target itu lebih ke functionality aja pokoknya functionality jalan tapi melupakan extendability, jadi sizing dari software ini sebenarnya untuk casuality design, casual basic. Casual basic itu nggak diperhitungkan kapasitas berapa jadi pasti kejadian. Atau dia lupa, dia rilis suatu software karena tidak mengikuti kaidah yang itu tadi begitu untuk level QOC atau level usernya dibawah belasan atau di bawah100 mungkin bisa, tapi kalau ribuan jutaan itu impact nya ada. Semua bisa jadi faktor itu tergantung waktu kita develop itu ada checklist nya, checklist itu diikutin nggak semuanya, kalau checklist semuanya diadopsi harusnya minimal, kita bisa prediksi sebenarnya. Kalau checklist domain-domain yang tadi ada 5 kriteria bisa 10 bisa macem-macem orang ngembangin kriterianya itu ada yang ga diadopsi di lifecyclenya nya pasti terjadi insiden. Tinggal ditunjuk aja sebenarnya inisden ini related ke misalnya kapasitas. Contohnya dia menggunakan yang gampang yang murah meriah software nya contohnya pake JBOSS, mestinya pake middleware yang lebih advance. Kalau JBOSS kan open source, gratisan, dikembangin. Kalau user nya dibawah 100 mungkin masih jalan, tapi usernya sudah hit ribuan bahkan jutaan itu bakal jadi isu besar karena pemilihan software platform yang tadi itu tidak mengakomodasi untuk level tersebut. Jadi fundamental architecture nya sudah salah. Banyak faktornya, nah tinggal masukin ke mana nih.

P : Kira-kira kalau SDLC mau masuk di sini itu jadi salah satu poin faktor baru atau bisa dimasukkan menjadi subfaktor?

R : Nggak ini kan ngomongnya insiden. Insiden di Software Development jadi fokus ke insidennya.

## (lanjutan)

P : Nggak khusus di Software Development saja, tapi untuk yang sudah dirilis Pak.

R : Iya maksudnya software yang udah di operation kan, tapi melihatnya kan kalau software di release, software ada dua, software yag custom build atau yang sudah platform, contohnya Microsoft itu kan software, Microsoft Word, Microsoft Excell ataupun yang lainnya itu kan platform kita pakai untuk di dalam perusahaan ada lagi yang custom build, CB itu adalah business prosess di indosat ditanam di dalam bentuk software misal menjadi aplikasi CRM, aplikasi billing. Kalau yang namanya paket, sudah komersil produk atau platform itu biasanya mereka merelease patch. Jadi setiap minggu mereka merelease patch misalnya tentang security, Jadi dia update patch terus supaya stabil. Itu untuk paket software. Kelemahan yang ini, banyak yg memonitor istilahnya, kalau yang opensource tapi kalau yang custom build hanya didesain, di develop di suatu perusahan itu saja, dan telko paling banyak yang dia membuat sendiri inhouse ataupun pake pertner. BP yang ada dibuat dalam bentuk software yang kita sebut aplikasi. Nah itu tergantung apakah di dev mereka punya standarisasi atau framework SDLC nggak. Kalau nggak punya pasti itu insiden nya di operation di rilis ada insiden. Tapi kalau mereka mengikuti SDLC atau ikut pemilihan paket software itu pasti minilam asal dia update secara regular baik itu patchnya, versionnya. Itu sebenarnya meminimalkan insiden. Makanya suka dirilis paket2 software. Kalau software itu sudah jadi, aplikasi itu custom disesuaikan dgn bisnis proses perusahaan. Ya sebenarnya mirip-mirip. Software yang dirilis di luar yang digunakan untuk original purpose missal Mic Word, itu ada mekanisme lifecysle nya dalam dev dan lifecycle dlm update patching, makanya kita harus ngikutin, kalau kita gak ngikutin itu misalnya 2 version belakangnya pasti sudah berantakan karena HW nya sudah baru tapi SW nya masih lama itu memicu insiden.

## (lanjutan)

P : Ini bisa masuk ke faktor ya update software?

R : Ini sebenarnya sudah OK. Saya cuman memberi tambahan bahwa untuk meminimalisasi insiden, insiden kriteria nya ini sudah OK. Kalau kita ngomongin insiden tidak hanya software di IT insiden nya ini. Kriteria nya bisa dipakai buat semuanya tidak terkhusus pada software. Detail nya ya bisa karena software failure, hardware failure, human error, macam- macam ya.

P : Kalau patch tadi termasuk di faktor yang mana ya Pak?

R : Oh itu masuk ke software upgrade. Software sudah dirilis, mungkin ditemukan dalam perjalanannya ada bug, itu kan butuh patch atau ada fitur yang harus diupdate,yang diextend dis itu, misalnya upgrade ver atau stability karna mungkin HW waktu dia kembangkan masih jadul, setelah 2 tahun dipakai softwarenya, ini HW nya sudah upgrade, saat ketemu software yang lama ini match. Nah saat itu dia butuh upgrade version. Karena isu hardware. Kalau ini OK semua sebenarnya.

P : Berarti Hardware juga perlu diupgrade? Upgrade hardware itu maksudnya hardrware nya diganti?

R : Iya, kalau hardware itu ada dua, bisa vertical upgrade atau horizontal upgrade. Kalau horizontal upgrade itu misalnya kapasitas, contohnya storage nya kurang, komputasi nya kurang, diclustering misalnya itu ke samping, jadi dari 1 ke 2 ke 3 ke 4 karna user nya mungkin bertambah dari kemarin seribu sekarang 100 ribu, gimana caranya? Ya dibuat clustering High Availability. Atau vertical upgrade lebih ke memory nya, prosesor nya ditambahin, macem2 lah. Atau memang replace, add on baru. Ini valid semua ini kalau dari perspektif insiden.

R : Jadi ini tambahin poin faktor baru yaitu upgrade HW pak?

## (lanjutan)

R : Bukan, itu solusi untuk mengatasi insiden. Kalau kita ngomong di operation ada namanya ITSM, di sini ada beberapa divisi. Saya div QA dan compliance, ada lg IT service transition (change management), kalau ada release baru dari project development masuk ke change management atau ada yang dari Operation kana da change nya. Ada lagi ITO, contract dan vendor mgmnt, itu tim yang ada di sini yang memegang IT Operation. Nah kalau yang insiden itu, incident problem management itu ada di IT Service Operation sebenarnya, di bawah ITSM. Kalau saya lebih kea rah QA nya, bagaiamana kematangan proses-proses yang ada. Kalau ini kan spot nya insiden ya. Ini OK sudah. Ini tinggal di mapping, biasanya kita punya mekanisme namanya root cause analysis, di mapp ke salah satu ini untuk mencari penyebab insiden.

P : Baik Pak, kemudian mengenai Change management Pak?

R : Change management bisa menyebabkan insiden, misalnya planning nya gak proper contohnya belum ada Businesss Impact Analysis. Stakeholder nya belum diundang , ada kontribusi potensi problem nggak kalau ada change, karna kan implikasi nya bisa ke surrounding system

P : Dari subfaktor di change management, ada yang perlu ditambahin tidak Pak?

R : Mungkin tambahan dari saya, ini kan masih pengelompokan ya, hanya planning the change. Baiknya lebih detail lagi, misal gak proper planningnya atau apa, atau tidak ada BIA begitu. Planning sih domainnya, tapi kenapa di palnning nya ? gak proper kah, atau gk ada BIA mengundang stakeholder terkait, kalau di run sekarang ada potensi apa di surrounding system nya.

P : Untuk ini memang hanya untuk domain-domain saja Pak.

## (lanjutan)

R : Saran saya sih didetailin apanya gitu, contohnya kayak subfaktor improper fault assignment itu kan jelas. **Jadi kalau di change management tadi misalnya unproper planning begitu, atau planning bagian apa yang berpotensi jadi insiden8**. Atau selama ini yang rootcause nya di change bagian apanya gitu, ini sebenarnya bisa di elaborate lagi. Karna di atas contohnya udah bener kayak DoS , worm, virus. Itu jelas itu penyebabnya. Karena ya kalau change management memang harus di planning

P : Untuk communication management bagaimana Pak?

R : Communication management terutama yang terkait kalau ada change tidak melibatkan atau share terhadap change itu gak dilakuin BIA kemudian kita melakukan upgrade di sisi kita padahal ada surrounding system yang terkait dengan aplikasi lain, kita upgrade, aplikasi tersebut masih ketinggalan, padahal output kita bisa jadi input ke sana. Itu bisa berdampak menyebabkan aplikasi di sana tidak jalan.

P : Itu berarti komunikasi ke mana ya Pak?

R : Ya itu, tidak melibatkan pihak terkait surrounding system nya tadi. Jadi kita punya aplikasi kemudian output kita jadi input ke system lain. Kita melakukan upgrade, awalnya jalan interfacing ke system lainnya. Saya melihat nya di situ

P : Kalau komunikasi terkait vendor bisa ditambahkan ke subfaktor communication management Pak?

R : Bisa, bisa jadi. Masuknya jadi vendor management juga bisa.

P : Dibuat jadi satu poin lagi atau ditambahin ke subfaktor communication management ?

R : Kalau saya melihat **communication management itu lebih ke internal9**

## (lanjutan)

P : Berarti ditambahin faktor Vendor Management ya Pak?

## R : Iya. Contohnya kualifikasi orang yang di sana begitu , qualified developer nggak, qualified engineer nggak, qualified tim nggak yang akan support kita10.

P : Berarti communication management itu internal

R : Kalau saya lihat iya lebih ke internal, mungkin leadership dari atasan nya untuk bisa nge track tim nya

P : Kalau terkait insider threat, bagaimana pendapat Bapak?

R : Insider threat itu sebenarnya ngga ada hubungan sama software. Insider threat itu kita punya aplikasi kemudian aplikasi itu dibocorin sama orang.

P : Tapi itu bisa menyebabkan insiden tidak Pak?

R : Itu lebih ke fraud sih, kalau fraud itu termasuk insiden ya bisa masuk sih. Banyak sebenarnya list nya untuk security violation ini. **Masukin aja fraud di situ11.**

P : Kalau untuk kriteria nya bagaimana Pak? Sebelumnya sudah wawancara juga dan ada pakr yang menyebutkan ada kriteria reliability, kalau menurut Bapak bagaimana?

## R : Ya masuk, reliabity sama resiliency12.

P : Sama satu lagi integrity bisa masuk ke kriteria tidak, Pak?

R : **Ya data integrity bisa13**. Karena integrity tidak hanya orang, tapi data juga butuh integrity, karena nanti potensi nya ada fraud, misssleading information. Proses jalan tapi masukin data algoritma nya beda antara satu sama lainnya

P : Kalau Resiliansi tadi bisa tolong djelaskan Pak?

## (lanjutan)

R : Daya tahan kalau terjadi insiden, kalau terjadi insiden adaptif tidak kalau terjadi insiden

P : Untuk konfirmasi, berarti tambahan dari Bapak untuk kriteria itu ada tiga reliability, resiliency, dan data integrity ya Pak.

R : Jadi desain nya sudah memenuhi itu tadi belum, baik reliability, resiliancy contohnya HW itu ada yang harus maksimal di level suhu 40 derajat celcius ada yang desain industrial PC didesain di atas 50 60 C masih berfungsi itu contohnya. Kriteria faktor-faktor tadi masuk tidak ke dalam desain, ini gak bisa di run di suatu lingkungan yang keras membuat dia mati, nge hang dll. Makanya waktu dibuat di awal ini objektif nya mau dipasang di mana. Kalau ini diperuntukkan ke suatu kondisi yang berat, itu disesuaikan, cari perangkat yang sesuai. Kalau tidak diakomodasi akan terjadi insiden

P : Sekarang ke Institusional Pressure Pak. Menurut Bapak ada kaitan nya sama vendor?

R : IP itu jadi gini, kan semuanya apapun itu butuh banyak resource dalam bentuk orang, people, process,technology, budget, kalau dipaksain misalnya dengan budget yang rendah jalan sih, tapi dicopotin kiri kanan, akhirnya gak maksimal desain nya ujung nya gagal. Orang yang nge run gak qualified, maka kita harus bisa ngitung harga nya. Ini harga yang pantas atau tidak, karena nanti partner juga deliver sesuai harga.

P : Perbedaan Low bid sama low budget request apa ya Pak kalau boleh dijelaskan?

R : Jadi harganya misal buth kendaraan tipikal mobil alphard harus baru tapi dikasi uang haya 200 juta, ada gak harga baru alhard 200 juta? minimal 1 M itu, nah itu contoh low bid. Kalaupun ada, mungkin dicopotin gagangnya.

## (lanjutan)

Kalau low budget request, saya maunya alphard, tapi karena saya gak tau komparasinya, benchmarkingnya, saya masukin budget nya aja sekian. Ini dari dalam, kalau low bid itu waktu biddingnya waktu kita procurement ke partnernya, misalnya harusnya harga projek 10 perak, harga tertingginnya 20. Di bawah 10 prak itu sudah berantakan. Tapi procurement memaksakan 5 perak, karna vendor ingin ngambil projeck tersebut, dia setuju tapi dalam deliver nya, baik orang, technology, software jug minimal akhirnya hasilnya gak sesuai. Kalau low budget req ini, waktu desain awal harusnya user nya benchmark untuk dapatin ini berapa budget yang pantas, tapi dia gak lakuin itu akhirnya dia request kerendahan. Orang finance tidak tau, yang tau kan yang meminta. Nah begitu, harusnya budgetnya 20 , 10 dia minta 5. Hasilnya sama, satu lagi kesalahan procurement, satu lagi kesalahan usernya gak paham. **Satu lagi itu time to marketnya14**, jadi gini harus rilis bulan depan saya butuh buat software, minggu depan sudah harus selesai, ini kan tidak logic. Kita kan harus memulai lifecycle tadi, requirement jelas, di desain awal, kemudiann di develop, mencari partnernnya, di testing kan butuh waktu, sampai beta tester, sampai release. Lifecycle saja butuh waktu belum develop software nya sendiri. Ini dipaksain karena time to market dari pihak marketing atau produk.

P : Ini masuk ke subfaktor Institusional pressure Pak, sama dengan underestimate of time requiremen tidak Pak?

R : Kalau itu sengaja underestimate, kalau yang ini orangnya tidak paham software development. Ini desakan pasar, kalau tidak kita tidak bisa berkompetensi di pasar. Sehingga dipaksain rilis sebelum waktunya dengan bypass proses-proses lifecycle tadi.

## Lampiran 6 Kerangka AHP pada Expert Choice 11

12/17/2015 1:02:07 PM Page 1 of 9

Model Name: AHP

Treeview

**Goal: Insiden Aplikasi Unextendibility (L: .080)**

**Software Failure (L: .182) System overload (L: .151)**

**Computational/Logic Error (L: .440) Resource Exhaustion (L: .149) Failed software upgrade (L: .137) Fault recovery routine (L: .123)**

**Human/Operator Error (L: .089) Configuration error (L: .413) Procedural error (L: .174) Miscellaneous accidents (L: .151) Organizational process (L: .045) Inadequate supervision (L: .074)**

**Planned inappropriate operations (L: .143) Hardware and Environmental Error (L: .241)**

**Device driver failure (L: .133) I/O error (L: .182)**

**Memory parity error (L: .082) Network hardware failure (L: .363) Power outages (L: .078) Overheating (L: .095)**

**High humidity (L: .041) Natural disaster (L: .026)**

**Fault Recovery (L: .082)**

**Improper fault assignment (L: .667) Complex fault recovery (L: .333)**

**Security Violations (L: .090) Password disclosures (L: .223)**

**Denial of service (DOS) attacks (L: .192) Worms and viruses (L: .055)**

rosareska

## (lanjutan)

12/17/2015 1:02:07 PM Page 2 of 9

**Browser vulnerabilities (L: .100) Authentication failures (L: .091) Insider threat (L: .339)**



**Change Management (L: .109)**

**Improper planning of change (L: .333) Improper testing of change (L: .667)**

**Communication Management (L: .091) Improper Internal Communication (L: .500) Improper External Communication (L: .500)**

**Institusional Pressure (L: .117) Unrealistically low bids (L: .118)**

**Unrealistically low budget requests (L: .244) Underestimates of time requirements (L: .167) Time to market (L: .471)**

**Inflexibility (L: .038) Software Failure (L: .167)**

**System overload (L: .160) Computational/Logic Error (L: .376) Resource Exhaustion (L: .186) Failed software upgrade (L: .148) Fault recovery routine (L: .130)**

**Human/Operator Error (L: .071) Configuration error (L: .413) Procedural error (L: .174) Miscellaneous accidents (L: .151) Organizational process (L: .045) Inadequate supervision (L: .074)**

**Planned inappropriate operations (L: .143) Hardware and Environmental Error (L: .146)**

**Device driver failure (L: .133) I/O error (L: .182)**

**Memory parity error (L: .082) Network hardware failure (L: .363) Power outages (L: .078) Overheating (L: .095)**

rosareska

## (lanjutan)

12/17/2015 1:02:07 PM Page 3 of 9

**High humidity (L: .041) Natural disaster (L: .026)**



**Fault Recovery (L: .061)**

**Improper fault assignment (L: .667) Complex fault recovery (L: .333)**

**Security Violations (L: .082) Password disclosures (L: .223)**

**Denial of service (DOS) attacks (L: .192) Worms and viruses (L: .055)**

**Browser vulnerabilities (L: .100) Authentication failures (L: .091) Insider threat (L: .339)**

**Change Management (L: .140)**

**Improper planning of change (L: .333) Improper testing of change (L: .667)**

**Communication Management (L: .169) Improper Internal Communication (L: .500) Improper External Communication (L: .500)**

**Institusional Pressure (L: .162) Unrealistically low bids (L: .131)**

**Unrealistically low budget requests (L: .235) Underestimates of time requirements (L: .337) Time to market (L: .297)**

**Unfunctionality (L: .134) Software Failure (L: .191)**

**System overload (L: .151) Computational/Logic Error (L: .440) Resource Exhaustion (L: .149) Failed software upgrade (L: .137) Fault recovery routine (L: .123)**

**Human/Operator Error (L: .119) Configuration error (L: .413) Procedural error (L: .174) Miscellaneous accidents (L: .151) Organizational process (L: .045)**

rosareska

## (lanjutan)

12/17/2015 1:02:07 PM Page 4 of 9

**Inadequate supervision (L: .074)**



**Planned inappropriate operations (L: .143) Hardware and Environmental Error (L: .124)**

**Device driver failure (L: .133) I/O error (L: .182)**

**Memory parity error (L: .082) Network hardware failure (L: .363) Power outages (L: .078) Overheating (L: .095)**

**High humidity (L: .041) Natural disaster (L: .026)**

**Fault Recovery (L: .097)**

**Improper fault assignment (L: .667) Complex fault recovery (L: .333)**

**Security Violations (L: .128) Password disclosures (L: .223)**

**Denial of service (DOS) attacks (L: .192) Worms and viruses (L: .055)**

**Browser vulnerabilities (L: .100) Authentication failures (L: .091) Insider threat (L: .339)**

**Change Management (L: .215)**

**Improper planning of change (L: .333) Improper testing of change (L: .667)**

**Communication Management (L: .071) Improper Internal Communication (L: .500) Improper External Communication (L: .500)**

**Institusional Pressure (L: .056) Unrealistically low bids (L: .118)**

**Unrealistically low budget requests (L: .244) Underestimates of time requirements (L: .167) Time to market (L: .471)**

**Unusability (L: .080) Software Failure (L: .247)**

**System overload (L: .151)**

rosareska

## (lanjutan)

12/17/2015 1:02:07 PM Page 5 of 9

**Computational/Logic Error (L: .440) Resource Exhaustion (L: .149) Failed software upgrade (L: .137) Fault recovery routine (L: .123) man/Operator Error (L: .172) Configuration error (L: .413) Procedural error (L: .174) Miscellaneous accidents (L: .151) Organizational process (L: .045) Inadequate supervision (L: .074)**

|  |  |
| --- | --- |
|  | **Hu** |
|  |
|  |
|  |
|  |
|  |
|  |
|  | **Ha** |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  | **Fau** |
|  |
|  |
|  | **Se** |
|  |
|  |
|  |
|  |
|  |
|  |
|  | **Cha**  **Co** |
|  |
|  |  |

**Planned inappropriate operations (L: .143) rdware and Environmental Error (L: .069) Device driver failure (L: .133)**

**I/O error (L: .182)**

**Memory parity error (L: .082) Network hardware failure (L: .363) Power outages (L: .078) Overheating (L: .095)**

**High humidity (L: .041) Natural disaster (L: .026) lt Recovery (L: .103)**

**Improper fault assignment (L: .667) Complex fault recovery (L: .333) curity Violations (L: .092)**

**Password disclosures (L: .223)**

**Denial of service (DOS) attacks (L: .192) Worms and viruses (L: .055)**

**Browser vulnerabilities (L: .100) Authentication failures (L: .091) Insider threat (L: .339)**

**nge Management (L: .211)**

**Improper planning of change (L: .333) Improper testing of change (L: .667) mmunication Management (L: .065) Improper Internal Communication (L: .500)**

rosareska

## (lanjutan)

12/17/2015 1:02:07 PM Page 6 of 9

 **Improper External Communication (L: .500) Institusional Pressure (L: .042)**

**Unrealistically low bids (L: .118) Unrealistically low budget requests (L: .244) Underestimates of time requirements (L: .167) Time to market (L: .471)**

**Untestability (L: .147) Software Failure (L: .163)**

**System overload (L: .151) Computational/Logic Error (L: .440) Resource Exhaustion (L: .149) Failed software upgrade (L: .137) Fault recovery routine (L: .123)**

**Human/Operator Error (L: .186) Configuration error (L: .413) Procedural error (L: .174) Miscellaneous accidents (L: .151) Organizational process (L: .045) Inadequate supervision (L: .074)**

**Planned inappropriate operations (L: .143) Hardware and Environmental Error (L: .170)**

**Device driver failure (L: .133) I/O error (L: .182)**

**Memory parity error (L: .082) Network hardware failure (L: .363) Power outages (L: .078) Overheating (L: .095)**

**High humidity (L: .041) Natural disaster (L: .026)**

**Fault Recovery (L: .059)**

**Improper fault assignment (L: .667) Complex fault recovery (L: .333)**

**Security Violations (L: .125) Password disclosures (L: .223)**

**Denial of service (DOS) attacks (L: .192)**

rosareska

## (lanjutan)

12/17/2015 1:02:07 PM Page 7 of 9

**Worms and viruses (L: .055) Browser vulnerabilities (L: .100) Authentication failures (L: .091) Insider threat (L: .339)**

**Change Management (L: .146)**

**Improper planning of change (L: .333) Improper testing of change (L: .667)**

**Communication Management (L: .078) Improper Internal Communication (L: .500) Improper External Communication (L: .500)**

**Institusional Pressure (L: .074) Unrealistically low bids (L: .118)**

**Unrealistically low budget requests (L: .244) Underestimates of time requirements (L: .167) Time to market (L: .471)**

**Unreliability (L: .235) Software Failure (L: .269)**

**System overload (L: .151) Computational/Logic Error (L: .440) Resource Exhaustion (L: .149) Failed software upgrade (L: .137) Fault recovery routine (L: .123)**

**Human/Operator Error (L: .078) Configuration error (L: .413) Procedural error (L: .174) Miscellaneous accidents (L: .151) Organizational process (L: .045) Inadequate supervision (L: .074)**

**Planned inappropriate operations (L: .143) Hardware and Environmental Error (L: .176)**

**Device driver failure (L: .133) I/O error (L: .182)**

**Memory parity error (L: .082) Network hardware failure (L: .363) Power outages (L: .078)**

rosareska

## (lanjutan)

12/17/2015 1:02:07 PM Page 8 of 9

**Overheating (L: .095) High humidity (L: .041) Natural disaster (L: .026)**



**Fault Recovery (L: .095)**

**Improper fault assignment (L: .667) Complex fault recovery (L: .333)**

**Security Violations (L: .157) Password disclosures (L: .223)**

**Denial of service (DOS) attacks (L: .192) Worms and viruses (L: .055)**

**Browser vulnerabilities (L: .100) Authentication failures (L: .091) Insider threat (L: .339)**

**Change Management (L: .145)**

**Improper planning of change (L: .333) Improper testing of change (L: .667)**

**Communication Management (L: .047) Improper Internal Communication (L: .500) Improper External Communication (L: .500)**

**Institusional Pressure (L: .033) Unrealistically low bids (L: .118)**

**Unrealistically low budget requests (L: .244) Underestimates of time requirements (L: .167) Time to market (L: .471)**

**Unintegrity (L: .287) Software Failure (L: .150)**

**System overload (L: .151) Computational/Logic Error (L: .440) Resource Exhaustion (L: .149) Failed software upgrade (L: .137) Fault recovery routine (L: .123)**

**Human/Operator Error (L: .205) Configuration error (L: .413) Procedural error (L: .174) Miscellaneous accidents (L: .151)**

rosareska

## (lanjutan)

12/17/2015 1:02:07 PM Page 9 of 9

**Organizational process (L: .045) Inadequate supervision (L: .074)**



**Planned inappropriate operations (L: .143) Hardware and Environmental Error (L: .049)**

**Device driver failure (L: .133) I/O error (L: .182)**

**Memory parity error (L: .082) Network hardware failure (L: .363) Power outages (L: .078) Overheating (L: .095)**

**High humidity (L: .041) Natural disaster (L: .026)**

**Fault Recovery (L: .074)**

**Improper fault assignment (L: .333) Complex fault recovery (L: .667)**

**Security Violations (L: .303) Password disclosures (L: .223)**

**Denial of service (DOS) attacks (L: .192) Worms and viruses (L: .055)**

**Browser vulnerabilities (L: .100) Authentication failures (L: .091) Insider threat (L: .339)**

**Change Management (L: .132)**

**Improper planning of change (L: .333) Improper testing of change (L: .667)**

**Communication Management (L: .051) Improper Internal Communication (L: .500) Improper External Communication (L: .500)**

**Institusional Pressure (L: .037) Unrealistically low bids (L: .118)**

**Unrealistically low budget requests (L: .244) Underestimates of time requirements (L: .167) Time to market (L: .471)**

rosareska

**(lanjutan)**

**Lampiran 7 Kuisioner *Pairwise Comparison***

**Kuisioner Karya Akhir**

**Analisis Faktor-faktor Penyebab Insiden Aplikasi: Studi Kasus PT Telkomsel dengan Metode *Analytic Hierarchy***

***Process* (AHP)**

## Bagian 1: Identitas Responden

**Identitas Responden**

Nama :

Jenis Kelamin :

Usia :

Pendidikan :

Jabatan :

Pengalaman : tahun (minimal 5 tahun)

## Bagian 2: Perbandingan Berpasangan

**Tujuan Kuisioner**

Tujuan dari kuisioner ini adalah untuk mendapatkan skala prioritas (*pairwise comparison*) antar elemen faktor penyebab insiden pada perusahaan telekomunikasi sesuai dengan skala penilaian yang direkomendasikan oleh Saaty. Responden dari kuisioner ini adalah para pakar di bidang *software quality management* di PT. Telkomsel. Hasil dari kusioner ini berupa skala prioritas antar elemen yang ada dalam struktur hierarki AHP yang telah

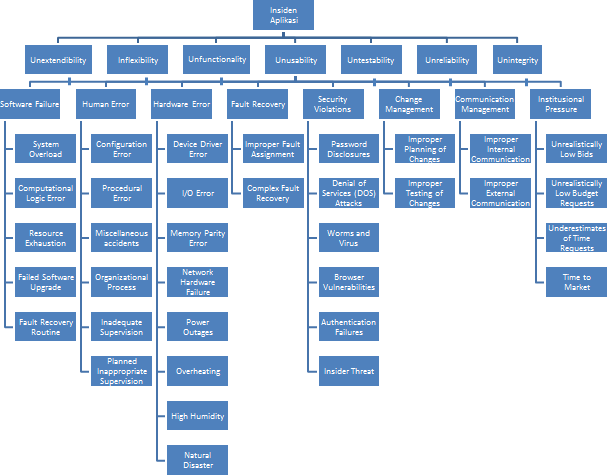
disususn berdasarkan studi literatur dan wawancara pada 3 pakar di bidang

**Protokol Kuisioner**

Kuisioner ini disebarkan pada beberapa pakar di bidang SQM dan Software Development dengan kriteria sebagai berikut :

* Memiliki pengalaman bekerja di bidang SD/SQM selama minimal 5 tahun
* Memiliki latar belakang pendidikan di bidang Teknologi Informasi

**STRUKTUR HIERARKI AHP**



Struktur hierarki AHP di atas menggambarkan hubungan antara tujuan, kriteria, dan alternatif. Level tertinggi dari hierarki merupakan tujuan yaitu insiden aplikasi. Level kedua merupakan kriteria aplikasi yang dapat menyebabkan insiden. Kriteria-kriteria ini akan menjadi indikator pembanding dari alternatif. Level paling akhir adalah alternatif yang merupakan faktor penyebab insiden pada aplikasi. Faktor-faktor ini yang akan mempengaruhi level di atasnya yaitu kualitas dari aplikasi. Setiap kategori faktor memiliki sub faktor yang merupakan akar penyebab masalah dari faktor tersebut.

## (lanjutan)

**Level 2: Kriteria**

Berikut merupakan penjelasan kriteria software yang dapat menyebabkan insiden aplikasi yang diteliti:

|  |  |
| --- | --- |
| **Kriteria Kegagalan Aplikasi** | **Definisi** |
| *Unextendability* | Ketidakmampuan aplikasi untuk dapat dikembangkan kapabilitas dan performance nya dengan mengembangkan fungsi saat ini atau dengan menambahkan fungsionalitas  baru |
| *Inflexibility* | Ketidakmudahan dalam mengubah tujuan, fungsi, atau data  aplikasi untuk memenuhi kebutuhan perubahan |
| *Unfunctionality* | Ketidakmampuan aplikasi untuk menyediakan fungsi yang sesuai dengan kebutuhan yang tersurat maupun tersirat  ketika aplikasi berada pada kondisi tertentu |
| *Unusability* | Ketidakmudahan untuk dipelajari dan dioperasikan |
| *Untestability* | Ketidakmudahan dalam menguji program untuk memastikan  bahwa aplikasi melakukan fungsi spesifik nya |
| *Unreliability* | Ketidakmampuan sebuah aplikasi untuk bebas dari kesalahan/kegagalan dalam waktu yang ditentukan dalam  lingkungan tertentu |
| *Unintegrity* | Ketidakmampuan sebuah aplikasi menjaga data tidak diubah  oleh pihak yang tidak berwenang |

## (lanjutan)

**Level 3: Kategori Faktor**

Berikut merupakan penjelasan kategori faktor yang dapat menyebabkan insiden aplikasi yang diteliti:

|  |  |
| --- | --- |
| **Kategori Faktor** | **Definisi** |
| *Software Error* | Kumpulan faktor yang berhubungan dengan  penyebab kegagalan aplikasi |
| *Human/Operator Error* | Kumpulan faktor yang berhubungan dengan  kesalahan manusia |
| *Hardware/Environmental*  *Failure* | Kumpulan faktor yang berhubungan dengan  kegagalan perangkat keras atau lingkungan |
| *Security Violations* | Kumpulan faktor yang berhubungan dengan  pelanggaran keamanan |
| *Fault Recovery Failure* | Kumpulan faktor yang berhubungan dengan kegagalan memperbaiki aplikasi setelah terjadi  kegagalan |
| *Communication*  *Management* | Kumpulan faktor yang berhubungan dengan  komunikasi |
| *Institusional Pressure* | Kumpulan faktor yang berhubungan dengan tekanan  institusi atau pihak ketiga |
| *Change Management*  *Failure* | Kumpulan faktor yang berhubungan dengan  perubahan |

## (lanjutan)

***Software*** *Error* Berikut merupakan penjelasan faktor-faktor dalam kategori

*Software Error* :

|  |  |  |
| --- | --- | --- |
| **Kategori Faktor** | **Subfaktor** | **Definisi** |
| *Software Error* | *System overload* | Ketidakmampuan untuk mengatasi pesan dan ekspektasi dari sejumlah sumber dalam  batas waktu tertentu |
| *Computational/Logic error* | Bug dalam program yang menyebabkan aplikasi beroperasi secara tidak benar, tetapi  tidak menyebabkan crash |
| *Resource exhaustion* | Keamanan komputer mengeksploitasi crash,  hang, atau menganggu program atau sistem yang menjadi target |
| *Failed software*  *upgrade* | Kegagalan yang terjadi saat mengupgrade  aplikasi |
| *Fault recovery routine* | Mekanisme dan teknik yang dilakukan untuk mengembalikan level operasi yang diinginkan dari sebuah komponen atau  sistem setelah terjadi kegagalan |

## (lanjutan)

***Human Error***: Berikut merupakan penjelasan faktor-faktor dalam kategori

*Human/Operator Error* :

|  |  |  |
| --- | --- | --- |
| Kategori  Faktor | Subfaktor | Definisi |
| *Human/Operat or Error* | *Configuration error* | Kegagalan yang terjadi karena kesalahan  dalam pengkonfigurasian aplikasi |
| *Procedural error* | Kegagalan yang terjadi karena prosedur  pengoperasian aplikasi tidak diikuti oleh operator |
| *Miscellaneous*  *accidents* | Berbagai kesalahan yang tidak disengaja |
| *Organizational process* | Penentuan proses dan tugas-tugas apa yang harus dilakukan untuk mencapai tujuan dan pengambilan keputusan tidak dilaksanakan dengan baik saat pengembangan /  pengoperasian aplikasi |
| *Inadequate*  *supervision* | Pengawasan yang tidak memadai saat  pengembangan / pengoperasian aplikasi |
| *Planned inappropriate*  *operations* | Merencanakan langkah operasi aplikasi dengan tidak tepat |

## (lanjutan)

***HW/Environment Error:*** Berikut merupakan penjelasan faktor-faktor dalam kategori *Hardware/Environmental Error*:

|  |  |  |
| --- | --- | --- |
| Kategori  Faktor | Subfaktor | Definisi |
| *Hardware and Environmental Error* | *Device driver failure* | Kegagalan yang disebabkan oleh device driver mengalami error. Salah satu sebabnya adalah serangan *malware*. Kegagalan suatu device driver biasa  ditandai dengan adanya pesan error |
| *I/O error* | Komputer tidak dapat melakukan tindakan input/output ketika mencoba mengakses  drive atau hard disk |
| *Memory parity error* | Satu atau lebih data yang disimpan membawa nilai yang berbeda ketika dipanggil kembali dengan nilai saat data tersebut disimpan. Parity error merupakan  jenis kerusakan data. |
| *Network hardware failure* | Kegagalan seluruh atau sebagian dari  komponen atau komponen jaringan karena sebuah kerusakan |
| *Power outages* | Hilangnya daya listrik dalam jangka waktu  yang pendek atau panjang |
| *Overheating* | Suhu hardware melebihi Batas normal  sehingga menghasilkan panas yang berlebihan |
| *High Humidity* | Hardware berada di lingkungan yang  tingkat kelembaban nya tinggi |
| *Natural disaster* | Bencana alam |

***Fault Recovery***: Berikut merupakan penjelasan faktor-faktor dalam kategori

*Fault Recovery*:

|  |  |  |
| --- | --- | --- |
| Kategori  Faktor | Subfaktor | Definisi |
| *Fault Recovery* | *Improper fault*  *assignment* | Proses menanggapi munculnya kesalahan  tidak diserahkan kepada pihak yang tepat |
| *Complex fault*  *recovery* | Prosedur penanganan fault susah dipahami |

## (lanjutan)

***Security Violations***: Berikut merupakan penjelasan faktor-faktor dalam kategori

*Security Violations*:

|  |  |  |
| --- | --- | --- |
| Kategori  Faktor | Subfaktor | Definisi |
| *Security Violations* | *Password*  *disclosures* | Pengungkapan password |
| *Denial of services (DOS) attacks* | Serangan terhadap sebuah komputer atau server di dalam jaringan internet dengan  cara menghabiskan resource yang dimiliki oleh komputer tersebut |
| *Worms and viruses* | Worms : Sebuah program komputer yang  dapat menggandakan dirinya secara sendiri dalam sistem komputer. |
| *Browser vulnerabilities* | Kerentanan dalam sistem operasi atau software dengan maksud untuk melanggar keamanan browser untuk mengubah pengaturan browser pengguna tanpa  sepengetahuan mereka |
| *Authentication*  *failures* | Kegagalan otentikasi |
| *Insider threat* | Adanya orang yang memiliki informasi  orang dalam mengenai praktek keamanan organisasi, data dan sistem komputer. |

***Change Management***: Berikut merupakan penjelasan faktor-faktor dalam kategori *Change Management*:

|  |  |  |
| --- | --- | --- |
| Kategori  Faktor | Subfaktor | Definisi |
| *Change Management* | *Improper planning*  *of change* | Perubahan dari sebuah aplikasi tidak  direncanakan dengan baik |
| *Improper testing of*  *change* | Perubahan dari sebuah aplikasi tidak diuji  dengan baik |

## (lanjutan)

***Communication Management***: Berikut merupakan penjelasan faktor-faktor dalam kategori *Communication Management*:

|  |  |  |
| --- | --- | --- |
| Kategori  Faktor | Subfaktor | Definisi |
| *Communicatio n Management* | *Improper internal*  *communication* | Komunikasi internal IT tidak dilakukan  dengan baik |
| *Improper external*  *communication* | Komunikasi ke pihak ketiga tidak  dilakukan dengan baik |

***Institusional Pressure***: Berikut merupakan penjelasan faktor-faktor dalam kategori *Institusional Pressure*:

|  |  |  |
| --- | --- | --- |
| Kategori  Faktor | Subfaktor | Definisi |
| *Institusional Pressure* | *Unrealistically low*  *bids* | Tawaran yang terlalu rendah untuk sebuah  pengembangan aplikasi |
| *Unrealistically low*  *budget requests* | Permintaan biaya yang terlalu rendah untuk  sebuah pengembangan aplikasi |
| *Underestimates of*  *time requests* | Tidak memperhitungkan secara tepat lama  waktu pengembangan ssebuah aplikasi |
| *Time to market* | Desakan pasar/kompetitor agar aplikasi  segera di-release |

Pada bagian ini, responden diberikan kuisoner dalam bentuk perbandingan berpasangan untuk membandingkan kriteria/faktor di sisi kiri (A) dan di sisi kanan (B) yang berkaitan dengan insiden aplikasi. Responden diminta untuk memilih salah satu nilai pada kolom yang disediakan dengan aturan skala penilaian sebagai berikut :

## (lanjutan)

|  |  |
| --- | --- |
| Nilai | Keterangan |
| 1 | A dan B memiliki tingkat potensi yang sama |
| 3 | A sedikit lebih berpotensi dari B |
| 5 | A jelas lebih berpotensi dari B |
| 7 | A sangat jelas berpotensi dari B |
| 9 | A mutlak lebih berpotensi dari B |
| 2,4,6,8 | Ragu-ragu |

**CONTOH PENGISIAN**

**Membandingan kriteria mana yang lebih berpotensi menyebabkan insiden antara A dan B**

**A sangat jelas berpotensi daripada B**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Kriteria | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Kriteria |
| A |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  | B |

Jika kriteria A sangat jelas lebih berpotensi menyebabkan insiden pada aplikasi disbanding kriteria B, maka repsonden harus memilih nilai 7 yang berarti “A sangat jelas berpotensi dari B” pada posisi kolom di sebelah kiri (sisi kriteria A)

## B sangat jelas berpotensi daripada A

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Kriteria | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Kriteria |
| A |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  | B |

Jika kriteria B sangat jelas lebih berpotensi menyebabkan insiden pada aplikasi disbanding kriteria A, maka repsonden harus memilih nilai 7 yang berarti “B sangat jelas berpotensi dari A” pada posisi kolom di sebelah kanan (sisi kriteria B)

## (lanjutan)

KUISIONER PERBANDINGAN BERPASANGAN PERBANDINGAN BERPASANGAN **ANTAR KRITERIA** APLIKASI YANG

BERPOTENSI MENYEBABKAN INSIDEN

Tabel skala penilaian :

|  |  |
| --- | --- |
| Nilai | Keterangan |
| 1 | A dan B memiliki tingkat potensi yang sama |
| 3 | A sedikit lebih berpotensi dari B |
| 5 | A jelas lebih berpotensi dari B |
| 7 | A sangat jelas berpotensi dari B |
| 9 | A mutlak lebih berpotensi dari B |
| 2,4,6,8 | Ragu-ragu |

## Kriteria manakah yang lebih berpotensi menyebabkan insiden? Berilah tanda X pada kriteria yang menurut Anda lebih berpotensi menyebabkan insiden pada aplikasi.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Kriteria** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **Kriteria** |
| ***Unextendibility*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Inflexibility*** |
| ***Unextendibility*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Unfunctionality*** |
| ***Unextendibility*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Unusability*** |
| ***Unextendibility*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Untestability*** |
| ***Unextendibility*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Unreliability*** |
| ***Unextendibility*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Unintegrity*** |
| ***Inflexibility*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Unfunctionality*** |
| ***Inflexibility*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Unusability*** |
| ***Inflexibility*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Untestability*** |
| ***Inflexibility*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Unreliability*** |
| ***Inflexibility*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Unintegrity*** |
| ***Unfunctionality*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Unusability*** |
| ***Unfunctionality*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Untestability*** |
| ***Unfunctionality*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Unreliability*** |
| ***Unfunctionality*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Unintegrity*** |
| ***Unusability*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Untestability*** |
| ***Unusability*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Unreliability*** |
| ***Unusability*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Unintegrity*** |
| ***Untestability*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Unreliability*** |
| ***Untestability*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Unintegrity*** |
| ***Unreliability*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Unintegrity*** |

**(lanjutan)**

PERBANDINGAN BERPASANGAN **ANTAR KATEGORI FAKTOR**

TERHADAP **KRITERIA**

Perbandingan Berpasangan **antar kategori faktor** penyebab insiden aplikasi terhadap kriteria **Unextendibility**

|  |  |
| --- | --- |
| Nilai | Keterangan |
| 1 | A dan B memiliki tingkat potensi yang sama |
| 3 | A sedikit lebih berpotensi dari B |
| 5 | A jelas lebih berpotensi dari B |
| 7 | A sangat jelas berpotensi dari B |
| 9 | A mutlak lebih berpotensi dari B |
| 2,4,6,8 | Ragu-ragu |

Tabel skala penilaian :

## Kategori faktor manakah yang lebih berpotensi menyebabkan insiden aplikasi berkenaan dengan kriteria *Unextendebility*? Berilah tanda X pada kategori faktor yang menurut Anda lebih berpotensi menyebabkan insiden pada aplikasi.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Kriteria** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **Kriteria** |
| ***Software*** *Error* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Human Error*** |
| ***Software*** *Error* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***HW/Environmental Error*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Fault Recovery*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Security Violations*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Change Management*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication***  ***Management*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional Pressure*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  | ***HW/Environmental***  ***Error*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Fault Recovery*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Security Violations*** |

**(lanjutan)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Kriteria** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **Kriteria** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Change Management*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication***  ***Management*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional***  ***Pressure*** |
| ***HW/Environemtal***  ***Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Fault Recovery*** |
| ***HW/Environemtal Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Security Violations*** |
| ***HW/Environemtal Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Change Management*** |
| ***HW/Environemtal***  ***Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication***  ***Management*** |
| ***HW/Environemtal Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional Pressure*** |
| ***Fault Recovery*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Security Violations*** |
| ***Fault Recovery*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Change***  ***Management*** |
| ***Fault Recovery*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication***  ***Management*** |
| ***Fault Recovery*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional Pressure*** |
| ***Security Violations*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Change Management*** |
| ***Security Violations*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication Management*** |
| ***Security***  ***Violations*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional***  ***Pressure*** |
| ***Change Management*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication Management*** |
| ***Change***  ***Management*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional***  ***Pressure*** |
| ***Communication Management*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional Pressure*** |

**(lanjutan)**

Perbandingan Berpasangan **antar kategori faktor** penyebab insiden aplikasi

## terhadap kriteria Inflexibility

Tabel skala penilaian :

|  |  |
| --- | --- |
| Nilai | Keterangan |
| 1 | A dan B memiliki tingkat potensi yang sama |
| 3 | A sedikit lebih berpotensi dari B |
| 5 | A jelas lebih berpotensi dari B |
| 7 | A sangat jelas berpotensi dari B |
| 9 | A mutlak lebih berpotensi dari B |
| 2,4,6,8 | Ragu-ragu |

## Kategori faktor manakah yang lebih berpotensi menyebabkan insiden aplikasi berkenaan dengan kriteria *Inflexibility*? Berilah tanda X pada kategori faktor yang menurut Anda lebih berpotensi menyebabkan insiden pada aplikasi

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Kriteria** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **Kriteria** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Human Error*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***HW/Environmental Error*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Fault Recovery*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Security Violations*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Change Management*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication Management*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional***  ***Pressure*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***HW/Environmental***  ***Error*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Fault Recovery*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Security Violations*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Change Management*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication***  ***Management*** |

**(lanjutan)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional Pressure*** |
| ***HW/Environemtal***  ***Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Fault Recovery*** |
| ***HW/Environemtal Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Security Violations*** |
| ***HW/Environemtal Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Change Management*** |
| ***HW/Environemtal***  ***Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication***  ***Management*** |
| ***HW/Environemtal Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional Pressure*** |
| ***Fault Recovery*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Security Violations*** |
| ***Fault Recovery*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Change Management*** |
| ***Fault Recovery*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication***  ***Management*** |
| ***Fault Recovery*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional Pressure*** |
| ***Security Violations*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Change Management*** |
| ***Security Violations*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication Management*** |
| ***Security Violations*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional Pressure*** |
| ***Change***  ***Management*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication***  ***Management*** |
| ***Change Management*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional Pressure*** |
| ***Communication***  ***Management*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional***  ***Pressure*** |

**(lanjutan)**

Perbandingan Berpasangan **antar kategori faktor** penyebab insiden aplikasi terhadap kriteria **Unfunctionality**

Tabel skala penilaian :

|  |  |
| --- | --- |
| Nilai | Keterangan |
| 1 | A dan B memiliki tingkat potensi yang sama |
| 3 | A sedikit lebih berpotensi dari B |
| 5 | A jelas lebih berpotensi dari B |
| 7 | A sangat jelas berpotensi dari B |
| 9 | A mutlak lebih berpotensi dari B |
| 2,4,6,8 | Ragu-ragu |

## Kategori faktor manakah yang lebih berpotensi menyebabkan insiden aplikasi berkenaan dengan kriteria *Unfunctionality*? Berilah tanda X pada kategori faktor yang menurut Anda lebih berpotensi menyebabkan insiden pada aplikasi.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Kriteria** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **Kriteria** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Human Error*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***HW/Environmental Error*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Fault Recovery*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Security Violations*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Change Management*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication***  ***Management*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional Pressure*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***HW/Environmental***  ***Error*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Fault Recovery*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Security Violations*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Change***  ***Management*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication Management*** |

**(lanjutan)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional Pressure*** |
| ***HW/Environemtal Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Fault Recovery*** |
| ***HW/Environemtal***  ***Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Security Violations*** |
| ***HW/Environemtal Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Change Management*** |
| ***HW/Environemtal***  ***Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication***  ***Management*** |
| ***HW/Environemtal Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional Pressure*** |
| ***Fault Recovery*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Security Violations*** |
| ***Fault Recovery*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Change Management*** |
| ***Fault Recovery*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication***  ***Management*** |
| ***Fault Recovery*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional Pressure*** |
| ***Security Violations*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Change Management*** |
| ***Security Violations*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication Management*** |
| ***Security Violations*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional Pressure*** |
| ***Change***  ***Management*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication***  ***Management*** |
| ***Change Management*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional Pressure*** |
| ***Communication***  ***Management*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional***  ***Pressure*** |

‘

## (lanjutan)

Perbandingan Berpasangan **antar kategori faktor** penyebab insiden aplikasi terhadap kriteria **Unusability**

Tabel skala penilaian :

|  |  |
| --- | --- |
| Nilai | Keterangan |
| 1 | A dan B memiliki tingkat potensi yang sama |
| 3 | A sedikit lebih berpotensi dari B |
| 5 | A jelas lebih berpotensi dari B |
| 7 | A sangat jelas berpotensi dari B |
| 9 | A mutlak lebih berpotensi dari B |
| 2,4,6,8 | Ragu-ragu |

## Kategori faktor manakah yang lebih berpotensi menyebabkan insiden aplikasi berkenaan dengan kriteria *Unusability*? Berilah tanda X pada kategori faktor yang menurut Anda lebih berpotensi menyebabkan insiden pada aplikasi

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Kriteria** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **Kriteria** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Human Error*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***HW/Environmental Error*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Fault Recovery*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Security Violations*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Change Management*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication***  ***Management*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional Pressure*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***HW/Environmental***  ***Error*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Fault Recovery*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Security Violations*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Change***  ***Management*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication***  ***Management*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional Pressure*** |

**(lanjutan)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ***HW/Environemtal Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Fault Recovery*** |
| ***HW/Environemtal***  ***Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Security Violations*** |
| ***HW/Environemtal***  ***Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Change***  ***Management*** |
| ***HW/Environemtal***  ***Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication***  ***Management*** |
| ***HW/Environemtal Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional Pressure*** |
| ***Fault Recovery*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Security Violations*** |
| ***Fault Recovery*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Change Management*** |
| ***Fault Recovery*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication***  ***Management*** |
| ***Fault Recovery*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional Pressure*** |
| ***Security***  ***Violations*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Change***  ***Management*** |
| ***Security***  ***Violations*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication***  ***Management*** |
| ***Security Violations*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional Pressure*** |
| ***Change***  ***Management*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication***  ***Management*** |
| ***Change Management*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional Pressure*** |
| ***Communication***  ***Management*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional***  ***Pressure*** |

‘

## (lanjutan)

Perbandingan Berpasangan **antar kategori faktor** penyebab insiden aplikasi terhadap kriteria **Untestability**

Tabel skala penilaian :

|  |  |
| --- | --- |
| Nilai | Keterangan |
| 1 | A dan B memiliki tingkat potensi yang sama |
| 3 | A sedikit lebih berpotensi dari B |
| 5 | A jelas lebih berpotensi dari B |
| 7 | A sangat jelas berpotensi dari B |
| 9 | A mutlak lebih berpotensi dari B |
| 2,4,6,8 | Ragu-ragu |

## Kategori faktor manakah yang lebih berpotensi menyebabkan insiden aplikasi berkenaan dengan kriteria *Untestabiltiy?* Berilah tanda X pada kategori faktor yang menurut Anda lebih berpotensi menyebabkan insiden pada aplikasi.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Kriteria** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **Kriteria** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Human Error*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***HW/Environmental Error*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Fault Recovery*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Security Violations*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Change Management*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication***  ***Management*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional Pressure*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***HW/Environmental***  ***Error*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Fault Recovery*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Security Violations*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Change***  ***Management*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication***  ***Management*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional Pressure*** |

**(lanjutan)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ***HW/Environemtal Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Fault Recovery*** |
| ***HW/Environemtal***  ***Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Security Violations*** |
| ***HW/Environemtal***  ***Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Change***  ***Management*** |
| ***HW/Environemtal***  ***Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication***  ***Management*** |
| ***HW/Environemtal Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional Pressure*** |
| ***Fault Recovery*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Security Violations*** |
| ***Fault Recovery*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Change Management*** |
| ***Fault Recovery*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication***  ***Management*** |
| ***Fault Recovery*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional Pressure*** |
| ***Security***  ***Violations*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Change***  ***Management*** |
| ***Security***  ***Violations*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication***  ***Management*** |
| ***Security Violations*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional Pressure*** |
| ***Change***  ***Management*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication***  ***Management*** |
| ***Change Management*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional Pressure*** |
| ***Communication***  ***Management*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional***  ***Pressure*** |

**(lanjutan)**

Perbandingan Berpasangan **antar kategori faktor** penyebab insiden aplikasi terhadap kriteria **Unreliability**

Tabel skala penilaian :

|  |  |
| --- | --- |
| Nilai | Keterangan |
| 1 | A dan B memiliki tingkat potensi yang sama |
| 3 | A sedikit lebih berpotensi dari B |
| 5 | A jelas lebih berpotensi dari B |
| 7 | A sangat jelas berpotensi dari B |
| 9 | A mutlak lebih berpotensi dari B |
| 2,4,6,8 | Ragu-ragu |

## Kategori faktor manakah yang lebih berpotensi menyebabkan insiden aplikasi berkenaan dengan kriteria *Unreliability?* Berilah tanda X pada kategori faktor yang menurut Anda lebih berpotensi menyebabkan insiden pada aplikasi.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Kriteria** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **Kriteria** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***HumanError*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***HW/Environmental Error*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Fault Recovery*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Security Violations*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Change Management*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication***  ***Management*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional Pressure*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***HW/Environmental***  ***Error*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Fault Recovery*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Security Violations*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Change***  ***Management*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication Management*** |

**(lanjutan)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional Pressure*** |
| ***HW/Environemtal***  ***Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Fault Recovery*** |
| ***HW/Environemtal***  ***Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Security Violations*** |
| ***HW/Environemtal***  ***Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Change***  ***Management*** |
| ***HW/Environemtal Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication Management*** |
| ***HW/Environemtal Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional Pressure*** |
| ***Fault Recovery*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Security Violations*** |
| ***Fault Recovery*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Change***  ***Management*** |
| ***Fault Recovery*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication Management*** |
| ***Fault Recovery*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional***  ***Pressure*** |
| ***Security***  ***Violations*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Change***  ***Management*** |
| ***Security Violations*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication Management*** |
| ***Security***  ***Violations*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional***  ***Pressure*** |
| ***Change Management*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication Management*** |
| ***Change***  ***Management*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional***  ***Pressure*** |
| ***Communication Management*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional Pressure*** |

**(lanjutan)**

Perbandingan Berpasangan **antar kategori faktor** penyebab insiden aplikasi terhadap kriteria **Unintegrity**

Tabel skala penilaian :

|  |  |
| --- | --- |
| Nilai | Keterangan |
| 1 | A dan B memiliki tingkat potensi yang sama |
| 3 | A sedikit lebih berpotensi dari B |
| 5 | A jelas lebih berpotensi dari B |
| 7 | A sangat jelas berpotensi dari B |
| 9 | A mutlak lebih berpotensi dari B |
| 2,4,6,8 | Ragu-ragu |

## Kategori faktor manakah yang lebih berpotensi menyebabkan insiden aplikasi berkenaan dengan kriteria *Unintegrity?* Berilah tanda X pada kategori faktor yang menurut Anda lebih berpotensi menyebabkan insiden pada aplikasi.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Kriteria** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **Kriteria** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Human Error*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***HW/Environmental Error*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Fault Recovery*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Security Violations*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Change Management*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication***  ***Management*** |
| ***Software Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional Pressure*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***HW/Environmental Error*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Fault Recovery*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Security Violations*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Change Management*** |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication***  ***Management*** |

**(lanjutan)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ***Human Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional Pressure*** |
| ***HW/Environemtal***  ***Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Fault Recovery*** |
| ***HW/Environemtal***  ***Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Security Violations*** |
| ***HW/Environemtal***  ***Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Change***  ***Management*** |
| ***HW/Environemtal Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication Management*** |
| ***HW/Environemtal Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional Pressure*** |
| ***Fault Recovery*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Security Violations*** |
| ***Fault Recovery*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Change***  ***Management*** |
| ***Fault Recovery*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication Management*** |
| ***Fault Recovery*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional***  ***Pressure*** |
| ***Security***  ***Violations*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Change***  ***Management*** |
| ***Security Violations*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication Management*** |
| ***Security***  ***Violations*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional***  ***Pressure*** |
| ***Change Management*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Communication Management*** |
| ***Change***  ***Management*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional***  ***Pressure*** |
| ***Communication Management*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Institusional Pressure*** |

**(lanjutan)**

**PERBANDINGAN BERPASANGAN ANTAR FAKTOR TERHADAP**

KATEGORI FAKTOR

Perbandingan Berpasangan **antar faktor** penyebab insiden aplikasi terhadap kategori faktor **Software Failure**

Tabel skala penilaian :

|  |  |
| --- | --- |
| Nilai | Keterangan |
| 1 | A dan B memiliki tingkat potensi yang sama |
| 3 | A sedikit lebih berpotensi dari B |
| 5 | A jelas lebih berpotensi dari B |
| 7 | A sangat jelas berpotensi dari B |
| 9 | A mutlak lebih berpotensi dari B |
| 2,4,6,8 | Ragu-ragu |

## Faktor manakah yang lebih berpotensi menyebabkan insiden aplikasi dalam kategori faktor *Software Failure?* Berilah tanda X pada faktor yang menurut Anda lebih berpotensi menyebabkan insiden pada aplikasi.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Kriteria** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **Kriteria** |
| ***System Overload*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Computational/***  ***Logic Error*** |
| ***System Overload*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Resource Exhaustion*** |
| ***System Overload*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Failed Software***  ***Upgrade*** |
| ***System Overload*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Fault Recovery Routine*** |
| ***Computational/***  ***Logic Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Resource***  ***Exhaustion*** |
| ***Computational/ Logic Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Failed Software Upgrade*** |
| ***Computational/***  ***Logic Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Fault Recovery***  ***Routine*** |
| ***Resource***  ***Exhaustion*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Failed Software***  ***Upgrade*** |
| ***Resource Exhaustion*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Fault Recovery Routine*** |
| ***Failed Software***  ***Upgrade*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Fault Recovery***  ***Routine*** |

**(lanjutan)**

Perbandingan Berpasangan **antar faktor** penyebab insiden aplikasi terhadap kategori faktor **Human/Operator Error**

Tabel skala penilaian :

|  |  |
| --- | --- |
| Nilai | Keterangan |
| 1 | A dan B memiliki tingkat potensi yang sama |
| 3 | A sedikit lebih berpotensi dari B |
| 5 | A jelas lebih berpotensi dari B |
| 7 | A sangat jelas berpotensi dari B |
| 9 | A mutlak lebih berpotensi dari B |
| 2,4,6,8 | Ragu-ragu |

## Faktor manakah yang lebih berpotensi menyebabkan insiden aplikasi dalam kategori faktor *Human/Operator Error?* Berilah tanda X pada faktor yang menurut Anda lebih berpotensi menyebabkan insiden pada aplikasi.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Kriteria** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **Kriteria** |
| ***Configuration Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Procedural Error*** |
| ***Configuration***  ***Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Miscellaneous***  ***Accidents*** |
| ***Configuration Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Organizational Process*** |
| ***Configuration***  ***Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Inadequate***  ***Supervision*** |
| ***Configuration Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Planned***  ***Inappropriate Operations*** |
| ***Procedural Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Miscellaneous Accidents*** |
| ***Procedural***  ***Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Organizational***  ***Process*** |
| ***Procedural Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Inadequate Supervision*** |
| ***Procedural Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Planned Inappropriate Operations*** |

**(lanjutan)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ***Miscellaneous Accidents*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Organizational Process*** |
| ***Miscellaneous***  ***Accidents*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Inadequate***  ***Supervision*** |
| ***Miscellaneous Accidents*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Planned Inappropriate***  ***Operations*** |
| ***Organizational Process*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Inadequate Supervision*** |
| ***Organizational Process*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Planned***  ***Inappropriate Operations*** |
| ***Inadequate Supervision*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Planned Inappropriate Operations*** |

**(lanjutan)**

Perbandingan Berpasangan **antar faktor** penyebab insiden aplikasi terhadap kategori faktor **Hardware/Environmental Error**

Tabel skala penilaian :

|  |  |
| --- | --- |
| Nilai | Keterangan |
| 1 | A dan B memiliki tingkat potensi yang sama |
| 3 | A sedikit lebih berpotensi dari B |
| 5 | A jelas lebih berpotensi dari B |
| 7 | A sangat jelas berpotensi dari B |
| 9 | A mutlak lebih berpotensi dari B |
| 2,4,6,8 | Ragu-ragu |

## Faktor manakah yang lebih berpotensi menyebabkan insiden aplikasi dalam kategori faktor *Hardware/Environmental Error?* Berilah tanda X pada faktor yang menurut Anda lebih berpotensi menyebabkan insiden pada aplikasi.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Kriteria** | | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **Kriteria** | |
| ***Device Failure*** | ***Driver*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***I/O Error*** | |
| ***Device***  ***Failure*** | ***Driver*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Memory***  ***Error*** | ***Parity*** |
| ***Device Failure*** | ***Driver*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Network Hardware Failure*** | |
| ***Device***  ***Failure*** | ***Driver*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Power Outages*** | |
| ***Device Failure*** | ***Driver*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Overheating*** | |
| ***Device Failure*** | ***Driver*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***High Humidity*** | |
| ***Device***  ***Failure*** | ***Driver*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Natural Disaster*** | |
| ***I/O Error*** | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Memory Error*** | ***Parity*** |
| ***I/O Error*** | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Network Hardware***  ***Failure*** | |

**(lanjutan)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ***I/O Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Power Outages*** |
| ***I/O Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Overheating*** |
| ***I/O Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***High Humidity*** |
| ***I/O Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Natural Disaster*** |
| ***Memory Parity Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Network Hardware Failure*** |
| ***Memory Parity Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Power Outages*** |
| ***Memory Parity Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Overheating*** |
| ***Memory Parity***  ***Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***High Humidity*** |
| ***Memory Parity Error*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Natural Disaster*** |
| ***Network Hardware***  ***Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Power Outages*** |
| ***Network Hardware Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Overheating*** |
| ***Network Hardware Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***High Humidity*** |
| ***Network Hardware Failure*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Natural Disaster*** |
| ***Power Outages*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Overheating*** |
| ***Power Outages*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***High Humidity*** |
| ***Power Outages*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Natural Disaster*** |
| ***Overheating*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***High Humidity*** |
| ***Overheating*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Natural Disaster*** |
| ***High Humidity*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Natural Disaster*** |

**(lanjutan)**

Perbandingan Berpasangan **antar faktor** penyebab insiden aplikasi terhadap kategori faktor **Fault Recovery**

Tabel skala penilaian :

|  |  |
| --- | --- |
| Nilai | Keterangan |
| 1 | A dan B memiliki tingkat potensi yang sama |
| 3 | A sedikit lebih berpotensi dari B |
| 5 | A jelas lebih berpotensi dari B |
| 7 | A sangat jelas berpotensi dari B |
| 9 | A mutlak lebih berpotensi dari B |
| 2,4,6,8 | Ragu-ragu |

## Faktor manakah yang lebih berpotensi menyebabkan insiden aplikasi dalam kategori faktor *Fault Recovery?* Berilah tanda X pada faktor yang menurut Anda lebih berpotensi menyebabkan insiden pada aplikasi.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Kriteria** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **Kriteria** |
| ***Improper Fault Assignment*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Complex Fault Recovery*** |

Perbandingan Berpasangan **antar faktor** penyebab insiden aplikasi terhadap kategori faktor **Security Violations**

Tabel skala penilaian :

|  |  |
| --- | --- |
| Nilai | Keterangan |
| 1 | A dan B memiliki tingkat potensi yang sama |
| 3 | A sedikit lebih berpotensi dari B |
| 5 | A jelas lebih berpotensi dari B |
| 7 | A sangat jelas berpotensi dari B |
| 9 | A mutlak lebih berpotensi dari B |
| 2,4,6,8 | Ragu-ragu |

## (lanjutan)

**Faktor manakah yang lebih berpotensi menyebabkan insiden aplikasi dalam kategori faktor *Security Violations?* Berilah tanda X pada faktor yang menurut Anda lebih berpotensi menyebabkan insiden pada aplikasi.**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Kriteria** | | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **Kriteria** | |
| ***Password Disclosure*** | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Denial Services***  ***Attacks*** | ***of (DOS)*** |
| ***Password***  ***Disclosure*** | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Worms***  ***Viruses*** | ***and*** |
| ***Password***  ***Disclosure*** | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Browser***  ***Vulnerabilities*** | |
| ***Password Disclosure*** | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Authentication Failures*** | |
| ***Password***  ***Disclosure*** | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Insider Threat*** | |
| ***Denial of Services (DOS)***  ***Attacks*** | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Worms Viruses*** | ***and*** |
| ***Denial of Services (DOS) Attacks*** | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Browser Vulnerabilities*** | |
| ***Denial of Services (DOS) Attacks*** | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Authentication Failures*** | |
| ***Denial of***  ***Services (DOS) Attacks*** | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Insider Threat*** | |
| ***Worms***  ***Viruses*** | ***and*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Browser***  ***Vulnerabilities*** | |
| ***Worms Viruses*** | ***and*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Authentication Failures*** | |
| ***Worms***  ***Viruses*** | ***and*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Insider Threat*** | |
| ***Browser Vulnerabilities*** | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Authentication Failures*** | |
| ***Browser Vulnerabilities*** | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Insider Threat*** | |
| ***Authentication***  ***Failures*** | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Insider Threat*** | |

**(lanjutan)**

Perbandingan Berpasangan **antar faktor** penyebab insiden aplikasi terhadap kategori faktor **Change Management**

Tabel skala penilaian :

|  |  |
| --- | --- |
| Nilai | Keterangan |
| 1 | A dan B memiliki tingkat potensi yang sama |
| 3 | A sedikit lebih berpotensi dari B |
| 5 | A jelas lebih berpotensi dari B |
| 7 | A sangat jelas berpotensi dari B |
| 9 | A mutlak lebih berpotensi dari B |
| 2,4,6,8 | Ragu-ragu |

## Faktor manakah yang lebih berpotensi menyebabkan insiden aplikasi dalam kategori faktor *Change Management?* Berilah tanda X pada faktor yang menurut Anda lebih berpotensi menyebabkan insiden pada aplikasi.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Kriteria** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **Kriteria** |
| ***Improper Planning of Changes*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Improper Testing of Changes*** |

**(lanjutan)**

Perbandingan Berpasangan **antar faktor** penyebab insiden aplikasi terhadap kategori faktor **Communication Management**

Tabel skala penilaian :

|  |  |
| --- | --- |
| Nilai | Keterangan |
| 1 | A dan B memiliki tingkat potensi yang sama |
| 3 | A sedikit lebih berpotensi dari B |
| 5 | A jelas lebih berpotensi dari B |
| 7 | A sangat jelas berpotensi dari B |
| 9 | A mutlak lebih berpotensi dari B |
| 2,4,6,8 | Ragu-ragu |

## Faktor manakah yang lebih berpotensi menyebabkan insiden aplikasi dalam kategori faktor *Communication Management?* Berilah tanda X pada faktor yang menurut Anda lebih berpotensi menyebabkan insiden pada aplikasi.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Kriteria** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **Kriteria** |
| ***Improper Internal Communication*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Improper External Communication*** |

Perbandingan Berpasangan **antar faktor** penyebab insiden aplikasi terhadap kategori faktor **Institusional Pressure**

Tabel skala penilaian :

|  |  |
| --- | --- |
| Nilai | Keterangan |
| 1 | A dan B memiliki tingkat potensi yang sama |
| 3 | A sedikit lebih berpotensi dari B |
| 5 | A jelas lebih berpotensi dari B |
| 7 | A sangat jelas berpotensi dari B |
| 9 | A mutlak lebih berpotensi dari B |
| 2,4,6,8 | Ragu-ragu |

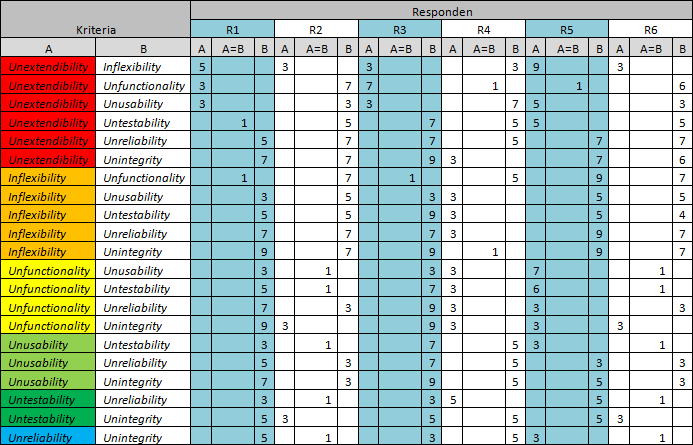
## (lanjutan)

**Faktor manakah yang lebih berpotensi menyebabkan insiden aplikasi dalam kategori faktor *Institusional Pressure?* Berilah tanda X pada faktor yang menurut Anda lebih berpotensi menyebabkan insiden pada aplikasi.**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Kriteria** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **Kriteria** |
| ***Unrealistically Low Bids*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Unrealistically Low Budget***  ***Requests*** |
| ***Unrealistically Low Bids*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Underestimates of Time***  ***Requirement*** |
| ***Unrealistically Low Bids*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Time To Market*** |
| ***Unrealistically Low Budget***  ***Requests*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Underestimates of Time***  ***Requirement*** |
| ***Unrealistically Low Budget***  ***Requests*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Time To Market*** |
| ***Underestimates of Time***  ***Requirement*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***Time To Market*** |

**Lampiran 8 Hasil Kuisioner *Pairwise Comparison***

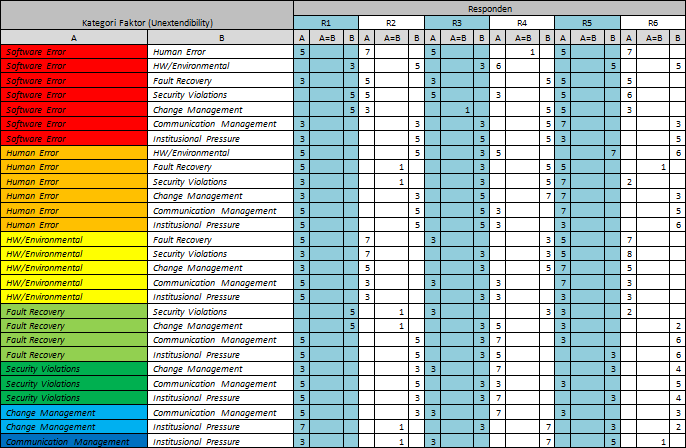
## Hasil Kuisioner Level 2 (Kriteria Aplikasi yang Menyebabkan Insiden)



**(lanjutan)**

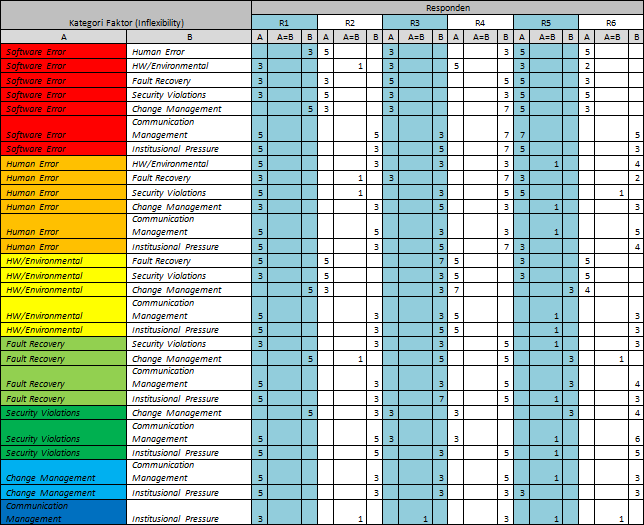
**Hasil Kuisioner Level 3 (Kategori Faktor terhadap Setiap Kriteria)**

1. Kriteria Unextendibility



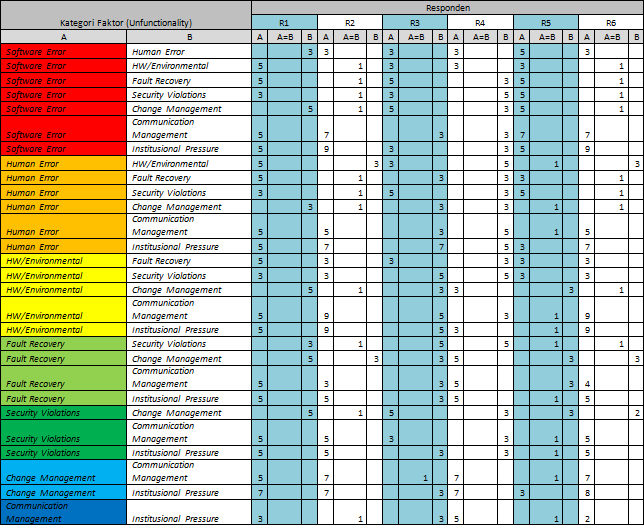
## (lanjutan)

1. Kriteria Inflexibility



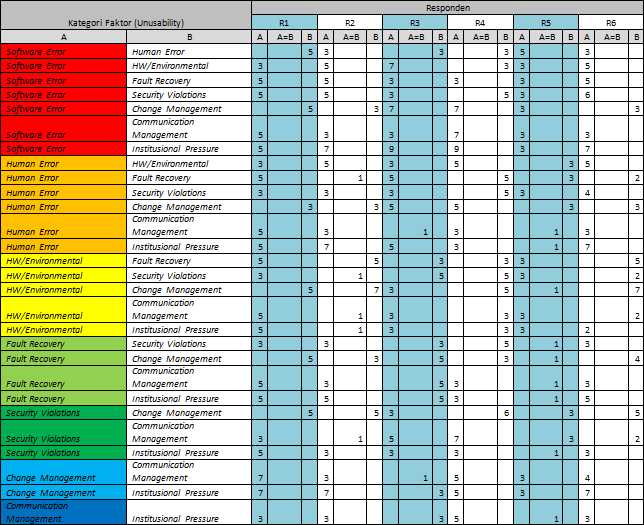
## (lanjutan)

1. Kriteria Unfunctionality



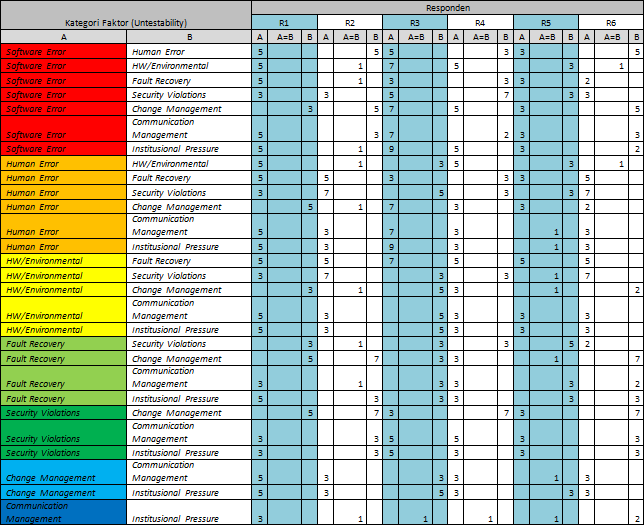
## (lanjutan)

1. Kriteria Unusability



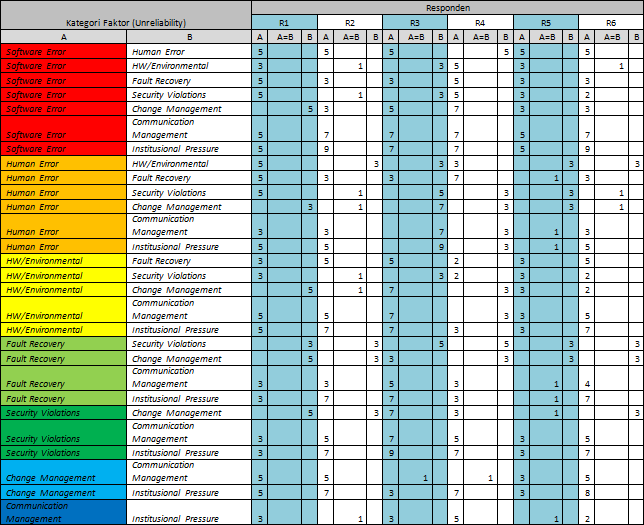
## (lanjutan)

1. Kriteria Untestability



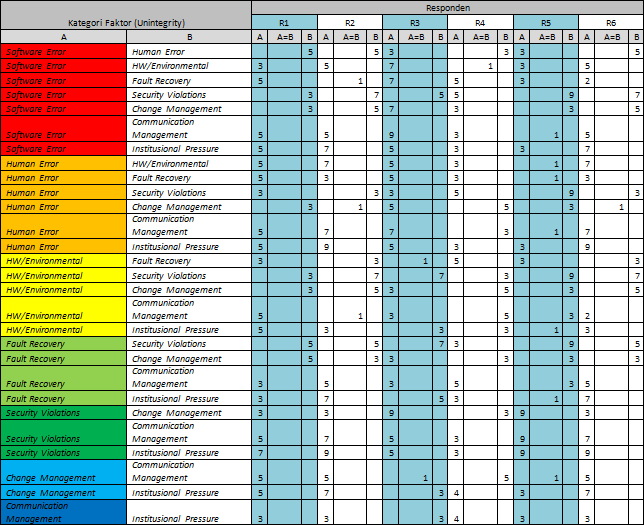
## (lanjutan)

1. Kriteria Unreliability



## (lanjutan)

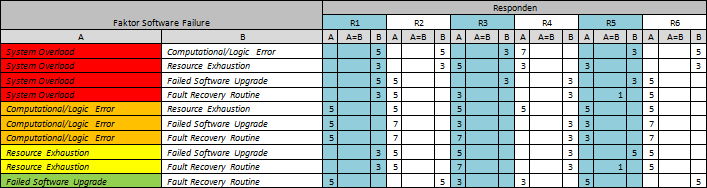
1. Kriteria Unintegrity



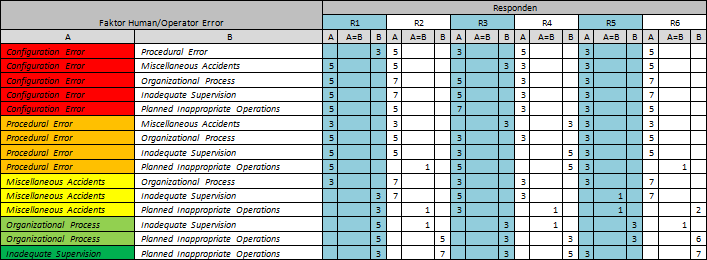
## (lanjutan)

**Hasil Kuisioner Level 3 (Faktor pada setiap Kategori Faktor)**

* 1. Software Failure

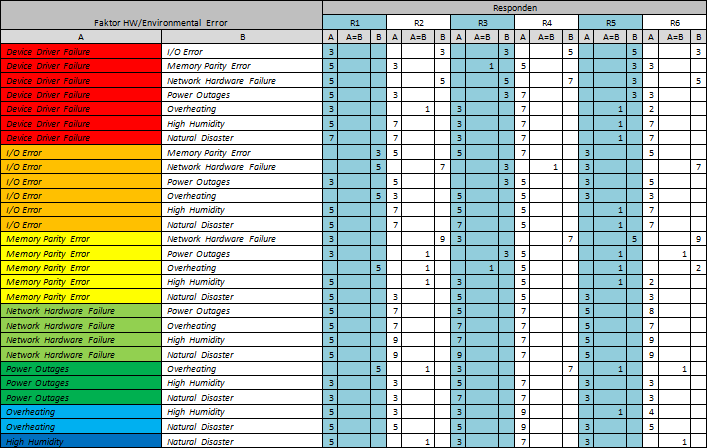


* 1. Human/Operator Error



## (lanjutan)

* 1. Hardware/Environmental Error

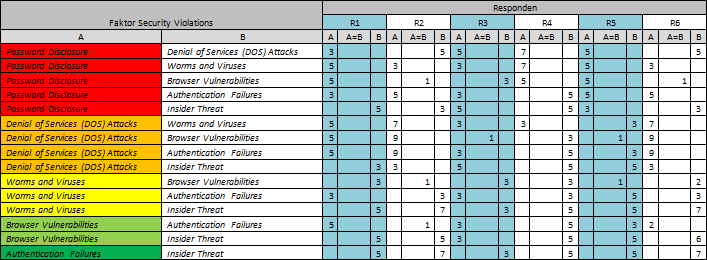


* 1. Fault Recovery



## (lanjutan)

* 1. Security Violations



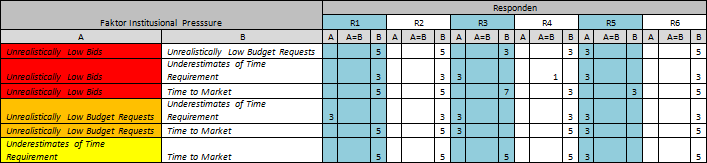
* 1. Change Management



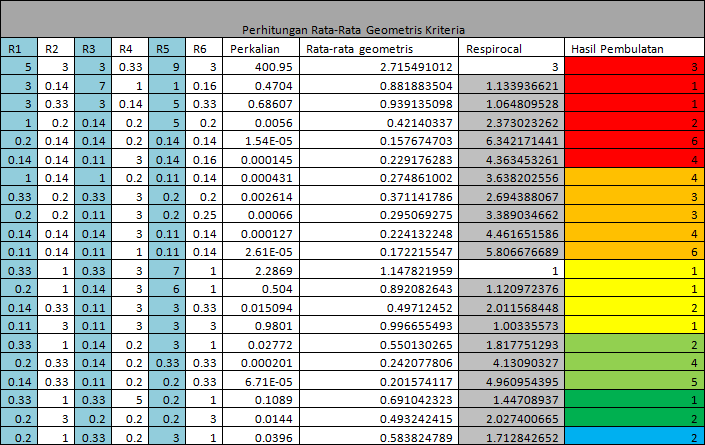
* 1. Communication Management



* 1. Institusional Pressure



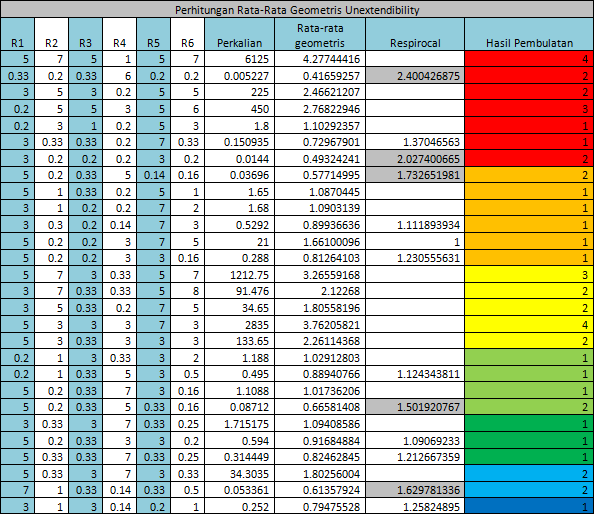
## Lampiran 9 Perhitungan Rata-Rata Geomteris Rata-Rata Geometris Kriteria Aplikasi yang Menyebabkan Insiden



**(lanjutan)**

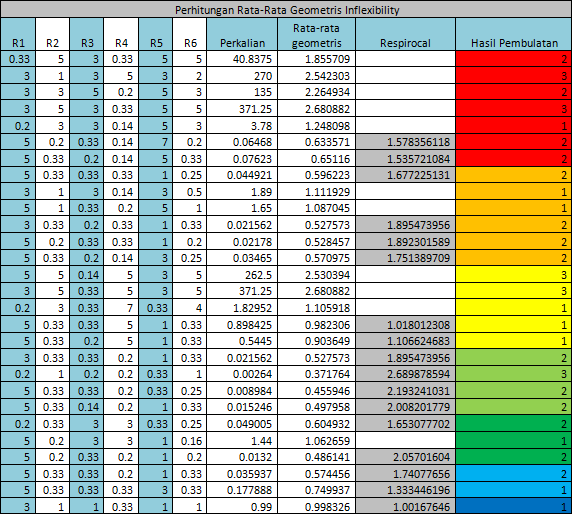
**Rata-Rata Geometris Kategori Faktor terhadap Setiap Kriteria**

1. Kriteria Unextendibility



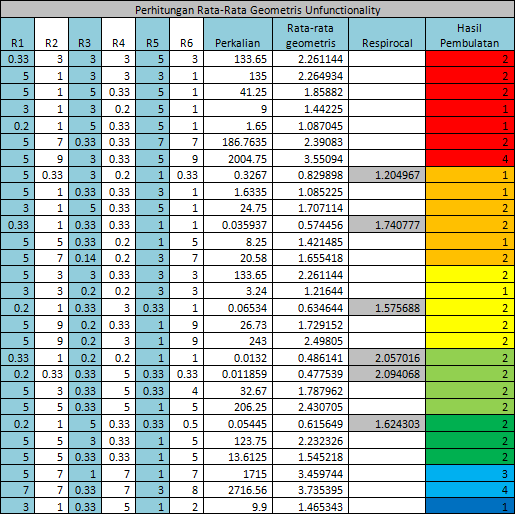
## (lanjutan)

1. Kriteria Inflexibility



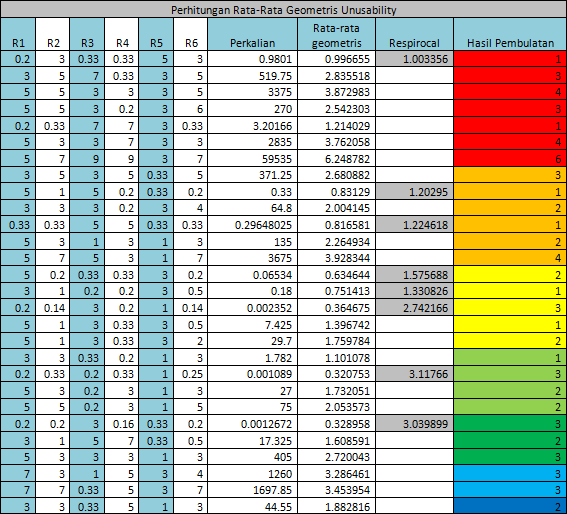
## (lanjutan)

1. Kriteria Unfunctionality



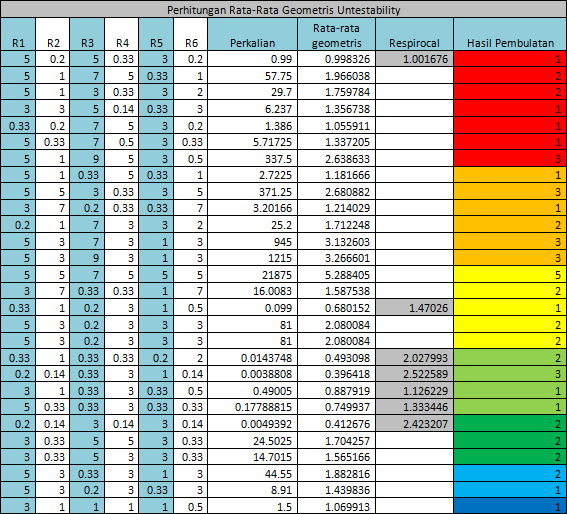
## (lanjutan)

1. Kriteria Unusability



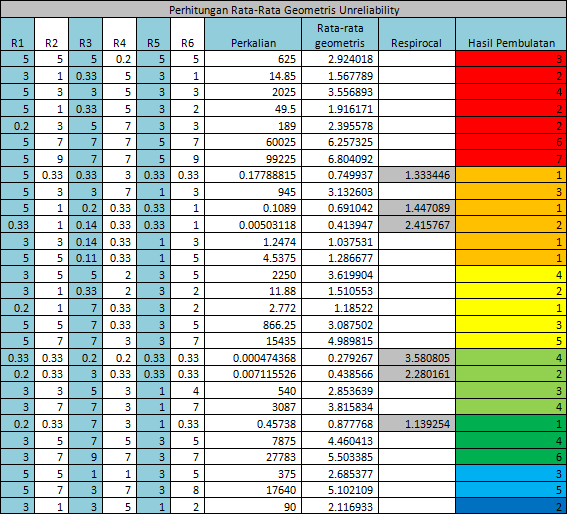
## (lanjutan)

1. Kriteria Untestability



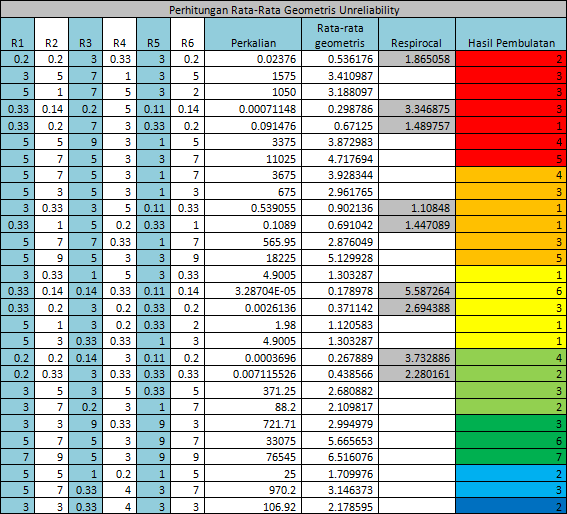
## (lanjutan)

1. Kriteria Unreliability



## (lanjutan)

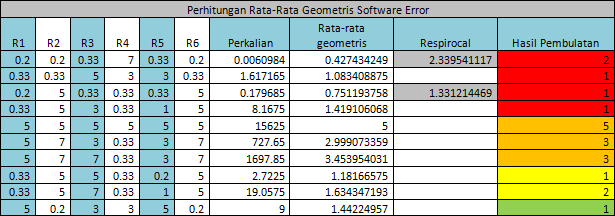
1. Kriteria Unintegrity



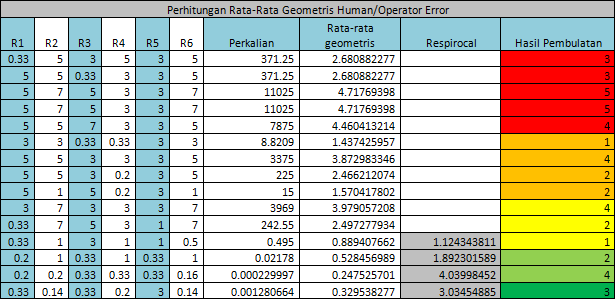
## (lanjutan)

**Hasil Kuisioner Level 3 (Faktor pada setiap Kategori Faktor)**

1. Software Failure

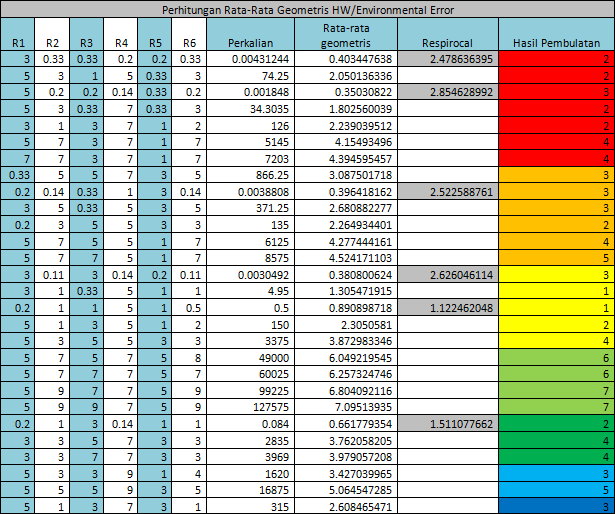


1. Human/Operator Error



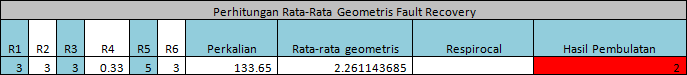
## (lanjutan)

1. Hardware/Environmental Error

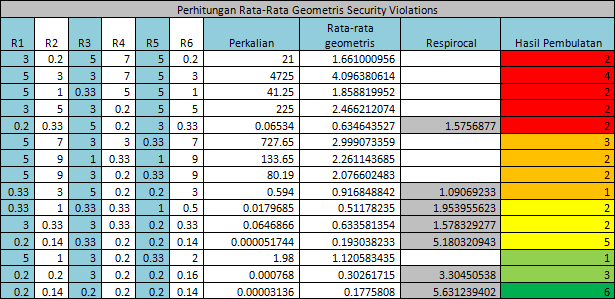


## (lanjutan)

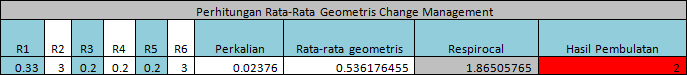
1. Fault Recovery



1. Security Violations

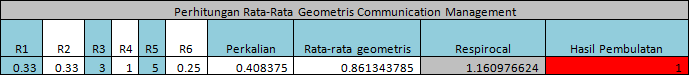


1. Change Management

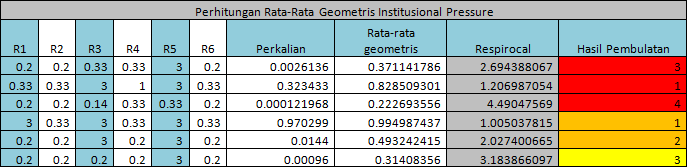


## (lanjutan)

1. Communication Management



1. Institusional Pressure



## Lampiran 10 Tranksrip Wawancara Tahap 2 (Pakar 4)

Nama : Henda S Barkah Tanggal Wawancara : 13 Desember 2016 Keterangan

P : Peneliti

R : Responden

P : Jadi, kemarin setelah menyebarkan kuisioner ternyata hasilnya Insider Threat menduduki peringkat kedua sebagai faktor yang paling berpotensi atau yang sering terjadi di Telkomsel. Menurut Bapak bagaimana?

R : Biasanya itu adalah terkait, insidernya ini harus dilihat dulu, insidernya ini apa gituloh apakah itu hanya perubahan konfigurasi, apakah itu memang ter-compromise insiden dari luar. Itu ada dua, biasanya kalau perubahan konfigurasi mereka akan konfirmasi dulu ke tim security kita, gitu.

P : Insider threat itu dari orang dalem kan, Pak? Menurut Bapak di Telkomsel apakah memang banyak terjadi?

R : Sebentar dulu, Insider Threat nya ini apa nih, seperti contohnya nih, jenis nya banyak soalnya.

P : Tolong ceritakan saja Pak tentang kasus Insider Threat yang pernah terjadi di Telkomsel

R : Tentang IT Security nya Telkomsel?

P : Bukan, tentang Insider Threat yang pernah terjadi dan gimana IT Security Telkomsel menangani kasus tersebut aja

R : Oh gitu.. Jadi kalau di tempat kita itu yang kita pegang itu namanya Managed Operation IT Security. IT Security yang ada di Telkomsel. Nah yang terkait di sini itu yang kita manajemen itu adalah server-server, yang kita

## (lanjutan)

monitor. Yang kedua itu adalah aplikasi-aplikasi, terus yang secara langsung nge hit ke bagian production dr Telkomsel.

P : Lalu celah keamanannya sehingga bisa terjadi banyak Insider Threat itu Bagaimana Pak?

## R : Kita itu monitoring nya 24 jam dengan melakukan alert conditioning di tempat.. ya kita memasang semacam spy di titik-titik itu, di aplikasinya, di homogennya, apa namanya, di servernya. Jadi setiap kali itu ada perubahan, siapapun yang meng-hit itu akan keluar alert itu di tempat kita, gitu.

P : Berarti sebenearnya sudah ada, ini ya langkah pencegahan R : Udah..

P : Tapi kenapa itu masih sering terjadi ?

R : **Nah, biasanya ketika ada alert di tempat kita, nah nanti kita punya tim semacam FBI, ya tim investigasinya, itu ada di level 1, Level 1 investigation. Nah itu mereka yang akan turun, misalkan ada perubahan di aplikasi apa, misalnya Reflex atau apa. Biasanya terjun ke orang Reflex nya, didatengin**, Pak apakah benar ada perubahan di aplikasi ini, apakah bapak yang melakukan atau tidak. Begitu kan kita tanya, path nya gini gini, masuknya dari sini, keluarnya dari sini, dijelaskan gitu kan. Oh iya pak itu aktivitas dari saya.

P : Emang untuk satu aplikasi semuanya bisa ngubah? R : Nggak lah, yang pemilik aplikasinya lah yang bisa

P : Berarti cuman pemilik aplikasinya saja yang bisa merubah ya Pak.

## (lanjutan)

R : Iya, custody nya saja. Makanya kita raise nya ke custody. Gitu, kalau sysadmin, ya ke sysadmin.

P : Yang biasanya mereka ubah itu apa ya Pak?

## R : Itu dia, path-path nya, tergantung mereka nanem apa, tergantung biar mereka bisa naro di aplikasi, suatu semcam kayak file naro di situ, gituloh. Tergantung, mereka bisa di aplikasi mereka menambahkan file atau apa segala macem.

P : Oh itu terserah mereka gitu ya? Sesuai kebutuhan mereka?

## R : Iya kebutuhan mereka. Tapi itu ntar terkonfirmasi memang dilakukan sama mereka. Tapi kalau misalnya penanaman nya itu bukan dari mereka itu kita jadikan insiden.

P : Itu sering, Pak?

## R : Ohh.. ada sering, sebulan sekali lah paling.

P : Lalu, tindakan nya apa Pak?

## R : Tindakannya itu langsung Lvel 2 turun ke sana. Level 2 jadi kalau di IT security ada 24x7 monitoring itu level 1, tugasnya memberikan analisa alert-alert yg masuk, kemudian raise ke custody, setelah raise ke custody dia dapet konfirmasi terus dia melakukan summary juga analisa di tingkat awal ini insiden apa bukan. Kalau itu insiden, maka itu nanti Level 2 yang akan turun. Level 2 itu misalkan ada brute force gitu di dalam aplikasi kita, biasanya mereka itu akan melakukan reset passwordnya. Kita itu ada tindakan gitu, ada preventif, forensic, terus detektif yang kita lakukan itu jadi kita lakukan secara menyeluruh.

**(lanjutan)**

P : Itu seringnya kan pas ketahuan, oh ini dari yang bukan punya aplikasi lalu diselidiki lagi kenapanya , ada contoh root cause nya Pak?

R : Ada banyak laporannya, tapi saya ga bisa share. P : Baik Pak, kalau contoh kasus nya saja ada Pak?

## R : Oh..Misalnya kayak kemaren, kasus dari MKios lah ya. Tiba-tiba akunnya tinggi banget, dia bisa transaksi sampai 1 M. Dia bisa beli pulsa sampe 1 M. Ternyata yang ngelakuin salah satu vendor di sini. Nah itu sebelumnya kita selidiki, ini bener gak nih transaksinya, terus kita cek 3 bulan ke belakang ritme nya dia. Ritme pembeliannya dia. Kalau misalkan ini ter-compromise misalkan dia ini ada kecurigaan dari hasil penyelidikan ya, kita udah kuat gitu dari segi ininya kita blok nomor itu

P : Kalau route cause nya dia bisa transaksi gede gitu gimana Pak?

R : Nah itu dia, kita punya alert nya. Kita ada alertnya di sini. Alert yang mengarah ke Mkios itu. Kita pasang semacam sirene lah di situ. Kalau misalkan dia transaksinya jomplang ada spy kita udah tau.

P : Nah itu ternyata kenapa dia bisa transaksi nya tinggi?

R : Iya kita lihat dulu, legitimate nggak. Sesuai prosedur nggak, by system nya sesuai prosedur nggak. Kemarin, ada yang memang sesuai prosedur. Ada juga yang tidak sesuai prosedur, kita langsung blok saja.

P : Oh langsung diblok saja ya Pak.

R : Iya, diblock aja. Ntar kalau misalkan mereka ini kan mereka bakal *nyap* nyap ke kita. Maksudnya ngomel-ngomel ini kenapa transaksi kita diblok gitu kan.

P :Tapi penyebab nya mereka bisa tiba-tiba punya transaksi tinggi?

## (lanjutan)

R : Iya biasanya mereka itu, biasanya ada semcam flick di luar itu, ya dia bisa pasang tunnel, terowongan lah untuk nge hit langsung ke production nya kita, untuk nambah segala macem. Ya banyak sih sebab-sebabnya. Ada lagi yang tahun kemarin itu, yang merubah apa ya.. lupa yang ditangkep itu dari salah satu vendor itu, tahun 2014 akhir. Duh saya lupa detailnya, soalnya banyak banget kasusnya neng tercompromise nya berbeda-beda. Kalau dari insider biasanya kebanyakan ngubah path-path itu segala macem. Kalau missal ada macem-macem ya langsung kita tangkep, soalnya udah ada ketahuan, ya tugas kita sebenarnya nangkep orang. **Kalo konfigurasi dari IT Security itu kan dari 24x7 itu kana da 4 domain service lagi yang mensupport 24x7 itu. Yang pertama itu ada insiden monitoring services, yang kedua itu adalah security assessment, itu yang melakukan VA dan hardening. Gunannya itu kalau VA mencari celah aplikasi, terus yang hardening itu ibaratnya menutup segala kemungkinan akses ke suatu aplikasi. Terus yang ketiga itu di dalam itu ada Security Development, biasanya ini biro namanya itu biro developer untuk mengembangkan alert-alert baru dari kalo serangan itu kan dia itu tercompromise nya data nya tuh beda-beda ya, kalau ada serangan-serangan baru, nah developer ini tugasnya nangkep, untuk mengcreate lagi alert-alert baru. Nah yang ke empat ada user access management, kita melakukan yang namanya mapping user access ke semuanya ke sistem telkomsel itu harus dari User Access Management. Itu aja kalau contoh kasusnya banyak, tapi gak bisa dishare.**

P : Baik Pak, itu saja dulu. Terimakasih waktunya Pak

## Lampiran 11 Tranksrip Wawancara Tahap 2 (Pakar 5)

Nama : Isa A Susianto Tanggal Wawancara : 13 Desember 2016 Keterangan

P : Peneliti

R : Responden

P : Jadi setelah kemarin diolah hasil kuisioner nya dari 6 responden itu, faktor yang paling berpotensi atau paling sering menyebabkan insiden itu adalah yang pertama computational/logic error, kemudian insider threat, improper testing of change, configuration error, sama network hardware failure. Jadi, di sini saya membutuhkan pendapat Bapak terkait peringkat ini. Mulai dari yang pertama, kenapa computational/logic error bisa sering terjadi di sini dan tipe-tipe insiden yang disebabkan oleh faktor ini bagaimana?

R : Computational/Logic Error ini sebenernya, sorry kalau dia ada perubahan masuknya kemana?

P : Ke change management, masuknya improper testing atau improper planning of change. Kalau Logic error itu bugs.

## R : Iya bug. Ya karena kita apa sih, karena kita di IT kan memang banyak application based ya service nya di IT. Seiring dengan kompleksitas produk kita yang bertambah terus kompleksitas integrase antara satu aplikasi dengan aplikasi lainnya makin ditambah pasti bakal banyak bugs yang ditemukan. Karena lebih ke implementasinya sih,karena misalkan satu produk kan kita banyaknya produk kita beli ya. Pasti kita akan kustomisasi sesuai dengan kebutuhan telkomsel, dan kebutuhan telkomsel itu mengikuti kebutuhan bisnis, nah itu sih sebenernya yang sering menyebabkan insiden. Misal bisnis maunya A, kita punya produk

**(lanjutan)**

**X, sementara untuk memenuhi kebutuhan bisnis, produk ini harus di kustomisasi. Nah itu yang mungkin menyebabkan bugs, terutama yang terkait integrase antar aplikasi.**

P : Ini juga terkait sama yang configuration error ya?

## R : Bedanya kalau configuration itu misalkan gini, kenapa itu sering terjadi soalnya yang tadi missal ada integrase A dan B. Nah itu pasti konfigurasinya harus match kan antara A dan B, misal yang simple aja, konfigurasi timeout. Timeout A dan B kalau dia saling terhubung kan. Terus ada lagi aplikasi di belakangnya ada C, D. Konfigurasinya harus match kan, gak boleh misalkan aplikasi depannya ini timeout nya lebih kecil daripada yang di belakang itu gak bisa atau dan kita harus tau behavior tiap aplikasi untuk set up konfugurasi yang pas. Nah di Telkomsel ini masih sering apa ya bukan trial dan error tapi lebih sering, kita punya analisis terus kita apply, ternyata ini kurang pas nih, terus kita ubah lagi. Kalau konfigurasi itu gitu sih.

P : Kalau yang computational/logic error tadi?

R : Lebih ke ya memang bugs di aplikasinya, terutama saat integrasi dengan aplikasi lain.

P : Berarti sebenarnya computational/logic error, changes, dan config ini terkait ya Pak?

## R : Sebenarnya bisa jadi keterkaitan, kalau changes itu yang pasti kalau misalkan ada perubahan, OK di aplikasi A dilakukan perubahan, nah aplikasi A ini berhubungan dengan aplikasi B nah kita suka miss bahwa ternyata impact juga ke aplikasi B gitu. Bahwa impact nya ke surrounding nya juga. Nah mungkin di situ, kalau di sisi testing nya

**(lanjutan)**

**kadang-kadang bisa jadi miss. Jadi misalnya kita scope testing nya harus jelas kan kalo ada perubahan itu, misalkan aplikasi surrounding nya apa aja, ketika surroundingnya banyak itu akan lebih memunculkan kemungkinan bahwa aka nada test case yang miss di situ, belum lagi kalau untuk point testing nya kalau yang aku lihat sih pengadaan resource testingnya kayak environment itu nggak ideal lah, jauh dari ideal lah kalau di Telkomsel. Kayak kita gak punya testing environment yang sesuai dengan production, kita gak semua aplikasi punya gitu. Jangankan yang sesuai dengan production, yang aplikasi yang punya testing environment aja gak semuanya kan. Nah itu sih sebenernya. Salah satu poin yang besar juga terkait environment.**

P : Secara garis besar, Bapak juga setuju kan ya dengan lima faktor teratas ini?

R : Iya setuju.

P : Kalau tentang insider threat bagaimana Pak?

## R : Aku sih taunya waktu jaman di problem itu memang banyak terjadi kasus fraud, baik dari luar atau pun dari dalam, ya memang security di kita juga gak bisa dibilang, ya maksudnya gini lah semakin canggih security di kita semakin banyak juga cara membobolnya. Memang security threat ini banyak banget sih. Lumayan banyak lah kasus di Telkomsel semenjak aku ada di Telkomsel. Gitu, aku nemuin waktu aku di Problem. Beberapa kayak ada orang yang masuk ke server kita lah atau yang ngubah-ngubah config kita.

P : Itu dari orang dalam Pak?

R : Ada yang orang dalam, ada orang luar. P : Kalau terkait Network Hardware Pak?

## (lanjutan)

R : Kalau Network ini masuk nya ke Datacomm itu kan. Network ini kan ada yang part of IT ada yang di luar IT.

P : Kita itu punya network sendiri atau menggunakan managed service?

R : Kita punya sendiri. Kalau untuk service nya managed service cuman hardware nya punya sendiri

P : Oh.. Hardware nya punya kita tapi yang mengurusi nya vendor?

R : Iya, tapi masih under supervise nya orang Telkomsel juga sih**. Sepengalaman ku kalau network ini banyak juga yang terkait dengan bugs sih, bug di perangkat network nya di firewall kah, di load balancer kah, tapi sebenernya bug bug juga sih, masalah config juga ada.**

P : Dalam sebulan ada berapa kejadian-kejadian ini Pak?

## R : Sebulan ya? Seminggu itu rata-rata bisa sekali kalau bugs. Kalau security violations kayak insider threat, atau DOS ini iya sebulan paling sekali tapi dampak nya besar, even dia cuma sekali. Mungkin dalam waktu 6 bulan cuman dua kali misalkan tapi impact nya besar. Yang banyak kecil-kecil itu masalah config.

P : Ketahuan nya?

R : Ketahuan nya ya itu bisa jadi di monitoring kita , bisa jadi pengalaman pelanggan, complaint. **Kalau yang config itu kecil-kecil sih biasanya, cuman contohnya ada sih cuman yang ini gak kecil waktu kemarin. Ada konfigurasi timeout, err bukan timeout juga sih, session gitu lah antara load balancer sama reflex aplikasi. Na itu efeknya ya banyak transaksi yang gagal karena session nya udah full. Jadi karena konfigurasi jumlah sessionnya yang gak match gitu kan.**

P : Kalau masalah improper testing of change?

## (lanjutan)

R : Kalau testing..Kalau standard change masuk gak ya? Kan kamu tau sendiri kalau issue nya testing itu karena standard change kan. Which is emang gak dites, dites tapi ya sama internal gak sama tim testing. Itu case nya banyak. Kalau untuk normal change, setau aku tahun 2016 sampe aku menjabat di testing itu ada gak ya insiden, gak ada deh.. kalau issue kalo gak salah ada. Kalau yang standard change banyak, tapi data nya aku gak tau pasti. Karena standard kan internal testingnya.

P : Kalau faktor configuration? R : Kenapa jumlahnya?

P : Jumlah sama contoh kasusnya.

R : Contoh kasusnya yang itu, yang reflex sama load balancer.

P : Berarti antara computational/logic error dan config ini terkait banget ya Pak?

R : Nggak, nggak maksud aku ada juga case config di situ, yang bugs itu ya banyak ya aplikasi nya nge hang, instance nya nge hang, itu banyak, kemungkinan nya banyak gitu lah

P : Berarti salah mismatch session tadi di config ya Pak?

R : Iya itu di config, karena itu konfigurasi levelnya. Kalau bug itu misalkan ya itu hang lah misalkan aplikasi nya atau instance nya nge hang, cuman kalo insiden sih udah jarang, issue yang banyak. Overall sekarang insiden lagi sedikit, kalau issue banyak. **Contohnya yang nyebabin insiden itu kayak misalkan ada instance yang nge hang di aplikasi CRM, dia jadi stuck order, jadinya issue. Sebenernya di kita itu issue insiden juga sih, tapi minor insiden. Kalau insiden yang bener-bener di broadcast itu major incident.**

## (lanjutan)

P : Kalau masalah testing kebanyakan di standard change karena itu internal testing ya Pak?

R : Iya, jadi ya itulah f**aktor nya bisa jadi karena environment, karena waktu, waktunya yang diburu-buru**

P : Network Hardware Pak?

R : Perangkatnya itu router gitu-gitu kan? **Contohnya kalau network itu misalkan FO cut. Fiber Optik. Fisik nya putus. Dulu sering jaman 2014-2015. 2015 paling sering. Sekarang udah jarang.**

P : Kenapa bisa putus?

## R : Macem-mace, kadang-kadang, karena putus sendiri, karena kan ada yang di dalem laut kan, ada yang di dalem tanah atau kecangkul. Dulu itu sering banget.

P : Tapi hardware nya ini punya kita diurusin sama vendor under supervise Telkomsel?

## R : Kalau FO kita pake punya Telkom, karena itu milik bersama. Ada lagi masalah port, misalkan port di router atau di switch itu kan ada portnya, portnya bermasalah, portnya rusak akhirnya kita ganti port aja. Nah itu kan kita juga gak tau penyebabnya apa, ya mungkin karena, ya hardware problem lah itu

P : Menurut Bapak, ada kaitan nya tidak karena service network nya di managed oleh vendor hardware nya sering failure?

## R : Kalau menurut ku sih nggak ya, karena si vendor pun under supervisinya orang Telkomsel juga sih. Kalau menurut ku memang ke perangkatnya. Mungkin ada masalah apa gitu, ya kerusakan perangkat.

**(lanjutan)**

P : Perangkat di Telkomsel sering diupgrade?

## R : Iya, biasanya kalau umurnya udah berapa tahun gitu ya bisa diupgrade. Persisnya berapa tahun tergantung produknya, ada setahun, ada yang lima tahun. Tergantung kebutuhan, tergantung rekomendasi dari yang punya produknya, tergantung teknologi juga, kalau misalkan ada teknologi baru yang Telkomsel rasa perlu, diganti.

P : Ada perangkat yang sudah lama sekali tapi nggak pernah diganti? R : Kayaknya selalu ada refresh deh

P : Okay, segitu aja Pak. Terimakasih banyak atas waktunya