

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Kajian pustaka yang digunakan pada penelitian ini adalah penelitian terdahulu oleh Nistala & Kumari yang berjudul “*An Approach to Carry Out Consistency Analysis on Requirements*”. Pada tabel 2.1 kajian pustaka 1 dijelaskan bahwa penelitian Nistala & Kumari bertujuan untuk mengukur kesesuaian perancangan menggunakan metode *consistency analysis*

Tabel 2. 1 Kajian Pustaka 1

Metodologi	Hasil
Penelitian ini bertujuan untuk mengukur dan mengetahui kesesuaian sebuah perancangan sistem terhadap kebutuhan fungsional sistem pada sebuah studi kasus industri. Penelitian ini berisi penjelasan dan uraian kerangka kerja “ <i>Consistency Analysis</i> ” meliputi istilah keselarasan kebutuhan, struktur konfigurasi kebutuhan, konsistensi analisis kebutuhan, matriks inkonsistensi kebutuhan, dan indeks konsistensi kebutuhan. Serta, penerapan metode kedalam sebuah studi kasus	Hasil studi kasus menunjukkan bahwa inkonsistensi persyaratan yang signifikan dapat ditemukan melalui kerangka struktur konfigurasi kebutuhan. analisis konsistensi kebutuhan menghasilkan kelengkapan dari beberapa item yang tidak sesuai. Jalur konsistensi dari tujuan, proses dan kebutuhan bisnis dapat divalidasi dan dianalisis melalui kerangka ini. Perhitungan indeks konsistensi bernilai 48% membuktikan rendahnya konsistensi sistem dengan kebutuhan pada studi kasus tersebut

Penelitian selanjutnya dari Kamalrudin & Sidek (2015) berjudul “*A Review On Software Requirements and Consistency Management*” yang dijelaskan pada tabel 2.2 kajian pustaka 2. Penelitian tersebut bertujuan untuk meninjau terhadap definisi 3C yang merupakan *correctness*, *completeness*, dan *correctness* terhadap kebutuhan sistem.

Tabel 2. 2 kajian pustaka 2

Metodologi	Hasil
Penelitian ini bertujuan untuk meninjau dari definisi 3C dan pemahamannya. Selanjutnya penelitian ini menjelaskan mengenai tinjauan menyeluruh terkait teknik pengelolaan konsistensi yang telah diidentifikasi. Penelitian ini mengidentifikasi berbagai kesenjangan yang ada dalam proses memvalidasi dan mengelola konsistensi kebutuhan untuk menghindari penemuan kembali. Penelitian ini didukung dengan representasi <i>heat map</i> terkait jenis kontribusi, teknik, spesifikasi dan semantik yang digunakan dalam manajemen konsistensi.	Hasil dari penelitian tersebut berupa pembahasan gagasan 3C konsistensi, kebenaran, dan konsistensi. Dalam pengelolaan konsisten <i>heat map</i> digunakan untuk menyajikan grafik dan dilakukan penyederhanaan dan mengelompokkan jenis kontribusi, spesifikasi, semantik dan teknik yang digunakan dalam konsistensi. Selain itu juga membandingkan pendekatan yang ada dan mengidentifikasi kelemahan dan kelebihan an pendekatan tersebut.

Penelitian terakhir dari Naung & Oo (2014) berjudul “*Information System Requirement Gathering using FAST Framework:Critical Analysis*” yang dijelaskan pada tabel 2.3 kajian pustaka 3. Penelitian tersebut menjelaskan bagaimana mengumpulkan kebutuhan sistem dalam pengembangan sistem menggunakan FAST.

Tabel 2. 3 kajian pustaka 3

Metodologi	Hasil
Penelitian ini bertujuan untuk mengumpulkan dan menganalisis kebutuhan dari sebuah sistem menggunakan kerangka FAST. Pada FAST penggunaan fase analisis kebutuhan dilakukan pada fase definisi lingkup, analisis permasalahan, analisis kebutuhan, dan desain logis	Penelitian tersebut menjelaskan bahwa pengumpulan kebutuhan merupakan proses yang vital karena dapat mempengaruhi perubahan biaya dan waktu. Penerapan pengumpulan kebutuhan pada fase FAST terbatas hingga fase desain logis untuk memodelkan fungsi pada sistem yang akan dikembangkan

2.2 Analisis dan Perancangan

Membuat sistem yang kompleks dibutuhkan analisis dan perancangan yang baik agar dapat meningkatkan nilai dari sistem yang dibuat atau dikembangkan. Analisis dan perancangan sistem merupakan kegiatan yang dilakukan sebelum membuat sistem. Tujuan dari analisis dan perancangan sistem untuk menentukan kebutuhan, permasalahan yang dapat diatasi dari adanya sebuah sistem yang akan dibangun, dan sistem seperti apa yang akan dibuat (Whitten & Bentley, 2007).

2.2.1 Analisis Sistem

Dalam pengembangan atau pembuatan sistem analisis sistem merupakan salah satu faktor penting yang berpengaruh dalam menentukan kualitas sebuah sistem yang ingin dikembangkan atau dibuat. Kualitas sistem yang baik dikembangkan berdasarkan analisis permasalahan dan kebutuhan yang baik.

Menurut Whitten & Bentley (2007) analisis sistem merupakan studi domain masalah bisnis untuk rekomendasi perbaikan dan spesifikasi persyaratan dan prioritas bisnis untuk solusi. Analisis sistem ditujukan untuk menyediakan pengembang dalam pemahaman secara menyeluruh terhadap masalah masalah dan kebutuhan yang memicu sebuah sistem. Informasi yang didapat terkait dengan kebutuhan sistem dipelajari dan dianalisis untuk memperoleh pemahaman yang lebih detail mengenai apa yang bekerja, apa yang tidak bekerja, dan apa yang dibutuhkan dalam sebuah sistem yang ingin dikembangkan.

2.2.2 Perancangan sistem

Setelah proses analisis sistem dilakukan hasilnya akan diterapkan pada desain atau perancangan sistem. Perancangan sistem dibuat berdasarkan kebutuhan, persyaratan, maupun informasi lain yang berkaitan dengan sistem yang telah dianalisis.

Menurut Whitten & Bentley (2007) perancangan atau desain sistem merupakan spesifikasi konstruksi solusi yang teknis dan berbasis computer untuk persyaratan bisnis yang diidentifikasi dalam analisis sistem. Kebanyakan perusahaan harus memilih antara membeli solusi yang cukup bagus atau membangun solusi kustom sendiri. Setelah alternatif teknis dipilih dan disetujui fase perancangan atau desain sistem mengembangkan cetak biru dan spesifikasi teknis yang dibutuhkan untuk mengimplementasikan database, program,

antarmuka pengguna dan jaringan sistem informasi yang akan diterapkan, pada saat dimana kita memilih untuk membeli perangkat lunak atau sistem daripada membangunnya, cetak biru berperan dalam menentukan bagaimana perangkat lunak yang akan dibeli dapat diintegrasikan ke dalam bisnis dengan sistem informasi lain. Dalam perancangan sistem setidaknya dapat menacakup dua aspek, *dynamic view* ataupun *static view*. *Dynaic view* biasanya memodelkan tingkah laku atau state sistem dari waktu ke waktu. *Static view* memodelkan komponen sistem yang tidak berubah pada umumnya semisal sebuah obyek atau kelas yang dimiliki sistem (Rumbaugh, et al., 2005).

2.3 Sistem Informasi

Sistem Informasi merupakan suatu sistem di dalam organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari sebuah organisasi untuk menyediakan kepada pihak tertentu dengan laporan laporan yang diperlukan (Sutabri, 2016)

2.4 Pemrograman Berorientasi Obyek

Metodologi berorientasi obyek merupakan strategi pembangunan sistem yang mengorganisasikan sistem sebagai kumpulan obyek yang berisi data dan operasi yang diberlakukan terhadapnya (Sukamto & Shalahuddin, 2014). Pendekatan berorientasi obyek akan memandang sistem yang akan dibangun sebagai sekumpulan obyek yang berkorespondensi dengan obyek obyek di dunia nyata.

2.5 Metode FAST (*Framework Application of System Thinking*)

Framework Application of System Thinking atau FAST merupakan kerangka kerja cerdas yang cukup fleksible untuk menyediakan tipe tipe berbeda proyek maupun strategi dan berisi gabungan dari praktik praktik penggunaan metode pengembangan sistem yang dapat ditemui dalam banyak metode refensi dan komersial (Whitten & Bentley, 2007).

Pada setiap metode pengembangan sistem menggunakan prinsip prinsip dasar yang mendukung keberhasilan dan kinerja sebuah pengembangan sistem. Prinsip prinsip mendasar metode FAST terdiri dari keterlibatan pengguna sistem, penggunaan pendekatan untuk memecahkan masalah, pembentukan fase dan aktivitas, dokumentasi, pembentukan standar, pengelolaan proses da proyek, penerapan sistem informasi sebagai investasi modal, pembatalan atau revisi lingkup, pembagian untuk memppermudah pemecahan masalah, dan desain sistem untuk pertumbuhan dan perubahan

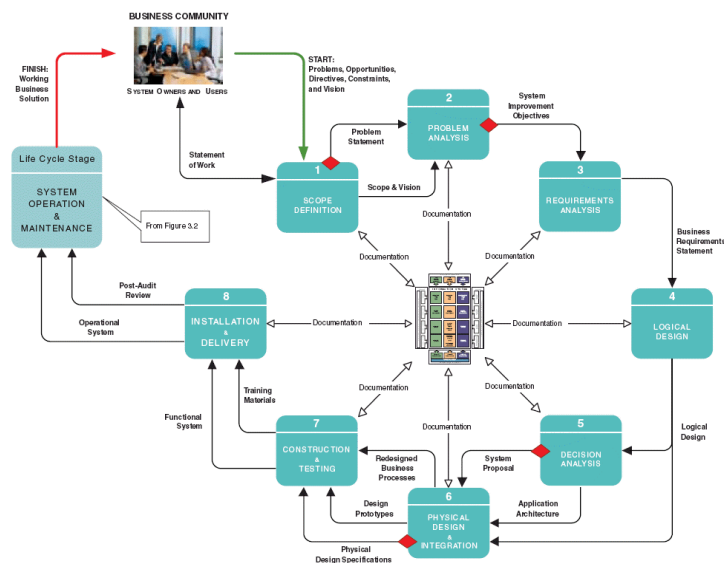
Tabel 2. 4 Fase FAST

FAST Phases	Classic Phases			
	Project Initiation	System Analysis	System Design	System Implementation
Scope Definition	x			

Problem Analysis	x	X		
Requirement Analysis		X		
Logical Design		x		
Decision Analysis	(A system analysis transition phase)			
Physical Design and integration			X	
Construction and Testing			X	
Instalation and Delivery				X

Sumber: Whitten & Bentley (2017)

FAST terdiri dari beberapa fase, tiap fase menghasilkan produk jadi yang selanjutnya digunakan dalam mengerjakan fase berikutnya. Produk yang dihasilkan pada tiap fase didokumentasikan untuk membantu proses pengembangan. Jumlah fase yang digunakan sebanyak 8 fase meliputi, Fase Analisis dan Perancangan (Definisi Lingkup, analisis masalah, analisis kebutuhan/persyaratan, desain logis), fase peralihan (analisis keputusan), dan fase implementasi (desain dan integrasi fisik, konstruksi dan pengujian, dan instalasi dan pengiriman).



Gambar 2. 1 Proses pengembangan pada FAST

Sumber: Whitten & Bentley (2007)

2.6 Scope Definition

Fase pertama pada metode FAST yaitu Definisi Lingkup atau *Scope Definition*. Tujuan dari fase ini yaitu untuk menjawab “Apakah proyek ini pantas diperhatikan?” dan untuk mengasumsikan bahwasannya masalah tersebut perlu diperhatikan. Fase ini menentukan ukuran dan batas batas proyek, visi proyek,

semua batasan atau *limit*, partisipan proyek yang dibutuhkan, anggaran, dan jadwal (Whitten, et al., 2007).

Fase ini menyimpulkan apakah sistem ini dapat dijalankan atau tidak dari pemilik sistem. Apakah para pemilik sistem setuju dengan lingkup, anggaran, dan jadwal proyek yang diusulkan. Atau pemilik sistem harus mengurangi lingkup (untuk mengurang waktu dan biaya) atau membatalkan proyek. Produk akhir yang dihasilkan dan paling penting adalah *statement of work* atau pernyataan kerja yang merupakan kontrak atau persetujuan untuk mengembangkan sistem informasi. Produk ini mengkonsolidasi kan pernyataan masalah, pernyataan lingkup, jadwal, dan anggaran untuk semua pihak yang terlibat dalam proyek. Beberapa hal yang dilakukan dalam fase ini:

1. Identifikasi masalah dan peluang.

Tugas paling penting dari definisi lingkup adalah membuat dasar permasalahan, kesempatan, dan atau arah dari proyek yang akan dibangun. Hal tersebut dinilai berdasarkan urgensi, visibilitas, dan prioritas .

2. Negosiasi lingkup dasar.

Identifikasi batasan proyek, menentukan aspek bisnis yang termasuk maupun tidak termasuk dalam proyek. Lingkup pada sebuah proyek dapat berubah ubah selama proyek berlangsung sehingga perlu rencana proyek awal yang ditetapkan sebagai landasan proyek.

3. Menilai kelayakan proyek.

Hal ini akan menjawab pertanyaan apakah proyek layak untuk dilanjutkan. Pada tahap ini berisi perkiraan terbaik dalam memperkirakan solusi permasalahan, pemanfaatan peluang, atau memenuhi arah proyek sesuai dengan biaya yang digunakan untuk mengembangkan sistem.

4. Mengembangkan jadwal awal dan anggaran.

Apabila proyek layak dilanjutkan. Kita dapat mengembangkan rencana proyek yang berisi pendahuluan rencana proyek termasuk penjadwalan, rencana kerja proyek. Rencana proyek tersebut akan diperbarui setiap akhir pengerjaan fase pengembangan sistem.

5. Komunikasi rencana proyek

Mengawasi rencana proyek agar sesuai dengan kesepakatan diawal sehingga perencanaan proyek ditetapkan sebagai prioritas tertinggi untuk landasan pembangunan proyek

2.6.2 Analisis PIECES

Fase definisi lingkup merupakan fase penting dalam sebuah perancangan sebuah sistem. Menurut Whitten & Bentley (2007) untuk mendukung analisis dan identifikasi kebutuhan dari sebuah sistem dapat menggunakan analisis PIECES. Analisis PIECES merupakan kerangka kerja yang digunakan untuk memperbaiki atau mengoreksi permasalahan yang ada berdasarkan kategori yang disebutkan

dalam tiap hurufnya *Performance, Information, Economic, Control, Efficiency, Service* (Whitten & Bentley, 2007). Dari permasalahan yang telah diidentifikasi dapat diambil beberapa permasalahan yang dihadapi untuk selanjutnya permasalahan tersebut dapat dipahami. Berikut merupakan daftar identifikasi permasalahan, kesempatan, dan perintah dari setiap huruf pada PIECES.

1) *Performance* (kinerja)

Unsur *performance* ini memiliki peran penting untuk menilai apakah proses atau prosedur yang ada masih mungkin ditingkatkan kinerjanya, dan melihat sejauh mana dan seberapa handalkah suatu sistem informasi dalam berproses untuk menghasilkan tujuan yang diinginkan. Hal-hal yang menjadi ukuran dalam *performance* yaitu:

- a) *Throughput*, yaitu jumlah pekerjaan/ *output/ deliverables* yang dapat dilakukan/ dihasilkan pada saat tertentu, bagian ini mendeskripsikan situasi mengenai jumlah pekerjaan yang dibutuhkan untuk melakukan kerja tertentu (dalam satuan waktu/ Jumlah orang).
- b) *Response time*, yaitu waktu yang dibutuhkan untuk menyelesaikan serangkaian kegiatan untuk menghasilkan *output* tertentu. Bagian ini mendeskripsikan keadaan waktu respon yang terjadi ketika proses transaksi masuk hingga proses transaksi tersebut direspon untuk diproses. Penundaan ini terjadi dapat diakibatkan oleh antrian transaksi sebelumnya.

2) *Information* (informasi)

Menilai apakah informasi mempunyai nilai guna untuk pengguna dalam hal konten, ketepatan waktu, akurasi, dan format informasi. Dalam hal ini dapat diukur dengan:

- a) Masukan (*inputs*): keperluan memasukkan suatu data, hingga di mana data disimpan.
 - Data tidak dimiliki, data apa yang tidak dimiliki, dampak dan penyebab
 - Data tidak dimiliki pada waktu yang tidak tepat. data apa yang tidak dimiliki, dampak dan penyebab
 - Data tidak dimiliki secara akurat. Data apa yang tidak dimiliki secara akurat, kriteria keakuratan, dampak dan penyebab
 - Data sulit dimiliki. data apa yang sulit di dimiliki
 - Data yang dimiliki berlebihan
 - Data yang dimiliki bersifat ilegal
- b) Keluaran (*outputs*): terjadinya produksi hasil keluaran berupa tampilannya.
 - Kurangnya informasi yang dibutuhkan atau relevan

- Terlalu banyak informasi yang dimiliki sehingga terkesan tidak rapi dan berserakan
 - Informasi tidak dalam format yang bermanfaat. Informasi sudah tersedia sehingga menyebabkan kesulitan dalam pemahamannya
 - Informasi tidak akurat, tidak sesuai dengan kenyataan
 - Informasi sulit diproduksi dapat diakibatkan oleh data yang tidak lengkap ataupun sumber informasi yang sulit di dapat
 - Informasi yang tidak tepat waktu dan kondisinya. Informasi yang diterima tidak sesuai kondisi yang dialami
- c) Data tersimpan (*data store*) proses dari bagaimana sistem menyimpan data dan informasi yang dimiliki
- Data tersimpan secara berlebihan pada banyak file maupun database yang dimiliki
 - Item data yang dimiliki sama namun memiliki nilai yang berbeda (integrasi data buruk)
 - Data yang disimpan tidak akurat (tidak sesuai dengan kenyataan) dampak dari data yang tidak akurat dan penyebabnya seperti apa
 - Data tidak aman, resiko terjadinya kecelakaan tinggi atau mudah dirusak
 - Data tidak dikelola dengan baik
 - Data tidak fleksibel atau sulit untuk memenuhi kebutuhan informasi baru dari data tersimpan
 - Data tidak dapat diakses

3) *Economic* (ekonomi)

Menilai apakah prosedur yang ada saat ini masih dapat ditingkatkan manfaatnya (nilai gunanya) atau menurunkan pengelolaan biaya penyelenggaraannya.

a) Biaya

- Biaya terkait dana yang dikeluarkan untuk informasi, proses bisnis, dan pengambilan keputusan tidak diketahui jumlahnya
- Biaya tidak dapat dilacak dari sumbernya sehingga tidak diketahui keakuratan biayanya
- Biaya yang dikeluarkan untuk produksi informasi, melakukan proses bisnis, dan pengambilan keputusan terlalu tinggi

b) Keuntungan

Keuntungan atau manfaat yang didapat dari penerapan sistem informasi dalam menjalankan proses bisnisnya.

- Pangsa pasar baru dapat dieksplorasi
- Dapat memperbaiki pemasaran
- Dapat meningkatkan pesanan

4) *Control* (pengendalian)

Menilai apakah prosedur yang ada saat ini masih dapat ditingkatkan sehingga kualitas pengendalian menjadi semakin baik, dan kemampuannya untuk mendeteksi kesalahan/ kecurangan menjadi semakin baik pula. Hal tersebut dilakukan untuk mengetahui akses pada sistem terhadap pengguna

a) Keamanan atau kontrol terlalu lemah

- Input data tidak diedit secara layak
- Kejahatan yang terjadi terhadap data
- Etika yang dilanggar terhadap data ataupun informasi seperti penyalhgunaan wewenang
- Data tersimpan secara tidak konsisten pada database atau file yang berbeda
- Peraturan atau panduan privasi dapat dilanggar oleh pihak tertentu
- Kesalahan pada saat pengambilan keputusan

b) Keamanan atau kontrol terlalu lemah

- Prosedur dalam birokrasi menghambat sistem
- Pengendalian mengganggu pengguna
- Pengendalian berlebih mengakibatkan penundaan proses tertentu

5) *Efficiency* (efisiensi)

Menilai apakah prosedur yang ada saat ini masih dapat diperbaiki, sehingga tercapai peningkatan efisiensi operasi. Kemampuan mengolah data dengan meminimalisasi langkah kerja yang dianggap tidak perlu.

a) Orang, mesin atau komputer membuang waktu

- Data dimasukan atau disalin secara berlebih
- Data diproses secara berlebih
- Informasi yang dihasilkan berlebihan

b) Orang, mesin atau komputer membuang material atau persediaan

c) Usaha yang dilakukan untuk tugas tertentu terlalu berlebih

d) Bahan untuk tugas tertentu digunakan secara berlebih

6) *Service* (layanan)

Menilai apakah layanan sistem dapat diandalkan, fleksibel, dan ditingkatkan kemampuannya. Kriteria pada yang *service* ini, antara lain: siapakah pengguna

layanan, adakah ada berbagai tipe pengguna, apakah sistem memperhatikan pengguna, petunjuk dan cara penggunaan perangkat harus dimasukkan dalam sistem, serta perlukah menyimpan dokumen.

- a) Sistem menghasilkan produk yang belum akurat
- b) Sistem menghasilkan produk yang belum konsisten
- c) Sistem menghasilkan produk yang belum belum dapat dipercaya
- d) Sistem sulit dipelajari
- e) Sistem sulit digunakan
- f) Sistem canggung untuk digunakan
- g) Sistem belum fleksibel terhadap kondisi baru
- h) Sistem belum fleksibel untuk perubahan tertentu
- i) Sistem belum terintegrasi dengan sistem lain

2.7 Problem Analysis

Problem analysis atau analisis masalah merupakan fase selanjutnya dari definisi lingkup. Fase analisis masalah mempelajari sistem yang ada dan menganalisa temuan – temuan untuk menyediakan tim proyek dengan pemahaman yang lebih mendalam akan masalah masalah yang akan memicu proyek (Whitten, et al., 2007).

Prasyarat untuk fase analisis adalah lingkup dan pernyataan masalah seperti didefinisikan dan di setujui dalam fase definisi lingkup. Produk jadi dari fase analisis masalah adalah satu set tujuan perbaikan sistem yang diperoleh dari pemahaman menyeluruh terhadap masalah-masalah bisnis. Tujuan ini tidak mendefinisikan input, output, atau proses, melainkan mendefinisikan kriteria bisnis dimana semua sistem baru akan dievaluasi. Seperti contoh berikut:

1. Mempercepat waktu antara pemrosesan pesanan dan pengapalan menjadi tiga hari.
2. Mengurangi kerugian kredit bermasalah sampai 45 persen
3. Menyesuaikan dengan persyaratan kualifikasi federal bantuan finansial pada 1 januari

Tujuan perbaikan sistem dapat disajikan kepada para pemilik dan pengguna sistem sebagai rekomendasi tertulis atau presentasi lisan. Tergantung kerumitan masalah dan jadwal proyek. Analisis masalah didokumentasikan berupa model bisnis kondisi sekarang “*as is model business*”. Berikut hal yang dilakukan dalam fase analisis masalah:

1. Memahami permasalahan

Selama fase analisis permasalahan hal pertama yang dilakukan adalah mempelajari sistem yang sudah ada atau sistem terkait. Setiap pemilik sistem, pengguna, dan analis memiliki tingkat pemahaman dan pendapat yang berbeda. Sebuah studi yang akan dilakukan dapat menjelaskan kepada semua pihak terkait permasalahan yang ada.

2. Menganalisis permasalahan dan peluang

Sebagai tambahan mempelajari sistem terkait, tim proyek harus bekerja sama dengan seluruh *stakeholder* sistem untuk menganalisis permasalahan dan peluang.

3. Menganalisis bisnis proses

Pada analisis proses bisnis hanya cocok untuk proyek perancangan proses bisnis atau pengembangan sistem yang berhubungan dengan perancangan bisnis proses. Pada proyek ini tim diminta untuk memeriksa proses bisnis yang lebih detail untuk mengukur nilai dari setiap proses pada sebuah organisasi.

4. Menetapkan tujuan perbaikan sistem

Setelah memahami sistem terkait, masalah dan peluang. Akan ditetapkan tujuan perbaikan sistem. Tujuan pada proses ini untuk menetapkan kriteria terhadap setiap peningkatan sistem yang akan diukur dan diidentifikasi kendala yang mungkin membatasi fleksibilitas dalam mencapai peningkatan. Kriteria sukses dapat diukur dari tujuan, selain mengidentifikasi tujuan diperlukan identifikasi kendala untuk menetapkan batasan pada pencapaian tujuan, waktu, anggaran, dan teknologi yang dibutuhkan.

5. Memperbarui dan menyempurnakan proyek

Berdasarkan jadwal dan *budget* dari fase definisi lingkup. Lingkup tersebut dapat berkembang maupun berubah ukuran dan kompleksitasnya. Pada fase ini evaluasi kembali terkait lingkup proyek dan mempertbaiki atau memperbarui rencana proyek yang sesuai dilakukan.

6. Mengkomunikasikan temuan dan rekomendasi

Fase analisis permasalahan menyimpulkan dengan mengkomunikasikan tugas. Tahap ini akan menjelaskan rekomendasi dari temuan pada sistem yang akan dibangun

2.8 Requirements Analysis

Fase selanjutnya setelah analisis masalah adalah analisis persyaratan/kebutuhan atau *requirements analysis*. Fase ini sangat penting dalam menciptakan sistem informasi baru. Sistem baru akan selalu dievaluasi, terutama seberapa besar persyaratan yang telah dipenuhi oleh sistem tersebut. Oleh karena itu, fase ini dapat menentukan persyaratan dalam sebuah sistem baru. Persyaratan dapat didefinisikan melalui analisis PIECES, jenis data, proses dan antar muka yang harus ada pada sistem. analisis kebutuhan meliputi (Whitten, et al., 2007):

1. Identifikasi dan menyatakan kebutuhan sistem.

Tugas awal dari fase ini adalah mengidentifikasi dan menyatakan kebutuhan sistem. Tugas ini terlihat mudah dan sepele. Namun, fase ini sering terjadi kesalahan, kelalaian dan konflik. Tugas ini dibuat dibuat dalam fase analisis

masalah ketika kita mengidentifikasi sasaran peningkatan sistem untuk diterjemahkan dalam kebutuhan fungsional dan non fungsional dari sistem.

2. Menentukan prioritas kebutuhan sistem

Keberhasilan sebuah sistem dapat diukur dari seberapa besar kebutuhan sistem yang telah dipenuhi. Apabila sebuah proyek melebihi jadwal atau anggaran. Maka identifikasi prioritas pemenuhan kebutuhan terkait sistem sangat diperlukan.

3. Memperbarui atau menyempurnakan rencana proyek

Aktifitas perbaruan dan penyempurnaan rencana proyek dilakukan untuk menyesuaikan pemahaman lingkup proyek dengan identifikasi kebutuhan/persyaratan yang telah dilakukan

4. Mengkomunikasikan kebutuhan sistem

Aktifitas komunikasi kebutuhan dilakukan untuk memberi informasi kepada *stakeholder* terkait sistem terkait dengan kebutuhan beserta preioritas dari kebutuhan tersebut. Fase ini dilakukan terus menerus untuk mengurangi resiko *missed communication*.

2.9 Logical Design

Fase *logical design* atau desain logis merupakan aktifitas lebih lanjut mengenai dokumen kebutuhan bisnis menggunakan model sistem yang menggambarkan struktur data, bisnis proses, alur data, dan antar muka pengguna. Dengan kata lain fase ini memvalidasi kebutuhan yang ditetapkan pada fase analisis kebutuhan. Beberapa hal yang dilakukan dalam fase ini meliputi (Whitten, et al., 2007):

1. Struktur kebutuhan fungsional

Aktifitas pertama dalam fase ini adalah menstruktur kebutuhan fungsional. Aktifitas ini akan menggambarkan atau memperbarui satu atau lebih model sistem untuk menggambarkan persyaratan fungsional. Hal ini mencakup kombinasi data, proses, model obyek yang secara akurat menggambarkan persyaratan bisnis maupun pengguna.

2. Prototipe kebutuhan fungsional

Prototipe dilakukan sebagai alternatif untuk pemodelan sistem. Terkadang pengguna mengalami kesulitan dalam menyatakan fakta-fakta yang diperlukan untuk menggambarkan model sistem yang tepat. Dalam hal ini maka pendekatan alternatif atau pelengkap untuk pemodelan sistem adalah dengan membangun prototipe penemuan.

3. Validasi kebutuhan fungsional

Model sistem dan prototipe merupakan representasi dari persyaratan pengguna. Keduanya harus divalidasi dalam hal kelengkapan dan kebenarannya

4. Menentukan penerimaan uji

Kebanyakan pakar setuju bahwa tidaklah terlalu dini untuk memulai sekalipun merencanakan pengujian sistem. Model dan prototipe sistem sangat efektif untuk menentukan persyaratan, pemrosesan, aturan data maupun aturan bisnis bagi sistem baru.

2.9.2 Unified Modelling Language


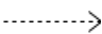

Pada perkembangan teknologi *software* (perangkat lunak), diperlukan adanya standarisasi bahasa atau pemodelan terhadap *software* yang akan dibuat agar orang di berbagai negara dapat mengerti pemodelan perangkat lunak. Pada perkembangan teknik pemrograman berorientasi obyek, muncul standarisasi bahasa pemodelan untuk pembangunan *software* yang dibangun dengan teknik pemrograman berorientasi obyek yaitu UML (*Unified Modelling Language*).




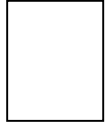



UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks pendukung (Sukamto & Shalahuddin, 2014).

2.9.3 Use case Diagram

Menurut Sukamto & Shalahuddin (2014) *use case* atau diagram use case merupakan pemodelan untuk tingkah laku sistem informasi yang akan dibuat. Diagram use case mendefinisikan interaksi antara satu atau lebih aktor sesuai sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa yang berhak menggunakan fungsi tersebut. Diagram use case terdiri dari beberapa simbol pada table 2.5:

Tabel 2. 5 Simbol Use case Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).



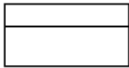




4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

Sumber: Sukamto & Shalahuddin (2014)

2.9.4 Class Diagram

Diagram kelas atau *class diagram* memodelkan struktur sistem dari segi pendefinisian kelas-kelas sesuai dengan sistem yang akan dibuat. Setiap kelas memiliki atribut dan metode operasi (Sukamto & Shalahuddin, 2014). Atribut merupakan variable yang dimiliki suatu kelas. Sedangkan metode atau operasi merupakan fungsi fungsi yang dimiliki suatu kelas. Diagram kelas dibuat agar *programmer* atau pembuat program dapat membuat fungsi dari sebuah sistem sesuai dengan yang ada pada dokumentasi perancangan. Definisi symbol dan kegunaanya dijelaskan pada Tabel 2.6. Susunan struktur diagram kelas yang baik harus memiliki jenis jenis kelas seperti, Kelas Main, kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan, Kelas View, kelas yang menangani tampilan sistem, Kelas Controller, kelas yang menangani fungsi-fungsi yang harus ada dan diambil dari pendefinisian sistem, Kelas Model, kelas yang mawadahi data dari sebuah sistem

Tabel 2. 6 Simbol Class Diagram






NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

Sumber : Sukamto & Shalahuddin (2014)

2.9.5 Activity Diagram

Diagram aktifitas atau *activity diagram* menggambarkan bagaimana aliran kerja atau aktifitas atau proses bisnis atau menu dari sebuah sistem. diagram aktifitas menggambarkan aktifitas sistem, bukan apa yang dilakukan oleh aktor (Sukamto & Shalahuddin, 2014). Diagram aktifitas banyak digunakan untuk hal-hal seperti Perancangan proses bisnis, Urutan atau tampilan dari sistem/*user interface*, Perancangan pengujian dan Perancangan menu yang ditampilkan pada sistem. Definisi simbol apa saja yang dimiliki pada diagram kelas dan kegunaanya dijelaskan pada Tabel 2.7.

Tabel 2. 7 Simbol Activity Diagram

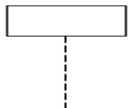
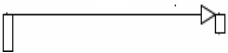
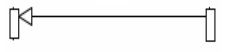
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

Sumber : Sukamto & Shalahuddin (2014)

2.9.6 Sequence Diagram

Diagram sekuen atau *sequence diagram* menggambarkan kelakuan obyek pada *use case* dengan mendefinisikan waktu hidup obyek dan pesan yang dikirimkan dan diterima antar obyek. Untuk menggambarkan diagram sekuen maka harus diketahui obyek-obyek yang terlibat dalam *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi obyek tersebut (Sukamto & Shalahuddin), 2014). Berikut symbol dari *sequence diagrams*.

Tabel 2. 8 Simbol Sequence Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

Sumber : Sukamto & Shalahuddin (2014)

2.9.7 Conceptual Data Model

CDM (*Conceptual Data Model*) atau model konsep data adalah konsep yang berhubungan dengan sudut pandang pengguna dalam menyimpan data pada tabel dalam basis data. CDM digambarkan dalam bentuk tabel dan relasi pada setiap tabel (Sukamto & Shalahuddin, 2014).

2.9.8 Physical Data Model

PDM (*Physical Data Model*) atau model relasional adalah model yang menggunakan tabel untuk menjelaskan data serta hubungan antar data. Setiap tabel memiliki sejumlah atribut dimana setiap atribut pada tabel terdiri dari atribut unik dan tipe data dari setiap atribut. PDM merupakan konsep yang menjelaskan detail dari bagaimana data disimpan dalam basis data. PDM juga merupakan bentuk fisik perancangan basis data yang siap untuk di implementasi ke dalam DBMS (Sukamto & Shalahuddin, 2014).

2.10 Decision Analysis

Tahap terakhir dalam perancangan sistem pada metode FAST adalah *decision analysis* atau analisis keputusan yang bertujuan untuk identifikasi calon solusi, menganalisa kandidat solusi, dan merekomendasikan sistem yang akan dirancang, dibangun, dan diimplementasikan. Dalam fase ini memperhatikan teknologi informasi dan arsitektur sistem seperti apa yang akan dibuat dalam sistem yang akan dibangun. Beberapa hal yang akan dilakukan pada fase ini meliputi (Whitten, et al., 2007):

1. Identifikasi solusi kandidat

Solusi kandidat dapat diketahui pada ide-ide dan opini terhadap perancangan dari pemilik, pengguna, dan tim pengembang sistem. Sehingga pada fase ini hanya untuk menentukan solusi kandidat yang dapat dipertimangkan.

2. Analisa solusi kandidat

Setiap solusi kandidat akan dianalisa untuk kelayakan, fase ini dilakukan setelah proses identifikasi kandidat dilakukan. Kriteria dalam solusi kandidat sedikitnya terdiri dari kelayakan teknis, operasional, ekonomi, dan jadwal.

3. Membandingkan solusi kandidat

Analisis kelayakan yang telah dilengkapi untuk setiap solusi kandidat akan dibandingkan untuk direkomendasikan pada pemilik sistem mana yang memiliki keuntungan lebih dominan

4. Memperbarui rencana proyek

Berdasarkan solusi yang direkomendasikan, evaluasi dan perbaikan dilakukan untuk mengetahui kesesuaian terhadap rencana proyek dan lingkup proyek

5. Merekomendasikan solusi sistem

Merekomendasikan sebuah solusi sistem kepada komunitas bisnis berdasarkan solusi kandidat yang telah diidentifikasi pada awal fase analisis keputusan

2.11 Physical design & integration

Fase perancangan dan integrasi merupakan fase yang mentransformasikan kebutuhan/persyaratan bisnis yang terdapat pada desain logis ke dalam spesifikasi desain fisik yang akan digunakan untuk konstruksi sistem (Whitten, et al., 2007).

Untuk pemodelan desain pada fase ini menghasilkan cetak biru tertulis untuk pedoman membangun sistem. Salah satu pemodelan yang digunakan pada fase ini adalah Unified Modelling Language (UML).

2.12 Construction & testing

Fase ini bertujuan untuk membangun dan menguji sebuah sistem fungsional yang memenuhi persyaratan bisnis dan desain untuk implementasi antarmuka antara sistem baru dan sistem produksi yang telah ada (Whitten, et al., 2007).

2.13 Implementation

Pada fase implementasi proses transisi dari sistem lama ke sistem baru dan membantu pengguna sistem untuk melakukan penyesuaian terhadap sistem yang dilah dibuat. Hasil dari fase implementasi adalah sistem operasional yang akan masuk ke tahap *operational support* dari siklus sistem (Whitten, et al., 2007).

2.14 Consistency analysis: Requirement Configuration Structure

Requirement consistency analysis merupakan metode untuk melakukan analisis konsistensi pada hasil perancangan sistem dengan pemanfaatan hubungan antar elemen perancangan (Nistala & Kumari, 2013). Dalam penerapannya terdapat 4 langkah kerja yaitu:

1) Layers and Configuration Items

Tahap ini mendeskripsikan asal dari 4 layer yang akan dianalisis. Layer tersebut antara lain :

- a) Business layer yang berisi tujuan organisasi yang diperoleh dari proses yang berjalan pada sebuah organisasi.
- b) *Process layer* yang Berisi proses dan sub-proses yang harus ada untuk mencapai tujuan organisasi.
- c) Requirements layer yang berisi kunci dari kebutuhan sistem berdasarkan proses dan sub-proses.
- d) Specification layer yang menghasilkan analisis kebutuhan dalam bentuk spesifikasi kebutuhan.

2) Configuration Structure

Tahap ini memberikan panduan dalam identifikasi layer dan menghubungkan 4 layer pada komponen yang pertama. Setiap elemen pada tiap layer akan dijelaskan pada tahap ini.

3) Consistency Analysis

Tahap ini berguna untuk memberikan validasi dari tahap kedua, dengan cara menggambarkan hubungan antara 4 layer yang telah didefinisikan dengan digambarkan dalam bentuk diagram consistency analysis.

4) Requirement Consistency Index

Requirement Consistency Index berfungsi untuk melakukan perhitungan terhadap persentasi konsistensi dalam pendefinisian kebutuhan. Proses perhitungan RCI dituliskan pada persamaan 2.1.

$$RCI = \frac{A}{B+C} \quad (2.1)$$

Keterangan

A : Jumlah elemen kebutuhan yang konsisten.

B : Jumlah total elemen kebutuhan.

C : Jumlah elemen kebutuhan yang terdefinisi secara tidak benar.

2.15 Correctness

Fokus pengujian perangkat lunak adalah dengan melihat sistem kandidat pada kebutuhan yang dipilih dan memeriksa apakah program kandidat tersebut memiliki fungsi yang sesuai dengan spesifikasi kebutuhannya (Mili & Tchier, 2014). Studi tentang *corectness* program mengarah pada tingkat granularitas yang tidak sesuai. Khususnya mengarah pada asumsi tentang perilaku sistem pada tahap tertentu dalam pelaksanaannya. Berikut merupakan contoh penggunaan *correctness* pada penerapannya.

Diumpamakan R berisi daftar spesifikasi kebutuhan yang diperlukan oleh pengguna.

$$R = \{(0,0), (0,1), (0,2), (1,1), (1,2), (1,3), (2,2), (2,3), (2,4), (3,3), (3,4), (3,5)\}$$

Terdapat 3 program yang memiliki fungsi yang disesuaikan dengan spesifikasi dan diumpamakan kedalam himpunan P.

$$P1 = \{(0,1), (1,2), (2,3), (3,4)\}$$

$$P2 = \{(0,0), (1,2), (2,4), (4,8), (5,10), (6,12)\}$$

$$P2 = \{(0,0), (1,2), (2,4), (3,6), (4,8), (5,10), (6,12)\}$$

Ketiga kandidat program diatas menjelaskan bahwa:

- P1 dikatakan sesuai dengan spesifikasi kebutuhan atau termasuk dalam kategori *correctness* karena setiap fungsi yang dimiliki p1 terdapat pada spesifikasi kebutuhan.
- P2 dikatakan tidak sesuai dengan spesifikasi kebutuhan atau termasuk dalam kategori *partially correctness* karena sebagian fungsi yang dimiliki p2 terdapat pada spesifikasi kebutuhan dan sebagian lagi tidak dimiliki.
- P3 dikatakan tidak sesuai dengan spesifikasi kebutuhan atau termasuk dalam kategori *terminate normally* karena fungsi yang dimiliki p3 tidak sesuai dengan spesifikasi kebutuhan.