**BAB II**

**LANDASAN TEORI**

**2.1. Sistem**

Sistem (system) dapat didefinisikan dengan pendekatan prosedur dan dengan pendekatan komponen. “Dengan pendekatan prosedur, Sistem dapat didefinisikan sebagai kumpulan dari prosedur-prosedur yang mempunyai tujuan tertentu”. Contoh sistem yang didefinisikan dengan pendekatan ini adalah sistem indiehome cinema. [2]

**2.2. Analisis Sistem**

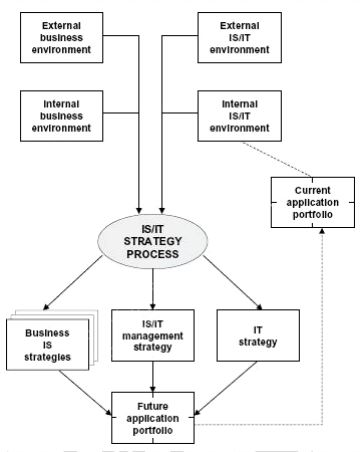
Analisa Sistem adalah teori sistem umum yang sebagai sebuah landasan konseptual yang mempunyai tujuan untuk memperbaiki berbagai fungsi didalam sistem yang sedang berjalan agar menjadi lebih efisien, mengubah sasaran sistem yang sedang berjalan, merancang/mengganti output yang sedang digunakan, untuk mencapai tujuan yang sama dengan seperangkat input yang lain (biasa jadi lebih sederhana dan lebih interatif) atau melakukan beberapa perbaikan serupa. [3]

**2.3. Perencanaan Strategis**

Perencanaan strategis SI/TI dalam sebuah organisasi merupakan proses yang berkelanjutan yang akan perlu sering diperbarui secara teratur sebagai respon terhadap dorongan eksternal, peluang dan kebutuhan bisnis, rencana kerja yang terjadwal, budaya organisasi dan kemanfaatan yang diperoleh dari penerapan strategi itu sendiri. Perbaruan itu bisa berupa revisi yang relatif kecil namun tidak tertutup kemungkinan dibutuhkan perubahan mendasar dan menyeluruh, bergantung pada keluasan cakupan proses strategis organisasi.

Selain merupakan proses yang berkelanjutan, perencanaan strategis juga merupakan proses belajar. Dalam situasi ini, baik dikalangan manajemen yang membidangi SI maupun yang membidangi bisnis organisasi, menjadi semakin waspada pada isu-isu bisnis dan teknologi. Mereka akan terus mencermati dan belajar mengidentifikasi serta menggali peluang secara bersama-sama dalam iklim kerjasama yang harmonis. Budaya kerjasama antara fungsi IS dan organisisasi secara keseluruhan akan mengubah orientasi mereka dalam memperlakukan informasi, yaitu menempatkan sistem dan teknologi menjadi sumber pokok dalam kegiatan bisnis mereka sehari-hari. Lebih dari itu, sistem dan teknologi juga akan merupakan inti bagi upaya pengembangan bisnis secara terus menerus. Ini seiring dengan tingkat kematangan fungsi sistem informasi dalam organisasi tersebut.

Perihal keterkaitan strategi SI/IT dengan strategi bisnis, karakteristik serta pendekatannya telah dikupas pada sub-bab terdahulu, pembahasan selanjutnya akan memusatkan perhatian pada kerangka kerja dan perumusannya. Ward dan Peppard (2002) menyajikan suatu model analisa tentang hal ini yang diwujudkan dalam bangun kotak-kotak yang menggambarkan bagian masukan, keluaran dan aktifitas-aktifitas pokoknya.



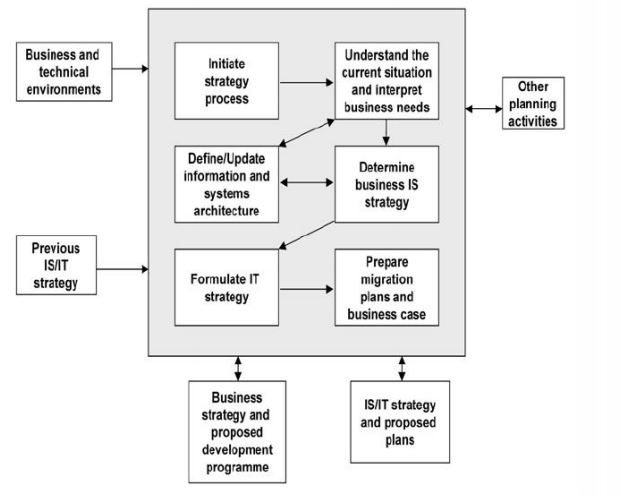
Gambar 2.1 Model Strategi SI/TI (Ward, 2002)

**Masukan** dalam model ini terdiri dari lingkungan internal dan eksternal bisnis, lingkungan internal dan eksternal SI/TI. Lingkungan internal bisnis meliputi strategi bisnis yang ada sekarang, tujuan, sumberdaya, proses serta budaya dan tata nilai organisasi.. Lingkungan eksternal meliputi situasi ekonomi, iklim usaha dan persaingan tempat organisasi menjalankan kegiatannya. Lingkungan internal SI/TI adalah perspektif terhadap bisnis, kematangan SI/TI, cakupan dan kontribusi, kemampuan/ketrampilan, sumberdaya dan infrastruktur teknologinya. Portofolio aplikasi sistem yang ada maupun sisten yang sedang dikembangkan atau yang baru dianggarkan. Lingkungan eksternal SI/TI terdiri dari tren perkembangan teknologi dan peluang yang ada serta sejauhmana organisasi lain juga telah menggunakannya, terutama pihak pelanggan, pesaing dan pemasok.

**Keluaran** dalam model perencaaan strategis SI/TI meliputi strategi manajemen TI yang merupakan elemen umum strategi yang diterapkan secara menyeluruh dalam organisasi, bila perlu harus dipastikan bahwa kebijakan diterapkan secara konsisten. Strategi bisnis SI, menyangkut bagaimana tiap unit atau fungsi dalam SI akan menyiapkan SI/TI guna tujuan-tujuan bisnis mereka. Demikian pula portofolio aplikasi yang hendak di bangun untuk masing-masing unit bisnis dan model bisnisnya, sehingga bisa tergambar arsitektur informasi masing-masing unit. Portotofolio hendaknya menckup juga keterangan tentang bagaimana SI/TI hendak dimanfaatkan dimasa yang akan datang untuk membantuk unit mencapai tujuan-tujuan bisnisnya. Pada bagian keluaran kerangka perumusan strategi SI/TI juga menyertakan kebijakan dan strategi pengelolaan teknologi dan sumberdaya spesialis yang ada.

Model strategis SI/TI menggambarkan keseluruhan komponen yang diperlukan dalam proses perumusan strategi serta hasil yang hendak diperoleh seusai proses perancangan strategi IS/IT. Adapun langkah-langkah dan perumusan strategi SI/TI serta deliverables yang dihasilkan, Ward dan Peppard menuangkannya dalam satu kerangka kerja runtut dan tersendiri. Gambar 2.6 menggambarkan komponen-komponen dalam kerangka kerja tersebut yang terdiri dari:

**Proses Inisiasi Strategi**, komponen ini merupakan bagian pembuka dalam perumusan strategi SI/TI yang meliputi langkah-langkah memastikan tujuan, cakupan dan deliverables, menentukan pendekatan kebutuhan sumberdaya, seperti misalnya perangkat-perangkat otomatis. Ditentukan juga identifikasi personal yang diperlukan dan akan dilibatkan dalam proses ini, dibentuk tim jika perlu bisa juga dilakukan pelatihanpelatihan. Langkah selanjutnya adalah pembuatan mekanisme pengendalian dan kontrol terhadap proses ini, jelaskan bagaimana hubungan kegiatan dengan bisnis dan apa yang bisnis akan dapatkan sebagai masukan. Lakukan identifikasi orang-orang yang akan berpartisipasi sehingga diperoleh gambaran waktu yang diperlukan untuk pengumpulan data dalam tahap analisa. Kemudian dilakukan penyusunan rencana kerja, kebutuhan waktu, peran dan tanggungjawab serta simpul dimana pengujian akan dilakukan.



Gambar 2.2 Kerangka Kerja perumusan dan perencanaan strategi SI/TI

(Ward, 2002)

**Memahami Situasi Saat Ini dan Merumuskan Kebutuhan Bisnis**, langkah ini bertujuan untuk membangun pemahaman yang luas tentang bisnis dalam lingkungannya dan untuk merumuskan kebutuhan bisnisnya, baik saat ini, atau sudah terencana maupun kemungkinannya di masa datang. Hal-hal yang harus dilaksanakan ada tahap ini bisa dikelompokan menjadi tiga kategori. Pertama, menganalisa strategi bisnis, tujuan serta faktor-faktor penentunya (critical success factors) permasalahan dan proses penentu, dalam rangka untuk merumuskan situasi saat ini, kelemahan dan kekuatan yang ada. Tentukan juga informasi yang dibutuhkan dalam kaitannya dengan memusatkan perhatian pada investasi pada sistem guna memenuhi kebutuhan tersebut. Kedua, lakukan evaluasi terhadap kegiatan SI/TI, sistemnya, catu informasinya, sumber, organisasi, ketrampilan dan layanan yang ada, agar bisa ditentukan cakupan serta kontribusinya dan dimana kiranya perbaikan dilakukan akan memberi manfaat. Ketiga, analisa terhadap lingkungan bisnis, baik eksternal maupun internal untuk melakukan identifikasi inovasi-inovasi bisnis yang bisa dilahirkan dengan menerapkan aplikasi SI/TI.

**Menentukan Strategi Bisnis SI**, akumulasi kebutuhan SI bisnis melahirkan suatu rekomendasi untuk menempatkan SI/TI pada aras unit bisnis maupun pada keseluruhan organisasi. Hal ini didokumentasikan dalam strategi manajemen dan SI bisnis. Pada tiap unit bisnis, sistem informasi secara konseptual dikonsolidasikan dan dipetakan dalam portofolio aplikasi yang mencerminkan posisinya saat ini, posisi yang dibutuhkan atau kemungkinan-nya di masa depan.

**Menentukan Arsitektur Sistem Informasi**, dalam langkah ini didayagunakan hasil analisa terhadap proses dan kebutuhan informasi guna membangun suatu model bisnis. Merupakan idealisasi model bisnis di masa depan, dalam kaitan proses, informasi dan sistem, model ini juga dibutuhkan untuk merancang arah saat ada rencana untuk melakukan migrasi. Tahap ini dimulai semenjak porses analisa lingkungan dimulai hingga akhir perumusan strategi bisnis SI.

**Merumuskan Proposal Penyediaan TI**, yang hendak dilaksanakan pada tahap ini adalah menentukan unsur atau bagian dari proposal penyediaan TI. Secara praktis, pada titik ini strategi SI dan proposal penyediaan TI perlu dicocokan kembali dengan rumusan dalam proses strategi bisnis, maksud utamanya adalah agar terjadi konsolidasi. Dengan demikian pihak manajemen bisa menentukan investasi pada program mana yang paling layak dan menguntungkan. Selanjutnya garis besar rencana bisa disusun agar bisa ditentukan rute dan titik-titik pencapaian tiap gagasan utama. Konsekuensinya, harus ada kerjasama yang erat antara SI/TI dengan unsur bisnis dalam rencana pengembangan yang sudah disetujui sehinga bisa ditetapkan garis besar langkah migrasi serta ringkasan “business case” masing-masing program. Rincian proposal masing-masing program masih akan tetap diperlukan saat dana yang dibutuhkan untuk program pengembangan ini diajukan.

Rangkaian langkah-langkah dalam proses strategi SI/TI tersebut diatas akan menghasilkan dua kategori keluaran, yaitu keluaran yang berupa hard deliverables dan soft deliverables. Keluaran yang termasuk sebagai hard deliverable adalah berupa berkas-berkas yang memuat rumusan-rumusan tentang strategi dan perencanaan, seringkali juga termasuk di dalamnya aplikasi yang memuat bahan rujukan (dictionary), matriks serta model-model analisa informasi. Sedangkan keluaran yang dikategorikan sebagai soft deliverable adalah hal-hal yang berkaitan dengan faktor manusia dalam pengembangan perencanaan strategi SI/TI, seperti misalnya ketrampilan, kepedulian serta motivasi. [4]

**2.4. Enterprise Architecture Planning**

Enterprise Architecture Planning se-lanjutnya disebut EAP, merupakan suatu metode yang digunakan untuk memban-gun sebuah arsitektur informasi. Menurut Steven H. Spewak, EAP adalah suatu me-tode pendekatan perencanaan kualitas data yang berorientasi pada kebutuhan bisnis serta bagaimana cara implementasi dari arsitektur tersebut dilakukan sedemikian rupa dalam usaha untuk men-dukung perputaran roda bisnis dan penca-paian isi sistem informasi dan organisasi.

Pada dasarnya EAP bukan merancang bisnis dan arsitekturnya, tetapi

mendefinisikan kebutuhan bisnis dan arsitekturnya. Dalam EAP, arsitektur menjelaskan mengenai data, aplikasi dan teknologi yang dibutuhkan untuk mendukung bisnis organisasi. Untuk hal tersebut tadi, Steven H Spewak menyatakan bahwa pemakaian istilah arsitektur terdiri dari arsitektur data, arsitektur aplikasi, dan arsitektur teknologi. Arsitektur disini dimaksudkan layaknya cetak biru, penggambaran, atau model.

Perbedaan EAP dengan Sistem Informasi Tradisional

EAP memiliki perbedaan dengan perancangan sistem informasi yang tradisional dalam empat hal, yaitu:

1. Arsitektur dapat ditemukan dalam model bisnis fungsional. Pendekatan tradisional untuk perencanaan sistem tidak dimulai dengan definisi secara keseluruhan dari bisnis tersebut, namun dimulai dengan kelompok sistem analis yang bertanya , “Sistem apa yang anda butuhkan?” Pendekatan modern untuk perencanaansistem dimulai dengan sistem analis yang bertanya, “Apa saja yang menjadi critical success factor?” atau dapat juga diinttepretasikan sebagai “informasi apa yang paling anda butuhkan?”

2. EAP mendefinisikan data sebelum aplikasi.

Pendekatan tradisional dimulai dengan menentukan aplikasi apa saja yang diperlukan untuk mendukung bisnis dan apa saja yang dapat diselesaikan oleh aplikasi tersebut. Langkah berikutnya adalah menentukan data apa saja yang harus diproses.

Pada EAP, arsitektur pertama mendefinisikan semua data yang diperlukan untuk mendukung bisnis. Setelah arsitektur tersebut selesai, arsitektur berikutnya mendefinisikan semua aplikasi yang dibutuhkan untuk mengolah data tersebut.

3. EAP menggunakan keterkaitan data untuk membatasi rencana implementasi.

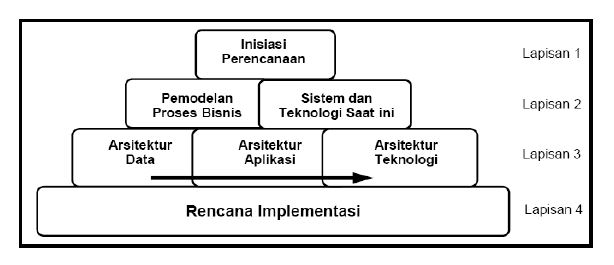
Pada pendekatan tradisional, setelah pertanyaan “Sistem apa yang anda butuhkan untuk mencapai tujuan?,” sistem analis akan bertanya, “sistem mana yang akan diimplementasikan terlebih dahulu?” pihak eksekutif perusahaan memegang peranan untuk menentukan prioritas pengembangan sistem. Pendekatan EAP menunjukan proiritas tersendiri. Keterkaitan data akan menentukan urutan ideal untuk pengembangan aplikasi. Selain untuk untuk pengembangan aplikasi, keterkaitan data ini akan mendorong rencana implementasi.

4. EAP mempertimbangkan baik operasional jangka pendek dan focus strategi jangka panjang. Dalam menggunakan informasi dan teknologi untuk mendukung bisnis.

Pendekatan tradisional berfokus secara eksekutif terhadap masalah jangka pendek. Biasanya pendekatan ini hanya memperhatikan area bisnis yang akan memberikan kontribusi keuntungan dalam waktu dekat. EAP mempertimbangkan tujuan bisnis jangka panjang. Oleh karena itu, EAP akan menunjukan rencana yang fleksibel untuk mencapai tujuan dengan biaya yang masuk akal.

Komponen Enterprise Architecture Planning (EAP).

Komponen dari EAP menurut Spe-wak menggunakan dasar dari dua layer dari John Zachman’s framework yaitu dari tin-jauan planner dan owner. Komponen EAP dapat digambarkan sebagai berikut: [5]

Gambar 2.3 Komponen EAP

**2.4. Perancangan Sistem**

Perancangan sistem adalah spesifikasi umum dan terperinci dari pemecahan masalah berbasis komputer yang telah dipilih selama tahap analisis. [6]

Perancangan sistem adalah bagian dari perancangan bisnis yang berhubungan dengan penyebaran sumber daya sistem informasi sistem informasi perusahaan yang mencakup manusia, perangkat keras dan perangkat lunak. [7]

**2.4.1. UML (Unified Modeling Language)**

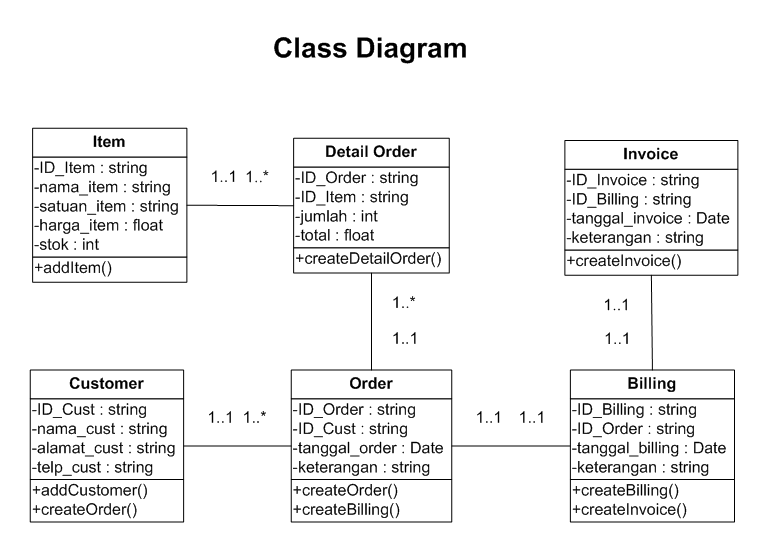
UML adalah bahasa spesifikasi standar untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. Jenis-jenis diagram UML [8] :

* 1. Use Case Diagram

*Gambar 2.1 Usecase Diagram*

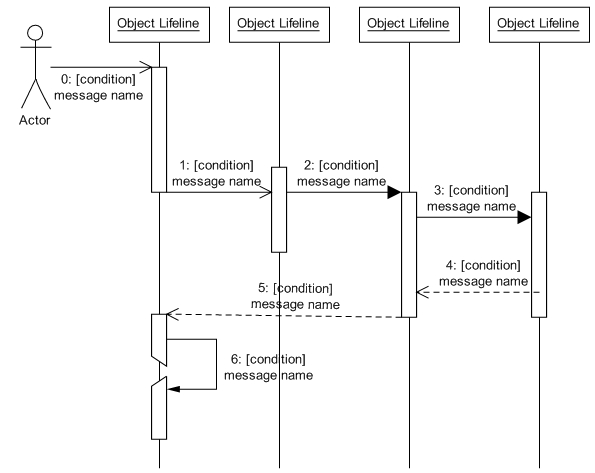
* 1. Class Diagram

Menggambarkan struktur object sistem. Diagram ini menunjukkan class object yang menyusun sistem dan juga hubungan antara class object tersebut.



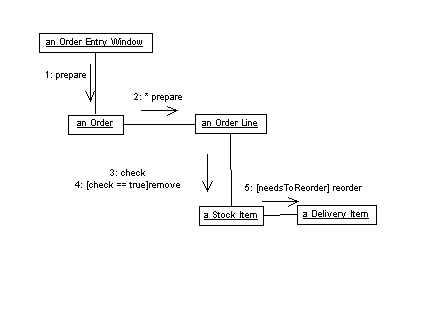
*Gambar 2.2 Class Diagram*

* 1. Sequence Diagram

Secara grafis menggambarkan bagaimana objek berinteraksi dengan satu sama lain melalui pesan pada sekuensi sebuah use case atau operasi.

*Gambar 2.3 Sequence Diagram*

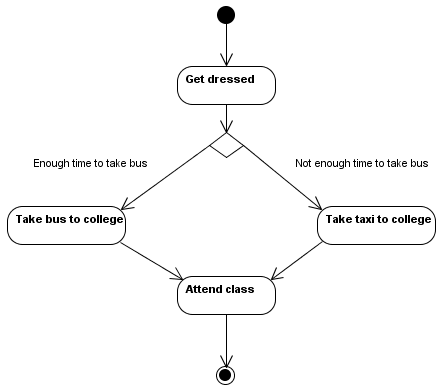
* 1. Collaboration Diagram

Menggambarkan kolaborasi dinamis seperti *sequence diagrams*. Dalam menunjukkan pertukaran pesan, *collaboration diagrams* menggambarkan objek dan hubungannya (mengacu ke konteks). Jika penekannya pada waktu atau urutan gunakan *sequence diagrams*, tapi jika penekanannya pada konteks gunakan *collaboration diagram*.

*Gambar 2.4 Collaboration Diagram*

* 1. Activity Diagram

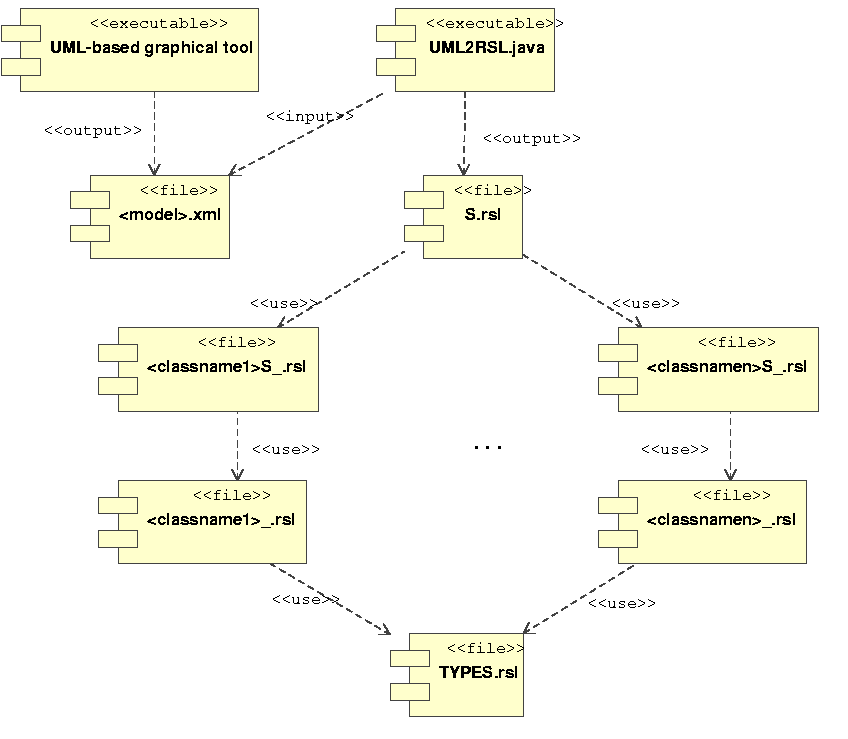
Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi.



*Gambar 2.5 Activity Diagram*

* 1. Component Diagram

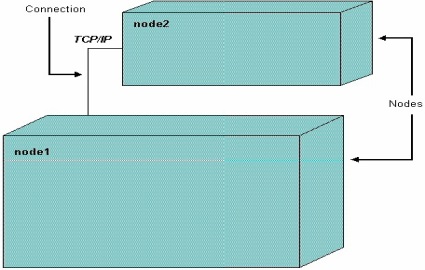
*Component software* merupakan bagian fisik dari sebuah sistem, karena menetap di komputer tidak berada di benak para analis. Komponent merupakan implementasi software dari sebuah atau lebih *class*. Komponen dapat berupa *source code*, komponen biner, atau *executable component*. Sebuah komponen berisi informasi tentang *logic class* atau *class* yang diimplementasikan sehingga membuat pemetaan dari *logical view* ke *component view*. Sehingga *component diagram* merepresentasikan dunia riil yaitu *component software* yang mengandung *component, interface* dan *relationship.*



*Gambar 2.6 Component Diagram*

* 1. Deployment Diagram

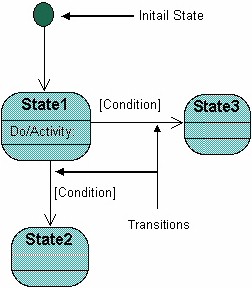
Menggambarkan tata letak sebuah sistem secara fisik, menampakkan bagian-bagian *software* yang berjalan pada bagian-bagian *hardware*, menunjukkan hubungan komputer dengan perangkat (*nodes*) satu sama lain dan jenis hubungannya. Di dalam *nodes, executeable component* dan objek yang dialokasikan untuk memperlihatkan unit perangkat lunak yang dieksekusi oleh *node* tertentu dan ketergantungan komponen.



*Gambar 2.7 Deployment Diagram*

* 1. State Diagram

Menggambarkan semua state (kondisi) yang dimiliki oleh suatu object dari suatu class dan keadaan yang menyebabkan state berubah. Kejadian dapat berupa object lain yang mengirim pesan. State class tidak digambarkan untuk semua class, hanya yang mempunyai sejumlah state yang terdefinisi dengan baik dan kondisi class berubah oleh state yang berbeda.



*Gambar 2.8 State Diagram*

**2.4. Metodologi TOGAF**

TOGAF dimulai awal 1990-an sebagai metodologi untuk pengembangan arsitektur teknis, dan telah dikembangkan oleh The Open Group ke dalam kerangka arsitektur enterprise yang luas.  Pada tahun 1995 , versi pertama dari TOGAF (TOGAF 1.0) disajikan. Versi ini terutama didasarkan pada Architecture Framework Teknis Pengelolaan Informasi (TAFIM), dikembangkan sejak tahun 1980 oleh an Departemen Pertahanan AS.

Pada bulan Desember 2001 TOGAF 7, “Edisi Teknis “, diterbitkan  TOGAF 8 (“Enterprise Edition”) pertama kali diterbitkan pada bulan Desember 2002 dan diterbitkan dalam bentuk diperbarui TOGAF 8.1 pada bulan Desember 2003. Sekitar tahun 2005 menjadi TOGAFTM merek dagang terdaftar dari The Open Group.  Pada bulan November 2006 Open Group dirilis TOGAF 8.1.1. Menurut The Open Group , pada Februari 2011, lebih dari 15.000 individu TOGAF Bersertifikat.  Pada September 2012 register resmi memiliki lebih dari 20.000 individu bersertifikat.

Versi terakhir adalah TOGAF 9.1, diluncurkan pada tanggal 1 Desember 2011. Sebuah perkembangan evolusi dari TOGAF 8, TOGAF 9  mencakup banyak fitur baru termasuk :

* Peningkatan kekakuan, termasuk Konten Metamodel resmi yang menghubungkan artefak TOGAF bersama-sama ( walaupun ada beberapa masalah dengan Metamodel tersebut ) .
* Penghapusan perbedaan yang tidak perlu.
* Banyak lagi contoh dan template.

Panduan dan teknik tambahan meliputi:

* Sebuah pendekatan bisnis berbasis formal arsitektur.
* Kemampuan bisnis berbasis perencanaan.
* Bimbingan tentang cara menggunakan TOGAF untuk mengembangkan Arsitektur dan Keamanan SOAs.

The Open Group menyediakan TOGAF gratis kepada organisasi untuk tujuan non-komersial internal mereka sendiri. Jadi The open group architecture framework (TOGAF) adalah suatu framework ntuk arsitektur perusahaan yang memberikan pendekatan yang komprehensif untuk merancang, perencanaan, pelaksanaan, dan tata kelola arsitektu informasi perusahaan. TOGAF merupakan level atas dan pendekatan holistik untuk desain, yang biasanya dimodelkan pada empat tingkat, yaitu bisnis, aplikasi, data, dan teknologi.

TOGAF memiliki pandangan sendiri, yang dapat ditentukan baik sebagai deskripsi formal dari suatu sistem, atau rencana rinci dari sistem pada tingkat komponen untuk memandu pelaksanaan, atau sebagai struktur komponen, hubungannya, prinsip-prinsip dan pedoman yang mengatur desain dan evolusi.

**Ruang Lingkup**

Awalnya TOGAF digunakan oleh Departemen Pertahanan Amerika Serikat namun pada perkembangannya TOGAF banyak digunakan pada berbagai bidang seperti perbankan, industri manufaktur, Departemen Negara dan juga pendidikan.

**Kelebihan dan Kekurangan**

**Kelebihan Togaf**

* Sifatnya yang fleksibel dan bersifat open source.
* Sistematis
* Focus pada siklus implementasi (ADM) dan proses
* Kaya akan area teknis arsitektur
* Recource base menyediakan banyak material referensi
* Karena melibatkan banyak pihak terutama industri, di TOGAF banyak memberikan best practice atau kejadian riil di dunia nyata

**Kekurangan Togaf**

* Tidak ada templates standart untuk seluruh domain (misalnya untuk membuat blok diagram)
* Tidak ada artefak yang dapat digunakan ulang (ready made)

TOGAF (The Open Group Architecture Framework) muncul dengan cepat dan merupakan kerangka kerja serta metode yang dapat diterima secara luas dalam pengembangan arsitektur perusahaan. Berawal dari Technical Architecture for Information Management atau (TAFIM) di Departemen Pertahanan Amerika Serikat, kerangka kerja itu diadopsi oleh Open Group pada pertengahan 1990an. Spesifikasi pertama TOGAF diperkenalkan pada tahun 1995, dan TOGAF 8 (Enterprise Edition) dirilis pada awal 2004. Pada saat ini sudah ada TOGAF 9 yang secara keseluruhan melengkapi versi sebelumnya.

TOGAF memberikan metode yang detil tentang bagaimana membangun dan mengelola serta mengimplementasikan arsitektur enterprise dan sistem informasi yang disebut dengan ADM (Architecture Development Method).

Tujuan dari arsitektur enterprise adalah untuk mengoptimalkan seluruh perusahaan ke lingkungan terpadu yang tanggap terhadap perubahan dan mendukung strategi bisnis. Arsitektur enterprise yang baik memungkinkan kita untuk mencapai keseimbangan yang tepat antara efisiensi teknologi informasi dan inovasi bisnis. Hal ini memungkinkan unit bisnis individu untuk berinovasi secara aman untuk mengejar keunggulan kompetitif mereka. Keuntungan yang dihasilkan dari arsitektur enterprise yang baik membawa manfaat bisnis yang penting, yang jelas terlihat dalam laporan laba atau rugi bersih dari perusahaan atau organisasi.

**Karakteristik Togaf**

Sebagai kerangka kerja perancangan arsitektur, TOGAF memiliki beberapa karakteristik, antara lain:

* Termasuk dalam 3 kerangka kerja perancangan arsitektur yang paling sering digunakan (Schekkerman, 2003).
* Merupakan kerangka kerja yang bersifat open-standard.
* Bersifat netral –> *fits all*
* Diterima oleh masyarakat internasional secara luas –> *fits all*
* Pendekatannya bersifat menyeluruh (holistic).
* Dibutuhkan metode yang *fleksibel* untuk mengintegrasikan unit-unit informasi dan juga sistem informasi dengan platform dan standar yang berbeda-beda.
* TOGAF mampu untuk melakukan integrasi untuk berbagai sistem yang berbeda-beda
* TOGAF adalah kerangka kerja umum dan dimaksudkan untuk digunakan dalam berbagai macam lingkungan, ia menyediakan konten kerangka kerja yang *fleksibel* dan extensible yang mendasari seperangkat pengiriman arsitektur generik.
* TOGAF cenderung bersifat *generik* dan *fleksibel*karena dapat mengantisipasi segala macam artefak yang mungkin muncul dalam proses perancangan (Resource base TOGAF menyediakan banyak material referensi), standarnya diterima secara luas, dan mampu mengatasi perubahan.
* Fokus pada siklus implementasi (ADM) dan proses –> *process driven*
* Kunci TOGAF adalah metode – TOGAF Architecture Development Method (ADM – Metode Pengembangan Arsitektur) – untuk mengembangkan suatu arsitektur enterprise yang membahas kebutuhan bisnis.
* TOGAF relatif mudah diimplementasikan –> *fits all*
* TOGAF bersifat *open source*, sehingga bersifat netral terhadap teknologi dari *vendor* tertentu –>*fits all*

**Struktur dan Komponen dari TOGAF**

Dalam bidang pendidikan TOGAF telah diimplementasikan oleh *Monash University*. Berikut ini adalah struktur dan komponen dari TOGAF :

**1. Architecture Development Method**

Architecture Development Method menjelaskan bagaimana menemukan sebuah arsitektur perusahaan/organisasi secara khusus berdasarkan kebutuhan bisnisnya. Ini merupakan bagian utama dari TOGAF.

**2. Foundation Architecture (Enterprise Continuum)**

Foundation Architecture merupakan sebuah “framework-within-a-framework” yang menyediakan hubungan bagi pengumpulan asset arsitektur yang relevan dan menyediakan bantuan petunjuk pada saat terjadinya perpindahan abstraksi level yang berbeda.

Foundation Architecture terdiri dari:

a. Technical Reference Model

Menyediakan sebuah model dan klasifikasi dari platform layanan generik.

b. Standard Information Base

Menyediakan standar-standar dasar dari informasi.

c. Building Block Information Base

Menyediakan blok-blok dasar informasi di masa yang akan datang.

**3. Resource Base**

Bagian ini memberikan sumber-sumber informasi  idelines, templates, checklists,latar belakang informasi dan detil  material pendukung yang membantu arsitek di dalam penggunaan Architecture Development Method.

TOGAF digunakan sebagai framework untuk arsitektur sistem informasi perguruan tinggi karena cocok dengan karakteristik perguruan tinggi dan sistem informasinya itu sendiri, yaitu:

1. Dibutuhkan suatu metoda yang fleksibel untuk mengintegrasikan unit-unit informasi dan mungkin juga sistem perencanaan sistem informasi (SI) dengan flatform dan standar yang berbeda-beda. TOGAF mampu untuk melakukan integrasi untuk berbagai sistem yang berbeda-beda.
2. TOGAF cenderung merupakan suatu metoda yang bersifat generik serta fleksibel yang dapat mengantisipasi segala macam artefak yang mungkin muncul dalam proses perancangan (karena TOGAF memiliki resource base yang sangat banyak), standarnya diterima secara luas, dan mampu mengatasi perubahan.
3. TOGAF mudah diimplementasikan.
4. TOGAF bersifat open source . [9]

**2.4.1 *Architecture Development Method* (ADM)**

*Architecture Development Method* (ADM) merupakan inti dari TOGAF sebagai hasil kontribusi dari banyak praktisi arsitektur teknologi informasi di dunia. Secara spesifik ADM dirancang untuk memenuhi kebutuhan bisnis dan teknologi informasi berskala *enterprise*. ADM dilengkapi dengan banyak alat bantu (*tools*) baik dalam perencanaan maupun prosesnya, antara lain:

i. Satu set arsitektur *view* yang mencakup *view* bisnis, data, aplikasi dan teknologi.

ii. Satu set *deliverables* yang direkomendasikan.

iii. *Linkages* dengan banyak studi kasus yang nyata.

iv. Metode untuk mengelola requirement.

**2.5. C# Programming**

C# (dibaca “C Sharp”) merupakan sebuah bahasa pemrograman yang digunakan untuk membangun berbagai macam aplikasi seperti aplikasi *desktop*, *website*, *games* dan *phone app* yang dapat berjalan dalam .NET *Framework*. C# *programming* menyediakan fitur *objectoriented* dan *functional programming*. C# juga menyediakan *service* seperti *runtime* *type checking*, *exception handling*, dan *thread management*.[10]

## 2.6. Visual Basic .NET 2010

Visual Basic .Net 2010 adalah salah satu bahasa pemrograman yang tergabung dalam Microsoft Visual Studio 2010. Visual Studio 2010 dan Microsoft .Net Framework 4.0 membantu developer menghasilkan performansi yang lebih baik dan menghasilkan aplikasi yang scapable. [11]

**2.7. Pengertian Database**

Database adalah kumpulan file-file yang mempunyai kaitan antara satu file dengan file lain sehingga membentuk satu bangunan data untuk menginformasikan suatu perusahaan instansi dalam batasan tertentu. [12]

Database adalah sekumpulan data yang saling terhubung secara logikal yang dirancang dalam rangka memenuhi kebutuhan informasi dalam sebuah organisasi. [13]