# BAB II

# TINJAUAN PUSTAKA

## 2.1. Penelitian Terkait

Surendro (2007), memaparkan pada organisasi yang telah berjalan lama, banyak ditemui adanya aplikasi sistem informasi dengan berbagai platform teknologi dan perangkat teknologi informasi pendukungnya. Sistem ini dikenal dengan istilah *"sistem legacy"*. Biasanya, sistem ini saling terpisah satu dengan yang lain, yang diiringi dengan banyak dan menyebarnya "pulau data" dalam organisasi. Keterpisahan ini memberikan dampak yaitu rendahnya tingkat ketersediaan, konsistensi dan efektivitas penyediaan data.

Kondisi tersebut membuat Sistem Informasi (SI) tidak dapat dimanfaatkan sesuai dengan misinya yaitu menyediakan dan mengolah informasi secara efektif bagi unit organisasi yang membutuhkannya, karena pengembangan sistem informasi tidak direncanakan secara baik. Berbagai macam pendekatan bisa digunakan untuk membuat rencana strategis sistem informasi, salah satunya adalah metodologi *Enterprise Architecture Planning* (EAP).

Perencanaan dengan metodologi EAP menghasilkan arsitektur data, arsitektur aplikasi, arsitektur teknologi, dan arah rencana implementasinya bagi *enterprise*. Kemudian dikombinasikan dengan analisis portofolio aplikasi untuk menentukan implementasi aplikasi dalam hubungannya dengan fungsi bisnis.

Mentang (2011), dalam penelitiannya menjelaskan persaingan yang kompetitif antara perusahaan jasa telekomunikasi, menuntut indosat *sales area*

kendari untuk memperbaiki dan mengembangkan sistem informasi pelayanan kepada pelanggan dalam meningkatkan kualitas layanan sesuai dengan tujuan perusahaan. Diperlukan model yang dapat menggambarkan hubungan antara proses bisnis, data, aplikasi, dan teknologi atas perubahan-perubahan yang terjadi.

Metodologi yang dipakai dalam memodelkan arsitektur *enterprise* yaitu EAP, dengan proses mendefinisikan arsitektur *enterprise* yang memfokuskan pada arsitektur data, aplikasi dan teknologi dalam mendukung bisnis serta rencana untuk mengimplementasikan arsitektur tersebut. Pemodelan arsitektur *enterprise* dengan fokus pada area bisnis sistem informasi pelayanan kepada pelanggan. Tujuan dari penelitian ini adalah membuat *blueprint* arsitektur *enterprise* (data, aplikasi, dan teknologi) sebagai bahan usulan kepada indosat sales area kendari dalam meningkatkan kualitas layanan kepada pelanggan sesuai dengan visi, misi, dan strategi perusahaan.

Fadillah (2011), memaparkan pemanfaatan luasnya jaringan internet sebagai area persaingan dan perluasan pasar bagi para pelaku bisnis, atau lebih dikenal dengan istilah *e-Commerce* atau *e-business*. Dengan mengintegrasikan keempat komponen bisnis proses PT. BaONG dalam *e-Commerce*, yaitu *Point of Sales* (POS), *Web store (End user Interface)*, *Supply Chain Management* (SCM), *Customer Relationship Management* (CRM), dibangunlah arsitektur sistem informasi *e-Commerce* yang terintegrasi dengan metodologi EAP.

Dari penelitian-penelitian di atas dapat disimpulkan bahwa EAP sebagai salah satu metoda atau kerangka acuan untuk membangun sebuah arsitektur sistem informasi yang berorientasi pada kebutuhan bisnis yang terdiri dari

arsitektur data, aplikasi dan teknologi serta rencana implementasi dari arsitektur yang telah dibuat untuk mendukung aktivitas bisnis demi pencapaian misi organisasi. Cakupan EAP dalam perancangan arsitektur sistem informasi, cukup untuk mewakili perusahaan seperti PT. AKAL Interaktif sehingga penulis menggunakan metodologi EAP pada penelitian ini.

Sebagai pembeda dari penelitian-penelitian di atas, model yang akan dikemukakan dalam penelitian ini adalah kombinasi antara metodologi EAP dengan metode *Function Point*. Kombinasi dilakukan secara berurutan dengan melakukan metode EAP terlebih dahulu, selanjutnya dilakukan pengukuran *volume software* menggunakan metode *Function Point* sehingga menghasilkan perkiraan sumber daya pengembangan yang dibutuhkan. Perkiraan ini memberikan dasar untuk perusahaan dalam mempersiapkan kebutuhan sumber daya manusia, waktu dan biaya pada proses implementasi aplikasi.

## 2.2. Sistem Informasi

### 2.2.1. Pengertian Sistem Informasi

Sistem informasi adalah sekumpulan komponen yang saling berhubungan dan fungsinya mengumpulkan, memproses, menyimpan, dan mendistribusikan informasi untuk membantu manajer dalam mengambil keputusan, menganalisis dan menggambarkan masalah yang kompleks dalam suatu organisasi (Laudon & Laudon, 2012).

### 2.2.2. Klasifikasi Sistem Informasi

Klasifikasi sistem informasi (Curry, Flett, & Hollingsworth, 2006) terdiri dari:

1) *Transacation Processing System* (TPS)

   *Transacation Processing System* (TPS) merupakan jenis sistem informasi tertua yang dikembangkan setelah Perang Dunia II di departemen akuntansi dari perusahaan besar. Sistem ini sangat terstruktur dan terperinci yang melibatkan sekumpulan data spesifik dalam format tertentu. TPS ini memeriksa kesalahan transaksi seperti data yang tidak konsisten atau hilang. Contoh jenis sistem ini adalah pengolahan cek dan debet langsung oleh bank, dimana sistem *online* melibatkan suatu hubungan langsung antara operator dan program TPS. Transaksi diproses secara bersamaan sebagai data yang dikumpulkan sebelumnya kemudian diikuti konfirmasi bahwa transaksi telah selesai.

2) *Management Reporting System* (MRS)

   *Management Reporting System* (MRS) merangkum semua laporan yang di produksi secara teratur dan terstruktur untuk *review* indikator kinerja bisnis dan bertindak sebagai umpan balik yang relevan. MRS secara efektif mengambil semua output dari TPS dan internal lainnya berupa sumber-sumber informasi dengan maksud untuk menyediakan informasi untuk manajemen. Contoh jenis sistem ini adalah mengklarifikasi dan menganalisis data penjualan yang diproses dari hari ke hari dengan memfokuskan atau

menyoroti kinerja data dari produk atau agen penjualan sehingga dapat diambil tindakan yang efektif untuk perbaikannya.

3) *Executive Information System and Support* (EISS)

*Executive Information System and Support* (EISS) sangat interaktif dan mengambil konsep dasar dari DSS. EISS dirancang untuk digunakan oleh eksekutif yang sangat senior dan memberikan fleksibel akses informasi untuk pemantauan hasil dan kondisi bisnis yang disajikan secara relevan dengan format *user-friendly*. Tujuan dari EISS tersebut adalah untuk mengatasi keterbatasan dari DSS yang cendrung menyediakan *tools* yang membutuhkan keahlian yang cukup tinggi didalam penggunaannya, sehingga dengan EISS eksekutif dengan mudah dapat mengakses langsung ke informasi tanpa perantara sistem dan teknis analisis. Contoh jenis sistem ini adalah informasi penjualan yang disajikan dalam bentuk antarmuka yang berisi informasi produk, laporan produksi lead time, proses kontrol, memo, ringkasan personel dan kebutuhan-kebutuhan lainnya.

4) *Decision Support System* (DSS)

*Decision Support System* (DSS) menyediakan sistem interaktif untuk membantu dalam proses pengambilan keputusan dalam situasi dimana tidak ada yang yakin apakah keputusan harus dibuat atau melaksanakan tugas yang telah ada. DSS menyediakan *tools* dan model informasi untuk mendukung atau memfasilitasi pembuatan keputusan yang tidak terstruktur atau situasi yang semi-terstruktur. DSS muncul dari meningkatnya kebutuhan sistem informasi yang didukungan oleh situasi yang tidak berulang atau tidak rutin

terjadi dimana kriteria keberhasilan tidak begitu jelas. Contoh jenis sistem ini adalah menganalisis pertimbangan terhadap rentang produk, fitur, harga, penjualan, marjin, jumlah iklan untuk membantu pemasaran dan manajemen memutuskan mengenai rencana masa depan, riset pasar dan memantau hasil dengan cara konsultasi menggunakan berbagai model dan *database*.

5) *Office Information System* (OIS)

*Office Information System* (OIS) biasanya digunakan oleh koperasi yang bertanggung jawab atas pelaksanaan dasar kegiatan usaha organisasi dengan menggunakan TPS sebagai dukungan dalam hal mengumpulkan dan menyimpan data yang berhubungan dengan transaksi bisnis.

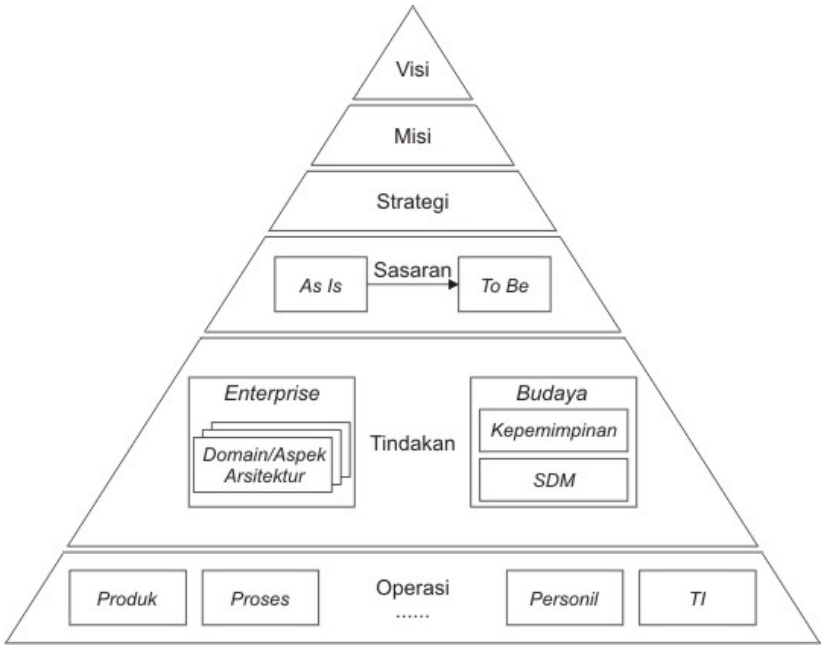
## 2.3. Arsitektur *Enterprise*

Arsitektur *enterprise* adalah kumpulan prinsip, metode dan model yang bersifat masuk akal yang digunakan untuk mendesain dan merealisasikan sebuah struktur organisasi *enterprise*, proses bisnis, sistem informasi dan infrastrukturnya (Surendro, 2009).

Berdasarkan definisi tersebut dapat disimpulkan bahwa arsitektur *enterprise* merupakan cara untuk menggambarkan model operasional *enterprise* yang mencakup aspek perencanaan bisnis, operasional bisnis, otomasi, hingga infrastruktur teknologi informasi pendukungnya. Arsitektur *enterprise* memiliki 4 (empat) komponen *(domain)* utama yaitu arsitektur bisnis, arsitektur informasi, arsitektur teknologi, dan arsitektur aplikasi. Sehubungan dengan keempat

komponen ini, produk arsitektur *enterprise* akan berupa grafik, model, dan narasi yang menjelaskan lingkungan dan rancangan *enterprise*.

Pada gambar 2.1 arsitektur *enterprise* diposisikan dalam konteks pengelolaan *enterprise*. Pada posisi puncak piramid, terdapat visi dari *enterprise*. Visi merupakan pernyataan tentang citra perusahaan dimasa depan. dan nilai-nilai yang dipegang oleh *enterprise*. Misi menggambarkan tentang produk perusahaan, pasar, dan penekanan dibidang teknogi dengan cara mencerminkan nilai dan prioritas para pengambil keputusan yang strategis. Selanjutnya terdapat strategi yang menyatakan tentang rute yang akan dijalani *enterprise* dalam mencapai visi dan misinya. Hal ini diterjemahkan ke dalam sasaran yang jelas yang memberikan arah dan tahapan dalam menjalankan strategi. Penerjemahan sasaran dalam perubahan yang nyata dalam operasi sehari-hari merupakan area aristektur *enterprise* berada.




Gambar 2.1 Arsitektur *enterprise* sebagai instrumen manajemen (Surendro, 2009).

### 2.3.1. Pengertian *Enterprise Architecture Planning* (EAP)

Menurut Spewak, EAP merupakan "proses mendefinisikan arsitektur untuk menggunakan informasi guna mendukung bisnis dan rencana untuk mengimplementasikan arsitektur tersebut" (Surendro, 2009).

EAP merupakan proses untuk mendefinisikan kedua *top layer* dari *framework* arsitektur sistem informasi *Zachman* EAP menghasilkan *blueprint* mengenai data, aplikasi dan teknologi yang menghasilkan solusi jangka panjang yang *cost-effective*, bukan hanya perbaikan secara cepat. Kedua *top layer* dari *framework* arsitektur sistem informasi *Zachman* dia atas dapat diuraikan dengan tabel 2.1.

**Tabel 2.1 EAP Dalam Kerangka Kerja Zachman (Surendro, 2009).**

| | **Apa (Data)** | **Bagaimana (Fungsi)** | **Lokasi (Jaringan)** |
|---|---|---|---|
| **Obyektif/Lingkup (Perencana)** | Daftar entitas yang penting untuk bisnis | Daftar fungsi bisnis yang dilakukan | Daftar lokasi tempat operasi bisnis |
| **Model Enterprise (Pemilik)** | Entitas bisnis dan hubungan-hubungannya | Dekomposisi fungsi dan proses | Hubungan komunikasi antar lokasi bisnis |

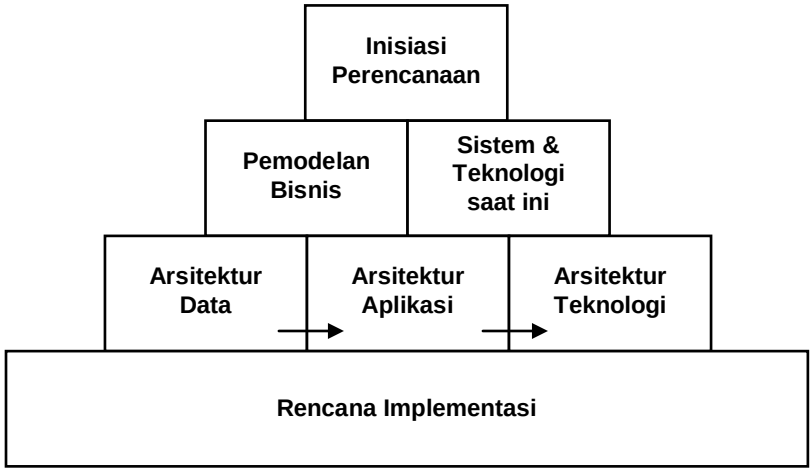
### *2.3.2.* Manfaat EAP

EAP berbeda dengan metoda perencanaan sistem tradisional, dimana pendekatan perencanaan tradisional bersifat *"technology driven"* sedangkan EAP bersifat "*business driven"*. Terdapat sejumlah manfaat dengan menerapkan EAP, diantaranya :

1) Fokus pada penggunaan strategi teknologi untuk mengelola data sebagai suatu aset.
2) Memfasilitasi komunikasi dengan *vocabulary* yang standar serta mengurangi inkonsistensi dan redundansi data.
3) Dokumentasi meningkatkan pemahaman pada bisnis.
4) Model-model yang dihasilkan dapat digunakan untuk menjelaskan bisnis dan menilai dampak perubahan bisnis.
5) Kebijakan-kebijakan pembuatan keputusan dapat di-*review*.
6) Dapat mempertimbangkan integrasi sistem saat ini dengan yang baru.
7) Memungkinkan pendekatan yang komprehensif, obyektif dan tidak parsial.
8) Perencanaan sistem jangka panjang melengkapi rencana bisnis.
9) Solusi *cost-effective*, jangka panjang dengan mempertimbangkan *rate of return.*
10) Melibatkan strategi migrasi yang *feasible* dengan pencapaian jangka pendek.
11) Lebih mudah untuk menilai manfaat dan dampak sistem dan *software* baru.
12) Lebih mudah untuk mengakomodasi perubahan bisnis yang dinamis seperti *merger, acquisition,* produk-produk baru, dan sebagainya.
13) Partisipasi manajemen menyediakan perspektif bisnis, kredibilitas, kepercayaan.

### 2.3.3. Komponen EAP

Komponen atau tahapan EAP, yang menjelaskan bagaimana mendefinisikan arsitektur dan perencanaan ditunjukan pada gambar 2.2.




Gambar 2.2 Komponen dan Lapisan EAP (Surendro, 2009).

Komponen-komponen tersebut terbentuk sebagai lapisan atau tahapan, dimana tiap tahap merepresentasikan fokus tugas yang berbeda, yaitu :

1) Tahap 1 dimana kita memulai inisiasi rencana.

   Memulai EAP pada jalur yang benar, termasuk menentukan metodologi yang digunakan, siapa yang harus dilibatkan dan *toolset* apa yang digunakan. Langkah-langkah yang dilakukan pada fase inisiasi perencanaan adalah :

   (1) Menentukan ruang lingkup dan sasaran perencanaan arsitektur enterprise.

   (2) Menentukan visi.

   (3) Menentukan metodologi.

2) Tahap 2 dimana kita saat ini

   (1) Pemodelan Bisnis.

   Menyusun *knowledge base* mengenai bisnis dan informasi yang digunakan untuk melaksanakan bisnis. Langkah-langkah yang dilakukan pada fase pemodelan bisnis adalah :

a) Analisis Rantai Nilai

Analisis rantai nilai, memberikan kerangka untuk identifikasi dan inventarisasi fungsi bisnis, dengan mengelompokkan area fungsional ke dalam aktivitas utama dan aktivitas pendukung.

b) Daftar Fungsi Bisnis

Untuk melengkapi dan lebih memastikan kelengkapan dekomposisi dalam suatu area fungsi, digunakan analisis siklus hidup sumber daya yang digunakan dalam metodologi *Business System Planning*. Keseluruhan analisis rantai nilai dengan dekomposisi dirangkum melalui model rantai nilai dengan siklus hidup sumber daya dan produk di dalam tiap-tiap area fungsi.

c) Model Bisnis

Setelah proses bisnis didefinisikan, selanjutnya dilakukan identifikasi struktur organisasi yang isinya adalah unit organisasi. Area fungsi beserta proses bisnisnya dipetasilangkan dengan unit organisasi, dengan tujuan untuk mengidentifikasi lingkup tanggung jawab pengambilan keputusan dan keterlibatan tiap unit organisasi dalam tiap area fungsi dan/atau proses bisnis.

(2) Sistem dan Teknologi saat ini.

Tahap ini bertujuan untuk mengidentifikasi dan mendokumentasikan sistem aplikasi dan *platform* teknologi yang digunakan *enterprise* dalam mendukung fungsi bisnis saat ini karena *enterprise* yang telah berjalan pada umumnya telah memiliki sistem dan teknologi untuk aplikasi-aplikasi

sistem informasinya. Hasil dokumentasi disebut sebagai Katalog Sumber Daya Informasi (*Information Resource Catalog* atau IRC) atau disebut juga *Systems Inventory*. IRC tidak menjabarkan setiap sistem secara terperinci, melainkan hanya ringkasannya saja.

3) Tahap 3 visi tentang di mana yang kita inginkan di masa depan

Panah pada *layer* ini menunjukkan bahwa arsitektur data didefinisikan terlebih dahulu, lalu berturut-turut mendefinisikan arsitektur aplikasi dan arsitektur teknologi. Hal ini berbeda dengan metoda perencanaan sistem tradisional yang melakukan sebaliknya, dimana pertama-tama menentukan *hardware*, kemudian aplikasi yang berjalan pada *hardware*, dan terakhir data yang perlu diproses.

(1) Arsitektur Data.

Tahap ini bertujuan untuk mengidentifikasi dan mendefinisikan jenis-jenis data utama atau entitas data yang diperlukan bagi *enterprise* guna mendukung fungsi-fungsi bisnis yang telah didefinisikan pada tahap pemodelan bisnis kemudian merelasikan entitas data tersebut dengan fungsi bisnis enterprise. Arsitektur data merupakan salah satu arsitektur enterprise untuk arsitektur sistem informasi, yaitu kolom data *(what)* pada *Zachman Framework*. Berikut langkah-langkah yang dilakukan pada perancangan arsitektur data:

a) Membuat daftar kandidat entitas data. Tujuannya adalah untuk mengidentifikasikan semua entitas data potensial yang dibutuhkan untuk mendukung bisnis. Hal ini dapat dilakukan dengan

memperhatikan kebutuhan data dari masing-masing *business process* yang telah didefinisikan.

b) Membuat diagram hubungan antar entitas data. Suatu entitas data dapat mendukung lebih dari satu area fungsi bisnis dan tidak berdiri sendiri, tetapi memiliki ketergantungan dan hubungan dengan entitas data lainnya. Pendekatan EAP mengambil ketergantungan dan hubungan antar entitas data ini untuk melandasi pembangunan *enterprise architecture*. Hal ini mempertimbangkan bahwa aplikasi-aplikasi berkaitan erat dengan basis-basis data sedangkan suatu basis data terdiri dari kumpulan entitas data dengan hubungan dan ketergantungannya, oleh karena itu entitas-entitas data perlu dirangkai sesuai dengan ketergantungan dan hubungannya dalam konteks area fungsi yang didukungnya. Pemodelan untuk menggambarkan hubungan antar entitas data menggunakan *Entity-Relationship Diagram* (E-RD). Hasil pemodelan E-RD untuk tiap area fungsi melengkapi kerangka kerja Zachman pada baris perspektif pemilik dan kolom data.

c) Merelasikan entitas data dengan fungsi bisnis. Setiap entitas data yang telah didefinisikan dihubungkan dengan area fungsi bisnis. Hubungan antara entitas data dengan area fungsi bisnis adalah dalam hal pengolahan dan penggunaan data untuk keperluan pemenuhan tujuan fungsi bisnis. Hubungan ini didefinisikan melalui sebuah matriks hubungan antara entitas data dengan fungsi bisnis. Masing-masing sel

dalam matriks untuk menentukan data entitas yang di *create* (C) yaitu fungsi untuk membuat data, *read/reference* (R) yaitu fungsi yang menggunakan data dan *update* (U) yaitu fungsi yang mengubah atau meng-*update* data.

(2) Arsitektur Informasi.

Tahap ini bertujuan untuk mengidentifikasi dan mendefinisikan jenis-jenis aplikasi utama yang dibutuhkan untuk mengelola data dan mendukung fungsi bisnis *enterprise*, kemudian merelasikan aplikasi dengan fungsi bisnis *enterprise*. Arsitektur aplikasi bukan rancangan sistem tetapi merupakan pendefinisian aplikasi apa saja yang dibutuhkan untuk mengelola data dan menyediakan informasi bagi user untuk melakukan fungsi bisnis. Adapun langkah-langkah pada perancangan arsitektur aplikasi adalah:

a) Membuat daftar kandidat aplikasi dan definisi aplikasi. Setelah fungsi-fungsi bisnis didefinisikan dan arsitektur data untuk masa depan dibangun maka dorongan bisnis dan dorongan data diarahkan untuk menentukan dan mendefinisikan aplikasi-aplikasi. Langkah ini bertujuan untuk mengidentifikasikan setiap kemungkinan aplikasi yang dibutuhkan bagi pengelolaan data dan dukungan fungsi bisnis. Langkah awal dalam tahap ini adalah menginventarisasikan kandidat-kandidat aplikasi yang diperlukan untuk mendukung *business process* dan mengelola data untuk masa depan. Kandidat-kandidat aplikasi dapat diperoleh dengan meninjau katalog sumber daya informasi dan

mengakomodasi berbagai masukan kebutuhan aktual dari unit-unit *enterprise* maupun dengan mengadaptasi perkembangan aplikasi sistem informasi.

b) Merelasikan aplikasi dengan fungsi bisnis. Langkah ini bertujuan untuk menentukan fungsi bisnis yang langsung didukung atau diakomodasi oleh aplikasi.

c) Melakukan analisis dampak pada aplikasi yang ada saat ini. Langkah ini merupakan penentuan atas pilihan-pilihan untuk tetap menggunakan aplikasi, memodifikasi, atau mengganti sistem *legacy*.

(3) Arsitektur Teknologi.

Tahap ini bertujuan untuk mengidentifikasi dan mendefinisikan prinsip teknologi yang dibutuhkan untuk menyediakan lingkungan yang mendukung aplikasi pada arsitektur aplikasi yang disusun sebelumnya dalam mengelola data dan mendukung fungsi bisnis. Arsitektur teknologi merupakan definisi dari teknologi yang akan mendukung fungsi bisnis dengan menyediakan lingkungan sharing data. Berikut langkah-langkah pada perancangan arsitektur teknologi:

a) Mengidentifikasikan prinsip teknologi yang akan digunakan. Langkah ini bertujuan untuk mengidentifikasikan prinsip-prinsip yang harus diperhatikan bagi pemilihan *platform* teknologi yang dibutuhkan *enterprise*.

b) Konfigurasi konseptual teknologi. Langkah ini dibangun berdasarkan pada kebutuhan strategi distribusi data dan aplikasi serta kebutuhan

sharing data diantara unit-unit organisasi dengan memperhatikan lokasi bisnis.

4) Tahap 4 bagaimana kita merencanakan untuk mencapainya.

   Tahap ini bertujuan untuk menyusun dan menyiapkan suatu rekomendasi untuk rencana pengimplementasian yang berdasarkan pada arsitektur yang telah dibuat. Adapun langkah-langkah pada tahap rencana implementasi adalah:

   (1) Menentukan urutan prioritas pengembangan aplikasi. Langkah ini diimplementasikan dari sekian banyak aplikasi yang telah didefinisikan dengan menggunakan prinsip aplikasi yang menciptakan *(create)* data terlebih dahulu diimplementasikan sebelum aplikasi yang mengubah *(update)* data atau menggunakan *(reference)* data.

   (2) Membuat estimasi-estimasi pelaksanaan implementasi. Langkah ini bertujuan untuk memperkirakan kebutuhan pada saat implementasi dilaksanakan.

   (3) Membuat kesimpulan perencanaan. Kesimpulan perencanaan merupakan laporan akhir dari perencanaan arsitektur *enterprise* berupa cetak biru.

## 2.4. *Value Chain*

Michael Porter menyatakan bahwa “setiap perusahaan merupakan kumpulan aktivitas yang dilakukan untuk merancang, menghasilkan, memasarkan, menyampaikan dan mendukung produk atau layanannya. Semua aktivitas ini

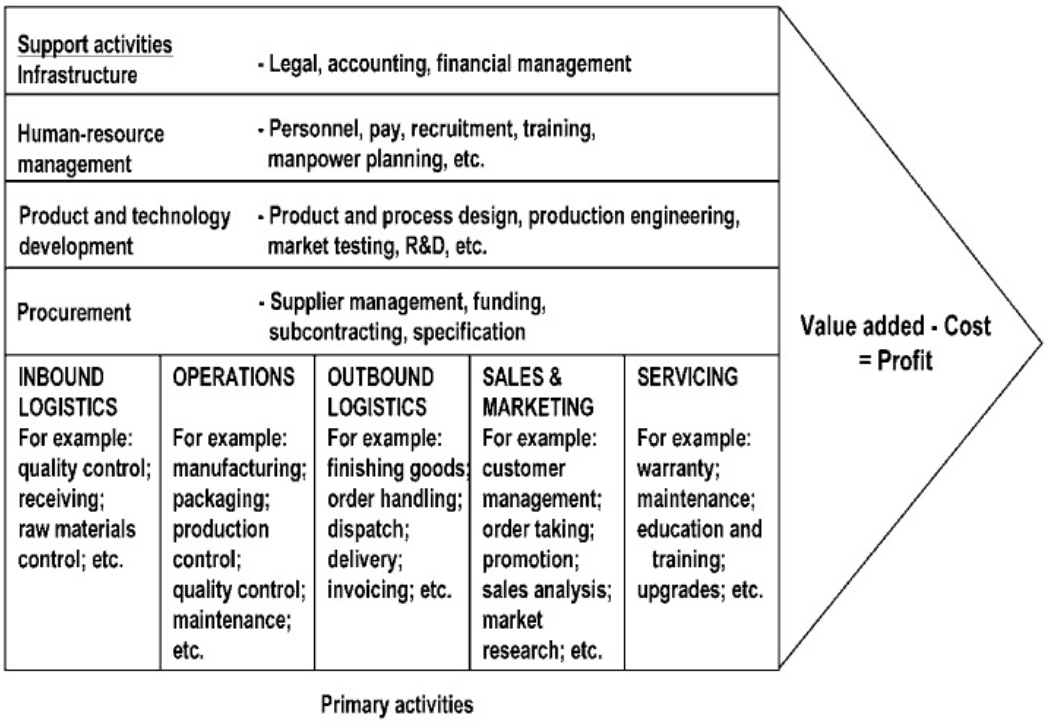
dapat disajikan dalam bentuk *value chain*. *Value chain* hanya dapat dipahami dalam konteks unit bisnis." (Ward & Peppard, 2002).

*Value chain analysis* dapat membantu institusi menentukan tipe kompetitif yang mana yang harus dicapai dan bagaimana mencapainya. Terdapat dua komponen *value chain analysis* : *industry value chain* dan *internal value chain* organisasi. *Industry value chain* terdiri atas aktivitas *value creating* pada industri. Porter mengidentifikasi 5 *competitive forces* dalam industri: intensitas persaingan di antara kompetitor, *barrier* untuk kompetitor baru, ancaman dari produk dan layanan substitusi, daya tawar *supplier*, daya tawar pembeli. Analisis terhadap tekanan-tekanan seperti ini menunjukkan keatraktifan fundamental industri, mengekspose pengendali profitabilitas industri, serta menunjukkan bagaimana profitabilitas dapat meningkat di masa yang akan datang, memberikan perubahan yang berbeda pada *supplier*, *channels, subtitutes, competitors*, atau *technology*.

Kunci analisis *value chain* adalah memahami aktivitas di dalam institusi yang menciptakan manfaat kompetitif serta pengaturan aktivitas tersebut lebih baik dari institusi lain pada industri. Porter mengemukakan bahwa aktivitas bisnis dapat dikelompokan menjadi dua :

1) Aktivitas utama *(primary activities)*, yang secara langsung berkaitan dengan produksi dan pengiriman produk atau layanan.
2) Aktivitas pendukung *(support activities)*, yang mendukung aktivitas utama, tidak terlibat langsung dalam produksi, namun memiliki potensi meningkatkan efisiensi dan efektivitas.

*Value chain analysis* ditujukan untuk melaksanakan proses internal serta mengidentifikasi aktivitas mana yang baik untuk diterapkan, yang baik untuk disediakan. Selengkapnya mengenai rantai nilai model Porter dapat dilihat pada Gambar 2.3.




Gambar 2.3. *Value Chain* (Ward & Peppard, 2002).

*Primary activities* terdiri atas :

1) *Inbound logistic* (input), material yang datang, diproses (bisa ditempat penyimpanan, gudang dan lain-lain) dan dalam pemrosesan ini ditambahkan nilai *(added value)*.
2) *Operations*, material digunakan dalam operasi sehingga memberikan nilai lagi pada produk atau layanan.

3) *Outbound logistic (storage and distribution)*, produk atau layanan perlu dipersiapkan untuk delivery (dapat berupa pengemasan, penyimpanan dan pengiriman).
4) *Marketing and sales*, berusaha menjual produk atau layanan ke konsumen, meningkatkan nilai produk atau layanan dengan menciptakan demand.
5) *Services*, merupakan after sales service yang diberikan pada konsumen untuk kembali memberikan nilai tambah.

*Support activities* terdiri atas :

1) *Organizational infrastructure*, terdiri atas sistem dan fungsi pendukung, contoh *: finance, planning, quality control* dan *general senior management*.
2) *Human resource management*, berhubungan dengan aktivitas rekruitmen, pengembangan, memotivasi, serta memberikan penghargaan pada tenaga kerja.
3) *Technology development*, berhubungan dengan aktivitas pengaturan pemrosesan informasi dan pengembangan serta proteksi "*knowledge"* dalam organisasi.
4) *Procurement*, berhubungan dengan bagaimana sumber daya dibutuhkan dalam organisasi (contohnya *sourcing* dan negosiasi dengan *supplier*).

## 2.5. *Four Stage Life Cycle Business System Planning* (BSP)

*Four stage life cycle* adalah *tool* yang digunakan untuk menemukan turunan dari fungsi bisnis yang terkait dengan produk atau layanan yang diberikan

oleh fungsi bisnis tersebut. *Four stage life cycle* pada BSP digunakan pada tahap pendefinisian proses bisnis. Keempat siklus hidup tersebut adalah:

1) Kebutuhan *(requirement)*

   Aktivitas yang menentukan banyaknya produk atau sumber daya yang diperlukan, rencana mendapatkannya serta pengukuran dan pengendalian terhadap rencana tersebut.

2) Akuisisi *(acquisition)*

   Aktivitas untuk mengembangkan produk atau jasa atau aktivitas untuk mendapatkan sumber daya yang akan digunakan dalam pengembangan.

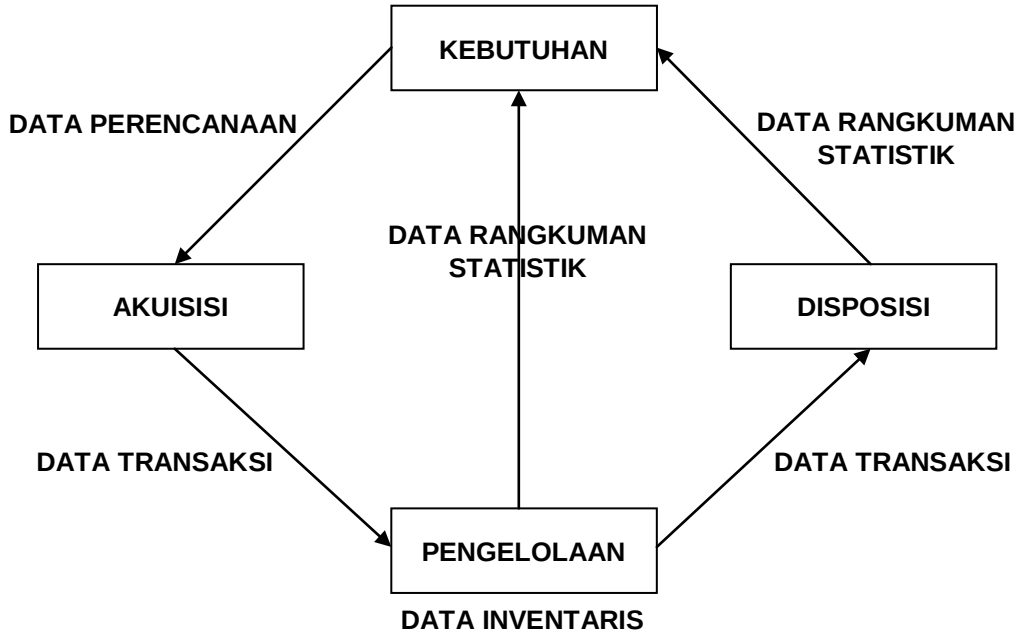
*3)* Pengelolaan *(stewardship)*

   Aktivitas untuk membentuk, memperbaiki, atau memelihara sumber daya pendukung dan untuk menyimpan atau melacak produk atau jasa.

4) Disposisi *(disposition)*

   Aktivitas dan keputusan yang mengakhiri tanggung jawab dari (unit) organisasi terhadap suatu produk, jasa atau suatu penanda untuk mengakhiri penggunaan suatu sumber daya.

Siklus dari *four stage life cycle* dapat dilihat pada Gambar 2.4. Pada gambar tersebut dapat dilihat bahwa pada proses antar siklus maupun sebuah siklus terdapat kelas-kelas data dari aktivitas yang dilakukan pada siklus tersebut. Kelas data dikategorikan ke dalam data perencanaan, data rangkuman statistik, data transaksi dan data inventaris.




Gambar 2.4. Model *four stage life cycle* (Surendro, 2009).

## 2.6. Dekomposisi Proses Bisnis

Daftar fungsi bisnis digunakan untuk menentukan konteks dan lingkup *enterprise* dengan cara mengidentifikasi dan menginventarisasikan area-area fungsi yang dijalankan organisasi. Dalam kerangka kerja Zachman, hasil-hasil ini mengisi baris perencana dan kolom fungsi. Tiap-tiap area fungsi dapat dikomposisikan sehingga menjadi proses-proses bisnis dalam berbagai tingkatannya. Dekomposisi diperlukan untuk menghasilkan model aktual dan nantinya arsitektur *enterprise* yang lebih utuh dengan definisi yang lebih lengkap.

## 2.7. *Business Process Modeling Notation* (BPMN)

BPMN merupakan salah satu metoda pemodelan proses bisnis dari BPMI *(Business Process Management Initiative)* dan model yang digunakan untuk tahapan awal dalam rangkaian seluruh aktivitas pemodelan proses bisnis. BPMN merupakan tools pemodelan proses bisnis yang masih baru, yang dirilis pada bulan Mei 2004. BPMN menyediakan BPD *(Business Process Diagram)*, yang berlandaskan pada teknik *flowchart* yang digunakan untuk membuat model proses bisnis. BPMN mendukung *swimlanes*, yaitu *Pool* dan *Lane*. *Pool* merepresentasikan participant dalam proses, sedangkan *Lane* merupakan dekomposisi atau sub partisi dari *Pool*.

## 2.8. *Information Resource Catalog* (IRC)

IRC digunakan untuk mendokumentasikan dan mendefinisikan semua landasan sistem dan teknologi yang sedang digunakan dalam *enterprise*. IRC tidak menjabarkan setiap sistem secara terperinci, melainkan ringkasannya saja. IRC juga bukan kamus data ataupun inventori peralatan komputasi. Spewak melaporkan keuntungan dari pembuatan dan pengelolaan IRC sebagai berikut :

1) IRC menyediakan rujukan akan semua sumber daya informasi.
2) IRC menunjukan distribusi sumber daya informasi dalam *enterprise*.
3) IRC dapat digunakan sebagai petunjuk lokasi informasi yang dibutuhkan manajemen.
4) IRC dapat digunakan untuk memberikan orientasi bagi personil baru kedalam Departemen SI.

5) IRC digunakan dalam EAP sebagai basis perencanaan.
6) Keputusan penganggaran dan kendali biaya dapat dihubungkan secara langsung dengan IRC.
7) IRC dapat dibuat dengan cepat dan dengan biaya yang layak.
8) IRC merepresentasikan penggunaan internal piranti dokumentasi.

IRC dapat dikerjakan secara terpisah atau bersamaan dengan EAP, akan tetapi lebih baik kalau IRC diselesaikan lebih dahulu sebelum melakukan pekerjaan-pekerjaan arsitektural.

## 2.9. *Entity-Relationship Diagram* (ERD)

Suatu entitas data bisa menunjang lebih dari satu area fungsi dan tidak berdiri sendiri, melainkan memiliki ketergantungan dan hubungan dengan entitas data lainnya. Pendekatan EAP mengambil ketergantungan dan hubungan antar entitas data ini untuk melandasai pembangunan arsitektur *enterprise*. Hal ini mempertimbangkan bahwa aplikasi-aplikasi terkait erat dengan basis-basis data, sedangkan suatu basis data terdiri dari kumpulan entitas data dengan hubungan dan ketergantungannya.

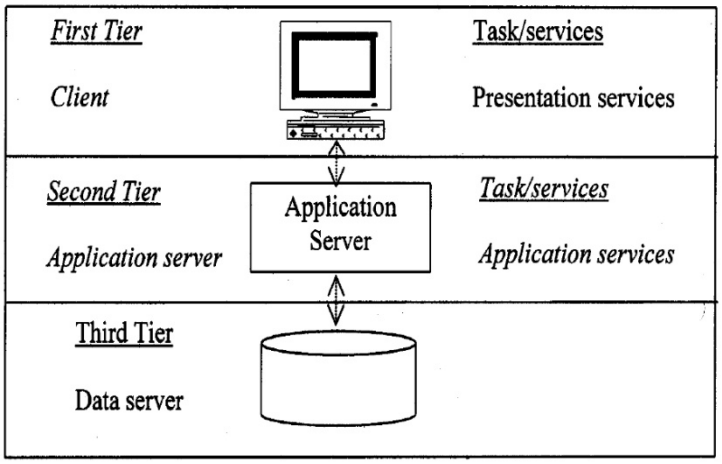
Untuk itu, entitas-entitas data perlu dirangkai sesuai dengan ketergantungan dan hubungannya dalam konteks area fungsi yang didukungnya. Dalam penelitian ini, pemodelan untuk hal ini dilakukan dengan ERD.

## 2.10. Matriks Proses Bisnis-Entitas Data

Hubungan antara area fungsi dan entitas data adalah dalam hal pembuatan, pengolahan, dan penggunaan data untuk keperluan pemenuhan tujuan fungsi bisnis. Hubungan ini didefinisikan melalui matriks proses terhadap entitas data, dimana masing-masing sel dalam matriks diisi dengan huruf : "C" *(create)*, "U" *(update)*, dan "R" *(reference)*. Penanda tersebut bermakna proses yang bersangkutan membuat *(create)*, melakukan *(upadate)*, dan menggunakan *(reference)* entitas data terkait. Proses yang melakukan "C" mengimplikasikan "U" dan "R", sedangkan proses yang melakukan "U" mengimplikasikan "R".

## 2.11. Arsitektur *Three Tier*

Arsitektur *Three Tier* merupakan inovasi dari arsitektur *Client Server*. Pada arsitektur *Three Tier* terdapat *Application Server* yang berdiri di antara *Client* dan *Database Server*. Arsitektur *Three Tier* banyak diimplementasikan dengan menggunakan *Web Application*,sehingga *Client Side* (Komputer *Client*) hanya akan melakukan instalasi *Web Browser*. Saat komputer *client* melakukan inputan data, maka data tersebut dikirimkan ke *Application Server* dan diolah berdasarkan *business process*-nya. Selanjutnya *Application Server* akan melakukan komunikasi dengan *database server*.

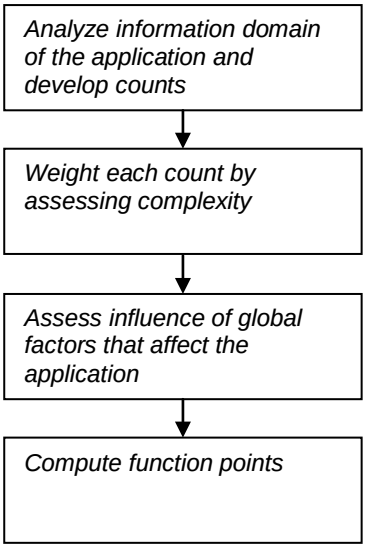



Gambar 2.5. Arsitektur *Three-tier Client/Server* (Gunadi, Julistiono, & Tantama, 2001)

## 2.12. Metode *Function Point*

Gagasan metode *Function Point* muncul dari keinginan untuk mendapatkan suatu teknik pengukuran *volume software* yang tidak hanya berdasar pada banyaknya baris kode program, namun lebih ke arah sesuatu yang dapat diukur lebih awal pada *software development life cycle*. Metode *Function Point* mempunyai keunggulan karena mampu menyediakan perkiraan *volume* proyek dalam bentuk sumber daya pengembangan yang dibutuhkan, sehingga memberikan dasar penting untuk perusahaan *software* dalam mempersiapkan proposal *tender* dan *project plan*. Metode Function Point terdiri dari banyak varian. Variasinya ada pada langkah/tahapan maupun pada isi dari tiap tahapan. Varian-varian ini timbul karena metode ini dapat diubah sesuai dengan kebijakan perusahaan pengembang software.

Tahapan-tahapan dalam menentukan *function point* dapat dilihat pada gambar 2.6. Tahapan tersebut berdasarkan publikasi varian yang populer seperti Gramus dan Herron (1996), IEEE (2000), Caldiera dkk (1998) yang menghasilkan manual penggunaan function point seperi IFPUG 3, IFPUG 4 dan Mark II. (Iskandar & Kusrini, 2008).




Gambar 2.6. Tahapan perhitungan *function point* (Iskandar & Kusrini, 2008)

Sebelum perhitungan dilakukan, yang pertama dilakukan adalah menganalisis aplikasi yang hendak dibuat. Setelah itu langkah selanjutnya, melakukan perhitungan. Secara singkat diterangkan sebagai berikut:

1) Menghitung *crude function points* (CFP). Jumlah dari komponen fungsional sistem pertama kali diidentifikasi dan dilanjutkan dengan mengevaluasi kuantitasi bobot kerumitan dari tiap komponen tersebut. Pembobotan tersebut kemudian dijumlahkan dan menjadi angka CFP, format penghitungan CFP seperti pada tabel 2.2.

**Tabel 2. 2. Format Penghitungan CFP** (Iskandar & Kusrini, 2008)

| TIPE KOMPONEN | LEVEL KOMPLEKSITAS | | | | | | | | | TOTAL CFP |
|---|---|---|---|---|---|---|---|---|---|---|
| | SEDERHANA | | | MENENGAH | | | KOMPLEKS | | | |
| | JUMLAH | BOBOT | POINT | JUMLAH | BOBOT | POINT | JUMLAH | BOBOT | POINT | |
| | A | B | C=AxB | D | E | F=DxE | G | H | I=GxH | J=C+F+I |
| Tipe Input | | 3 | | | 4 | | | 6 | | |
| Tipe Output | | 4 | | | 5 | | | 7 | | |
| Tipe Query | | 3 | | | 4 | | | 6 | | |
| Tipe File | | 7 | | | 10 | | | 15 | | |
| Tipe Interface Ext | | 5 | | | 7 | | | 10 | | |
| **TOTAL CFP** | | | | | | | | | | |

2) Menghitung *Relative Complexity Adjustment Factor* (RCAF). RCAF berfungsi untuk menghitung kesimpulan kompleksitas dari subyek perspektif yang paling berpengaruh terhadap usaha pengembangan. Penilaian dilakukan dengan skala pengukuran 0 sampai 5 yang diberikan pada tiap subyek penilaian. Format penilaian RCAF seperti pada Tabel 2.3.

**Tabel 2.3. Format Penilaian RCAF** (Iskandar & Kusrini, 2008)

| NO | SUBYEK | NILAI |
|---|---|---|
| 1. | Tingkat kompleksitas kehandalan backup/recovery | 0 1 2 3 4 5 |
| 2. | Tingkat kompleksitas komunikasi data | 0 1 2 3 4 5 |
| 3. | Tingkat kompleksitas pemrosesan terdistribusi | 0 1 2 3 4 5 |
| 4. | Tingkat kompleksitas kebutuhan akan kinerja | 0 1 2 3 4 5 |
| 5. | Tingkat kebutuhan lingkungan operasional | 0 1 2 3 4 5 |
| 6. | Tingkat kebutuhan knowledge pengembang | 0 1 2 3 4 5 |
| 7. | Tingkat kompleksitas updating file master | 0 1 2 3 4 5 |
| 8. | Tingkat kompleksitas instalasi | 0 1 2 3 4 5 |
| 9. | Tingkat kompleksitas aplikasi input, output, query dan file | 0 1 2 3 4 5 |
| 10. | Tingkat kompleksitas pemrosesan data | 0 1 2 3 4 5 |
| 11. | Tingkat ketidakmungkinan penggunaan kembali/reusable kode program | 0 1 2 3 4 5 |
| 12. | Tingkat variasi organisasi pelanggan | 0 1 2 3 4 5 |
| 13. | Tingkat kemungkinan perubahan/flekibilitas | 0 1 2 3 4 5 |
| 14. | Tingkat kebutuhan kemudahan penggunaan | 0 1 2 3 4 5 |
| | **TOTAL RCAF** | |

3) Menghitung *Function Point* dengan persamaan:

FP = CFP x (0.65 +0.01 x RCAF)