BAB 2
LANDASAN TEORI

2.1 Sistem Informasi

Sistem informasi adalah kumpulan elemen yang saling berhubungan satu sama lain yang membentuk satu kesatuan untuk mengintegrasikan data, proses dan menyimpan, serta mendistribusikan informasi (Sutedjo, 2002). Sistem informasi dapat didefinisikan sebagai seperangkat komponen yang saling terkait bekerja sama untuk mengumpulkan atau mengambil, memproses, menyimpan, dan menyebarkan informasi untuk mendukung pengambilan keputusan, koordinasi, dan pengendalian dalam suatu organisasi (Laudon, 2010). Selain itu, sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari waktu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan (Jogiyanto, 2005). Sistem informasi juga merupakan kombinasi antar prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi (Kadir, 2005).

2.1.1 Komponen Sistem Informasi

Istilah dalam komponen sistem informasi adalah blok bangunan yang dapat dibagi menjadi enam blok (Jogiyanto, 2005), diantaranya:

1. Blok masukan, data-data yang masuk ke dalam sistem informasi, dapat berupa dokumen-dokumen dasar yang dapat diolah menjadi suatu informasi tertentu.
2. Blok model, terdiri dari kombinasi prosedur, logika dan model matematik yang akan mengolah data input untuk menghasilkan suatu informasi yang dibutuhkan.
3. Blok keluaran, informasi yang menghasilkan sekumpulan data yang nantinya akan disimpan berupa data cetak laporan.
4. Blok teknologi, penunjang utama dalam berlangsungnya sistem informasi. Memiliki beberapa komponen yaitu alat memasukan data, alat untuk menyimpan dan mengakses data (*storage device)*, alat untuk menghasilkan

dan mengirimkan keluaran *(output device)* dan alat untuk membentuk pengendalian sistem secara keseluruhan (*control device*).

5. Blok basis data, kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan oleh perangkat lunak untuk memanipulasinya.
6. Blok kendali, beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah bila terlanjur terjadi kesalahan-kesalahan kerusakan dalam penggunaan sistem.

2.2 *Enterprise Architecture*

Enterprise architecture merupakan salah satu disiplin dalam teknologi informasi, memiliki deinisi sebagai berikut :

1. *Enterprise architecture* merupakan deskripsi misi para stakeholder yang mencakup parameter informasi, fungsionalitas, lokasi, organisasi, dan kinerja. Arsitektur enterprise menjelaskan rencana untuk membangun sistem atau sekumpulan sistem (Osvalds, 2001).
2. *Enterprise architecture* merupakan Pendekatan logis, koprehensif, dan holistik untuk merancang dan mengimplementasikan sistem dan komponen sistem yang bersama (Parizeau, 2002).
3. *Enterprise architecture* merupakan Basis aset informasi strategis, yang menentukan misi, informasi dan teknologi yang dibutuhkan untuk melaksanakan misi, dan proses transisi untuk mengimplementasikan teknologi baru sebagai tanggapan terhadap perubahan kebututhan mis (Chief Information Officer Council, 2001).

Berdasarkan definisi diatas 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 empat komponen/domain utama yaitu : arsitektur bisnis, arsitektur informasi, arsitektur teknologi, dan arsitektur aplikasi.

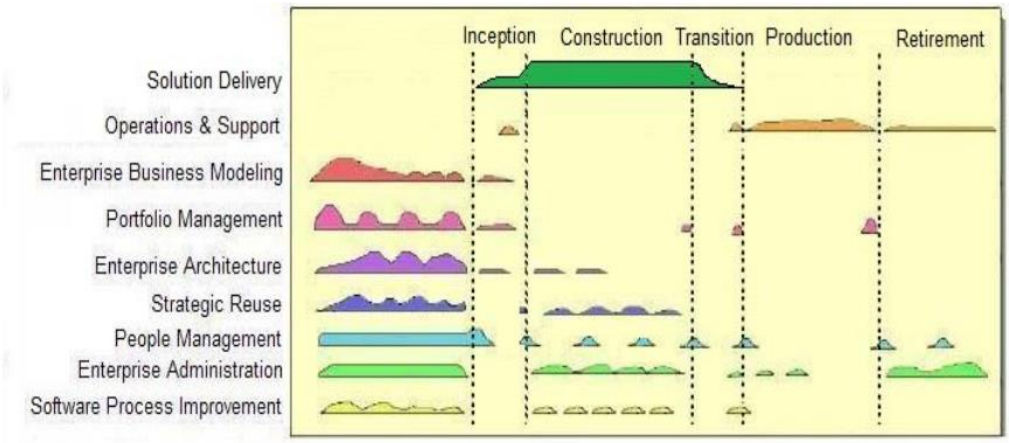
2.2.1 Keuntungan Arsitektur *Enterprise*

Terdapat beberapa keuntungan dari arsitektur *enterprise*, diantaranya (Esmaeil, 2012):

1. Dapat menangkap fakta tentang misi, fungsi, dan landasan bisnis dalam bentuk yang dipahami untuk mendukung perencanaan dan pengambilan keputusan.
2. Dapat memperbaiki komunikasi di antara organisasi Teknologi Informasi dan bisnis dalam perusahaan menggunakan kosakata standar.
3. Fokus pada penggunaan strategi dari teknologi untuk pengelolaan informasi perusahaan yang baik dan meningkatkan konsistensi, akurasi, tepat waktu, integritas, kualitas, ketersediaan dan berbagi informasi pengelolaan Teknologi Informasi.
4. Mencapai skala ekonomi dengan menyediakan mekanisme berbagi layanan di seluruh bagian perusahaan
5. Mempercepat integrasi sistem yang eksis, migrasi, dan yang baru.
6. Memastikan pemenuhan hukum dan regulasi.

2.3 *Enterprise Unified Process*

Proses perangkat lunak yang efektif sangat penting bagi keberhasilan sebuah organisasi. *Unified Process* (UP) adalah suatu kerangka kerja dari proses perangkat lunak yang instantiated, baik *Rational Unified Process* (RUP) maupun *Enterprise Unified Process* (EUP). Ruang lingkup dari RUP adalah dari pengembangan sistem/*software*, sedangkan EUP perluasan dari RUP untuk menjadi siklus penuh teknologi informasi yang meliputi siklus hidup sistem secara lengkap (multi sistem) dengan menambahkan dua fase baru yaitu *production* dan *retirement* serta menambahkan tujuh *enterprise discipliness* (Ambler, 2005).



Gambar 2.1 Fase *Enterprise Unified Process* (EUP)

2.3.1 Fase-Fase EUP

Fase-fase dalam EUP terdiri dari enam fase (Ambler, 2005), yaitu :

1. *Inception Phase*, Tahapan ini merupakan tahapan paling awal dimana aktivitas penilaian terhadap sebuah proyek perangkat lunak dilakukan. Tujuannya adalah untuk mendapatkan kesepakatan dari stakeholder sehubungan dengan tujuan dan dana proyek. Tahap ini juga digunakan untuk mendapatkan gambaran umum kebutuhan, persyaratan dan fungsi-fungsi utama perangkat lunak.
2. *Constraction Phase*, Tujuan dari tahapan ini adalah membangun perangkat lunak sampai dengan saat perangkat lunak tersebut siap digunakan. Titik berat tahapan ini adalah pada penentuan tingkat prioritas kebutuhan / persyaratan, melengkapi spesifikasinya, analisis lebih dalam, disain solusi yang memenuhi kebutuhan dan persyaratan, pengkodean dan pengujian perangkat lunak. Jika dimungkinkan versi awal dari perangkat lunak diuji cobakan untuk mendapatkan masukan dari pengguna.
3. *Transition Phase*, Tahap ini difokuskan pada bagaimana menyampaikan perangkat lunak yang sudah jadi pada pengguna. Perangkat lunak akan secara resmi diuji oleh baik oleh penguji (*tester*) yang kompeten maupun oleh pengguna. Beberapa aktivitas seperti pemindahan pusat data dan pelatihan pengguna dan staf pendukung harus dilakukan pada tahap ini.

4. *Production Phase*, fase ini dilakukan setelah melakukan deployment terhadap sistem untuk memastikan bahwa sistem berjalan dengan baik (*monitoring*), mengelola permintaan perubahan dan perbaikan sistem jika terjadi kerusakan.
5. *Retirement Phase*, fase terakhir ini mengenai penghapusan sistem yang lama yang sudah tidak terpakai dan digantikan dengan sistem baru sehingga tidak terjadi penumpukan pengoperasian terhadap pengguna sistem.

2.3.2 Disiplin EUP

EUP mencakup tujuh disiplin enterprise management yang mendukung cross-system untuk menjadi kunci kesuksesan teknologi informasi (Ambler,2005). Berikut ini tujuh disiplin enterprise management EUP :

1. *Enterprise Business Modeling*, disiplin ini untuk mengeksplorasi struktur dan proses bisnis perusahaan. Ini memberikan pemahaman umum business activities, customer, dan supplier bisnis. *Enterprise business modeling* membantu mengidentifikasi masalah dan sistem yang akan diotomatiskan. Disiplin ini memiliki komponen untuk menjelaskan setiap fasenya, yaitu :
   a) *Enterprise Vision* adalah sebuah pernyataan dari tujuan utama dari sebuah organisasi.
   b) *Enterprise Mission Statement* adalah sebuah pernyataan dari strategi yang harus diikuti untuk mencapai enterprise vision.
   c) *Enterprise Business Process Model* adalah menangkap proses bisnis fundamental, entitas dan alur kerja
2. *Portfolio Management*, disiplin ini untuk membantu meningkatkan efisiensi dan efektivitas Teknologi Informasi dalam suatu organisasi dengan memilih dan kemudian mengelola proyek pengembangan perangkat lunak yang layak. Pada disiplin ini terdapat *Inventory Current System* yaitu kegiatan menginventarisasi sistem saat ini yang ada di perusahaan. Tujuan kegiatan ini adalah untuk mengetahui sistem yang sudah ada dalam perusahaan.
3. *Enterprise Architecture*, disiplin ini membahas masalah arsitektur keseluruhan yang terkait dengan organisasi, membantu memastikan konsistensi sistem dan

aplikasi untuk memudahkan upaya arsitektur. Disiplin ini memeliki beberapa *workflow* diantaranya :

a) *Define Architectural Requirements* adalah mendefinisikan kebutuhan arsitektur perusahaan meliputi arsitektur data, aplikasi dan teknologi.
b) *Define Candidate Architecture* adalah untuk mengidentifikasi dan kemudian memvalidasi kandidat entitas data, kandidat aplikasi dan kandidat teknologi yang merupakan blok pembangunan arsitektur yang akan dibangun perusahaan.

4. *Strategic Reuse*, disiplin ini mempromosikan pengembangan dan penggunaan kembali aset Teknologi Informasi yang sudah ada, tujuannya adalah untuk memungkinkan pengembangkan aplikasi yang lebih efisien dalam hal penggunaan komponen yang dapat digunakan bersama.
5. *People Management*, disiplin ini menjelaskan proses pengorganisasian pelatihan, pemantauan, dan mengarahkan orang-orang untuk keberhasilan proyek-proyek dalam organisasi.
6. *Enterprise Administration*, disiplin ini menentukan bagaimana suatu organisasi menciptakan, memelihara, mengelola data aset fisik dan menyebarkan informasi.
7. *Software Process Improvement*, disiplin ini menjelaskan kebutuhan metode pengembangan software yang diusulkan. Tujuannya adalah untuk sinkronisasi dengan perencanaan arsitektur enterprise.

2.3.3 Keuntungan EUP

Ada beberapa keuntungan pengembangan arsitektur enterprise dengan menggunakan pendekatan EUP yaitu (Ambler, 2005) :

1. Meningkatkan kualitas Teknologi Informasi pada suatu perusahaan dengan cara melakukan *cross-system* terhadap semua kebutuhan dengan manambahkan disiplin *Enterprise Management* berupa *Enterprise Business Modelling, Portfolio Management, Enterprise Architecture, Strategic Reuse, People Management, Enterprise Administration* dan *Software Process Improvement.*

2. Pendekatan EUP mendukung secara eksplisit pengembangan infrastruktur multisistem seperti pengembangan organisasi secara luas.
3. *Risk focused*, EUP memungkinkan para pengembang arsitektur enterprise untuk bisa mengetahui risiko vital di setiap awal tahapan pengerjaan. Dengan demikian faktor-faktor yang sekiranya mempunyai resiko yang paling vital bisa lebih mendapat perhatian terlebih dahulu, sehingga nantinya tidak mengganggu proses pengembangan arsitektur enterprise selanjutnya. Selain itu EUP juga menghendaki agar risiko di setiap fase bisa segera diselesaikan pada fase itu juga, sehingga tidak menghambat fase-fase selanjutnya.
4. Meningkatkan produktivitas *end-user* dengan menambahkan fase baru yaitu production dan retirement. Pada fase tersebut dilakukan monitoring terhadap sistem dan penghapusan sistem yang tidak terpakai sehingga dilakukan *software process improvement*.
5. Perubahan user requirement lebih mudah diterapkan. Hal ini terdapat pada *fase transition*.

2.4 *Unified Modeling Language*

Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem (Dharwiyanti, dkk., 2003).

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan class dan operation dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C. Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan syntax/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML

syntax mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (Object-Oriented Design), Jim Rumbaugh OMT (Object ModelingTechnique), dan Ivar Jacobson OOSE (Object-Oriented Software Engineering).

Abstraksi konsep dasar UML yang terdiri dari structural classification, dynamic behavior, dan model management, bisa kita pahami dengan mudah apabila kita melihat gambar diatas dari Diagrams. Main concepts dapat dipandang sebagai term yang akan muncul pada saat kita membuat diagram. Dan view adalah kategori dari diagaram tersebut.

1. *Use Case Diagram*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

2. *Class Diagram*

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

3. *Statechart Diagram*

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu state ke state lainnya) suatu objek pada sistem sebagai akibat dari stimuli yang diterima. Pada umumnya statechart diagram menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu statechart diagram).

4. *Activity Diagram*

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin

terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

5. *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.

6. *Collaboration Diagram*

Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*. Setiap *message* memiliki *sequence number*, di mana *message* dari level tertinggi memiliki nomor 1. *Messages* dari level yang sama memiliki prefiks yang sama.

7. *Component Diagram*

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) diantaranya. Komponen piranti lunak adalah modul berisi kode, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time, link time*, maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil.

Komponen dapat juga berupa interface, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

8. *Deployment Diagram*

Deployment/physical diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal Sebuah node adalah server, *workstation*, atau piranti keras lain yang digunakan untuk men-deploy komponen dalam lingkungan sebenarnya. Hubungan antar node (misalnya TCP/IP) dan requirement dapat juga didefinisikan dalam diagram ini.

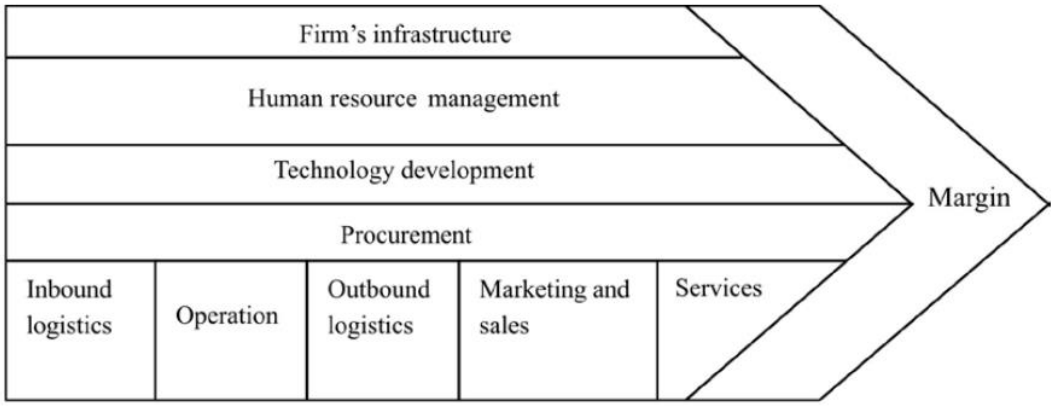
2.5 Analisis Rantai Nilai

Analisis Rantai Nilai merupakan alat kunci utama untuk memahami lingkungan internal oraganisasi (Assauri, 2013). Dengan analisis ranai nilai, suatu perusahaan akan lebih dapat memahami tentang bagian dari operasi usahanya yang menciptakan nilai/ *value,* dan bagian mana yang tidak. Model rantai nilai pertama kali diusulkan oleh Porter, yang terdiri dari satu rangkaian aktivitas yang menciptakan dan membangun suatu nilai yang dapat menghasilkan margin nilai tambah bagi organisasi. Rantai nilai (*value chain*) memberikan kerangka untuk mengidentifikasi dan menginventarisasikan area-area fungsi bisnis, yaitu dengan pengelompokkan area-area fungsional ke dalam (Porter, 1985) :

1. Aktivitas utama (*Primary activities*), yang berupa:
    a. Logistik masukan (*inbound logistics*): aktivitas yang berhubungan dengan penerimaan, penyimpanan dan menyebarkan masukan.
    b. Operasi (*operations*): aktivitas yang mentransformasikan masukan menjadi keluaran menjadi produk akhir.
    c. Logistik keluaran (*outbound logistics*): aktivitas yang berhubungan dengan menyebarkan produk/jasa ke pelanggan.
    d. Pemasaran dan penjualan (*marketing and sales*): aktivitas yang berhubungan dengan pemasaran dan penjualan seperti promosi dan sebagainya.

e. Layanan (*service*): aktivitas yang berhubungan dengan penyedia layanan untuk meningkatkan pemeliharaan produk seperti pelatihan, perbaikan dan perawatan.

2. Aktivitas pendukung (*Support activities*), yang berupa:
   a. Infrastruktur perusahaan (*firm infrastructure*): aktivitas yang terkait dengan biaya serta aset yang berhubungan dengan manajemen umum, *accounting* dan keuangan, keamanan dan keselamatan sistem informasi dan fungsi lainnya.
   b. Manajemen sumber daya manusia (*human resources management*): aktivitas yang terkait dengan penerimaan, pelatihan, pengembangan dan kompensasi untuk semua tipe personil dan mengembangkan tingkat keahlian pekerja.
   c. Pengembangan teknologi (*technology development*): aktivitas yang terkait dengan biaya yang berhubungan dengan produk, perbaikan proses, perancangan peralatan, pengembangan perangkat lunak komputer, sistem telekomunikasi, kapabilitas basis data baru dan pengembangan dukungan sistem berbasis komputer.
   d. Pengadaan (*procurement*): aktivitas yang terkait dengan bagaimana sumber daya diperoleh seperti fungsi pembelian input yang digunakan dalam value chain organisasi.



Gambar 2.2 *Value Chain*

2.6 Basis Data

Basis data merupakan suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh informasi (Kadir, 2005). Basis data adalah mekanisme yang digunakan untuk menyimpan informasi atau data. Informasi adalah sesuatu yang kita gunakan sehari-hari untuk berbagai alasan. Dengan basis data, pengguna dapat menyimpan data secara terorganisasi. Setelah data disimpan, informasi harus mudah diambil. Kriteria dapat digunakan untuk mengambil informasi. Cara data disimpan dalam basis data menentukan seberapa mudak mencari informasi berdasarkan banyak kriteria. Data pun harus mudah ditambahkan ke dalam basis data, dimodifikasi dan dihapus (J. Simamarta, dkk, 2006).

Basis data sendiri dapat didefinisikan dalam sejumlah sudut pandang (Fathayansyah, 2004), seperti:

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
3. Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

2.7 *Entity Relationship Model*

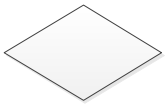
Model E-R adalah suatu model yang digunakan untuk menggambarkan data dalam bentuk entitas, atribut, dan hubungan antar entitas (Kadir, 2009). Model E-R berarti juga Diagram E-R, yaitu tidak mencerminkan bentuk fisik yang nantinya akan disimpan dalam database, melainkan hanya bersifat konseptual. E-R model didasarkan atas persepsi terhadap dunia nyata yang terdiri dari sekumpulan objek yang disebut entitas dan hubungan antar objek yang disebut relasi (Kusrini, 2007). Terdapat tiga notasi pada E-R Model yaitu *entity sets, relationship sets,* dan *attributes. Entity* adalah sebuah benda atau objek di dunia

nyata yang dapat dibedakan dari semua objek lainnya. Entitas mempunyai atribut yang membedakannya dengan entitas lainnya. *Entity sets* dilambangka dengan bentuk persegi panjang, seperti yang terlihat pada Gambar 2.3.



Gambar 2.3 *Entity*

Relationship adalah hubungan di antara beberapa entity. *Relationship sets* adalah sekumpulan relasi yang mempunyai tipe yang sama. *Relationship sets* digambarkan dengan diamond seperti yang telihat pada Gambar 2.4.



Gambar 2.4 *Relationship*

Atribut merupakan sebutan untuk mewakili suatu *entity*. Atribut dalam ERD dilambangkan dengan bentuk elips seperti yang terlihat pada Gambar 2.5.



Gambar 2.5 Atribut

2.8 Penelitian yang Terkait

Beberapa penilitian yang mempunyai keterkaitan dengan penelitian yang dilakukan beserta pembahasan dari hasil penelitian tersebut, diantaranya:

1. *Pemodelan Enterprise Arsitektur untuk Perguruan Tinggi di Indonesia*. (Yunis, 2009)

   Dalam penelitiannya, digunakan model arsitektur *enterprise* menekankan bagaimana cara penerapan TOGAF ADM sebagai salah satu metode yang bisa digunakan untuk melakukan pengembangan arsitektur *enterprise*. TOGAF ADM dikombinasikan dengan metode pengembangan sistem *Rational Unified Process* (RUP) dengan tujuan untuk mendapatkan rancangan atau *blueprint* dari sistem informasi perguruan tinggi yang akan dikembangkan. Dari hasil penelitian ini hanya menggambarkan penekanan model arsitektur secara umum saja dan belum menggambarkan integrasi dari masing-masing bagian antara fungsi bisnis utama dan fungsi bisnis pendukung pada perguruan tinggi.

2. *Analisis Pemodelan Arsitektur Enterprise untuk Mendukung Sistem Informasi Akademik dengan TOGAF (The Open Group Architecture Framework) (Studi Kasus AMIK AMIKOM Surakarta)*. (Hadi, dkk, 2013)

   Dalam penelitiannya, pemodelan arsitektur *enterprise* memberikan panduan dalam membuat blueprint untuk pengembangan sistem informasi akademik untuk data, aplikasi, bisnis dan teknologi. Tetapi pada pembahasannya tidak tampak penggunaan kerangka kerja TOGAF.

3. *Penerapan Metode EAP (Enterprise Architecture Planning) pada Pembuatan Blueprint Sistem Akademik*. (Miftahudin, dkk, 2013)

   Dalam penelitiannya, dianalisis mengenai proses bisnis dan merancang arsitektur data, arsitektur aplikasi dan arsitektur teknologi. Didalamnya, dibahas perancangan arsitektur data menggunakan ERD, arsitektur aplikasi digambarkan dengan menggunakan diagram konteks. Tetapi pada penelitian ini tidak menggambarkan keterkaitan antara fungsi bisnis, arsitektur data dan arsitektur aplikasi. Kemudian, didalamnya tidak dijelaskan juga mengenai perencanaan penerapan aplikasi tersebut.

4. *Perancangan Sistem Informasi Pemetaan Penelitian Dosen menggunakan Kerangka Zachman di STMIK Pontianak*. (Kasma, 2013)

   Pada penelitiannya, perancangan menggunakan kerangka Zachman yang terdiri dari 5 baris dan 3 kolom. Didalamnya dibahas mengenai arsitektur data menggunakan ERD, arsitektur aplikasi menggunakan *use case diagram* dan *class diagram* dan arsitektur teknologi. Pada penelitian ini juga dibahas mengenai *user interface* dari sistem informasi yang akan dibangun. Akan tetapi, pada penelitian ini tidak membahas perencanaan penerapan aplikasi tersebut.

5. *Perencanaan Arsitektur Enterprise Perguruan Tinggi dengan menggunakan Enterprise Unified Process (EUP)*. (Farid, 2009)

   Dalam penelitiannya, perencanaan arsitektur enterprise untuk menciptakan keselarasan antara bisnis dan teknologi informasi bagi kebutuhan organisasi. Metode yang digunakan adalah metode EUP. Pada penelitian ini, dibahas mengenai arsitektur data, arsitektur aplikasi dan arsitektur teknologi dan belum menggambarkan sebuah framework yang terintegrasi pada setiap bagian atau divisi pada perguruan tinggi.