

BAB II

TINJAUAN PUSTAKA

2.1 Pemerintah Daerah

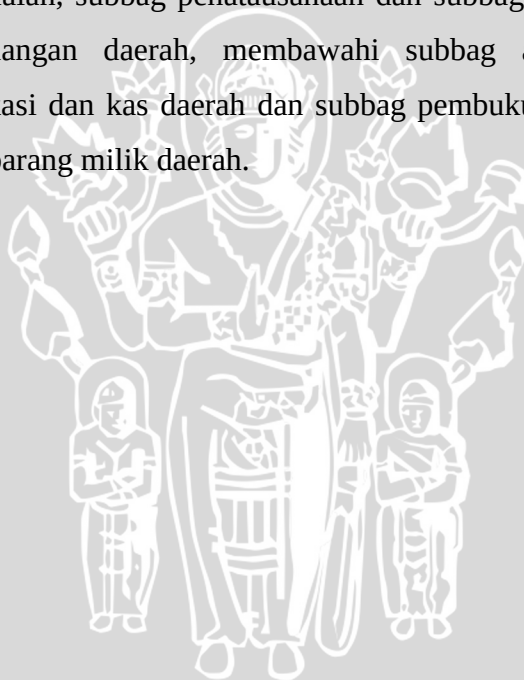
Undang-undang Dasar Republik Indonesia Tahun 1945 menyebutkan bahwa Pemerintah Daerah adalah penyelenggara urusan pemerintahan oleh Pemerintah Daerah dan Dewan Perwakilan Rakyat Daerah (DPRD) menurut asas otonomi dan tugas pembantuan dengan prinsip otonomi seluas-luasnya. Sedangkan Pemerintah Daerah adalah gubernur, bupati, dan/atau walikota, dan perangkat daerah sebagai unsur penyelenggara pemerintahan daerah. Selain itu, UUD 1945 juga mengamanatkan adanya hubungan keuangan, pelayanan umum serta pemanfaatan sumber daya alam dan sumber daya lainnya antara Pemerintah Pusat dan Pemerintah Daerah diatur dan dilaksanakan secara adil dan selaras berdasarkan Undang-undang. Untuk melaksanakan hal ini, ditetapkan Undang-undang No.33 tahun 2004 tentang Perimbangan Keuangan antara Pemerintah Pusat dan Pemerintah Daerah [OES-10].

Kabupaten Lombok Tengah terbentuk menjadi otonom berdasarkan Undang-undang Nomor 69 Tahun 1958 tentang Pembentukan daerah-daerah Tingkat II dalam wilayah daerah-daerah Tingkat I Bali, Nusa Tenggara Barat dan Nusa Tenggara Timur. Undang-undang tersebut disahkan pada tanggal 14 Agustus 1958. Namun demikian, sebelum terbentuk sebagai wilayah pemerintahan, entitas Lombok Tengah telah ada jauh sebelum itu. Beberapa momentum historis yang menandai keberadaan Lombok Tengah, antara lain adalah dengan dikeluarkan STB Nomor 248 Tahun 1898, kemudian pasca proklamasi Lombok Tengah secara integral menjadi bagian dari NKRI ditandai dengan pelantikan secara formal Kepala Pemerintahan setempat Lombok Tengah yang pertama pada tanggal 15 Oktober 1945.

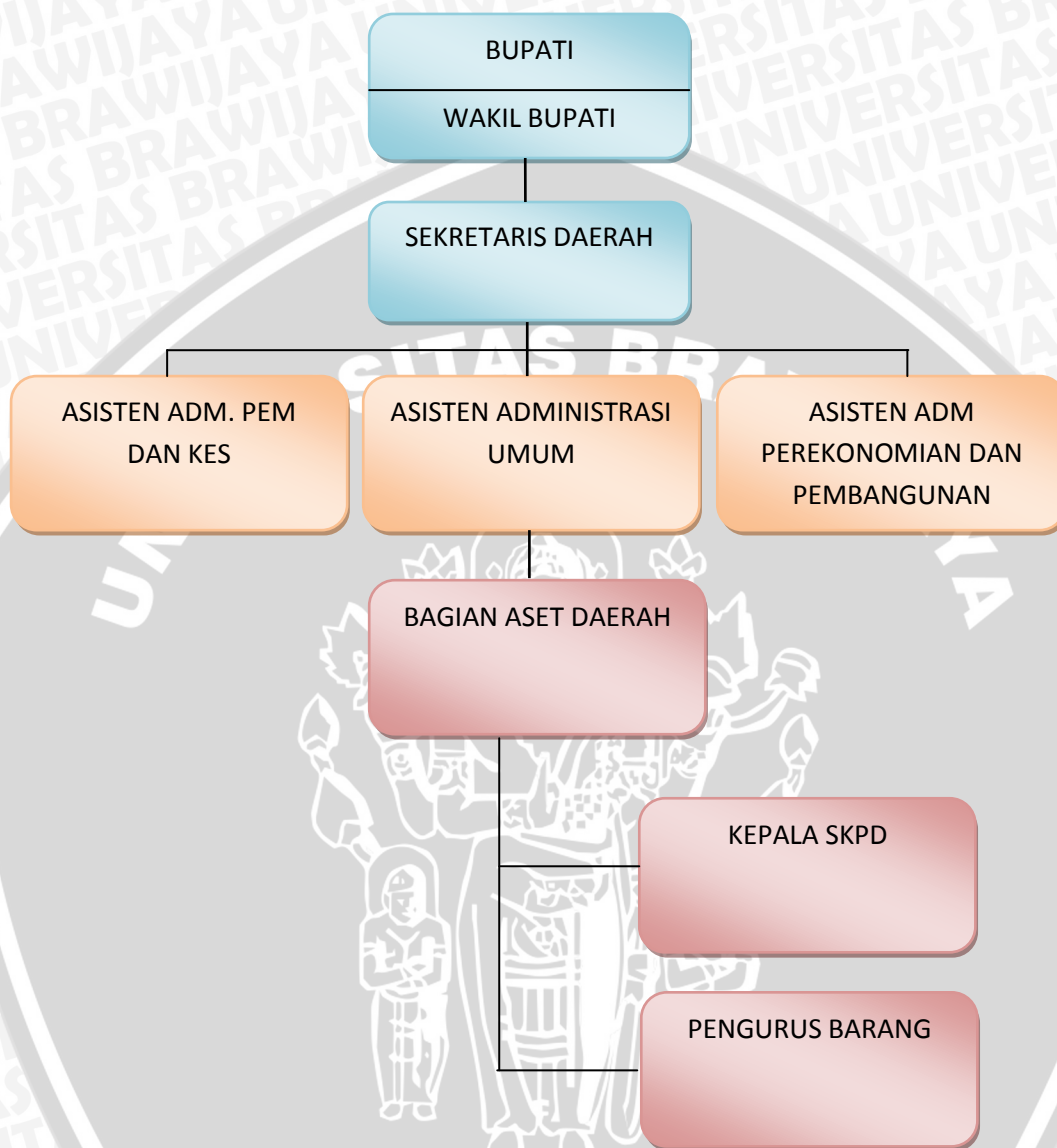
Kabupaten Lombok Tengah adalah salah satu Daerah Tingkat II di Provinsi Nusa Tenggara Barat. Ibu kota daerah ini ialah Praya. Kabupaten Lombok Tengah memiliki luas wilayah 1.208,39 km² dengan populasi sebanyak 860.209 jiwa. Sekretariat Daerah (Setda) Lombok Tengah (Loteng) dibentuk berdasarkan

Peraturan Daerah Kab.Lombok Tengah Nomor 3 Tahun 2008 Tentang Organisasi dan Tata Kelola Organisasi Perangkat Daerah Kab.Lombok Tengah.

Dalam sistem pengelolaan aset daerah,banyak subsistem yang terkait. Kepala daerah selaku penguasa pengelolaan aset/barang milik daerah, sekretaris daerah selaku pengelola barang milik daerah. Sekretariat Daerah Kabupaten Lombok Tengah dibentuk berdasarkan Peraturan Daerah Kab.Lombok Tengah Nomor 3 Tahun 2008 Tentang Organisasi dan Tata Kerja Organisasi Perangkat Daerah Kab.Lomboh Tengah yang terdiri dari 1 (Satu) Sekretaris Daerah, 3 (Tiga) Asisten , 12 (Dua Belas) Kepala Bagian dan 36 (Tiga Puluh Enam Kepala Sub Bagian). Dalam pengelolaan aset ini yang berperan adalah sekretaris daerah selaku pengelola barang milik daerah, bagian aset daerah, membawahi subbag pemanfaatan dan penilaian, subbag penatausahaan dan subbag pengendalian dan evaluasi, bagian keuangan daerah, membawahi subbag anggaran, subbag perbendaharaan,verifikasi dan kas daerah dan subbag pembukuan, kepala SKPD selaku pengguna aset/barang milik daerah.



STRUKTUR ORGANISASI SEKRETARIS DAERAH KABUPATEN LOMBOK TENGAH



Gambar 2.1 Struktur Organisasi Sekertaris Daerah

2.2 Keuangan Daerah

Keuangan Daerah adalah semua hak dan kewajiban daerah dalam rangka penyelenggara pemerintahan daerah yang dapat dinilai dengan uang termasuk didalamnya segala bentuk kekayaan yang berhubungan dengan hak dan kewajiban daerah tersebut. Pengelolaan Keuangan Daerah meliputi keseluruhan kegiatan yang meliputi perencanaan, pelaksanaan, penatausahaan, pelaporan,

pertanggungjawaban, dan pengawasan keuangan daerah. Dalam Peraturan Pemerintah Nomor 58 tahun 2005 tentang Pengelolaan Keuangan Daerah disebutkan bahwa salah satu lingkup pengelolaan keuangan daerah adalah pengelolaan barang milik daerah. Oleh karena itu, efektifitas dan efisiensi pengelolaan barang milik daerah akan mempengaruhi efektivitas dan efisiensi pengelolaan keuangan daerah [OEP-10].

2.2.1 Laporan Keuangan PEMDA

Laporan keuangan pemerintah daerah adalah suatu bentuk pertanggungjawaban pemerintah daerah kepada *stakeholder* yang di dalamnya mencakup berbagai macam pekerjaan yang membutuhkan keuangan, termasuk komponen aset yang tercermin dalam neraca daerah di mana setiap tahun dibuatkan laporannya setelah pelaksanaan anggaran. Pembuatan laporan keuangan ini dilakukan setelah adanya laporan keuangan Satuan Kerja Pemerintah Daerah (SKPD) dengan batasan waktu dua bulan setelah tahun anggaran berakhir. Menurut Standar Akuntansi Pemerintah Nomor 24 Tahun 2005 Laporan Keuangan pemerintah terdiri atas Laporan Arus Kas, Laporan Realisasi Anggaran, catatan atas Laporan Keuangan, dan Neraca SKPD yang selanjutnya dikonsolidasikan menjadi laporan keuangan pemerintah daerah [YSF-10].

2.2.2 Manajemen Aset

Definisi manajemen menurut Ensiklopedia Administrasi Indonesia, dalam LAN (2007:3) adalah segenap kekuatan menggerakkan sekelompok orang yang mengerahkan fasilitas dalam satu usaha kerja untuk mencapai tujuan tertentu.

Pengertian aset secara umum menurut Siregar (2004:178) adalah barang (*thing*) atau sesuatu barang (*anything*) yang mempunyai nilai ekonomi (*economic value*), nilai komersial (*commercial value*) atau nilai tukar (*exchange value*) yang dimiliki oleh badan usaha, instansi atau individu (perorangan).

Berdasarkan Undang-undang No.1 Tahun 2004 yang dimaksud dengan Barang Milik Daerah adalah semua barang yang dibeli atau diperoleh atas beban APBD atau yang berasal dari perolehan lainnya yang sah. Manajemen aset sangat penting karena akan mendukung pelaksanaan tugas dan fungsi instansi pemerintah. Alasan pentingnya manajemen aset meliputi kebutuhan untuk

menegaskan posisi hukum setiap aset terutama tanah dan bangunan yang seringkali menjadi objek sengketa antar lebih dari satu instansi, kebutuhan perawatan aset, penegasan pihak yang bertanggung jawab mengelola aset ini.

Makna manajemen aset daerah adalah melaksanakan pengelolaan aset/Barang Milik Daerah (BMD) berdasarkan prinsip dasar-dasar manajemen aset terhadap aset/BMD dengan mengikuti landasan kebijakan yang diatur berdasarkan Undang-undang, Peraturan Pementah, Keppres, Kepmen dan Surat keputusan lainnya yang berhubungan dengan pengaturan/pengelolaan aset daerah [IDS-13].

2.2.3 Aset/Barang

Aset atau barang daerah merupakan potensi ekonomi yang dimiliki oleh daerah. Potensi ekonomi bermakna adanya manfaat *financial* dan ekonomi yang bisa diperoleh pada masa yang akan datang, yang bisa menunjang peran dan fungsi pemerintah daerah sebagai pemberi pelayanan publik kepada masyarakat.

Aset daerah banyak macamnya dan menurut peraturan pemerintah, aset pemerintah termasuk aset daerah terdiri atas enam golongan serta aset lainnya [YSF-10] :

1. Golongan Tanah
2. Golongan Peralatan dan Mesin
3. Golongan Gedung dan Bangunan
4. Golongan Jalan, Irigasi dan Jaringan
5. Golongan Aset Tetap Lainnya
6. Golongan Kontruksi dalam Pengerjaan
7. Golongan Aset Lainnya

2.2.4 Penggolongan Barang Milik Daerah

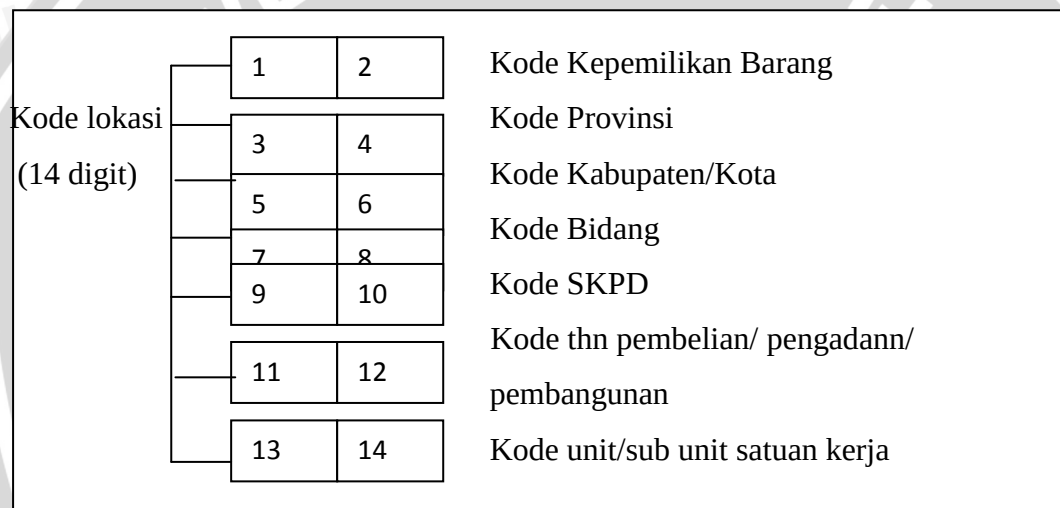
Pada saat pencatatan dalam KIB, terlebih dahulu dilakukan penggolongan dan kodifikasi barang milik daerah. Penggolongan barang ini akan terdiri dari 2 kode yaitu kode lokasi dan kode barang. Kedua kode ini terdiri dari 2 14 digit. Sehingga setiap barang, nantinya akan memiliki 2 kode yang masing-masing terdiri dari 14 digit. Untuk tertib administrasi, setiap barang akan diberikan kode yang terdiri dari kedua macam kode tersebut.

Jadi setiap barang yang dimiliki pemerintah daerah akan diberikan kode yang terkait dengan kode lokasi dan kode barang. Kode lokasi terkait dengan kepemilikan dari barang tersebut, sedangkan kode barang terkait dengan klasifikasi/penggolongan barang. Setiap kode barang dan kode lokasi ini harus dicantumkan pada setiap barang inventaris.

2.2.4.2 Kode Lokasi

Kode lokasi barang menerangkan mengenai kepemilikan barang beserta tahun perolehan/pembelian/pembangunan dari barang milik daerah yang dimiliki suatu SKPD.

Bagan 2.1 Kode Lokasi Barang



Sumber : Diolah dari Permendagri nomor 17 tahun 2007

Digit pertama dan kedua dari kode lokasi menerangkan kode kepemilikan untuk masing-masing tingkatan pemerintah.

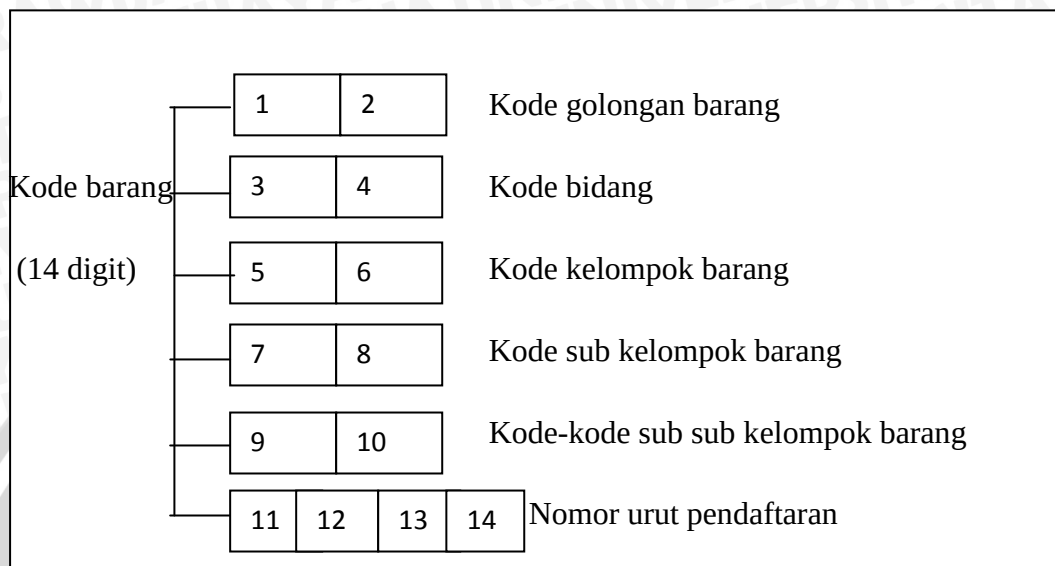
- a 12 untuk barang milik pemerintah kabupaten/kota
- b 11 untuk barang milik pemerintah provinsi
- c 00 untuk barang milik pemerintah pusat

Digit ke-3 dan 4 diisi kode Provinsi dan untuk kode Provinsi Nusa Tenggara Barat adalah 15. Digit ke-5 dan 6 untuk kode kabupaten/kota. Digit ke-7 dan 8 kode bidang tugas yang terdapat dalam suatu pemerintah daerah. Digit ke-9 dan 10 adalah nomor kode Satuan Kerja Pemerintah Daerah (SKPD). Digit ke-11 dan 12 kode tahun pembelian/pengadaan/pembangunan barang. Digit ke-13 dan 14 diisi nomor kode Sub Unit Kerja.

2.2.4.3 Kode Barang

Kode barang juga terdiri dari 14 digit. Kode ini menggambarkan klasifikasi/penggolongan barang.

Bagan 4.2 Kode Barang



Sumber : Diolah dari Permendagri nomor 17 tahun 2007

Untuk digit ke-1 dan 2 kode barang adalah golongan barang. Nomor kode golongan barang diklasifikasikan menjadi 6 golongan, yaitu :

- Untuk tanah, kodenya 01
- Untuk mesin dan peralatan, kodenya 02
- Untuk gedung dan bangunan, kodenya 03
- Untuk jalan, irigasi dan jaringan, kodenya 04
- Untuk aset tetap lainnya, kodenya 05
- Untuk konstruksi dalam pengerjaan, kodenya 06

2.2.5 Buku Induk Inventaris, Buku Inventaris dan Kartu Inventaris Barang

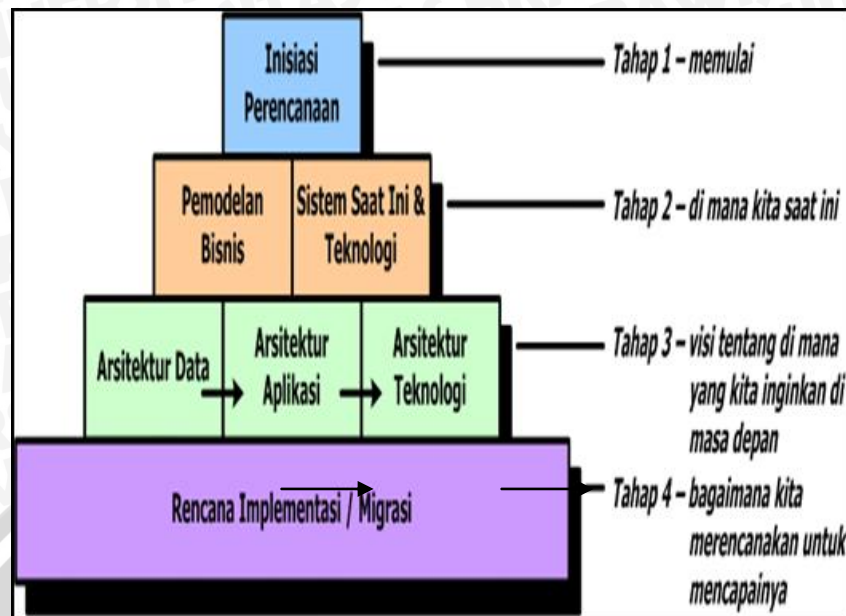
Buku Induk Inventaris (BII) merupakan gabungan/kompilasi buku inventaris (Tingkat Pengelola). Sedangkan Buku Inventaris (BI) adalah himpunan catatan data teknis dan administratif yang terdiri atas catatan kartu barang inventaris (sebagai hasil sensus) di tiap-tiap Satker/SKPD yang dilaksanakan secara serentak pada waktu tertentu. Kartu Inventaris Barang (KIB) merupakan kartu yang digunakan untuk mencatat data dari barang/aset yang ada seperti kode barang, kode lokasi barang jumlah barang dan lain-lain.

2.3 Arsitektur Enterprise

Arsitektur lazimnya dimengerti sebagai studi atau pekerjaan merancang bangunan. Sesungguhnya pengertian arsitektur tidak terbatas akan rancangan bangunan. Arsitektur (*Architecture*) adalah cara dimana sebuah sistem yang terdiri dari *network*, *hardware* dan *software* di strukturkan. Arsitektur pada dasarnya menceritakan bagaimana bentuk konstruksi sebuah sistem, bagaimana setiap komponen sistem disusun, dan bagaimana semua aturan dan *interface* (penghubung sistem) digunakan untuk mengintegrasikan seluruh komponen yang ada tersebut. Arsitektur juga mendefinisikan fungsi, deskripsi dari format data dan prosedur yang digunakan komunikasi diantara setiap node dan *workstation* [TSY-09].

Kata enterprise dapat didefinisikan sebagai organisasi (atau badan lintas organisasi) yang mendukung lingkup bisnis dan misi yang telah ditetapkan. *Enterprise* mencakup sumber daya yang saling berkaitan (manusia, organisasi, dan teknologi) yang harus mengkoordinasikan fungsinya dan berbagi informasi dalam mendukung misi bersama (atau sekumpulan misi yanag berkaitan) [TSY-09].

Seperti yang dikatakan oleh John Zachman bahwa *enterprise architecture* sudah bukan lagi suatu pilihan tetapi sudah menjadi suatu kewajiban. Setiap perusahaan mencari tingkatan kinerja misi mereka. *Enterprise architecture* adalah satu praktek manajemen untuk memaksimalkan kontribusi dari sumber daya perusahaan, investasi IT, dan aktivitas pembangunan sistem untuk mencapai tujuan kinerjanya. Untuk mencapai misi organisasi melalui kinerja optimal dari proses bisnis dengan efisiensi lingkungan teknologi informasi (TI) maka penerapan *enterprise architecture* harus dimasukkan kedalam *roadmap* dari perusahaan. *enterprise architecture* menolong mengorganisir dan memperjelas hubungan di antara tujuan strategis perusahaan, investasi, solusi bisnis dan peningkatan kinerja terukur. Untuk mencapai peningkatan kinerja sasaran, *enterprise architecture* harus kuat dan sepenuhnya terintegrasi dengan area praktek lainnya termasuk perencanaan strategis, perencanaan modal dan investasi [IYS-10].



Gambar 2.2 Lapisan Perencanaan Arsitektur Enterprise [SUR-09]

Dari tahun 2004 dan tahun 2005 bahwa ada kemajuan yang cepat didalam pengimplementasian *enterprise architecture* di negara-negara seperti India, Singapura dan Brazil. Berdasarkan survei dari pada tahun 2003 *enterprise architecture* diterapkan yang paling utama adalah untuk mengatur kompleksitas (*managing complexity*), memberikan peta jalan (*road map*) tetapi lambat laun hanya dalam tempo 2 tahun yaitu pada tahun 2005 penerapan *enterprise architecture* berubah menjadi sarana untuk mendukung didalam pengambilan keputusan (*supports decision making*) dan untuk mengatur portofolio TI bagi perusahaan (*manage IT portofolio*). Jadi kecenderungan motivasi dari perusahaan untuk menerapkan *enterprise architecture* adalah untuk mengatur segala sesuatu yang kompleks di perusahaan dan membuat *road map*. Dan semakin mudahnya level manajemen dalam mengambil keputusan dan membuat skala prioritas pekerjaan mana yang akan dilakukan. Beberapa alasan nyata didalam penerapan *enterprise architecture* yaitu pada tahun 2004 terjadi perkembangan yang cepat untuk transformasi *road map* dan terjadi penurunan pada bisnis dan TI yang selaras sebagai alasan penerapan *enterprise architecture*. Pada tahun 2005 bisnis dan TI yang selaras yang menjadi alasan paling utama untuk penerapan *enterprise architecture*, sedangkan transformasi *road map* hampir sama dengan tahun

sebelumnya. Untuk pembaharuan infrastruktur (*infrastructure renewel*) mengalami penurunan [IYS-10].

Enterprise architecture sangat penting untuk meningkatkan system informasi dan meningkatkan system informasi dan mengembangkan system baru yang dapat mengoptimalkan nilai misi perusahaan. Ini dipenuhi pada kondisi bisnis (misalnya misi, fungsi bisnis, aliran keterangan, dan lingkungan sistem) dan kondisi teknis (misalnya perangkat lunak, perangkat keras, komunikasi), dan meliputi suatu rencana peralihan untuk transisi dari lingkungan dasar ke lingkungan sasaran [IYS-10].

2.4 Kerangka Kerja Arsitektur Enterprise

Untuk dapat memaksimalkan peran *enterprise architecture*, diperlukan sebuah *framework* yang memiliki model simbolis untuk menspesifikasikan berbagai fase *enterprise architecture*. Dari sebuah model simbolis diinterpretasikan menjadi model semantik, model ini mengekspresikan makna dari masing-masing symbol pada sebuah model. Untuk dapat mengerti relasi antar model semantik dengan arsitektur, maka harus dipahami tujuan dari modeling yaitu untuk memprediksi realitas dari keadaan yang sebenarnya [SOF-11].

Diantara *framework* yang paling banyak dan lazim digunakan dalam dunia industry maupun pemerintahan adalah *Zachman Framework*, *TOGAF*, *The Federal Enterprise Architecture Framework (FEAF)*, *The Integrated Archtitecture Framework (IAF)* dan *The Department Of Defense Architectural Framework (DoDAF)* [SOF-11].

Kajian yang sering dijadikan sebagai acuan kriteria pengukuran seperti yang dikeluarkan oleh Roger Session yang melakukan perbandingan dengan memberikan 12 aspek sebagai perbandingan. Kriteria pengukuran yaitu :

- a *Taxomony Completeness*, kriteria seberapa baik pengklasifikasikan dalam *framework*.
- b *Process Completeness*, seberapa jelas langkah dan panduan yang dalam implementasinya.
- c *Referensi Model Guidance*, seberapa bermanfaat dalam perancangan *reference models*.

- d *Practice Guidance*, seberapa berperan dalam praktek sehari-hari di perusahaan.
- e *Maturity Model*, seberapa efektif dan mature di perusahaan.
- f *Business Focus*, seberapa besar peranan *framework* untuk mengurangi biaya atau meningkatkan pendapatan.
- g *Governance Guides*, seberapa membantu sebuah *framework* dapat menciptakan tata kelola (*governance*) yang efektif.
- h *Partitioning Guidance*, seberapa baik dalam memandu perancangan *autonomous partitions* dari perusahaan, khususnya untuk menangani kompleksitas yang dihadapi.
- i *Prescriptive Catalog*, seberapa baik untuk membuat catalog dari *architectural asset* yang dapat di *reuse* di masa yang akan datang.
- j *Vendor Neutrality*, menekankan bahwa perusahaan harus terbebas dari tingkat ketergantungan atau *intervensi* dengan vendor.
- k *Information Availability*, menekankan kualitas dan kemudahan untuk memperoleh informasi.
- l *Time to Value Refers*, kriteria ini mengacu waktu yang diperlukan untuk implementasi bagi perusahaan.

Tabel 2.1 Perbandingan dari kerangka kerja Arsitektur Enterprise

Criteria	Ratings			
	Zachman	TOGAF	FEA	Gartner
Taxonomy Completeness	4	2	2	1
Process Completeness	1	4	2	3
Reference Model Guidance	1	3	4	1
Practice Guidance	1	2	2	4
Maturity Model	1	1	3	2
Business Focus	1	2	1	4
Governance Guidance	1	2	3	3
Partitioning Guidance	1	2	4	3
Prescriptive Catalog	1	2	4	2
Vendor Neutrality	2	4	3	1
Information Availability	2	4	2	1
Time to Value	1	3	1	4

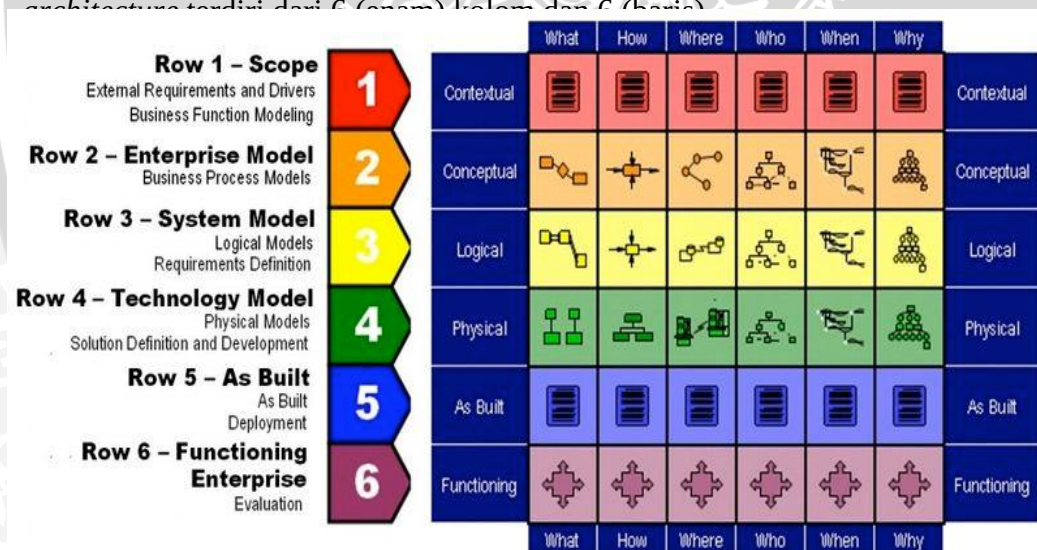
(Sumber: Sessions, Roger. (2007))

Dari hasil tersebut terlihat bahwa masing-masing dari *framework* memiliki kelemahan dan kekuatan [SOF-11].

2.5 Zachman Framework

Kerangka kerja Zachman pertama kali diperkenalkan oleh John Zachman pada tahun 1987, kemudian diperluas dan diformulasikan oleh Sowa dan Zachman pada tahun 1992. Setiap model kerangka kerja mendefinisikan entitas-entitas arsitektur ke dalam baris-baris dan atribut untuk setiap entitas ke dalam kolom-kolom [DYM-12].

Kerangka kerja Zachman adalah pendekatan klasifikasi artifak *enterprise architecture* yang diterima sebagai standar *de-facto*. Kerangka kerja ini disanjung karena keunikannya dalam klasifikasi arsitektur dalam perspektif enterprise. Kerangka kerja Zachman bukan suatu metodologi untuk mengembangkan *enterprise architecture*, akan tetapi kerangka kerja *Zachman* merupakan kerangka kerja untuk mengkategorikan artifak *enterprise architecture*. Kerangka kerja Zachman dapat dimanfaatkan untuk menentukan apakah suatu metodologi meliputi semua aspek dalam *enterprise architecture* atau aspek apa saja yang dicakup oleh metodologi [DYM-12]. Kerangka kerja Zachman untuk *enterprise architecture* terdiri dari 6 (enam) kolom dan 6 (enam) baris.



Gambar 2. 3 Kerangka Kerja Zachman Framework [JUC-01]

Secara umum tiap kolom merepresentasikan fokus, abstraksi atau topik *enterprise architecture* [DYM-12].

1. *What* (data) : menggambarkan kesatuan yang dianggap penting dalam bisnis. Kesatuan tersebut adalah hal-hal yang informasinya perlu dipelihara.
2. *How* (fungsi) : mendefinisikan fungsi atau aktivitas. Input dan output juga dipertimbangkan pada kolom ini.
3. *Where* (jaringan) : menunjukkan lokasi geografis dan hubungan antara aktivitas dalam organisasi, meliputi lokasi geografis bisnis yang utama.
4. *Who* (orang) : mewakili manusia dalam organisasi dan metric untuk mengukur kemampuan dan kinerjanya. Kolom ini juga berhubungan dengan *user interface* dan hubungan antara manusia dan pekerjaan yang menjadi tanggung jawabnya.
5. *When* (waktu) : mewakili waktu atau kegiatan yang menunjukkan criteria kinerja. Kolom ini berguna untuk mendesain jadwal dan memproses arsitektur.
6. *Why* (motivasi) : menjelaskan motivasi dari organisasi dan pekerjanya. Disini terlihat tujuan, sasaran, rencana bisnis, arsitektur pengetahuan, alasan pikiran dan pengambilan keputusan dalam organisasi.

Setiap baris pada kerangka kerja Zachman mewakili perspektif yang berbeda dan unik [DYM-12].

1. Perspektif Perencana (*Ballpark View*), yaitu menetapkan konteks, latar belakang dan tujuan *enterprise*.
2. Perspektif Pemilik (*Owner's View*), yaitu menetapkan model-model konseptual dari *enterprise*.
3. Perspektif Perancang (*Designer's View*), yaitu menetapkan model-model system informasi sekaligus menjembatani hal-hal yang diinginkan pemilik dan hal-hal yang dapat direalisasikan secara teknis dan fisik.
4. Perspektif Pembangunan (*Builder's View*), yaitu menetapkan rancangan teknis dan fisik yang digunakan dalam mengawasi implementasi teknis dan fisik.
5. Perspektif Subkontraktor (*Subcontractor*), yaitu menetapkan peran dan rujukan bagi pihak yang bertanggung jawab untuk melakukan

pembangunan secara teknis dan fisik serta mengadakan komponen-komponen yang diperlukan.

6. Perspektif Fungsi Sistem, yaitu merepresentasikan perspektif pengguna dan wujud nyata hasil implementasi.

2.6 BPMN

BPMN adalah singkatan dari *Business Process Modeling Notation*, yaitu suatu metodologi baru yang dikembangkan oleh *Business Process Modeling Initiative* sebagai suatu standar baru pada pemodelan proses bisnis, dan juga sebagai alat desain pada system yang kompleks. Dikembangkan oleh *Business Process Management Initiative* (BPMNI). Spesifikasi BPMN versi 1.0 diperkenalkan pada Mei 2004. Spesifikasi ini merepresentasikan usaha BPMI *Notation Working Group* selama lebih dari dua tahun. [ROS-07]

Tujuan utama dari BPMN adalah menyediakan notasi mudah digunakan dan bisa dimengerti oleh semua orang yang terlibat dalam bisnis, yang meliputi bisnis analis yang memodelkan proses bisnis, pengembang teknik yang membangun system yang melaksanakan bisnis, dan berbagai tingkatan manajemen yang harus dapat membaca dan memahami proses diagram dengan cepat sehingga dapat membantu dalam pengambilan keputusan. [NEW-04]

BPMN menggambarkan suatu *Business Process Diagram* (BPD) yang mana BPD ini didasarkan kepada teknik diagram alur (*Flowchart Technique*), dirangkai untuk membuat model-model grafis dari operasi-operasi bisnis. Sebuah *Business Process Model* kemudian sebuah objek-objek grafis dari jaringan, dimana terdapat aktifitas-aktifitas dan control-kontrol alur yang mendefinisikan urutan kerja. [KAH-07]

Sebuah BPD terbuat dari serangkaian elemen-elemen grafis. Elemen-elemen ini mempermudah pengembangan dari diagram-diagram sederhana yang terasa familiar bagi analisis bisnis (misal : diagram alur). Elemen-elemen ini dipilih untuk bedakan satu diantara lainnya dan menggunakan bentuk-bentuk yang familiar bagi kebanyakan pemodel. Sebagai contoh, aktivitas digambarkan dengan segi empat (*rectangles*), keputusan dilambangkan dengan lambang berlian (*diamonds*). Penggambaran ini seharusnya menekankan bahwa tujuan dari pengembangan BPMN ini adalah untuk menciptakan mekanisme sederhana bagi

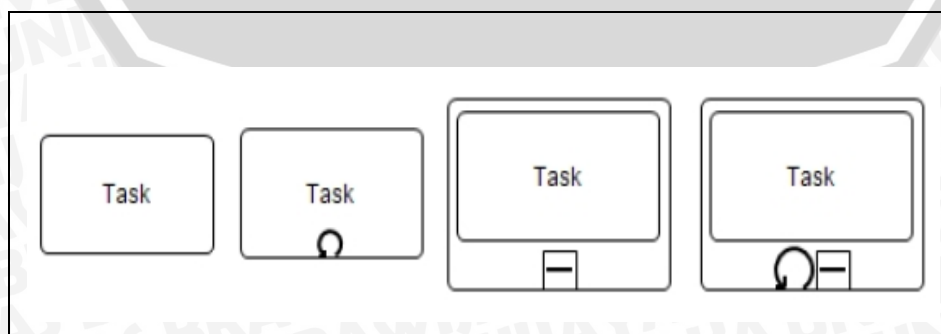
pembuatan model-model proses bisnis, dimana pada saat yang sama mampu mengenai kompleksitas inheren dari proses bisnis. Pendekatan ini digunakan untuk menangani persyaratan-persyaratan yang bertentangan yang digunakan untuk mengorganisasikan aspek-aspek grafis dari notasi dalam ke dalam kategori-kategori dasar dari elemen-elemen, informasi dan variasi tambahan dapat ditambahkan untuk mendukung persyaratan-persyaratan dari kompleks tanpa merubah secara dramatis dari bentuk diagram dasar. Empat kategori dasar dari elemen-elemen ini adalah : [OWE-03]

1. *Flow Object*, terdiri atas :
 - a. Event dipresentasikan dalam bentuk lingkaran dan menjelaskan apa yang terjadi saat itu. Ada tiga jenis event, yaitu *start*, *intermediate*, dan *end*. Masing-masing mewakili kejadian dimulainya proses bisnis, interupsi proses bisnis, dan akhir dari proses bisnis. Untuk setiap jenis event tersebut sendiri terbagi atas beberapa jenis, misalnya message start, yang dilambangkan seperti start event namun mendapatkan tambahan langsung amplop di dalamnya yang berarti ada pesan event tersebut dimulai dengan masuknya pesan :



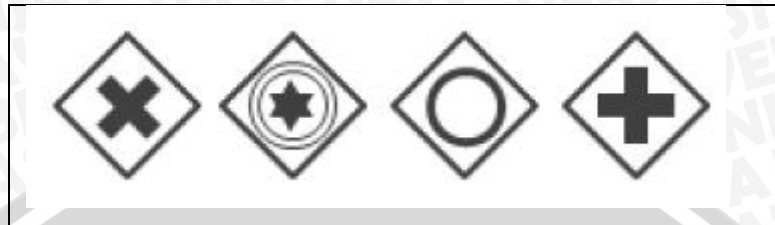
Gambar 2. 4 Element Start, Intermediete, dan end Event [BRA-08]

- b. *Activity* merepresentasikan pekerjaan (*task*) yang harus diselesaikan. Ada empat macam activity, yaitu *task*, *looping task*, *sub process*, dan *looping subprocess*. Keempatnya secara terurut diperlihatkan pada gambar di bawah ini.



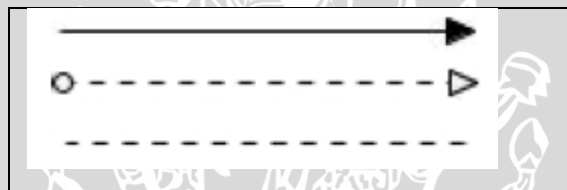
Gambar 2. 5 Element Start, Intermediete, dan End Event [BRA-08]

- c. *Gateway* merepresentasikan pemecahan alur yang terdapat di dalam proses bisnis. Ada berbagai macam gateway, yaitu *exclusive data based*, *exclusive event based*, *inclusive event based*, dan *parallel*. Keempatnya secara terurut ditunjukkan dalam gambar dibawah ini.



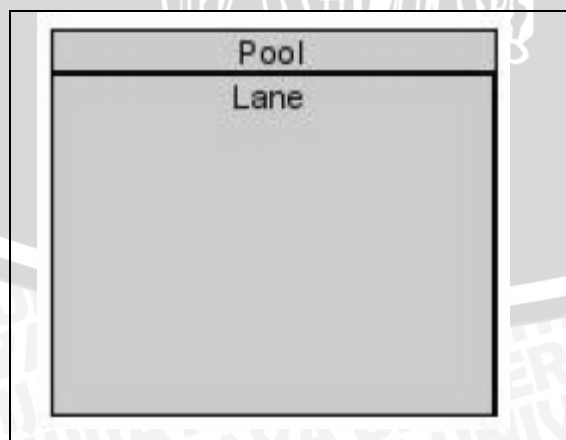
Gambar 2. 6 Element-element Activity [BRA-08]

2. *Connecting Object*, terdiri atas :
 - a. *Sequence flow*, merepresentasikan pilihan default untul menjalankan proses.
 - b. *Message flow*, merepresentasikan aliran antar proses.
 - c. *Association*, digunakan untuk menghubungkan elemen dengan artifact.



Gambar 2. 7 Element *sequence flow*, *message flow*, and *association* [BRA-08]

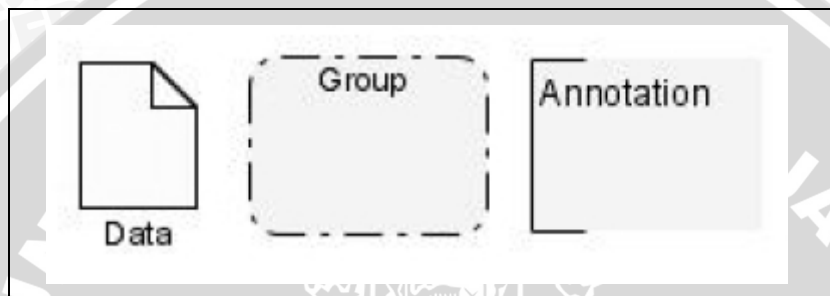
3. *Swimlanes*. Elemen ini digunakan untuk mengkategorikan secara visual seluruh elemen dalam diagram. Ada dua jenis *swimlanes*, yaitu *pool* dan *lane*. Perbedaannya adalah *lane* terletak di bagian dalam *pool* untuk mengkategorisasi elemen-elemen di dalam *pool* menjadi lebih spesifik.



Gambar 2.8 Pool dan Lane [BRA-08]

4. *Artifacts*. Elemen ini digunakan untuk member penjelasan di diagram. Elemen ini terdiri atas tiga jenis, yaitu :

- Data object*, digunakan untuk menjelaskan data apa yang dibutuhkan dalam proses.
- Group*, untuk mengelompokkan sejumlah aktivitas didalam proses tanpa mempengaruhi proses yang sedang berjalan.
- Annotation*, digunakan member catatan diagram menjadi lebih mudah dimengerti.



Gambar 2. 9 Element data object, group, and Annotation [BRA-08]

2.7 Diagram UML (Unified Modelling Language)

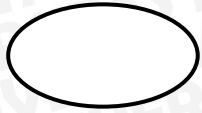
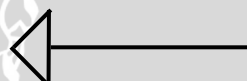
Unified Modelling Language (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem [DRS-03].

2.7.1 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 [DRS-03]. Elemen – elemen use case diagram ditunjukkan pada tabel 2.2 :

Tabel 2. 2 Elemen – elemen *Use Case Diagram*

No	Nama Elemen	Fungsi	Notasi
1	Aktor	Menggambarkan tokoh atau sistem yang memperoleh keuntungan dan berada di luar sistem. Aktor ditempatkan di	

No	Nama Elemen	Fungsi	Notasi
		luar <i>subject boundary</i> .	
2	<i>Use Case</i>	Mewakili sebuah bagian dari fungsionalitas sistem dan ditempatkan dalam <i>boudary system</i> .	
3	<i>Association Relationship</i>	Menghubungkan aktor untuk berinteraksi dengan <i>use case</i> .	
4	<i>Include Relationship</i>	Menunjukkan <i>inclusion</i> fungsionalitas dari sebuah use case dengan use case lainnya. Arah panah dari base use case ke included use case.	
5	<i>Generalization Relationship</i>	Menunjukkan generalisasi dari <i>use case</i> khusus ke yang lebih umum.	

(Sumber : Desanti, 2010)

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 [DRS-03].

2.7.2 Activity Diagram

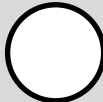
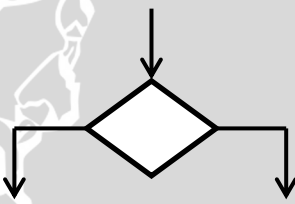
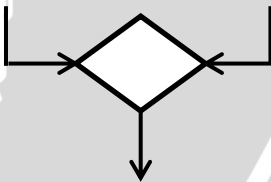
Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan

bagaimana aktor menggunakan sistem untuk melakukan aktivitas [DRS-03].

Elemen – elemen *activity diagram* ditunjukkan pada tabel 2.3 :

Tabel 2. 3 Elemen – elemen *activity diagram*

No	Nama Elemen	Fungsi	Notasi
1.	Aktifitas	Untuk mewakili kumpulan aksi	
2.	<i>Control Flow</i>	Menunjukkan rangkaian pelaksanaan	
3.	<i>Initial Node</i>	Menandakan awal dari kumpulan aksi atau aktivitas	
4.	<i>Final – activity Node</i>	Untuk menghentikan seluruh <i>control flows</i> atau <i>object flows</i> pada sebuah aktivitas atau aksi	
5.	<i>Decision Node</i>	Untuk mewakili suatu kondisi pengujian yang bertujuan untuk memastikan bahwa <i>control flow</i> hanya menuju ke satu arah	
6.	<i>Merge Node</i>	Untuk menyatukan kembali <i>decision path</i> yang dibuat dengan menggunakan <i>decision node</i>	

(Sumber : Desanti, 2010)

2.7.3 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).

Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok :

1. Nama (dan stereotype)
2. Atribut
3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

- *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan
- *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya
- *Public*, dapat dipanggil oleh siapa saja

Hubungan Antar *Class*:

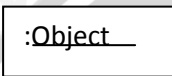
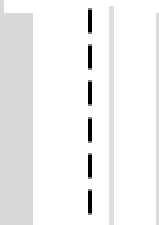
1. Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain. Panah *navigability* menunjukkan arah *query* antar *class*.
2. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas..”).
3. Pewarisan, yaitu hubungan hirarkis antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metoda *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
4. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di-*passing* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence diagram* yang akan dijelaskan kemudian [DRS-03].

2.7.4 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 atas 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[DRS-03].

Tabel 2. 4 Komponen-komponen *Sequence Diagram*

Komponen	Nama Komponen	Keterangan
	<i>Object</i>	Merupakan <i>instance</i> dari sebuah <i>class</i> dan dituliskan tersusun secara <i>horizontal</i> . Digambarkan sebagai sebuah <i>class</i> (kotak) dengan nama <i>object</i> didalamnya.
	<i>Actor</i>	<i>Actor</i> juga dapat berkomunikasi dengan <i>object</i> , maka <i>actor</i> juga dapat diurutkan sebagai kolom.
	<i>Lifeline</i>	Mengidentifikasi keberadaan sebuah <i>object</i> dalam basis waktu. Notasi untuk <i>Lifeline</i> adalah garis putus-putus vertikal yang ditarik dari sebuah <i>object</i> .
	<i>Activation</i>	<i>Activation</i> dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah <i>lifeline</i> . Mengindikasikan sebuah objek yang akan melakukan sebuah aksi.
	<i>Boundary</i>	<i>Boundary</i> terletak di antara sistem dengan dunia sekelilingnya. Semua form,

Komponen	Nama Komponen	Keterangan
		laporan-laporan, antar muka ke perangkat keras seperti <i>printer</i> atau <i>scanner</i> dan antarmuka ke sistem lain termasuk dalam kategori.
	<i>Control</i>	<i>Control</i> berhubungan dengan fungsionalitas seperti pemanfaatan sumber daya, pemrosesan terdistribusi atau penayangan kesalahan.
	<i>Entity</i>	<i>Entity</i> digunakan untuk menangani informasi yang mungkin akan disimpan secara permanen. <i>Entity</i> bisa juga merupakan sebuah tabel pada struktur basis data.
	<i>Message</i>	<i>Message</i> digambarkan dengan anak panah <i>horizontal</i> antara <i>Activation</i> <i>Message</i> mengidentifikasi komunikasi antara <i>object-object</i> .
	<i>Self-message</i>	<i>Self-message</i> atau panggilan mandiri mengidentifikasi komunikasi kembali ke dalam sebuah objek itu sendiri.

(Sumber : Ulum, 2010)

2.8 Business Process Reengineering

Michael Hammer dan James Champy menyatakan bahwa *Business Process Reengineering* BPR adalah “Pemikiran dan perancangan ulang suatu sistem bisnis secara mendasar (*fundamental*) dan radikal untuk mendapatkan perbaikan secara dramatis pada saat kritis, dengan mengukur kinerja saat ini melalui elemen-elemen biaya, kualitas, pelayanan, dan kecepatan.” Dalam definisi dari Michael Hammer diatas, terdapat empat kata kunci yaitu fundamental, radikal, dramatis dan proses [HMC-95].

Ada dua pendekatan utama untuk merancang ulang proses guna tercapainya perbaikan kinerja [PJR-95]:

2.8.1 Perancangan ulang sistematis

Mengidentifikasi dan memahami proses-proses yang ada kemudian mendesain kembali proses-proses tersebut secara sistematis untuk menciptakan proses-proses baru, guna memberikan hasil yang diinginkan.

Tujuan perancangan proses pada pendekatan ini adalah meningkatkan nilai tambah pada proses yang ada sekarang dengan cara mengeliminasi semua kegiatan yang tidak bernilai tambah dan merampingkan kegiatan yang bernilai tambah. Ada 4 prinsip dalam perancangan ulang sistematis yaitu :

1 *Eliminate*

Proses menghapus aktifitas proses bisnis yang tidak diperlukan, tahap-tahap dalam proses bisnis yang tidak bernilai tambah harus dieliminasi. Contoh-contoh kegiatan yangb sering ada dan cenderung tidak bernilai tambah sehingga mempunyai potensi untuk dieliminasi :

- Waktu Menunggu, dalam waktu tunggu ada biaya tambahan untuk material, atau sumber daya manusia. Pekerjaan akan terhambat atau terhenti karena harus menunggu material tersebut tiba, menunggu, keputusan-asaan dan sebagainya.
- Pemrosesan, pemrosesan harus diklasifikasi lagi apakah proses tersebut bisa memberi nilai tambah, efisien dan jika mengeliminasi penyebab variabilitasnya atau dengan meningkatkan kepastian hasil proses dapat memberi nilai tambah bagi pengguna.

- *Paperwork*, paperwork dan formulir yang berlebihan cenderung menghasilkan ketidakefisienan proses karena menambah birokrasi dan biasanya hanya sedikit kontribusinya yang secara aktual akan bermanfaat bagi pengguna.
- Duplikasi tugas, setiap tugas yang dilakukan harus memberikan nilai tambah dengan cara-cara tertentu. Jika sebuah tugas diulang, ini tidak menambah nilai, tetapi hanya menambah biaya. Bertambahnya *papaerwork* dan pemasukan data ke dalam sistem komputer sering ditemukan berulang-ulang terjadi dalam kebanyakan perusahaan. Akibat duplikasi tugas ini adalah timbulnya kemungkinan kesalahan dan ketidaksesuaian antara pengerjaan pertama dan pengerjaan selanjutnya.

2 Simplify

Proses penyederhanaan aktifitas proses bisnis yang terjadi di beberapa tempat atau yang diselenggarakan oleh beberapa bagian disesuaikan dengan sumber daya manusia atau alat yang digunakan. Contoh-contoh kegiatan yang sering ada dan cenderung tidak bernilai tambah sehingga mempunyai potensi untuk disederhanakan.

- Prosedur, prosedur-prosedur yang ada biasanya terlalu rumit dan sulit dipahami.
- Komunikasi, baik dengan pelanggan maupun antar karyawan harus jelas dan dapat dipahami oleh semua pihak. 'Bahasa' yang digunakan harus jelas dan sederhana.
- Teknologi, teknologi yang diterapkan perlu diperhatikan kesesuaiannya dengan tugas yang sedang dilaksanakan, solusi teknologi tinggi untuk mengatasi tugas yang tidak dapat diatasi oleh teknologi rendah.
- Aliran, urutan tugas dapat diubah untuk menyederhanakan aliran material ataupun *papaerwork* dan membuat pekerjaan berikutnya lebih mudah.
- Proses, dapat juga disederhanakan dan dirampingkan dengan mengetahui kapan proses tersebut melayani produk atau pasar berbeda. Dengan memecah proses dan mengidentifikasi kegiatan yang paling tepat ditujukan bagi segmen pelanggan tertentu, proses tersebut dapat dibuat lebih sederhana. Kadangkala proses yang sama mencoba memuaskan

pelanggan dengan kebutuhan yang cukup berbeda. Proses itu tidak cukup memadai untuk melayani segmen-segmen yang berbeda tersebut dan yang sering terjadi adalah penekanan pada salah satu segmen tertentu saja.

3 *Integreted*

Proses menggabungkan aktifitas proses bisnis yang serupa, biasanya tugas yang disederhanakan harus diintegrasikan agar dapat menghasilkan aliran yang lancar dalam penyampaian kebutuhan pelanggan dan tugas pelayanan pelanggan. Contoh-contoh kegiatan yang sering ada dan cenderung tidak bernilai tambah sehingga mempunyai potensi untuk diintegrasikan :

- Pekerjaan, dimungkinkan untuk menggabungkan pekerjaan menjadi satu. Melalui pemberdayaan seorang pekerja untuk menyelesaikan rangkaian tugas yang telah disederhanakan sehingga aliran material atau informasi dalam organisasi akan menjadi lebih cepat.
- Tim, perluasan logis dari tugas-tugas yang disatukan adalah penggabungan para ahli ke dalam tim-tim, dimana tidak mungkin bagi seseorang secara sendiri dapat melakukan seluruh rangkaian kegiatan. Diusahakan agar suatu tim ditempatkan bersama-sama saling berdekatan secara fisik untuk meminimalkan jarak yang harus ditempuh material, informasi dan *paperwork* serta meningkatkan komunikasi antar setiap orang yang bekerja dalam proses tersebut.

4 *Automate*

Teknologi informasi dapat menjadi alat yang kuat untuk mempercepat proses dan memberikan layanan pelanggan yang lebih bermutu jika diterapkan pada proses yang tepat/logis. Jika proses tersebut bermasalah maka otomatis akan dapat memperparah situasi. Oleh karena itu otomatis diterapkan setelah mengeliminasi, setelah tahapan otomatisasi, dimungkinkan untuk kembali pada tahap yaitu pengeliminasian, penyederhanaan, dan pengintegrasian tugas-tugas. Beberapa kondisi proses yang dapat dipertimbangkan untuk diotomatisasi adalah sebagai berikut :

- Tugas yang berulang merupakan calon yang paling baik untuk diotomatisasi. Tugas-tugas ini dapat berupa tugas *shop floor*, tugas-tugas

klerikal seperti tugas mencocokkan item-item dalam formulir dan sebagainya.

- Pengumpulan data jika dilakukan dengan mesin, waktu proses lebih cepat dan akurasi akurat. Contoh teknologi ini adalah *barcode reader* di toko-toko grosir.
- Transfer data, mentransfer data dari satu format ke format yang lain, dari satu orang ke orang lain atau satu sistem ke sistem lain, jika memang harus dilakukan atau tidak dapat dihilangkan merupakan calon utama yang lain diotomatisasi.

2.9 Analisa Fit/Gap

2.9.1 Definisi Analisa *fit/gap*

Menurut Hoffman dan Bateson (2006) Analisa *fit/gap* adalah suatu alat yang digunakan untuk mengetahui mengenai kondisi aktual yang sedang berjalan di perusahaan tersebut, untuk kemudian diperbandingkan dengan sumber daya perusahaan tersebut. Hal ini dilakukan agar dapat mengetahui apakah suatu perusahaan sudah bergerak di proses bisnisnya secara optimal untuk memaksimalkan kinerja perusahaan tersebut [HDB-06].

Dalam penggunaan Analisis *fit/gap* dengan *service quality*, menurut Hoffman dan Bateson (2006) bahwa terdapat 5 *quality perspective* dari *service quality* yaitu [HDB-06]:

- 1 *Service Gap*, yaitu mengindikasikan bahwa adanya perbedaan antara pengharapan antara keinginan yang diinginkan oleh pelanggan dengan keadaan yang telah mereka terima sekarang.
- 2 *Knowledge Gap*, yaitu pengharapan yang diinginkan oleh pelanggan dan pengharapan yang diinginkan oleh manajemen perusahaan.
- 3 *Standard Gap*, adalah terjadinya ketimpangan antara persepsi manajemen perusahaan dengan pelanggan, yang dimaksudkan disini adalah standar dari *delivery* standar.
- 4 *Delivery Gap*, adalah terjadinya persepsi yang diinginkan perusahaan kepada pelanggan dengan keadaan yang telah terjadi sebenarnya di perusahaan tersebut.

- 5 *Communication Gap*, adalah terjadinya antara kesenjangan pelanggan dengan komunikasi yang terdapat atau yang dimiliki oleh perusahaan tersebut, dalam hal ini adalah mengantarkan informasi akurat, tepat dan jelas kepada pelanggan mengenai produk atau jasa yang ditawarkan.

2.9.2 Tujuan Analisa *Fit/Gap*

Tujuan dari Analisis *Fit/Gap* adalah mengidentifikasi gap antara alokasi optimal dan integrasi dari input, dan tingkat alokasi pada saat ini. Ini membantu perusahaan dalam menyediakan pemahaman mengenai area-area yang dapat ditingkatkan. Analisis *Fit/Gap* merupakan pembelajaran formal mengenai apa yang dilakukan oleh bisnis dan kemana kita akan berada pada masa yang akan datang. Analisis *Fit/Gap* dapat dilakukan dalam beberapa perspektif antara lain [HDB-06] :

- 1 Organisasi (sebagai contoh : Sumber Daya)
- 2 Tujuan bisnis
- 3 Proses bisnis
- 4 Teknologi Informasi

Analisa *Fit/Gap* menyediakan dasar untuk mengukur investasi dari waktu, biaya, dan sumber daya yang dibutuhkan untuk mencapai hasil yang diharapkan.

2.9.3 Ranking Requirements

Tahapan ini mendukung tim proyek dan sponsor proyek untuk memastikan proses bisnis dapat diakomodasi selama implementasi sistem yang baru. Selain itu, berfungsi untuk memastikan tim proyek berfokus pada area yang paling penting bagi organisasi agar *functionality* yang baru dapat memberikan nilai tambah bagi perusahaan dalam meningkatkan proses bisnis. *Requirement* harus diidentifikasi sesuai dengan tingkat prioritasnya. Adapun tingkat prioritasnya akan dijelaskan sebagai berikut [NAE-10]:

- *High Critical Requirement* : merupakan requirement yang sangat penting untuk kegiatan operasi dan tanpa *requirement* tersebut perusahaan tidak dapat berfungsi, termasuk di dalamnya kebutuhan akan pelaporan internal dan eksternal yang penting.

- *Medium Critical Requirement* : merupakan requirement dimana ketika dipenuhi akan meningkatkan proses bisnis perusahaan.
- *Low Critical Requirement* : merupakan requirement yang hanya menambah nilai yang kecil / minor *value* bagi proses bisnis perusahaan apabila *requirement* tersebut dipenuhi. Adapun requirement tersebut akan dikelompokkan berdasarkan kategori, yaitu :
 - Operasional : *Requirement* pada kategori operasional merupakan *requirement* yang bersifat sebagai peningkatan *produktivitas* karyawan seperti efisiensi waktu, dan penyempurnaan operasional.
 - Strategis : *Requirement* pada kategori strategis merupakan *requirement* yang bersifat sebagai alat pendukung pengambilan keputusan bagi pihak manajemen.

2.9.4 Degree of Fit

Tahap selanjutnya tahap analisis adalah menentukan tingkat kesesuaian diantara kebutuhan pengguna dan perangkat lunak. Berikut ini akan diuraikan kode-kode yang digunakan dalam menentukan tingkat kesesuaian untuk analisis *fit/gap* [NAE-10].

- *Fit* : aktifitas saat ini sudah baik, sehingga tidak ada alternatif untuk perbaikan.
- *Gap* : aktifitas saat ini tidak efektif dan tidak efisiensi sama sekali.
- *Partial* : aktifitas proses saat ini cukup baik akan tetapi diperlukan alternatif agar dapat meningkatkan efektifitas dan efisiensi aktifitas tersebut.

2.10 Pengujian Perangkat Lunak

Sasaran pengujian adalah penemuan semaksimal mungkin kesalahan dengan usaha yang dapat dikelola pada rentang waktu realistik. Pengujian perangkat lunak merupakan tahap kritis dalam penjaminan kualitas perangkat lunak dan merupakan *review* menyeluruh terhadap spesifikasi perancangan dan pengkodean [HRB – 04].

2.10.1 Black-box Testing

Konsep Black-box testing digunakan untuk mempresentasikan sistem yang cara kerja di dalamnya tidak tersedia untuk diinspeksi. Di dalam kotak hitam, item-item yang diuji dianggap gelap karena logiknya tidak diketahui, yang diketahui hanya apa yang masuk dan apa yang keluar. Pada black-box testing, kasus-kasus pengujian berdasarkan pada spesifikasi sistem. Pada black-box testing, dicobakan beragam masukan dan memeriksa keluaran yang dihasilkan. Teknik black-box testing juga dapat digunakan untuk pengujian berbasis skenario di mana isi di dalam sistem tidak tersedia untuk diinspeksi tapi masukan dan keluaran yang didefinisikan dengan *usecase* dan informasi analisis yang lain [HRB – 04].

Black-box testing berusaha menemukan kesalahan dalam kategori sebagai berikut [PRE – 10]:

1. Fungsi – fungsi yang tidak benar atau hilang
2. Kesalahan antarmuka
3. Kesalahan dalam struktur data atau akses *database*
4. Kesalahan kinerja