BAB II

TINJAUAN PUSTAKA

2.1 Sistem Informasi

Sistem informasi adalah sistem komputerisasi yang mengolah data (fakta) dan menghasilkan informasi (Shim 2000). Siklus pengolahan informasi terdiri dari empat operasi yaitu input, proses, output, dan penyimpanan. Informasi memiliki nilai dan berguna karena dapat membantu pengambil keputusan.

Pembangunan sistem informasi dapat menjadi tugas yang rumit dan mencakup banyak pegawai untuk bekerja sama. Dalam proses ini, banyak sistem informasi kecil yang sudah ada dikombinasikan menjadi sistem yang besar sehingga semua fungsi bisnis dapat diintegrasikan melalui sistem informasi, atau sistem informasi yang baru. Membangun sistem informasi di dalam kerangka kerja terencana dapat menghasilkan sistem yang lebih baik dan membantu organisasimenghindari kebutuhan menggabungkan sekumpulan sistem informasi yang tidak sesuai. Oleh karena itu, pembangunan sistem informasi membutuhkan pemahaman terhadap proses bisnis, ilmu, kebijakan organisasi, keahlian di bidangnya, visi, dan kemampuan komunikasi yang baik. Menangkap peluang atau memecahkan masalah bisa menjadi dasar dalam pembangunan sistem informasi yang dapat dijelaskan sebagai berikut:

Menangkap peluang. Peluang didefinisikan sebagai kemungkinan peningkatan di dalam bisnis seperti meningkatkan penjulan, operasi yang lebih efisien, atau mendapatkan keuntungan kompetitif. Menangkap peluang

dipertimbangkan menjadi aktifitas proaktif di dalam bisnis, sedangkan pemecahan masalah dipertimbangkan sebagai reaktif.

Memecahkan masalah. Masalah didefinisikan sebagai situasi yang tidak diinginkan atau keadaan yang tidak terduga. Sistem informasi yang baik dapat memberikan solusi terhadap masalah yang dihadapi.

2.2 *Enterprise*

Enterprise adalah sekumpulan korporat atau entitas atau lembaga dengan tugas yang mendukung entitas fungsional dan memiliki sekumpulan tujuan atau suatu perintah (Minoli 2008). Dalam konteks ini, suatu *enterprise* adalah divisi atau department dari suatu perusahaan, sekelompok organisasi yang terpisah secara geografis namun dihubungkan bersama oleh kepemilikan secara administratif, agen pemerintah (atau sekumpulan agen) pada setiap tingkat yuridiksi, sekumpulan agen pemerintah, dan seterusnya (Group 2011).

Organisasi besar dan entitas pemerintah dapat terdiri dari beberapa *enterprise*. Suatu arsitektur mendukung penyatuan operasi multi organisasional. *Framework* arsitektural dapat memberikan dasar pengembangan repositori arsitektur untuk penyatuan dan penggunaan kembali model dan rancangan data dasar.

2.3 Arsitektur

Berdasarkan American National Standards Institute/Institute of Electrical and Electronics Engineers (ANSI/IEEE) Std 1471-2000, suatu arsitektur adalah dasar organisasi suatu sistem, diwujudkan di dalam komponennya, hubungannya terhadap komponen lain dan lingkungannya, dan prinsip mengatur rancangan dan

evolusi sistem tersebut. Arsitektur dapat digambarkan sebagai *blueprint*TI/korporat untuk perencanaan suatu kota atau pembangunan yang besar (Minoli 2008).

Arsitektur mencakup bagian berikut ini:

1. Arsitektur bisnis: suatu perumusan arsitektural dari fungsi bisnis.
2. Arisktektur informasi: suatu perumusan arsitektural dari informasi fungsi melalui model data.
3. Arsitektur sistem/aplikasi: defenisi arsitektural dari sistem/aplikasi fungsi.
4. Arsitektur teknologi infrastruktur : perumusan (penjelasan) arsitektural dari infrastruktur teknologi fungsi.

2.4 *Enterprise Architecture*

Enterprise Architecture (EA) adalah kumpulan proses bisnis, aplikasi, teknologi, dan data yang mendukung strategi bisnis suatu *enterprise* (Minoli 2008). Arsitektur *enterprise* adalah suatu rencana perekaman, cetak biru dari struktur, susunan, pengaturan, pengelompokan fungsional, antar muka, data, protokol, logika fungsional, integrasi, teknologi, dari sumber IT dan dibutuhkan untuk mendukung fungsi bisnis atau misi korporat atau organisasi (Minoli 2008).

Tujuan dari EA adalah membuat lingkungan TI yang bersatu (sistem *software* dan *hardware* yang terstandarisasi) melintasi perusahaan atau semua unit bisnis perusahaan, yang dihubungkan kepada sisi bisnis dan strategi organisasi.Secara spesifik, tujuannya adalah untuk mempertimbangkan penjajaran, standarisasi, penggunaan kembali aset TI, dan berbagi metode yang sudah ada

untuk pengelolaan proyek dan pembangunan *software* antar organisasi. Hasil akhirnya adalah membuat TI lebih murah, strategis, dan responsif.

Maksud dari EA adalah membuat pemetaan aset TI dan proses bisnis dan sekumpulan prinsip tata kelola yang mengendalikan strategi bisnis dan bagaimana hal tersebut dapat dijelaskan melaui TI. Hasil dari EA dapat berupa sekumpulan strategi TI, standar EA yang baru atau dimodifikasi, *roadmap* yang menjelaskan proyek TI yang dibutuhkan untuk mengimplementasikan arisktektur baru dan mencapai target, dan suatu rencana pengembangan/penyebaran.

2.5 TOGAF

TOGAF adalah suatu *framework* (rincian metode dan sekumpulan tools pendukung) untuk mengembangkan arsitektur *enterprise*. (Group 2011)

TOGAF adalah *framework* arsitektur yang memberikan metode dan tools untuk membantu dalam penerimaan, produksi, penggunaan, dan perawatan arsitektur *enterprise*. TOGAF didasarkan pada model proses iteratif yang didukung oleh best practice dan sekumpulan aset arsitektur yang bisa digunakan kembali.

TOGAF mendefenisikan *enterprise* sebagai sekumpulan organisasi yang memiliki sekumpulan tujuan (*goal*) (Group 2011). Sebagai contoh, *enterprise* bisa berupa agen pemerintah, suatu perusahaan, suatu bagian dari perusahaan, sautu *department*, atau organisasi yang terpisah secara geografis namun mempunyai pemilik yang sama.

Istilah *enterprise* di dalam arsitektur *enterprise* dapat digunakan untuk menunjukan keseluruhan *enterprise* – menunjukan seluruh layanan teknologi daninformasi, proses-proses, dan infrastruktur maupun domain tertentu di dalam *enterprise*. Arsitektur melintasi berbagai sistem dan berabagai kelompok fungsional di dalam *enterprise*.

Tujuan dari arsitektur *enterprise* adalah mengoptimasi proses antar *enterprise* yang terpisah (baik manual atau otomatis) menjadi satu lingkungan terintegrasi yang merespon perubahan dan mendukung strategi bisnis. Arsitektur *enterprise* yang baik memungkinkan pencapaian keseimbangan antara efesiensi TI dan inovasi bisnis. Arsitektur *enterprise* memungkinan unit bisnis untuk berinovasi untuk mendapatkan keuntungan strategis.

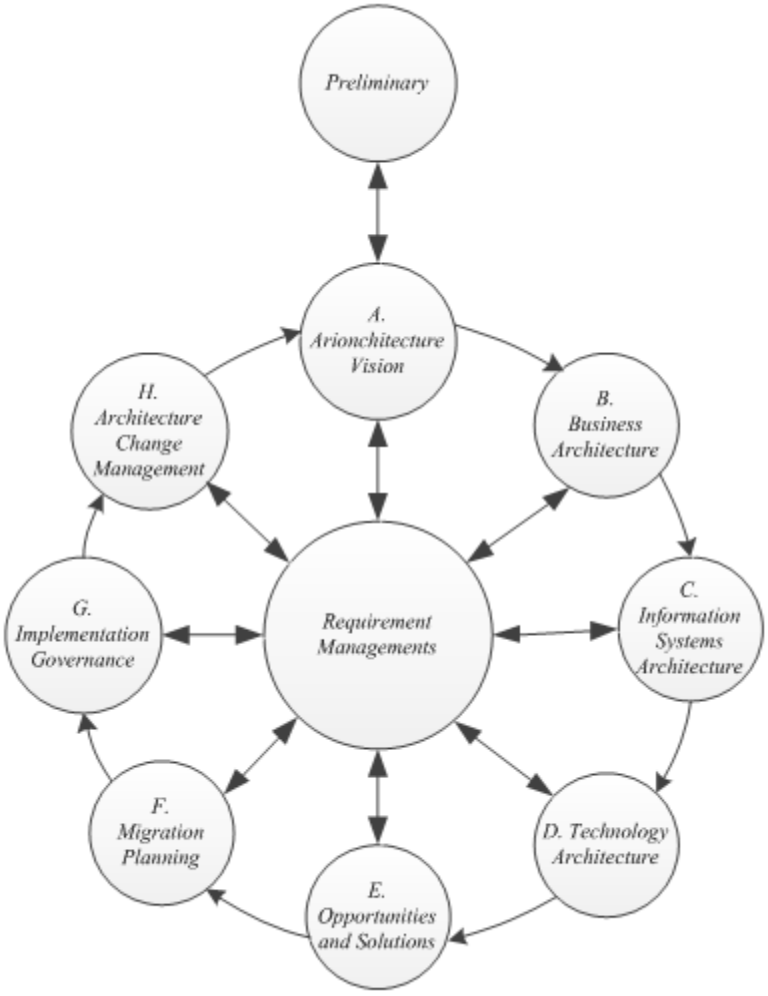
Framework arsitektur merupakan struktur dasar, atau sekumpulan struktur, yang dapat digunakan dalam membangun arsitektur yang berbeda(Group 2011).*Framework* tersebut harus bisa menjelaskan metode perancangan target *enterprise* di dalam sekumpulan bagian dan menunjukan bagaimana bagian-bagian tersebut saling terhubung. *Framework* mengandung sekumpulan *tools* dan memberikan penjelasan mengenai istilah yang digunakan dan sekumpulan rekomendasi standar yang bisa digunakan di dalam bagian tersebut.

TOGAF sudah dikembangkan oleh lebih dari 300 perusahaan anggota forum arsitektur yang berasal dari perusahaan dan organisasi besar. Mengembangkan dan mendukung arsitektur *enterprise* merupakan teknik yang rumit karena mencakup berbagai *stakeholder* dan proses pengambilan keputusan di dalam organisasi. TOGAF memiliki peran penting di dalam standarisasi dan

menjelaskan proses pengembangan arsitektur. TOGAF memberikan praktik kerangka kerja terbaik untuk menambah nilai, dan memungkinkan organisasi untuk membangun solusi yang dapat dilakukan dan ekonomis dan mengakomodasi persoalan dan kebutuhan bisnis.

2.5.1 TOGAF *Architecture Development Method*

TOGAF *Architecture Development Method* (ADM) memberikan proses teruji dan dapat diulang untuk mengembangkan arsitektur (Group 2011). ADM mencakup penetapan *framework* arsitektur, membangun konten arsitektur, peralihan, dan mengatur realisasi arsitektur.



Gambar 2.1. Siklus TOGAF ADM (Group 2011).

Semua aktifitas tersebut ada di dalam suatu siklus iteratif dari defenisi dan realisasi arsitektur yang memungkinkan organisasi mengubah perusahaan menjadi bisa dikontrol yang berhubungan dengan tujuan dan peluang bisnis.

Fase-fase yang ada di dalam ADM adalah sebagai berikut:

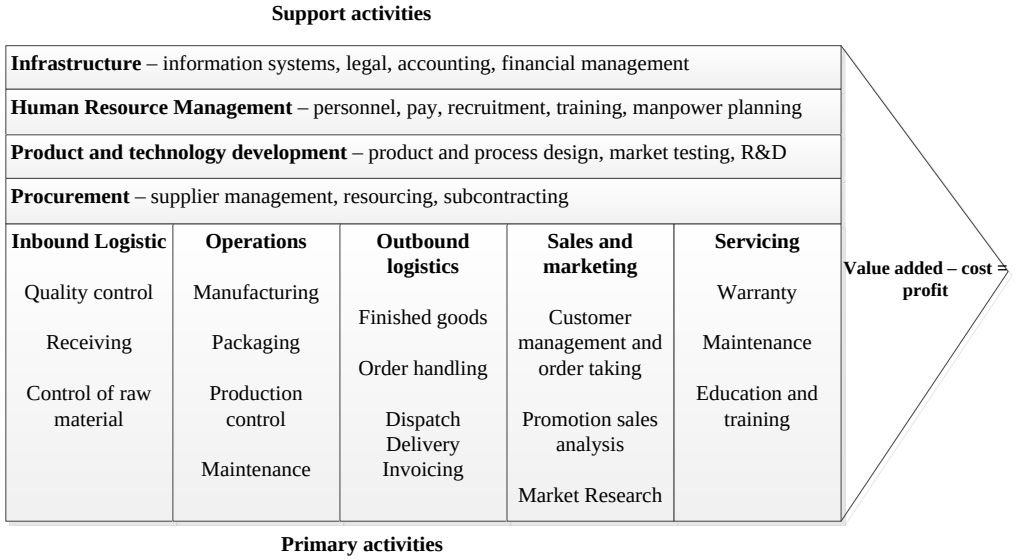
1. Fase *preliminary* menjelaskan aktifitas persiapan dan permulaan yang dibutuhkan untuk membuat *Architecture Capability* termasuk penyesuaian TOGAF dan pendefinisian prinsip arsitektur.
2. Fase A: *Architecture Vision* menjelaskan fase awal dari siklus pengembangan arsitektur yang mencakup informasi mengenai penentuan ruang lingkup pembangunan arsitektur, identifikasi *stakeholder*, pembuatan visi arsitektur, dan memperoleh izin untuk membangun arsitektur.
3. Fase B: *Business Architecture* menjelaskan pembangunan *Business Architecture* untuk mendukung *ArchitectureVision* yang sudah disetujui.
4. Fase C: *Information* System *Architecture* menjelaskan pembangunan arsitektur sistem informasi untuk mendukung *ArchitectureVision* yang sudah disetujui.
5. Fase D: *Technology Architecture* menjelaskan pembangunan arsitektur teknologi untuk mendukung *ArchitectureVision* yang sudah disetujui.
6. Fase E: Peluang dan Solusi berhubungan dengan perencanaan implementasi awal dan identifikasi sarana pengiriman untuk arsitektur yang didefenisikan pada fase sebelumnya.

7. Fase F: *Migration Planning* menjelaskan bagaimana bergerak dari garis awal ke target arsitektur dengan mengakhiri rincian implementasi dan rencana perpindahan.
8. Fase G: *Implementation Governance* memberikan kesalahan implementasi yang arsitektural.
9. Fase H: *Architecture Change Management* menetapkan prosedur untuk mengelola perubahan terhadap arsitektur yang baru.
10. *Requirement management* memeriksa proses pengelolaan kebutuhan arsitektur melalui ADM.

2.6 *Business Value Chain Model*

Value chain model fokus terhadap aktifitas tertentu di dalam bisnis dimana strategi kompetitif dapat diaplikasi dengan baik dan dimana sistem informasi paling banyak memberikan dampak strategis (Kenneth C. Laudon 2006).

Model ini mengidentifikasi poin kritis tertentu dimana perusahaan dapat menggunakan teknologi informasi yang paling efektif untuk mencapai posisi kompetitif. *Value chain model* melihat perusahaan sebagai sekumpulan atau rantai aktifitas dasar yang bisa menambah nilai terhadap produk atau layanan perusahaan. Aktifitas tersebut dapat dibagi menjadi aktifitas utama dan aktifitas pendukung.



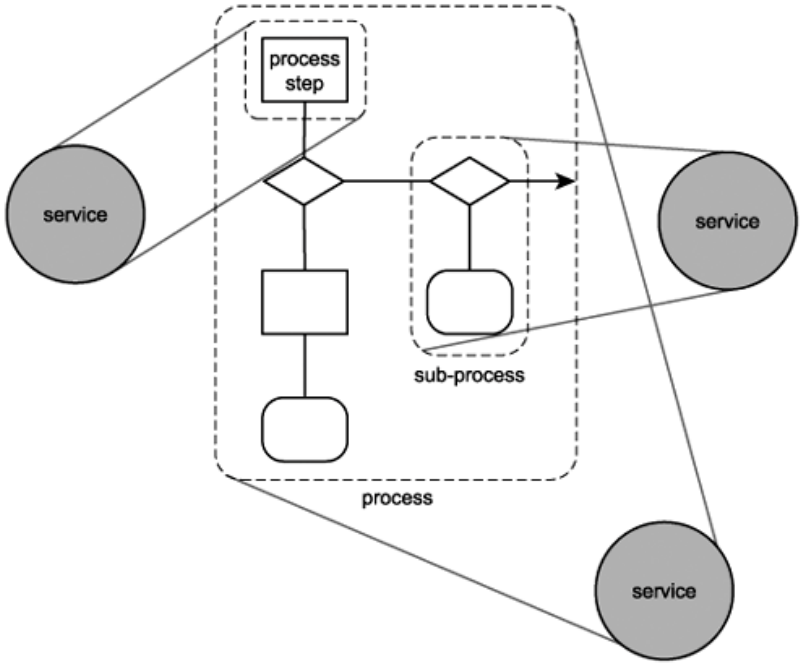
Gambar 2.2.*Value chain* Organisasi (Adrienne Curry 2006).

Aktifitas utama berhubungan langsung terhadap produksi dan distribusi produk dan layanan perusahaan, yang menghasilkan nilai bagi konsumen. Aktifitas utama mencakup *inbound logistic*, operasi, *outbound logistic*, penjualan dan pemasaran, dan layanan.*Inbound logistic* mencakup penerimaan dan penyimpanan bahan untuk didistribusikan ke produksi. Operasi mengubah input menjadi produk akhir. *Outbound logistic* membutuhkan penyimpanan dan pendistribusian produk akhir. Penjualan dan pemasaran mencakup aktifitas promosi dan penjualan produk perusahaan. Aktifitas layanan mencakup perawatan dan perbaikan barang dan jasa perusahaan.

Aktifitas pendukung membuat aktifitas utama dapat dilakukan dan terdiri dari infrastruktur organisasi (administrasi dan manajemen), sumber daya manusia (perekrutan, penggajian, dan pelatihan pegawai), dan *procurement* (input pembelian).

2.7 *Service-Oriented Architecture*(SOA)

Service-oriented architecture adalah istilah yang merepresentasikan model penyusunan logika otomasi menjadi lebih kecil, memperjelas sekumpulan logika(Erl 2005). Jika dikumpulkan, unit-unit tersebut akan membangun bagian logika otomasi bisnis yang lebih besar. *Service-Oriented Architecture* (SOA) mendorong sekumpulan logika individual agar tetap ada secara otonomi dan tidak saling terisolasi.

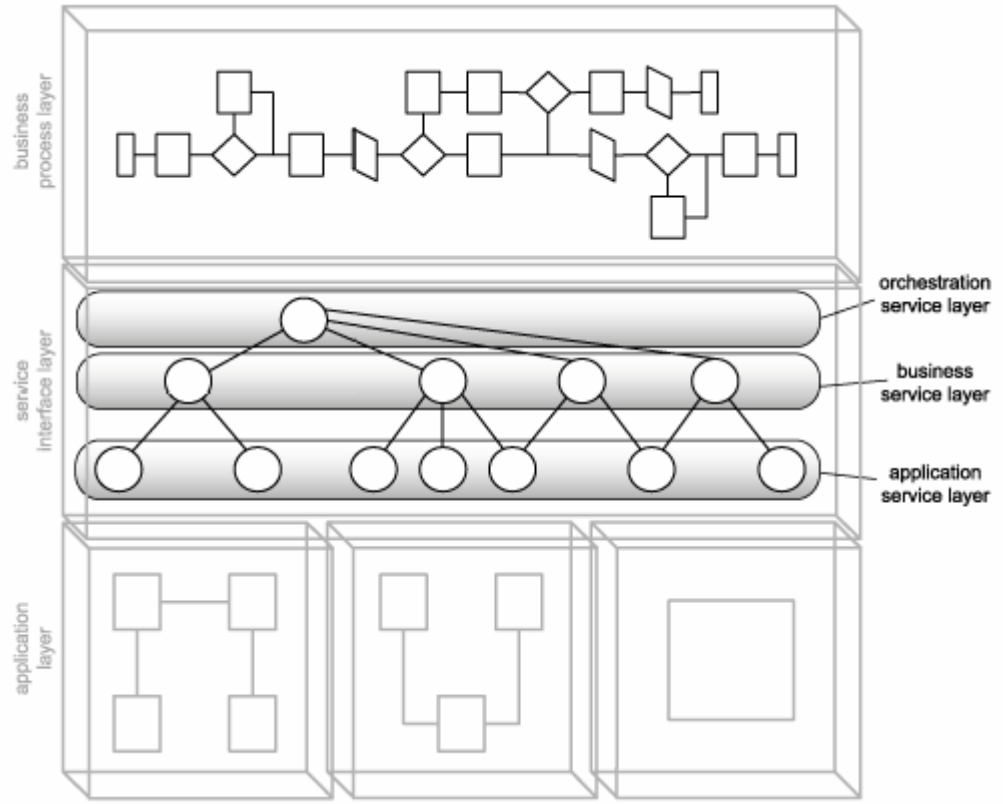


Gambar 2.3. Layanan dapat mengenkapsulasi berbagai logika.

Seperti pada gambar 2.4, ketika membangun solusi otomasi yang terdiri dari *service*, setiap *service* dapat mengenkapsulasi operasi yang dilakukan oleh langkah individu atau subproses yang terdiri dari sekumpulan langkah. *Service* bahkan dapat mengenkapsulasi seluruh logika proses.

2.7.1 Lapisan SOA

Setiap lapisan dapat meringkaskan aspek solusi tertentu, mengakomodasi salah satu masalah yang sudah diidentifikasi (Erl 2005). Hal ini mengurangi beban untuk membangun *service* yang mengakomodasi bisnis, aplikasi, dan agilitas sekaligus.



Gambar 2.4. Tiga lapisan utama layanan.

1. *Application ServiceLayer*

   Lapisan *service* aplikasi merupakan landasan yang menyatakan fungsionalitas teknologi secara spesifik. Tujuannya adalah memberikan fungsi yang dapat digunakan kembali yang berhubungan terhadap pengolahan data di dalam lingkungan aplikasi yang baru atau sudah ada.

2. *Business servicelayer*

   Jika layanan aplikasi bertanggung jawab terhadap representasi teknologi dan logika aplikasi, lapisan *service* bisnis memperkenalkan suatu *service* yang hanya terkait dengan representasi logika bisnis. Lapisan ini bertanggung jawab untuk menyatakan logika bisnis melalui orientasi *service* dan membawa representasi model bisnis korporat ke dalam wilayah *web service*. Lapisan ini mengacu pada pembuatan dua model layanan bisnis yaitu *task-centric* dan *entity-centric*. *Task-centric* adalah suatu *service* yang mengenkapsulasi logika bisnis tertentu menjadi tugas atau proses bisnis. *Entity-centric* adalah suatu layanan yang mengenkapsulasi entitas bisnis tertentu.

3. *Orchestration service layer*

   Orkestrasi lebih bernilai dari pada proses bisnis standar karena memungkinkan menghubungkan logika proses ke interaksi *service* di dala logika *workflow*. Lapisan *service* orkestrasi memperkenalkan tingkatan induk yang mengurangi kebutuhan *service* lain untuk mengelola rincian interaksi yang dibutuhkan untuk menjamin bahwa operasi layanan dieksekui dalam urutan tertentu.

2.8 *Unified Modeling Language* (UML)

Unified Modeling Language (UML) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi,menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis *OO*(*Object-Oriented*). UML adalah bahasa standar yang diterima dengan luas

untuk spesifikasi dan rancangan sistem software yang menggunakan pendekatan object-oriented (Laplante 2007).

UML adalah bahasa pemodelan terstandarisasi untuk tujuan secara umum di bidang *software* engineering. UML mencakup sekumpulan teknis penggambaran notasi untuk membuat model abstrak dari suatu sistem (Roshen 2009). UML memungkinkan pengguna untuk memodelkan proses bisnis, struktur aplikasi, *application behavior*, struktur data, dan arsitektur. Oleh karena itu UML dapat dimanfaatkan untuk tiga lapisan atas arsitektur.

UML mendefinisikan 12 tipe diagram, yang dibagi menjadi tiga kategori: empat diagram yang merepresentasikan struktur aplikasi statis, lima diagram merepresentasikan aspek *dynamic behavior*, dan tiga diagram merepresentasikan cara mengorganisasi dan mengelola modul aplikasi.

2.8.1 *Use case*

Use case adalah penjelasan statis bagaimana sistem atau bisnis digunakan, oleh konsumen, pengguna, atau oleh sistem lain(O'Docherty 2005). Diagram *use case* memperlihatkan bagiamana keterhubungan sistem dan bagaiamana pengguna berhubungan dengan sistem tersebut. Setiap lingkaran elips pada *use case* merepresentasikan*use case* dan lambang orang merepresentasikan pengguna.

Use case menggunakan partisipan yang disebut dengan aktor. Setiap *use case* akan memberikan nilai bagi aktor. Nilai dapat berarti hal yang berbeda bagi orang yang berbeda. Nilai tersebut bisa berupa beberapa informasi yang ingin diterima oleh aktor, beberapa efek yang diharapkan oleh aktor karena terhubung

dengan sistem, beberapa uang, pembelian, atau hal lain yang memotifasi aktor. *Use case* membantu dalam menemukan objek, atribut, dan operasi.