

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Kajian Penelitian Sejenis**

Sebagai referensi dalam penelitian ini, penulis mengkaji beberapa penelitian sejenis yang terdapat di perpustakaan STIKI dan beberapa literatur lainnya. Penulis menemukan beberapa penelitian sejenis diantaranya sebagai berikut:

##### **2.1.1 Kajian Penelitian Woldy (2013).**

Penelitian yang dilaksanakan oleh Woldy ini mengambil objek penelitian di Sekolah Tinggi Informatika Dan Komputer Indonesia. Sistem informasi tersebut bertujuan untuk memangkas beberapa proses sehingga memudahkan dan lebih efisien dalam segala hal yang berkaitan dengan workshop dan seminar seperti pengajuan acara ke lembaga, proses penjaringan peserta, pendaftaran peserta, pembayaran online, proses administrasi pelaksanaan hingga pencetakan sertifikat.

Kelemahan dari penelitian tersebut adalah Gravitas Woldy belum mengkaji instrumen yang sesuai dengan permendikbud nomor 81 tahun 2014 sehingga belum bisa dijadikan patokan tugas khusus di STIKI Malang.

### **2.1.2 Kajian Penelitian El Khoiro (2015).**

Wida Afif El Khoiro melakukan penelitian dengan objek penelitian Sekolah Tinggi Informatika dan Komputer Indonesia. Pengembangan sistem informasi seminar bertujuan memberikan kemudahan perihal pengelolaan kegiatan seminar dan workshop secara kepada semua civitas akademik terkait.

Kelemahan yang terdapat pada sistem tersebut adalah informasi yang diberikan jumlah kuota tidak berdasarkan ruangan yang ada dan belum ada parameter penilaian sebuah acara.

### **2.2 Gambaran Umum Perusahaan/ Instansi**

Sekolah Tinggi Informatika & Komputer Indonesia (STIKI) Malang merupakan lembaga pendidikan tinggi yang memfokuskan diri pada bidang informatika. STIKI Malang merupakan salah satu perguruan tinggi di Kota Malang, yang mencetak lulusan Sarjana dan Ahli Madya di bidang Informatika. STIKI Malang berdiri sejak tahun 1985 dan saat ini memiliki 2 program studi, yaitu Teknik Informatika (S1) dan Manajemen Informatika (D3) dan program setara Diploma-1 dengan 3 spesialisasi Desain grafis dan multimedia, Teknisi komputer dan jaringan, Digital marketing dan web. Saat ini, program studi yang ada masing-masing telah terakreditasi oleh BAN-PT dengan SK Akreditasi No. 025/BAN-PT/Ak-XIII/S1/XI/2010 tanggal 12

November 2010 untuk program studi Teknik Informatika dan SK Akreditasi No. 011/BAN-PT/Ak-XII/Dpl-III/VI/2012 tanggal 15 Juni 2012 untuk program studi Manajemen Informatika. Selain itu masing-masing program studi tercatat aktif memiliki ijin penyelenggaraan program studi, yang merupakan syarat utama penyelenggaraan perguruan tinggi. Mulai tahun 2014 STIKI telah menambah program studi baru yaitu program studi Desain Komunikasi Visual (DKV) dengan SK Mendikbud RI No. 381/E/O/2013 tentang Izin Penyelenggaraan Program Studi Desai Komunikasi Visual (S-1) pada Sekolah Tinggi Informatika & Komputer Indonesia (STIKI) di Malang yang Diseleggarakan oleh Yayasan Perguruan Tinggi Teknik Nusantara di Malang.

Sejalan dengan Visi Pendidikan Nasional tahun 2025 yaitu : INSAN INDONESIA CERDAS & KOMPETITIF serta adanya UU Sisdiknas tahun 2003, STIKI sejak tahun 2009 telah berbenah diri dengan memperbaharui sistem tata kelola, pengembangan kemahasiswaan, pengembangan dosen, pemberdayaan alumni serta menjalin kerjasama dengan berbagai pihak sejalan dengan Visi STIKI yaitu ‘Menjadi satu dari sepuluh perguruan tinggi ICT berkualitas di Indonesia, dan diakui dunia Internasional’

Di dalam Era Globalisasi saat ini, transformasi pendidikan yang menjadi motor penggerak dari masyarakat secara luas, telah bergerak sangat cepat sehingga tidak saja menggerakkan masyarakat berkembang menjadi masyarakat maju, tetapi juga menjadi masyarakat berbasis pengetahuan. Peranan ilmu pengetahuan dan teknologi bagi masyarakat berbasis

pengetahuan sangatlah dominan. Pemanfaatan Iptek didalam kehidupan masyarakat serta adanya transfer teknologi, akan mendorong penguasaan teknologi masyarakat Indonesia secara optimal.

Sebagai salah satu perguruan tinggi informatika di Indonesia, STIKI memiliki peran penting untuk mendorong terciptanya masyarakat berbasis pengetahuan, dengan memanfaatkan Iptek didalam seluruh aspek kehidupan masyarakat. Peranan mahasiswa, alumni, dosen untuk menghasilkan karya ICT berkualitas sangatlah diharapkan untuk meningkatkan jumlah aset intelektual bangsa Indonesia. Selain itu kegiatan pengabdian masyarakat yang berkualitas di bidang ICT merupakan perwujudan kepedulian civitas akademika didalam memampukan masyarakat Indonesia dalam penguasaan ICT, sehingga menjadi masyarakat berbasis pengetahuan.

### **2.2.1 Visi, Misi, Tata Nilai STIKI**

#### **Visi:**

Menjadi satu dari sepuluh Perguruan Tinggi ICT berkualitas di Indonesia dan diakui dunia Internasional.

#### **Misi :**

1. Menyelenggarakan pendidikan, pengabdian serta penelitian yang berkualitas serta mampu bekerjasama dengan lembaga terkemuka lain untuk menjawab tantangan serta perkembangan lokal dan global melalui pendekatan ICT.

2. Menghasilkan lulusan profesional yang menjunjung tinggi prinsip dan nilai ELANG, budaya Indonesia serta memiliki jiwa leadership dan entrepreneurship.

**Tujuan Stiki :**

1. Menjadi perguruan tinggi ICT berkualitas dengan tata kelola yang baik (good governance).
2. Mewujudkan jejaring kerjasama dengan lembaga terkemuka nasional maupun internasional.
3. Menjadi pusat inovasi dan Inkubator di bidang ICT yang memiliki prinsip & nilai ELANG, budaya serta mampu menumbuhkan jiwa leadership dan entrepreneurship.

**Prinsip & Nilai:**

1. Expert
2. Loyal
3. Active
4. Nationalist
5. Gentle

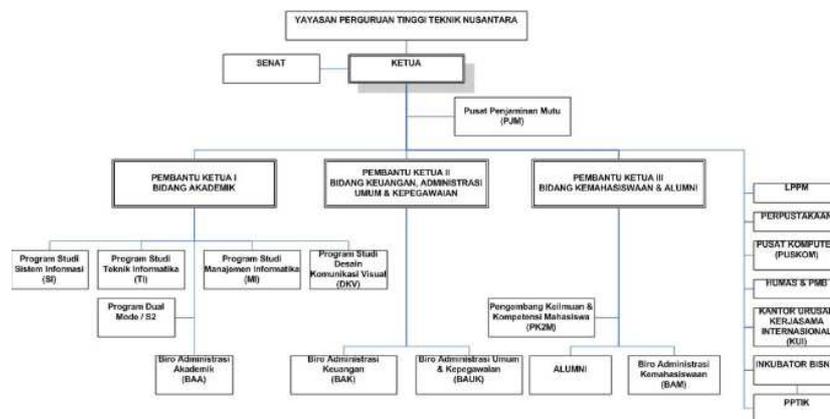
## Karakteristik Lulusan

1. Menguasai ICT
2. Memiliki Jiwa Leadership
3. Memiliki Jiwa Entrepreneurship

## 2.2.2 Struktur Organisasi Instansi

I. STRUKTUR ORGANISASI STIKI PERIODE 2013-2017  
 Dalam rangka untuk mencapai tujuan yang telah dicanangkan dan sebagai piranti operasionalisasi manajemen institusi disusun struktur organisasi sebagai berikut :

### A. BAGAN STRUKTUR ORGANISASI STIKI



Gambar Struktur Organisasi Sekolah Tinggi Informatika & Komputer Indonesia (STIKI)

**Gambar 2.1** Bagan Struktur Organisasi STIKI

**NAMA – NAMA PEJABAT STRUKTURAL**

<b>JABATAN</b>	<b>NAMA PEJABAT</b>
<b>Ketua</b>	Dr. Eva Handriyantini, S.Kom, M.MT
<b>Pembantu Ketua I Bid. Akademik</b>	Evy Poerbaningtyas, S.Si, M.T
- Ketua Program Studi Teknik Informatika (TI)	Daniel Rudiaman S., S.T, M.Kom
- Ketua Program Studi Manajemen Informatika (MI)	Anita, S.Kom, M.T
- Ketua Program Studi Desain Komunikasi Visual (DKV)	Saiful Yahya, S.Sn, M.T.
- Ketua Program Studi Sistem Informasi (SI)	Anita, S.Kom, M.T.
- Koordinator Program S2	Anita, S.Kom, M.T
- Kepala Biro Administrasi Akademik (BAA)	Erni Dwi Wahyuni, S.ST.
<b>Pembantu Ketua II Bid. Keuangan, Administrasi Umum &amp; Kepegawaian</b>	Laila Isyriyah, M.Kom
- Kepala Biro Administrasi Keuangan (BAK)	Ely Sulistyorini, S.E
- Kepala Biro Administrasi Umum & Kepegawaian (BAUK)	Diah Arifah P., S.Kom, M.T
<b>Pembantu Ketua III Bid. Kemahasiswaan &amp; Alumni</b>	Sugeng Widodo, M.Kom
- Kepala Pengembangan Keilmuan & Kompetensi Mahasiswa (PK2M)	Dedy Ari P., S.Kom
- Koordinator Alumni	Dedy Ari P., S.Kom
- Koordinator Biro Administrasi Kemahasiswaan (BAM)	U'un Setiawati
<b>PUSAT PENJAMINAN MUTU (PJM)</b>	
Kepala Pusat Penjaminan Mutu	Diah Arifah P., S.Kom, M.T
<b>UNIT PENUNJANG (UPT)</b>	
- Koordinator LPPM	Subari, M.Kom
- Kepala Perpustakaan	Endah Wulandari, S.E.
- Kepala Pusat Komputer (PUSKOM)	Koko Wahyu Prasetyo, S.Kom, M.TI.
- Koordinator PMB	Septa Noviana Yudaningtyas, S.Kom
- Koordinator Humas	Setiabudi Sakaria, M.Kom
Kepala Kantor Urusan Internasional (KUI)	Dr. Eva Handriyantini, S.Kom, M.MT
Kepala Inkubator Bisnis	Laila Isyriyah, M.Kom
Kepala PPTIK	Meivi Kartikasari, S.Kom

**Gambar 2.2**Daftar Pejabat Struktural

## 2.3 Sistem Informasi

### 2.3.1 Pengertian Sistem

Definisi sistem menurut Jerry Fitz Gerald, Andra F. Fitz Gerald, Warren D. Stallings, Jr. (1973), “Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu”

Menurut Sutabri (2012), “Sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen atau variabel-variabel yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu.”

Menurut West Churchman (1968), “Sebuah sistem dapat didefinisikan sebagai serangkaian komponen yang dikoordinasikan untuk mencapai serangkaian tujuan”

Berdasarkan penjelasan di atas penulis dapat menyimpulkan, bahwa sistem adalah gabungan dari dua atau lebih komponen dari sifat-sifat tertentu yang saling berhubungan dan berinteraksi antara satu dengan yang lainnya sehingga membentuk satu kesatuan untuk mencapai suatu tujuan tertentu.

## **2.3.2 Pengertian Seminar dan Workshop**

### **2.3.2.1 Pengertian Seminar**

Seminar pada umumnya merupakan sebuah bentuk pengajaran akademis, baik di sebuah universitas maupun diberikan oleh suatu organisasi komersial atau profesional. Kata *seminar* berasal dari kata Latin *seminarum*, yang berarti "tanah tempat menanam benih".

Sebuah seminar biasanya memiliki fokus pada suatu topik yang khusus, di mana peserta yang hadir dapat berpartisipasi secara aktif. Seminar seringkali dilaksanakan melalui sebuah dialog dengan seorang moderator seminar, atau melalui sebuah presentasi hasil penelitian dalam bentuk yang lebih formal. Biasanya, para peserta bukanlah seorang pemula dalam topik yang didiskusikan (di universitas, kelas-kelas seminar biasanya disediakan untuk mahasiswa yang telah mencapai tingkatan atas). Sistem seminar memiliki gagasan untuk lebih mendekatkan mahasiswa kepada topik yang dibicarakan. Di beberapa seminar dilakukan juga pertanyaan dan debat. Seminar memiliki sifat lebih informal dibandingkan sistem kuliah di kelas dalam sebuah pengajaran akademis.

Perlu dicatat bahwa di beberapa universitas Eropa, sebuah seminar dapat berarti kelas kuliah yang besar, khususnya ketika dibawakan oleh ahli yang termasyhur (tanpa memperhatikan jumlah hadirin atau jangkauan mahasiswa yang berpartisipasi dalam diskusi).

### **2.3.2.2 Pengertian Lokakarya (workshop)**

Lokakarya adalah program pendidikan dan pelatihan yang padat dan singkat. Pemimpin lokakarya memberi tugas kepada peserta yang harus dikerjakan pada waktu itu juga. Kegiatan lokakarya identik dengan seminar yaitu suatu pertemuan ilmiah untuk membahas masalah tertentu oleh para pakar dalam bidang tertentu pula

Perbedaan mendasar antara lokakarya dengan seminar hanya menekankan pada hasil yang didapat dari lokakarya menjadi sebuah produk yang dapat digunakan peserta lokakarya dalam proses pembelajaran di kelas. Sedangkan seperti seminar kali ini adalah hanya sebagai pencetus ide yang jika tepat dapat ditindak lanjuti dan jika tidak dapat digunakan bahan pemikiran dan acuan berfikir bagi kalangan pendidik di masa yang akan datang. Karna ada kalanya suatu pemikiran yang baik membutuhkan momen yang tepat bagi pelaksanaannya. Hal tersebut tergantung pada permasalahan yang ditimbulkan oleh pemikiran tersebut.

### **2.3.2.3 Ciri – Ciri Lokakarya**

- a. Masalah yang dibahas bersifat life centered dan muncul dari peserta sendiri
- b. Cara yang digunakan ialah metode pemecahan masalah musyawarah dan penyelidikan

- c. Menggunakan resource person dan resource materials yang memberbantuan yang besar sekali dalam mencapai hasil yang sebaik-baiknya.

#### **2.3.2.4Prosedur Pelaksanaan Lokakarya**

- a. Merumuskan tujuan workshop (output yang akan dicapai).
- b. Merumuskan pokok-pokok masalah yang akan dibahas secara terperinci.
- c. Menentukan prosedur pemecahan masalah.

#### **2.3.2.5Kelemahan dan Kelebihan Lokakarya**

##### **a. Kelebihan Lokakarya**

- 1) Peserta mendapatkan keterangan teoritis yang luas dan mendalam tentang masalah yang dibahas.
- 2) Peserta mendapatkan petunjuk-petunjuk praktis untuk melaksanakan tugasnya.
- 3) Peserta dibina untuk bersikap dan berfikir secara ilmiah, Terpupuknya kerja sama antar peserta, Terhubungnya lembaga pendidikan dan masyarakat.

##### **b. Kelemahan Lokakarya**

- 1) Memerlukan persiapan yang relatif lama.
- 2) Memerlukan tenaga dan biaya yang besar.

- 3) Melibatkan banyak orang sehingga menyita waktu guru untuk melaksanakan pembelajaran di kelasnya.
- 4) Menimbulkan banyak pro dan kontra sehingga menimbulkan potensi konflik di antara pengamat pendidikan dan pelaksana kebijaksanaan.

## **2.4 Tugas Khusus Di STIKI Malang**

Seminar dan workshop merupakan salah satu dari beberapa kegiatan yang menjadi prasyarat tugas khusus di STIKI. Sedangkan tugas khusus itu sendiri merupakan salah satu mata kuliah wajib yang harus ditempuh sebagai syarat kelulusan. Berdasarkan peraturan tugas khusus terbaru tahun 2014 yang telah ditetapkan oleh bidang akademik berikut penjelasan terkait tugas khusus.

### **2.4.1 Definisi Tugas Khusus**

Tugas Khusus merupakan mata kuliah mandiri yang bertujuan untuk meningkatkan kemampuan *hard skill* dan *soft skill* yang dinilai berdasarkan aturan yang telah ditetapkan.

Pada mata kuliah ini mahasiswa harus mengumpulkan point yang didapatkan dari mengikuti kegiatan/kompetisi ilmiah yang diselenggarakan didalam maupun diluar kampus, baik dalam peran serta pemakalah, peserta maupun penulis karya ilmiah.

## **2.5 Gambaran Umum Android**

### **2.5.1 Sejarah Android**

Android (/ˈæn.drɔɪd/; AN-droyd) adalah sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet. Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya Open Handset Alliance, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler. Ponsel Android pertama mulai dijual pada bulan Oktober 2008.

Antarmuka pengguna Android umumnya berupa manipulasi langsung, menggunakan gerakan sentuh yang serupa dengan tindakan nyata, misalnya menggeser, mengetuk, dan mencubit untuk memanipulasi objek di layar, serta papan ketik virtual untuk menulis teks. Selain perangkat layar sentuh, Google juga telah mengembangkan Android TV untuk televisi, Android Auto untuk mobil, dan Android Wear untuk jam tangan, masing-masingnya memiliki antarmuka pengguna yang berbeda. Varian Android juga digunakan pada komputer jinjing, konsol permainan, kamera digital, dan peralatan elektronik lainnya.

Android adalah sistem operasi dengan sumber terbuka, dan Google merilis kodenya di bawah Lisensi Apache. Kode dengan sumber terbuka dan lisensi perizinan pada Android memungkinkan perangkat lunak untuk dimodifikasi secara bebas dan didistribusikan oleh para pembuat perangkat, operator nirkabel, dan pengembang aplikasi. Selain itu, Android memiliki sejumlah besar komunitas pengembang aplikasi (apps) yang memperluas fungsionalitas perangkat, umumnya ditulis dalam versi kustomisasi bahasa pemrograman Java. Pada bulan Oktober 2013, ada lebih dari satu juta aplikasi yang tersedia untuk Android, dan sekitar 50 miliar aplikasi telah diunduh dari Google Play, toko aplikasi utama Android. Sebuah survei pada bulan April-Mei 2013 menemukan bahwa Android adalah platform paling populer bagi para pengembang, digunakan oleh 71% pengembang aplikasi bergerak. Di Google I/O 2014, Google melaporkan terdapat lebih dari satu miliar pengguna aktif bulanan Android, meningkat dari 583 juta pada bulan Juni 2013.

Faktor-faktor di atas telah memberikan kontribusi terhadap perkembangan Android, menjadikannya sebagai sistem operasi telepon pintar yang paling banyak digunakan di dunia, mengalahkan Symbian pada tahun 2010. Android juga menjadi pilihan bagi perusahaan teknologi yang menginginkan sistem operasi berbiaya rendah, bisa dikustomisasi, dan ringan untuk perangkat berteknologi tinggi tanpa harus mengembangkannya dari awal. Sifat Android yang terbuka juga telah mendorong munculnya sejumlah besar komunitas pengembang aplikasi untuk menggunakan kode sumber

terbuka sebagai dasar proyek pembuatan aplikasi, dengan menambahkan fitur-fitur baru bagi pengguna tingkat lanjut atau mengoperasikan Android pada perangkat yang secara resmi dirilis dengan menggunakan sistem operasi lain.

Pada November 2013, Android menguasai pangsa pasar telepon pintar global, yang dipimpin oleh produk-produk Samsung, dengan persentase 64% pada bulan Maret 2013. Pada Juli 2013, terdapat 11.868 perangkat Android berbeda dengan beragam versi. Keberhasilan sistem operasi ini juga menjadikannya sebagai target litigasi paten "perang telepon pintar" antar perusahaan-perusahaan teknologi. Hingga bulan Mei 2013, total 900 juta perangkat Android telah diaktifkan di seluruh dunia, dan 48 miliar aplikasi telah dipasang dari Google Play.

## **2.6 UML (Unified Model Language)**

Menurut Booch (2005:7) UML adalah Bahasa standar untuk membuat rancangan software. UML biasanya digunakan untuk menggambarkan dan membangun, dokumen artifak dari software –intensive system.

Menurut Nugroho (2010:6), UML (Unified Modeling Language) adalah ‘bahasa’ pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek’. Pemodelan (modeling) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami.

Menurut Nugroho (2009:4), UML (Unified Modeling Language) adalah Metodologi kolaborasi antara metoda-metoda Booch, OMT (Object Modeling Technique), serta OOSE (Object Oriented Software Engineering) dan beberapa metoda lainnya, merupakan metodologi yang paling sering digunakan saat ini untuk analisa dan perancangan sistem dengan metodologi berorientasi objek mengadaptasi maraknya penggunaan bahasa “pemrograman berorientasi objek” (OOP).

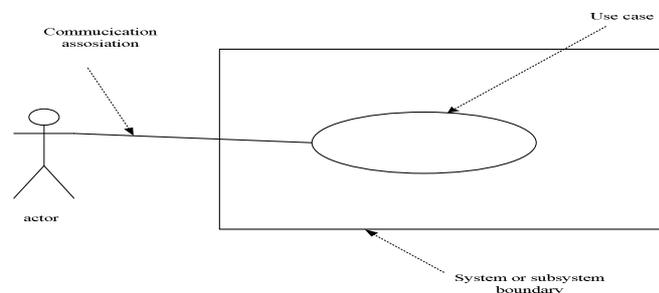
Menurut Herlawati& Widodo (2011:10), bahwa beberapa literature menyebutkan bahwa UML menyediakan sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa diagram yang digabung, misanya diagram komunikasi, diagram urutan dan diagram pewaktuan digabung menjadi diagram interaksi.

Menurut Herlawati &Widodo (2011:10), “Beberapa literature menyebutkan bahwa UML menyediakan sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa diagram yang digabung, misanya diagram komunikasi, diagram urutan dan diagram pewaktuan digabung menjadi diagram interaksi”. Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis.

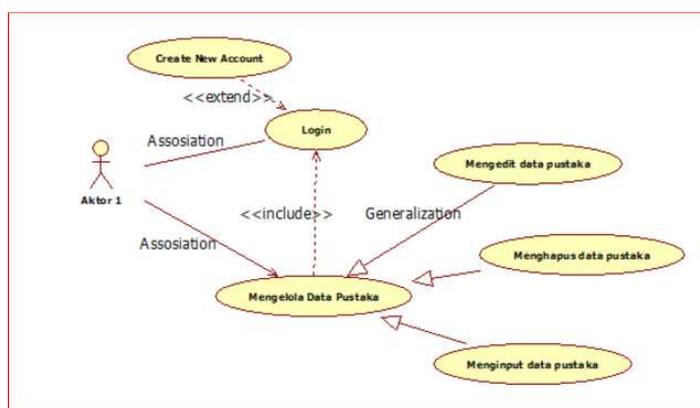
### 2.6.1 Use Case

Menurut Booch (2005), suatu *use case diagram* menampilkan kumpulan *use case* dan aktor (pelaku) dan hubungan di antara *use case* dan aktor tersebut. *Use case diagram* digunakan untuk penggambaran *use case* statistik dari suatu sistem. *Use case diagram* penting dalam mengatur dan memodelkan kelakuan dari suatu sistem.

*Use case* menjelaskan apa yang dilakukan sistem (atau subsistem) tetapi tidak menspesifikasikan cara kerjanya. *Flow of event* digunakan untuk menspesifikasikan kelakuan dari *use case*. *Flow of event* menjelaskan *use case* dalam bentuk tulisan dengan sejelas-jelasnya, diantaranya bagaimana, kapan *use case* dimulai dan berakhir, ketika *use case* berinteraksi dengan aktor, obyek apa yang digunakan, alur dasar dan alur alternatif.



**Gambar 2.3** Notasi *Use Case Diagram* (Simonn Bennet, Steve Marcob dan Ray Farmer :2006, p146)



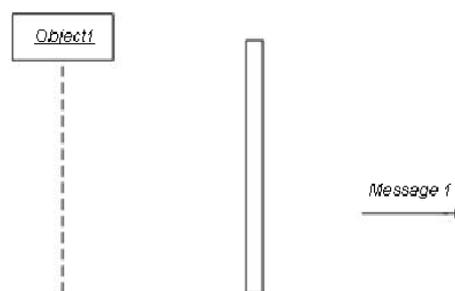
**Gambar 2.4** Contoh Use Case Diagram

## 2.6.2 Sequence Diagram

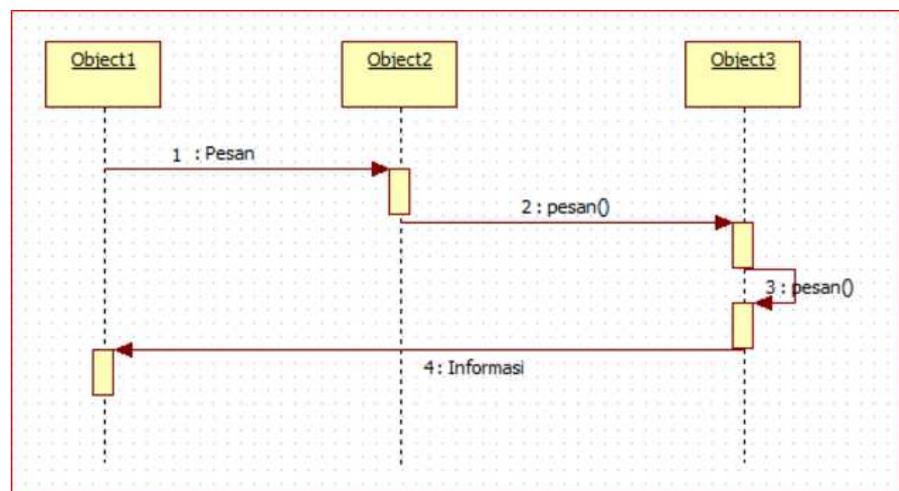
Menurut Booch (2005), suatu *sequencediagram* adalah suatu diagram interaksi yang menekankan pada pengaturan waktu dari pesan-pesan. Diagram ini menampilkan sekumpulan peran dan pesan-pesan yang dikirim dan diterima oleh instansi yang memegang peranan tersebut. *Sequence diagram* menangkap objek dan *class* yang terlibat dalam skenario dan urutan pesan yang ditukar antara objek diperlukan untuk melaksanakan fungsionalitas skenario. *Sequence diagram* berasosiasi dengan *usecase* selama proses pengembangan. Dalam *Unified Model Language (UML)*, objek dalam *sequencediagram* digambar dengan segiempat yang berisi nama objek yang diberi garis bawah. Objek dapat diberi nama dengan tiga cara :

(nama objek), (nama objek dan *class*) atau (hanya nama *class* (*anonymous object*)).

Berikut notasi *sequence diagram* seperti terlihat pada gambar dibawah ini :



**Gambar 2.5** Notasi *Sequence Diagram*



**Gambar 2.6** Contoh *Sequence Diagram*

### 2.6.3 Class Diagram

Menurut Booch (2005), *class diagram* menunjukkan sekumpulan kelas, antarmuka, dan kerjasama serta hubungannya. *Class diagram* digunakan untuk memodelkan

perancangan statik dari gambar sistem. Biasanya meliputi permodelan *vocabulary* dari sistem, permodelan kerjasama, atau permodelan skema.

*Class diagram* dapat digunakan untuk membangun sistem yang dapat dieksekusi

melalui teknik *forward and reverse*, selain untuk penggambaran, penyesuaian, dan pendokumentasian struktur model.

*Class Diagram* terdiri dari:

- a) Nama *Class*.
- b) Atribut.
- c) Operasi/Method.

**Tabel 2.1** *Class Diagram* (Wahono, R.S, 2003)

Nama Class
Atribut

Operasi/method
----------------

Atribut dan Operasi/*method* dapat memiliki tiga sifat berikut:

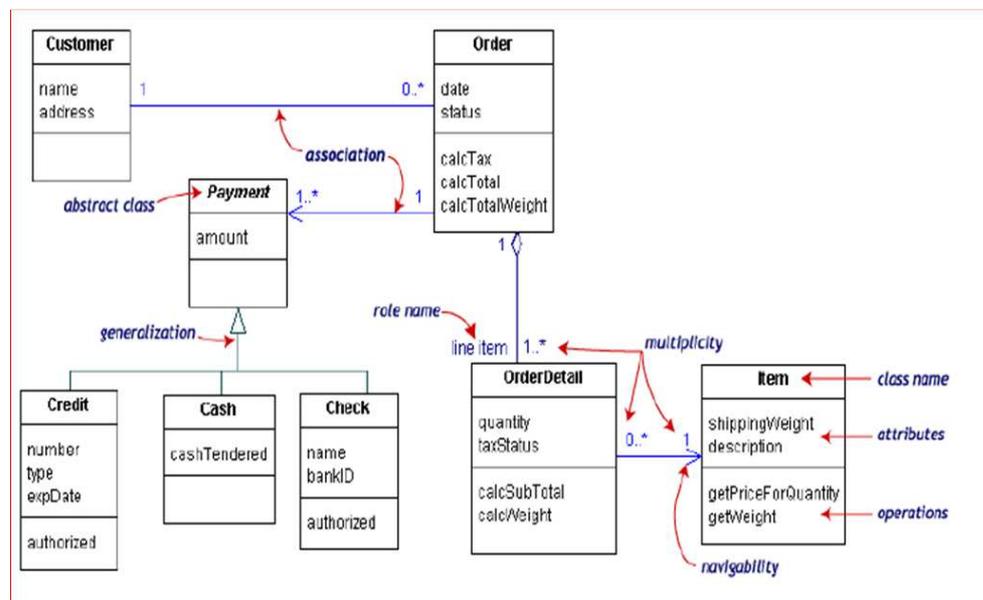
- *Public*, dapat dipanggil oleh *class* apa saja.
- *Protected*, hanya dapat dipanggil atau diakses oleh *class* yang bersangkutan dan *class* turunannya.
- *Private*, hanya dapat dipanggil oleh dirinya sendiri (tidak dapat diakses dari luar *class* yang bersangkutan).

Hubungan antar *class*

1. Asosiasi, yaitu hubungan yang bersifat statis dalam *class*. Asosiasi menggambarkan *class* yang memiliki atribut berupa *class* lain atau *class* yang harus mengenal adanya *class* lain.
2. Agregasi, merupakan hubungan antara satu *object* dengan *object* lainnya dimana *object* satu dengan *object* lainnya sebenarnya terpisah namun

disatukan, sehingga tidak terjadi kebergantungan (*Object* lain bisa ada walau *object* penampungnya tidak ada).

3. Pewarisan, yaitu hubungan hirarki antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metode *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 (*messaging*) yang di-*passing* dari satu *class* kepada *class* lain.

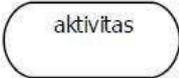
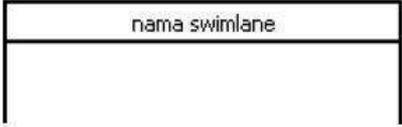


**Gambar 2.7** Contoh *Class Diagram*

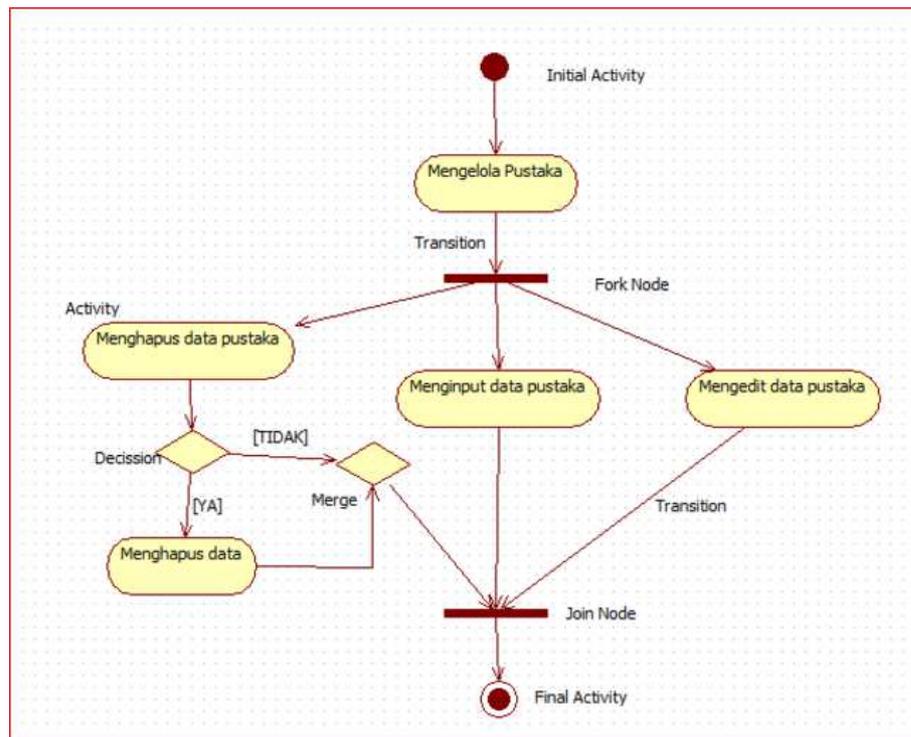
#### 2.6.4 Activity Diagram

Menurut Booch, et al (2007), activity diagram memberikan gambaran visual aliran kegiatan, baik dalam sistem, bisnis, alur kerja atau proses lainnya.

Diagram ini berfokus pada kegiatan yang dilakukan dan siapa yang bertanggung jawab atas kinerja kegiatan tersebut.

Simbol	Deskripsi
status awal 	status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
aktivitas 	aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
percabangan / decision 	asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
penggabungan / join 	asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
status akhir 	status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
swimlane 	memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi
<i>fork,</i> 	digunakan utk menunjukkan kegiatan yg dilakukan secara paralel
<i>join,</i> 	digunakan utk menunjukkan kegiatan yg digabungkan

**Gambar 2.8**Notasi Activity Diagram



**Gambar 2.9** Contoh *Activity Diagram*

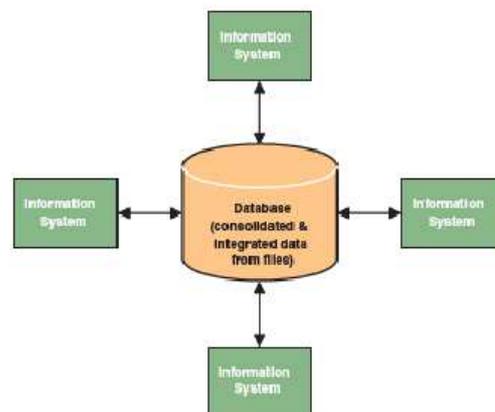
## 2.7 Basis Data

Menurut Connolly dan Begg (2005:14), “*Database* adalah kumpulan data yang saling terhubung satu sama lain yang digunakan secara bersama-sama dan kumpulan data ini dirancang untuk memenuhi kebutuhan informasi suatu perusahaan”.

Basis data (*Database*) adalah sebuah kumpulan dari *files* yang saling berkaitan. (Whitten & Bently, 2007). Teknologi database memiliki kelebihan dalam cara penyimpanan data dengan format yang fleksibel. hal ini memungkinkan dikarenakan *database* terpisah dari sistem informasi dan aplikasi yang menggunakannya. *database* juga dapat berkembang sesuai dengan perubahan kebutuhan organisasi.

Dari definisi mengenai *database* diatas, maka dapat diambil kesimpulan bahwa *database* adalah kumpulan dari data yang terintegrasi satu sama lain dan apabila akan disimpan maka diletakkan pada media penyimpanan tertentu serta untuk mengambil dan memanipulasinya diperlukan prosedur yaitu dalam bentuk program yang berfungsi sebagai perintah.

Whitten menggambarkan posisi *database* dalam sistem adalah seperti gambar berikut :



**Gambar 2.10** Posisi *Database* dalam Sistem

Whitten juga menjelaskan komponen-komponen penyusun sebuah *database* sebagai berikut :

a. Field

*field* adalah unit terkecil dari data yang mempunyai arti untuk disimpan kedalam sebuah file atau *database*. Terdapat empat jenis *field* yang dapat tersimpan.

- *Primary Key*

*Primary key* adalah sebuah *field* atau kelompok dari *field* yang bersifat unik sebagai identitas dari sebuah *record*.

- *SecondaryKey*

*Secondarykey* adalah sebuah *field* yang menjadi identitas dari sebuah *record* atau sebuah bagian dari *record* yang berkaitan.

- *ForeignKey*

*Foreignkey* adalah sebuah *field* yang mengacu kepada *record* yang terdapat pada *file* berbeda dalam sebuah *database*.

- *Descriptivekey*

*Descriptivefield* adalah *field* lain yang tidak menjadi *key*.

b. Record

Record adalah sebuah kumpulan dari field-field yang tersusun dan terikat berdasarkan format tertentu yang sudah ditentukan sebelumnya.

c. *File* dan *Tables*

*File* dalam konteks *database* adalah himpunan semua kejadian yang terjadi pada struktur *record*. *Tables* adalah *database* yang mempunyai relasi setara dengan sebuah *files*.

### 2.7.1 Istilah Istilah Dasar Basis Data

Berikut beberapa istilah dalam basis data yang perlu diketahui diantaranya sebagai berikut :

1. *Entity* adalah objek dalam enterprise yang akan disajikan di dalam database. Contoh : supplier, gudang, pegawai, part, dll.

2. Atribut (elemen data).

Setiap *entity* memiliki atribut-atribut yang mencirikan *entity* tersebut.

Contoh: *entity* pegawai terdiri atas atribut : kode pegawai, alamat, gaji, dan lain-lain.

3. Relasi adalah suatu asosiasi hubungan antara dua atau lebih *entity*. Contoh : mahasiswa mengambil mata kuliah, pasien dirawat dokter, dan lain-lain.

Dalam hubungan antar *entity* dikenal ada dua kunci data penghubung, yaitu:

1. *Primary Key* (Kunci Utama)

Satu *attribute/field* atau satu set *attribute* yang mengidentifikasi secara unik suatu kejadian yang spesifik pada *entity*.

2. *Foreign Key* (Kunci Tamu)

Satu *attribute* atau satu set *attribute* yang melengkapi suatu hubungan yang menunjukkan ke *entity* induknya. Kunci tamu berada pada *entity* anak.

Setiap proyek dalam sistem database dapat dipresentasikan dengan menggunakan beberapa *entity* dan setiap *entity* dapat terdiri dari beberapa elemen data (atribut). Dalam database model relasi dikenal beberapa tipe relasi antara *entity* dan relasi antara elemen data.

Tipe relasi tersebut yaitu :

1. Relasi satu ke satu (One to One)

Menggambarkan hubungan satu ke satu, yaitu satu *record* pada *entity* pertama berhubungan dengan satu *record* pada *entity* kedua atau sebaliknya. Misalnya setiap pasien yang dirawat akan menempati satu tempat tidur untuk satu periode perawatan.

2. Relasi satu ke banyak (One to Many)

Menggambarkan hubungan satu ke banyak, yaitu satu *record* pada *entity* pertama berhubungan dengan banyak *record* pada *entity* kedua atau sebaliknya. Misalnya antara file data karyawan dengan file pendapatan hariannya. Seorang karyawan mempunyai beberapa *record* data pendapatan hariannya, yang keseluruhannya menunjukkan gaji karyawan tersebut pada periode tertentu.

3. Relasi banyak ke banyak (Many to Many)

Menggambarkan hubungan banyak ke banyak, yaitu lebih dari satu *record* pada *entity* pertama berhubungan dengan lebih dari satu *record* pada *entity* kedua atau sebaliknya. Dibutuhkan *associative entity* atau *entity* perantara yang berisi *field* kunci dari masing-masing *entity*.

### 2.7.2 Komponen Sistem Database

Komponen Sistem Database meliputi:

a. Data

Data adalah jantung dari DBMS. Ada dua jenis data. Pertama, kumpulan informasi yang diperlukan oleh suatu organisasi. Kedua, informasi mengenai *database*.

Data harus bersifat :

- Dipakai bersama
- Terintegrasi/terpadu

b. Perangkat keras (*Hardware*)

*Hardware* merupakan sistem komputer yang digunakan untuk menyimpan dan mengakses *database*.

c. Perangkat lunak (*Software*)

*Software* beserta *utility software* adalah DBMS yang aktual. DBMS memungkinkan para user untuk berkomunikasi dengan *database* dengan *user*. Fasilitas yang harus tersedia adalah :

- Membuat file
- Menyisipkan data
- Menampilkan data
- Menghapus data
- Kontrol keamanan

- Kontrol keterpaduan

d. Pemakai (*User*)

Ada sejumlah *user* yang dapat mengakses atau mengambil data sesuai dengan kebutuhan pengguna aplikasi-aplikasi dan *interface* yang disediakan oleh DBMS, antara lain :

- *Database administrator* adalah orang atau *group* yang bertanggungjawab mengimplementasikan sistem *database* di dalam suatu organisasi.
- *End user* adalah orang yang berada di depan *workstation* dan berinteraksi secara langsung dengan sistem.
- *Programmer aplikasi*, orang yang berinteraksi dengan *database* melalui cara yang berbeda.

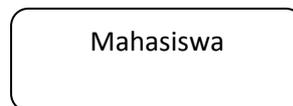
Menurut Whitten & Bentley (2007) Pemodelan data adalah sebuah teknik untuk mengorganisasi dan mendokumentasikan data dalam sistem, terkadang disebut pula pemodelan database. Terdapat beberapa notasi dalam pemodelan data. Pemodelan data yang sering digunakan adalah entity relationship diagram (ERD).

*Entity Relationship Diagram* (ERD) adalah sebuah pemodelan data yang menyediakan beberapa notasi untuk menggambarkan data kedalam bentuk entitas

serta keterkaitan antar data yang dideskripsikannya. Berikut adalah notasi-notasi yang digunakan dalam ERD :

a. Entitas (*Entity*)

*Entity* adalah sebuah kelas dari orang, tempat, objek, kejadian atau suatu konsep dimana perlu untuk dihimpun dan disimpan datanya. Notasi yang digunakan untuk menggambarkan sebuah *entity* adalah seperti berikut



**Gambar 2.11 Notasi *Entity***

Setiap *entity* dalam pemodelan data, memiliki satu atau lebih *attributes*. *Attributes* adalah sebuah properti deskriptif atau karakteristik dari sebuah *entity*. Sehingga *entity* yang memiliki *attributes* dinotasikan seperti gambar berikut.



**Gambar 2.12 Notasi *Entity&Attributes***

b. *Relationship*

*relationship* adalah sebuah asosiasi bisnis secara alami antara satu atau lebih *entity.relationship* data dapat berupa sebuah kejadian yang menghubungkan entitas-entitas atau hanya berupa keterkaitan logika yang terdapat diantara entitas-entitas. Suatu *relationship* digambarkan seperti berikut



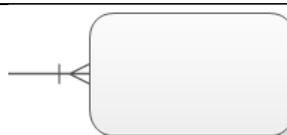
**Gambar 2.13 Notasi *Relationship***

c. Kardinalitas

Kardinalitas adalah jumlah minimal dan maksimal dari kejadian dari satu entitas yang mungkin berkaitan dengan sebuah kejadian di entitas lain. Berikut adalah notasi yang digunakan untuk menggambarkan kardinalitas data antar *entity*.

**Tabel 2.2. Notasi ERD**

Interprestasi Kardinalitas	Minimal <i>Instances</i>	Maksimal <i>Instances</i>	Notasi Gambar
-------------------------------	-----------------------------	------------------------------	---------------

Tepat satu (satu dan dengan hanya satu)	1	1	 Atau 
Nol atau satu	0	1	
Satu atau lebih	1	Banyak (>1)	
Nol, satu atau lebih	0	Banyak (>1)	
Lebih dari satu	>1	>1	

## 2.8 MVC (Model View Controller)

MVC adalah sebuah bentuk pemrograman yang memisahkan berdasarkan logika penanganan tampilan, logika pengontrolan dan logika model. MVC bertujuan supaya pada pengembangan perangkat lunak yang besar mudah untuk dilakukan maintenance (perbaikan/penambahan atau pengurangan code).

Model adalah komponen yang berfungsi mengambil data dari database/sumber data. Fungsi yang terdapat dalam Model akan dipanggil oleh Controller.

View adalah komponen yang berfungsi menyajikan tampilan kepada user. View menampilkan data yang diperoleh Controller.

Controller adalah komponen yang berfungsi untuk memanggil fungsi yang ada di dalam Model dan mengirim hasilnya melalui View, Controller juga berfungsi mengambil input dari user yang selanjutnya akan diolah oleh Model.

Model adalah komponen yang berfungsi mengambil data dari database/sumber data. Fungsi yang terdapat dalam Model akan dipanggil oleh Controller.

## **2.9 Web Service**

Web service menurut W3.org mendefinisikan web service sebagai “sebuah software aplikasi yang dapat teridentifikasi oleh URI dan memiliki interface yang didefinisikan, dideskripsikan, dan dimengerti oleh XML dan juga mendukung interaksi langsung dengan software aplikasi yang lain dengan menggunakan message berbasis XML melalui protokol internet”.

Web service adalah sebuah software aplikasi yang tidak terpengaruh oleh platform, ia akan menyediakan method-method yang dapat diakses oleh network. Ia juga akan menggunakan XML untuk pertukaran data, khususnya pada dua entities bisnis yang berbeda.

Definisi lain : Web service adalah sistem software yang dirancang untuk mendukung interoperabilitas mesin-ke-mesin yang dapat berinteraksi

melalui jaringan. Web service memiliki antarmuka yang dijelaskan dalam format mesin-processable (khusus WSDL). Sistem lain berinteraksi dengan Web service dalam cara ditentukan oleh deskripsi dengan menggunakan pesan SOAP, biasanya disampaikan menggunakan HTTP dengan serialisasi XML dalam hubungannya dengan Web lainnya yang terkait standar.

Dalam pengertian yang sederhana , XML Web Services dapat di definisikan sebagai aplikasi yang diakses oleh aplikasi yang lain. Mungkin orang berpendapat itu semacam web site, tetapi itu bukan demikian. Ada perbedaan – perbedaan yang membedakan dengan web site.

Perbedaan tersebut dapat dilihat dibawah ini :

- **Web Site**

1. Memiliki web interface
2. Dibuat untuk ber interaksi langsung dengan user
3. Dibuat untuk bekerja pada web browser.

- **Web Services**

1. Tidak memiliki interface yang bagus
2. Dibuat untuk ber interaksi langsung dengan aplikasi yang lain baik beda OS / Konsep sekalipun.

3. Dibuat untuk bekerja pada semua tipe client aplikasi / perangkat device

Beberapa karakteristik dari web service adalah:

- Message-based
- Standards-based
- Programming language independent
- Platform-neutral

Beberapa key standard didalam web service adalah: XML, SOAP, WSDL and UDDI.

**SOAP** (*Simple Object Access Protocol*) adalah sebuah XML-based *mark-up language* untuk pergantian pesan diantara aplikasi-aplikasi. SOAP berguna seperti sebuah amplop yang digunakan untuk pertukaran data object didalam network. SOAP mendefinisikan empat aspek didalam komunikasi: Message envelope, Encoding, RPC call

convention, dan bagaimana menyatukan sebuah message didalam protokol transport.

Sebuah SOAP *message* terdiri dari SOAP *Envelop* dan bisa terdiri dari *attachments* atau tidak memiliki attachment. SOAP envelop tersusun dari SOAP header dan SOAP body, sedangkan SOAP attachment membolehkan non-XML data untuk dimasukkan kedalam SOAP message, di-encoded, dan diletakkan kedalam SOAP message dengan menggunakan MIME-multipart.

**WSDL** (*Web Services Description Language*) adalah sebuah XML-based language untuk mendeskripsikan XML. WSDL menyediakan *service* atau layanan yang mendeskripsikan *service request* dengan menggunakan protokol-protokol yang berbeda dan juga encoding. WSDL memfasilitasi komunikasi antar aplikasi. WSDL akan mendeskripsikan apa yang akan dilakukan oleh *web service*, bagaimana menemukannya dan bagaimana untuk mengoperasikannya. Spesifikasi WSDL mendefinisikan tujuh tipe element:

1. Types – element untuk mendefinisikan tipe data. Data yang akan mendefinisikan tipe data (seperti string atau integer) dari element didalam sebuah message.
2. Message – abstract, pendefinisian tipe data yang akan dikomunikasikan.

3. Operation – sebuah deskripsi abstract dari sebuah action yang didukung oleh service.
4. Port Type – sebuah koleksi abstract dari operations yang didukung oleh lebih dari satu endpoints.
5. Binding – mendefinisikan penyatuan dari tipe port (koleksi dari operasi-operasi) menjadi sebuah protokol transport dan data format (ex. SOAP 1.1 pada HTTP). Ini adalah sebuah protokol konkret dan sebuah spesifikasi data format didalam tipe port tertentu.
6. Port – mendefinisikan sebuah komunikasi endpoint sebagai kombinasi dari binding dan alamat network. Bagi protokol HTTP, sebuah bentuk dari URL sedangkan bagi protokol SMTP, ini adalah sebuah form dari email address.
7. Service – satu set port yang terkorelasi atau suatu endpoints.

**WSDL** mendefinisikan service sebagai sebuah koleksi dari endpoints network. Sebuah definisi abstrak dari endpoints dan messages adalah ia bersifat terpisah dari pembangunan network atau penyatuan data format. Pembagian ini menyebabkan penggunaan kembali *abstract description* dari data yang akan dipertukarkan (message exchange) dan abstract collection dari operasi (*ports*) Protokol konkret dan spesifikasi data format bagi tipe port tertentu menentukan binding yang dapat digunakan kembali(reusable). Sebuah

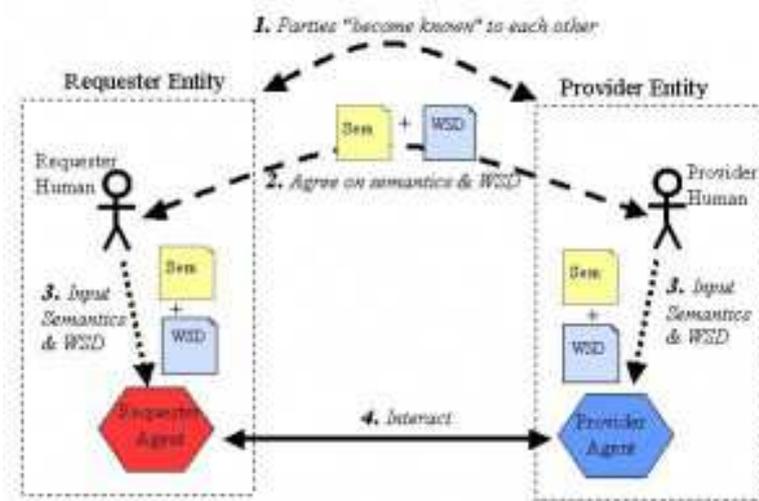
port adalah sebuah network address yang dikombinasikan reusable binding; sebuah service adalah koleksi dari port-port.

Sedangkan **UDDI** (*Universal Description, Discovery and Integration*) adalah sebuah service registry bagi pengalokasian web service. UDDI mengkombinasikan SOAP dan WSDL untuk pembentukan sebuah registry API bagi pendaftaran dan pengenalan service. Ia menyediakan sebuah area umum dimana sebuah organisasi dapat mengiklankan keberadaannya dan service yang diberikan (*web service*).

Semantik pada Web service adalah harapan bersama tentang perilaku layanan, khususnya dalam menanggapi pesan yang dikirim ke tujuan. Akibatnya, ini adalah “kontrak” antara entitas pemohon dan badan penyedia tentang tujuan dan konsekuensi dari interaksi. Meskipun kontrak ini merupakan keseluruhan perjanjian antara entitas penanya dan entitas penyedia tentang bagaimana dan mengapa masing-masing agen akan berinteraksi, itu belum tentu tertulis atau eksplisit dinegosiasikan. Ini mungkin eksplisit atau implisit, lisan atau tertulis, mesin processable atau manusia berorientasi, dan mungkin suatu perjanjian hukum atau kesepakatan informal (non-hukum).

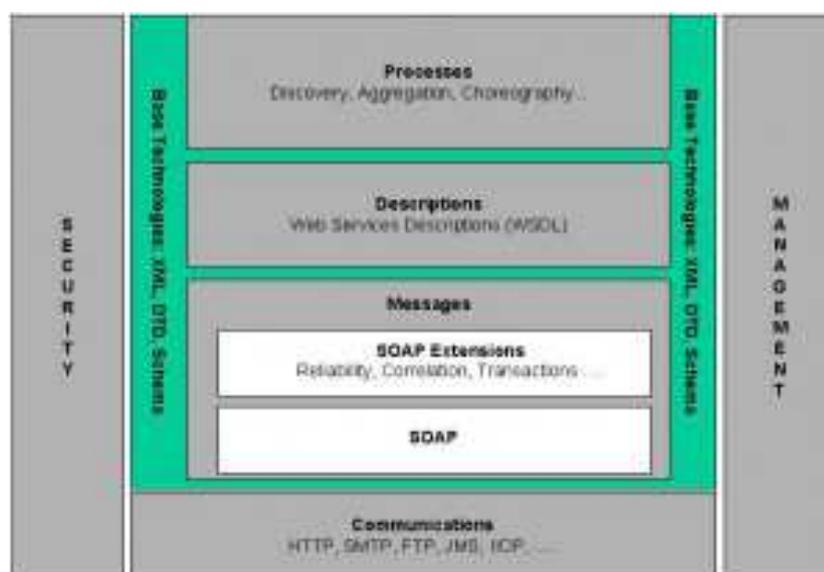
Ada banyak cara bahwa entitas penanya mungkin terlibat dan menggunakan Web service. Secara umum, langkah-langkah yang luas berikut yang diperlukan, seperti yang diilustrasikan pada Gambar 1. (1) pemohon dan penyedia entitas menjadi dikenal satu sama lain (atau setidaknya satu menjadi tahu untuk yang lain); (2) peminta dan penyedia entitas entah bagaimana

setuju pada deskripsi layanan dan semantik yang akan mengatur interaksi antara pemohon dan agen penyedia; (3) deskripsi layanan dan semantik direalisasikan oleh pemohon dan agen penyedia, dan (4) pemohon dan agen penyedia bertukar pesan, sehingga melakukan beberapa tugas atas nama pemohon dan badan penyedia. (Ie, pertukaran pesan dengan agen penyedia merupakan wujud nyata dari berinteraksi dengan layanan Web penyedia entitas.)



**Gambar 2.14.** Proses Umum Web Service

Arsitektur Web service melibatkan teknologi berlapis banyak dan saling terkait. Ada banyak cara untuk memvisualisasikan teknologi ini, seperti halnya ada banyak cara untuk membangun dan menggunakan Web service. Gambar 2 di bawah ini memberikan sebuah ilustrasi dari beberapa keluarga teknologi.



**Gambar 2.15** *Arsitektur Web Service*

Web Services itu digunakan saat kita akan mentransformasi sebuah bisnis logik / sebuah class dan object yang terpisah dalam 1 ruang lingkup yang menjadi satu, sehingga tingkat keamanan dan *security* dapat di tangani dengan baik. Selain itu Web Service juga lebih mudah dalam process deploymentnya, karena tidak memerlukan registrasi khusus ke dalam sistem operasi. *Web Service* cukup diupload ke *Web Server* dan siap diakses oleh pihak-pihak yang telah diberikan otorisasi. Web Service berjalan di port 80 yang merupakan protokol standar HTTP, dengan demikian mengurangi resiko terblokir oleh firewall. Kendala arsitektur COM/DCOM adalah memerlukan konfigurasi khusus di sisi firewall, dan ini tidak perlu dilakukan untuk mengakses *Web Service*.

Beberapa vendor luar negeri mulai berkolaborasi satu sama lain dengan konsep web services , diantaranya : IBM, Microsoft , SUN , ORACLE

Diantaranya contoh web services yang sudah jadi dan dipakai adalah web services keluaran Microsoft ( Microsoft Passport ) – web services untuk user name dan password yang sudah dipasang di web site Microsoft dan HOTMAIL.