BAB 2
LANDASAN TEORI

2.1 *Application Service Provider* (ASP)

Secara jelas belum ada definisi umum atau pasti mengenai ASP, tetapi banyak yang telah mengemukakan pendapat mengenai definisi dari ASP. Secara konsep, ASP atau sering juga disebut "***Rent-An-App***" merupakan konsep dari *digital business design* dimana sebuah perusahaan melakukan perubahan dari cara konvensional yang menggunakan kertas sebagai sarana utama penyampaian informasi ke cara digital dimana berusaha mengurangi pemakaian kertas sebagai penyampaian informasi (*paperless*).

Ada perbedaan yang cukup jauh namun mendekati di antara dua konsep ini. ASP merupakan perubahan baru penyampaian aplikasi. ASP sendiri mewakili suatu bentuk baru dari model bisnis yang ada pada saat ini. Perusahaan ASP yang biasanya tergabung dari *software provider* dan *service provider* menyediakan atau menyewakan aplikasi-aplikasi dari *office tool* sampai aplikasi-aplikasi bentuk proses bisnis yang khusus secara *online* bagi perusahaan kecil, menengah dan perusahaan besar.

Pada dasarnya ASP menyediakan aplikasi dan jasa yang diperuntukkan bagi perusahaan kecil dan perorangan dengan pembayaran setiap pemakaian (*pay-per-use*) atau ijin/iuran tahunan. Sedangkan perusahaan besar lebih memilih mengembangkan ASP untuk keperluan sendiri, memindahkan aplikasi dari PC/*personal* komputer ke suatu aplikasi *server* dimana *server* tersebut dirancang untuk menyediakan aplikasi bagi *workstation* yang ada.

uraian di atas, maka definisi umum ASP adalah :

Service Provider (ASP) adalah sebuah perusahaan yang menyediakan aplikasi dan jasa kepada perorangan atau perusahaan yang seharusnya aplikasi tersebut berada di personal atau *enterprise* komputer secara *online* (Mattson, Paul Robert, 2000, p15).

Sedangkan definisi ASP secara teknis :

ASP adalah *provider* atau penyedia sumber daya komputer, dimana sumber daya tersebut dioperasikan secara *online*, diletakkan dan diakses melalui jarak jauh (*remotely*) pada suatu bentuk *data center* dengan kepercayaan dari konsumen pemakai dan juga bertanggung jawab untuk menyediakan segala aktivitas secara spesifik dan profesional untuk pemeliharaan aplikasi perangkat lunak (anonymous,www.cherrytreeco.com).

Stability, reliability, availability, scaleability, performance dan *security* merupakan unsur dasar konsep ASP (anonymous,www.cherrytreeco.com).

Ada beberapa faktor yang menyebabkan ASP sangat diperlukan sekarang ini. Saat ini seluruh dunia sedang mempersiapkan diri untuk menghadapi pasar bebas dimana semua perusahaan akan bersaing secara bebas dan terbuka. Perusahaan-perusahaan pada kelas kecil dan menengah harus dapat bersaing dengan perusahaan-perusahaan besar berskala internasional.

Perusahaan-perusahaan tersebut harus dapat melakukan perubahan-perubahan dalam profesionalisme dan teknologi dalam hal komputerisasi. Terutama persaingan dalam hal teknologi yang menyangkut *brainware*, perangkat lunak dan perangkat keras. Hal-hal inilah yang menyebabkan mengapa sampai diperlukan ASP.

Ada 2 faktor utama dan 1 faktor pendukung yang menyebabkan munculnya ASP, yaitu faktor dari segi bisnis dan faktor dari segi teknis. Faktor-faktor dari segi bisnis adalah (anonymous,www.cherrytreeco.com) :

1. *Minimize Total Cost of Ownership* (TCO).

   Pilihan alternatif untuk menggunakan ASP dapat membuat perusahaan melakukan penghematan sekitar 30% - 50% dari dana yang biasanya dikeluarkan untuk TCO.

2. *Predictability of cash flows.*

   Konsep ASP dapat membuat perusahaan untuk lebih pasti dalam merancang anggaran belanja dengan menghilangkan ketidakpastian berapa besar biaya untuk perancangan dan pengembangan aplikasi perusahaan.

3. *Focus on core competencies and strategic objectives.*

   Dengan menggunakan pihak ketiga dalam hal ini ASP, untuk menangani aplikasi yang dipakai dapat membuat perusahaan lebih terfokus untuk meningkatkan bisnis perusahaan.

4. *Improve coordination efforts on a global basis.*

   Konsep ASP dapat membuat organisasi/perusahaan yang menggunakan teknologi terkini untuk mengkoordinasikan operasi global baik dalam maupun luar.

Faktor dari segi teknis adalah (anonymous,www.cherrytreeco.com) :

1. *Shortage of skilled IT labor.*

   Perusahaan kecil belum tentu mampu untuk membiayai pencarian, pelatihan dan pengkajian para staf IT.

2. *Utilization of emerging technologies and "best of breed" applications.*

   ASP dapat membuat perusahan-perusahaan kecil untuk dapat mengunakan aplikasi-aplikasi khusus seperti *Customer Relationship Management* (CRM). Saat ini aplikasi-aplikasi tersebut hanya mampu digunakan oleh perusahaan besar.

3. *Rapidly changing and increasing complexity of technology.*

   Departemen/bagian IT dalam sebuah perusahaan harus menghadapi pengembangan teknologi dan pengembangan aplikasi. Konsep ASP dapat menghilangkan semua asumsi biaya yang dikeluarkan untuk pengembangan aplikasi.

4. *Obtain technical expertise.*

   Banyak ASP saat ini yang terfokus pada satu bidang usaha. Hal ini yang membuat perusahaan mengunakan ASP sesuai spesifikasi kebutuhannya.

Selain faktor dari segi bisnis dan teknis, ada pula faktor pendukung yang menyebabkan ASP dapat terwujud. Faktor pendukung tersebut adalah faktor kemampuan teknologi yang terdiri atas (anonymous,www.cherrytreeco.com) :

1. *Ubiquity of the Internet.*

   ASP dapat terwujud karena perkembangan Internet dan pengembangan *web-enabled* secara terus menerus.

2. *Shared applications in a client/server environment.*

   Remote access pada konsep ASP bukanlah suatu hal yang baru, karena *user* telah terbiasa dengan teknologi *client/server.*

3. *Browsers as an accepted GUI application.*

   Kemampuan *browser* untuk menerima *graphical user interface* (GUI) telah meningkatkan kepopuleran dari *web-enabled* dan *thin-client* komputer.

Akan tetapi dari semua hal yang telah disebutkan diatas, ada satu hal yang benar-benar diperhatikan oleh ASP industri. ASP yang bergerak secara *vertical* terhadap satu bidang usaha atau yang biasa disebut sebagai *Vertical Service Provider* (VSP) memiliki kendala yang lebih banyak dari ASP lainnya. Walaupun semua perusahaan yang bergerak pada bidang usaha yang sama memiliki *line of business* yang sama akan tetapi tetap memiliki *rule-rule*/aturan-aturan yang berbeda.

Oleh karena itu, VSP harus menyediakan aplikasi CRM yang terintegrasi dengan proses bisnis sesuai dengan *rule-rule* yang dimiliki oleh *client*. VSP menyediakan *end-to-end user consulting, development* dan *integration* untuk menghadapi masalah tersebut.

2.2 *Multilevel Marketing* (MLM)

MLM biasanya disebut juga sebagai *network marketing, consumer direct marketing* atau *seller assisted marketing* dan istilah-istilah lainnya yang banyak muncul ke permukaan.

MLM merupakan sebuah perusahaan yang menjual produk-produknya melalui jaringan yang telah dibuat oleh para anggotanya/distributor. Anggota ini bekerja secara mandiri tanpa campur tangan perusahaan secara langsung (Andrias Harefa, 1999, p3). Sebagai imbalannya para anggota/distributor tersebut memperoleh potongan harga dan bonus/komisi yang besarnya ditetapkan oleh perusahaan sesuai dengan besarnya transaksi penjualan distributor tersebut.

MLM biasa disamakan dengan bisnis apa saja yang akan memberi bayaran apabila memiliki dua *level* atau lebih. Sebagai contoh, apabila ingin melakukan transaksi

penjualan, maka si penjual akan memperoleh poin untuk dirinya pribadi dan orang yang ada di atasnya.

Network marketing merupakan seorang distributor *network* yang dibutuhkan untuk membangun bisnis ini. Biasanya bisnis seperti itu juga disebut MLM disebabkan karena pembayaran dilakukan apabila telah memiliki lebih dari satu *level*.

Network marketing biasanya salah diartikan bahwa bisnis ini mengunakan jaringan *supplier* untuk melakukan penjualan produk. Hal ini biasanya digunakan untuk memberi gambaran pasti bahwa program yang digunakan lebih baik dibandingkan program-program penjualan lainnya.

Sedangkan *consumer direct marketing* menggambarkan bahwa jaringan distributor lebih dianggap sebagai konsumen daripada sebagai distributor. Hal ini yang menjadikan mengapa distributor lebih memilih membeli produk untuk penggunaan pribadi dibandingkan untuk dijual kembali kepada orang lain.

Seller assisted marketing plans merupakan istilah yang digunakan untuk menggambarkan beberapa jenis usaha termasuk MLM. Program ini mengharuskan seseorang untuk melakukan investasi berupa uang yang besarnya telah ditentukan untuk mendukung program ini, yaitu *seller assisted marketing plan*.

Setiap perusahaan MLM pasti memiiiki peraturan-peraturan yang berbeda. Akan tetapi, peraturan-peraturan tersebut harus sesuai dengan hukum yang berlaku. Secara umum konsep MLM terdiri dari tiga jenis, yaitu MLM murni, MLM *binary* dan MLM yang dikenal sebagai *money game*.

Konsep dari MLM murni adalah perusahaan yang bergerak dalam bidang penjualan produk-produk yang diproduksi atau dikeluarkan hanya oleh perusahaan tersebut. Produk-produk tersebut dijual dengan memanfaatkan jaringan yang telah

dibangun oleh anggota perusahaan tersebut baik itu untuk pemakaian pribadi maupun penjualan langsung kepada konsumen.

Setiap perusahaan yang bergerak dalam satu bidang usaha yang sama pasti memiliki proses bisnis yang sama, akan tetapi aturan proses bisnisnya pasti berbeda. Begitu pula halnya dengan MLM. Perbedaan bidang usaha MLM dengan bidang usaha lain adalah pada proses bisnisnya, karena pada MLM proses bisnis yang sama hanya terletak pada bentuk tujuan/konsep dari MLM tersebut.

Persamaan lainnya hanya terletak pada bentuk jaringan yang dibangun oleh anggota perusahaan tersebut, bentuk pendaftaran anggota (hanya dalam hal syarat menjadi anggota yang berbeda) dan bentuk penjualan produk yang terdiri dari harga anggota, harga konsumen dan poin atas setiap produk. Diluar hal tersebut setiap MLM tidak memiliki begitu banyak kesamaan. Perbedaan dari setiap MLM murni terletak pada bentuk akumulasi perhitungan poin anggota dan tingkatan-tingkatan anggota.

2.3 *Use Case*

Hal pertama yang harus diketahui dalam membuat sebuah aplikasi bisnis secara online yang dapat digunakan secara global oleh beberapa perusahaan adalah proses bisnis yang dijalankan. Tanpa mengetahui proses bisnis, aplikasi tersebut tidak dapat terwujud.

Metode yang dapat digunakan dalam melakukan proses perekaman bisnis suatu perusahaan adalah *Use Case*. *Use Case* ini biasanya digunakan untuk menggambarkan keperluan-keperluan yang dibutuhkan oleh suatu sistem. *Use Case* juga dapat digunakan untuk membuat sebuah jadwal proyek dan membantu dalam menentukan hal-hal apa saja yang harus dilakukan dalam proyek tersebut.

Use Case yang berisi tentang rekaman atas gambaran proses bisnis yang dijalankan oleh suatu perusahaan dinamakan *Business Use Case. Business Use Case* ini terdiri atas *System Level Business Use Case* dan *Business Use Case* itu sendiri ditambah dengan uraian dari *Business Use Case*.

System Level Business Use Case merupakan gambaran dari proses bisnis yang dijalankan oleh perusahaan secara umum. *Business Use Case* merupakan gambaran sub proses dari setiap proses yang ada pada *System Level Business Use Case*. Sedangkan uraiannya merupakan penjelasan langkah demi langkah yang dilakukan pada setiap sub proses yang terdapat dalam *Business Use Case*.

Use Case sendiri merupakan gambaran dari rancangan sistem yang dibuat. *Use Case* juga terdiri dari *System Level Use Case*, *Use Case* itu sendiri berikut uraiannya.

Sama halnya dengan *System Level Business Use Case, System Level Use Case* juga merupakan gambaran sistem secara umum yang akan dibuat. Dari *System Level Use Case* juga dapat dipecah kembali menjadi beberapa *Use Case*. *Use Case* ini merupakan gambaran sub proses dari setiap proses yang terdapat dalam *System Level Use Case.*

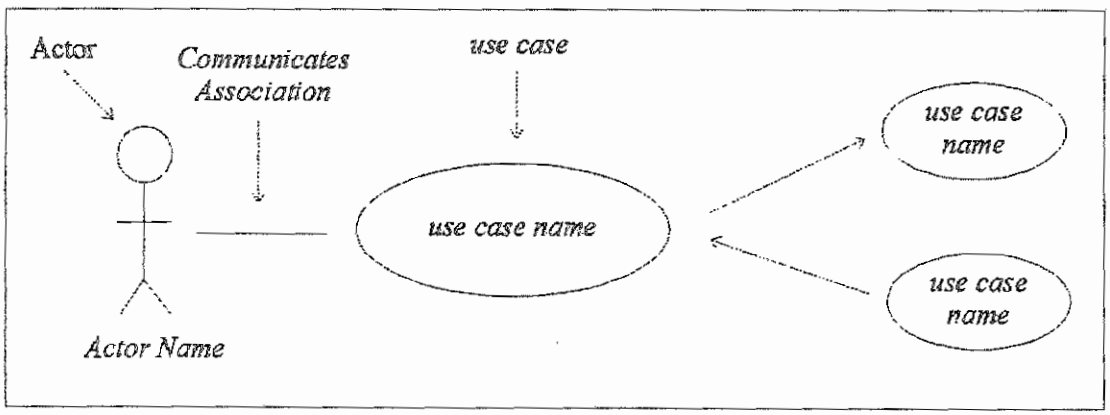
Use Case ini juga biasanya dilengkapi dengan uraian. Uraian ini juga merupakan bentuk penjelasan dari apa yang ada dalam *Use Case*.

Sebuah *Use Case* diharapkan dapat berupa *Use Case* yang terstruktur, tetapi memiliki cara yang informal dalam mengekspresikan kebutuhan-kebutuhan sistem. Selain itu, *Use Case* juga diharapkan dapat dengan mudah diakses.

Dalam membuat sebuah *Use case* yang baik, informasi-informasi yang dikumpulkan haruslah jelas. Informasi-informasi tersebut dapat berupa deskripsi proyek, faktor pasar yang mempengaruhi, faktor resiko yang akan dihadapi dan beberapa asumsi yang berkaitan dengan proyek tersebut.

Komponen utama dari sebuah *Use Case* adalah *actor*. *Actor* adalah segala sesuatu yang dapat berinteraksi dengan sistem yang dibuat (Geri Schneider, Jason P. Winters, 1998, p12). Ada beberapa contoh dari *actor*, yaitu manusia, perangkat lunak, perangkat keras, *data store* dan jaringan. Setiap *actor* memiliki peranan masing-masing dalam sistem. Apabila seseorang memiliki peranan yang berbeda-beda, maka dapat direpresentasikan oleh beberapa *actor*. Sebaliknya apabila beberapa orang memiliki peranan yang sama, maka dapat direpresentasikan oleh sebuah *actor*.

Selain *actor*, nama dari *Use Case* itu sendiri juga merupakan komponen dari diagram *Use Case*. Untuk lebih jelasnya perhatikan Gambar 2.1 di bawah ini.



Gambar 2.1 Komponen Diagram *Use Case* (Gary K.Evans,William F. Nazzaro, 1999, p7)

Sebuah *Use Case* yang lengkap harus memuat semua rincian yang dibutuhkan. Seperti telah dijelaskan di atas, biasanya sebuah *Use Case* dilengkapi dengan sebuah *description*. *Description* ini berfungsi untuk menjelaskan cara-cara kerja dari *Use Case* tersebut yang terdiri dari (Geri Schneider, Jason P. Winters, 1998, p25) :

1. *Brief Description* .

Brief Description ini berisi penjelasan secara umum tentang isi dari *Use Case*.

2. *Flow of Events*

   Flow of Events ini berisi tentang penjelasan mengenai langkah-langkah kerja dari *Use Case*. *Flow of Events* terbagi menjadi dua bagian, yaitu :

   a. *Basic Flows* yang menjelaskan langkah-langkah yang harus dijalankan oleh *Use Case*.

   b. *Alternative Flows* yang menjelaskan langkah-langkah pilihan yang dapat dilakukan.

3. *Special Requirement*

   Special Requirement ini berisi tentang syarat-syarat khusus yang dibutuhkan untuk menjalankan *Use Case*.

4. *Pre Conditions*

   Pre Conditions ini merupakan langkah awal yang harus dilakukan untuk menjalankan *Use Case*.

5. *Post Conditions*

   Post Conditions ini merupakan hasil akhir yang diperoleh bila *Use Case* dijalankan.

Sebuah *Use Case* juga biasanya dilengkapi dengan *Activity Diagram*. *Activity Diagram* ini merupakan gambaran dari langkah-langkah kerja suatu *Use Case* yang disajikan dalam bentuk sebuah diagram (Geri Schneider, Jason P. Winters, 1998, p55).

Setiap langkah dalam *Use Case* akan digambarkan dalam sebuah kotak yang ujungnya berbentuk setengah lingkaran dan perpindahan dari setiap langkah ditunjukkan dengan sebuah tanda panah. Bentuk belah ketupat yang terdapat dalam sebuah *Activity Diagram* menunjukkan bahwa langkah tersebut merupakan langkah pilihan yang dapat dipilih oleh *user*.

Bentuk nyata dari informasi-informasi yang telah dikumpulkan adalah representasi dari informasi-informasi tersebut. Bentuk konkrit tersebut merupakan sebuah *user interface* yang dibuat dalam bentuk *storyboard*.

2.4 System Development Life Cycle

System Development Life Cycle (SDLC) merupakan sebuah metodologi untuk mengembangkan sistem. SDLC menyediakan langkah-langkah pengerjaan secara konsisten untuk membangun sebuah sistem (Southwest Texas State University, 1997).

Metodologi SDLC dapat dimodifikasi untuk diikutsertakan hanya pada proyek tertentu yang bertujuan untuk pembangunan sebuah sistem, baik itu secara otomatis atau manual dan merupakan sistem baru atau kelanjutan dari sistem yang telah ada.

SDLC memiliki enam langkah (Southwest Texas State University, 1997), yaitu :

1. *System Initiation and Feasibility Study*.

   Melakukan dokumentasi atas sistem yang ada saat ini, tujuan dan manfaat pembuatan sistem baru atau pengembangan dari sistem yang telah ada, indentifikasi untuk alternatif solusi sistem dan penentuan solusi secara pasti.

2. *System Analysis*.

   Membuat sketsa awal untuk perencanan proyek dan dokumentasi untuk sistem baru.

3. *System Design*.

   Menghasilkan spesifikasi *file/database*, spesifikasi *input/output*, spesifikasi modul program dan rencana proyek yang lebih rinci. Secara rinci sistem perancangan menggambarkan bagaimana memenuhi kebutuhan untuk pembangunan yang telah ditentukan pada saat melakukan *system analysis*.

4. *Programming.*

   Membuat dan menghasilkan program yang dibutuhkan untuk sistem baru.

5. *Implementation.*

   Secara rinci implementasi melakukan *user testing* terhadap program yang telah dibuat, *files/database conversion*, *user training* dan melakukan pengenalan terhadap sistem baru.

6. *Post-Implementation Evaluation.*

   Melakukan *review* apakah sistem ini memenuhi kebutuhan *user*, penyaranan untuk pengembangan proses SDLC dan membuat jadwal secara berkala untuk melakukan sistem *review*.