

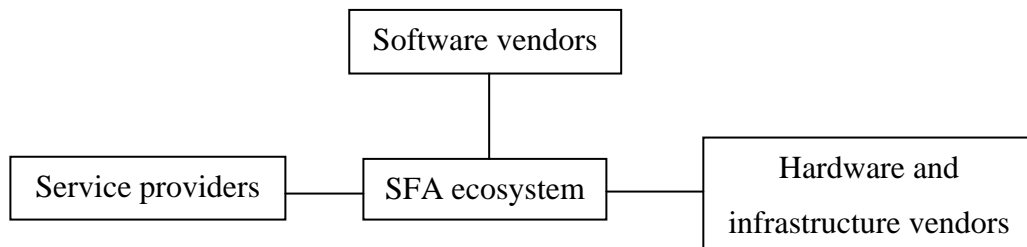
2. LANDASAN TEORI

2.1. *Sales Force Automation (SFA)*

SFA adalah aplikasi berbasis teknologi informasi yang mendukung fungsi kerja penjualan (Buttle, Ang, dan Iriana, 2006). Teknologi informasi yang dimaksud terdiri dari *hardware* dan *software*. *Software* merupakan pengaplikasian SFA itu sendiri, sementara *hardware* tidak terbatas kepada *desktop* atau *laptop* saja namun juga berbagai *device* yang dibutuhkan. Berikut daftar fungsi-fungsi yang ditawarkan oleh *software* SFA.

1. *Account management*
2. *Contact and activity management*
3. *Event management*
4. *Order and contract management*
5. *Document management and product encyclopaedias*
6. *Incentive management*
7. *Lead, opportunity and pipeline management*
8. *Product configuration and visualization*
9. *Sales forecasting*
10. *Territory management*
11. *Work-flow and process development*
12. *Proposal generation and quotation management*

SFA tidak mampu berdiri sendiri. Ada banyak aspek yang mendukung jalannya SFA. Gambar 2.1 merupakan diagram ekosistem dari SFA.



Gambar 2.1. Diagram ekosistem SFA

Sumber: Buttle, Ang, dan Iriana (2006).

Dari diagram di atas dapat diketahui bahwa SFA tidak bisa berjalan hanya dengan vendor *software* namun SFA juga membutuhkan *hardware* sebagai wadah untuk menjalankan *software* tersebut seperti Android. Selain itu, proyek SFA yang sudah kompleks kerap membutuhkan koneksi, sebagai contoh untuk mengirim *e-mail* dan fungsi lain yang membutuhkan jaringan internet.

Sementara itu, hasil penelitian mengenai faktor yang mendorong perusahaan dalam mengimplementasikan SFA dapat dilihat di Tabel 2.1.

Tabel 2.1. Motivasi perusahaan dalam mengimplementasikan SFA

<i>Motivation</i>	<i>% of sample reporting</i>
<i>Improve efficiencies</i>	72
<i>Improve customer contact</i>	44
<i>Increase sales</i>	33
<i>Reduce costs</i>	26
<i>Improve accuracy</i>	21

Sumber: Erffmeyer dan Johnson (2001).

Keuntungan pengaplikasian SFA:

1. Bagi *Sales Force*: siklus penjualan yang lebih pendek, semakin banyak pelanggan baru, peningkatan kemungkinan sukses.
2. Bagi manajer penjualan: peningkatan produktivitas *Sales Force*, peningkatan relasi terhadap *customer*, laporan yang lebih akurat, pengurangan pengeluaran dana pada penjualan.
3. Bagi petinggi perusahaan: arus kas semakin cepat, peningkatan pendapatan penjualan, pertumbuhan pangsa pasar, peningkatan profitabilitas.

Berbagai penelitian dan riset telah dilakukan tentang bagaimana pengimplementasian SFA, *training* apa yang perlu dilakukan, siapa saja yang harus terlibat, teknologi apa yang digunakan dalam *software*, dan apa saja yang berubah semenjak dilakukan pengimplementasian SFA. Dari riset-riset tersebut, diambil suatu kesimpulan *yaitu critical success factor* apa saja yang mempengaruhi pengimplementasian SFA, yang tertera dalam Tabel 2.2.

Tabel 2.2. *Critical Success Factor* dari SFA

Category of variable	Item
Organizational/cultural	Senior management facilitation and buy-in; organizational encouragement and support; customer orientation; adaptive cultural norms; entrepreneurial values.
Project-related	Well-developed project plan; user group involvement; customer involvement; deployment of a multidisciplinary implementation team; appropriate-to-task training.
Inter-personal	Sensitive management of user expectations; information-sharing culture; high levels of inter-personal trust.
Intra-personal	Strong belief that the technology will enable desired outcomes; adaptability; perceived usefulness of the technology; perceived ease-of-use; skills in computer use.
Technical	Access to a competent IT/IS resource; data availability; fragmented and poor quality data; development costs; degree of change required; technology fit to job role.

Sumber: Buttle, Ang, dan Iriana (2006).

2.2. Taksonomi

Konsep dari taksonomi didasarkan pada pengelompokan (*categorizing*) komponen-komponen konsep ke dalam struktur yang logis (Zach Wahl, 2006). Taksonomi tradisional biasanya menggunakan aturan-aturan pengelompokan yang detail, tidak boleh berulang dalam klasifikasinya, dan ketelitian (*granularity*) yang sangat tinggi. Tetapi karena perkembangan teknologi saat ini yang membuat *knowledge base* digunakan oleh jenis pengguna yang lebih luas, maka konsep taksonomi tradisional lebih sering gagal. Sebagai contoh staf umum dalam sebuah perusahaan cenderung tidak menemukan keuntungan penggunaan taksonomi. Taksonomi bisnis didesain lebih utamanya untuk klasifikasi, dan untuk *usability*. Taksonomi bisnis sering digunakan untuk klasifikasi metadata dari *knowledge*, navigasi bagi *user interface*, atau keduanya. Taksonomi bisnis yang baik harus didesain agar intuitif bagi seluruh jenis pengguna.

Aturan-aturan dalam taksonomi bisnis adalah:

1. Taksonomi menggunakan istilah-istilah yang sederhana. Hindari penggunaan jargon-jargon, atau istilah teknis yang kompleks. Desainer taksonomi harus menggunakan prinsip '*lowest common denominator*', yaitu bahwa setiap istilah harus dipahami oleh sebanyak mungkin pengguna yang paling umum. Ketika pengguna melihat level paling tinggi dari sebuah taksonomi pengguna harus langsung dapat memahami apa

yang sedang mereka lihat, dan dapat melanjutkan pencarian menuju konsep yang dicari.

2. Taksonomi tradisional menggunakan hirarki konsep yang sangat panjang. Seringkali terdapat 100 node pada level tertinggi, dan setiap node dapat memiliki sekitar 12 level hirarki. Taksonomi bisnis sebaiknya memiliki rata-rata 8 node pada level tertinggi, dan 3 level hirarki. Penggunaan *tree* yang lebih kecil dapat memastikan pengguna memahami keseluruhan konsep dan klasifikasi konsep dalam taksonomi tersebut.
3. Karakteristik lain adalah fleksibilitas. Karena kebutuhan sebuah bisnis selalu berubah, maka klasifikasi konsep harus mendukung evolusi taksonomi dalam jangka panjang sesuai umpan balik dari pengguna. Untuk mencapai hal ini diperlukan metode pengembangan taksonomi yang iteratif, yaitu evaluasi terus menerus dari taksonomi yang telah dibuat.

2.3. Solr

Solr adalah *platform* pencarian yang populer, sangat cepat, dan *open source* dari Apache Lucene (The Apache Software Foundation, 2011-2012). Fitur utamanya yaitu ketepatan dalam pencarian *full-text*, menemukan *highlight*, ragam, pengindeksan yang hampir *real-time*, pengelompokan yang dinamis, berintegrasi dengan *database*, kaya dokumen (misalnya, Word, PDF), dan pencarian secara geospasial. Solr sangat dapat diandalkan, terukur, dan kesalahannya dapat ditoleransi, menyediakan pengindeksan yang terdistribusi, replikasi, dan *query* yang ringan, *failover* dan *recovery* otomatis, konfigurasi terpusat, dan banyak lagi. Solr melakukan fitur pencarian dan navigasi dari banyak situs-situs internet terbesar di dunia.

2.4. PhoneGap

PhoneGap adalah sebuah kerangka kerja atau *framework open source* yang dipakai untuk membuat aplikasi *cross-platform mobile* dengan HTML5, CSS3, dan JavaScript. PhoneGap menjadi suatu solusi yang ideal untuk pengembang aplikasi *web* dalam pembuatan aplikasi di *smartphone*. PhoneGap juga merupakan solusi ideal untuk membuat sebuah aplikasi yang dapat berjalan pada beberapa perangkat *smartphone* dengan basis kode yang sama. Artinya, cukup hanya

dengan satu kali *coding* saja dapat membuat aplikasi untuk *smartphone* iPhone, Android, Blackberry, Symbian dan Windows Phone. PhoneGap juga tidak memerlukan *coding* secara terpisah untuk masing-masing *platform*. PhoneGap dapat membantu menghemat waktu dalam membuat aplikasi untuk beberapa *smartphone* dengan sekaligus dan hanya dibekali pengetahuan tentang HTML, CSS, dan JavaScript semua dapat menggunakan PhoneGap. Hal ini bisa disebut dengan *cross-platform* karena PhoneGap dapat membuat aplikasi pada beberapa *smartphone* dengan hanya satu *coding*.

Ilustrasi pada Gambar 2.2 merupakan contoh bahasa pemrograman PhoneGap untuk melakukan pengecekan terhadap tipe koneksi internet pada *mobile device*.

```
function checkConnection() {
  var networkState =
  navigator.connection.type;

  var states = {};
  states[Connection.UNKNOWN] =
  'Unknown connection';
  states[Connection.ETHERNET] =
  'Ethernet connection';
  states[Connection.WIFI] =
  'WiFi connection';
  states[Connection.CELL_2G] =
  'Cell 2G connection';
  states[Connection.CELL_3G] =
  'Cell 3G connection';
  states[Connection.CELL_4G] =
  'Cell 4G connection';
  states[Connection.CELL] =
  'Cell generic connection';
  states[Connection.NONE] = 'No
  network connection';

  alert('Connection type: ' +
  states[networkState]);
}

checkConnection();
```

Gambar 2.2. *Check Connection Function* di PhoneGap

Sumber: www.phonegap.com

2.5. SMS Gateway

SMS Gateway adalah suatu *platform* yang menyediakan mekanisme untuk menghantar dan menerima SMS dari peralatan *mobile* yang menggunakan *keyword* tertentu. SMS Gateway merupakan SMS interaktif yang memiliki

komunikasi dua arah. *SMS Gateway* juga bisa dimanfaatkan untuk keperluan lain seperti melakukan *polling*, transaksi perbankan, kuis berhadiah, dan lain-lain.

Mekanisme kerja *SMS Gateway* yaitu *SMS Gateway* menerima SMS dari *user* sesuai dengan format SMS (*keyword*) yang telah ditentukan sebelumnya kemudian akan melakukan validasi data serta melakukan pemrosesan terhadap data-data yang telah dikirimkan oleh *user*. Kemudian pada umumnya sistem akan mengirimkan informasi hasil dari pemrosesan data tersebut dalam bentuk format SMS berdasarkan *keyword* yang sesuai dengan permintaan *user* tersebut.

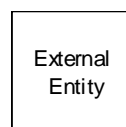
2.6. *Data Flow Diagram (DFD)*

Menurut Romney & Steinbart (2000), DFD merupakan skema yang menggambarkan jalannya alur data dalam suatu organisasi yang digunakan untuk mendokumentasikan sistem yang sudah ada dan untuk merencanakan serta mendesain sistem yang baru. DFD dapat dibagi menjadi beberapa level untuk menggambarkan kompleksitas pergerakan data secara lebih jelas.

Simbol-simbol DFD dapat digambarkan sebagai berikut:

- *External Entity*

Simbol ini digunakan untuk melambangkan orang atau organisasi yang memberi *input* data dan menerima data dari sistem. Bentuk simbolnya ini dapat dilihat pada Gambar 2.3.



Gambar 2.3. Simbol dari *External Entity*

- *Data Flow*

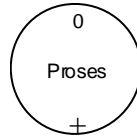
Simbol ini digunakan untuk melambangkan aliran perpindahan data masuk atau keluar dari suatu proses dan *entity*. Bentuk simbolnya ini dapat dilihat pada Gambar 2.4.



Gambar 2.4. Simbol dari *Data Flow*

- Proses

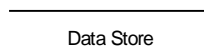
Simbol ini digunakan untuk melambangkan proses yang mentransformasi dari suatu *input* menjadi suatu *output*. Bentuk simbolnya ini dapat dilihat pada Gambar 2.5.



Gambar 2.5. Simbol dari Proses

- *Data Storage*

Simbol ini digunakan untuk melambangkan tempat untuk penyimpanan dan mengambil data. Bentuk simbolnya ini dapat dilihat pada Gambar 2.6.



Gambar 2.6. Simbol dari *Data Storage*

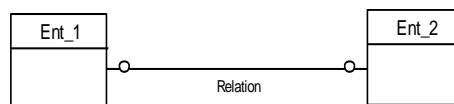
2.7. *Entity Relationship Diagram (ERD)*

Menurut Rommey & Steinbart (2000), ERD adalah diagram yang dipakai untuk mendokumentasikan skema *database* dan disebut sebagai *entity relationship* karena menggambarkan entitas dan hubungannya.

Relasi-relasi yang ada dalam ERD adalah sebagai berikut:

- *One to One Relationship*

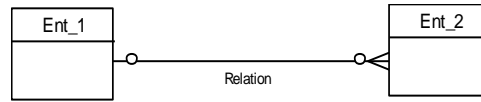
One to One Relationship (1:1) merupakan suatu model hubungan di mana satu anggota *entity* memiliki hubungan dengan satu anggota *entity* yang lain. Untuk lebih jelasnya dapat dilihat pada Gambar 2.7.



Gambar 2.7. *One to One Relationship*

- *One to Many Relationship*

One to Many Relationship (1:N) merupakan suatu relasi atau hubungan antara satu anggota *entity* dengan N anggota *entity* pada kelas yang berbeda. Untuk lebih jelasnya dapat dilihat pada Gambar 2.8.



Gambar 2.8. *One to Many Relationship*

- *Many to Many Relationship*

Many to Many Relationship (M:N) merupakan hubungan yang terjalin antara M anggota *entity* yang satu dengan N anggota *entity* yang lainnya. Kedua belah pihak dapat memiliki hubungan lebih dari satu dengan beberapa anggota *entity*. Untuk lebih jelasnya dapat dilihat pada Gambar 2.9.



Gambar 2.9. *Many to Many Relationship*