

INTEGRASI APLIKASI DAN INFORMASI FILOSOFI

Konsep Dan Penerapannya



Rika Yulliana



INTEGRASI APLIKASI DAN INFORMASI

Konsep Dan Penerapannya

Rika Yuliana



**INTEGRASI APLIKASI DAN INFORMASI
KONSEP DAN PENERAPANNYA**

Penulis:

Rika Yuliana

Desain Cover:

Fawwaz Abyan

Tata Letak:

Handarini Rohana

Editor:

N. Rismawati

ISBN:

978-623-459-162-0

Cetakan Pertama:

September, 2022

Hak Cipta 2022, Pada Penulis

Hak Cipta Dilindungi Oleh Undang-Undang

Copyright © 2022

by Penerbit Widina Bhakti Persada Bandung

All Right Reserved

Dilarang keras menerjemahkan, memfotokopi, atau memperbanyak sebagian atau seluruh isi buku ini tanpa izin tertulis dari Penerbit.

PENERBIT:

WIDINA BHAKTI PERSADA BANDUNG

(Grup CV. Widina Media Utama)

Komplek Puri Melia Asri Blok C3 No. 17 Desa Bojong Emas
Kec. Solokan Jeruk Kabupaten Bandung, Provinsi Jawa Barat

Anggota IKAPI No. 360/JBA/2020

Website: www.penerbitwidina.com

Instagram: [@penerbitwidina](https://www.instagram.com/penerbitwidina)

PRAKATA PENULIS

Alhamdulillah, puji syukur kehadiran Allah SWT yang hanya dengan hidayah-Nya sehingga buku referensi dengan judul Integrasi Aplikasi dan Informasi ini dapat diselesaikan dengan baik dan lancar. Pada dasarnya tujuan pembuatan buku referensi ini adalah untuk menjelaskan berbagai konsep beserta penerapan dari *integrase* aplikasi dan informasi secara umum dari perspektif akademisi. Adapun penjelasan dari isi/konten buku ini ditinjau dari berbagai hal terkait dalam *integrase* aplikasi dan informasi, baik berupa arsitektur maupun metode terkait dalam domain rekayasa informasi. Hal ini menjadi penting karena proses pengintegrasian aplikasi dan informasi ini melibatkan komponen-komponen yang memiliki karakteristik dan kegunaan yang spesifik sehingga diperlukan adanya cara yang tepat untuk menghindari terjadinya kesalahan dan kesenjangan dalam *system* informasi yang dihasilkan.

Buku referensi ini dapat digunakan oleh semua kalangan yang sedang membutuhkan petunjuk mengenai cara mengintegrasikan aplikasi dengan informasi, baik dari kalangan mahasiswa, *professional* maupun pihak lain yang terlibat baik secara langsung maupun tidak langsung dalam pekerjaan terkait. Adapun atas berbagai kekurangan yang masih ada dalam pengembangan buku ini, saya berharap berbagai masukan dan saran dari pembaca/pengguna buku ini agar bisa menjadi sumber rujukan yang dapat memberikan manfaat dan dampak yang lebih besar kepada masyarakat luas di masa yang akan datang.

Akhir kata, tak ada gading yang tak retak, dan karenanya tidak ada karya yang sempurna. Terlebih, kesempurnaan hanyalah milik Allah Swt semata, sang pemilik ilmu yang sesungguhnya. Tak lupa pula saya ucapkan terima kasih kepada semua pihak yang telah membantu dalam menyelesaikan buku referensi ini. Harapan saya, semoga buku ini bisa memberikan manfaat sebesar-besarnya bagi para pembaca dari berbagai kalangan. Selamat membaca!

Banda Aceh, September 2022

Penulis

DAFTAR ISI

PRAKATA PENULIS	iii
DAFTAR ISI	iv
BAB 1 PENDAHULUAN	1
BAB 2 BERBAGAI TIPE <i>SYSTEM INFORMASI ENTERPRISE</i>, EKONOMI DIGITAL DAN <i>HYPERCOMPETITIVE</i>	15
A. <i>Enterprise Information System</i>	15
B. <i>Ekonomi Digital</i>	25
C. <i>Hypercompetitive</i>	29
BAB 3 PERAN, STRATEGI, DAN TRANSFORMASI BISNIS SERTA PERKEMBANGAN TEKNOLOGI DIGITAL	37
A. <i>Peran Bisnis Digital</i>	37
B. <i>Strategi Bisnis Digital</i>	38
C. <i>Transformasi Bisnis</i>	45
D. <i>Perkembangan Teknologi Digital</i>	48
BAB 4 MODEL DATA	55
BAB 5 INTEGRASI DATA	63
BAB 6 KUALITAS INFORMASI	69
A. <i>Kualitas Informasi</i>	71
B. <i>Nilai Informasi</i>	72
BAB 7 <i>DATA-LEVEL ENTERPRISE INTEGRATION</i>	73
A. <i>Contoh Integrasi pada Level Database</i>	76
BAB 8 <i>APPLICATION-LEVEL ENTERPRISE INTEGRATION</i>	81
A. <i>Integration Typologies</i>	89
B. <i>Classifying The Integration Problem Types</i>	90
BAB 9 <i>METHOD-LEVEL ENTERPRISE INTEGRATION</i>	95
BAB 10 <i>INTEROPERABILITY AND ITS APPLICATIONS IN ENTERPRISE</i>	101
A. <i>Inter-Layer</i>	105
B. <i>Inter-Fw</i>	109
C. <i>Inter-Meth</i>	110
BAB 11 PENUTUP	113
A. <i>Kesimpulan</i>	113
DAFTAR PUSTAKA	114
INDEKS	116
GLOSARIUM	117
PROFIL PENULIS	118



BAB
1

PENDAHULUAN

Dilatarbelakangi oleh berbagai keuntungan yang diperoleh oleh pengguna dari sebuah sistem informasi yang terintegrasi, saat ini mulai banyak organisasi mulai berlomba-lomba dalam mengembangkan sistem informasi *enterprise* yang terintegrasi dengan unit/pihak lain yang terkait karena didorong oleh pengguna tersebut. Hal ini bisa terjadi dalam berbagai sektor organisasi, baik *public* maupun *private*. Keuntungan dari integrasi sistem ini adalah membaiknya suatu arus informasi dalam sebuah organisasi. Keuntungan ini merupakan alasan yang kuat untuk mengutamakan (menggunggulkan) sistem informasi terintegrasi karena tujuan utama dari sistem informasi adalah memberikan informasi yang benar pada saat yang tepat kepada pengguna. Disamping itu, *business driver* untuk melakukan integrasi adalah meningkatkan efisiensi, meningkatkan kepuasan pelanggan, dan perkembangan organisasi.

Sistem terintegrasi (*integrated system*) merupakan sebuah rangkaian proses untuk menghubungkan beberapa sistem-sistem komputerisasi dan *software* aplikasi baik secara fisik maupun secara fungsional. Sistem terintegrasi akan menggabungkan komponen sub-sub sistem ke dalam satu sistem dan menjamin fungsi-fungsi dari sub sistem tersebut sebagai satu kesatuan sistem. Integrasi informasi dari sebuah sistem diperlukan karena:

1. Adanya kebutuhan konstituen untuk bekerja sama antar bagian dalam suatu korporasi.
2. Terjadinya pengolahan data antar sistem informasi tiap bagian yang saling terkait, sehingga untuk melengkapi suatu informasi dibutuhkan proses pertukaran data dengan sistem informasi yang lain.
3. Dapat memungkinkan penyediaan *realtime* pengaksesan data.

4. Mengubah data untuk analisis bisnis dan pertukaran data, mengatur penempatan data untuk kinerja, mata uang dan ketersediaan. [1]

Untuk Membangun Sistem informasi terintegrasi ada beberapa metode pilihan yang dapat kita gunakan, diantaranya adalah:

1. **Vertical Integration**, merupakan proses mengintegrasikan sub-sub sistem berdasarkan fungsionalitas dengan menghubungkan sub-sub sistem yang sudah ada tersebut supaya bisa berinteraksi dengan sistem terpusat dengan tetap berpijak pada arsitektur sub sistem yang lama.
2. **Star Integration**, atau lebih dikenal sebagai *spaghetti integration*, adalah proses mengintegrasikan sistem dengan cara menghubungkan satu sub sistem ke semua sub-sub sistem lainnya.
3. **Horizontal Integration**, atau ada yang mengistilahkan dengan *Enterprise Service Bus (ESB)*, merupakan sebuah metode yang mengintegrasikan sistem dengan cara membuat suatu *layer* khusus yang berfungsi sebagai interpreter, dimana semua sub-sub sistem yang sudah ada akan berkomunikasi ke *layer* tersebut. Model ini lebih menawarkan fleksibilitas dan menghemat biaya integrasi, karena yang perlu difokuskan dalam implementasi proses pengintegrasian hanya *layer* interpreter tersebut. Untuk menangani ekspansi proses kerja juga hanya perlu diimplementasikan di *layer* interpreter itu juga, dan sub sistem baru yang akan menangani *interface* dari proses bisnis ekstensi tersebut akan berkomunikasi langsung ke *layer* dan *layer* akan menyediakan keperluan-keperluan data/*interface* untuk sub sistem lain yang memerlukannya. Metode *Enterprise Service Bus (ESB)* ini memiliki banyak kelebihan jika diadopsi dalam merancang arsitektur sistem terintegrasi, yaitu antara lain:
 - a. Lebih cepat dalam melakukan penyesuaian dengan sistem yang telah ada
 - b. Meningkatkan fleksibilitas, mudah untuk diperbaharui mengikuti perubahan keperluan sistem (*system requirements*)
 - c. Membuat standar sistem sehingga bisa diaplikasikan di sub sistem mana pun
 - d. Porsi pekerjaan *software development* lebih banyak di “konfigurasi” daripada “menulis *code*” untuk integrasi
 - e. Dapat diterapkan mulai ruang lingkup kecil hingga di level *enterprise*

Namun metode *horizontal integration* atau *Enterprise System Bus (ESB)* yang tampaknya ideal ini bukan berarti tidak ada kelemahan. Beberapa kelemahan yang cukup signifikan pengaruhnya antara lain:

1. Pembuatan standar sistem dalam *Enterprise Message Model* banyak berkulat di aspek analisis dan manajerial, biaya analisis benar-benar tinggi karena perlu berkolaborasi dengan analis-analis yang bertanggung jawab terhadap arsitektur dan desain sistem-sistem yang telah ada.
2. Secara khusus memerlukan perangkat keras (*hardware*) yang spesifik, seperti misalnya *business-logic-server* yang independen dan tidak integral dengan salah satu atau sebagian dari sub sistem yang telah ada.
3. Perlu tambahan tenaga (SDM) berupa *Middleware Analyst* yang akan mengkonfigurasi, merawat, dan mengoperasikan *layer Enterprise Service Bus*.
4. Karena biasanya ESB mempergunakan XML sebagai bahasa komunikasi antar sistem, tentu akan memerlukan *resources* dan komputasi berlebih untuk melakukan *parsing-reparsing* dalam komunikasi data.
5. Memerlukan *effort* yang cukup tinggi dalam mengimplementasikan ESB karena cukup banyak *layer/tingkatan* aplikasi yang harus ditangani, tidak hanya aplikasi-aplikasi *interface* dari sub-sub sistem saja, melainkan juga *layer* interpreter yang juga memiliki karakteristik sebagai aplikasi juga.

Ada dua Pendekatan Integrasi dalam Sistem Informasi:

1. Pendekatan Total & Homogen
 - Melakukan integrasi di semua aspek bisnis dengan suatu kerangka/*framework standart* & dilakukan serentak di setiap bidang.
 - Komponen yang homogen diharapkan mempermudah proses integrasi
 - Contohnya: Implementasi *product vendor* ERP seperti SAP, *Oracle Application/Peoplesoft, Sage Group, IBM Websphere* dll
 - Mahal & Implementasi membutuhkan waktu yang panjang (tergantung kematangan TI suatu organisasi)
2. Pendekatan Bertahap
 - Mulai dari bawah & memanfaatkan sistem informasi *existing*
 - Sistem informasi-sistem informasi dirangkai mengikuti pola integrasi dan kebutuhan informasi akan datang
 - Butuh waktu yang lama dan konsisten agar tidak gagal
 - Relatif lebih murah
 - Butuh strategi khusus (*Non Teknis Termasuk Political Will* dari Pimpinan)

Sistem informasi terintegrasi atau biasa juga dikenal dengan istilah *Enterprise Application Integration* atau disebut EAI, merupakan proses mengintegrasikan/menghubungkan aplikasi dalam satu organisasi bersama dalam rangka untuk menyederhanakan dan mengotomasi proses bisnis seluas mungkin, sementara pada saat yang sama menghindari keharusan membuat perubahan besar terhadap aplikasi yang ada. Dengan kata lain, *Enterprise Application Integration* adalah penggunaan perangkat lunak dan prinsip-prinsip arsitektur sistem komputer untuk mengintegrasikan sekumpulan aplikasi komputer. Grup Gartner mengatakan "EAI adalah berbagi data tak terbatas dan menghubungkan aplikasi proses bisnis dalam suatu perusahaan". Salah satu tantangan besar EAI adalah adanya perbedaan kondisi lingkungan dari sistem yang perlu dihubungkan bersama. Perbedaan tersebut mencakup: 1) sistem operasi, 2) solusi *database*, 3) bahasa pemrograman yang digunakan. Dalam beberapa kasus adalah sistem warisan (lama) yang tidak lagi didukung oleh vendor pembuatnya [6]. Jika integrasi diterapkan tanpa mengikuti pendekatan EAI, koneksi *point-to-point* (p2p) tumbuh di seluruh organisasi.

Ketergantungan akan bertambah secara spontan, hingga berantakan dan kusut. Ini biasanya disebut sebagai *spaghetti*, dalam pemrograman sama dengan "*spaghetti code*". Sebagai contoh, banyaknya koneksi yang dibutuhkan untuk menghubungkan n titik secara penuh adalah $n(n-1)/2$. Dengan demikian, jika ada 10 aplikasi yang diintegrasikan dengan pendekatan p2p maka diperlukan 45 koneksi. Ada beberapa pola penerapan EAI, diantaranya:

- 1) Mediasi. Pada pola mediasi, sistem EAI bertindak sebagai perantara (*interface*) beberapa aplikasi. Setiap kali sebuah peristiwa penting terjadi dalam aplikasi (misalnya: informasi baru dibuat, transaksi baru selesai, atau kejadian lainnya) modul integrasi dalam sistem EAI diberitahu. Modul kemudian menyebarkan perubahan ke aplikasi lain yang relevan.
- 2) Federasi. Pada pola federasi, sistem EAI bertindak sebagai *fasad* menyeluruh di beberapa aplikasi. Semua *event* dari eksternal ke salah satu aplikasi akan diterima oleh *front-end* sistem EAI. Sistem EAI dikonfigurasi untuk mengekspos hanya informasi yang relevan dan *interface* dari aplikasi eksternal, dan melakukan semua interaksi dengan aplikasi atas nama aplikasi eksternal tersebut.

ENTERPRISE INTEGRATION

Ruang lingkup struktur suatu perusahaan (yang dapat dijelaskan oleh EA) mencakup komponen perusahaan utama seperti tujuan dan sasaran perusahaan, struktur organisasi, proses bisnis, dan infrastruktur informasi.

Karena infrastruktur informasi berada dalam ruang lingkup EA, istilah "perusahaan" seperti yang digunakan dalam EA umumnya berkaitan dengan arsitektur informasi dalam organisasi industri. EA sangat relevan dengan integrasi informasi industri, karena integrasi informasi industri berkaitan dengan aliran informasi di dalam organisasi industri secara keseluruhan.

Metode integrasi utama dapat diringkas dalam hal cakupan integrasi, perspektif integrasi, lapisan integrasi, dan tingkat integrasi.

Lingkup integrasi: Ini terutama membedakan pendekatan integrasi ke dalam dan antar organisasi, seperti yang disebutkan sebelumnya. Integrasi intra organisasi selanjutnya dibagi menjadi dua subkategori: integrasi horizontal dan integrasi vertikal. "Integrasi horizontal" bermaksud untuk memotivasi kerja sama unit-unit dalam perusahaan untuk efektivitas operasi, sementara "integrasi vertikal" berfokus pada hubungan efektif antara struktur hulu dan hilir. Dengan kata lain, integrasi horizontal terutama menekankan pada kemungkinan unit yang berbeda untuk bekerja sama dalam tugas yang sama secara bersamaan, dan integrasi vertikal berfokus pada koordinasi hierarki untuk secara efektif melaksanakan tugas yang menjadi tanggung jawab mereka bersama.

Dalam lingkup intra-organisasi, integrasi horizontal mengacu pada integrasi hubungan lintas fungsi atau horizontal dalam organisasi. Tujuannya adalah untuk memungkinkan kerja sama lintas fungsi melalui integrasi data, fungsionalitas, dan proses berdasarkan proses bisnis inti organisasi, untuk mengurangi efisiensi rendah yang disebabkan oleh fungsionalitas dan proses yang dirancang secara tidak tepat dan untuk meningkatkan efektivitas dan efisiensi. Dalam proses ini, integrasi data merupakan langkah penting dalam integrasi horizontal karena memungkinkan pemrosesan transaksi terintegrasi, yang dapat mengurangi inkonsistensi data dan meningkatkan efisiensi pemrosesan transaksi. Tingkat integrasi hubungan horizontal yang lebih tinggi adalah tujuan dari generasi awal solusi perusahaan. Integrasi horizontal menargetkan integrasi fungsi dan komponen yang umumnya dianggap sebagai tugas utama solusi perusahaan, termasuk aplikasi terkait yang telah dikembangkan. Contoh khas dari aplikasi horizontal adalah solusi perusahaan yang dicontohkan oleh *Enterprise Resource Planning*, di mana sebuah organisasi mencoba untuk mengoptimalkan seluruh operasi oleh departemen yang berbeda secara horizontal dalam organisasi. Integrasi vertikal bertujuan agar integrasi dapat diterapkan pada berbagai tingkatan organisasi, guna mengupayakan proses bisnis yang lebih efisien dan efektif melalui berbagai tingkatan dalam hierarki organisasi. Dengan kata lain, integrasi vertikal mencoba mengintegrasikan data, fungsionalitas, dan proses yang diimplementasikan pada berbagai tingkat organisasi dengan cara yang

inovatif. Salah satu tujuan integrasi vertikal adalah untuk mengembangkan alternatif operasional melalui perancangan ulang fungsi dan proses bisnis pada tingkat yang berbeda. Dari perspektif sistem, secara umum, tidak diinginkan untuk mempertimbangkan integrasi horizontal dan integrasi vertikal secara terpisah. Untuk efisiensi operasional yang lebih baik, integrasi horizontal dan integrasi vertikal tidak dapat dipisahkan dan harus didekati secara sistematis dan holistik untuk mencapai kinerja yang lebih baik, karena integrasi horizontal dan vertikal saling melengkapi satu sama lain dari sudut pandang sistem. Mendekati posisi relatif integrasi horizontal dan vertikal secara sistematis akan mendorong perusahaan untuk mencapai kinerja yang lebih baik, karena kompleksitas sistem yang terlibat.

"Integrasi antar organisasi" terutama mewakili integrasi aplikasi dari organisasi yang berbeda. Ini melibatkan aplikasi dalam lingkungan e-bisnis termasuk B2B dan B2C. Ini terkait dengan lapisan ketiga dalam penelitian alur kerja. Konsep rantai pasok berkaitan erat dengan integrasi antar organisasi. Integrasi rantai suplai tidak hanya memperluas cakupan integrasi intra organisasi tradisional tetapi juga memperluas cakupan integrasi antar organisasi untuk membentuk konsep rantai suplai terintegrasi. Model B2B adalah contoh rantai pasokan terintegrasi, karena B2B memfasilitasi kolaborasi di perusahaan yang diperluas dengan memungkinkan integrasi sistem di luar tembok perusahaan dari satu perusahaan. Diperkirakan bahwa dengan pengembangan teknologi EAI antar-organisasi, solusi perusahaan, dan rantai pasokan, rantai pasokan secara "keseluruhan" pada akhirnya dapat diintegrasikan bersama untuk membentuk platform e-bisnis kolaboratif generasi baru.

Perspektif integrasi: Karena integrasi berfokus pada pemodelan dan otomatisasi proses kompleks di lingkungan yang heterogen, dan karena integrasi yang efektif bergantung pada metode integrasi kualitas, selama dekade terakhir, upaya signifikan telah dilakukan dalam pengembangan metode integrasi. Upaya tersebut telah dilakukan berdasarkan perspektif pemodelan yang berbeda; pada kenyataannya, beberapa menekankan satu pandangan di atas yang lain. Setiap aspek integrasi dapat dimodelkan dari perspektif yang berbeda. Ini adalah beberapa perspektif dari mana integrasi dapat didekati:

1. Perspektif konseptual: Perspektif ini menekankan pandangan desainer tentang integrasi pemodelan.
2. Perspektif eksternal: Perspektif ini berfokus pada pandangan pengguna tentang integrasi (yang terkait erat dengan perspektif konseptual).
3. Perspektif pemrograman dan implementasi: Perspektif ini berfokus pada aspek pemrograman dan implementasi.

4. Perspektif sistem: Perspektif ini menganggap EAI sebagai sistem yang kompleks.

Lapisan integrasi: Seperti disebutkan sebelumnya, integrasi dapat dilihat dari segi lapisan: lapisan integrasi data, lapisan integrasi proses bisnis, lapisan integrasi aplikasi, lapisan integrasi platform, serta lapisan integrasi standar. Lapisan model bisnis dapat didekomposisi lebih lanjut menjadi lapisan antarmuka aplikasi, lapisan metode, dan lapisan antarmuka pengguna. Integrasi dapat terjadi pada tiga atau lebih lapisan. Lapisan pertama berkaitan dengan "integrasi bisnis," yang mewakili koordinasi dan orkestrasi proses bisnis dan menghubungkan aktivitas di antara banyak unit berbeda dalam suatu organisasi. Metodologi perwakilan termasuk CIMOSA dan lain-lain. Tujuan awal CIMOSA adalah untuk menyediakan kerangka kerja konseptual untuk pemodelan perusahaan di mana "arsitektur generik" berlaku untuk semua perusahaan manufaktur, "arsitektur parsial" berlaku untuk sektor industri tertentu, dan "arsitektur khusus" memodelkan struktur perusahaan tertentu. Lapisan kedua sesuai dengan "integrasi data dan aplikasi," yang berfokus pada interoperabilitas aplikasi pada platform heterogen termasuk berbagi data dan mengakses layanan aplikasi jarak jauh. Ini melibatkan konsep pemodelan data terpadu. Alat yang digunakan dalam proses integrasi dan orkestrasi termasuk bahasa seperti HTML dan XML, dan platform seperti CORBA dan OMG, selain WfMS. Lapisan ketiga menyangkut "aplikasi sistem fisik" seperti integrasi sistem operasi dan komunikasi. Metode dan alat yang representatif mencakup standar ISO/OSI yang relevan dan protokol khusus seperti SOAP.

Tingkat integrasi: Integrasi seharusnya terjadi pada beberapa tingkatan, seperti integrasi proses bisnis, integrasi aplikasi semantik, integrasi aplikasi sintaksis, integrasi fisik, dan juga integrasi intra dan antar organisasi. Dengan demikian, untuk membangun sistem terintegrasi seperti solusi perusahaan yang mampu menyediakan fungsionalitas yang diperlukan, ketidaksesuaian tersebut (ketidaksesuaian antara jaringan komunikasi dan protokol yang berbeda yang ada pada tingkat fisik, dan ketidaksesuaian antara data dan perangkat lunak, dll., yang ada di berbagai tingkatan) harus diatasi. Sebagai contoh, meskipun beberapa aplikasi yang ada menggunakan XML, ada ketidakcocokan di kedua model data dan skema. Klasifikasi lain dari pendekatan integrasi didasarkan pada integrasi tingkat semantik, sintaksis, perangkat keras, dan platform. "Tingkat semantik" mengacu pada pertukaran data di mana ada representasi eksplisit dari makna data. "Tingkat sintaksis" mengacu pada pertukaran data tanpa memperhatikan representasi eksplisit dari maknanya. Ini termasuk melewati parameter, akses data eksternal, dan

mekanisme waktu. Integrasi semantik dapat membantu mengintegrasikan aplikasi yang lebih kuat dan berkualitas lebih baik. Tingkat perangkat keras melibatkan perbedaan dalam perangkat keras, jaringan, dll. Tingkat platform melibatkan perbedaan dalam platform basis data, sistem operasi, dll.

Berhasil tidaknya suatu proyek sistem terintegrasi, bergantung kepada seberapa dalam pengembang proyek menguasai teori dan metode *Enterprise Architectur Integration* (EAI). Istilah EAI secara umum digunakan untuk mendeskripsikan semua metode yang digunakan untuk menghubungkan berbagai sistem yang berbeda, agar tidak terjadi “*spaghetti architecture*” akibat dari monopoli API tunggal dari suatu sistem. Kesimpulannya, EAI adalah solusi bagi siapa saja yang ingin agar berbagai sistem (*software*) dalam organisasinya menjadi saling terintegrasi. Menurut Linthicum (1999), terdapat 4 Level yang bisa digunakan untuk mengimplementasikan EAI, yaitu

- 1) *Data*,
- 2) *Application Interface*,
- 3) *Method* dan
- 4) *User Interface*

Berbagai sistem yang akan dihubungkan sering berada pada kondisi:

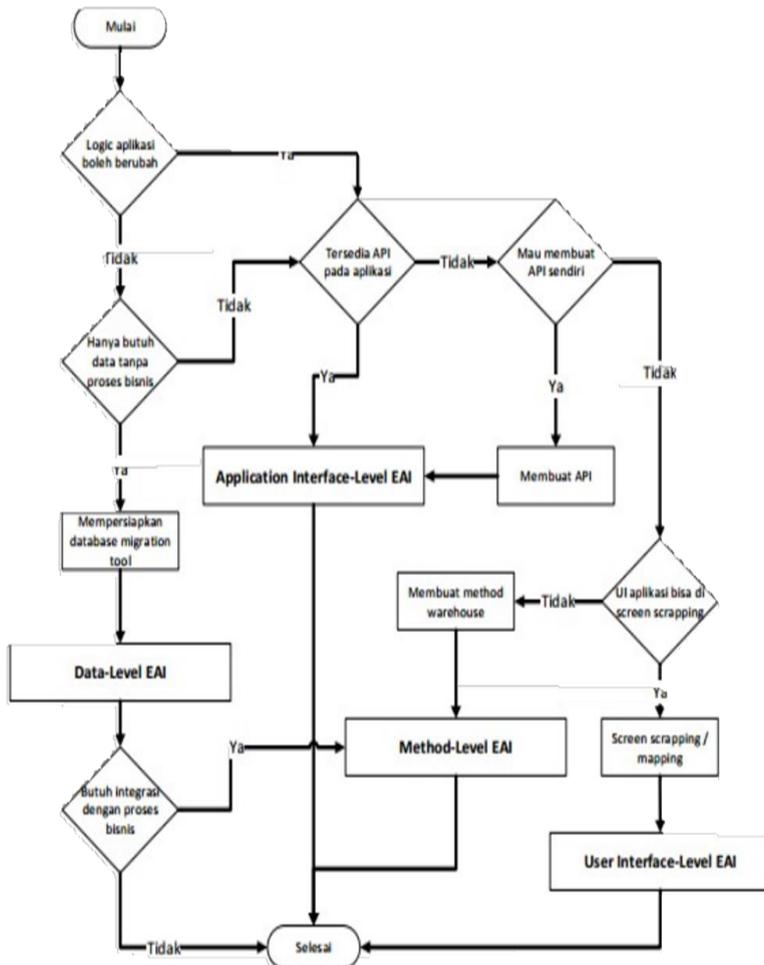
- Memiliki berbagai *Operating System* berbeda;
- Menggunakan berbagai bahasa komputer;
- *Legacy system* yang tidak didukung oleh *vendor* yang membuat. Contoh: program *Clipper* yang sudah tidak ada;
- *Stove pipe System*.

Oleh karena itu, penting bagi organisasi untuk melakukan berbagai tahapan dalam mempersiapkan Integrasi, yaitu:

- Harus memahami proses bisnis;
- Harus memahami data;
- Menentukan proses dan data yang perlu diintegrasikan.

Contohnya dalam Integrasi aplikasi, Integrasi dapat dilakukan melalui antar muka aplikasi atau melalui *method* sehingga integrasi dimungkinkan melalui *sharing business logic* (*method* di *shared* dengan meletakkannya pada sebuah *server* pusat (*application server*) atau dengan mengakses *method method* antar aplikasi (*distributed object*).

Untuk lebih jelasnya, perhatikan diagram penentuan level EAI pada Gambar 1.1 dibawah.



Gambar 1.1 Penentuan Level EAI

Berikut penjelasan Tabel 1.1 menurut masing-masing Level:

Tabel 1.1 Ringkasan *Data-Level EAI*

<i>Data-Level EAI</i>	
Kelebihan	<ol style="list-style-type: none"> 1. Sudah tersedia banyak <i>tool</i> untuk membantu integrasi data antar <i>database</i> 2. Akses informasi <i>database</i> tidak perlu mengubah struktur <i>database</i> itu sendiri maupun sisi <i>logic (source code)</i> aplikasi 3. Sisi <i>logic</i> aplikasi hampir tidak perlu diubah, oleh karena itu perusahaan tidak perlu khawatir saat harus melakukan <i>testing</i> ulang pada aplikasi

Kekurangan	<ol style="list-style-type: none"> 1. <i>Database</i> yang ada di perusahaan bisa sangat beragam, jumlah tabel sangat banyak dan dengan banyak pula <i>constraint</i>, <i>view</i>, <i>trigger</i>, <i>function</i> dan <i>procedure</i> yang harus dipahami pengembang 2. Pengembang haru menentukan frekuensi data, apakah <i>real-time</i>, <i>in-time</i> atau <i>one-time</i> 3. <i>Data-Level</i> EAI hanya menyediakan <i>integrasi</i> dalam bentuk data saja, tanpa ada proses bisnis
Kondisi yang harus dipenuhi	<ol style="list-style-type: none"> 1. <i>Data-Level</i> EAI baru dapat dijadikan solusi jika mengubah <i>logic/source code</i> aplikasi tidak dapat dilakukan 2. <i>Data-Level</i> EAI baru dapat terwujud jika pengembang sudah memahami teknologi <i>database</i> yang digunakan, kamus data seluruh tabel beserta <i>constraint</i>-nya, beserta aliran informasi dalam perusahaan.
Keterangan lain	Pada prakteknya, sangat sulit mengintegrasikan sistem jika hanya bermodal data saja. Integrasi sistem membutuhkan data dan proses bisnis (<i>application logic</i>). Itu sebabnya solusi <i>data-level</i> EAI ini biasanya diimplementasikan bersama dengan <i>method-level</i> EAI.
Produk/ Hasil Akhir	Sebuah <i>middle-tier</i> berupa <i>database migration software</i> , yang berfungsi untuk mengekstrak, memformat, dan meng- <i>update</i> data dari satu <i>database</i> aplikasi ke <i>database</i> aplikasi lain.

Tabel 1.2 Ringkasan *Application Interface-Level* EAI

<i>Application Interface-Level</i> EAI	
Kelebihan	<ol style="list-style-type: none"> 1. Saat ini, sudah banyak aplikasi di level <i>enterprise</i> yang sudah siap berbagi data dan proses bisnis dengan cara menyediakan API. 2. Jika aplikasi tidak menyediakan API, pengembang tetap masih dimudahkan dengan banyaknya <i>framework</i> ataupun <i>tool</i> untuk membuat API.
Kekurangan	<ol style="list-style-type: none"> 1. <i>Vendor</i> aplikasi harus sadar pentingnya menyediakan API dalam aplikasi buatannya. 2. Jika aplikasi tidak menyediakan API, maka pengembang harus membuat API tersebut.

Kondisi yang harus dipenuhi	<ol style="list-style-type: none"> 1. Saat <i>data-level</i> EAI tidak bisa menjadi solusi (karena <i>database</i> terlalu kompleks dan integrasi sistem membutuhkan informasi tidak hanya data, tetapi juga proses bisnis), maka <i>application interface-level</i> EAI ini bisa menjadi solusi alternatif 2. Informasi yang dihasilkan dari pemanggilan API harus diolah oleh sebuah <i>middleware</i> (bisa berupa <i>message broker</i>, <i>message queuing</i>, atau <i>application server</i>) untuk kemudian ditransfer ke sistem tujuan.
Keterangan lain	-
Produk/ Hasil Akhir	Sistem/aplikasi sumber (<i>source system</i>) menghasilkan informasi yang diminta oleh sistem/aplikasi tujuan (<i>target system</i>) dengan cara mengonsumsi API yang disediakan. Informasi yang dihasilkan akan diantri dan ditransfer via <i>middleware</i> .

Tabel 1.3 Ringkasan *Method-Level* EAI

<i>Method-Level</i> EAI	
Kelebihan	<ol style="list-style-type: none"> 1. Mengintegrasikan sistem berdasarkan pada <i>method</i> (proses bisnis) aplikasi tidak hanya menciptakan kumpulan <i>method</i> yang siap untuk di-<i>share</i>, tapi juga menciptakan infrastruktur yang lebih handal. 2. Produk akhir dari <i>method-level</i> EAI ini (berupa <i>method warehouse</i>) adalah wadah yang tepat bagi <i>method</i> aplikasi yang tidak hanya <i>reuse</i>, tetapi juga siap untuk di-<i>share</i>, di suatu <i>server</i> yang tersentralisasi.
Kekurangan	<ol style="list-style-type: none"> 1. Pengembangan <i>method</i> yang <i>reuse</i>, siap di-<i>share</i>, dan tersentralisasi ini membutuhkan <i>re-develop</i>, <i>re-test</i>, dan <i>re-deployment</i>. Dimana kesemua hal tersebut membutuhkan waktu yang lama dan biaya yang besar. 2. Solusi <i>method-level</i> EAI ini juga termasuk yang paling sering gagal. Alasan gagalnya pun beragam, mulai dari infrastruktur yang belum siap sampai konflik kepentingan internal.

Kondisi yang harus dipenuhi	<ol style="list-style-type: none"> 1. Untuk mensukseskan solusi ini, semua aplikasi yang ada harus terikat (saling terhubung) agar proses <i>sharing</i> proses bisnis berjalan lancar. 2. <i>Method</i> yang sudah jadi harus disimpan dalam sebuah <i>distributed objects</i>, seperti CORBA atau COM.
Keterangan lain	-
Produk/ Hasil Akhir	<i>Method Warehouse</i> , adalah wadah tersentralisasi bagi <i>method</i> yang <i>reuse</i> dan siap untuk di <i>share</i> oleh aplikasi yang membutuhkan.

Tabel 1.4 User Interface-Level EAI

User Interface-Level EAI	
Kelebihan	<ol style="list-style-type: none"> 1. Solusi ini menjembatani apa yang tidak ada pada solusi-solusi EAI di atas, yaitu mengintegrasikan sistem dengan menyediakan sebuah <i>user interface</i> (UI). 2. Selama proses implementasinya, solusi ini sama sekali tidak membutuhkan perubahan <i>logic</i> (<i>source code</i>) aplikasi. 3. Sudah tersedia <i>tool</i> untuk melakukan proses <i>screen scrapping</i>.
Kekurangan	Pengembang harus mengetahui secara detil kegunaan tiap UI yang ada, aliran <i>input</i> dan <i>output</i> data yang dihasilkan oleh UI yang bersangkutan, termasuk juga aliran informasi dan transformasinya (<i>raw data</i> atau diolah terlebih dahulu) yang masuk ke <i>database</i> .
Kondisi yang harus dipenuhi	<ol style="list-style-type: none"> 1. Jika perusahaan tidak punya waktu dan biaya yang cukup, maka UI-level EAI ini bisa menjadi solusi untuk <i>Application-level</i> EAI. 2. Solusi ini pun bisa menjadi alternatif jika memahami <i>database</i> aplikasi adalah hal yang hampir mustahil, yang menyebabkan data-level EAI tidak bisa dilakukan. 3. Solusi ini adalah solusi yang paling konvensional, sehingga harus menjadi opsi terakhir dalam EAI.

Keterangan lain	Sekilas memang <i>UI-level EAI</i> mirip dengan <i>Application Interface-level EAI</i> , namun <i>UI-level EAI</i> mengolah proses bisnis dengan berfokus pada UI yang ada, bukan pada <i>interface</i> aplikasi ataupun <i>database</i>
Produk/ Hasil Akhir	<i>Screen catalog</i> , adalah wadah untuk hasil dari proses <i>screen scrapping</i>



BAB
2

BERBAGAI TIPE SYSTEM INFORMASI ENTERPRISE, EKONOMI DIGITAL DAN *HYPERCOMPETITIVE*

A. ENTERPRISE INFORMATION SYSTEM

- ⦿ *Enterprise* adalah sebuah sistem dari manusia, peralatan, material, data, kebijakan dan prosedur yang muncul untuk menyediakan sebuah produk atau pelayanan, dengan tujuan mendapatkan keuntungan.
- ⦿ Sistem Informasi *Enterprise* merupakan pengembangan dari konsep yang sudah ada yaitu *Executive Information system* dan DSS yang diperluas untuk domain seluruh perusahaan.
- ⦿ Sistem *enterprise* mendukung struktur organisasi yang sebelumnya tidak mungkin untuk menciptakan budaya organisasi yang lebih disiplin.

Tipe-tipe system Informasi secara umum

⦿ ***Transaction Processing System (TPS)***

Sistem yang digunakan untuk menyimpan data rutin harian yang akan menjadi sumber data penting bagi tipe sistem-sistem informasi yang lainnya, berfokus pada penyimpanan data rutin (transaksi). TPS/Sistem Pengolahan data adalah aplikasi sistem informasi yang menangkap dan memproses data tentang (atau untuk) transaksi bisnis dengan cara:

1) *Batch Processing*

Data transaksi dikumpulkan terlebih dahulu selama periode tertentu baru kemudian diproses. Contoh: Bank memproses seluruh cek yang diterima dalam *batch* dilakukan pada malam hari.

2) *Realtime processing*

Data transaksi langsung diproses seketika setelah transaksi terjadi. Contoh: Bank memproses penarikan ATM dengan segera.

Contoh

Sistem Penjualan dan Pemasaran

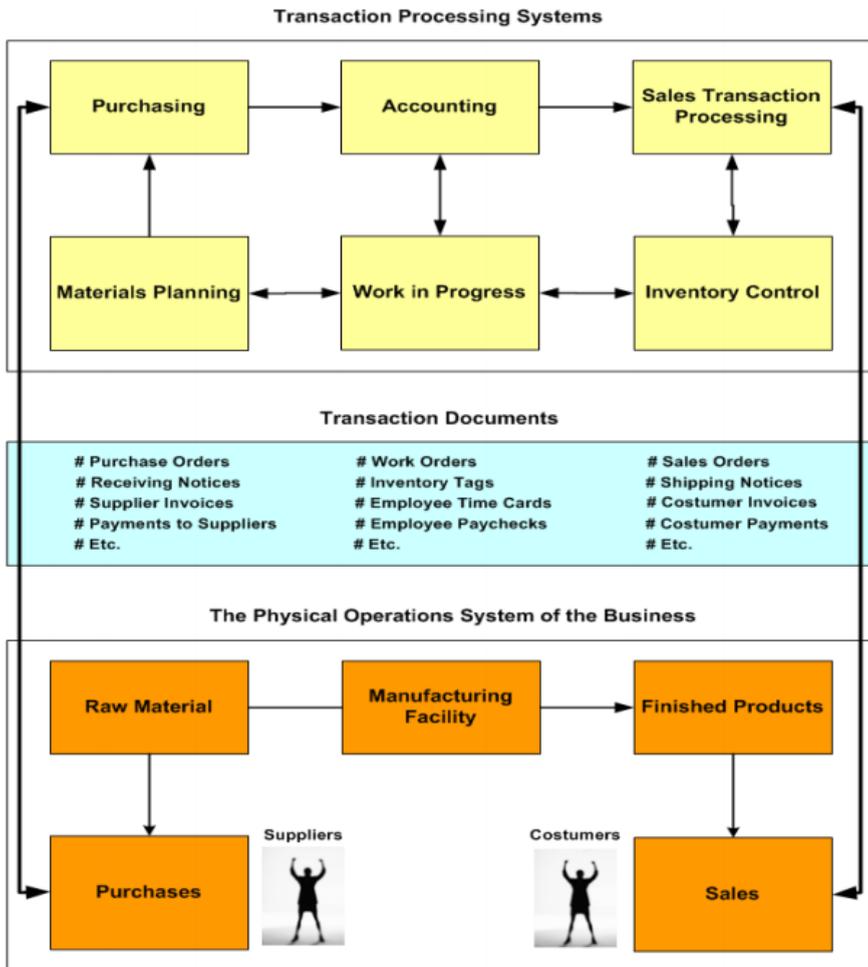
- Aplikasi-aplikasi utama dari sistem ini antara lain Sistem Info Order Penjualan; Sistem Riset Pemasaran; Sistem Penetapan Harga.

Sistem Produksi dan Pabrikasi:

- Aplikasi-aplikasi utama dari sistem ini adalah Sistem Perencanaan Sumber Daya Material; Sistem Kontrol Pesanan Pembelian; Sistem Perekayasaan; Sistem Pengendalian Mutu

Sistem Informasi Akademik

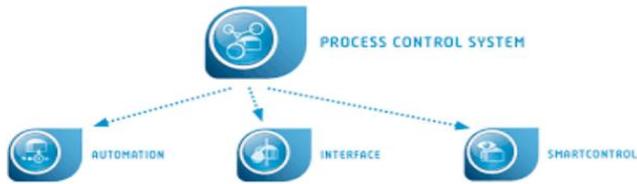
- Aplikasi-aplikasi utama dari sistem ini adalah Sistem Pendaftaran; Sistem pemasukan Kartu Rencana Studi; Sistem Pelaporan Hasil Studi, dan lain-lain



Gambar 2.2 Operasi fisik bisnis dan TPS

© **Process Control Systems (PCS)**

PCS adalah tipe sistem informasi yang melakukan pengambilan keputusan rutin untuk memantau dan mengontrol proses-proses operasi. Keputusan rutin ini dilakukan secara otomatis oleh *computer* tanpa menunggu pertimbangan dari pengguna. Contoh: sensor elektronik yang dihubungkan dengan perangkat *computer* pada proses-proses pengolahan. Contoh lain adalah pada mesin ATM, proses pengecekan PIN akan dilakukan otomatis oleh perangkat *computer* ATM.



☉ **Enterprise Collaboration Systems (ECS)**

ECS adalah tipe sistem informasi yang menggunakan berbagai macam teknologi informasi untuk membantu manusia bekerja bersama-sama. ECS mendukung cakupan luas aktivitas kantor bisnis yang menyediakan peningkatan komunikasi dan arus pekerjaan antar para pekerja. Tujuan dari ECS adalah untuk meningkatkan produktivitas dan kreativitas tim dan kelompok kerja dalam organisasi bisnis modern. Contoh: email, *discussion* forum, video *conference*, chat, dll

☉ **Managerial Support System**

MSS adalah tipe sistem informasi yang difokuskan untuk menyediakan informasi dan dukungan pada manajer agar dapat mengambil keputusan yang efektif.

☉ **Management Information System (MIS)**

System yang berfungsi menghasilkan informasi yang berguna untuk semua tingkatan manajemen, baik berupa rincian detail maupun rangkuman dari basis data yang dihasilkan oleh TPS dan aktivitas di berbagai area fungsional bisnis seperti akuntansi, keuangan, pemasaran.

Tipe laporan MIS:

Periodic Scheduled Reports

- Laporan dibuat dalam format yang telah ditentukan dan disampaikan secara periodic (misalnya: setiap hari atau seminggu sekali)

Exception reports

- Pada beberapa kasus, laporan hanya diproduksi ketika suatu kondisi tertentu terjadi

Demand reports and responses

- Informasi tersedia kapanpun manajer menginginkannya. Dalam hal ini manajer tidak perlu menunggu laporan periodic untuk memperolehnya.

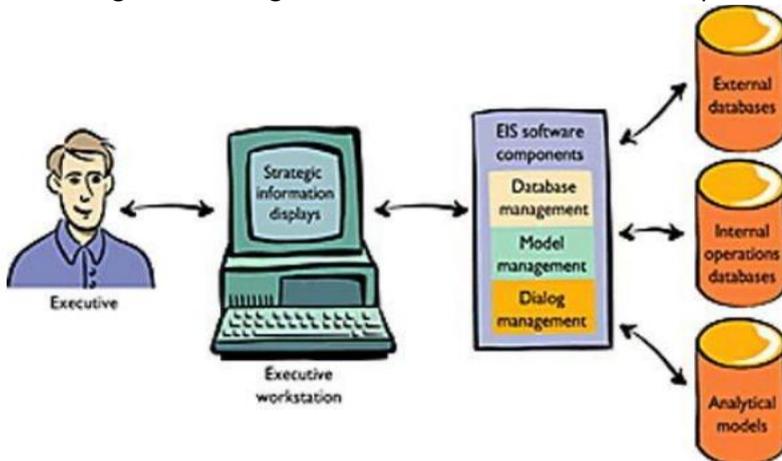
⦿ **Decision Support System (DSS)**

Sistem yang bersifat fleksibel, interaktif dan adaptif yang dibangun khusus untuk mendukung proses pengambilan keputusan atas masalah manajemen yang tidak terstruktur. Data yang diproses berasal dari internal & eksternal. DSS menggunakan pemodelan analitis, simulasi, pencarian data, dan kemampuan menyampaikan informasi untuk membantu manajer.

⦿ **Executive Informatin System (EIS)**

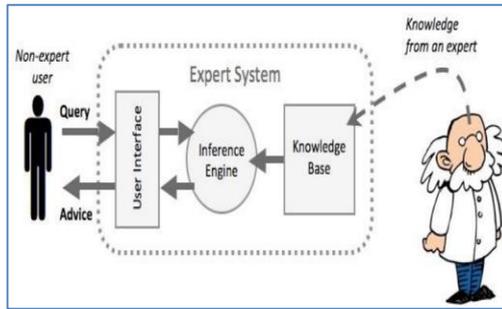
System yang berfungsi dalam proses pengambilan keputusan tingkat strategis sehingga dapat:

- mempengaruhi visi dan misi organisasi,
- perencanaan strategis dan sasaran bisnis,
- struktur organisasi,
- manajemen krisis dan manajemen risiko,
- serta strategi untuk mengendalikan dan memonitor semua operasi



⦿ **Expert system**

ES biasa dikenal juga sebagai *Knowledge-based Information System*, menggunakan pengetahuan (*knowledge*) pada suatu bidang tertentu dan bertindak sebagaimana seorang pakar/konsultan bagi seorang pengguna.

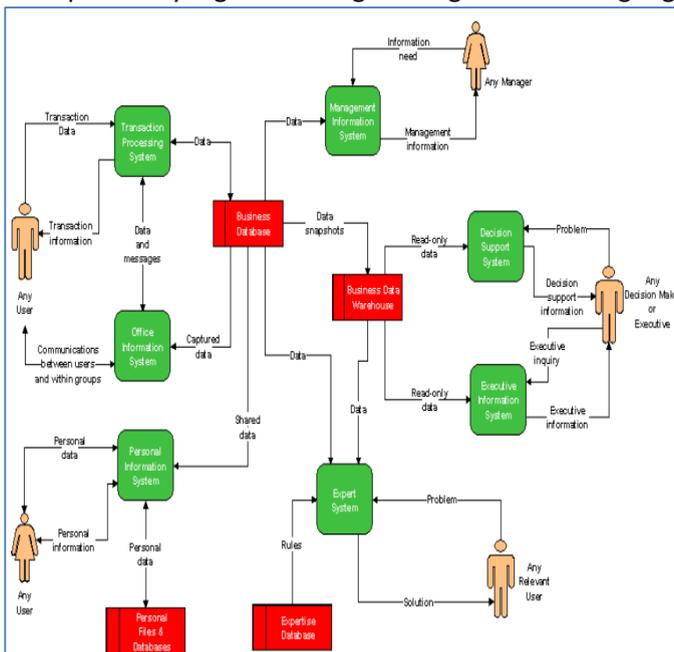


☉ **Knowledge Management System**

KMS dikembangkan oleh banyak perusahaan didasarkan pada pertimbangan-pertimbangan untuk mengelola pengetahuan yang mereka miliki. KMS membantu pekerja yang berpengalaman mengkreasi, mengorganisasi, dan membagi pengetahuan bisnis yang penting kapanpun dan dimanapun.

☉ **Geographic Information System**

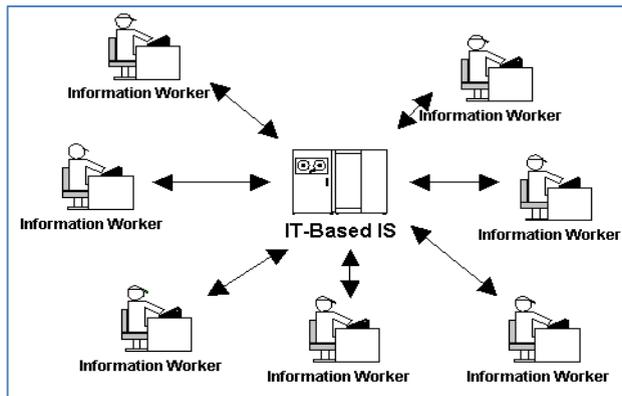
GIS adalah sistem informasi yang mengintegrasikan grafik komputer dengan data geografis dengan elemen-elemen DSS. Sistem ini akan membuat atau menghasilkan tampilan peta dan grafis-grafis lainnya untuk membantu pengambilan keputusan yang berhubungan dengan informasi geografis.



Gambar 2.3 Hubungan Antar Tipe Sistem Informasi

Komponen EIS

- 1) Perangkat Keras
- 2) Program
- 3) Data
- 4) Prosedur
- 5) Orang



Gambar 2.4 Komponen EIS

Teknologi-teknologi Pendukung Integrasi dan Migrasi

1) *Cluster*

Cluster adalah teknik atau metode menghubungkan beberapa komputer dalam suatu jaringan dan menjadikannya bertindak seperti sebuah mesin, dimana komputer tersebut dapat bekerja sama dalam pemrosesan suatu masalah, dari penggabungan beberapa komputer dalam satu jaringan tentu komputer tersebut dapat menghasilkan kecepatan yang sangat tinggi dalam prosesnya. Perusahaan-perusahaan besar sering meng-*cluster server-server*nya untuk mendistribusikan tugas komputasi yang intensif dan berisiko. Jika satu *server* dalam sebuah *cluster* gagal, maka sistem operasi dapat memindahkan proses ke *server* lainnya. Dengan demikian, pengguna dimungkinkan untuk dapat terus bekerja di saat *server* yang gagal sedang dalam proses pemulihan kembali.

2) *Grid*

Grid computing adalah sebuah asosiasi sumber daya komputer dari beberapa domain administrasi untuk mencapai tujuan bersama dengan berbagai layanan kepada pengguna. *Grid Computing* merupakan salah satu dari tipe Komputasi Paralel, adalah penggunaan sumber daya yang melibatkan banyak komputer terpisah secara geografis namun tersambung via jalur komunikasi (termasuk Internet) untuk memecahkan persoalan

komputasi skala besar. Semakin cepat jalur komunikasi terbuka, maka peluang untuk menggabungkan kinerja komputasi dari sumber-sumber komputasi yang terpisah menjadi semakin meningkat. Dengan demikian, skala komputasi terdistribusi dapat ditingkatkan secara geografis lebih jauh lagi, melintasi batas-batas domain administrasi yang ada.

3) **Web service**

Web service adalah suatu sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan. *Web service* digunakan sebagai suatu fasilitas yang disediakan oleh suatu *web site* untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan (*service*) yang disediakan oleh suatu sistem yang menyediakan *web service*. *Web service* menyimpan data informasi dalam format XML, sehingga data ini dapat diakses oleh sistem lain walaupun berbeda *platform*, sistem operasi, maupun bahasa *compiler*.

Web service bertujuan untuk meningkatkan kolaborasi antar pemrogram dan perusahaan, yang memungkinkan sebuah fungsi di dalam *Web Service* dapat dipinjam oleh aplikasi lain tanpa perlu mengetahui detail pemrograman yang terdapat di dalamnya.

4) **Web server**

Web server dapat diartikan sebagai perangkat keras (*hardware*) maupun sistem yang terstruktur melalui perangkat lunak (*software*). Pengertian dari *web server* merupakan *software* yang memberikan layanan yang berfungsi menerima HTTP atau HTTPS dari klien yang sering dikenal dengan web browser dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman web yang berbentuk HTML. Jadi dapat disimpulkan *web server* adalah penyimpanan data web dan *apache* dan lain-lain dan juga aplikasi pengatur untuk memfilter sesuai dengan apa yang di *request* oleh *client*.

5) **API (Application Programming Interface)**

API (*Application Programming Interface*) atau dalam bahasa Indonesianya antarmuka pemrograman aplikasi memiliki arti sebuah bahasa atau format pesan yang digunakan oleh program untuk berkomunikasi dengan sistem operasi atau program pengendali lainnya seperti sistem manajemen *database* (DBMS) atau komunikasi *protocol* yang diimplementasikan dengan menulis fungsi sintaks dalam program, yang menyediakan sarana yang diperlukan untuk meminta layanan program.

6) **Data warehouse**

Data warehouse adalah kumpulan data dari berbagai sumber internal maupun eksternal yang dikumpulkan menjadi satu dalam tempat penyimpanan yang berukuran besar yang bisa digunakan untuk pengambilan keputusan, dan bersifat berorientasi subjek, terintegrasi, *time-variant*, dan *non-volatile*.

7) **Database**

Database atau basis data merupakan kumpulan data atau berbagai informasi yang disimpan di dalam komputer secara sistematis yang dimana data tersebut dapat diubah atau dimanipulasi dengan menggunakan aplikasi tertentu.

8) **OTT (Over The Top)**

Over The Top merupakan sebuah layanan yang disampaikan melalui jaringan operator, tetapi tidak secara langsung melibatkan operator dimana layanan ini berjalan melalui internet. Layanan tersebut berisi konten yang bisa berupa data, informasi, dan multimedia. Beberapa contoh layanan OTT yang sering pada umumnya kita ketahui, seperti Yahoo, Google, iTunes, Facebook, Twitter, Line, dan lain-lain.

9) **Socket**

Definisi dari *socket* adalah *interface* pada jaringan yang menjadi titik komunikasi antarmesin pada Internet Protokol, tanpa adanya komunikasi ini, tidak akan ada pertukaran data dan informasi jaringan. *Socket* terdiri dari elemen-elemen utama yaitu, Protokol, *Local IP*, *Local Port*, *Remote IP*, dan *Remote Port*.

Contoh komunikasi sederhana adalah komunikasi antara komputer X dan komputer Z. Baik komputer X maupun komputer Z harus memiliki identitas unik, yang direpresentasikan oleh IP masing-masing. Komunikasi yang terjadi melalui *port*, sehingga baik komputer X maupun komputer Z harus memiliki *port* yang dapat diakses satu sama lain.

10) **Data mining**

Data mining merupakan sebuah atau serangkaian proses untuk menggali nilai tambah atau pencarian secara otomatis yang berupa informasi berguna dalam tempat penyimpanan data yang besar. Terdapat beberapa istilah lain dari *data mining* yaitu *Knowledge Discovery (mining) in Database (KDD)*, *Knowledge Extraction*, *Business Intelligence*, dan merupakan alat yang penting untuk memanipulasi data dalam penyajian informasi sesuai

kebutuhan pengguna dengan tujuan untuk membantu dalam analisis koleksi pengamatan perilaku.

11) Open data

Open data atau data terbuka merupakan informasi atau data yang tersedia yang bebas digunakan, digunakan kembali (*reusable*) dan didistribusikan kembali (*redistributed*) oleh siapa saja. Data dapat dikatakan terbuka penuh apabila:

- Data harus tersedia utuh dan tidak memerlukan biaya reproduksi yang berlebihan seperti, data bisa diunduh dari internet. Data juga harus tersedia dalam bentuk yang mudah digunakan (*convenient*) dan dapat diubah (*modifiable*).
- Penggunaan kembali dan penyebaran kembali data harus dilakukan melalui syarat-syarat yang berlaku bagi penggunaan-kembali dan penyebaran-kembali, termasuk pencampuran dengan set data lain.
- Data harus dapat digunakan, digunakan kembali, dan disebarluaskan kembali oleh semua orang, tidak boleh ada diskriminasi atas bidang usaha, orang, atau kelompok. Misalnya, batasan '*non-komersial*' yang melarang penggunaan '*komersial*', atau batasan penggunaan untuk tujuan tertentu (mis, hanya untuk pendidikan), tidak dibolehkan.

12) Cloud

Cloud sebenarnya merupakan sebuah istilah yang diberikan pada teknologi jaringan internet. Internet sendiri merupakan kependekan dari *Interconnection network* yang memiliki arti sebuah sistem teknologi komunikasi yang bisa menghubungkan antar satu komputer dengan komputer lainnya sehingga akan menjadi sebuah jaringan, jaringan ini mencakup seluruh dunia. Terdapat istilah *cloud computing* yang dikenal dalam bahasa Indonesianya komputasi awan yang memiliki arti gabungan atau kombinasi pemanfaatan dari teknologi komputer dengan pengembangan berbasis internet.

13) Smart city

Smart City adalah sebuah konsep kota cerdas yang membantu masyarakatnya dengan mengelola sumber daya yang ada dengan efisien dan memberikan informasi yang tepat kepada masyarakat atau mengantisipasi kejadian yang tak terduga. *SmartCity* dapat disimpulkan bahwa memanfaatkan sumber informasi dan menggunakan teknologi yang canggih untuk mempermudah kehidupan.

14) *Internet of things*

Internet of thing sebuah istilah yang belakangan ini mulai ramai ditemui namun masih banyak yang belum mengerti arti dari istilah ini. Sebetulnya hingga saat ini belum ada definisi standar mengenai *Internet of Things*, namun secara singkat *Internet of Things* bisa dibilang adalah di mana benda-benda di sekitar kita dapat berkomunikasi antara satu sama lain melalui sebuah jaringan seperti internet.

15) *Machine to machine*

M2M merupakan salah satu dari bidang kategori ICT (*Information and Communication Technology*), yaitu penggabungan teknologi komunikasi, komputer dan daya sehingga memungkinkan terjadinya komunikasi jarak jauh antara manusia dan mesin melalui interaksi fisik, kimia serta sistem dan proses biologi. Sebagai tambahan, M2M merupakan aplikasi komputasi baru dimana data dialirkan dari dan ke fisik serta biologi. Berikut ini merupakan aplikasi dari M2M:

- Sensor-sensor suhu, tekanan, kelembaban, kehadiran, serta sensor-sensor yang berkaitan dengan diagnosa kesehatan.
- Layanan pelacakan posisi untuk manusia, binatang, kendaraan, barang dan properti.
- Pemantauan jarak jauh untuk efisiensi kehidupan, kesehatan, properti dan pekerjaan.
- Pengendalian jarak jauh dan/atau pengelolaan mesin sederhana (elektronika konsumen, *vending machine*), *micro-robotics*, sampai dengan mesin kompleks seperti otomasi pabrik, pengendalian industri, dll.
- Identitas, pengelolaan kepemilikan, autentikasi, transfer keuangan, layanan posisi, konsep ekonomi digital.

B. EKONOMI DIGITAL

Internet masa kini bukan lagi menjadi barang mahal bagi sebagian besar penduduk dunia. Internet, sebagai sesuatu yang dapat dikatakan baru mampu menjadikan dirinya sebagai sebuah gaya hidup manusia modern. Bahkan, susah saja ketika internet disebut sebagai kebutuhan pokok manusia. Hal ini disebabkan karena manusia tidak dapat dipisahkan oleh internet dan kehidupan mereka hampir sebagian besar bersinggungan dengan internet. Perlu pula diketahui bahwa internet adalah buah dari globalisasi yang mendunia. Globalisasi menjadikan manusia menjadi lebih modern karena salah satu dampak globalisasi adalah melek teknologi Internet masa kini bukan lagi menjadi barang mahal bagi sebagian besar penduduk dunia.

Internet, sebagai sesuatu yang dapat dikatakan baru mampu menjadikan dirinya sebagai sebuah gaya hidup manusia modern. Bahkan, sah-sah saja ketika internet disebut sebagai kebutuhan pokok manusia. Hal ini disebabkan karena manusia tidak dapat dipisahkan oleh internet dan kehidupan mereka hampir sebagian besar bersinggungan dengan internet. Perlu pula diketahui bahwa internet adalah buah dari globalisasi yang mendunia. Globalisasi menjadikan manusia menjadi lebih modern karena salah satu dampak globalisasi adalah melek teknologi Internet masa kini bukan lagi menjadi barang mahal bagi sebagian besar penduduk dunia. Internet, sebagai sesuatu yang dapat dikatakan baru mampu menjadikan dirinya sebagai sebuah gaya hidup manusia *modern*. Bahkan, sah-sah saja ketika internet disebut sebagai kebutuhan pokok manusia. Hal ini disebabkan karena manusia tidak dapat dipisahkan oleh internet dan kehidupan mereka hampir sebagian besar bersinggungan dengan internet. Perlu pula diketahui bahwa internet adalah buah dari globalisasi yang mendunia. Globalisasi menjadikan manusia menjadi lebih *modern* karena salah satu dampak globalisasi adalah melek teknologi.

Secara keseluruhan, jumlah pengguna internet di seluruh dunia diproyeksikan akan mencapai 3 miliar orang pada tahun 2015. Kemudian, tiga tahun setelahnya, tahun 2018, diperkirakan akan meningkat menjadi 3,6 miliar manusia di bumi akan mengakses internet, setidaknya sekali dalam satu bulan. Tidak hanya berimbas pada negara-negara maju saja, Indonesia sebagai salah satu negara berkembang dengan jumlah penduduk yang tinggi juga ternyata berhasil mengikuti *trend* baru teknologi digital. Menurut lembaga riset pasar *e-Marketer*, populasi *netter* Tanah Air mencapai 83,7 juta orang pada tahun 2014.

Angka yang berlaku untuk setiap orang yang mengakses internet setidaknya satu kali setiap bulan itu mendudukkan Indonesia di peringkat ke-6 di dunia dalam hal pengguna internet. Pada tahun 2017, *e-Marketer* memperkirakan, jumlah *netter* Indonesia akan mencapai 112 juta orang atau mengalahkan Jepang pada peringkat ke-5 yang pertumbuhan jumlah pengguna internetnya lebih lambat. Indonesia dan India masih memiliki ruang pertumbuhan jumlah pengguna internet yang besarnya bisa mencapai dua digit setiap tahun. Di atas Indonesia, lima besar negara pengguna internet di dunia secara berurutan untuk saat ini diduduki oleh Tiongkok, Amerika Serikat, India, Brasil, dan Jepang.

Internet dan *gadget* merupakan 2 hal yang tidak dapat dipisahkan. Hal ini dikarenakan penggunaan *gadget* juga meningkat seiring dengan meningkatnya jumlah pengguna internet di dunia. *Gadget* sama halnya dengan internet merupakan barang vital yang tidak terlepas dari keseharian

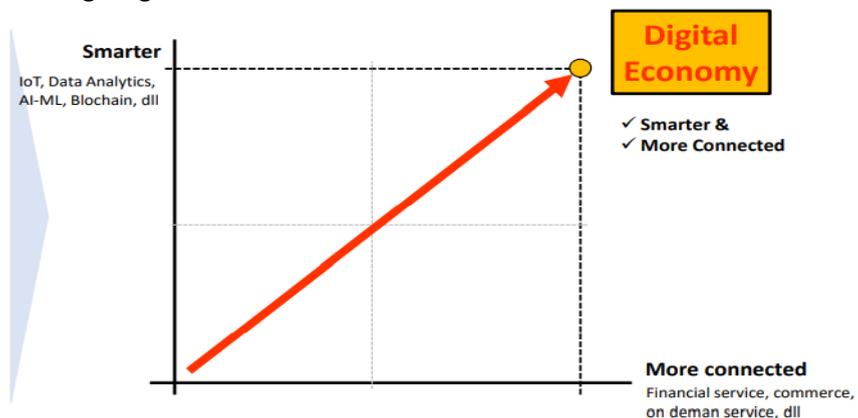
manusia bahkan tidak terlepas dari tangan manusia. *Gadget* menjadi sangat *popular* pada kehidupan masa kini oleh karena bentuknya yang sangat simpel dan nyaman digenggam sehingga mudah dibawa kemanapun dan kapanpun. Tidak hanya itu, *gadget* juga merupakan media dari sebuah internet karena internet akan menjadi sebuah komunikasi ketika berada dan dijalankan oleh *gadget*. Hal inilah yang kemudian merumuskan perkembangan antara internet dan *gadget* yang selaras.

Internet dan *gadget* diperuntukkan untuk banyak hal oleh manusia. Ketika awal perkembangan internet pada tahun 1969, internet hanya diperuntukkan untuk hal kemiliteran. Kemudian, beberapa tahun selanjutnya internet pun dipergunakan oleh kalangan akademis khususnya bagi universitas-universitas dalam pengembangan situs *website* maupun dalam rangka menambah pengetahuan lewat *browsing* internet. *Trend* terbaru perkembangan internet masa kini yaitu komunikasi, sosial media, dan belanja *online*. Komunikasi lewat internet memungkinkan terjalannya komunikasi antar individu, antara individu dengan kelompok, maupun kelompok dengan kelompok baik secara tertulis, grafis, audio, dan atau video.

Sosial media merupakan salah satu hal baru namun sudah menjadi sebuah gaya hidup bagi seluruh kalangan usia manusia. Sosial media memungkinkan seseorang untuk memiliki sebuah akun yang berisi tentang identitas seseorang yang dapat diakses oleh orang lain sehingga orang lain pun saling mengetahui satu sama lain di dunia maya. Dunia yang tidak terbatas oleh ruang dan waktu menyebabkan pertemanan di sosial media menjadi tanpa batas dan tidak dibatasi oleh apapun termasuk jarak. Sosial media sebagai sebuah kebutuhan kemudian menjadi sesuatu yang nyaman bagi manusia karena sifatnya yang sangat fleksibel dan mudah digunakan serta juga mampu memudahkan kehidupan manusia. Sosial media yang menjadi sangat hits dan kekinian saat ini ternyata juga membawa imbas pada gaya hidup manusia yang lain yaitu konsumtif sebagaimana Indonesia terkenal sebagai salah satu negara yang cukup konsumtif.

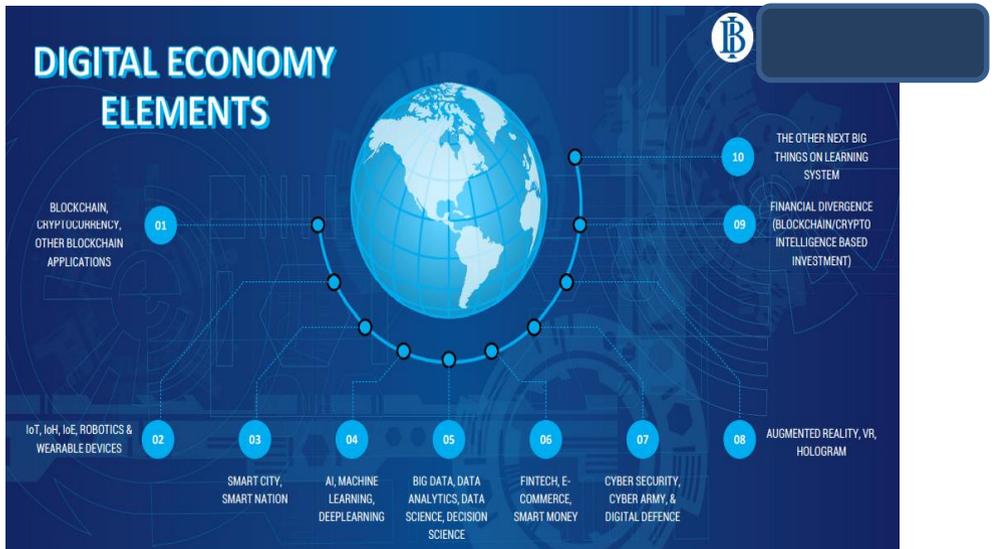
Sifat konsumtif manusia masa kini tak lagi sepenuhnya konvensional seperti yang terjadi di masa lalu. Ketika dulu masyarakat datang ke pasar, menemukan barang yang diinginkan, berinteraksi dengan penjual, dan melakukan transaksi, maka masyarakat masa kini terkadang tak perlu harus meninggalkan tempat dan bertatap mata langsung dengan penjual. *Gadget* merupakan media penghubung antara penjual dan pembeli di tempatnya masing-masing. *Gadget* sebagai alat ajaib mampu mempertemukan keinginan konsumen terhadap sesuatu yang ditawarkan oleh produsen.

Transaksi pasar maya tidak hanya menguntungkan antara penjual dan pembeli namun juga berimbas pada perekonomian wilayah yang bersangkutan. Perekonomian berbasis teknologi seperti ini dikenal sebagai ekonomi digital. Ekonomi digital merupakan sebuah spesialisasi dari ekonomi regional. Ekonomi digital adalah kolaborasi komplit dari berbagai elemen yang saling berhubungan satu dengan yang lain. Ekonomi ini mencoba membuat serangkaian proses berlanjut yang dimulai dari pedagang/penyedia jasa menjajakan barang/jasa mereka di sebuah ruang tertentu di internet, pembeli/konsumen yang membeli barang/menggunakan jasa pada ruang tersebut, pembayaran langsung/dengan bank tertentu, pengiriman barang langsung/menggunakan jasa pengiriman barang, serta penggunaan jasa secara langsung.



Gambar 2.5 Terminology Ekonomi digital (BI InSTITUTE, 2018)

Ekonomi digital di Indonesia telah berhasil masuk ke berbagai lapisan masyarakat dalam segala usia. Membaurnya masyarakat masa kini terhadap ekonomi digital terjadi oleh karena kenyamanan yang dirasakan atas ekonomi berbasis teknologi yang sangat memudahkan pekerjaan manusia. Ada banyak macam aplikasi dan implementasi dari ekonomi digital di Indonesia diantaranya industri, penyediaan jasa layanan antar atau logistik, *provider* telekomunikasi, produsen perangkat pintar, dan lain-lain. Salah satu industri ekonomi digital yang sangat *booming* adalah *e-commerce*.



Gambar 2.6 Elemen-elemen ekonomi Digital (BI Institute, 2018)

E-commerce atau secara sederhana dikenal sebagai ruang jual beli *online* menjadi sangat *in* di masyarakat karena para calon konsumen tidak perlu melakukan sebuah pergerakan untuk memperoleh barang yang diinginkan dan dengan duduk di tempat saja seseorang sudah dapat menikmati barang yang diinginkan. Sedangkan bagi pedagang bahwa mereka tidak perlu mencari lokasi tempat berjualan namun dengan sendirinya barang yang dijual akan laku. Hal ini menjadi sangat efektif dan efisien secara ruang dan waktu bagi manusia masa kini yang sangat sibuk.

Secara umum, terdapat 3 kategori *e-commerce* yang ada di Indonesia yaitu *Online forums and classifieds*, *Marketpalce*, dan *others*. *Online forums and classifieds*: Kaskus, OLX Indonesia, Jualo, Lamudi, Rumah123, Rumah, Carmudi, Mobil123, Lazada Indonesia, Matahari Mall, Bhinneka, Blibli, Zalora Indonesia, Shopee, Berrybenka, Paraplou, Bobobobo, Tiket, Traveloka, Groupon Indonesia, dan Grivy. *Marketplace*: Tokopedia, Bukalapak, Qoo10 Indonesia, Elevenia, Blanja, dan Rakuten Belanja Online. *Others*: *Indonetwork*, *Indo Trading*.

C. HYPERCOMPETITIVE

a. Strategi mencapai keunggulan bersaing

Strategi adalah suatu rencana aksi yang menyelaraskan sumber-sumber dan komitmen organisasi untuk mencapai kinerja unggul. Keunggulan bersaing/kompetitif adalah suatu manfaat yang ada ketika suatu perusahaan

mempunyai dan menghasilkan suatu produk dan atau jasa yang dilihat dari pasar targetnya lebih baik dibandingkan dengan para kompetitor terdekat.

Menurut Michael Porter, ada tiga landasan strategi yang dapat membantu organisasi memperoleh keunggulan kompetitif, yaitu keunggulan biaya, diferensiasi, dan fokus. Porter menamakan ketiganya strategi umum (strategi generik). Keunggulan biaya menekankan pada pembuatan produk standar dengan biaya per unit sangat rendah untuk konsumen yang peka terhadap perubahan harga. Diferensiasi adalah strategi dengan tujuan membuat produk yang menyediakan jasa yang dianggap unik di seluruh industri dan ditujukan kepada konsumen yang tidak terlalu peduli dengan perubahan harga. Fokus berarti membuat produk dan menyediakan jasa yang memenuhi keperluan sejumlah kelompok kecil konsumen.

Strategi Porter mensyaratkan adanya penataan organisasi, prosedur pengendalian, sistem intensif yang berbeda. Perusahaan besar dengan akses sumber daya yang besar biasanya bersaing dengan landasan keunggulan biaya dan atau dengan diferensiasi, sedangkan perusahaan kecil sering bersaing dengan landasan fokus.

Porter menekankan pentingnya perencanaan strategi dan melakukan analisis biaya manfaat untuk mengevaluasi berbagai peluang diantara unit-unit bisnis yang sudah ada dan unit bisnis yang potensial dalam perusahaan. Berbagai aktivitas dan sumber daya dapat meningkatkan keunggulan kompetitif karena dengan demikian biaya berkurang dan diferensiasi meningkat. Selain itu, Porter juga menekankan perlunya perusahaan mengalihkan keterampilan dan keahlian diantara unit bisnis otonomi secara efektif agar memperoleh keunggulan kompetitif.

Berdasarkan analisis kompetitif, Porter menyatakan bahwa walaupun suatu perusahaan memiliki banyak kekuatan dan kelemahan dalam berhadapan dengan para pesaing. Terdapat dua jenis dasar keunggulan kompetitif yang dapat dimiliki oleh suatu perusahaan yaitu biaya rendah dan diferensiasi yang sangat ditentukan oleh struktur industri. Keduanya dihasilkan dari kemampuan perusahaan dalam *strategies*). Strategi fokus terdiri dari dua varian yaitu fokus biaya dan fokus diferensiasi.

1) Keunggulan Biaya (*Overall Cost Leadership*)

Keunggulan biaya menanggulangi kelima kekuatan dengan lebih baik dibandingkan pesaingnya. Untuk kepentingan inilah maka Porter kemudian menyarankan tiga strategi yang harus di pertimbangkan oleh perusahaan yaitu strategi keunggulan biaya (*overall cost leadership*), diferensiasi (*differentiation*), dan fokus (*focus*) yang disebutnya sebagai strategi generik (*generic* merupakan strategi yang paling jelas dari ketiga strategi). Dengan konsep ini, perusahaan bersiap menjadi produsen berbiaya rendah di dalam

industri. Sumber keunggulan biaya bervariasi dan bergantung pada struktur industri. Produsen berbiaya rendah harus menemukan dan mengeksploitasi semua sumber keunggulan biaya. Apabila perusahaan dapat mencapai dan mempertahankan keseluruhan keunggulan biaya, maka perusahaan akan menjadi perusahaan berkinerja tinggi di dalam industri. Asalkan perusahaan dapat menguasai harga pada rata-rata industri.

Strategi keunggulan biaya yang sukses biasanya mewarnai seluruh hal dalam perusahaan, terbukti dari efisiensi yang tinggi, biaya *overhead* yang rendah, kebocoran yang kecil, tidak toleran terhadap pemborosan, penyaringan ketat terhadap permintaan anggaran, rentang kendali yang lebar, penghargaan dikaitkan dengan pengurangan biaya, dan partisipasi karyawan yang luas dalam usaha pengendalian biaya. Beberapa contoh perusahaan yang terkenal karena strategi keunggulan biaya adalah *Wal-Mart*, *Lincoln Electric*, dan *Briggs and Stratton*.

2) Diferensiasi (*differentiation*)

Dalam strategi diferensiasi, perusahaan berusaha menjadi unik dalam industri pada berbagai dimensi yang secara umum dihargai oleh pembeli. Cara melakukan diferensiasi berbeda untuk setiap industri. Diferensiasi dapat didasarkan pada produk itu sendiri. Sistem penyerahan produk yang dipergunakan untuk menjualnya, pendekatan pemasaran, dan faktor lain. Perusahaan yang dapat mencapai dan mempertahankan diferensiasi akan menjadi perusahaan berkinerja tinggi dalam industri. Logika dari strategi diferensiasi mengharuskan perusahaan memilih atribut untuk mendiferensiasikan diri secara berbeda dengan atribut pesaingnya.

Strategi diferensiasi yang berhasil memungkinkan perusahaan menetapkan harga lebih tinggi untuk produknya yang memperoleh loyalitas pelanggan karena konsumen bisa begitu terikat dengan fitur-fitur diferensiasi. Fitur-fitur yang membedakan produk suatu perusahaan bisa mencakup pelayanan yang sangat unggul, ketersediaan suku cadang, desain teknis, kinerja produk, umur manfaat produk, hemat bahan bakar, atau kemudahan penggunaan.

3) Fokus (*focus*)

Fokus berbeda dengan strategi lain karena menekan pilihan akan cakupan bersaing yang sempit dalam suatu industri. Dengan mengoptimalkan strategi untuk segmen pasar, penganut strategi fokus berusaha untuk mencapai keunggulan bersaing di dalam segmen sasaran walaupun tidak memiliki keunggulan bersaing secara keseluruhan. Strategi fokus memiliki dua varian, dalam fokus biaya, perusahaan mengusahakan keunggulan biaya dalam

segmen sasarannya, sedangkan dalam fokus diferensiasi, perusahaan mengusahakan diferensiasi dalam segmen sasarannya. Apabila perusahaan dapat mencapai keunggulan biaya yang dapat dipertahankan (fokus biaya) atau diferensiasi (fokus diferensiasi) dalam segmennya dan segmen tersebut menarik secara struktural, maka penganut strategi fokus akan menjadi perusahaan yang berkinerja tinggi di dalam industrinya.

Strategi fokus akan sangat efektif ketika konsumen mempunyai pilihan atau persyaratan tertentu yang dapat dipenuhi oleh perusahaan dan ketika perusahaan pesaing tidak berusaha untuk melakukan spesialisasi dalam segmen konsumen yang sama. Contoh perusahaan yang melakukan strategi fokus adalah BMW Jerman yang secara eksklusif memfokuskan pada pembuatan mobil-mobil mewah kelas atas. Strategi BMW tersebut bertentangan dengan paradigma umum industri mobil, yaitu memproduksi mobil untuk pasar masal. Richard A. D'Aveni (1994), mengemukakan bahwa konsep keuntungan neo-klasik maupun konsep *dynamic strategy* dari Porter (1991) adalah berjangka pendek (*short-life*) yang bersifat statis. Sekarang ini keadaannya berubah, yang diperlukan adalah untuk jangka panjang (*long-life*) dan dinamis. Untuk menghadapi kondisi yang semakin dinamis (*hypercompetitive*) seperti sekarang ini, Richard A. D'Aveni (1994), mengemukakan suatu ide dasar, bahwa perusahaan harus menekankan pada strategi yang memfokuskan pada pengembangan kompetensi inti (*building core competency*), pengetahuan dan keunikan *intangible asset*, untuk menciptakan keunggulan. Oleh karena itu, D'Aveni mengajukan tujuh kunci keberhasilan perusahaan dalam *hypercompetitive environment*, yang dikenal dengan "The New 7-S's".

Konsep "The New 7-S's" ini meliputi pokok-pokok dasar sebagai berikut:

- **Superior stakeholder satisfaction**
Strategi yang pertama dari New 7-S's adalah memberi kepuasan yang istimewa kepada *stakeholder* yaitu semua unsur yang memiliki kepentingan dalam perusahaan dengan tanpa terkecuali seperti pemasok, karyawan, manajer, pemegang saham, konsumen, pemerintah dan masyarakat sekitarnya.
- **Strategic soothsaying**
Strategi kedua adalah strategi yang memfokuskan pada sasaran, artinya perusahaan harus mencari posisi yang tepat bagi produk dan jasa-jasa yang dihasilkan perusahaan.

- **Positioning for speed**
Strategi ketiga adalah strategi dalam memposisikan perusahaan secara cepat di pasar. Perusahaan harus segera mengkomunikasikan produk dan jasa-jasa tersebut ke pasar supaya segera dikenal konsumen.
- **Positioning for surprise**
Strategi keempat adalah membuat posisi yang mencengangkan melalui barang dan jasa-jasa baru yang lebih unik dan berbeda serta memberikan nilai tambah baru, sehingga konsumen lebih menyukai barang dan jasa yang diciptakan perusahaan.
- **Shifting the role of the game**
Strategi kelima adalah mengubah pola-pola persaingan perusahaan yang dimainkan sehingga pesaing terganggu (*disruption*) dengan pola-pola yang berbeda.
- **Signaling strategic intent**
Strategi keenam adalah mengutamakan pada perasaan. Kedekatan dengan para karyawan, relasi, dan konsumen merupakan strategi yang ampuh untuk meningkatkan kinerja perusahaan.
- **Simultaneous and sequential strategic thrusts**
Strategi ketujuh adalah mengembangkan faktor-faktor pendorong atau penggerak strategi secara simultan dan berurutan (*thrusts*) melalui penciptaan barang-barang dan jasa-jasa yang selalu memberi kepuasan kepada konsumen.

b. Dasar Keunggulan Bersaing

Untuk mencapai keunggulan bersaing, seorang wirausahawan harus mampu mengenali berbagai unsur dasar untuk mencapai keunggulan bersaing, yakni sebagai berikut:

1) Harga atau nilai

Seorang pengusaha harus mampu menghasilkan produk dan atau jasa rendah biaya, sehingga strategi dalam menetapkan harga (tidak terlalu tinggi dibandingkan dengan produk/jasa para pesaing). Jika mampu dapat juga ditambahkan bahwa produk/jasa memiliki nilai (bernilai) lebih dibandingkan dengan harga pesaing. Dengan demikian, produk/jasa kita memiliki keunggulan dari segi harga dan nilai. Pelanggan yang sensitif terhadap harga, biasanya selisih harga Rp1.000 saja dengan produk dan atau jasa pesaing akan menjadi pertimbangan dalam membeli produk yang lebih rendah harganya.

2) Menyenangkan konsumen

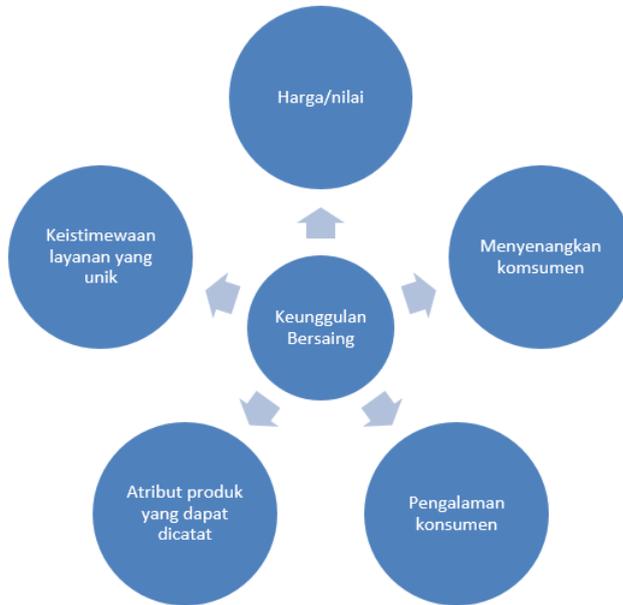
Keunggulan kedua yang harus diupayakan agar produk/jasa dapat bersaing dengan para kompetitor adalah diupayakan agar produk/jasa dapat menyenangkan konsumen. Menyenangkan dalam hal ini dari berbagai aspek, seperti kualitas produk/jasa yang bermutu dan memberi kepuasan. Misalnya, pelayanan yang memuaskan, komunikasi yang memuaskan, dan tanpa komplain atau setidak-tidaknya bila dikomplain segera ditanggapi atau tidak ditunda-tunda.

3) Pengalaman konsumen

Pengalaman baik atau buruk yang kita sampaikan dan yang dialami oleh konsumen, umumnya akan menjadi catatan penting (sering kali melekat seumur hidup). Untuk itu, berikanlah pengalaman yang paling menyenangkan atau memuaskan bagi para pemangku kepentingan, lebih-lebih bagi para konsumen pelanggan. Pengalaman yang baik akan dikenang sepanjang masa, bahkan sering akan ditularkan kepada handai tolan konsumen. Demikian sebaliknya, pengalaman buruk akan cepat menyebar dari mulut ke mulut, baik kepada sahabat maupun tetangga terdekat, bahkan di era teknologi informasi dewasa ini, kadang kala keluhan atau pengalaman buruk dipublis secara meluas melalui jejaring sosial. Agar produk/jasa kita unggul, maka di samping dari sudut harga, nilai, dan menyenangkan konsumen, berikanlah pengalaman kepada konsumen sebaik mungkin (*do your best*).

4) Atribut produk yang dapat dicatat

Keunggulan berikut yang harus dicapai oleh seorang pengusaha adalah seluruh atribut produk/jasa yang melekat di dalamnya harus dicatat. Manfaat dari catatan atribut produk/jasa adalah agar produk/jasa dapat ditingkatkan dari atribut yang sudah ada sebelumnya. Minimal seluruh atribut produk/jasa dapat dikenalkan tidak hanya kepada konsumen, namun juga kepada para pegawai atau karyawan kita. Dengan demikian, baik konsumen maupun karyawan kita mampu mengenali seluruh atribut produk/jasa kita, sehingga tidak ada yang dirahasiakan agar konsumen tidak merasa ditipu, baik secara sengaja maupun tidak sengaja.



Gambar 2.7 Dasar Keunggulan Bersaing. Sumber: Longenecker, dkk. (2005)

5) Keistimewaan layanan yang unik

Jika keempat unsur tersebut telah mencapai posisi unggul, hal yang tidak kalah pentingnya adalah bagaimana keistimewaan layanan yang unik dapat ditampilkan. Contoh yang baik, misalnya bagaimana keistimewaan layanan yang diberikan oleh Singapore Airlines. Keunikan yang dapat dicatat antara lain meskipun pesawat tidak atau belum penuh *sit* atau penumpangnya, tetapi kalau jadwal penerbangan sudah menunjukkan angka sama seperti angka yang tertera pada tiket sesuai jadwal *take-off*, maka Singapore Airlines tidak pernah “menunda jadwal penerbangan”. Di dalam pesawat, kita dapat mengetahui berbagai informasi tentang jadwal, tempat rekreasi, tempat penginapan atau hotel, jalur, peta, dan berbagai informasi lain tentang Singapura. Bila anda telah menjadi anggota (melakukan penerbangan dengan Singapore Airlines lebih dari 60.000 km penerbangan pada kelas eksekutif), anda akan mendapat layanan khusus. Anda akan memperoleh berbagai informasi tentang menu-menu kesukaan anda di berbagai hotel di Singapura, *overload* bagasi dimungkinkan, dan pelayanan plus lainnya. Jadi, yang harus dilakukan oleh seorang wirausahawan adalah bagaimana memberikan layanan yang unik dibandingkan dengan produk/jasa pesaing terdekat kita, sehingga pada akhirnya konsumen akan menjadi konsumen yang loyal dengan perusahaan anda.



PERAN, STRATEGI, DAN TRANSFORMASI BISNIS SERTA PERKEMBANGAN TEKNOLOGI DIGITAL

A. PERAN BISNIS DIGITAL

Perkembangan teknologi informasi yang makin pesat telah membawa dunia memasuki era digital (Rahayu, Silfia, & Juita, 2017). Asromawardi (2016) mendefinisikan era digital sebagai suatu istilah yang digunakan dalam kemunculan teknologi digital, jaringan internet khususnya teknologi informasi. Kemunculan teknologi informasi ini telah membawa dampak pada berbagai bidang kehidupan seperti ekonomi, politik, sosial & budaya, pertahanan dan keamanan. Namun, bidang ekonomi lebih banyak mendapatkan perhatian karena berkaitan dengan hajat hidup orang banyak, khususnya lagi pada sektor bisnis.

Sektor bisnis pada era digital ini mengalami perubahan proses bisnis yang sangat mendasar akibat dari kemunculan dari teknologi informasi. Hal ini juga dikemukakan oleh Hall & Singleton (2009) yang mengungkapkan bahwa teknologi informasi telah menginspirasi rekayasa ulang berbagai proses bisnis tradisional. Bodnar & Hopwood (2006) menyatakan bahwa proses bisnis dapat diklasifikasikan menjadi dua bagian, yakni proses bisnis primer dan pendukung. Proses bisnis primer terdiri dari logistik penjualan *inbound* & *outbound*, operasi, pemasaran, dan jasa. Sedangkan, proses bisnis pendukung terdiri dari prokuremen, pengembangan teknologi, sumber daya manusia, dan infrastruktur perusahaan. Oleh karena itu dapat dikatakan terjadi perubahan yang sangat mendasar karena mempengaruhi dari hulu hingga hilir aktivitas sebuah perusahaan.

Selanjutnya Hammer dan Champy (1993) yang dikutip dalam Hartono (2005) menyatakan bahwa terdapat tiga tekanan atau faktor yang menyebabkan perubahan bisnis, yakni pelanggan (*customer*), kompetisi (*competition*), dan perubahan (*change*). Tekanan pertama berasal dari pelanggan, dimana pelanggan saat ini menyukai kemudahan akses informasi terhadap barang dan jasa yang mereka inginkan. Hal ini seiring dengan perubahan perilaku pelanggan yang kian dekat dengan teknologi informasi.

Tekanan kedua berasal dari kompetisi antara perusahaan. Persaingan bisnis yang makin tajam membuat perusahaan harus melakukan hal-hal yang baru dan inovatif dalam menghadapi perubahan teknologi informasi. Selanjutnya, tekanan ketiga berasal dari perubahan. Hartono (2005) mengungkapkan bahwa pasar, produk, jasa, lingkungan bisnis dan teknologi akan terus berubah dan berkembang dengan cepat. Hal ini menuntut perusahaan untuk bisa beradaptasi dengan segala bentuk perubahan, agar mampu bertahan dan memenangkan persaingan.

Ketiga tekanan tersebut mendorong terjadinya perubahan bisnis dari sistem tradisional menjadi sistem yang berbasis digital. Proses perubahan ini lebih dikenal dengan sebutan transformasi digital. Transformasi digital secara umum digambarkan sebagai integrasi antara teknologi informasi dan bisnis menjadi tunggal dan selaras. Teknologi informasi menghasilkan informasi yang sangat dibutuhkan dalam menunjang kegiatan bisnis perusahaan yang pada akhirnya berdampak pada kinerja perusahaan. Hal ini dibuktikan oleh riset yang dilakukan oleh Capgemini *Consulting* & MIT Sloan *Management* (2012) yang menemukan bahwa perusahaan yang melakukan transformasi digital dapat meningkatkan pendapatan sebesar 9 (Sembilan) %, keuntungan sebesar 26 (Dua puluh enam) % dan nilai pasar sebesar 12 (dua belas) %.

B. STRATEGI BISNIS DIGITAL

Definisi sistem informasi berbeda dengan definisi teknologi informasi, dimana teknologi informasi mengacu secara khusus pada teknologi, seperti *hardware*, *software*, dan jaringan telekomunikasi. Teknologi informasi memfasilitasi akuisisi, pemrosesan, penyimpanan, pengiriman, dan penyebaran informasi. Beberapa sistem informasi secara total bisa diotomatisasi dengan menggunakan teknologi informasi. Dengan cara ini, informasi dapat secara sempurna terhubung dan memenuhi seluruh permintaan yang ada secara *real time*. (Oktavia, 2011)

Akuisisi suatu sistem informasi dalam suatu perusahaan/organisasi, bukanlah suatu masalah mudah karena sistem informasi merupakan bagian dari investasi perusahaan yang membutuhkan usaha yang cukup besar serta melibatkan beberapa aspek, seperti manusia, uang, waktu, dll. Diperlukan

perumusan strategi yang tepat agar perumusan sistem informasi dapat tepat guna dan hasil yang diperoleh sesuai dengan tujuan yang diharapkan. Beberapa permasalahan yang mungkin timbul jika perumusan strategi perusahaan tidak sesuai di antaranya: (1) Hilangnya kesempatan bisnis. Sistem dan investasi teknologi yang diterapkan kadang kala tidak mendukung tujuan bisnis, bahkan menjadi batasan terhadap perkembangan bisnis; (2) Kurangnya integrasi sistem dan pengaturan informasi yang tidak efektif, informasi yang tidak akurat dan tidak memadai dalam menjalankan bisnis; (3) Prioritas yang tidak berdasarkan kebutuhan bisnis, level sumber daya yang tidak optimal, dan rencana proyek yang secara konsisten berubah, sehingga mengakibatkan performa bisnis yang tidak meningkat, biaya yang tinggi, solusi bisnis dengan kualitas dan produktivitas yang rendah; (4) Strategi teknologi yang kacau, pilihan yang tidak kompatibel, dan besarnya biaya yang dikeluarkan tidak sesuai. (5) Kurangnya pemahaman dan pengarahan di antara pengguna, manajemen senior dan IS/IT *specialist* dalam penanganan masalah, perumusan solusi, dan penyalahgunaan sumber daya.

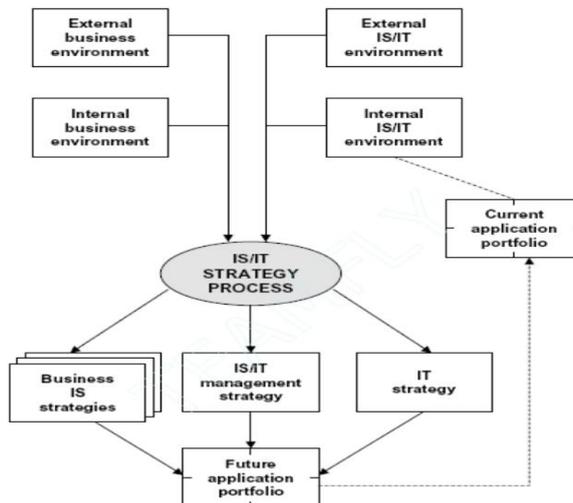
Ward & Peppard (2002, p.44) mengungkapkan bahwa strategi sistem informasi harus mampu memenuhi kebutuhan suatu organisasi atau kebutuhan atas informasi dan sistem untuk mendukung strategi bisnis suatu organisasi. Penyampaian informasi tidaklah mudah karena informasi haruslah dihasilkan dan disampaikan kepada mereka yang benar-benar membutuhkannya baik dari sisi waktu dan kegunaannya. Menurut Ward & Peppard (2002, p.26) strategi sistem mempunyai empat karakteristik umum, yaitu: (1) Penyebaran informasi melalui sistem teknologi dengan *customers*, *consumers*, dan *suppliers* dan mengubah sifat hubungan; (2) Menghasilkan integrasi yang lebih efektif dari penggunaan informasi dalam proses penambahan *value* perusahaan; (3) Memungkinkan suatu organisasi untuk mengembangkan, memproduksi, memasarkan, serta mengirimkan produk atau jasa baru berdasarkan informasi; (4) Menyediakan informasi bagi manajemen eksekutif guna mendukung pengembangan dan pelaksanaan strategi (khususnya di mana informasi eksternal dan internal yang relevan terintegrasi dalam analisis).

Tujuan utama mengembangkan strategi sistem informasi adalah untuk mengidentifikasi nilai tambah dari portofolio aplikasi yang akan meningkatkan performa suatu organisasi/perusahaan. Hal ini dapat diperoleh dengan menghubungkan kebutuhan sistem informasi pada strategi bisnis dengan eksplorasi kesempatan untuk IS/IT untuk membentuk strategi bisnis. Tujuan dari strategi IS/IT seharusnya tidak berfokus pada orientasi obyek, teknologi *database*, internet, HTML, spesifikasi *hardware*, atau *end user* karena seluruh aspek tersebut termasuk dalam permasalahan dalam implementasi. Ciri khas

dari strategi sistem informasi harus sesuai dengan fokus bisnis, seperti meningkatkan pelayanan pelanggan, meningkatkan produktivitas atau menyediakan diferensiasi produk. Tantangan yang timbul dari tujuan tersebut adalah bagaimana mendefinisikan dan mengukur dampak strategi serta bagaimana relasi antara penyusunan strategi sistem informasi terhadap performa perusahaan.

Untuk mencapai keberhasilan strategi, perlu diperhatikan dan dianalisis faktor-faktor apa saja yang memungkinkannya. Ward & Peppard (2003, p.35), mempunyai pendapat bahwa faktor keberhasilan dalam strategi sistem informasi, terdiri dari: (1) Fokus pada eksternal bukan internal: memperhatikan pelanggan, pesaing, *supplier*, bahkan bisnis lainnya, serta hubungan bisnis dan kesamaan dengan dunia bisnis di luar. Secara tradisional IS/IT difokuskan pada proses internal dan masalah-masalah yang ada; (2) Nilai tambah, bukan penurunan biaya: meskipun penurunan biaya yang mungkin bertambah disebabkan oleh ekspansi bisnis dengan biaya marginal berkurang. Hal ini konsisten dengan persyaratan perusahaan untuk membedakan diri dengan produk dan jasa pesaing yang lebih baik; (3) Berbagi keuntungan dalam suatu organisasi, dengan *supplier*, *customer*, *consumer*, dan juga *competitor*. Dalam beberapa kasus yang telah terjadi, tidak dilakukan pembagian keuntungan dalam suatu organisasi, tetapi keuntungan yang diperoleh digunakan untuk memperluas organisasi. Hal ini akan mengurangi keuntungan dan tidak memungkinkan untuk bertahan.

Dalam melakukan perencanaan strategi sistem informasi, Ward & Peppard mengenalkan sebuah model yang dijabarkan pada Gambar 3.1 dibawah ini.

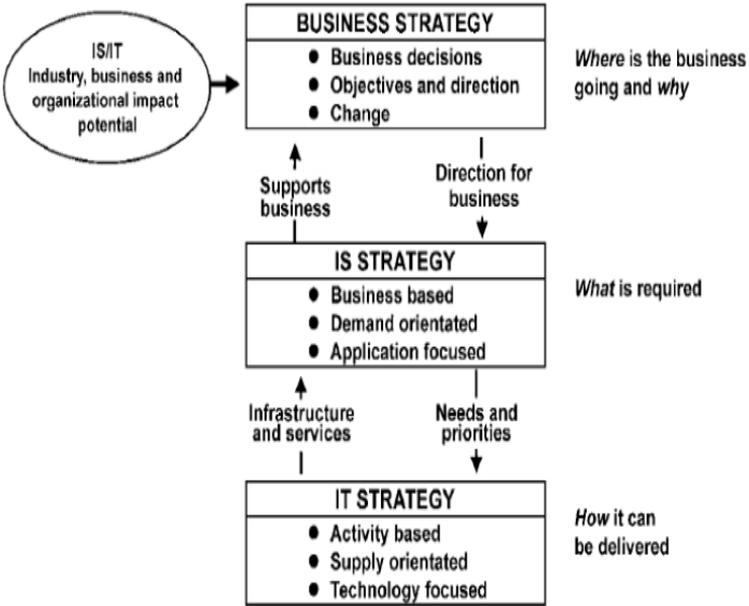


Gambar 3.1 Model penyusunan strategi IS/IT

Input meliputi: (1) *internal business environment* – strategi bisnis saat ini, tujuan, sumber daya, proses, dan nilai budaya kerja; (2) *external business environment* – ekonomi, industri dan iklim persaingan dalam suatu organisasi; (3) *the internal IS/IT environment* – sudut pandang IS/IT dalam bisnis, tahap puncak, cakupan bisnis dan kontribusi, kemampuan, sumber daya dan infrastruktur teknologi; (4) *the external IS/IT environment* – *trend* teknologi dan peluang serta penggunaan IS/IT terutama bagi *customers, competitors,* dan *suppliers*.

Output meliputi: (1) IS/IT strategi manajemen – aspek umum dari suatu strategi yang diterapkan organisasi, memastikan kebijakan yang konsisten jika diperlukan; (2) *business IS strategies* – bagaimana setiap unit akan menerapkan IS/IT untuk mencapai tujuan bisnis; (3) strategi IT – kebijakan dan strategi untuk mengatur teknologi dan sumber daya ahli.

Model penyusunan perencanaan sistem informasi harus disesuaikan dengan strategi bisnis dan perencanaan strategi sistem informasinya. Adapun hubungan antara strategi bisnis dengan strategi sistem informasi dan teknologi informasi dapat dilihat pada Gambar 3.2 di bawah ini:



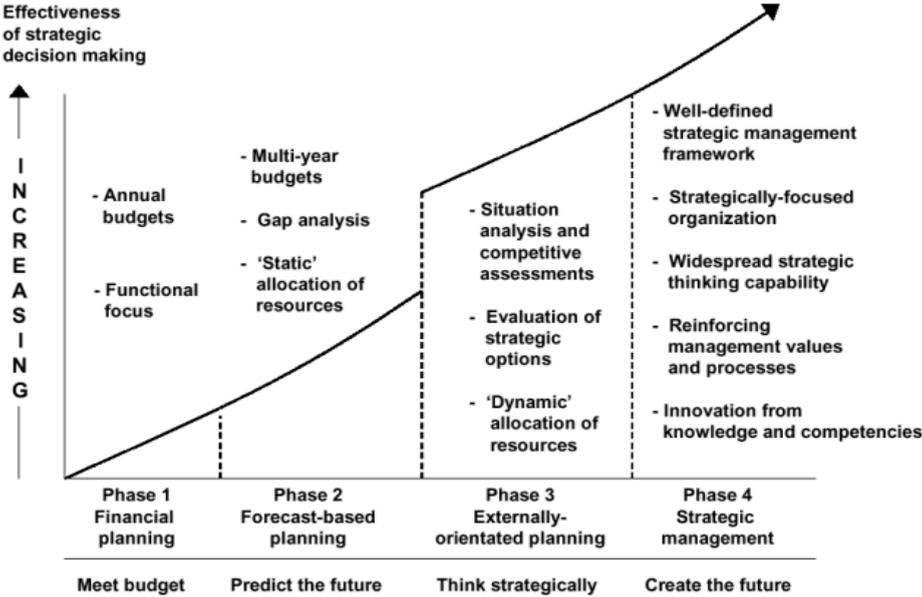
Gambar 3.8 Hubungan bisnis dengan strategi IS/IT

Selanjutnya, mengidentifikasi efek potensial pertama, kemudian mengevaluasi informasi dan sistem yang diperlukan untuk dihubungkan dengan strategi dan menentukan bagaimana cara memperoleh sistem informasi melalui teknologi. Dalam menentukan strategi IS/IT yang

diharapkan dapat mendukung visi dan misi suatu organisasi, diperlukan suatu pemahaman yang mendalam mengenai strategi bisnis organisasi. Adapun pemahaman yang diperlukan, di antaranya: mengapa suatu bisnis dijalankan, kemana arah tujuan bisnis perusahaan, kapan target pencapaian tujuan, bagaimana cara mencapai tujuan dan adakah perubahan yang perlu dilakukan. Oleh karena itu, perlu adanya penyesuaian seluruh aspek tersebut dengan strategi bisnis organisasi.

Seluruh organisasi/perusahaan memiliki beberapa bentuk strategi, baik secara implisit maupun eksplisit, di mana inti dari strategi bisnis tersebut adalah menciptakan keunggulan kompetitif yang lebih cepat dari pesaingnya. Evolusi strategi dan strategi *planning* suatu perusahaan dikembangkan dalam suatu model untuk menjelaskan peningkatan suatu perusahaan (Gambar 3.3). Seluruh organisasi/perusahaan memiliki beberapa bentuk strategi, baik secara implisit maupun eksplisit, di mana inti dari strategi bisnis tersebut adalah menciptakan keunggulan kompetitif yang lebih cepat dari pesaingnya. Evolusi strategi dan strategi *planning* suatu perusahaan dikembangkan dalam suatu model untuk menjelaskan peningkatan suatu perusahaan (Gambar 3.3). Meskipun terdapat beberapa perubahan, sejak tahun 1980 model ini sudah bisa menggambarkan bagaimana inti permasalahan yang terlibat, sejalan dengan pendekatan untuk mengembangkan dan mengimplementasikan strategi. Pada fase pertama perusahaan berfokus pada aliran kas dan perencanaan keuangan tahunan dan melibatkan beberapa teknik sederhana untuk mengembangkan *medium-term budgets*. Sedangkan pada fase kedua, difokuskan pada percobaan prediksi – meramalkan apa yang mungkin terjadi pada tiga atau empat tahun ke depan. Hal ini bisa dilakukan dengan mengacu pada *historical performance*, analisis dan proyeksi ke depan menggunakan *internal trend* dan parameter eksternal, seperti ekonomi dan riset data pasar yang bisa meramalkan penjualan dan perkembangan pasar serta memprediksi efek terhadap pemasukan dan biaya. Pada fase ketiga, organisasi untuk pertama kalinya mengharapkan eksternal untuk memperoleh keuntungan melalui pemahaman terhadap *nature* dari persaingan dalam industrinya, dengan menilai dan sadar terhadap rintangan potensial serta posisi perusahaan untuk memperoleh keuntungan kompetitif. Organisasi harus merevisi portofolio produk untuk menyesuaikan dengan permintaan dalam sektor pasar yang lebih atraktif, atau meningkatkan nilai/fitur pada produk dan servis, atau dengan mengurangi biaya produksi. Sedangkan pada fase keempat, organisasi diarahkan pada inovasi dan berubah menjadi organisasi yang mampu membentuk lingkup bisnisnya sendiri. Organisasi harus mampu merespon perubahan pasar dari sisi produk baru dan persaingan yang ketat. Nilai suatu organisasi, budaya dan struktur

harus bisa mendorong proses dan kompetensi yang dibutuhkan untuk berkembang dan menjadi pimpinan dalam industri. Untuk tetap bisa menjadi pemimpin dalam pasar diperlukan inovasi secara berkelanjutan. Ketika beberapa organisasi mampu membentuk strategi yang kreatif, perlu juga dilakukan pengawasan terhadap keadaan lingkungan sekitar.

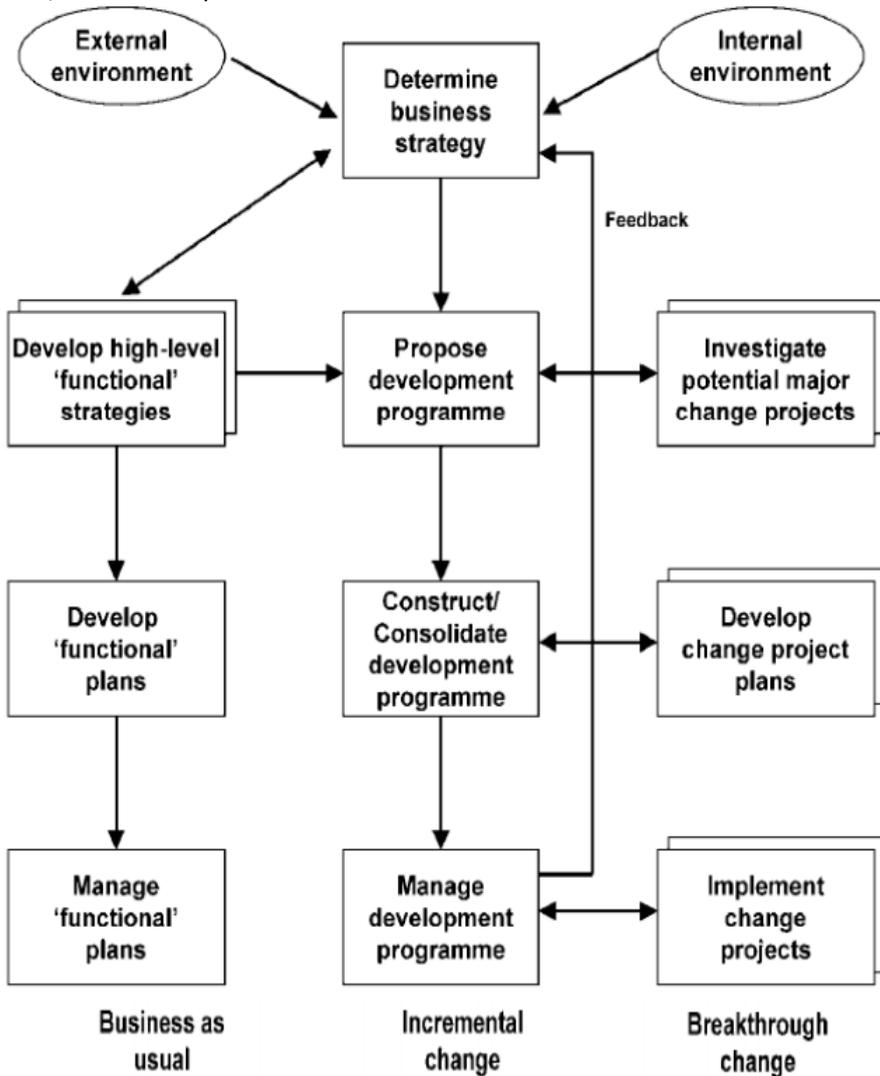


Gambar 9.3 Evolution of strategic management maturity

Beberapa teknik analisis perumusan strategi dibutuhkan untuk berfokus pada permasalahan strategi yang ada, seperti analisis pesaing, keunggulan portofolio produk, atau hubungan dari beberapa aspek terhadap pengambilan keputusan. Meskipun demikian, terdapat konteks yang luas dalam teknik dan *tool* yang digunakan, dengan kata lain sebagai kerangka kerja strategis.

Kerangka kerja melibatkan beberapa faktor dalam pengaturan strategi bisnis yang terbagi menjadi tiga *layer*, yaitu: lingkungan eksternal, tekanan kelompok (pemegang saham, kompetitor, pelanggan, karyawan, pemerintah, dll.), dan *stakeholders*, strategi bisnis internal dan perencanaan. Setelah mengetahui faktor yang perlu diperhatikan dari lingkungan eksternal dan ancaman serta kesempatan yang diperoleh dari tekanan kelompok, organisasi/perusahaan harus mengidentifikasi, mengevaluasi, dan menentukan strategi yang tepat untuk mencapai tujuan bisnis dengan perencanaan dan penggunaan sumber daya yang efektif. Kunci masalah dalam beberapa proses strategi adalah menentukan lingkup. Apakah bisa

mencakup keseluruhan perusahaan atau hanya bagian kecil saja? Menganalisis strategi bisnis memerlukan beberapa tuntutan, di antaranya: (1) Mengidentifikasi strategi yang digunakan saat ini dan sebelumnya, (2) Menginterpretasikan dan menganalisis strategi, dan menggambarkan struktur metode, (3) Menyusun dan mengkonfirmasi kebutuhan IS. Berikut contoh kerangka kerja strategi bisnis yang terintegrasi menurut Ward & Peppard (2002, Gambar 3.4).



Gambar 3.10 *Integrated business strategy framework*

C. TRANSFORMASI BISNIS

Dewasa ini, teknologi hampir mampu membuat segalanya menjadi mungkin (Royana, 2018). Sebut saja sebagai contoh bagaimana perkembangan teknologi yang disematkan pada ponsel dalam satu dekade ini, kehadiran layanan *cloud*, teknologi sensor, kemampuan analisis pada Big Data, serta *Internet of Things* (Das *et al.*, 2016). Teknologi memberikan efek kombinatorial yang mempercepat kemajuan diberbagai aspek baik dibidang bisnis maupun kehidupan bermasyarakat secara eksponensial (WEF, 2017). Ini adalah konteks dimana inovasi digital kini “menggangu” bisnis dan model operasi, dan membuat beberapa dampak yang signifikan terhadap kehidupan sosial masyarakat, (Head, 2016).

Era digital merupakan revolusi yang sedang terjadi di dunia. Revolusi digital ini didorong oleh 4 (empat) teknologi yang telah disebutkan di atas semakin berkembang beberapa tahun ini yang terus memberikan dampak yang signifikan di dalam ekonomi global (Das *et al.*, 2016). Menurut McKinsey (2016) revolusi digital juga sedang terjadi di Indonesia, walaupun ditemukan di Indonesia sedikit agak lambat dalam mengadopsi potensi digital jika dibandingkan dengan negara lain.

Digitalisasi merupakan salah satu kunci penting dalam peningkatan produktivitas, yaitu dengan membangun teknologi digital seperti *remote sensors*, *intelligent machine*, *big data*, dan *real time communication* yang meningkatkan efisiensi proses, kualitas produk dan layanan, dan optimalisasi alokasi sumber daya, sehingga mampu mengurangi waktu proses menjadi lebih cepat, operasional yang lebih ramping, dan kepuasan pelanggan yang lebih baik (Das *et al.*, 2016).

Transformasi digital adalah perubahan organisasi yang melibatkan orang, proses, strategi, struktur, melalui penggunaan teknologi dan model bisnis untuk meningkatkan kinerja (Westerman *et al.*, 2011). Transformasi digital akan melakukan banyak sekali inovasi yang mengubah perusahaan menjadi lebih efektif dan efisien di dalam menjalankan bisnis (Westerman George, Bonnet Didier, 2014).

Transformasi digital juga didefinisikan sebagai penggunaan teknologi yang secara radikal meningkatkan kinerja atau pencapaian tujuan perusahaan (Westerman *et al.*, 2011), transformasi digital membawa serta banyak tantangan bahwa organisasi harus mempertimbangkan lebih hati-hati dari sebelumnya, (Fitzgerald *et al.*, 2013).

Laju perkembangan teknologi meningkatkan tekanan pada organisasi untuk melakukan transformasi digital karena para pesaing terus-menerus menyesuaikan teknologi baru, teknologi juga telah memberdayakan konsumen untuk lebih menuntut dan menetapkan standar kepuasan yang

lebih tinggi dari biasanya, misalnya, layanan yang lebih baik dan lebih cepat kapanpun dan dimanapun dan dengan perangkat apapun (Fitzgerald *et al.*, 2013).

Menurut McKinsey (2016), banyak sektor industri yang akan bertransformasi ke arah penggunaan digital. Sebagai contoh misalnya industri *retail store* yang digital akan membutuhkan hampir tidak ada sumber daya manusia, sehingga memungkinkan pelanggan untuk berbelanja di toko dan langsung menerima notifikasi promosi yang sesuai dengan historis pembelinya, penggunaan *augmented reality* akan mempermudah cara pelanggan mencoba produk yang ditawarkan, kemudian pembayarannya pun akan berlangsung secara otomatis, dan rak-rak akan melakukan penambahan dan pengurangan stok secara otomatis berdasarkan analisis pelanggan dan data penjualan.

Hampir setiap keputusan strategis yang pilih oleh kebanyakan organisasi sangat bergantung pada teknologi untuk berhasil, pemahaman dan komunikasi persyaratan teknis, menjadi sangat penting bagi pengambil keputusan bisnis untuk membuat keputusan terbaik (Ross, J. W., Weill, P., & Robertson, 2007). Namun, berdasarkan penelitian terdahulu, para pengambil keputusan sering kali kurang memahami gambaran besarnya sehingga dalam membuat perencanaan strategi dan pengembangan bisnisnya sering mengalami kesulitan, (Westerman George, Bonnet Didier, 2014).

Studi yang komprehensif mengenai topik ini telah dilakukan oleh MIT *Center for Digital Business* dan Cap Gemini *Consulting*. Westerman mewawancarai 157 eksekutif dari 50 perusahaan di 15 negara dan di 8 industri selama beberapa tahun dan Fitzgerald *et al.* mensurvei 1559 eksekutif dari seluruh dunia mengenai topik ini.

Penelitian yang dilakukan oleh Westerman dkk. telah mengidentifikasi bahwa transformasi digital menyentuh organisasi di tiga bidang utama, yaitu Pengalaman Pelanggan, Proses Operasional, Model bisnis.

Pengalaman Pelanggan dipecah menjadi tiga sub segmen: *customer understanding, top line growth and customer touch points*. Gagasan menarik tentang bagian ini adalah bagaimana lebih memahami pengalaman pelanggan. Disini Westerman dkk. memaparkan gambaran yang lebih komprehensif dengan menambahkan unsur-unsur penjualan digital dan proses pemasaran (*top line growth*), serta sarana baru untuk memahami dan berinteraksi dengan pelanggan di ranah digital.



Gambar 3.11 Building blocks of the digital transformation

Pada bagian ini, transformasi yang paling jelas adalah cara baru dalam berinteraksi dengan pelanggan di ranah digital dengan pemanfaatan web dan seluler, ini memungkinkan perusahaan mampu memahami pelanggan lebih baik dan bahkan menghadirkan *co-creation* atau penciptaan bersama (Paul Gray, Quarterly, 2013). Fenomena menarik yang terjadi dewasa ini adalah tentang pengalaman pelanggan, karena digital menyediakan alat baru untuk dapat lebih mengenali pelanggan secara efektif dalam skala besar serta dapat mengembangkan dan meningkatkan pengalaman pelanggan dengan lebih baik (Lin, Hsia and Wu, 2009).

Organisasi juga dapat menggunakan teknologi digital untuk meningkatkan dan mengotomatisasi **proses operasional**. Westerman dkk. membagi transformasi proses operasional menjadi tiga segmen: proses digitalisasi, pemberdayaan pekerja dan manajemen kinerja.

Dari ketiganya, proses digitalisasi adalah yang paling jelas. Tidak ada yang baru dengan ini, setelah semua perusahaan membeli sistem ERP dan sejenisnya, tepatnya untuk tujuan penyelarasan IT dengan Bisnis, dan otomatisasi proses bisnis, dalam jumlah besar sejak tahun 1990an (Rettig, 2007). Namun, seiring kemajuan teknologi, kemungkinan baru masih terus

terbuka untuk mengembangkan dan mengotomatisasi proses bisnis lebih jauh lagi.

Begitu proses dan semua datanya ada dalam format digital, maka sangat memungkinkan organisasi untuk memantau dan mengelola kinerjanya dengan lebih akurat dengan bantuan *Key Performance Indicators* (KPI). Segala informasi penting perusahaan dapat lebih mudah dibagi ke seluruh organisasi untuk meningkatkan transparansi serta bisa digunakan dalam membantu pengambilan keputusan.

Namun, pemberdayaan pekerja sebagai yang paling menarik dari ketiganya. Ada banyak transformasi untuk itu. Pertama, ini tentang memberi karyawan alat untuk bisa melakukan pekerjaan dengan lebih baik. Bagian terpentingnya adalah tentang menjadi lebih mampu dalam berbagi pengetahuan dan berkomunikasi lintas batas organisasi untuk saling berkolaborasi. Ini adalah tantangan di semua organisasi besar dan menjadi hal yang sangat penting bagi karyawan untuk dapat melakukan yang terbaik dalam peningkatan kinerjanya.

Bidang ketiga transformasi adalah bahwa **model bisnis**. Digitalisasi dapat memungkinkan untuk merambah ke pasar global atau perluasan pasar baru. Bagian ketiga dari area ini adalah penggunaan digital untuk memberikan sesuatu yang baru bagi bisnis yang ada, dalam bentuk penambahan nilai, penambahan produk atau jasa dengan komponen digital.

Menggarisbawahi tiga bidang transformasi digital sebagaimana dipaparkan diatas adalah bidang yang disebut digital *capabilities*. Ini adalah dasar untuk dapat melaksanakan transformasi digital. Kemampuan ini bukan hanya tentang memiliki sistem TI yang tepat, namun merupakan konsep yang jauh lebih holistik. Ini mencakup komponen seperti seberapa baik TI & bisnis selaras, seperti apa data dan proses organisasi diintegrasikan, serta kemampuan aktual dari organisasi untuk memberikan solusi bagi kebutuhan yang muncul.

Dari sudut pandang yang lebih praktis, sebuah organisasi dapat menggunakan model tersebut untuk merencanakan sudut pandang pendekatan mereka dalam penerapan transformasi digital. Perusahaan mengawali inisiasi transformasi digital mulai dari pengalaman pelanggan dan kemudian memikirkan cara mengoptimalkan proses operasional dan menyempurnakan model bisnis.

D. PERKEMBANGAN TEKNOLOGI DIGITAL

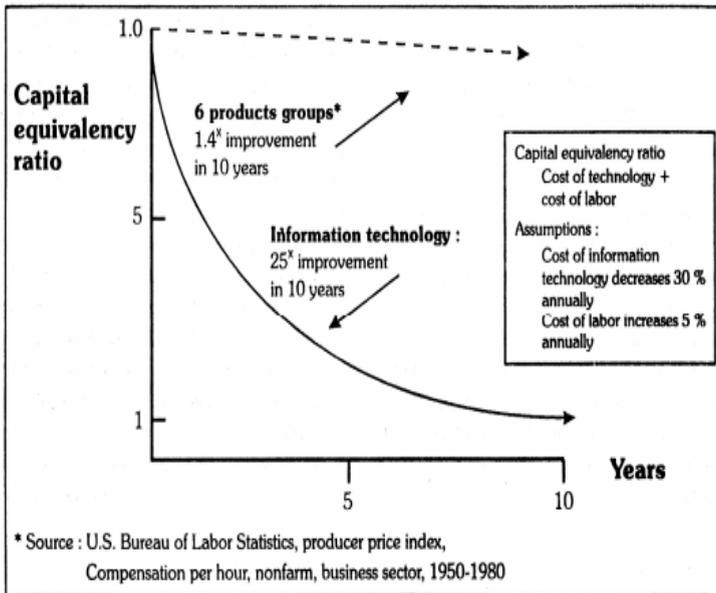
Perubahan lingkungan bisnis yang dinamis dan *turbulent* yang berinteraksi dengan perkembangan teknologi informasi telah menyebabkan transformasi bisnis dan organisasi. Berbagai studi dan penelitian telah

menghasilkan kerangka untuk menjadi pedoman bagi bisnis dalam menyikapi dengan sebaik-baiknya Teknologi Informasi tersebut (Indriantoro, 1996).

Istilah teknologi informasi bisa mempunyai konotasi yang berbeda antar orang yang membicarakan. Penulis menggunakan istilah teknologi informasi dengan mengacu pada definisi yang digunakan oleh *The Management in the 1990s Research Program* yang diselenggarakan oleh MIT Sloan School of Management yaitu enam elemen yang semakin terintegrasi dan berevolusi:

- 1) Perangkat-keras. Ini berkisar dari *mainframe* skala besar sampai *microcomputer* skala kecil.
- 2) Perangkat-lunak. Ini berkisar dari bahasa tradisional seperti COBOL dan ekuivalen generasi keempatnya sampai ke *expert system* yang muncul seiring dengan perkembangan dalam *artificial intelligence*.
- 3) Jaringan. Jaringan telekomunikasi ini berkisar dari publik ke *private*, *broadband* ke *narrowband*.
- 4) *Workstation*. Ini berkisar dari yang dirancang untuk insinyur, dengan kapabilitas komputasi besar dan kemampuan untuk menyajikan grafik warna tiga-dimensi, sampai *workstation* profesional yang digunakan oleh karyawan bagian pinjaman suatu bank atau analis pasar perusahaan *consumer goods*. Yang terakhir ini mengandalkan model, heuristik, dan grafik sederhana dan sering mempunyai *database* yang sangat besar dalam sistemnya.
- 5) Robotik. Ini berkisar dari robot dengan "pandangan" dan "tangan" yang digunakan di pabrik ke berbagai alat yang sangat biasa bagi umumnya orang, seperti *automatic teller machine* (ATM).
- 6) *Smart chips*. *Chips* "pintar" yang digunakan pada produk untuk meningkatkan fungsionalitas atau reliabilitas. Misalnya, yang digunakan pada sistem rem mobil untuk mencegah tergelincir dan pada elevator untuk memperbaiki kecepatan respons dan untuk mendeteksi kerusakan. Dalam bentuk yang paling sederhana sekarang ini muncul dalam produk seperti *active card* yang digunakan untuk menelusuri melalui satelit pergerakan kontainer yang dikapalkan.

Perkembangan teknologi informasi tersebut selalu dikaitkan dengan semakin meningkatnya *power* dan pada saat yang bersamaan menurunnya biaya sehingga menghasilkan kombinasi *price/performance* yang semakin baik. Gambar 3.6 berikut ini menunjukkan *trend* tersebut.



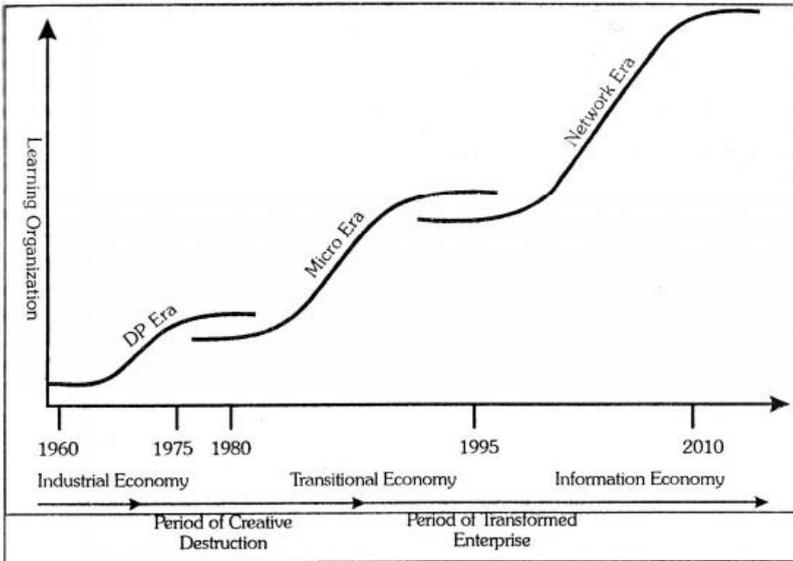
Gambar 3.12 Rasio Ekuivalensi Kapital: Teknologi Informasi vs Enam Kelompok Produk

Salah satu kerangka yang sering digunakan untuk mempelajari perkembangan teknologi informasi adalah dengan menggunakan *stages theory*. *Stages Theory* yang dikembangkan oleh Nolan misalnya, (Bradley *et al.*, 1993; Nolan & Crosson, 1995) membagi perkembangan teknologi informasi kedalam tiga tahap yaitu Era DP (*data processing*), Era *Micro*, dan Era *Network* yang diwujudkan dalam tiga kurva-S dalam durasi 15 sampai dengan 20 tahun (lihat gambar 3.7). Teori Perkembangan Nolan ini kemudian akan disampaikan secara lengkap.

1. Era DP

Era DP, kira-kira dari tahun 1960 sampai dengan tahun 1980, didominasi oleh produk industri yang mendukung komputasi dengan *mainframe*. *Minicomputer* dikelola dengan cara sentralisasi seperti halnya dengan *mainframe*. Permintaan, atau "pasar internal," dalam perusahaan untuk komputasi *mainframe* adalah otomasi klerikal tingkat rendah dan pekerjaan pabrik. Paradigma aplikasi yang dikembangkan adalah "otomasi," dimana komputer digunakan untuk membuat organisasi lebih efisien. Yang paling sering adalah mengotomasikan proses dan prosedur yang ada di perusahaan; jarang sekali ini mengakibatkan perubahan fundamental pada struktur organisasi hierarkis perusahaan. Setelah melakukan investasi selama 10 sampai dengan 15 tahun, masa kritis otomasi dicapai dimana pada saat itu sejumlah perusahaan akan melakukan *down-sizing* tenaga kerja pabrik dan

klerikal sebanyak kira-kira 20% sampai 30%, yang menyebabkan struktur biaya lebih rendah dalam industri tersebut. Penyusutan tenaga kerja pabrik dan klerikal dimulai tahun 1970 di hampir semua industri dan secara agresif berjalan terus sampai tahun 1980an.



Gambar 3.13 Stages Theory

Dengan permintaan *mainframe* komputasi menurun pada pertengahan tahun 1970an, seperti dicerminkan oleh menurunnya pertumbuhan dalam kurva-S Era DP, industri komputer mulai menjajaki permintaan teknologi berbasis komputer di tingkat menengah dalam organisasi. Karena tingkat menengah ini terdiri dari profesional, seperti insinyur, akuntan, dan manajer, permintaannya secara fundamental berbeda dengan komputasi di tingkat bawah organisasi. Paradigma otomasi Era DP biasanya diupayakan pada tingkat ini, dan biasanya gagal. Kerja yang dilakukan oleh profesional tingkat menengah cenderung terlalu rumit dan esoterik untuk mengotomasikan dengan cara yang sama pada waktu kerja pabrik dan klerikal diotomasikan.

2. Era Micro

Pada Era *Micro*, paradigma baru diperkenalkan untuk mengatasi masalah profesional tingkat menengah dan membedakannya dengan otomasi. Pada era ini diperkenalkan suatu istilah baru yaitu "*informate*." Tujuan *Informating*, berbeda dengan otomasi, tidaklah untuk mengganti profesional dengan komputasi, tetapi menggunakan komputasi untuk mendongkrak kerja profesional. Mikrokomputer, dengan inovasi terkait seperti pemrograman

tanpa *programmer* (misal: *spreas-heet*, pengolah-kata, *computer-aided design* [CAD], dan sebagainya) dan *graphic user interface*, memicu Era *Micro*. Dengan sebagian besar profesional diperlengkapi dengan mikrokomputer, mulailah pengurangan tenaga kerja secara agresif dan permanen di tingkat menengah organisasi, sama dengan apa yang kita lihat pada tingkat lebih bawah pada pertengahan tahun 1970an dan tahun 1980an.

Pada Era *Micro*, komputer, terutama mikrokomputer, telah menyebabkan permintaan lain yang kurang dipahami di organisasi. Mikrokomputer telah semakin dimasukkan secara langsung ke produk dan jasa perusahaan. Banyak mobil sekarang ini yang memasukkan (*embedded*) ke sistemnya setara dengan 15 mikro-komputer. Kartu kredit diperlengkapi dengan *chip* yang dapat mencatat data dan berkomunikasi dengan jaringan komputer. Semakin banyak produk, mulai dari produk elektronik, peralatan rumah-tangga, mesin *foto-copy*, dan mainan anak-anak sampai dengan tata lampu, sistem keamanan dan pengendalian lingkungan, telah dibuat "lebih pintar" dengan memasukkan mikroprosesor dalam rancangannya.

3. Era *Network* (Jaringan)

Investasi untuk komputer dalam tiga bidang utama atau pasar - otomasi kerja pabrik dan klerikal, *informating* pekerja informasi, dan mendukung pengembangan produk dan jasa pintar - merupakan basis permintaan yang kemudian muncul yaitu permintaan *network* (jaringan). Jaringan *wide-band* muncul atas dasar sistem otomasi. *Local Area Network* (LAN) mula-mula digunakan untuk secara elektronik menghubungkan tim profesional (misalnya insinyur, akuntan, dan eksekutif) kemudian dihubungkan dengan jaringan *wide-band* sehingga memungkinkan kelompok profesional lokal untuk berinteraksi dengan kelompok lokal lain yang terpisah secara geografis melalui *workstations* mereka. Produk seperti mobil dan pesawat-terbang, dirancang oleh kelompok yang dijaringan secara global yang berkolaborasi setiap hari, mengirimkan "cetak-biru" ulang-alik ke seluruh dunia dan mendiskusikannya lewat telepon atau lewat konferensi video atau papan-tulis elektronik.

Bahkan komputer yang dimasukkan dalam produk diintegrasikan ke dalam jaringan, dan kemudian diintegrasikan lebih lanjut ke jaringan elektronik perusahaan tersebut. Langkah kearah jaringan komputer elektronik yang terintegrasi dan saling terkait ini mempunyai dampak yang besar terhadap struktur organisasi dan nilai jasa yang diberikan ke pelanggan.

Meskipun banyak perusahaan yang melakukan investasi besar-besaran dalam komputasi dan LAN, sebagian besar belum secara resmi memikirkan perancangan, manajemen dan ekspektasi global jaringan elektronis ini. Inilah yang akan menjadi faktor yang semakin penting dalam hal bagaimana perusahaan bersaing di tahun 1990an.

A square graphic with a dark gray background and a white border. Inside, the word "BAB" is written in white serif font at the top, and a large white number "4" is centered below it.

BAB 4

MODEL DATA

Dalam konteks merancang model *database*, model adalah pertimbangan penting dari beberapa konsep, bahasa, grafik yang biasanya digunakan dalam mendeskripsikan struktur data dan proses operasi suatu *database*. Fokus dari model data adalah mendeskripsikan daripada metode apa yang digunakan. (Alisha Safira, 2020)

Model data merupakan sekumpulan konsep yang terintegrasi untuk mendeskripsikan data, hubungan antar data, dan batasannya dalam organisasi. Model data harus menyediakan konsep dasar dan notasi yang memungkinkan perancang basis data dan pemakai untuk dapat mengkomunikasikan pemahamannya mengenai organisasi data. Adapun model data tersusun atas beberapa komponen sebagai berikut:

- Bagian struktural → Memuat aturan untuk konstruksi basis data;
- Bagian manipulasi → Melakukan definisi tipe operasi yang diizinkan;
- Aturan mengenai integritas.

Model data dibangun menggunakan tiga bangunan blok, yaitu konsep, bahasa dan grafis. Kata "konsep" di konteks pemodelan data memiliki arti khusus dan unik. Hal ini mengacu tidak hanya untuk ide-ide abstrak, tetapi juga untuk dunia nyata berwujud dan tidak berwujud fitur/fenomena yang relevan dengan kebutuhan informasi *database* pengguna. Sejak identifikasi fitur ini/fenomena di pemodelan data sebagian besar merupakan proses mental, terutama pada tahap awal, kata "Konsep" adalah istilah yang tepat untuk digunakan dalam konteks ini. Sebagai pemodelan data proses berkembang, konsep diidentifikasi menjadi "entitas" dalam bahasa dari model *database* relasional atau "objek" dalam bahasa berorientasi objek

model *database*. Dalam *database*, konsep diidentifikasi adalah sering disebut sebagai "objek data" dan kehadirannya atau kejadian di *database* disebut sebuah "contoh".

Model data dapat dikelompokkan berdasarkan konsep pembuatan deskripsi struktur basis data, yaitu:

- a) Model data konseptual (*high level*) menyajikan konsep tentang bagaimana *user* memandang atau memperlakukan data. Dalam model ini dikenalkan tiga konsep penyajian data yaitu:
 - *Entity* (entitas), merupakan penyajian obyek, kejadian atau konsep dunia nyata yang keberadaannya secara eksplisit didefinisikan dan disimpan dalam basis data, contohnya: Mahasiswa, Matakuliah, Dosen, Nilai dan lain sebagainya.
 - *Attribute* (atribut) berisi keterangan-keterangan yang menjelaskan karakteristik dari suatu entitas seperti: NIM, Nama, Fakultas, Jurusan untuk entitas Mahasiswa.
 - *Relationship* (hubungan) yaitu hubungan atau interaksi antara satu entitas dengan yang lainnya, misalnya entitas pelanggan berhubungan dengan entitas barang yang dibelinya.
- b) Model data fisik (*low level*) merupakan konsep bagaimana deskripsi detail data disimpan ke dalam komputer dengan menyajikan informasi tentang format rekaman, urutan rekaman, dan jalur pengaksesan data yang dapat membuat pencarian rekaman data lebih efisien.
- c) Model data implementasi (*representational*) merupakan konsep deskripsi data disimpan dalam komputer dengan menyembunyikan sebagian detail deskripsi data sehingga para *user* mendapat gambaran global bagaimana data disimpan dalam komputer. Model ini merupakan konsep model data yang digunakan oleh model hierarki, jaringan dan relasional.

Secara garis besar, model data dapat dikelompokkan dalam tiga macam yaitu;

1) Model data berbasis *object* (*object based data model*)

Model data berbasis objek menggunakan konsep entitas, atribut dan hubungan antar entitas. Terdiri dari:

- ***Entity Relationship model***

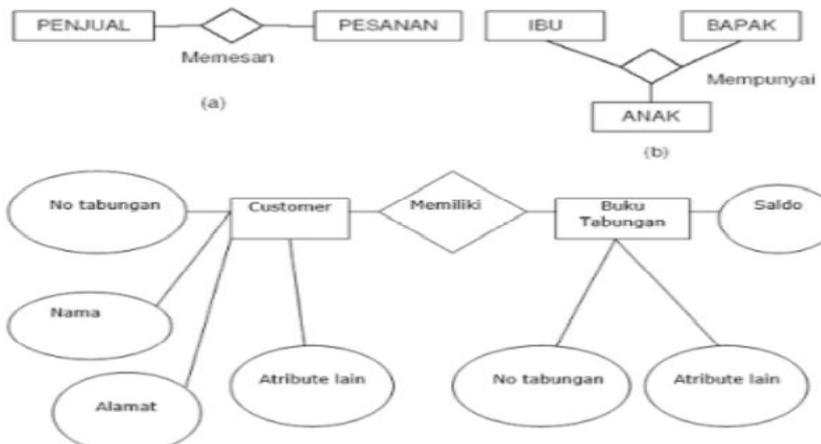
Model *entity-relationship* pertama kali diperkenalkan oleh Peter Chen pada tahun 1976. Dalam pemodelan ini dilakukan dengan tahapan sebagai berikut:

- a) Memilih entitas-entitas yang akan disusun dalam basis data dan menentukan hubungan antar entitas yang telah dipilih.

- b) Melengkapi atribut-atribut yang sesuai pada entitas dan hubungan sehingga diperoleh bentuk tabel normal penuh (ternormalisasi).

Model untuk menjelaskan hubungan antar data dalam basis data berdasarkan suatu persepsi bahwa *real word* terdiri dari objek-objek dasar yang mempunyai hubungan atau relasi antara *object-object* tersebut. E-R MODEL berisi ketentuan/aturan khusus yang harus dipenuhi oleh isi *database*. Aturan terpenting adalah *MAPPING CARDINALITIES*, yang menentukan jumlah *entity* yang dapat dikaitkan dengan *entity* lainnya melalui *relationship-set*.

Antara entitas diasosiasikan dalam suatu hubungan (*relationship*). Suatu relasi dapat memiliki beberapa atribut. Jumlah kelas entitas dalam suatu relasi disebut derajat relasi. Gambar di bawah ini merupakan contoh dari relasi berderajat dua (a) dan relasi berderajat tiga (b).



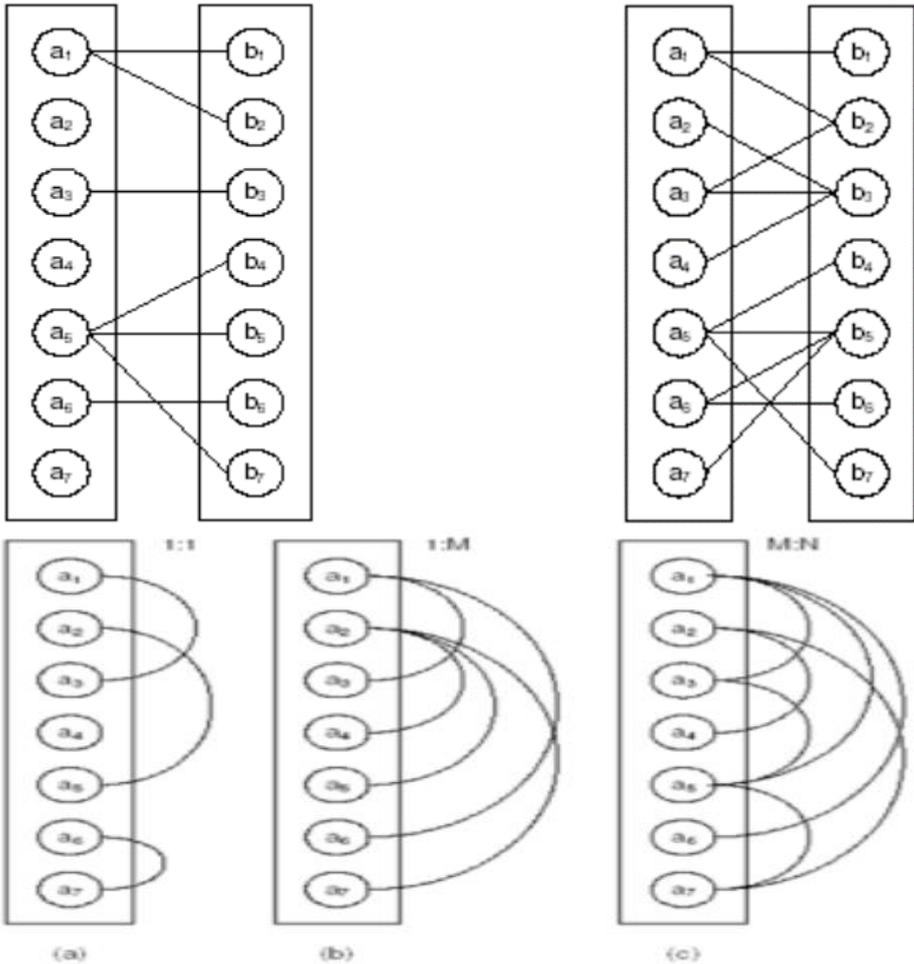
Gambar 4.14 Contoh kasus ER-Model

- **Binary model**

Relasi memiliki tiga tipe biner yaitu:

- One-to-one* (1:1). Hubungan terjadi bila setiap instansi entitas hanya memiliki satu hubungan dengan instansi entitas lain.
- One-to-many* (1:M). Relasi ini terjadi bila setiap instansi dapat memiliki lebih dari satu hubungan terhadap instansi entitas lain tetapi tidak kebalikannya.
- Many-to-many* (M:N). Hubungan saling memiliki lebih dari satu dari setiap instansi entitas terhadap instansi entitas lainnya.

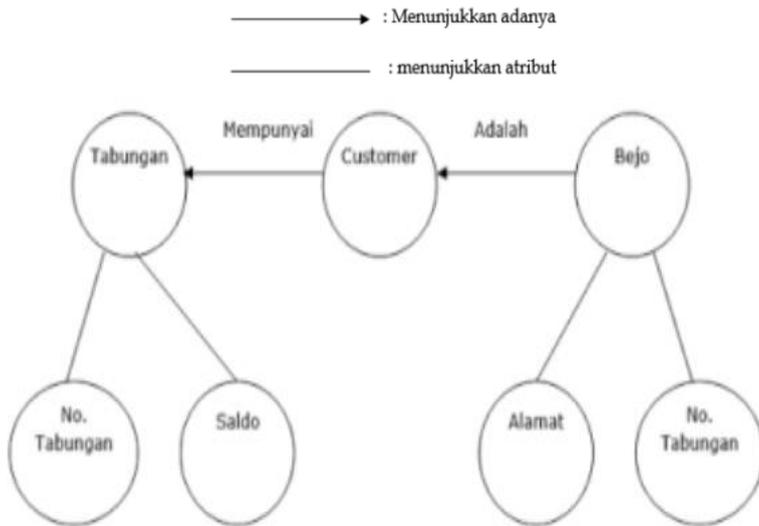
Selain relasi antara dua entitas, terdapat juga hubungan terhadap entitasnya sendiri yang disebut dengan *recursive relationship (self relation)*. Hubungan ini dapat mempunyai tipe biner, seperti yang sudah dibahas sebelumnya.



- **Semantic data model**

Hampir sama dengan *Entity Relationship* model dimana relasi antara objek dasar tidak dinyatakan dengan simbol tetapi menggunakan kata-kata (*Semantic*). *Semantic Model* digunakan untuk menjelaskan hubungan antar data dalam basis data kepada pemakai secara logik. *Semantic* model digambarkan dalam bentuk diagram: **diagram semantic**. Sebagai contoh, dengan masih menggunakan relasi pada Bank X sebagaimana contoh sebelumnya, dalam *semantic* model adalah seperti terlihat pada gambar di

atas. Tanda-tanda yang menggunakan dalam *semantic* model adalah sebagai berikut:



Gambar 4.15 Contoh kasus *Semantic* model

2) Model data berbasis *record* (*record based data model*)

Model ini berdasarkan pada *record* untuk menjelaskan kepada *user* tentang hubungan *logic* antar data dalam basis data. Pada *record based* data model disamping digunakan untuk menguraikan struktur logika keseluruhan dari suatu *database*, juga digunakan untuk menguraikan implementasi dari *system database* (*higher level description of implementation*).

Klasifikasi berdasarkan model data, klasifikasi ini terdiri dari 3 data model pada *record based* data model yaitu model data hierarki, model data jaringan, model data relasional.

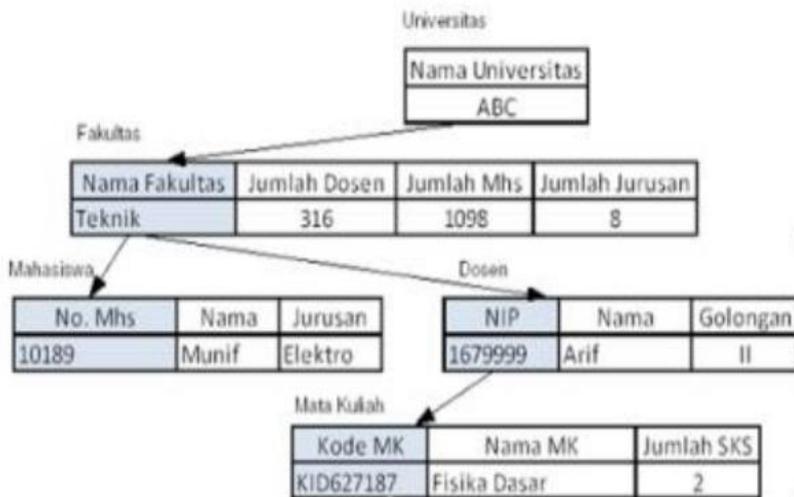
a) Model data hierarki

Dalam model ini, data disusun menurut struktur pohon yang merupakan bentuk lain dari abstraksi data untuk basis data akademik. Pada puncak hierarki disebut dengan akar (*root*). Tiap entitas tingkat atas (*parent*) mempunyai satu atau lebih sub-entitas (*children*) sehingga setiap entitas hanya boleh mempunyai satu induk, tetapi dapat mempunyai banyak anak.

Pada mode data hierarki, hubungan antar entitas dinyatakan dalam satu-banyak (*one to many*) atau satu satu (*one to one*). Dalam satu Universitas terdapat banyak Fakultas dan setiap Fakultas terdapat banyak Dosen atau banyak Mahasiswa, dan seterusnya. Tanda panah menunjukkan derajat keterhubungan “banyak”. Untuk menampilkan semua mata kuliah pada Fakultas tertentu harus dilakukan dalam dua tahap. Yang pertama adalah

menampilkan rekaman semua Dosen yang mengajar di Fakultas tersebut, kemudian baru mata kuliah yang dipegang oleh para Dosen. Dalam hal ini penampilan data terlihat kurang efisien, sebab menggunakan entitas perantara (dosen) yang harus ditampilkan juga. Dikarenakan kunci data yang digunakan untuk menghubungkan antar entitas diberi kode dalam struktur data, maka untuk jumlah entitas perantara yang sedikit masih dapat dikatakan efisien.

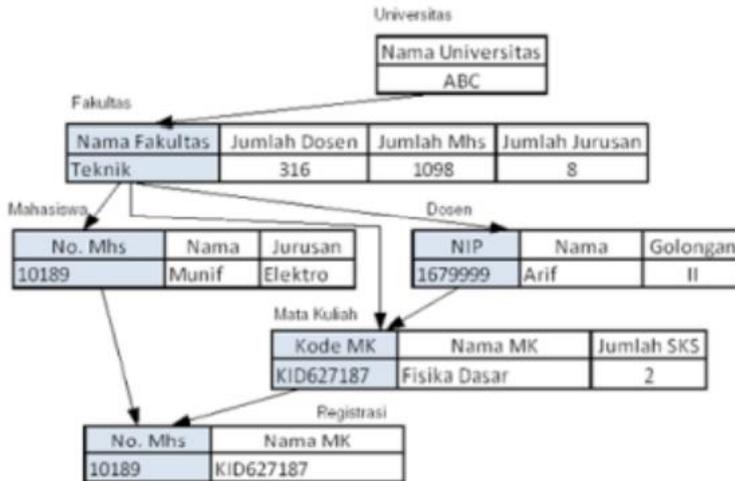
Kelemahan lain pada model data hierarki adalah tidak dapat melakukan pencarian data pada *field*. Misalnya dalam entitas mata kuliah tidak dapat ditampilkan hanya mata kuliah dengan jumlah SKS tertentu, sebab *field* "Jumlah SKS" bukan sebagai kunci data. Hal ini masih dapat dilakukan dengan mengubah struktur data dengan memberi hubungan khusus yang digunakan untuk mengubah struktur *database*. Kelebihan model ini adalah sangat mudah dipahami dan mudah dalam pembaharuan data [Waliyanto2000].



Gambar 4.16 Organisasi rekaman data pada model hierarki

b) Model data Jaringan

Dalam model ini setiap entitas dapat mempunyai banyak induk dan banyak anak. Pada gambar menunjukkan entitas mata kuliah mempunyai dua induk, yaitu langsung berhubungan dengan Fakultas dan Dosen. Dalam model ini lebih sedikit terdapat data rangkap, namun lebih banyak terdapat hubungan antar entitas, sehingga akan menambah informasi hubungan yang harus disimpan dalam *database*. hal ini akan menambah volume dan kerumitan dalam penyimpanan berkas data.



Gambar 4.17 Organisasi rekaman data model jaringan

c) Model data Relasional

Dalam model ini setiap *field* dapat dijadikan kunci data. Data rekaman disusun dari nilai yang berhubungan (*record*). Baris-baris ini akan membentuk tabel yang umumnya tersimpan dalam satu berkas (*file*). Dengan menggunakan model ini, pencarian *field* dari suatu tabel atau banyak tabel dapat dilakukan dengan cepat. Pencarian atribut yang berhubungan pada tabel yang berbeda dapat dilakukan dengan menghubungkan terlebih dahulu tabel-tabel tersebut dengan menggunakan atribut yang sama (*joint operation*). Keuntungan yang didapat dengan menggunakan model ini adalah sebagai berikut [Waliyanto2000]:

- Model ini lebih luwes karena nilai data dalam tabel tidak ada pembatasan dalam berbagai proses pencarian data;
- Model ini mempunyai latar belakang teori matematika;
- Pengorganisasian model relasional sangat sederhana, sehingga mudah dipahami;
- Basis data yang sama biasanya dapat disajikan dengan lebih sedikit terjadi data rangkap (*redundancy data*).

Sedangkan beberapa kelemahan model ini adalah [Waliyanto2000]:

- lebih sulit dalam implementasinya terutama untuk data dengan jumlah yang besar dan tingkat kompleksitasnya tinggi;
- Proses pencarian informasi lebih lambat, karena beberapa tabel tidak dihubungkan secara fisik;

- Dalam manipulasi data yang menggunakan beberapa tabel akan memerlukan waktu yang lama, karena tabel-tabel harus dihubungkan terlebih dahulu.

3) Model data fisik (*physical based data model*)

Digunakan untuk menguraikan data pada internal level. Model ini digunakan untuk menguraikan data di tingkat internal atau menjelaskan kepada pemakai bagaimana data-data dalam basis data disimpan dalam media penyimpanan secara fisik. Model ini jarang digunakan karena kerumitan dan kompleksitasnya yang justru menyulitkan pemakai. Beberapa model yang umum digunakan adalah *Unifying* model dan *Frame memory*.

A square graphic with a dark gray background and a white border. Inside, the word "BAB" is written in white capital letters at the top, and a large white number "5" is centered below it.

BAB
5

INTEGRASI DATA

Integrasi data atau informasi merupakan suatu tindakan yang dilakukan untuk menyatukan atau menggabungkan data atau informasi dari berbagai sumber yang tersebar untuk memenuhi kebutuhan informasi pengguna secara lebih baik (Daromo, 2016). Pada penerapan integrasi data terdapat beberapa kendala yang terjadi salah satu penyebabnya adalah heterogenitas skema (keanekaragaman skema basis data) yang digunakan oleh setiap sistem informasi (Muslih dkk, 2014). Keanekaragaman skema merupakan perbedaan dalam penamaan definisi skema yang meliputi tipe data, format serta presisi data. Keanekaragaman skema basis data ini disebabkan oleh jumlah informasi yang disimpan dalam basis data terus bertambah, sehingga menyebabkan perlunya data tersebut disimpan kedalam beberapa basis data yang berbeda dan integrasi data menjadi salah satu aspek penting dalam menjaga konsistensi diantara basis data tersebut.

Integrasi data merupakan proses yang dilakukan untuk menggabungkan beberapa dari berbagai sumber data yang berbeda sehingga dapat mendukung pengguna dalam melihat kesatuan data (Lenzerini, 2002). Proses tersebut dapat terjadi diberbagai macam proses bisnis pada suatu lembaga atau bagian. Suatu lembaga atau organisasi membutuhkan integrasi data karena organisasi tersebut akan membutuhkan data-data dan informasi dari bagian-bagian yang berbeda yang berada pada organisasi tersebut. Penerapan integrasi data yang tepat dapat mendukung rencana dan pengembangan suatu lembaga yang nantinya akan memberikan nilai tambah berupa *competitive advantage* dalam menghadapi persaingan (Wuryanto, 2018).

Integrasi data merupakan keterkaitan data antara beberapa sistem maupun sub sistem yang terhubung ke dalam suatu penyimpanan data seperti data *warehouse* dimana dengan adanya integrasi data tersebut dapat memudahkan dalam hal berbagi data dan analisis data untuk mendukung pengelolaan informasi (Giordano, 2011). Dalam penerapan proses integrasi data terdapat beberapa permasalahan yang dihadapi. Untuk dapat menyelesaikan masalah tersebut dapat dikelompokkan kedalam beberapa cara, adapun salah satunya adalah integrasi basis data. Integrasi basis data dilakukan berdasarkan skema basis data yang bertujuan untuk membangun pemetaan serta pencocokan antar skema basis data.

Schema matching (pencocokan skema) adalah bagian dari *enterprise application integration* (EAI) yang mempunyai tugas untuk mengatasi berbagai permasalahan integrasi yang disebabkan oleh keanekaragaman skema basis data (Sutanta dkk, 2015). *Schema matching* dimanfaatkan untuk menemukan hubungan serta kesamaan antar elemen dari beberapa skema basis data (Martinek, 2009). *Schema matching* memiliki peranan penting dalam aplikasi yang membutuhkan interoperabilitas diantara sumber data yang beragam.

Schema matching dapat digunakan dengan berbagai jenis metode antar lain seperti *linguistic*, *instance* dan lain sebagainya serta dapat juga digabungkan baik secara *composite* maupun *hybrid*. *Schema matching* dapat dilakukan dengan cara individu dan gabungan atau kombinasi. *Schema matching* individu dilakukan berdasarkan skema dan data yang digunakan sedangkan dengan cara kombinasi dapat dilakukan secara *hybrid* atau *composite*. *Schema matching* secara *hybrid* adalah dengan menggabungkan beberapa pencocokan skema individu ke dalam satu algoritma secara bersamaan, sedangkan *composite*, menggunakan pencocokan skema individu secara individu untuk mendapatkan nilai kemudian dicocokkan serta dikombinasikan hasil yang telah didapat dari masing-masing pencocokan skema individu.

Proses dalam *schema matching* dapat melibatkan berbagai macam algoritma, misal untuk menentukan data yang akan dicocokkan, transformasi pemetaan, atau melakukan penggabungan. Berdasarkan algoritma yang digunakan, model *schema matching* dapat diklasifikasikan ke dalam beberapa kategori.

Studi pustaka ini menemukan 34 model dan prototipe *schema matching* dalam 71 publikasi ilmiah (dkk, 2015). Masing-masing model dan prototipe dapat menggunakan *input* berupa **schema model relasional (RDF/Relational Database Format)**, **model XML (DTD/Document Type Definition atau XSD/W3C XML Schema)**, atau **model ontologi (OWL/Web Ontology Language)**.

Klasifikasi model *schema matching* oleh Madhavan dkk (2001) dan Rahm dkk (2001) terdiri atas *schema based vs. instance based, element vs. structure granularity, linguistic based, constraint based, matching cardinality, auxiliary information*, serta *individual vs. combinational*. Klasifikasi model *schema matching* menurut Do (2005) terdiri atas *schema based, instance based, dan reuse oriented*. Klasifikasi menurut Shvaiko (2005) membagi model berdasarkan level dan jenis informasi yang dieksplorasi meliputi elemen *schema* dan struktur *schema*, dan berdasarkan jenis informasi yang dieksplorasi meliputi aspek terminologi (melibatkan aspek linguistik (terdiri atas *language based* dan *linguistic based*) atau tidak melibatkan aspek linguistik (*string based*), berdasarkan aspek struktural (meliputi aspek internal (*constraint based*) dan *relational* (terdiri atas *alignment reuse, graph based, taxonomy based, serta repository structure*), dan semantik (terdiri atas *upper level formal ontology* dan *model based*). Menurut Cruz dkk (2009), model *schema matching* diklasifikasikan berdasarkan level komponen yang dicocokkan (konseptual dan struktur), tingkat intervensi pengguna (manual dan otomatis), metode yang digunakan (*stand-alone* dan gabungan), dan tipe komponen yang digunakan sebagai dasar pencocokan (menggunakan *schema* atau menggunakan *schema* dan *instance*). Klasifikasi model *schema matching* menurut Karasneh dkk (2009) terdiri atas RSM (*relations schemas matcher*), ANM (*attribute relations name matcher*), DTM (*data type matcher*), CM (*constraint matcher*), dan IDM (*instance data matcher*). Dengan cara yang berbeda, Villanyi dkk (2009) mengelompokkan algoritma *schema matching* menjadi tiga, yaitu *linguistic matcher* NTA (*name, connected terms, attributes*) *linguistic matcher, prefix/suffix based matcher for name, dan prefix/suffix based matcher for types*), *vocabular matchers* (*WordNet based word matcher for names* dan NTA (*name, connected terms, attributes*) *related terms similarity*), dan *structural matchers* (*flooding similarity, WordNet based ancestor context similarity, string comparison based child context similarity, child context similarity, dan direct ancestor similarity using string comparison*). Klasifikasi model *schema matching* yang mirip dengan sebelumnya diberikan oleh Bernstein dkk (2011), meliputi *linguistic matching, auxiliary information, instance based matching, structure based matching, constraint based matching, rule based matching, dan hybrid matching*.

Menurut Bernstein dkk (2011), beberapa metode lain pernah dikembangkan termasuk yang memanfaatkan informasi tambahan, yaitu *graph matching, usage based matching, document content similarity, dan document link similarity*. Klasifikasi kombinasi model *schema matching* untuk *schema* berukuran besar yang diberikan oleh Bernstein yaitu strategi independen atau eksekusi berurutan atau eksekusi kombinasi, *parallel*

matching, self-tuning match work flow, early search space pruning, partition based matching, dan optimasi schema. Model schema matching dalam sebuah domain tertentu, yaitu reuse based matching dan pencocokan holistik merupakan kelompok pendekatan berbeda yang telah diusulkan. Beberapa strategi yang menyertakan interaksi dan umpan balik dari pengguna dalam proses pencocokan dikelompokkan oleh, meliputi dukungan GUI (graphical user interface), pencocokan incremental, pencocokan Top-k, dan kolaboratif. Kelompok strategi lain adalah penggunaan algoritma untuk memperluas pencocokan semantic, misal semantic tagging dan conditional tagging. Dalam referensi lain, Özsü dkk (2011) mengelompokkan model schema matching menjadi dua, yaitu schema based dan instance based. Model schema based terdiri atas model element based dan structure based. Model element based terdiri atas linguistic based dan constraint based, sementara level struktur dikembangkan berdasarkan constraint based. Model instance based dikembangkan berdasarkan element level yang terdiri atas linguistic based, constraint based, dan learning based.

Problem utama dalam schema matching adalah sering kali ditemukan penamaan yang tidak jelas dalam schema, kesulitan ditemukan sinonim penamaan, dan perbedaan bahasa sehingga model schema matching tidak mungkin menghasilkan keluaran yang 100% tepat dalam pemetaan schema. Dengan alasan tersebut, maka schema matching tidak bisa sepenuhnya dilakukan secara otomatis, biasanya harus dikoreksi oleh pengguna/pakar untuk memperoleh hasil akhir yang benar. Terdapat dua kasus di mana schema matching akan mengalami kegagalan sehingga memerlukan keterlibatan pengguna, pertama apabila elemen schema sumber tidak dapat dicocokkan dengan satupun elemen schema target menggunakan aturan yang digunakan, atau kedua, apabila elemen schema sumber menghasilkan beberapa elemen yang dianggap cocok dalam schema target dan sistem tidak dapat menentukan elemen yang cocok terbaik secara otomatis. Umumnya pencocokan dua schema membutuhkan informasi yang tidak selalu ditampilkan di dalam schema dan tidak dapat dilakukan secara otomatis, sehingga memerlukan pengguna yang cakap untuk me-review dan menentukan saran pada hasil schema matching. Proses schema matching tidak akan pernah bisa dilakukan secara otomatis penuh selama belum ada model pencocokan semantik yang lengkap pada integrasi sistem informasi. Alasan lain yang menyebabkan schema matching tidak bisa dilakukan secara otomatis karena adanya masalah konflik penamaan dan konflik level abstraksi.

Model schema matching dapat dikembangkan secara individual matcher atau combinational matcher. Model individual matcher memiliki kelebihan dan kelemahan, dan hanya sesuai pada kasus tertentu, sehingga umumnya

memerlukan lebih dari sebuah *matcher* yang dikombinasikan. Model *combinational matchers* dapat diimplementasikan secara terpisah (*composite*) atau bersamaan (*hybrid*). Istilah *composite matchers* sinonim dengan *inter-matcher parallelism*, sedangkan *hybrid matcher* sinonim dengan *intra-matcher parallelism*. Model *hybrid matcher* menggunakan beberapa kriteria pencocokan secara bersamaan, sedangkan *composite matchers* menjalankan algoritma secara terpisah atau independen dan mengkombinasikan pada hasilnya. *Hybrid matcher* menggabungkan beberapa metode secara bersamaan untuk melakukan pencocokan antar elemen *schema*, dan harus memberikan hasil yang lebih baik dan perbaikan kinerja (efektivitas) daripada individual *matcher*. Secara sederhana, *hybrid matcher* menggabungkan dua metode berbeda yang diproses secara *simultaneous*, sedangkan *composite matcher* menggabungkan dua metode yang diproses secara berurutan yaitu sebuah metode akan dilaksanakan setelah metode yang lain selesai dilaksanakan.



BAB
6

KUALITAS INFORMASI

Menurut Sedarmayanti (2001), bahwa “Kualitas adalah suatu ukuran yang menyatakan seberapa jauh telah dipenuhi berbagai persyaratan, spesifikasi, dan harapan”. Sedangkan Informasi menurut Cangara (2005) adalah “kegiatan untuk mengumpulkan, menyimpan data, fakta dan pesan, opini dan komentar, sehingga orang bisa mengetahui keadaan yang terjadi diluar dirinya apakah itu dalam lingkungan daerah, nasional atau Internasional”. Dari kedua definisi diatas peneliti menarik kesimpulan bahwa kualitas informasi adalah suatu ukuran yang menyatakan seberapa jauh telah dipenuhi berbagai persyaratan, spesifikasi, dan harapan dari sebuah kegiatan untuk mengumpulkan, menyimpan data, fakta dan pesan, opini dan komentar, sehingga orang bisa mengetahui keadaan yang terjadi diluar dirinya apakah itu dalam lingkungan daerah, nasional atau Internasional. (Nursyihab, 2020)

Faktor-Faktor Kualitas Informasi Menurut Wing Wahyu Winarno (2004), baik buruknya kualitas informasi dipengaruhi oleh tiga penentu, yaitu:

- 1) Isi informasi, berbagai hal yang dilaporkan.
- 2) Waktu penyajian, informasi yang terlambat disajikan tidak ada gunanya lagi, dan
- 3) Bentuk informasi, tipe jenis informasi memiliki karakteristik yang berbeda, sehingga memerlukan bentuk yang berbeda pula.

a) Isi Informasi

Menurut Abdul Kadir (2002) yang menentukan isi informasi, yaitu:

- Akurasi → menyatakan derajat kebenaran terhadap informasi dan menentukan kehandalan atau reliabilitas informasi. Informasi yang benar-benar bebas kesalahan dikatakan sangat akurat. Penyebab

ketidakakurasian adalah bias dan kesalahan acak. Bias diakibatkan oleh kesalahan ketika dapat diukur, dihimpun, diproses atau disajikan. Kesalahan entri data atau kealpaan dalam mencatat transaksi merupakan penyebab ketidakakurasian informasi yang dihasilkan. Akurasi dapat ditingkatkan melalui ketelitian yang lebih tinggi dalam mengumpulkan dan memproses data. Proses umpan balik dengan cara mengirimkan informasi ke pemakai dan meminta tanggapan terhadap informasi tersebut merupakan contoh lain untuk meningkatkan akurasi.

- Relevan → berarti bahwa informasi benar-benar memberikan manfaat bagi pemakainya. Relevansi informasi untuk setiap pemakai berbeda-beda.
- Idealnya, informasi yang penting bagi pengambilan keputusan haruslah lengkap (tidak ada yang hilang) sehingga dapat mengurangi faktor ketidakpastian.
- Kadang kala informasi yang terlalu detail tidak memberikan hasil yang lebih baik, tetapi malah sebaliknya, karena informasi semakin sulit untuk diresap dan dipahami. Selain itu, penerima informasi mungkin tidak punya waktu banyak untuk melakukan interpretasi. Istilah *over load*/informasi yang berlebihan merupakan istilah yang ditujukan untuk menyatakan adanya informasi yang terlalu melimpah-ruah dan membingungkan. Pada keadaan seperti ini, ada kemungkinan penerima informasi malah mengabaikan informasi yang formal dan berganti menggunakan petunjuk-petunjuk tidak formal (misalnya gosip atau prasangka). Dalam pengambilan keputusan akibatnya sangat berisiko. Oleh karena itu, sering kali informasi yang rinci perlu dikemas dalam bentuk yang lebih ringkas berupa ringkasan/ikhtisar. Hal yang terpenting, informasi harus diringkas agar sesuai dengan kebutuhan penerima informasi sebagai berikut, pihak manajemen tingkat tinggi memerlukan data yang lebih ringkas daripada pihak manajemen tingkat menengah.

b) Waktu penyajian informasi

Yang menentukan baik buruknya sebuah informasi dalam waktu penyajiannya. Menurut Abdul Kadir (2002) bahwa ketepatan waktu (*time liness*) menyatakan usia data yang sesuai dengan upaya pengambilan keputusan. Artinya, informasi tersebut tidak usang/kedaluwarsa ketika sampai ke penerima, sehingga masih ada waktu untuk menggunakan informasi tersebut sebagai bahan pengambilan keputusan. Sedangkan menurut Wing Wahyu Winarno (2004) menyatakan bahwa keterkinian

(*currency*) informasi, yaitu informasi harus benar-benar mencerminkan keadaan paling baru.

c) Bentuk Informasi

Menurut Wing Wahyu Winarno (2004) yang menentukan baik buruknya bentuk informasi ditentukan oleh:

- Kejelasan (*clarity*) dari informasi. Informasi harus disajikan secara jelas, dengan cara meminta tanggapan atau umpan balik terhadap informasi tersebut.
- Rincian (*detail*) laporan harus dapat ditampilkan dan dapat disediakan bila diperlukan oleh pembacanya. Biasanya, manajemen hanya memerlukan informasi secara garis besar saja. Misalnya total penjualan. Cara penyajian bisa dilakukan dengan grafik, dengan warna, dengan kata-kata, dengan perbandingan, dengan garis runtutan, dan lain sebagainya. Informasi yang berbeda memerlukan cara penyajian yang berbeda pula.
- Sarana (media) → pelaporan informasi dapat dilaporkan melalui berbagai media, misalnya e-mail, laporan tercetak/buku, tampilan dilayar, laporan melalui pesan singkat di telepon seluler, informasi berbentuk film di *disk*, atau tampilan di internet informasi yang sederhana dapat disampaikan melalui media yang sederhana, media yang rumit harus disampaikan melalui media tercetak agar bisa dibaca berulang kali dengan mudah.

A. KUALITAS INFORMASI

Kualitas dari suatu informasi tergantung dari tiga hal, yaitu informasi harus akurat, tepat pada waktunya dan relevan.

- 1) Akurat, berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bisa menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke penerima informasi kemungkinan banyak terjadi gangguan (*noise*) yang dapat merubah atau merusak informasi tersebut.
- 2) Tepat pada waktunya, berarti informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi. Karena informasi merupakan landasan di dalam pengambilan keputusan. Bila pengambilan keputusan terlambat, maka dapat berakibat fatal untuk organisasi. Dewasa ini mahalnya nilai informasi disebabkan harus cepatnya informasi tersebut didapat, sehingga diperlukan teknologi-teknologi mutakhir untuk mendapatkan, mengolah dan mengirimkannya.

3) Relevan, berarti informasi tersebut mempunyai manfaat untuk pemakaiannya. Relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda. Misalnya informasi mengenai sebab musabab kerusakan mesin produksi kepada akuntan perusahaan adalah kurang relevan dan akan lebih relevan bila ditujukan kepada ahli teknik perusahaan. Sebaliknya informasi mengenai harga pokok produksi untuk ahli teknik merupakan informasi yang kurang relevan, tetapi relevan untuk akuntan.

B. NILAI INFORMASI

Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya. Kegunaan informasi adalah untuk mengurangi hal ketidakpastian di dalam proses pengambilan keputusan tentang sesuatu keadaan. Masalahnya adalah berapa harus dibayar oleh perusahaan untuk mendapatkan informasi tersebut. Apakah informasi yang didapat sepadan dengan biaya yang dikeluarkan untuk mendapatkannya? Misalnya suatu perusahaan minyak membeli hak pengeboran sebesar 10 juta dollar US dan yakin bahwa investasi tersebut akan sangat bernilai jika pemilikan tersebut mengandung paling sedikit 5 juta barrel minyak mentah. Sedang perusahaan belum mengetahui seberapa banyak minyak mentah yang dikandung di dalam pemilikan tersebut. Ketidakyakinan ini dapat dikurangi dengan mendapatkan informasi tambahan yang berkualitas. Misalnya dengan mengadakan alat pengukur *seismic shot* atau meminta pendapat dari ahli geologi minyak. Untuk maksud mendapatkan informasi tersebut sepadan atau lebih besar atau lebih efektif dan biaya-biaya yang dikeluarkan untuk mendapatkan informasi tersebut, maka dikatakan informasi tersebut bernilai.

Akan tetapi perlu diperhatikan bahwa informasi yang digunakan di dalam suatu *system* informasi umumnya digunakan untuk beberapa kegunaan. Sehingga tidak memungkinkan dan sulit untuk menghubungkan suatu bagian informasi pada suatu masalah yang tertentu dengan biaya untuk memperolehnya. Karena sebagian besar informasi tidak dapat persis ditaksir keuntungannya dengan suatu nilai usang, tetapi dapat ditaksir nilai efektivitasnya. Pengukuran nilai investasi biasanya dihubungkan dengan analisis *cost effectiveness*, atau analisis *cost-benefit*.



DATA-LEVEL ENTERPRISE INTEGRATION

Dalam lingkungan bisnis modern, perusahaan yang bekerja sama perlu mengintegrasikan berbagai aplikasi perusahaan ke dalam satu sistem informasi yang stabil dan terintegrasi. Dalam hubungan bisnis-ke-bisnis (B2B) setidaknya ada dua sistem independen yang perlu bekerja sama dan berkomunikasi untuk mencapai tujuan bisnis. Tantangan utamanya adalah bagaimana mencapai pertukaran data yang dapat dioperasikan antara dua sistem tersebut.

Interaksi antara dua sistem dapat terjadi setidaknya pada dua tingkat: tingkat data dan tingkat proses bisnis. Untuk integrasi tingkat data, sistem harus dapat membaca data satu sama lain atau mereka harus dapat menerima data dalam beberapa format yang diketahui, menafsirkan data, dan memanggil layanan bisnis yang sesuai. Pendekatan yang paling populer untuk pertukaran data adalah untuk bertukar struktur data tekstual menggunakan *Extensible Markup Language* (XML). Meskipun sintaks independen pertukaran data tersedia, masalah mengenali dan memvalidasi semantik data masih perlu diselesaikan. Ontologi domain memberikan deskripsi semantik yang digunakan untuk definisi formal konsep dalam domain tertentu. *Web Ontology Language* (OWL) adalah bahasa yang memungkinkan representasi semantik data secara formal serta penalaran otomatis yang mengikuti dari semantik data. Tingkat interaksi proses bisnis antara sistem dapat dicapai dengan menggunakan arsitektur berorientasi layanan (SOA) dan teknologi Layanan Web yang menggunakan beberapa teknologi terkait (yaitu, SOAP, XSDL, UDDI).

Pada kedua tingkat interaksi, yang digunakan adalah teknologi *platform*-independen dan bebas karena solusi yang dapat dioperasikan memerlukan kemandirian dan portabilitas teknologi. Integrasi tingkat data adalah cara integrasi sistem yang paling kuat, tetapi dapat menjadi solusi yang bergantung pada *platform*. Persyaratan penting untuk solusi integrasi adalah independensi *platform*, cara interaksi yang mudah antar sistem, dan peran mediasi antara aplikasi *vendor* yang berbeda yang direalisasikan menggunakan teknologi yang berbeda dan yang bekerja pada *platform* yang berbeda.

Dalam EAI, integrasi data berarti berbagi data yang relevan antara dua atau lebih aplikasi. Integrasi data juga baik untuk standarisasi data. Tujuan utama dalam setiap tugas integrasi data meliputi peningkatan akurasi data, peningkatan produktivitas, kelincahan dan fleksibilitas yang lebih besar, dan memfasilitasi penggantian sistem/merger organisasi. Integrasi data memerlukan tugas konversi data dan pemetaan yang melibatkan skema sumber, skema target, dan pemetaan di antara mereka. Diperlukan pemahaman dan pemeliharaan skema yang mendasarinya untuk mengatasi perubahan secara teratur. Proses integrasi data terutama terdiri dari operasi seperti ekstraksi, transportasi, transformasi, dan penyisipan. Proses ekstraksi adalah menyiapkan data untuk memenuhi persyaratan aplikasi target dalam format yang dapat diterima untuk transportasi. Selama transportasi data, beberapa faktor penting (misalnya, keandalan dan keamanan) perlu mendapat perhatian khusus. Pada langkah transformasi, data dari aplikasi sumber diformat untuk memenuhi persyaratan aplikasi target. Pada langkah penyisipan, data masuk diimpor ke sistem target. Berbagai macam produk integrasi yang menyediakan infrastruktur untuk melakukan tugas-tugas yang disebutkan sebelumnya tersedia. *Point-to-point*, *spoke-and-hub*, *messaging bus*, dan integrasi layanan perusahaan dengan BPM adalah beberapa pendekatan integrasi umum yang telah digunakan. Dalam "pendekatan *point-to-point*," program integrasi digunakan untuk menghubungkan setiap sumber dan setiap aplikasi target. Dengan kata lain, objek antarmuka yang sesuai dikembangkan antara dua titik pertukaran data yang diperlukan. Operasi utama adalah mengekstrak data target yang sesuai dari sistem sumber data melalui objek antarmuka. "Pendekatan *spoke-and-hub*" memanfaatkan komponen yang dapat digunakan kembali untuk melakukan fungsi integrasi umum; karena pendekatan ini dapat dikonfigurasi, diperluas, dan dapat digunakan kembali, ini memungkinkan integrasi cepat dengan berbagai sistem. Penelitian tentang integrasi data terutama berkaitan dengan pemindahan data antara sumber data yang heterogen di bawah sistem operasi yang berbeda.

Sebagaimana halnya sistem itu sendiri, di dalam sebuah organisasi yang memanfaatkan sistem berbasis teknologi informasi dengan ketersediaan *database* di dalamnya, maka kemungkinan untuk adanya proses integrasi pada level *database* bisa saja terjadi. Apalagi di dalam sebuah organisasi dengan kebutuhan sistem yang makin kompleks, proses integrasi dapat terjadi pada satu atau ketiga level sekaligus (*database*, aplikasi, *middleware*). Integrasi pada level pertama adalah integrasi di level *database*.

Integrasi di level *database* dapat diartikan sebagai sebuah proses penggabungan atau penyusunan bersama (*integration*) dua buah *database* atau lebih, yang telah ada di sebuah organisasi. Terdapat empat buah alasan mengapa sebuah organisasi (misalkan perusahaan) melakukan integrasi di level *database*. Yaitu:

- Kebutuhan akan proses penggabungan (integrasi) dari dua buah *database* atau lebih yang dimiliki, untuk dapat menyederhanakan proses dan layanan. Adanya beragam *database* yang dimiliki oleh sebuah organisasi, yang berasal dari beragam layanan yang disediakan oleh sistem berbasis teknologi informasi (misalkan memiliki sejumlah sistem informasi dengan *datasenya* masing-masing), maka akan menimbulkan masalah di dalam proses pengelolaan, manajemen, penanganan masalah teknis sistem (*Troubleshooting*), pemeliharaan sistem, hingga kenyamanan pengguna (untuk *login* ke setiap sistem). Data dan informasi yang berasal dari beragam *database* ini, dapat disatukan melalui sebuah *database* virtual hasil integrasi, baik dengan menggunakan *Single Share Database* (dibahas pada bagian bawah).
- Keinginan untuk meningkatkan produktivitas dan kualitas layanan yang diberikan oleh organisasi kepada konsumen. Integrasi di level *database* diharapkan mampu meningkatkan produktivitas dan kualitas layanan yang diberikan kepada pengguna (konsumen, nasabah), dengan kesederhanaan dan kesatuan *database* yang dilakukan. Dari sisi penyedia layanan dan pemilik sistem (misalkan Administrator), mereka dapat lebih terbantu di dalam kemudahan penanganan data dalam jumlah banyak dan berasal dari sejumlah *database* berbeda (dalam hal ini berbeda sistem, berbeda platform, hingga berbeda DBMS). Misalkan kemudahan untuk berbagi data dan informasi, pertukaran data dan informasi, serta pemakaian bersama data dan informasi dari beragam *database* yang berasal dari unit-unit yang ada di dalam organisasi tersebut.
- Dari poin 1 dan 2 di atas, diharapkan akan berujung kepada peningkatan pemasukan (pendapatan) dari organisasi bersangkutan karena adanya peningkatan kualitas layanan dan peningkatan produktivitas. Misalkan

sebuah bank yang mampu mewujudkan integrasi *database* dengan baik, akan mampu meningkatkan kualitas layanan kepada nasabah, yang berujung kepada peningkatan kepercayaan nasabah dan pendapatan bank tersebut.

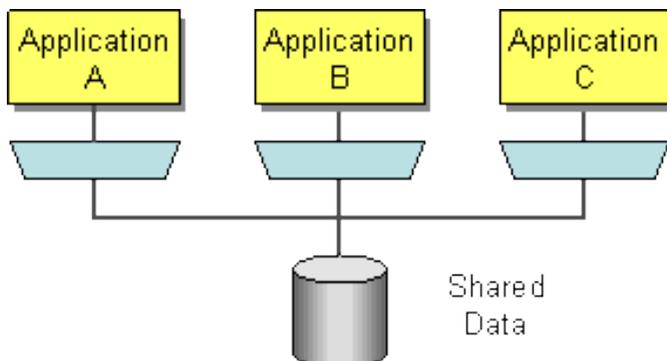
- Kemudahan untuk pengembangan aplikasi dan layanan, dilihat dari sisi pengembang (*Developer*), baik pengembang dari organisasi bersangkutan (misalkan *staff IT*) ataupun dari pihak ketiga. Kemudahan dengan pemanfaatan *Sandbox* dan *Version Control System (VCS)* akan dijelaskan pada pembahasan di bawah.

A. CONTOH INTEGRASI PADA LEVEL *DATABASE*

Contoh paling sederhana dari penerapan integrasi pada level *database* di dalam sebuah organisasi adalah sebagai berikut. Asumsikan terdapat sebuah organisasi berbentuk perusahaan (sebut saja Perusahaan X) yang memiliki sejumlah sistem berbasis teknologi informasi (dalam bentuk sistem informasi dan aplikasi) yang berbeda-beda, beserta dengan *datasenya* masing-masing. Setiap *database* dapat sama atau pun berbeda dalam hal *Data Base Management System (DBMS)* serta *platform* yang digunakan. Perusahaan X ingin melakukan integrasi di level *database*.

Solusi paling sederhana yang dapat diberikan untuk Perusahaan X, adalah dengan menciptakan sebuah *Single Share Database*. *Single Share Data* bertindak sebagai media untuk menyimpan data dan informasi dari sejumlah *database* yang akan diintegrasikan tersebut. Hal ini akan memungkinkan adanya kemudahan di dalam pertukaran data dan informasi antar sistem beserta dengan *datasenya* masing-masing.

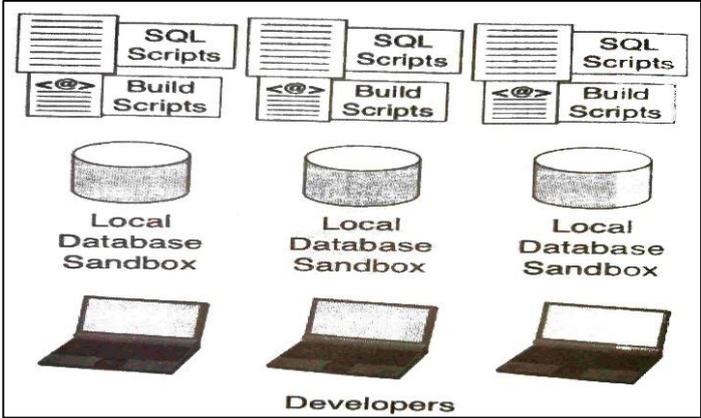
Gambar berikut ini mengilustrasikan penjelasan di atas, mengenai *Single Share Database*, yang dalam contoh ini, melibatkan tiga buah aplikasi berbeda:



Gambar 7.1 Contoh integrasi di level *database* menggunakan *Single Share Database* (sumber: <http://www.enterpriseintegrationpatterns.com/img/SharedDatabaseIntegration.gif>)

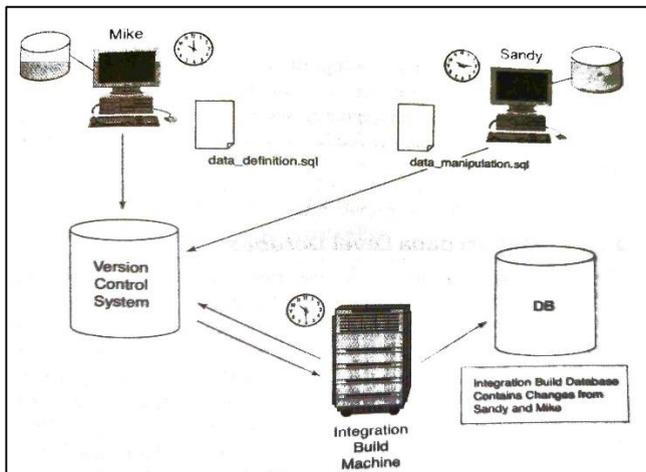
Bagaimana dengan kondisi sebuah organisasi yang memiliki beberapa buah sistem informasi dengan *database* dan pengembang masing-masing? Organisasi dengan kondisi seperti ini, akan memerlukan adanya *Local Database Sandbox*. Setiap pengembang dari sistem berbasis teknologi informasi (berserta *database* di dalamnya) wajib membuat sebuah *Sandbox* untuk *database* dari sistem mereka.

Sandbox ini menjadi semacam *Local Instance* (lingkungan lokal) bagi para pengembang, untuk dapat melakukan uji coba, pembaruan, hingga proses integrasi *database* dan perubahan di dalamnya, tanpa merusak sistem dan struktur *database* itu sendiri. Pada contoh ilustrasi di bawah ini, setiap pengembang dari 3 buah sistem berbeda (dengan *database* masing-masing) akan membuat sendiri *Local Database Sandbox* mereka, untuk kemudahan di dalam proses integrasi pada level *database*.



Gambar 7.2 Contoh integrasi di level *database* dengan melibatkan *Local Database Sandbox* (sumber: http://cdn.ttgtmedia.com/digitalguide/images/Misc/Continteg_CH5_Fig.5-3.jpg)

Contoh lainnya lagi untuk integrasi di level *database* (yang melibatkan pengembang berbeda, sistem, dan *database* yang berbeda) adalah pemanfaatan *Version Control System* (VCS), yang berfungsi sebagai kontrol untuk sinkronisasi perubahan dan pembaruan di dalam *database*, yang dilakukan oleh masing-masing pengembang. Perhatikan gambar ilustrasi berikut ini.



Gambar 7.3 Integrasi di level *database* dengan menggunakan *Version Control System* (Sumber:http://cdnutmedia.com/digitalguide/images/Misc/Continteg_CHS_Fig.5-4.jpg)

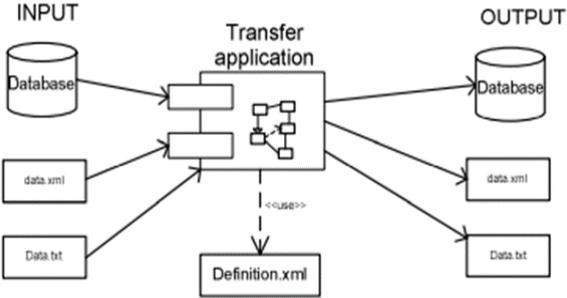
Dari contoh ilustrasi di atas, misalkan terdapat dua buah sistem berbeda dengan *database* dan pengembang masing-masing), sebut saja bernama Mike dan Sandy. Maka apabila terjadi perubahan (manipulasi) terhadap *database* yang dilakukan masing-masing oleh Mike dan Sandy, sistem akan mencatat waktu perubahan dan melakukan sinkronisasi perubahan pada integrasi *database*, berdasarkan kepada repositori dari *Version Control* yang disediakan oleh *Version Control System* (VCS).

Hal yang harus diperhatikan pada proses integrasi di level *database* ini adalah konsistensi data dan informasi di dalamnya secara *real time*. Sebagai contoh, misalkan terdapat tiga buah *database* dari 3 buah sistem berbeda (A, B, dan C) yang akan diintegrasikan. Maka apabila terjadi pembaruan data ataupun proses manipulasi data pada *database* A, maka pada sistem *database* hasil integrasi tersebut, haruslah terdapat pembaruan yang sama untuk data yang berasal dari *database* A. Sistem akan melakukan pembaruan (*update*) secara otomatis. Solusi lainnya untuk hal ini adalah dengan penyediaan *Transaction Management System* (TMS).

Solusi lainnya untuk integrasi pada level data dapat menggunakan *Extensible Markup Language* (XML). *There need to be two elements of the solution:*

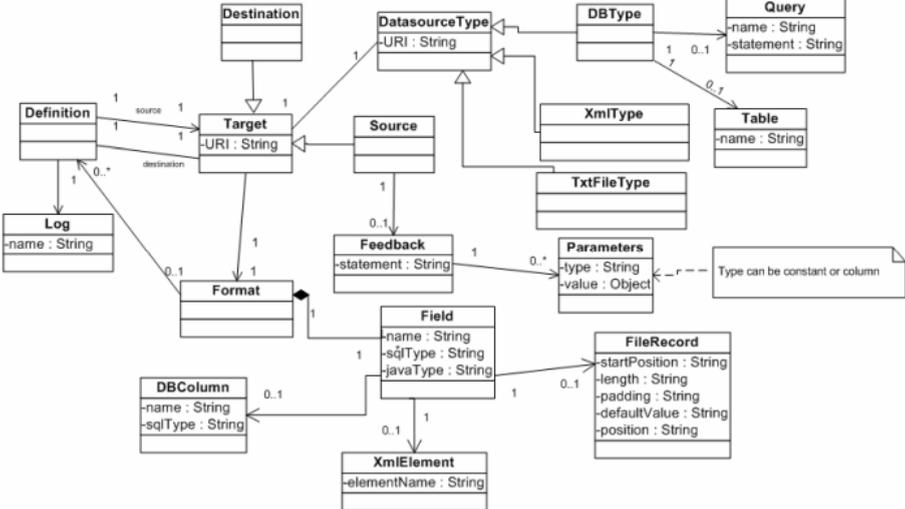
- Komponen yang akan menggunakan definisi ini untuk membaca data dari sumber data (*database*, memori, dll) dan menulis data ke tujuan yang sesuai (*database*, file tekstual, file XML) seperti yang dapat dilihat pada Gambar 7.4.

- Definisi deklaratif tipe sumber data, tipe tujuan data, pemetaan antar *field*, tipe data *field*, dan format seperti terlihat pada Gambar 7.5.



Gambar 7.4 Component Model

Extensible Markup Language (XML) digunakan untuk mencapai deskripsi deklaratif untuk perilaku komponen (aplikasi) selama transfer data dari satu sumber data ke sumber data lainnya (yaitu dari satu *database* ke *database* lainnya, atau dari *database* ke format file yang memadai). Dalam teks selanjutnya, kita akan menggunakan istilah "file definisi" untuk deskripsi deklaratif ini. Pertama-tama, file definisi perlu dijelaskan secara ketat dan formal karena aplikasi transfer yang menggunakan file definisi mengharapkan format yang ditentukan secara tepat. Meta-model untuk file definisi ditunjukkan pada gambar di bawah ini: yaitu model objek yang dibuat oleh aplikasi transfer setelah file definisi dimuat. Setelah pembuatan model objek, kelas lainnya memiliki tanggung jawab untuk menyelesaikan transfer data.



Gambar 7.5 Definition file meta-model

Kelas Definisi adalah titik dasar dan kelas itu berisi dua target: sumber dan tujuan. Kelas sumber mendefinisikan sumber data dari mana sistem akan membaca data. Jika kita menggunakan tabel *database* untuk sumber, maka Sumber memiliki asosiasi dengan kelas *DBType* yang telah mendefinisikan pernyataan SQL. Kelas tujuan juga memiliki tipe sumber data yang mewakili tipe tujuan. Ini bisa berupa tabel di *database* kedua (*DBType*), file tekstual (*TxtFileType*), atau file XML (*XmlType*). Dalam hal ini, *DBType* berisi kelas Tabel yang dapat memberikan informasi tentang tabel tertentu tempat komponen menyisipkan data. Kelas tujuan memiliki Format yang menjelaskan pemetaan antara kolom yang diambil dari Sumber ke kolom tabel tujuan yang sesuai atau file tekstual atau bahkan elemen. Jika kita ingin membuat hubungan *master-detail* untuk data ekspor dalam file XML, maka kelas Format dapat dikaitkan dengan kelas Definisi baru.

Dengan menggunakan model ini, kita dapat memperoleh file Definition.xml. Pada tabel di bawah ini ada dua contoh file definition.xml.

Setelah file definisi selesai, aplikasi transfer mem-*parsing file* ini dengan menggunakan teknologi SAX dan DOM dan membuat model objek.

<i>Describe In/out data transfer</i>	<i>Describe payment transfer orders export</i>
<pre> definition> <source> URI="jdbc:microsoft:sqlserver://10.21.46.200\ITBR:1433; DatabaseName=InfoTime_Transfer;User=sa;Password=br123"> <dataSourceType>DBType</dataSourceType> <SQLQuery name="IN/OUT time data"> select id,regtime,direction, regloc,typeid,READSTATUS from it_reg where readStatus=0 and typeid='E' order by regtime </SQLQuery> <feedback statement="update it_reg set READSTATUS=? where ID=? and REGTIME=? and DIRECTION=? and REGLOC=? and TYPEID=?"> <parameter type="constant" value="1"> </parameter> <parameter type="column" value="id"> </parameter> -- <parameter type="column" value="typeid"> </parameter> </feedback> </source> <destination> URI="jdbc:oracle:thin:@dhost:1521:dbaid;username;password"> <dataSourceType>DBType</dataSourceType> <table name="GATE"> </table> <format> <field name="id" SQLType="NUMERIC" JavaType="java.math.BigDecimal"> <dbcolumn name="employeeid" sqltype="NUMERIC"> </dbcolumn> </field> <field name="regtime" SQLType="DATE" JavaType="java.sql.Date"> <dbcolumn name="time" sqltype="DATE"> </dbcolumn> </field> <field name="direction" SQLType="VARCHAR" JavaType="java.lang.String"> <dbcolumn name="direction" sqltype="VARCHAR"> </dbcolumn> </field> -- <field name="typeid" SQLType="VARCHAR" JavaType="java.lang.String"> <dbcolumn name="typeid" sqltype="VARCHAR"> </dbcolumn> </field> </format> </destination> <log filename="c:\log\transfer.log"> </log> </definition> </pre>	<pre> <definition> <source> URI="jdbc:oracle:thin:@dhost:1521:dbaid;usernam e; password"> <dataSourceType>DBType</dataSourceType> <SQLQuery name="PAYMENT_TRANSFER_ORDER"> select ppid, date,place,nameN,accountN,callNoN, nameM,accountM,calNom, payCode,purpose,amount from payment_t_order where ppid='A' </SQLQuery> </source> <destination> URI="c:\paymentTransferOrders\YUBank\pto.txt"> <dataSourceType>TxtFileType</dataSourceType> <format> <field name="ppid" SQLType="NUMERIC" JavaType="java.math.BigDecimal"> <filerecord startPosition="1" length="3" defaultValue="X" position="10" padding=""> </filerecord> </field> <field name="date" SQLType="DATE" JavaType="java.sql.Date"> <filerecord startPosition="4" length="6" position="10"> </filerecord> </field> -- <field name="place" SQLType="VARCHAR" JavaType="java.lang.String"> <filerecord startPosition="10" length="12" position="10" padding=""> </filerecord> </field> </format> </destination> <log filename="c:\log\dataexport.log"> </log> </definition> </pre>



APPLICATION-LEVEL ENTERPRISE INTEGRATION

Oleh karena itu, sistem informasi saat ini terdiri dari aplikasi yang dikembangkan dalam periode yang berbeda, oleh tim pengembangan yang berbeda, sebagian besar waktu tanpa koordinasi nyata, dan menggunakan teknologi yang berbeda. Dalam sebagian besar kasus, aplikasi baru dibuat menggunakan teknologi terbaru pada saat itu, sering kali tanpa benar-benar mempertanyakan perlunya penerapan teknologi semacam itu, dan melupakan bahwa teknologi "modern" menjadi usang dalam sekejap.

Selain itu, seperti yang telah kita lihat, aplikasi ini pada awalnya tidak dirancang untuk berkomunikasi satu sama lain. Tautan dan antarmuka yang memungkinkan mereka untuk bekerja sama harus dibuat di antara setiap aplikasi. Pengenalan paket perangkat lunak komersial dalam sistem informasi hanya memperkuat masalah ini karena mereka dirancang untuk menanggapi masalah tertentu tanpa disibukkan dengan komunikasi dengan sistem yang ada.

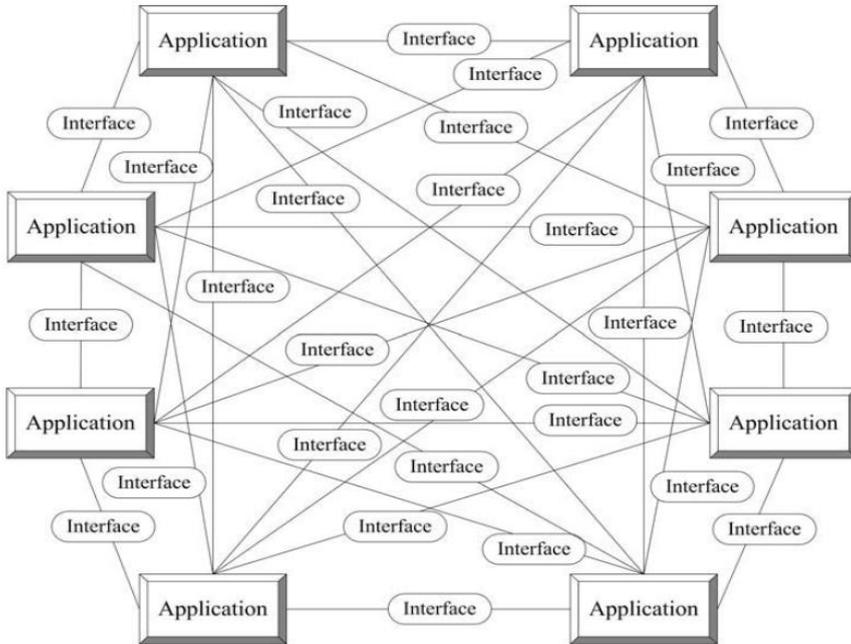
Jika kita mempertimbangkan kasus ekstrim di mana semua aplikasi harus berkomunikasi dengan semua aplikasi lain dalam sistem informasi, maka jumlah antarmuka dua arah yang diperlukan adalah hasil langsung dari penerapan rumus:

$$i = n(n - 1)/2$$

di mana i adalah jumlah antarmuka dan n adalah jumlah aplikasi.

Untuk menghubungkan 6 aplikasi, 15 antarmuka harus dibuat. Untuk menghubungkan 150 akan membutuhkan 11.175!

Dengan cara ini, dan tentu saja, kami akhirnya menciptakan sistem "spageti" (Gambar 8.1), di mana sangat sulit untuk menganalisis potensi dampak modifikasi pada aplikasi atau untuk mengukur ruang lingkup perubahan yang diperlukan untuk mengintegrasikan aplikasi baru. Manajer TI mencapai titik di mana mereka ragu sebelum memodifikasi sistem informasi mereka karena mereka tidak lagi mampu mengendalikan konsekuensi dari perubahan mereka sendiri. Sistem itu sendiri mulai menolak perubahan, dan hasilnya bisa disebut fosilisasi sistem informasi.



Gambar 8.1 A "spaghetti" system

Sistem informasi perusahaan sebagian besar dicirikan oleh:

- heterogenitas dalam platform perangkat keras dan aplikasi;
- perkalian antarmuka antara aplikasi;
- ketidakmampuan yang terkenal untuk bertukar informasi dengan mudah, baik secara internal maupun dengan dunia luar.

Dalam hal kecepatan evolusi dan komunikasi, sistem seperti itu benar-benar tidak dapat diadaptasi dengan baik, tetapi seperti yang telah kita lihat, justru kualitas-kualitas itulah yang sangat diperlukan untuk kelangsungan hidup perusahaan dalam konteks ekonomi saat ini. Memang, saat ini harus dimungkinkan untuk – misalnya – mendelokalikasi sebagian dari sistem

informasi atau menggabungkan *back-office* untuk mengoptimalkan biaya dan memusatkan investasi di area yang menambah nilai dan membantu membedakan perusahaan dari pesaingnya.

Oleh karena itu, sistem informasi harus disesuaikan dengan kebutuhan perusahaan saat ini, khususnya dengan menempatkan infrastruktur untuk mengelola pertukaran. Justru di sini kita memasuki domain solusi integrasi aplikasi. Sebelum memeriksa karakteristik mereka, pertama-tama perlu untuk mengidentifikasi masalah yang harus mereka selesaikan dan kendala yang harus mereka pertanggungjawabkan.

Secara umum, integrasi di level aplikasi (*Application Level Integration*) merupakan proses integrasi yang dilakukan pada dua buah aplikasi (*software*) atau lebih pada suatu organisasi. Integrasi di level aplikasi juga dapat diartikan sebagai proses integrasi sistem pada suatu organisasi, yang menghusus pada perangkat lunak komputer (*Software*) atau aplikasi yang digunakan oleh organisasi bersangkutan.

Untuk mewujudkan integrasi di level aplikasi, organisasi harus memahami terlebih dahulu mengenai konsep-konsep dasar di dalamnya. Salah hal yang perlu diketahui terkait dengan integrasi di level pendekatan (*Approach*) yang digunakan untuk mewujudkan integrasi di level aplikasi.

Sebuah pendekatan (*Approach*) merupakan pedoman di dalam mengimplementasikan sesuatu, baik sebuah sistem, *Software*, *Hardware*, maupun produk riset. Demikian pula dengan pendekatan yang digunakan untuk mengimplementasikan integrasi di level aplikasi. Terkait dengan hal ini, *Mercator*, sebagai sebuah perusahaan teknologi informasi skala dunia, yang bergerak di bidang integrasi sistem *enterprise*, menyatakan di dalam tulisannya yang berjudul *Leveraging Web Services Application Integration*, mengenai empat jenis pendekatan (*Approach*) yang ada di dalam proses integrasi pada level aplikasi.

Keempat jenis pendekatan integrasi di level aplikasi ini, dibedakan berdasarkan kepada orientasi yang digunakan di dalamnya. Adapun keempat jenis pendekatan pada proses integrasi di level aplikasi tersebut, meliputi *Portal Oriented Application Integration*, *Process Integration Oriented Application Integration*, *Application Service Oriented Application Integration*, dan *Information Oriented Application Integration*. Uraian masing-masing pendekatan, akan disajikan di bawah ini.

a. *Portal Oriented Application Integration*

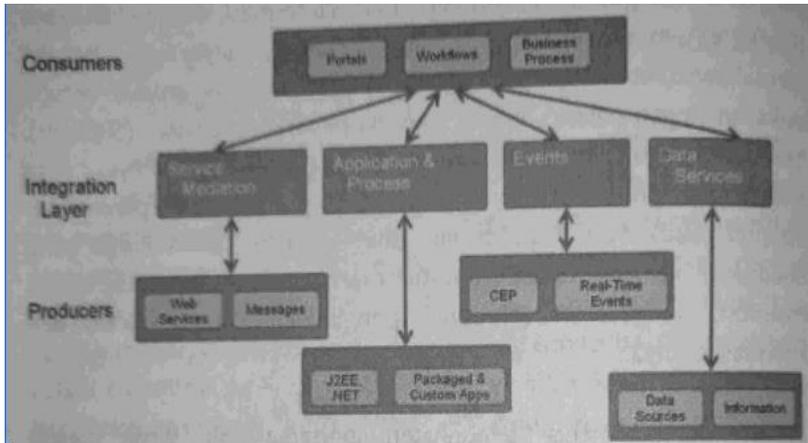
Pendekatan pertama yang dapat digunakan di dalam proses integrasi di level aplikasi pada sistem berbasis teknologi informasi pada sebuah organisasi adalah *Portal Oriented Application Integration*. Menurut David S. Lithicum di

dalam tulisannya di tahun 2000 yang berjudul *Portal Oriented B2B Application Integration*, disebutkan bahwa pada jenis pendekatan ini, kunci utamanya adalah penyediaan sebuah Portal bagi pihak yang terlibat dengan organisasi, terkait dengan jenis hubungan B2B (*Business to Business*). Portal didefinisikan sebagai sebuah aplikasi (umumnya berupa *web based*), yang mampu menghubungkan banyak pengguna sekaligus, untuk berbagi data dan informasi, dengan hak akses (*Privilege*) masing-masing. Sebagai contoh, organisasi berupa perusahaan (sebut saja perusahaan X) yang menjual produk berupa makanan ringan (*snack*), dapat memiliki beberapa penyalur (*Supplier*), baik perorangan maupun organisasi (perusahaan) lainnya, yang menyalurkan produk *snack* tersebut ke konsumen akhir (perorangan maupun toko). Para *Supplier* ini dapat *login* ke dalam Portal yang disediakan oleh perusahaan X, untuk melihat informasi produk, harga, inventori, pengecekan (*Tracking*) pengiriman dan penjualan, dan informasi lainnya.

Terkait dengan jenis pendekatan pertama ini, pihak Oracle (www.oracle.com) sebagai perusahaan teknologi informasi kelas dunia yang bergerak di bidang penyediaan produk dan layanan utamanya berupa *database*, mendefinisikan proses integrasi di level aplikasi untuk Portal *Oriented Application Integration* ini, ke dalam tiga buah *layer* berbeda. Ketiga buah *layer* tersebut (dari atas ke bawah) meliputi *Consumers*, *Integration Layer*, dan *Producers*. *Consumer Layer* meliputi tiga hal yang berhubungan langsung dengan konsumen akhir dan *supplier*. yaitu *Portals*, *Workflows*, dan *Business Process*.

Pada *Integration Layer*, terdapat empat hal yang berhubungan dengan *Workflows* pada *Consumers Layer* di atas. Yaitu *Service Mediation*, *Application And Process*, *Events*, dan *Data Services*. Terakhir, pada *Producers Layer*, terdapat delapan hal yang berhubungan dengan keempat hal pada *Integration Layer*. Antara lain adalah *Web Service* dan *Message* (yang berhubungan dengan *Service Mediation*), *J2EE*, *.NET*, *Package And Custom Apps* (yang berhubungan dengan *Application And Process*), *CEP* dan *Real Time Events* (yang berhubungan dengan *Events*), serta *Data Sources* dan *Information* (yang berhubungan dengan *Data Services*).

Gambar berikut ini menunjukkan bagan dari penjelasan mengenai *Portal Oriented Application Integration*.

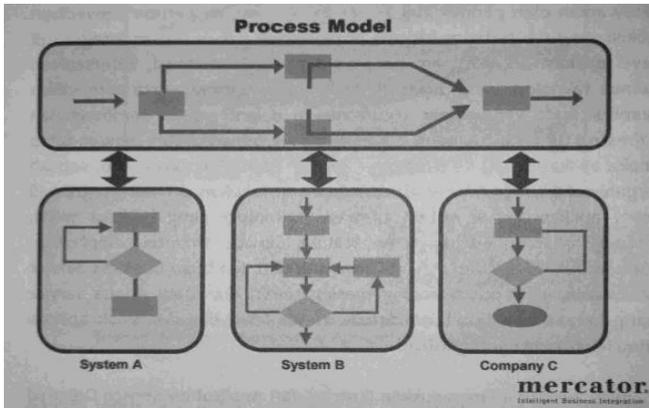


Gambar 8.2 Bagan *Portal Oriented Application Integration*

b. *Process Integration Oriented Application Integration*

Pada jenis pendekatan berupa *Process Integration Oriented Application Integration* ini, ide yang digunakan di dalamnya adalah memanfaatkan integrasi dari sejumlah proses, untuk dapat menjelaskan integrasi di level aplikasi. Setiap aplikasi memiliki sistem, proses, *database*, dan proses bisnis masing-masing. Maka, sebelum melakukan integrasi di level aplikasi, dilakukan terlebih dahulu pemodelan proses dengan cara menggabungkan proses bisnis dari masing-masing aplikasi yang diintegrasikan tersebut.

Sebagai contoh, terdapat tiga buah sistem berbeda (A,B, dan C) dengan aplikasi, *database*, dan proses bisnis masing-masing. Ketiga buah *system* ini kemudian digabungkan (melalui aplikasi dalam bentuk integrasi di level aplikasi), di mana proses bisnis dari masing-masing sistem digabungkan menjadi sebuah proses bisnis gabungan. Dari proses bisnis gabungan inilah, maka patokan untuk membuat *Compsite Application* dapat diwujudkan. Gambar di bawah ini menunjukkan ilustrasi dari penjelasan mengenai *Process Integration Oriented Application Integration* ini:



Gambar 8.3 *Process Integration Oriented Application Integration*

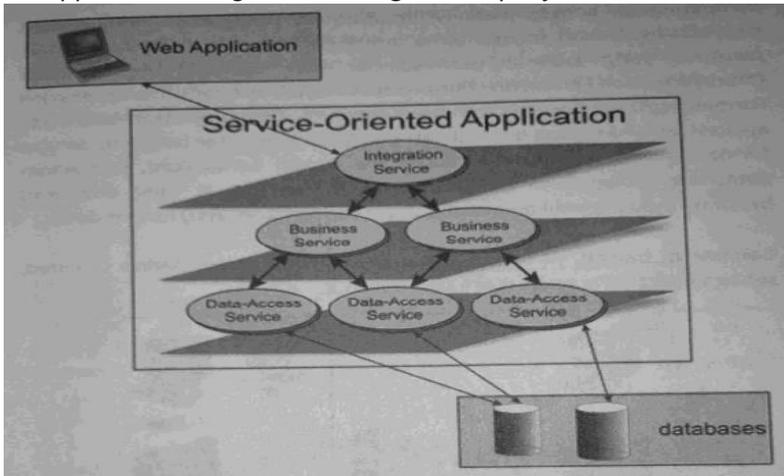
(Sumber: http://www.omg.org/news/meetings/workshops/presentations/Web_Services_2002/062_L_Linthicum-Leveraging_WS_for_Application_Intergration.pdf)

c. *Application Service Oriented Application Integration*

Jenis pendekatan ketiga ini, memuat hal yang lebih kompleks dibandingkan pendekatan pertama dan kedua (yang telah dibahas di atas) maupun pendekatan keempat (yang akan dijelaskan selanjutnya di bagian bawah). Mengapa dikatakan lebih kompleks? Hal ini disebabkan oleh karena *Application Service Oriented Application Integration* menyediakan metode integrasi di level aplikasi, dengan cara menciptakan sebuah aplikasi gabungan berbentuk *Composite Integration*. Di dalam *Composite Integration* tersebut, terdapat proses berbagi (*Sharing*) dalam bentuk sumber kode, *Business Logic*, serta proses (program, objek, transaksi).

Namun, di balik kompleksitas yang dimilikinya, tersimpan nilai lebih yang ditawarkan oleh pendekatan ketiga ini. Antara lain berupa penyediaan solusi yang tak terbatas kepada organisasi di dalam proses integrasi di level aplikasi untuk sistem berbasis teknologi informasi, ketersediaan semua teknologi yang dapat digunakan di dalamnya, serta penyediaan sumber kode yang dapat digunakan di dalam proses pembangunan infrastruktur untuk *Composite Application*. Sebagai contoh, dengan tatap muka aplikasi (GUI) ke pengguna berupa aplikasi berbasis web, sebuah organisasi (misalkan perusahaan) dapat melakukan proses integrasi di level aplikasi untuk sistem berbasis teknologi yang mereka miliki, menggunakan pendekatan *Application Service Oriented Application Integration*. *Integration Service* terbentuk dari dua buah *Business Service* atau lebih, yang mana masing-masing terdiri atas *Data Access Service* yang berasal dari dua buah *database* atau lebih dan dua buah aplikasi atau lebih yang diintegrasikan.

Gambar berikut ini menunjukkan ilustrasi dari *Application Service Oriented Application Integration*, sebagaimana penjelasan di atas:



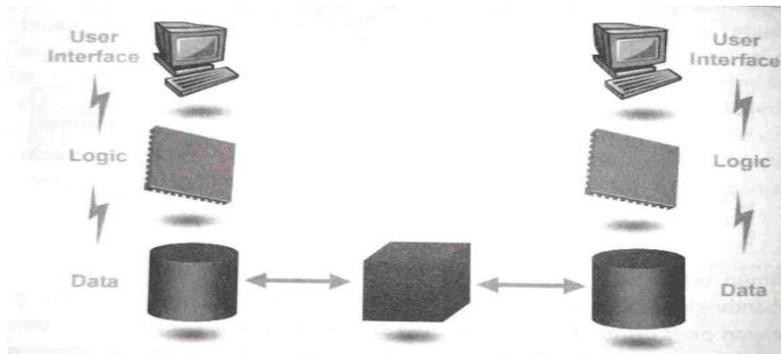
Gambar 8.4 *Application Service Oriented Application Integration*
(Sumber: http://cdn.ttgmedia.com/digitalguide/images/Misc/soa_bd_1.gif)

Namun tentu saja, untuk kompleksitas yang dimilikinya tersebut (jika dibandingkan dengan ketiga pendekatan lainnya), harus diimbangi dengan pengorbanan yang lebih dari sisi waktu, biaya, perencanaan, dan arsitektur sistem. *Application Service Oriented Application Integration* lebih cocok diterapkan pada organisasi skala besar.

d. Information Oriented Application Integration

Pada jenis pendekatan *Information Oriented Application Integration* ini, logik dan *database* yang berbeda dari dua buah aplikasi pada dua buah komputer yang berbeda, dapat diintegrasikan melalui sebuah media yang menggunakan konsep *Business to Business (B2B)*. B2B memudahkan penyediaan sebuah format baku untuk data, yang berasal dari semua platform yang berasal dari sumber dan aplikasi yang berbeda (*Transformation Formatting/Data Format Transformation*). *Transformation Formatting/Data Format Transformation* ini, memungkinkan dua buah aplikasi berbeda, yang berasal dari dua buah komputer berbeda, dengan format data yang berbeda, dapat melakukan komunikasi, pertukaran data, dan integrasi di level aplikasi, dengan format yang disepakati bersama atau melakukan perubahan (*Convert/konversi*) format data.

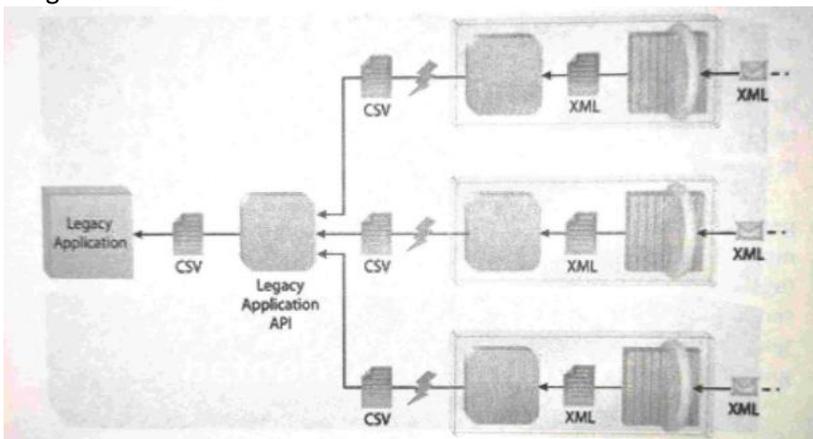
Gambar di bawah ini menunjukkan ilustrasi dari *Information Oriented*, sebagaimana yang telah dijelaskan tersebut:



Gambar 8.5 Ilustrasi *Information Oriented Application Integration*

(Sumber: http://www.omg.org/news/meetings/workshops/presentations/Web_Services_2002/062_L_Linthicum-LeveragingWS_for_Application_Intergration.pdf)

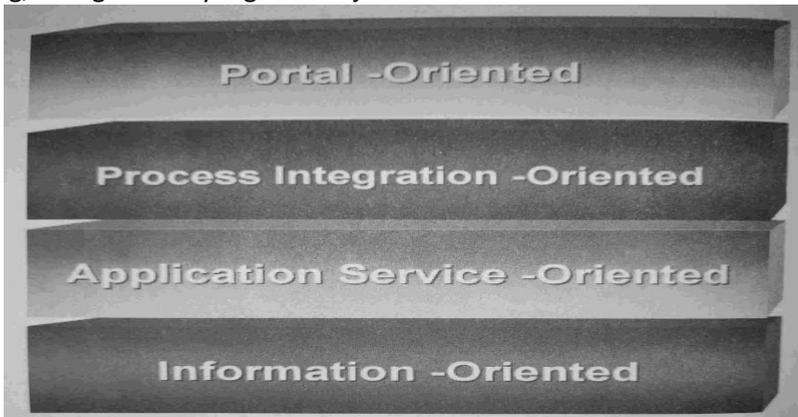
Sebagai contoh untuk *Transformation Formatting* atau *Data Format Transformation*, misalkan dua buah aplikasi berbeda pada komputer yang berbeda (aplikasi X terinstal pada komputer A dan aplikasi Y terinstal pada komputer V) dengan format berbeda (aplikasi X dengan format CSV, aplikasi Y dengan format XML), dapat melakukan integrasi di level aplikasi menggunakan pendekatan *Information Oriented* ini, dengan menyediakan *Legacy Application API (Application Programming Interface)* dan fungsi untuk melakukan konversi dari format CSV ke XML dan sebaliknya. Perhatikan ilustrasi gambar di bawah ini:



Gambar 8.6 Bagan Data Format *Transformation*

(sumber: http://soapatterns.org/static/images/figures/data_format_transformation/fig1.png)

Secara umum, gambar berikut ini menunjukkan rangkuman dari keempat buah jenis integrasi di level aplikasi berdasarkan kepada orientasinya masing-masing, sebagaimana yang telah dijelaskan di atas:



Gambar 8.7 Keempat jenis integrasi di level aplikasi (versi *Mercator*)
(sumber: http://www.omg.org/news/meetings/workshops/presentations/WebServices_2002/2_L_Linthicum-LeveragingWS_for_Application_Intergration.pdf)

A. INTEGRATION TYPOLOGIES

Seperti yang dicatat oleh Gartner Group, apa pun solusi yang diterapkan, semuanya berkisar pada penanganan satu atau lebih dari bidang masalah berikut:

- penyebaran dan konsistensi data;
- pengelolaan proses multi-langkah;
- pembuatan aplikasi komposit.

Untuk mencapai tujuan tersebut, solusi harus memperhatikan karakteristik dan kendala dari aplikasi yang sudah ada dalam sistem informasi, antara lain:

- aplikasi *batch*;
- aplikasi transaksional;
- aplikasi klien/*server*;
- Aplikasi *web*;
- aplikasi waktu nyata;
- paket perangkat lunak komersial.

B. CLASSIFYING THE INTEGRATION PROBLEM TYPES

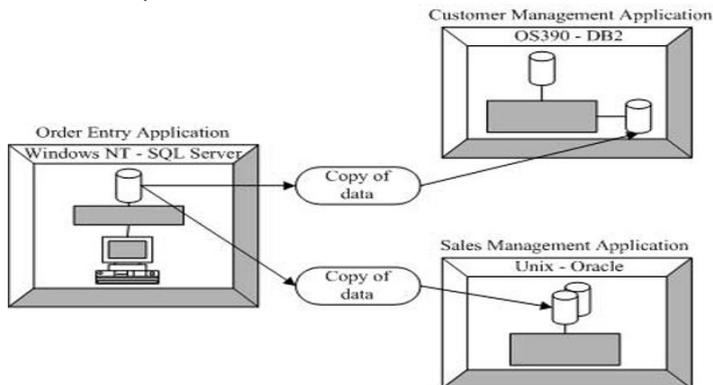
a. Data propagation and data fidelity

Alasan utama untuk menerapkan solusi integrasi aplikasi biasanya menyebarkan data dan memaksakan konsistensi di atasnya. Ini melibatkan memastikan konsistensi data yang disimpan secara berlebihan oleh aplikasi yang berbeda untuk alasan geografis, teknis, atau organisasi. Jenis integrasi ini berdampak pada banyak proses bisnis, terdiri dari beberapa langkah dengan interaksi asinkron dan searah yang berfungsi pada platform independen.

Misalnya, pertimbangkan tiga aplikasi, satu untuk menerima pesanan, satu untuk mengelola pelanggan, dan satu untuk mengelola penjualan. Setiap aplikasi beroperasi pada platform yang berbeda dan menggunakan teknologi yang sesuai dengan platform tersebut, dan masing-masing menggunakan informasi tentang pelanggan perusahaan. Konsistensi informasi di antara ketiga aplikasi ini harus dipastikan.

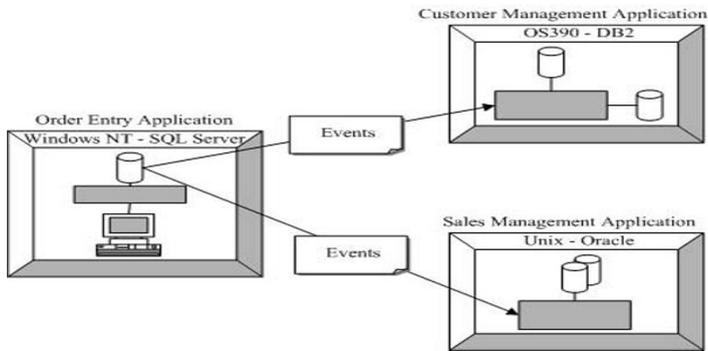
Dua metode dapat digunakan:

- Pendekatan repositori: salin data langsung dari *database* (atau dari file) yang digunakan oleh aplikasi asli (pengambilan pesanan) ke *database* (atau file) yang digunakan oleh dua aplikasi lainnya (Gambar 8.8). Pendekatan ini berjalan melalui data. Ini mengandaikan bahwa model semantik dan data yang digunakan oleh aplikasi kompatibel. Selain itu, tidak menjaga integritas data dari aplikasi penerima karena tidak memungkinkan untuk menerapkan logika dan aturan manajemen yang diperbarui dari aplikasi lain tersebut.



Gambar 8.8 Propagate data and impose consistency: going through the data

- Pendekatan berbasis peristiwa: mengirimkan peristiwa dari aplikasi asli ke dua aplikasi lainnya, yang kemudian akan memastikan pemrosesannya, memperbarui informasi mereka sendiri (Gambar 8.9).



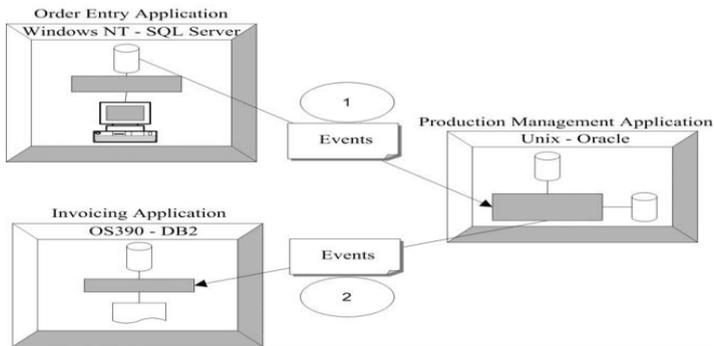
Gambar 8.9 Propagate data and impose consistency: going through the events

Tidak peduli metode mana yang digunakan, data dapat disalin atau peristiwa ditransmisikan dalam sesi (*batch* untuk menyalin informasi atau mengirim peristiwa) atau secara *real-time* (menyalin atau mengirim setiap proses yang dilakukan dalam aplikasi asli).

b. *The management of multi-step processes*

Manajemen proses multi-langkah membahas proses bisnis di perusahaan. Integrasi memungkinkan proses bisnis yang terdiri dari beberapa langkah aplikasi dengan interaksi asinkron dan satu arah, tetapi secara logis saling bergantung, dan dengan pemrosesan yang dilakukan oleh setiap langkah bergantung pada pemrosesan yang dilakukan di hulu oleh langkah lainnya. Aplikasi ini tentu saja dapat berfungsi pada platform independen. Proses semacam itu dapat melibatkan intervensi manusia, tetapi sebagian besar waktu, itu sepenuhnya otomatis, menunjukkan apa yang kita sebut pemrosesan ekspres, atau "pemrosesan langsung" (STP).

- Sebagai contoh, pertimbangkan tiga aplikasi lain: satu untuk menerima pesanan, satu untuk mengelola produksi, dan satu lagi untuk menangani penagihan. Sekali lagi, mereka semua berjalan pada *platform* yang berbeda dan menggunakan teknologi yang sesuai dengan platform tersebut. Tujuannya adalah integrasi proses lengkap dalam pendekatan STP: ambil pesanan – buat artikelnya – buat fakturnya. Integrasi ini akan memerlukan pengurutan otomatis dari ketiga aplikasi dengan juga mengelola transmisi peristiwa manajemen terkait, misalnya, pesanan pembuatan dan permintaan untuk mengeluarkan faktur.



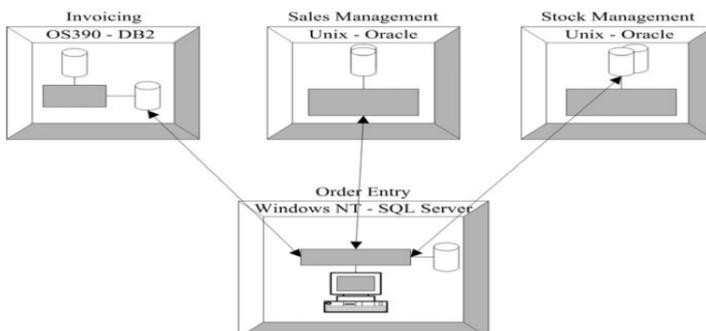
Gambar 8.10 Management of multi-step processes: event-based approach

- Acara dapat ditransmisikan dalam *batch* (menyalin informasi atau mengirim acara) atau secara *real-time* (menyalin atau mengirim saat dan ketika aplikasi asli dijalankan). Transmisi *real-time*, tentu saja, lebih tepat untuk mencapai STP.

c. **Creating composite applications**

Ini adalah pertanyaan membuat aplikasi baru dengan menggunakan data atau layanan yang tersedia di aplikasi yang ada, yang menjelaskan denominasi "aplikasi komposit". Ini adalah jenis integrasi yang paling sering ditemukan pada dasar penerapan aplikasi klien di Web. Aplikasi ini akan, misalnya, meminta satu atau lebih transaksi di komputer pusat, dan akan memanggil paket perangkat lunak di Unix atau NT.

Sebagai contoh, pertimbangkan aplikasi untuk menangani pesanan di Web. Aplikasi dikembangkan dengan memanggil data yang disimpan dalam *database* aplikasi klien pada platform MVS di bawah DB2; itu akan memanggil layanan dalam paket manajemen penjualan (beroperasi di Unix), dan itu akan menginterogasi *database* inventaris (pada platform Unix lain).



Gambar 8.11 Creation of a composite application

Contoh integrasi ini menyangkut proses bisnis tunggal, satu langkah, yang memerlukan interaksi sinkron dan dua arah, melalui pertukaran kesatuan secara *real-time*. Paling sering, itu akan menggunakan pendekatan yang menggabungkan data dan peristiwa.

Untuk menempatkan solusi integrasi aplikasi, maka pertama-tama melibatkan penentuan masalah yang akan diselesaikan (propagasi data, pemrosesan langsung, atau aplikasi komposit), kemudian metode untuk menyelesaikannya (integrasi melalui data, atau melalui peristiwa), tetapi perlu juga diketahui karakteristik dan kendala dari aplikasi yang akan diintegrasikan.



BAB
9

METHOD-LEVEL ENTERPRISE INTEGRATION

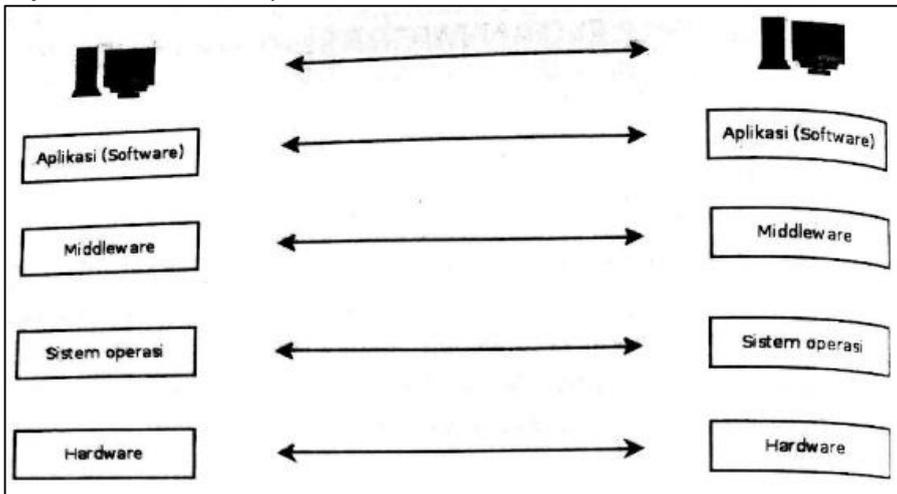
Ini adalah spesialisasi yang lebih jarang digunakan dari metode integrasi tingkat aplikasi yang ditunjukkan di atas. Di sini, kami menggabungkan operasi umum pada beberapa aplikasi ke dalam satu aplikasi yang menjadi depan aplikasi terintegrasi. Ini umumnya digunakan ketika setiap aplikasi terintegrasi memiliki seperangkat API atau metode fungsional yang serupa. Aplikasi terintegrasi harus mendukung *Remote Procedure Call* (RPC) atau teknologi komponen terdistribusi. Kerugian utama dari pendekatan ini adalah lagi-lagi aplikasi kopling yang ketat di komponen depan. Mereka akan rusak ketika perubahan dilakukan pada API aplikasi terintegrasi, dan masalah ini akan menyebar ke aplikasi lain yang bergantung padanya. Ini digunakan ketika kami telah mendistribusikan komponen atau teknologi CORBA.

Bagaimana memilih metode integrasi? Ini benar-benar latihan dalam pemodelan berbasis kendala. Anda harus melihat setiap sistem dan menentukan antarmuka yang mungkin untuk aplikasi itu. Dalam beberapa kasus, aplikasi tidak memiliki API apa pun; oleh karena itu penyimpanan data *backend* mewakili satu-satunya pilihan. Dalam kasus lain, API dan infrastruktur CORBA mungkin ada; jadi gunakan integrasi tingkat aplikasi.

Integrasi pada level ini memerlukan pemahaman bukan saja dari sisi perangkat lunak komputer (*Software*), tapi juga juga dari sisi perangkat keras komputer (*Hardware*) sehingga biasa juga disebut dengan integrasi pada level *middleware*. Secara umum dapat dikatakan bahwa *Middlewared* merupakan jenis dari perangkat lunak komputer (*software*) yang berfungsi sebagai penghubung (*Connector*) dan memfasilitasi proses integrasi (*Integrator*) aplikasi-aplikasi yang berjalan pada *layer* atas (*Application Layer, Translation Layer*) dengan perangkat keras komputer (*Hardware*) yang berjalan di *layer*

bawah, di mana penghubung keduanya adalah sistem operasi. *Middleware* memungkinkan aplikasi-aplikasi dapat saling berkomunikasi satu sama lain, meskipun berbeda platform maupun berbeda *vendor*.

Gambar berikut ini menunjukkan ilustrasi dari *Middleware* pada dua buah komputer yang sedang berkomunikasi, yang berada di antara aplikasi (*Software*) dengan perangkat keras komputer (*Hardware*), yang mana di antara keduanya tersebut, juga dibatasi oleh sistem operasi, sebagaimana penjelasan dari definisi pertama di atas:



Gambar 9.1 Ilustrasi dari *Middleware* dan lokasinya di antara *Software*, sistem operasi, dan *Hardware*

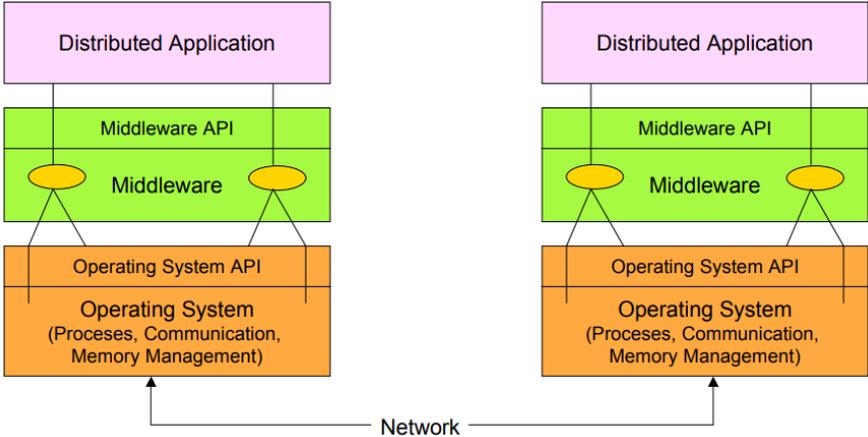
Definisi kedua menyatakan bahwa *Middleware* merupakan perangkat lunak komputer penghubung (*Connectivity Software*) yang memperbolehkan beberapa buah proses untuk berjalan pada satu atau beberapa buah komputer sekaligus pada jaringan komputer. Definisi ini menekankan fungsi dari *Middleware* untuk membantu proses migrasi sistem dari aplikasi yang bersifat *Mainframe* ke aplikasi yang bersifat *Client Server*, dengan menyediakan komunikasi antar aplikasi, meskipun aplikasi-aplikasi tersebut berbeda *platform* dan *vendor*.

Masih mengacu pada definisi pertama dan kedua di atas, jika dihubungkan dengan posisi dari *Middleware* yang berada di antara aplikasi (*Software*) dan sistem operasi (serta di bawah sistem operasi merupakan *layer* untuk *Hardware*), terdapat *Application Programming Interface (API)*, yang menjembatani tiga hal pada sistem operasi dengan aplikasi-aplikasi yang umumnya bersifat terdistribusi pada jaringan komputer (disebut dengan *Distributed Application*). Keempat proses pada sistem operasi tersebut,

meliputi proses (*Process*), komunikasi (*Communication*), dan manajemen memori (*Memory Management*). API pada *Middleware* berkomunikasi dengan API pada sistem operasi.

Dari hal ini, sebenarnya terlihat bahwa *Middleware* memiliki peran tambahan untuk membantu proses pengembangan perangkat lunak komputer (*Software*) menjadi lebih sederhana dan cepat diselesaikan. Hal ini disebabkan oleh karena *Middleware* memiliki kemampuan untuk menyediakan abstraksi pemrograman (*Programming Abstraction*) bagi para pengembang *Software*, detail dari pemrograman yang memanfaatkan bahasa tingkat rendah (*Low Level Programming*) yang membutuhkan komunikasi langsung dengan *Hardware*, serta dukungan terhadap sebanyak mungkin sistem operasi, *Hardware*, dan *platform* pengembangan. Aplikasi-aplikasi yang dijalankan secara paralel dan terdistribusi pada jaringan komputer, haruslah bebas, dalam artian tidak terikat pada satu *vendor*, *platform*, atau sistem operasi tertentu.

Gambar di bawah ini menunjukkan ilustrasi dari penjelasan di atas, mengenai API pada *Middleware* dengan API pada sistem operasi, serta hubungannya dengan *Distributed Application* dan jaringan komputer (*Network*):



Gambar 9.2 API pada *Middleware* dan API pada sistem operasi

(Sumber: https://www.dcl.hpi.uni-potsdam.de/LV/Components04/VL1/02_middleware.pdf)

Jika merunut kembali ke belakang, sejarah awal mengenai teknologi *Middleware* adalah pada tahun 1990, ketika *Middleware* mulai diperkenalkan ke publik, untuk mendukung interoperabilitas antar aplikasi pada arsitektur jaringan *Client Server*. Beberapa buah *Middleware* yang terkenal hingga saat ini, antara lain adalah *Common Object Request Broker Architecture* (CORBA),

Component Object Model/Distributed Component Object Model (COM/ DCOM), dan *Distributed Computing Environment (DCE)*. *Middleware-Middleware* ini, diprakarsai maupun disponsori oleh perusahaan (misalkan oleh Microsoft) maupun memiliki bentuk organisasi berupa *Open Software Foundation (OSF)* dan *Object Management Group (OMG)*.

Namun, jika meninjau lebih lanjut mengenai kategori standarisasi untuk *Middleware*, maka dapat disebutkan adanya tujuh buah kategori standarisasi pada *Middleware*, yang umum digunakan di dunia teknologi informasi. Ketujuh buah kategori standarisasi pada *Middleware*, meliputi *Object Request Broker (ORB)*, *Distributed Computing Environment (DCE)*, *Database Access System (DAC)*, *Transaction Processing Monitoring (TPM)* *Remote Procedure Call (RPC)*, *Message Passing (MP)*, dan *Enterpris Service Bus (ESB)*. Hampir semua dari standarisasi pada *Middleware* ini diterapkan pada bentuk aplikasi dan layanan *Web Service* dan implementasi *Service Oriented Architecture (SOA)* lainnya.

Di dalam integrasi pada level *Middleware* yang dilakukan oleh suatu organisasi, terdapat representasi dari kemampuan aplikasi-aplikasi pada sistem milik organisasi bersangkutan. Hal ini bertujuan untuk dapat menjalankan *Runtime Service* pada proses komunikasi antarkomputer, kemampuan operasional, pemantauan sistem (*System Monitoring*), serta eksekusi proses integrasi pada level aplikasi.

Integrasi di level *Middleware* juga membutuhkan sumber daya (*Resource*), baik mencakup sumber daya komputasi maupun Sumber Daya Manusia (SDM), agar proses integrasi dapat berjalan dengan baik. Berdasarkan kepada jenis dari sumber daya (*Resource*) yang digunakan, maka integrasi level *Middleware* dapat dipetakan ke dalam empat domain integrasi berbeda. Keempat domain integrasi tersebut meliputi:

1. *Cloud integration*

Pada domain *Cloud Integration*, sumber daya yang digunakan untuk proses integrasi pada level *Middleware* adalah ketiga jenis layanan pada *Cloud Computing* (*IAAS Cloud* untuk layanan infrastruktur, *PAAS Cloud* untuk layanan pengembangan *platform*, dan *SAAS Cloud* untuk layanan aplikasi) serta empat model *deployment* pada *Cloud Computing* (*Private Cloud*, *Hybrid Cloud*, *Public Cloud*, *Community Cloud*). Domain ini juga tidak menutup kemungkinan untuk memanfaatkan teknologi *Web Service* sebagai media layanan dan *deployment Cloud Computing*, serta kolaborasi dengan *Business to Business (B2B)*.

2. *Application integration*

Pada domain *Application Integration*, sumber daya yang digunakan untuk proses integrasi pada level *Middleware* adalah aplikasi-aplikasi yang saling berkomunikasi satu sama lain (dengan meminimalkan peran dari manusia), dalam bentuk *Application to Application*. Jenis aplikasi yang digunakan, umumnya terdiri atas aplikasi berbasis *Cloud Computing* (memanfaatkan jenis layanan *SAAS Cloud*) serta aplikasi untuk *Remote System* (memanfaatkan protokol pada jaringan komputer, baik HTTP, FTP, SSH, maupun teknologi *Web Service*).

3. *Data integration*

Pada domain *Data Integration*, sumber daya yang digunakan untuk proses integrasi pada level *Middleware* adalah data-data bisnis milik organisasi bersangkutan, yang didukung dengan aplikasi dan sistem untuk mengelola data-data tersebut. Beberapa di antaranya adalah *database*, file bisnis, aplikasi *Business Intelligence*, dan aplikasi *Data Warehouse*.

4. *Business integration*

Pada domain *Business Integration*, sumber daya yang digunakan untuk proses integrasi pada level *Middleware* adalah pihak-pihak yang terlibat bisnis dengan organisasi, yang membentuk hubungan *Business to Business* (B2B). Di antaranya meliputi konsumen, pelanggan, distributor/*suppliers*, serta perusahaan-perusahaan lain yang menjadi partner bisnis.

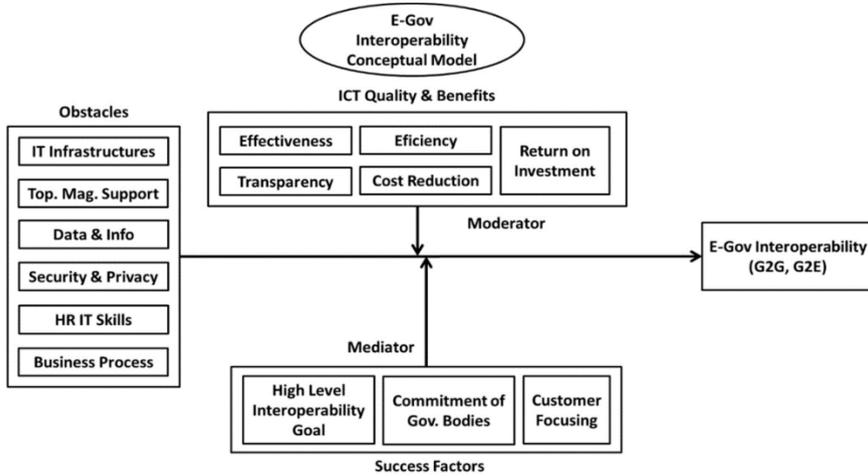
A square box with a double border. Inside, the word "BAB" is at the top, and the number "10" is in the center, both in white text on a dark gray background.

BAB
10

INTEROPERABILITY AND ITS APPLICATIONS IN ENTERPRISE

Interoperability merupakan hal yang harus diperhatikan saat melakukan integrasi *system digital*. "Interoperabilitas" dicapai hanya jika interaksi antara dua sistem setidaknya dapat berlangsung di tiga tingkat: data, sumber daya, dan proses bisnis dengan semantik yang didefinisikan dalam konteks bisnis. Interoperabilitas juga mengacu pada kemampuan sistem (atau proses) untuk menggunakan informasi dan/atau fungsionalitas sistem (atau proses) lain dengan mengikuti standar umum. Secara umum, interoperabilitas adalah kemampuan dua atau lebih entitas yang berbeda (misalnya, unit bisnis, proses, perangkat lunak, dan sistem) untuk melakukan interoperasi. Interoperabilitas membantu mewujudkan integrasi. Ini mempromosikan gagasan bahwa integrasi harus dicapai dengan menggunakan kerangka kerja, arsitektur, dan standar yang relevan.

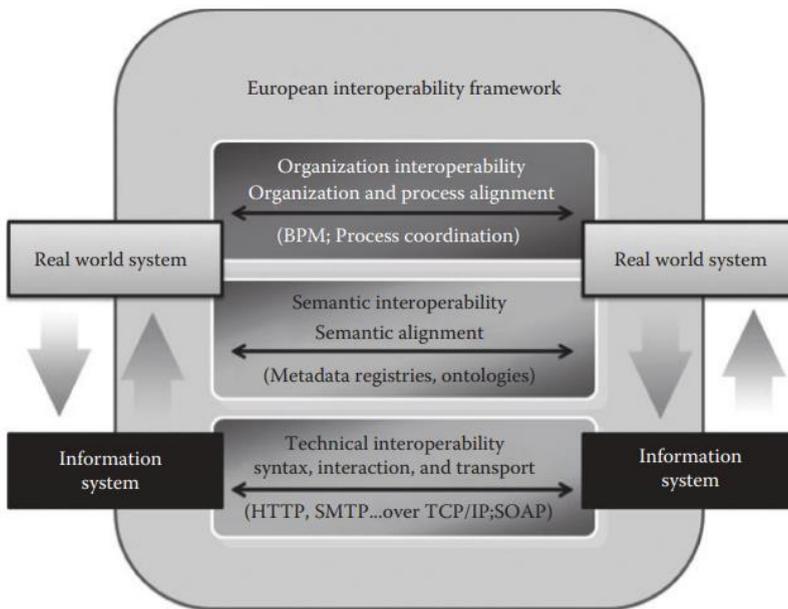
Tampaknya ada kebutuhan untuk klarifikasi konseptual dan pengembangan gagasan interoperabilitas *e-government*. Pertama, konsep integrasi dan interoperabilitas perlu dibedakan secara konseptual. Integrasi (sebagai konsep yang lebih abstrak) bisa berarti menyatukan beberapa bagian dan menjadikannya lubang yang koheren. Integrasi tersebut dapat berarti bahwa sistem informasi yang berbeda diintegrasikan ke dalam satu sistem. Integrasi bagaimanapun juga bisa berarti bahwa bagian-bagian tetap sebagai entitas yang terpisah tetapi mereka bekerja sama dengan cara yang berfungsi dengan baik (sebagai sistem federasi). Sistem *interoperable* adalah sistem yang dapat bekerja sama, tetapi tetap merupakan sistem yang terpisah. Mereka dihubungkan bersama melalui beberapa pertukaran data.



Gambar 10.1 *egovernment interoperability conceptual model* (Sulehat & Taib, 2016)

Interoperabilitas antar sistem yang heterogen dapat dipahami dan melibatkan:

- Interoperabilitas Teknis, yang terkait dengan komponen, sistem, dan platform perangkat keras/lunak yang memungkinkan terjadinya komunikasi mesin-ke-mesin. Interoperabilitas semacam ini sering berpusat pada protokol (komunikasi) dan infrastruktur yang diperlukan agar protokol tersebut dapat beroperasi.
- *Syntactical Interoperability*, yang berhubungan dengan format data.
- Interoperabilitas Semantik, yang dikaitkan dengan makna konten dan menyangkut interpretasi manusia daripada mesin dari konten.
- Interoperabilitas Organisasi, yang merupakan kemampuan organisasi untuk berkomunikasi secara efektif dan mentransfer (bermakna) data (informasi) di berbagai sistem informasi melalui infrastruktur yang sangat berbeda. Interoperabilitas organisasi tergantung pada tiga yang pertama.



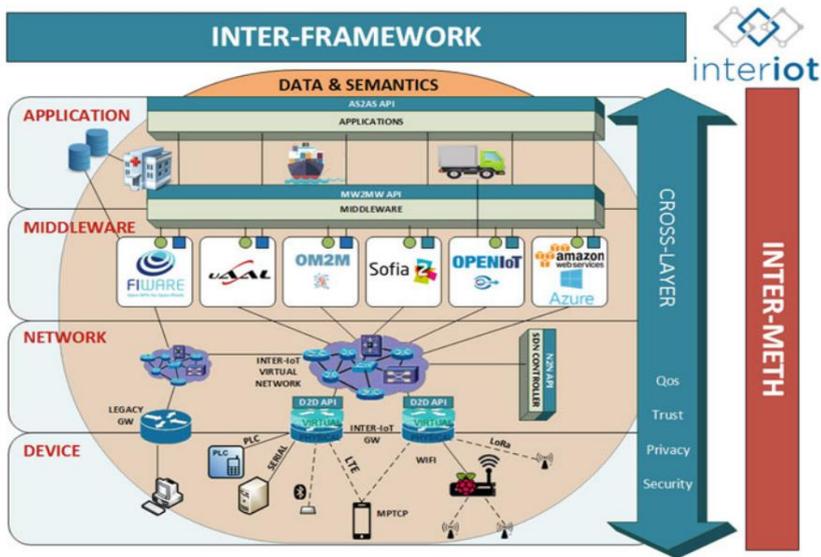
Gambar 10.2 *European interoperability framework* (Li Da Xu, 2014)

Menurut beberapa kerangka kerja, dapat dianggap bahwa masalah interoperabilitas terutama berkaitan dengan masalah konseptual, teknis, dan organisasi:

- Pada tingkat konseptual, data, sumber daya, dan proses bisnis dari sistem informasi yang berbeda harus dihubungkan meskipun strukturnya heterogen dan interpretasinya berbeda. Masalahnya adalah sintaksis dan semantik.
- Pada tingkat teknis, tujuannya adalah untuk menyatukan berbagai aplikasi, teknologi, sistem, dan infrastruktur komunikasi yang digunakan oleh para mitra.
- Di tingkat organisasi, konteks bisnis kolaborasi harus dijelaskan: Bagaimana mitra berinteraksi? Data mana yang dipertukarkan? Sumber daya mana yang mereka ekspos kepada orang lain? Model proses dan data adalah contoh solusi untuk memodelkan interoperabilitas pada level ini.

Solusi yang diadopsi oleh INTER-IoT mencakup tiga solusi utama untuk memberikan interoperabilitas sukarela (lihat Gambar 10.3):

- *INTER-LAYER*: metode dan alat untuk menyediakan interoperabilitas di antara dan di setiap lapisan (gerbang/perangkat virtual, jaringan, *middleware*, layanan aplikasi, data, dan semantik) platform IoT. Secara khusus, kami akan mengeksplorasi *gateway* nyata/virtual untuk komunikasi perangkat-ke-perangkat, sakelar virtual berbasis SDN untuk interkoneksi jaringan-ke-jaringan, super *middleware* untuk integrasi *middleware* ke *middleware*, dan broker layanan untuk orkestrasi lapisan layanan, dan mediator semantik untuk data dan interoperabilitas semantik.



Gambar 10.3 The *INTER-IoT* abstract architecture highlighting *INTER-IoT* multi-layer solutions (device, networking, middleware, application, data and semantics), their interconnection (cross-layer), and their tools (*INTER-FrameWork* and *INTER-METH*)

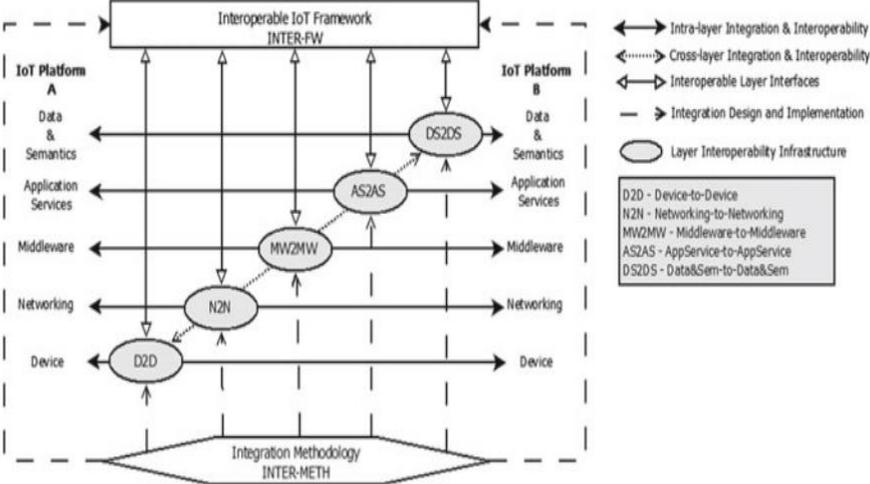
- *INTER-FW*: kerangka kerja global (berdasarkan meta-arsitektur dan model meta-data yang dapat dioperasikan) untuk memprogram dan mengelola platform IoT yang dapat dioperasikan, termasuk API untuk mengakses komponen *INTER-LAYER* dan memungkinkan pembuatan ekosistem aplikasi dan layanan IoT. *INTER-FW* akan menyediakan fungsi manajemen yang dikhususkan untuk interkoneksi antar lapisan. API yang disediakan mencakup fitur keamanan dan privasi dan akan mendukung pembuatan komunitas pengguna dan pengembang.

- *INTER-METH*: metodologi rekayasa berdasarkan alat CASE (*Computer Aided Software Engineering*) untuk secara sistematis mendorong integrasi/interkoneksi *platform* IoT heterogen yang tidak dapat dioperasikan.

A. INTER-LAYER

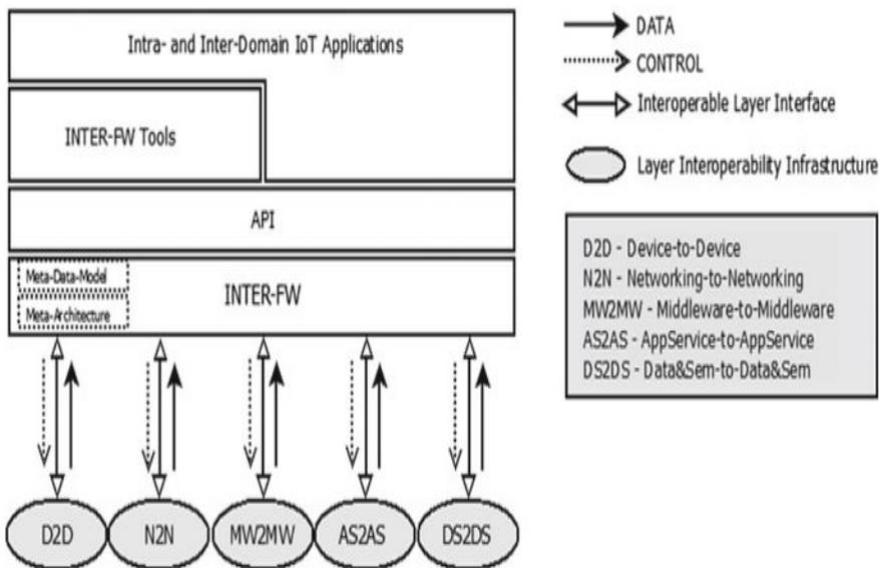
Berbeda dari pendekatan interoperabilitas saat ini, INTER-IoT menggunakan pendekatan berorientasi lapisan untuk sepenuhnya mengeksplorasi fungsi spesifik setiap lapisan (perangkat, jaringan, *middleware*, layanan aplikasi, data, dan semantik). Meskipun pengembangan pendekatan berorientasi lapisan merupakan tantangan penelitian, dibandingkan dengan pendekatan tingkat aplikasi, pendekatan ini memiliki potensi yang lebih tinggi untuk menghadirkan integrasi dua arah yang ketat di antara platform IoT yang heterogen, terutama menjamin independensi, sehingga memberikan kinerja, modularitas, kemampuan beradaptasi yang lebih tinggi, fleksibilitas, keandalan, keamanan, privasi, dan kepercayaan. Selain itu, yang sangat penting, INTER-IoT akan memberikan kontrol lebih besar atas kebutuhan fungsional dan *non-fungsional*.

Seperti yang disorot pada Gambar 10.4, pendekatan INTER-IoT didasarkan pada infrastruktur interoperabilitas lapisan nyata/virtual berikut di antara lapisan rekan yang menjamin interoperabilitas/integrasi: Perangkat-ke-Perangkat (D2D); Jaringan-ke-Jaringan (N2N); *Middleware-ke-Middleware* (MW2MW); Layanan Aplikasi-ke-Aplikasi Layanan (AS2AS); Data dan Semantik-ke-Data dan Semantik (DS2DS).



Gambar 10.4 Abstract schema of INTER-LAYER

Setiap infrastruktur interoperabilitas lapisan (dapat diterapkan dalam perangkat keras, perangkat lunak, atau keduanya) tidak hanya menyediakan sambungan yang kuat antara lapisan rekan tetapi juga memperlihatkan antarmuka, yang dapat diprogram untuk mengontrol/berinteraksi dengan komponen. Antarmuka akan dikendalikan oleh kerangka meta-level untuk menyediakan interoperabilitas global (lihat Gambar 10.5). Selain itu, infrastruktur interoperabilitas lapisan dapat berkomunikasi satu sama lain untuk menyediakan *cross-layering* yang bertujuan untuk memperkuat integrasi antar lapisan sehingga memberikan efisiensi dan keandalan yang lebih tinggi, dengan tetap mendukung fleksibilitas dan kemampuan beradaptasi. Komponen lintas lapisan sepenuhnya ditujukan untuk dukungan dan koordinasi mekanisme dan layanan keamanan dan privasi untuk seluruh *INTER-Layer*.



Gambar 10.5 Abstract schema of *INTER-FW*

- *Device*

Karena sensor, aktuator, dan perangkat pintar menjadi lebih kecil, lebih fleksibel, biaya lebih rendah, dan lebih hemat daya, mereka digunakan dalam jumlah yang lebih besar, baik sebagai perangkat tujuan khusus atau disematkan ke produk lain. Dengan penyatuan dan konvergensi dari sejumlah besar *platform* yang telah digunakan, aksesibilitas (API dan antarmuka) *platform* ke pengembang aplikasi, memerlukan interoperabilitas. Ponsel cerdas adalah komponen utama dalam komunikasi dan interoperabilitas

Perangkat-ke-Perangkat (D2D), namun, ada banyak jenis perangkat lain yang saat ini digunakan, baik secara independen (misalnya, jam tangan pintar dan perangkat yang dapat dikenakan lainnya) maupun sebagai bagian dari perangkat dan *platform* lain (misalnya, elektronik konsumen atau Sistem Cyber-Fisik).

- *Networking*

Produk IoT akan mencakup skenario komunikasi data yang berbeda. Beberapa mungkin melibatkan sensor yang jarang mengirim paket data kecil dan tidak memprioritaskan pengiriman tepat waktu. Lainnya mungkin melibatkan penyimpanan untuk mempertahankan periode ketika tautan komunikasi terputus (mis., Jaringan Toleran Penundaan). Orang lain mungkin membutuhkan *bandwidth* tinggi tetapi dapat menerima latensi tinggi. Dan yang lain mungkin membutuhkan kualitas tinggi, *bandwidth* tinggi, dan latensi rendah. Sejumlah besar lalu lintas dengan ukuran paket yang relatif pendek akan membutuhkan manajemen lalu lintas yang canggih dan prosedur rekayasa lalu lintas. Protokol dan mekanisme manajemen yang lebih efisien akan membantu mengurangi *overhead* tetapi dapat menghadirkan tantangan bagi integritas sistem, keandalan, dan skalabilitas. Standarisasi antarmuka diinginkan agar objek IoT dapat berkomunikasi dengan cepat dan efisien, dan memungkinkan mobilitas antar platform IoT yang dapat dioperasikan. Objek IoT akan membutuhkan cara untuk menemukan satu sama lain dengan cepat dan mudah dan mempelajari kemampuan tetangga mereka.

- *Middleware*

Middleware, banyak digunakan dalam sistem terdistribusi konvensional, adalah alat dasar untuk desain dan implementasi perangkat IoT dan sistem IoT. Mereka menyediakan abstraksi umum dan khusus (misalnya, model komputasi objek, komunikasi antar-objek, antarmuka sensorik/aktuasi, layanan penemuan, manajemen pengetahuan), serta alat pengembangan dan penyebaran, melalui perangkat IoT, sistem IoT, dan aplikasi terkaitnya. dapat dengan mudah dibangun. Memang, *middleware* (i) memungkinkan konektivitas untuk sejumlah besar komponen beragam yang ada di Lapisan Perangkat, (ii) mewujudkan interoperabilitas tanpa batas di Lapisan Jaringan, dan (iii) memastikan transparansi operasional di Lapisan Layanan Aplikasi. Sedemikian rupa, perangkat IoT dan sistem IoT yang heterogen, sering kali kompleks, dan sudah ada, milik domain aplikasi yang berbeda dan awalnya tidak dirancang untuk dihubungkan, dapat dengan mudah diintegrasikan, dikelola secara efektif, dan dieksploitasi bersama. Oleh karena itu, peran *middleware* dalam skenario IoT cyber-fisik, heterogen, berskala besar, dan

saling berhubungan bahkan lebih penting daripada dalam sistem terdistribusi konvensional.

- *Application Services*

Terlepas dari tingkat abstraksi, kesamaan yang mereka semua miliki adalah bahwa mereka semua berjalan dalam jaringan yang dirancang untuk *bandwidth* tinggi dan penundaan singkat, yang asing bagi jaringan IoT yang dirancang untuk efisiensi dan konsumsi rendah, dan di mana kegagalan komunikasi intermiten diharapkan. Layanan *cloud* untuk IoT memiliki spektrum yang luas termasuk penyimpanan data, sinkronisasi informasi, analisis data, dan komunikasi M2M, serta lainnya yang lebih spesifik, seperti geolokasi atau *streaming*. Sering kali, mereka tidak spesifik IoT, dan dapat sangat diuntungkan dari lapisan yang mengelola interaksi antara keduanya, sehingga memudahkan kedua belah pihak untuk beroperasi dengan cara yang efisien. Misalnya, sensor berdaya rendah yang sangat sederhana dapat menggunakan layanan berbasis *cloud* tingkat sangat tinggi tanpa menurunkan masa pakai baterainya melalui interoperabilitas di jaringan, *middleware*, dan beberapa layanan aplikasi *caching* tambahan. Itu adalah mekanisme yang sangat kuat yang memungkinkan kami untuk menghubungkan jaringan IoT yang ada dengan Layanan *Cloud* yang ada tanpa perlu memodifikasi juga.

- *Data and Semantics*

Interoperabilitas semantik dapat dikonseptualisasikan sebagai pendekatan untuk memfasilitasi "penggabungan" beberapa platform IoT. Kasus paling sederhana, menggabungkan dua platform IoT, dapat diatasi dengan mengembangkan penerjemah satu-ke-satu ("gerbang") untuk memungkinkan "pemahaman semantik" di antara mereka. Namun, pendekatan ini tidak berskala, seperti untuk setiap entitas berikutnya yang bergabung dengan perakitan platform N. Dengan demikian, N penerjemah harus dibuat. Dua pendekatan utama untuk menghindari masalah ini, dan menangani interoperabilitas semantik adalah: (i) standar komunikasi umum; (ii) pemrosesan data ontologi dan semantik.

- *Cross layering*

INTER-IoT secara khusus bertujuan untuk menciptakan interoperabilitas dan integrasi lintas-lapisan antara platform IoT yang heterogen. Pendekatan lintas lapisan sangat penting untuk membuat *interoperable*/mengintegrasikan seluruh lapisan dalam tumpukan (perangkat, jaringan,

middleware, layanan aplikasi, data, dan semantik) platform IoT. Oleh karena itu, pelapisan silang akan didasarkan pada hasil poin sebelumnya.

Selain itu, persyaratan dan fitur penting seperti *Quality of Service* (QoS), *Quality of Experience* (QoE), Keamanan, Privasi, Kepercayaan, dan Keandalan, perlu ditangani pada setiap lapisan dengan mekanisme yang berbeda. Pendekatan transversal semacam itu memungkinkan mempertahankan manfaat arsitektur berlapis (misalnya, modularitas, interoperabilitas, dll.) tetapi menambahkan, pada saat yang sama, fleksibilitas (misalnya, optimasi, desain yang dapat disesuaikan, dll.) untuk komponen-komponen yang membutuhkannya. Mempertimbangkan heterogenitas dan penyebaran perangkat IoT dan aplikasi IoT, jelas bahwa pilihan desain seperti itu lebih dari cocok untuk mendukung (i) QoS dan QoE dinamis (yang pertama, pada dasarnya bertujuan untuk membagi lalu lintas menjadi kelas prioritas dan mencoba untuk menjamin metrik kinerja tertentu, yang terakhir menggabungkan aspek yang lebih subjektif terkait dengan persepsi pengguna dalam mengevaluasi layanan); (ii) teknik keamanan dan privasi baru (yang mempertimbangkan sifat fisik siber perangkat IoT serta konteks aplikasi IoT); model kepercayaan yang diperluas (di mana aktor yang tidak konvensional, seperti jejaring sosial, memainkan peran penting) dan (iv) mekanisme keandalan yang ditingkatkan (untuk menangani kegagalan perangkat IoT yang terbatas sumber daya, kurangnya jangkauan dari jaringan akses di beberapa wilayah, aplikasi yang cepat beralih konteks, dll).

B. INTER-FW

Kerangka Kerja IoT *Interoperable* (*INTER-FW*) bertujuan untuk menyediakan interoperabilitas tingkat platform global dan terbuka di antara *platform* IoT heterogen yang digabungkan melalui LII yang dikembangkan secara khusus. *INTER-FW* akan mengandalkan meta-model arsitektur untuk *platform* yang dapat dioperasikan IoT dan pada model metadata untuk semantik yang dapat dioperasikan IoT. Gambar 10.5 menunjukkan skema abstrak *INTER-FW*. Ini menyediakan perpustakaan pemrograman (yaitu, *INTER-API*) yang akan digunakan baik oleh alat *INTER-FW*, menyediakan manajemen tingkat global dari *platform* IoT terintegrasi, dan oleh aplikasi IoT baru, mungkin lintas domain, yang dikembangkan di atas *INTER-FW* dan yang akan dikembangkan di WP4 (*work package 4*), sepenuhnya sesuai dengan model meta-arsitektur dan metadata *INTER-IoT* yang dirancang.

Dengan demikian, *INTER-FW* memajukan *negara-of-the-art* dengan menyediakan metode umum dan efektif untuk interoperabilitas antar-*platform*, menangani di tingkat global: *real-timeliness*, keandalan, keamanan, privasi, dan kepercayaan. Secara khusus, *INTER-FW* akan dirancang dan

diimplementasikan dengan mempertimbangkan kebutuhan untuk menghormati, jika berlaku, privasi data pengguna (misalnya, anonimisasi, ID tersembunyi, penggunaan basis data terpisah untuk identifikasi dan konten data dengan akses terkontrol) dan akses aman ke data (hanya perangkat resmi, memastikan otentikasi dan *non*-penolakan). Selanjutnya, akses ke data oleh pihak yang tidak berwenang harus dicegah (terutama yang jahat). Setiap persyaratan fungsional dan *non*-fungsional lainnya (misalnya, keandalan atau keramahan pengguna) akan dimasukkan ke dalam spesifikasi dan implementasi *INTER-FW*, termasuk alat, *INTER-API*, dan mesin fleksibel interoperabilitas. *INTER-FW* juga mencakup mekanisme manajemen dan API yang digunakan untuk mengakses dan mengoordinasikan berbagai lapisan *INTER-IoT*. Aspek *INTER-FW* ini terutama diperlukan untuk aspek-aspek seperti penemuan, pendaftaran perangkat dan objek pintar, dan juga untuk manajemen keamanan dan privasi.

C. *INTER-METH*

Metodologi rekayasa *INTER-METH* bertujuan untuk mendukung proses integrasi platform IoT yang heterogen untuk mendapatkan interoperabilitas di antara mereka dan memungkinkan implementasi dan penyebaran aplikasi IoT di atasnya. Sampai saat ini, tidak ada proposal yang menyediakan metodologi sistematis yang mendorong implementasi integrasi. Diakui secara luas bahwa menggunakan metodologi rekayasa merupakan hal mendasar dalam domain aplikasi rekayasa apa pun (misalnya, rekayasa perangkat lunak, desain bersama perangkat keras/lunak, teknik sipil, dll). Penerapan teknik, metode, dan kerangka kerja yang kompleks secara manual dan *non*-sistematis kemungkinan besar akan menyebabkan peningkatan tingkat kesalahan selama integrasi. Proses *INTERMETH* ditunjukkan pada Gambar 10.3. Ini dibayangkan sebagai iteratif, termasuk enam fase berikut: Analisis, Desain, Implementasi, *Deployment*, Pengujian, dan Pemeliharaan. Setiap fase menghasilkan produk kerja yang merupakan *input* untuk fase berturut-turut. Iterasi dapat melibatkan fase tunggal, serangkaian fase berturut-turut, atau seluruh proses, sehingga memastikan kemampuan beradaptasi dengan elemen baru.

Khususnya pada:

- 1) Fase Analisis mendefinisikan persyaratan integrasi, baik fungsional maupun *non*-fungsional (misalnya, ketepatan waktu nyata, keandalan, keamanan, privasi, kepercayaan).
- 2) Fase Desain menghasilkan desain integrasi dalam bentuk diagram (i) infrastruktur interoperabilitas lapisan dan antarmuka terkait, dan (ii)

pemrograman dan pola manajemen *INTER-FW*, untuk memenuhi persyaratan yang diperoleh.

- 3) Tahap Implementasi berfokus pada implementasi produk kerja desain untuk mendapatkan sistem yang berfungsi penuh (diimplementasikan perangkat keras dan/atau perangkat lunak).
- 4) Fase *Deployment* melibatkan pengaturan operasi dan konfigurasi *platform* IoT terintegrasi.
- 5) Fase Pengujian memungkinkan melakukan pengujian untuk memvalidasi *platform* terintegrasi sesuai dengan persyaratan fungsional dan *non-fungsional*.
- 6) Fase Pemeliharaan mengelola peningkatan dan evolusi sistem.

A square graphic with a dark gray background and a white border. Inside, the word "BAB" is written in white at the top, and the number "11" is written in a large, white, serif font in the center.

BAB
11

PENUTUP

A. KESIMPULAN

Era digital sebagai suatu istilah yang digunakan dalam kemunculan teknologi digital, jaringan internet khususnya teknologi informasi. Kemunculan teknologi informasi ini telah membawa dampak pada berbagai bidang kehidupan seperti ekonomi, politik, sosial & budaya, pertahanan dan keamanan. Oleh karena itu, untuk mempermudah konektivitas di zaman yang serba terhubung satu sama lain seperti saat ini dan di masa mendatang, maka konsep integrasi menjadi sesuatu yang krusial demi menjaga performa berbagai aktivitas terkait dengan teknologi informasi. Sistem terintegrasi (*integrated system*) merupakan sebuah rangkaian proses untuk menghubungkan beberapa sistem-sistem komputerisasi dan *software* aplikasi baik secara fisik maupun secara fungsional. *Enterprise application integration* adalah penggunaan perangkat lunak dan prinsip-prinsip arsitektur sistem komputer untuk mengintegrasikan sekumpulan aplikasi komputer. Sedangkan sistem informasi *enterprise* merupakan pengembangan dari konsep yang sudah ada yaitu *executive information system* dan DSS yang diperluas untuk domain seluruh perusahaan. Integrasi data atau informasi merupakan suatu tindakan yang dilakukan untuk menyatukan atau menggabungkan data atau informasi dari berbagai sumber yang tersebar untuk memenuhi kebutuhan informasi pengguna secara lebih baik penerapan integrasi data yang tepat dapat mendukung rencana dan pengembangan suatu lembaga yang nantinya akan memberikan nilai tambah berupa *competitive advantage* dalam menghadapi persaingan.

DAFTAR PUSTAKA

- Alisha Safira, dkk. (2020, November 9). *Model Data, Metadata dan Kategori Aplikasi Basisdata*. Retrieved from academia.edu: https://www.academia.edu/31467484/Model_Data_Metadata_dan_Kategori_Aplikasi_Basisdata
- B. Manouvrier and L. Ménard, *Application Integration: EAI, B2B, BPM and SOA*, London: ISTE Ltd, 2008.
- Edhy, S. dkk (2015). KAJIAN MODEL DAN PROTOTIPE SCHEMA MATCHING (Studi untuk Menemukan Peluang Pengembangan Model dan Prototipe Baru). *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*. Yogyakarta.
- Gravina, R., Palau, C.E., Manso, M., Liotta, A., Fortino, G. (2018). Erratum to: Integration, Interconnection, and Interoperability of IoT Systems. In: Gravina, R., Palau, C., Manso, M., Liotta, A., Fortino, G. (eds) *Integration, Interconnection, and Interoperability of IoT Systems*. Internet of Things. Springer, Cham. https://doi.org/10.1007/978-3-319-61300-0_11
- H. Abd Rahman Rahim, et all. 2016. Kewirausahaan Lankutan. Lembaga Perpustakaan dan Penerbitan Universitas Muhammadiyah Makassar. <https://media.neliti.com/media/publications/68821-ID-implementasi-prinsip-usability-f-shape-p.pdf>
- https://www.researchgate.net/publication/330779973_EKONOMI_DIGITAL_DALAM_RANGKA_PENINGKATAN_PEREKONOMIAN_INDONESIA
- I. P. A. E. Pratama, *Integrasi dan Migrasi Sistem (Teori dan Praktik)*, Bandung: Informatika, 2016.
- I. P. A. E. Pratama, *Integrasi dan Migrasi Sistem (Teori dan Praktik)*, Bandung: Informatika, 2016.
- I. Putu. Agus. Eka. Pratama, *Integrasi dan Migrasi Sistem (Teori dan Praktik)*, Bandung: Informatika, 2016
- Indriantoro, N. (1996). TRANSFORMASI ORGANISASI DENGAN TEKNOLOGI INFORMASI SEBAGAI ENABLER. *The Journal of Indonesian Economy and Business*, Vol 11, No 1.
- J. Fenner. [Online]. Available: <http://www-icm.cs.ucl.ac.uk/staff/W.Emmerich/lectures/3C05-02-03/aswe21-essay.pdf>. [Accessed 24 July 2022].

- L. D. Xu, *Enterprise Integration and Information Architecture: A Systems Perspective on Industrial Information Integration*, Boca Raton: CRC Press, 2015.
- Li. Da. Xu, *Enterprise Integration and Information Architecture: A Systems Perspective on Industrial Information Integration*, Boca Raton: CRC Press, 2015.
- Nursyihab, I. (2020), November 9). Retrieved from slideshare.net: <https://www.slideshare.net/imamnursyihab/kualitas-informasi>
- Oktavia, T. (2011). PERAN SERTA STRATEGI SISTEM INFORMASI TERHADAP KEBERHASILAN PENERAPAN TEKNOLOGI INFORMASI PERUSAHAAN. *Comtech*, Vol.2 No.1 p.42-51.
- Rahayu, R., Silfia, R., & Juita, V. (2017). Pengaruh Kapabilitas Teknologi Informasi Terhadap Kinerja dan Nilai Perusahaan. *Prosiding SNA XX Jember*.
- Royana, A. (2018). *STRATEGI TRANSFORMASI DIGITAL*. Jakarta: Thesis Binus.
- Selvi. [Online]. Available: <https://sulselprov.go.id/welcome/post/penerapan-integrasi-sistem-dalam-pemerintahan>. [Accessed 25 Januari 2022].
- Vujasinovic, M., Marjanovic, Z. (2006). Data Level Enterprise Applications Integration. In: Bussler, C.J., Haller, A. (eds) *Business Process Management Workshops. BPM 2005. Lecture Notes in Computer Science*, vol 3812. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11678564_35

INDEKS

A

Abraham Maslow 35 Adaptasi karakteristik 29 Adversity intelligence 6 Adversity quotient 11, 17, 61, 66, 73, 74
Affective commitment 57
Aktualisasi diri 36
Akuntan 36
Altruism 13
Amerika Serikat 23
Availability of alternative jobs 56

B

Bakat 64
Bakat seni 29
Bali 12
Bank 10

Basic tendencies 29 Budaya perusahaan 4

C

Civic virtue 13
CO2RE 66
Conscientiousness 13
Continuance commitment 57

D

Digital Generation 23

E

Edwin Locke 39
Emotional Quotient (EQ) 17
Endurance 67
Eropa 24
Etika kerja 26

GLOSARIUM

A

Additive Rule: Aturan atau hukum penjumlahan.

Aggregate: Keseluruhan unsur-unsur atau variabel-variabel

Aksioma: Pernyataan yang dapat diterima sebagai kebenaran tanpa pembuktian.

Almost Nil: Kerugian yang hampir tidak mungkin terjadi

Aproksimasi: Pendekatan atau pembulatan nilai terhadap hasil pengukuran.

B

Burgery: Terminologi dari *burglary* dan/atau *house breaking* didefinisikan sebagai pencurian, yang melibatkan aktivitas 'masuk' atau 'keluar' dari premis bertanggung yang disertai dengan paksaan dan adanya tanda-tanda kerusakan

PROFIL PENULIS

Rika Yuliana, M.T.



Penulis saat ini bekerja sebagai salah satu seorang staf pendidik (dosen) Pegawai Negeri Sipil pada program studi Teknologi Informasi, Fakultas Sains dan Teknologi, Universitas Islam Negeri Ar-Raniry Banda Aceh sejak 2014 hingga sekarang. Sebelumnya penulis sempat bekerja di berbagai instansi swasta baik sebagai praktisi maupun akademisi sehingga hal tersebut turut menambah wawasan penulis dalam pengintegrasian bidang keilmuan teknologi informasi. Lahir dan besar di kota Banda Aceh, pendidik TK hingga SMA diselesaikan di kota Banda Aceh.

INTEGRASI APLIKASI DAN INFORMASI FILOSOFI

Konsep Dan Penerapannya



Era internet of things mengubah wajah dunia secara cepat. Ketika semua serba terkoneksi dan terintegrasi sempurna dalam sebuah sistem informasi, segala sesuatu terkait pekerjaan maupun bisnis lebih mudah diakses. Bahkan, tanpa mengenal perbedaan tempat dan waktu. Dewasa ini, makin banyak perusahaan mengandalkan sistem demikian dalam manajemen operasionalnya. Misalnya, bagaimana sistem ini mempermudah aktivitas harian perusahaan, seperti memproses keuangan, mengelola urusan SDM, sampai menjangkau calon pelanggan lewat digital marketing dan membangun sistem penjualan online. Tanpa sistem seperti ini, Anda tidak akan mengenal aplikasi-aplikasi yang disediakan untuk di akses dan dipergunakan sebaik mungkin. Pemerintah bahkan telah menerapkan sistem terintegrasi ini dalam melayani berbagai keperluan administrasi penduduk. Berkat sistem yang tertata dan terkoneksi satu sama lain, layanan pemerintahan bisa berlangsung efektif dan efisien sehingga warga pun merasa terbantu.

Integrasi sistem informasi merupakan suatu proses teknik menggabungkan berbagai komponen atau subsistem menjadi satu kesatuan sistem yang besar. Prosedur tersebut menghubungkan beberapa komponen terpisah yang kadang diproduksi vendor berbeda. Sistem terintegrasi juga menambah nilai suatu sistem dengan penyediaan fungsi baru. Sistem terintegrasi menjadi solusi tepat bagi perusahaan yang memiliki kesulitan dalam mengoperasikan beberapa subsistem terpisah. Anda pun tidak perlu menguras energi dan membuang waktu untuk memasukkan ulang data ke tiap sistem secara manual. Tentu ini menekan risiko terjadinya kesalahan-kesalahan kecil yang bisa berakibat fatal di kemudian hari. Itulah sebabnya pemakaian sistem informasi meningkat secara drastis di semua lini bisnis. Semua demi menjaga keunggulan perusahaan agar tetap bisa berkompetisi dengan para pesaingnya. Sulit membayangkan bisnis tanpa sistem yang efektif sekaligus terintegrasi. Menarik untuk ditelusuri bagaimana perusahaan menekankan pentingnya integrated information system dan apa saja manfaatnya untuk keberlangsungan bisnis yang dijalani, maka dari itu buku ini layak untuk dikonsumsi semaksimal, terutama pelajar sebagai cikal bakal penerus yang akan melanjutkan semua generasi pada masing-masing bidangnya.