

# BAB 1

## Perkembangan Web Service

*Service Oriented Architecture* (SOA) (McGovern dkk, 2003) merupakan paradigma yang baru muncul untuk aplikasi terdistribusi dan pemrosesan e-business yang berasal dari pemrograman berorientasi object dan pemrograman berbasis komponen. Satu tujuan utama SOA adalah memungkinkan developer untuk membangun jaringan terintegrasi dan aplikasi kolaboratif, tanpa memperhatikan baik platform dimana aplikasi atau service berjalan dan bahasa pemrograman yang digunakan untuk mengembangkan. Web Service merupakan teknologi pendukung SOA. Web Service menyediakan platform untuk pengembangan aplikasi dengan memanfaatkan infrastruktur internet.

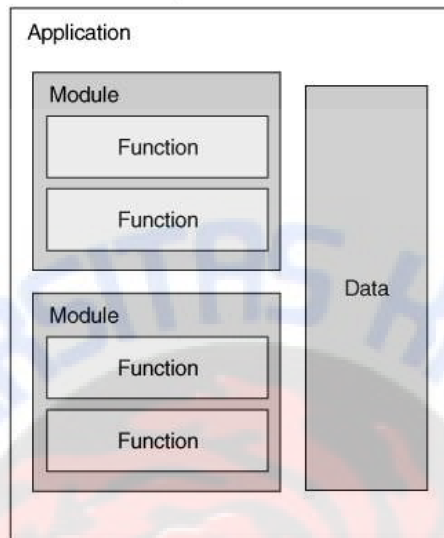
### 1.1 Evolusi software

SOA (Estefan, 2000) merupakan rangkaian panjang dalam upaya rekayasa software menekankan penggunaan komponen software. Langkah pertama evolusi ini dikembangkan melalui konsep fungsi. Dalam konsep fungsi, program dipecah kedalam subprogram dan penulisan kode dipusatkan pada API (*application programming interface*). API merepresentasikan kontrak yang harus dijalankan oleh komponen software.

*Structured design and development* (Yourdon dan Constantine, 1975), dapat dilihat pada diagram gambar 2.1. yaitu pemecahan proses yang lebih besar ke proses yang lebih kecil. Tujuannya adalah untuk mengurangi kompleksitas dan meningkatkan kemampuan penggunaan ulang. Perancangan terstruktur menekankan pada perilaku system software yang terpisah dari data. Pemecahan sistem ke lebih kecil membantu mengembangkan aplikasi yang lebih kompleks, tetapi pengelolaan data di dalam aplikasi termasuk sulit, karena fungsi yang berbeda bekerja pada data yang sama.

Pada model pengembangan terstruktur penyembunyian informasi dilakukan dengan menyembunyikan struktur program dan proses tetapi tidak menyembunyikan rincian data di dalam program. Prinsip perancangan ini dikenal sebagai *information hiding* (Parnas 1972). Namun pengembangan berorientasi objek memungkinkan pengembang menyembunyikan data dan *behavior* ke dalam objek.

## Structured Development



**Gambar 1.1 Pengembangan Terstruktur yang melibatkan pemecahan dari proses lebih besar ke proese lebih kecil. (McGovern dkk, 2003)**

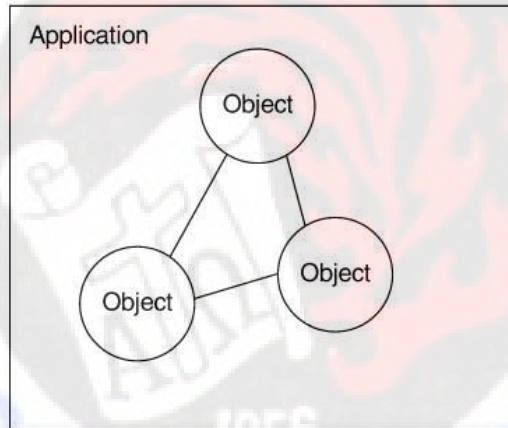
Langkah utama kedua adalah pengembangan konsep objek. Sebuah objek merupakan blok bangunan dasar yang berisi data dan fungsi didalam unit yang terbungkus. Dengan paradigma berorientasi objek ide tentang kelas, pewarisan dan polimorfisme mulai diperkenalkan.

*Object-oriented development* (Booch 1990), yang disajikan dalam Gambar 1.2., memungkinkan desainer software dan developer untuk membungkus (*encapsulate*) data dan *behavior* ke dalam kelas dan objek. Keuntungan tambahan dari Orientasi Objek ini adalah bahwa struktur software lebih mudah dipetakan ke entitas dunia nyata.

Pengembangan berorientasi objek meningkatkan prinsip penyembunyian informasi.

Namun sejumlah besar objek yang saling terkoneksi menciptakan ketergantungan yang akan sulit dikelola. Objek secara relative bersifat *fine-grained*. Walaupun interface mengendalikan akses ke internal objek, jumlah besar dari objek yang *fine-grained* membuat ketergantungan yang sulit dikendalikan dalam aplikasi besar. Pengembangan berbasis komponen membantu membangun aplikasi berorientasi objek yang besar.

Object Based Development

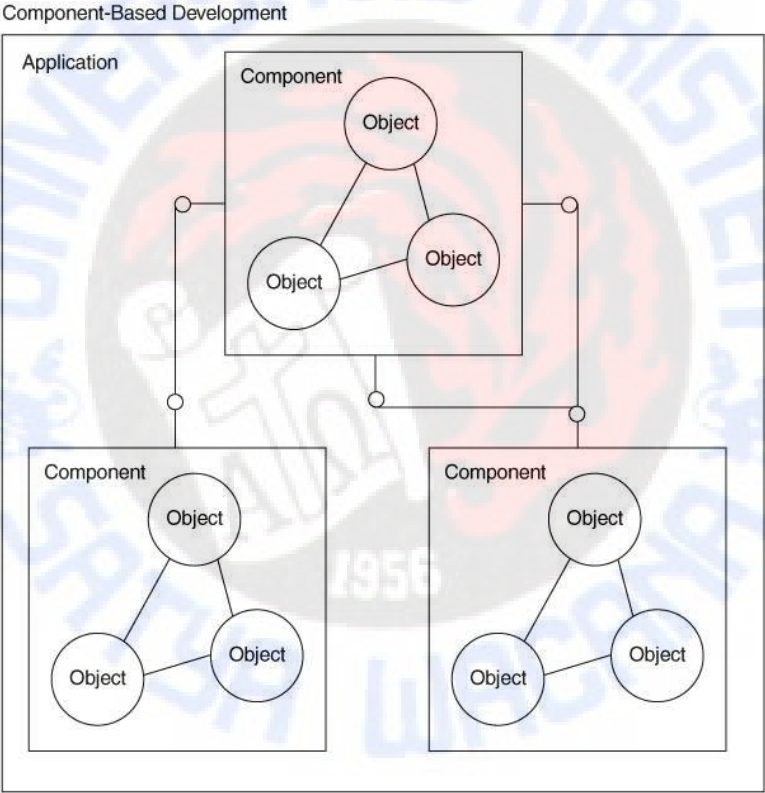


**Gambar 1.2 Pengembangan berorientasi objek yang membungkus data dan *behavior* ke dalam kelas dan objek. (McGovern dkk, 2003)**

Komponen lebih besar dan *coarser grained* daripada objek dan tidak memungkinkan program mengakses private data, *structure*, atau state. *Component-based development* memungkinkan developer membuat system berkualitas tinggi yang lebih kompleks daripada

sebelumnya, sebab pendekatan ini lebih baik dalam mengelola kompleksitas dan ketergantungan di dalam sebuah aplikasi.

Szyperski (1998) mendefinisikan komponen sebagai kelompok kecil objek dengan interface yang ditentukan secara kontrak, bekerja bersama untuk menyediakan fungsi aplikasi, seperti yang tampak pada Gambar 1.3.

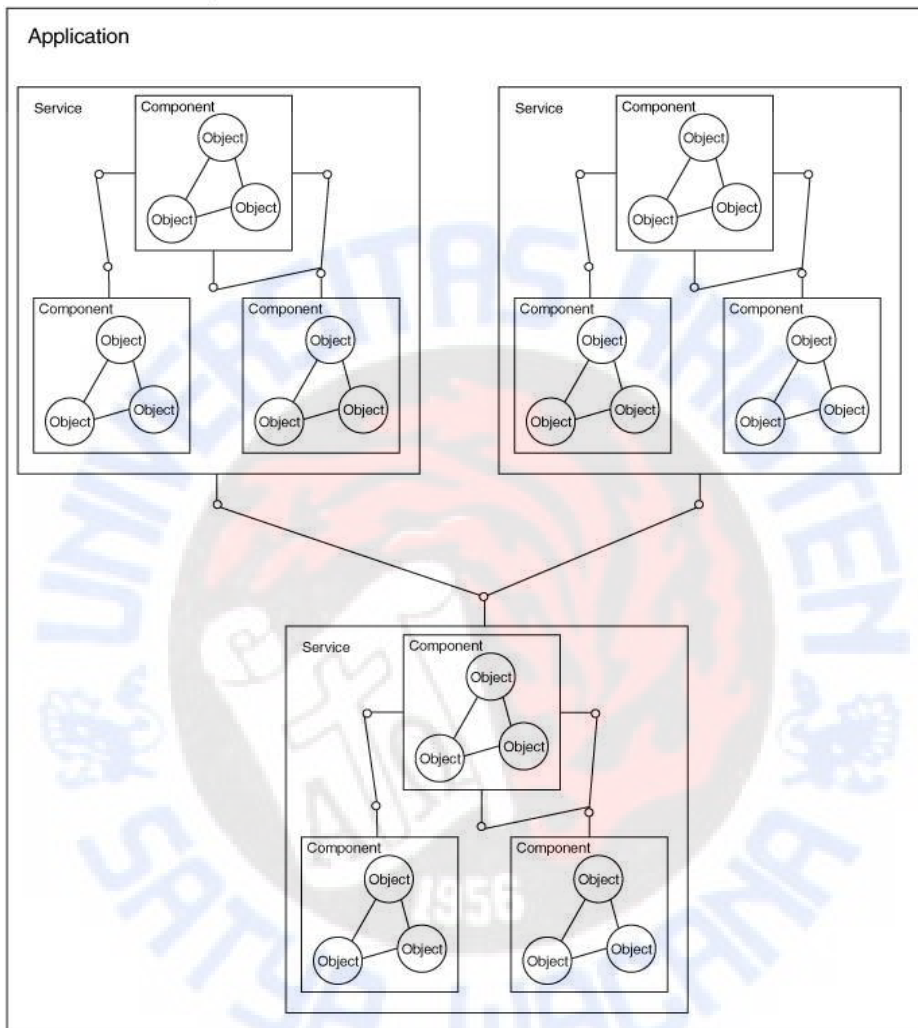


**Gambar 1.3 Pengembangan berbasis komponen yang melibatkan kelompok kecil objek dengan interface, bekerja bersama menyediakan fungsi aplikasi. (McGovern dkk, 2003)**

Protocols yang digunakan untuk mengakses komponen tidak dapat dengan mudah melewati *firewall*, yang membuat protocol ini interoperabilitasnya kurang. Komponen juga digabungkan ke dalam aplikasi pada waktu developer melakukan compile program (*compile time*). Aplikasi yang menggunakan sebuah komponen harus mempunyai definisi interface pada waktu *compile*, sehingga interface dapat diikat dan dieksekusi pada saat *runtime*. Pengembangan berbasis layanan membantu masalah ini.

Dengan pengembangan berbasis layanan (*service-based development*), seperti dalam Gambar 1.4, layanan biasanya berupa komponen yang dibungkus dalam sebuah lapisan layanan. Sebuah layanan biasanya mempunyai banyak komponen dalam sebuah interface sehingga *coarser-grained*. Konsumer layanan tidak tahu lokasi atau interface layanan sampai runtime. Ini yang dinamakan "[late binding](#)". Konsumer menemukan lokasi layanan saat *runtime* melalui pencarian di *registry*, yang juga berisi penunjuk ke kontrak layanan. Kontrak mendeskripsikan interface untuk layanan dan bagaimana melakukan invoke.

## Service-Based Development



**Gambar 1.4** Dalam pengembangan berbasis layanan, layanan dibungkus dalam lapisan layanan. (McGovern dkk, 2003)

Kontrak layanan ditemukan secara dinamis pada runtime, mengikat dan kemudian mengeksekusi. Feature dari pengembangan

berbasis layanan memungkinkan konsumen menampilkan negosiasi kontrak secara real time dengan beberapa layanan.

Pengembangan berbasis layanan telah memecahkan isu interoperabilitas dengan mengadopsi standard interoperabilitas berbasis web. Web services menggunakan protokol HTTP untuk transfer data dan XML untuk format data. Hal ini memungkinkan layanan melakukan *passing* melalui *firewalls*. Pengembangan berbasis layanan dimungkinkan melalui penemuan internet yang sudah tersebar luas. Pengembangan berbasis layanan juga melakukan penyembunyian informasi antara bagian aplikasi dengan memisahkan interface layanan dari implementasinya.

Potensi penggunaan layanan kembali juga meningkat karena penggunaan teknologi standard yang juga memungkinkan meluasnya penggunaan layanan diantara organisasi. Definisi pesan XML dalam komunitas partner dagang juga mulai distandardkan.

Dengan cara ini kelas dipandang sebagai sebuah pola. Langkah berikut dalam evolusi adalah diperkenalkannya Web Service.

## **1.2 Web Services**

W3C (W3C, 2007) memberikan definisi resmi Web Service yaitu : “Web Service merupakan system software yang diidentifikasi melalui URI yang public interface dan binding nya didefinisikan dan dideskripsikan menggunakan XML. Definisi ini dapat ditemukan oleh system software lain. System ini bisa berinteraksi dengan Web Service



dengan cara yang sudah ditentukan melalui definisi, menggunakan pesan berbasis XML yang dikonversi melalui protocol internet”

Web Services merupakan teknologi yang berkembang secara evolusioner dan tidak seketika. Tidak ada revolusi dalam Web Service, Web Service dilihat sebagai evolusi berbasis protocol internet yang sudah ada. Web Service merupakan langkah berikutnya dalam internet.

Pada awalnya ada HTTP yang merupakan protocol komunikasi untuk mengirim informasi dari satu titik ke titik lain pada internet. Informasi yang dikirim melalui kabel dapat disusun menggunakan XML (eXtensible Markup Language) (W3C, 2007). XML metalanguage mendefinisikan format dan semantic dari informasi dan menjadi fondasi dasar untuk lapisan di atasnya. Framework Web Service didasarkan pada tiga teknologi yaitu SOAP, WSDL and UDDI. SOAP (W3C, 2007) (Simple Object Access Protocol) adalah protocol yang mendefinisikan bagaimana proses harus berinteraksi satu dengan yang lain. SOAP didasarkan pada XML dan mendeskripsikan struktur dokumen yang digunakan untuk meminta (*invoke*) layanan. Web Service memungkinkan menyelesaikan masalah integrasi dengan sistem operasi, model objek dan bahasa pemrograman yang berbeda. WSDL (Christenses, 2001) (*Web Services Description Language*), mendiskripsikan operasi yang dipasok oleh layanan, termasuk parameter dan *return values*. UDDI digunakan untuk melakukan *publish* dokumen.



**Gambar 1.5 Model Pemrograman Web Services (W3C, 2007)**

Model pemrograman Web Service berisi tiga komponen: *service consumers*, *service providers* dan *service registrar*. *Service registrar* bertindak sebagai *intermediary* antara provider dan consumer, sehingga dapat saling menemukan satu sama lain. *Service provider* melakukan *publish service*, *service consumer* mencoba menemukan layanan menggunakan registrar, jika menemukan layanan yang diharapkan maka *service consumer* akan menyusun kontrak dengan *provider* dengan tujuan mengkonsumsi service tersebut. Gambar 2.5 menunjukkan model dan protocol umum pemilihan implementasi interaksi antara komponen seperti SOAP, WSDL dan UDDI.

### 1.3 Service Oriented Architecture

Integrasi system bisnis yang berjalan pada lokasi yang berbeda merupakan aspek paling menantang dalam pengembangan aplikasi dalam sebuah enterprise. Saat ini, arsitektur berorientasi layanan untuk

integrasi aplikasi dan pertukaran informasi di dalam enterprise sedang mencapai momentum. Hal ini dapat dilihat dari penggunaan teknologi internet yang canggih yang mencakup aplikasi bisnis seperti e-commerce, Supply Chain Management, Customer Relationship Management, dsb.

Integrasi aplikasi memerlukan pemahaman domain bisnis dan kebutuhan enterprise dalam jangka panjang, yang memungkinkan solusi dan pembangunan arsitektur yang dapat diterapkan untuk lingkungan komputasi yang berbeda. Untuk membangun arsitektur semacam ini yang melibatkan kombinasi *real-time* dari beberapa aplikasi dan fungsionalitas merupakan proses yang rumit. Pendekatan tradisional untuk integrasi aplikasi seperti kombinasi aplikasi dan basisdata secara langsung tidak lagi mencukupi. Arsitektur Berorientasi Layanan atau *Services Oriented Architecture (SOA)* memecahkan masalah ini dengan menyediakan layanan yang memanfaatkan aplikasi dengan memecah ke level komponen dan kombinasi beberapa pendekatan dan teknologi. Saat ini pendekatan untuk integrasi aplikasi sedang berpindah dari integrasi berorientasi informasi menuju ke integrasi berbasis layanan. Pendekatan berorientasi informasi merupakan integrasi langsung basisdata, *API proprietary*, tanpa perlu merubah struktur aplikasi (Linthicum, 2004). Integrasi berbasis layanan menggabungkan aplikasi pada level layanan yang mencakup strukturnya, transaksi dan objek terdistribusi dengan memanfaatkan teknologi internet untuk membentuk aplikasi gabungan, proses dan aliran informasi dari berbagai sumber daya dalam enterprise.

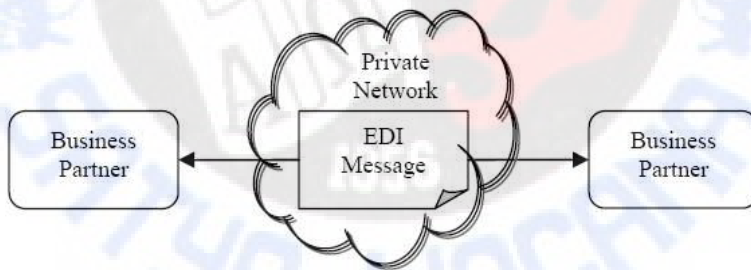
Integrasi berbasis layanan memungkinkan *interface well-defined* yang dipandang sebagai layanan yang tersedia melalui internet. Hal ini juga memerlukan perubahan struktur aplikasi baik target maupun sumber, yang membuat integrasi berbasis layanan ini mejadi eksklusif dan lebih canggih daripada pendekatan tradisional. Pendekatan semacam ini menyediakan peluang dalam mengimplementasi solusi untuk masalah integrasi. Integrasi berbasis layanan memperluas domain aplikasi dari level departemen ke enterprise yang lebih luas dalam penyusunan arsitektur yang memuaskan kebutuhan bisnis.

Integrasi B2B merupakan kombinasi dua atau lebih logika aplikasi dalam membagi aliran informasi dan fungsionalitas aplikasi dengan cara seotomatis mungkin antara partner komunikasi yang berbeda. Partner secara individual mempunyai definisi aturan dan regulasi untuk distribusi layanan dan logika aplikasi dalam enterprise.

Integrasi B2B tradisional melibatkan API yang *strongly-typed* dan tidak fleksibel dalam mendukung perubahan bisnis. Kebutuhan integrasi B2B secara dinamis melibatkan struktur komputasi yang memungkinkan integrasi dari system partner melalui interface.

Interface yang mendukung integrasi B2B diketahui dari aliran proses dan aliran data di dalam organisasi. Karena kebutuhan untuk penggabungan struktur aplikasi menjadi menyeluruh untuk enterprise, maka integrasi berbasis layanan menyediakan mekanisme untuk membagi logika bisnis umum dan membuat aplikasi gabungan dengan meningkatkan layanan dari system remote dan heterogen. Tool dan teknik untuk integrasi B2B memberi peluang partner bisnis untuk

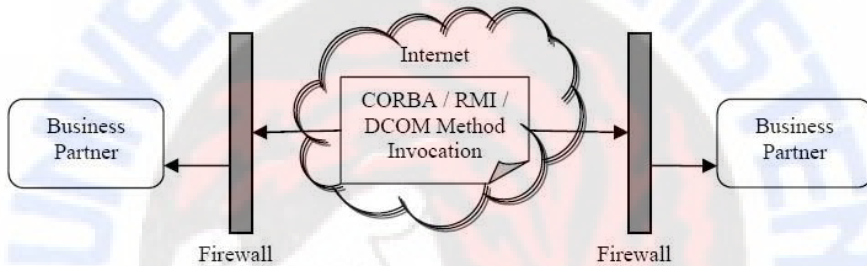
membuat infrastruktur yang layanannya dapat dibuat, dites dan di-deploy. Teknologi awal untuk pengembangan integrasi B2B adalah **Electronic Data Interchange (EDI)** (McGovern, 2003) dibangun untuk memungkinkan pertukaran data elektronik melalui jaringan. EDI mendefinisikan format pesan dan cara pertukaran data diantara partner yang terlibat dalam komunikasi. EDI tidak mendefinisikan rincian jaringan dan protocol untuk enterprise. Organisasi yang menggunakan EDI harus membangun jaringan private dan memungkinkan pihak lain untuk melakukan koneksi. Pendekatan ini cukup umum di periode awal namun mahal dalam implementasinya dan dihasilkan dari struktur aplikasi berhak milik, dan system tertutup. EDI digunakan untuk organisasi besar. Sebelum adanya internet EDI merupakan satu-satunya pilihan untuk e-business dan komunikasi antara dua titik yang terdistribusi secara geografis dan fisik. Gambar dibawah ini menggambarkan pendekatan EDI.



**Gambar 1.6 Electronic Data Interchange dalam Enterprise (McGovern dkk, 2003)**

Sebelum pengembangan Web Service, *CORBA*, *COM/DCOM* dan *RMI* merupakan pendekatan umum untuk komunikasi enterprise,

namun masih ada kesulitan yang membuat teknologi ini tidak diterima secara luas sebagai solusi untuk integrasi enterprise. Keunggulan teknologi ini membawa enterprise menggunakan internet, menjalankan praktek protocol jaringan yang diakui secara umum, interface layanan yang disetujui, dan perkembangan infrastruktur komunikasi dari pendekatan berorientasi informasi sampai ke pendekatan berbasis layanan. Kelemahannya adalah teknologi ini kurang fleksibel karena adanya firewall, dan tidak menyediakan pilihan interoperabilitas.



**Gambar 1.7 CORBA, RMI dan DCOM dalam Enterprise (McGovern dkk, 2003)**

Teknologi *Web services* berhasil memecahkan beberapa masalah yang dihadapi oleh teknologi sebelumnya dalam pengembangan integrasi B2B. SOAP salah satu teknologi Web Service, menstandarisasi format data dan protocol dengan menggunakan XML dengan platform bahasa bebas untuk deskripsi data, dan penggunaan standard teknologi internet yang lain. Web services menyediakan spesifikasi untuk interoperabilitas dan memudahkan integrasi dari aplikasi tersebar untuk komunikasi B2B.



**Gambar 1.8 Simple Object Access Protocol dalam Enterprise (McGovern dkk, 2003)**

Kekurangan utama dengan Web Service adalah Web Service tidak menyediakan kemampuan untuk melakukan kolaborasi bisnis dan pemrosesan informasi. Untuk tujuan ini, **electronic business XML (ebXML)** diusulkan oleh United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) yang bergabung dengan Organization for the Advancement of Structured Information Standards (OASIS) di tahun 2001 (Anonim, 2001). ebXML mengajukan framework berbasis XML untuk percakapan (*conversation*) bisnis dengan lingkungan computer dan definisi proses bisnis antara bermacam-macam organisasi. ebXML menyediakan struktur umum dan syntax pesan lantar enterprise dan standard layanan untuk keamanan, penanganan kesalahan dan eksepsi.

Framework memanfaatkan model proses bisnis yang dapat dikerjakan dengan UML atau bahasa pemodelan lain. ebXML *Business Process Specification Schema* (BPSS) diproduksi sebagai hasil pemodelan proses dengan menerapkan beberapa teknik pemodelan

dan tools, dan dihasilkan sebagai dokumen XML untuk representasi use case.

*Collaboration-Protocol Profile* (CPP) merupakan dokumen XML yang mendeskripsikan fungsi organisasi dan property teknis yang mencakup informasi mengenai proses bisnis, rincian protocol untuk transport data dan pesan dan aspek keamanan dalam lingkungan komputasi. CPP digunakan oleh organisasi lain untuk menemukan layanan yang tersedia dan menjalin kolaborasi dengan organisasi.

*Collaboration-Protocol Agreement* (CPA) merupakan dokumen XML yang digunakan untuk definisi persetujuan (*agreement*) level system untuk pertukaran data antar partner didasarkan atas dokumen CPP. Pada saat pihak-pihak yang terlibat setuju dengan CPA, maka komunikasi dan percakapan (*conversation*) akan terjadi.

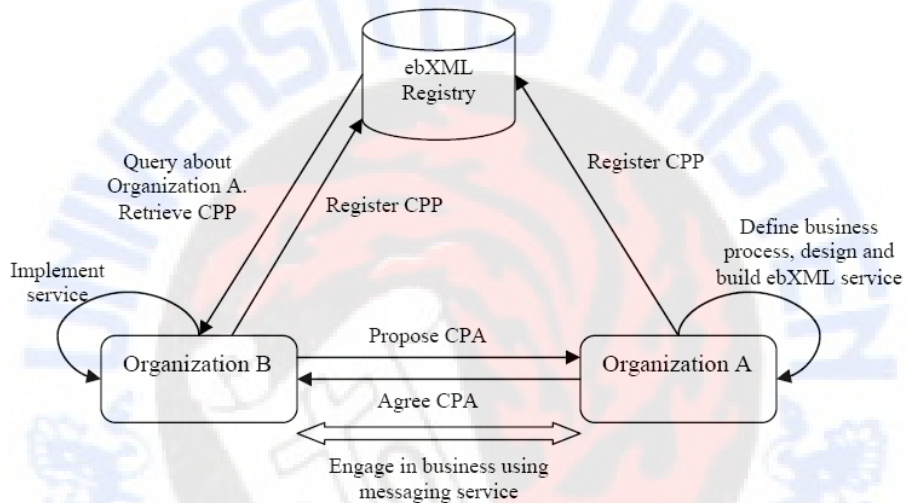
Layanan yang tersedia dan proses dapat ditemukan menggunakan *ebXML registry*, yang menunjukkan kesamaan dengan spesifikasi UDDI. *ebXML registry* menyimpan informasi mengenai organisasi, layanannya sendiri, semantic bisnis, dan proses untuk partner lain dalam menemukan dan memanfaatkannya. Baik *ebXML registry* dan UDDI merupakan spesifikasi dari konsorsium OASIS dan menyediakan fungsionalitas yang overlapping, kecuali untuk beberapa arsitektur yang berbeda.

*ebXML Messaging Service Specification* menyediakan mekanisme penanganan aliran informasi antar organisasi dan mendeskripsikan karakteristik pertukaran pesan bisnis. Paket pesan dalam *ebXML* merupakan ekstensi dari pesan SOAP dan dikerjakan



melalui skema SOAP header. Paket berisi SOAP envelope dan *payload*, yang mendeskripsikan informasi tertentu organisasi yang diperlukan.

Layanan pesan diperlukan saat runtime untuk kolaborasi. Gambar dibawah ini menggambarkan langkah yang disederhanakan dari interaksi partner dalam ebXML.



**Gambar 1.9 Langkah Kolaborasi Bisnis dengan ebXML (McGovern dkk, 2003)**

Teknologi untuk integrasi B2B berlanjut untuk mendukung kebutuhan bisnis sicanggih mungkin, Tabel dibawah ini membandingkan teknologi yang telah dibahas diatas yang digunakan untuk mendukung integrasi e-business.

Web services mencakup teknologi untuk penciptaan layanan, deksripsi dan registri layanan, ebXML melapisi beberapa aspek Web Service dan menyediakan penggunaan proses integrasi B2B sehingga

efektif dan operable. Namun ebXML bukan merupakan solusi umum yang diterima luas. Teknologi Business Process Management System memanfaatkan Web Service dan menyediakan teknologi tambahan untuk kolaborasi bisnis dan penggunaan proses informasi.

#### **1.4 Integrasi Berorientasi Proses Bisnis**

Integrasi berorientasi proses bisnis (*Business process oriented integration*) melibatkan integrasi logika aplikasi yang dibungkus dalam system remote melalui penemuan proses bisnis. Integrasi berorientasi proses bisnis menyediakan mekanisme untuk aktivasi proses dalam urutan sekuensial untuk memenuhi fungsionalitas yang dibutuhkan bisnis.

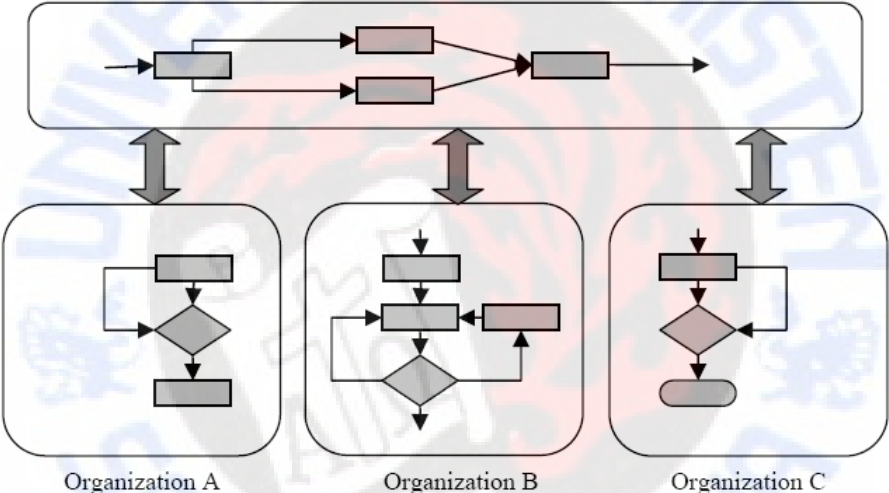
Proses ini bernilai bisnis karena mencerminkan aliran informasi yang akurat dan urutan eksekusi aplikasi bisnis. Mereka dibangun sebagai lapisan terpisah pada aplikasi yang mencakup system berorientasi objek, paket aplikasi, dan layanan, dan mengelola dan mengendalikan sumberdaya enterprise ketika dieksekusi. Proses merupakan asset bisnis yang fungsionalitasnya dideskripsikan sebagai kecil, unit terdesain-baik (*well-designed units*) yang memungkinkan pertukaran data antara system yang berpartisipasi.

Proses bisnis didefinisikan sebagai rangkaian kegiatan yang harus dieksekusi dalam urutan untuk mengerjakan fungsionalitas tertentu atau sasaran bisnis dalam lingkungan struktur organisasi dan

enterprise. Definisi proses mencakup deskripsi jaringan kegiatan, relasinya, unit yang berpartisipasi termasuk aplikasi, organisasi dan orang-orang, aliran data antara kegiatan dan property yang mengelilingi proses seperti kondisi proses untuk memulai dan mengakhiri eksekusi. **Business Process Management (BPM)** menyediakan infrastruktur untuk perancangan, deployment, execution, maintenance, dan monitoring proses bisnis. System BPM digunakan untuk menafsirkan definisi proses, pemodelan, pengembangan dan manajemen proses ketika dieksekusi. System menjamin bahwa *sequence* item kerja yang ditugaskan untuk partisipan dan sumberdaya yang sesuai akan di-*invoke* ketika diperlukan. *Workflow* merupakan otomasi proses bisnis. *Workflow* dapat dilihat bahwa system BPM menyediakan deskripsi yang sama untuk otomasi proses yang disediakan oleh Workflow Management Coalition (WfMC), dan tersebar luas khususnya setelah penemuan Web Service.

Workflow Management Coalition, organisasi internasional non profit, mengidentifikasi struktur produk yang umum dan mengusulkan spesifikasi untuk menggunakan workflow melalui terminology umum, yang memungkinkan interoperabilitas dan konektivitas antar produk (Guner, 2005). System manajemen workflow mendefinisikan dan menciptakan *instance* workflow dari definisi proses, yang memerlukan *process definition tool*, dan mengeksekusi *instance* melalui penggunaan software yang disebut *workflow engine*. Tool definisi proses dibutuhkan untuk membuat dan mendemonstrasikan deskripsi proses dalam bentuk yang dapat dieksekusi mesin. Tool ini memanfaatkan bahasa definisi

proses untuk memodelkan definisi proses dalam bentuk tekstual atau grafis atau notasi bahasa formal. BPM menyediakan strategi yang memusatkan pada definisi proses bisnis dan integrasi di dalam dan diantara enterprise, daripada pengembangan struktur aplikasi individual yang *tightly coupled*. Integrasi proses bisnis melibatkan integrasi beberapa aplikasi dengan memanfaatkan berbagai metadata, platform dan proses.

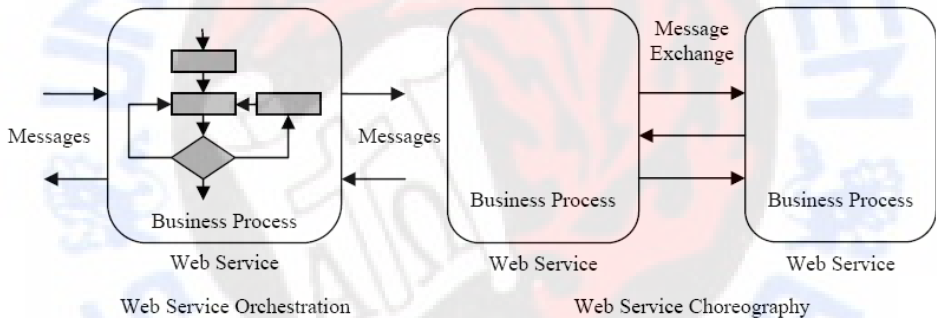


**Gambar 1.10 Pendekatan Integrasi Berbasis Proses (McGovern dkk, 2003)**

Web Services merupakan teknologi yang diterima untuk penafsiran proses bisnis. Sistem BPM menggunakan Web Services sebagai teknologi untuk deskripsi proses bisnis dan menyediakan

pendekatan baru yang mengganti sistem workflow tradisional yang menerapkan bahasa definisi proses yang berhak milik.

**Orchestration** dan **Choreography** dalam Web services menunjukkan kombinasi Web Service dalam urutan *consequencial* untuk membuat proses bisnis yang dapat dieksekusi. Perbedaan antara orchestration dan choreography adalah orchestration mengacu ke proses bisnis yang dapat dieksekusi yang dikendalikan oleh satu pihak bisnis yang terlibat, sementara itu, choreography mendeskripsikan kerja kolaboratif untuk tiap pihak yang terlibat di dalam interaksi keseluruhan dari Web Service.

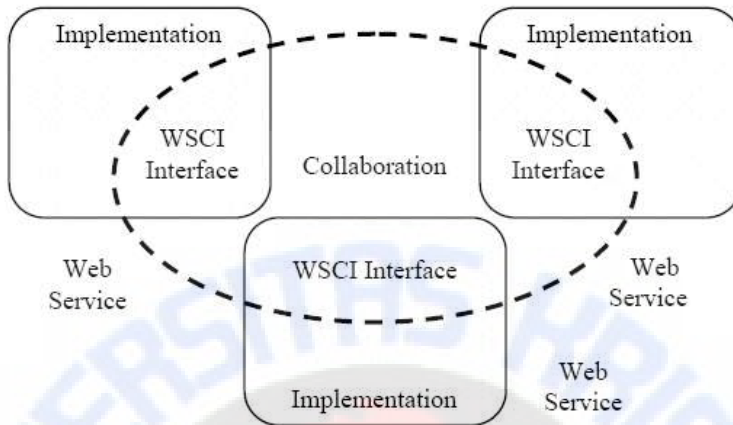


Gambar 1.11 Orchestration dan Choreography pada Web Service (McGovern dkk, 2003)

Bahasa definisi proses digunakan untuk mendeskripsikan rangkaian urutan yang mana Web Service di-invoke untuk menyelesaikan fungsi bisnis dan bagaimana operasi Web Service dikorelasikan untuk membentuk percakapan (*conversation*) bisnis karena operasi pada Web Service bersifat *stateless*. Walaupun

berbagai bahasa definisi proses diajukan dari organisasi dan vendor, masih belum ada standard yang secara universal diterima sebagai bahasa untuk deskripsi proses bisnis. Setiap bahasa ini mempunyai perbedaan kekuatan dan kelemahan dalam mengekspresikan proses bisnis.

Satu bahasa yang digunakan untuk definisi proses adalah **Web Service Choreography Interface (WSCI)**, (W3C, 2002) yang merupakan bahasa berbasis XML yang diajukan oleh Intalio, Sun Microsystems, SAP dan BEA Systems. Bahasa yang mendeskripsikan aliran pertukaran pesan diantara Web Service yang berinteraksi dengan menyediakan definisi proses yang global dan berorientasi pesan. WSCI merupakan bahasa choreography yang berarti WSCI mendeskripsikan perilaku yang dapat diobservasi antara Web Service tanpa berurusan dengan definisi dari proses bisnis yang dieksekusi dan property transaksional. WSCI mendeskripsikan ketergantungan diantara operasi Web Service sehingga beberapa client dapat memahami bagaimana berinteraksi dengan layanan dalam konteks proses yang diberikan dan dapat mengantisipasi perilaku yang diharapkan dari layanan semacam itu pada beberapa point dalam siklus hidup proses. WSCI mendeskripsikan rincian perilaku Web Service di dalam proses yang eksekusinya dapat diawali melalui pesan yang diterima. Interface WSCI tunggal mendeskripsikan pertukaran pesan dari sudut pandang Web Service. Gambar 2.12 mendeskripsikan relasi antara interface WSCI dan Web Services.



**Gambar 1.12 Interface WSCI dan Web Service (McGovern dkk, 2003)**

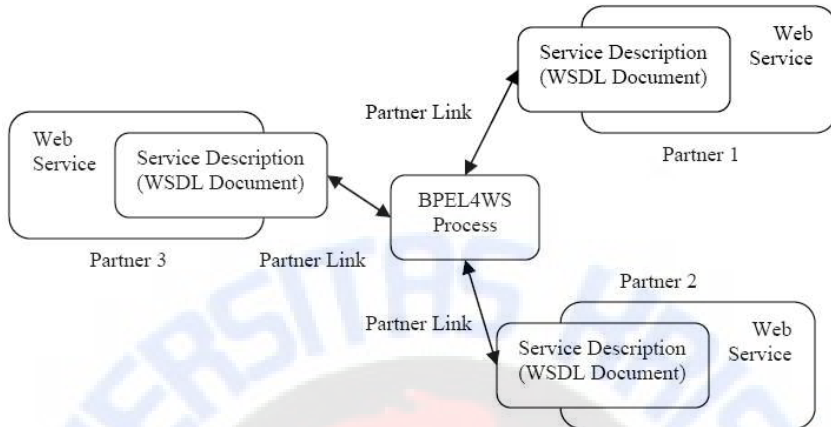
Bahasa definisi proses yang paling umum adalah **Business Process Execution Language for Web Services (BPEL4WS)**, (OASIS, 2003) yang merupakan sebuah spesifikasi yang ditulis bersama oleh IBM, BEA, Microsoft, SAP, dan Siebel. BPEL4WS merupakan bahasa penyatuan yang featuranya dari IBM's Web Service Flow Language (WSFL) dan Microsoft's XLANG. BPEL4WS memanfaatkan *grammar* berbasis XML untuk menciptakan definisi proses dan terletak pada lapisan atas WSDL dalam mendeskripsikan komponen Web Service tertentu untuk pendefinisian pesan yang dipertukarkan, operasi yang dieksekusi dan port type yang diperlukan.

Bahasa yang digunakan untuk mendukung dua penggunaan scenario terpisah adalah :

- **Abstract process** adalah untuk definisi aturan protocol bisnis dan identifikasi perilaku pertukaran pesan antara pihak-pihak yang terlibat dalam protocol dengan penyembunyian perilaku internal mereka.
- **Executable process** mengidentifikasi perilaku actual dari partisipan dalam interaksi bisnis dengan mendefinisikan urutan eksekusi Web Service antar setiap partner bisnis. Ia mendefinisikan berapa banyak interaksi layanan dengan partner yang dikoordinasi dan memperkenalkan mekanisme sistematis untuk urusan dengan eksepsi bisnis dan kegagalan pemrosesan.

Definisi proses BPEL4WS berisi elemen-elemen yang mendeskripsikan aliran kendali (*control flow*), *asynchronous interactions*, *correlation*, *faults*, *compensation* dan komponen lain di dalam proses bisnis. Definisi proses mendefinisikan proses dalam acuan interaksi dengan partner. Partner akan menyediakan layanan ke proses, pemanggilan layanan dari proses atau berkontribusi dalam interaksi dua arah dengan proses. *Partner links* mengidentifikasi bentuk relasi dengan partner dengan mendefinisikan pesan dan port types yang digunakan dalam interaksi dalam kedua arah. Gambar 2.13. mengidentifikasi relasi antara proses BPEL4WS dan partnernya.





**Gambar 1.13 Proses dan Partner BPEL4WS (McGovern dkk, 2003)**

Masih ada berbagai spesifikasi yang ada seperti *WS Choreography Model*, atau bahasa definisi proses, seperti *Business Process Modeling Language (BPML)*, yang mendukung dan memperluas teknologi Web Service agar memungkinkan penggabungan layanan untuk mempunyai sifat *state-full* dan interaksi *long-running* antara partner bisnis dalam enterprise. System BPM dengan Web Service menyediakan tool dan strategi *well-defined* untuk passing informasi antara system yang berpartisipasi, pemodelan dan visualisasi proses bisnis dalam aplikasi dan level enterprise, dan otomasi diantara mereka.

Integrasi proses lebih canggih daripada pendekatan lain dalam arti bahwa instance proses bisnis menutupi banyak solusi integrasi aplikasi tradisional dan memperluas batas dari komunikasi dua atau lebih aplikasi ke integrasi level enterprise dengan menyediakan sebuah

*process model*. *Process model* menyediakan sebuah abstraksi untuk aturan bisnis dan nilai bisnis dan menentukan bagaimana setiap system berinteraksi dengan lainnya agar mencapai sasaran bisnis.

Integrasi berbasis proses menyediakan pendekatan integrasi yang memuaskan untuk bisnis dengan mempertimbangkan kebutuhan enterprise dan memungkinkan integrasi secara real-time. Ini merupakan pendekatan berpusat bisnis yang memberikan bisnis secara individual kemungkinan untuk mengkonfigurasi model proses tanpa memerlukan perubahan permintaan dalam model implementasi.

