

Aplikasi Enterprise

Pertemuan 1
Wiratmoko Yuwono

Deskripsi Matakuliah

- Matakuliah ini akan menjelaskan arsitektur, konsep, dan metode-metode yang ada dalam perancangan dan implementasi IT pada perusahaan skala enterprise, beserta implementasi pemrogramannya.
 - Matakuliah ini akan membantu memberi pemahaman khusus tentang peran **service oriented architectures** (SOA) dan perkembangannya dalam analisa kebutuhan, perancangan dan penerapannya pada skala enterprise.
 - Selain itu matakuliah ini juga membahas arsitektur/teknologi yang dapat langsung diterapkan dalam platform yang berbeda-beda, mendukung teknologi berbasis komponen, berbasis service dan sistem terdistribusi (menggunakan Webservice pada Java dan .NET)
-

Kompetensi Matakuliah

Setelah menyelesaikan matakuliah ini, diharapkan mahasiswa akan:

- Dapat menjelaskan arsitektur *enterprise*,
 - Mengerti dan mengenal kapan dan dimana arsitektur *enterprise* dapat diterapkan,
 - Mampu menjelaskan Service Oriented Architecture
 - Mengembangkan suatu arsitektur berbasis layanan (service) dengan teknologi *web service*,
 - Mempelajari pemrograman arsitektur enterprise dengan teknologi J2EE dan .NET
-

Referensi

- *Web Service and SOA Technologies* by Glenn Hostetler, *Practicing Safe Tech*, 2009
- *Web Services: Concepts, Architectures and Applications*, by Gustavo Alonso, Fabio Casati, et. al, Springer, 2004
- *Service Oriented for Dummies*, by Judith Hurwitz, Robin Bloor, Carol Baroudi, and Marcia Kaufman, Willey, 2007
- *Service-Oriented Architecture: Concepts, Technology, and Design* by Thomas Erl, Prentice Hall PTR, 2005
- *Web Services* by Gustavo Alonso, Fabio Casati, Harumi Kuno, Vijay Machiraju, Springer, 2003
- *Enterprise SOA: Service-Oriented Architecture Best Practices* by Dirk Krafzig, Karl Banke, Dirk Slama, Prentice Hall PTR, 2004
- *Enterprise Integration: An Architecture for Enterprise Application and Systems Integration* by Cummins, Fred A, John Wiley & Sons, 2002
- *Real World XML* by Steven Holzner, New Riders Publishing, 2003
- *Understanding Enterprise SOA* by Eric Pulier, Hugh Taylor, Manning Publications Co, 2006
- *Patterns of Enterprise Application Architecture*, by Martin Fowler et. al, Addison Wesley, 2002

MATERI I AASE

Pengantar Arsitektur Enterprise

- Menurut kamus, “enterprise” berarti:
 - Keberanian berusaha, kegiatan memulai usaha
 - Perusahaan, firma.
 - Menurut *Developing Enterprise Java Applications with J2EE and UML* by Khawar Zaman Ahmed, Cary E. Umrysh, istilah enterprise mengacu pada suatu organisasi atau individu sebagai suatu **kesatuan**, yang bekerja **bersama-sama** untuk mencapai beberapa tujuan **umum**.
 - Enterprise berkaitan erat dengan **B2B** (Business to Business) dan **B2C** (Business to Customer).
 - Kata enterprise biasa digunakan untuk menyebut perusahaan dalam skala besar, seperti Microsoft, Intel, Google, Yahoo!, atau Ebay.
-

8 Elemen Lingkungan Perusahaan

Masyarakat



8 Elemen Lingkungan Perusahaan

Tanggung jawab utama dari area Bisnis

Area bisnis Elmn-ling	<i>Keuangan</i>	<i>SDM</i>	<i>Jasa Inf</i>	<i>Manufaktur</i>	<i>Pemasaran</i>
Pelanggan	X				X
Pemasok			X	X	
Pemegang saham dan pemilik	X				
Serikat Buruh		X		X	
Pemerintah	X	X	X	X	X
Masy.Keuangan	X				
Masy.Global		X			
Pesaing					X

Enterprise era in the Digital World

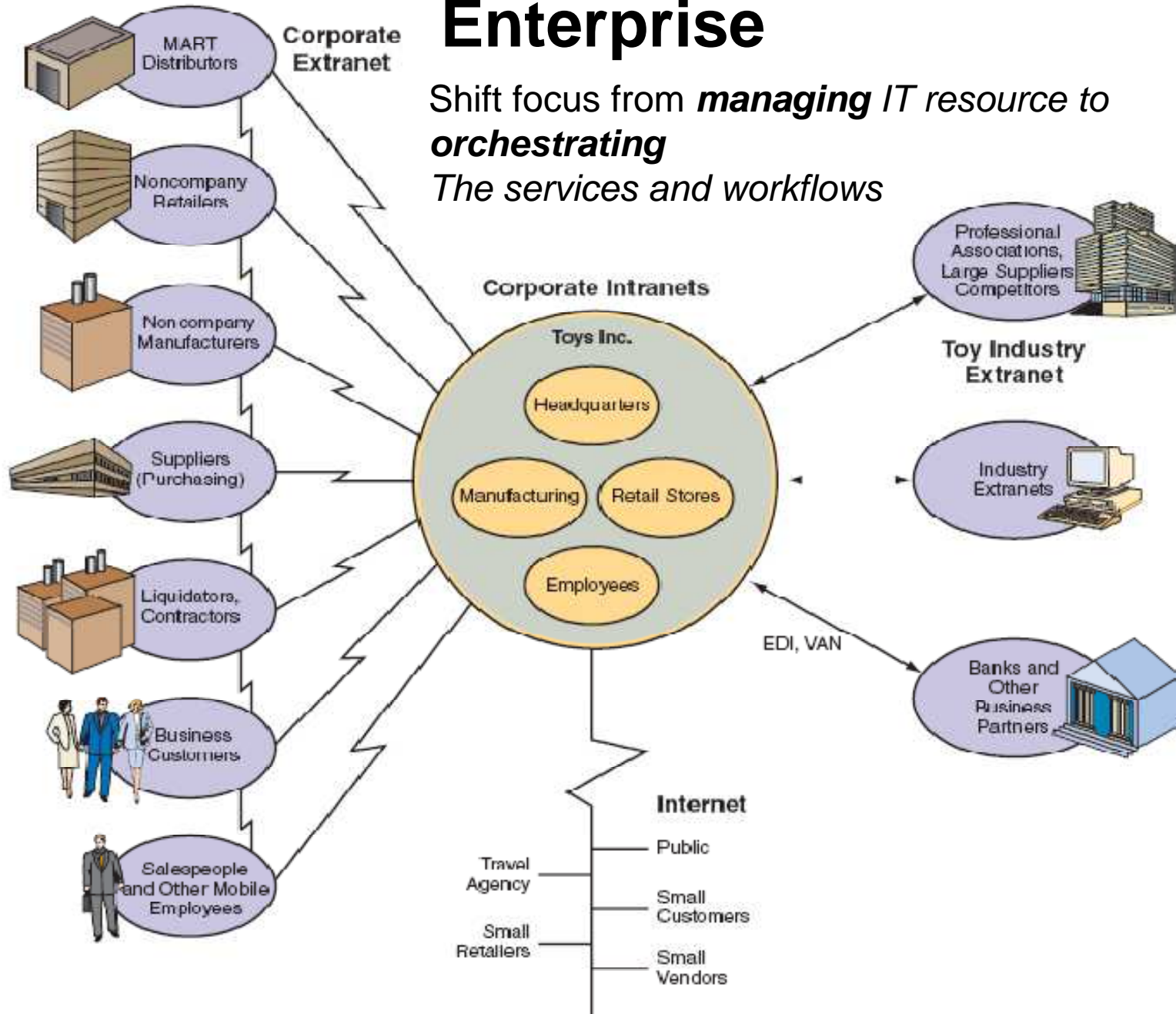
- Globalization
 - The digitization of goods
 - Speeds
 - The merger of products and services
 - New markets and pricing
 - The Net Generation:
 - Digital
 - Connected
 - Social
-

Enterprise's Need

- Information sharing and processing - IT
 - Asset management and tracking - PLM
 - Resource planning - ERP
 - Business process automation - BI
 - Customer or client management - CRM
 - Boost employee productivity - HRM
 - Protection of business knowledge - Security
-

Enterprise

Shift focus from *managing IT resource* to *orchestrating The services and workflows*



What is architecture? - IEEE

- “The **organizational structure** of a system or component.”
 - IEEE Standard 610, “Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries”
 - “The fundamental **organization** of a system embodied in its **components**, their **relationships** to each other, and to the environment, and the principles guiding its **design** and **evolution**.”
 - IEEE Standard 1471, “IEEE Recommended Practice for Architectural Description of Software-Intensive Systems.
-

Arsitektur

- Suatu cara dimana **komponen-komponen** dalam sebuah sistem diorganisasikan dan diintegrasikan
 - Dari definisi itu **komponen-komponen** pembentuk sistem sangat **penting** untuk kesuksesan sebuah arsitektur.
 - **Pemahaman** terhadap komponen-komponen penyusun arsitektur tersebut diperlukan agar kita bisa memahami arsitektur, karakteristik sistem dan keterkaitannya dengan sistem lainnya.
 - Kita perlu **mengintegrasikan** sistem dalam suatu perusahaan sehingga terjadi komunikasi antar elemen-elemennya.
-

Arsitektur (2)

- Sangatlah penting untuk mengerti sistem dasar suatu organisasi dan mencatat bagaimana komponen-komponen dalam sistem tersebut bekerja **tetapi tidak harus mengetahui detail bagaimana fungsi-fungsi** komponen-komponen tersebut.
-

Enterprise Architectures includes...

- Business Architecture
- Information Architecture
- Technology Architecture



Business architecture

- Business architecture
 - How the business **works**
 - **Business process**
 - What is the business **targets**
 - What are the vision and mission.
 - What is the constraints and demands from IT
 - How IT could be implemented
-

Information architecture

- How information is being managed?
 - Where to store data?
 - It's about data schemas
 - How to collaborate data
 - Mapping of unused / missing information
 - Information availability
 - Information security
-

Technology architecture

- Needed **infrastructure** to support the systems.
 - Collection of selected technologies and tools to implement chosen architectures
 - Development
 - Communication
 - Backups
 - Standards
-

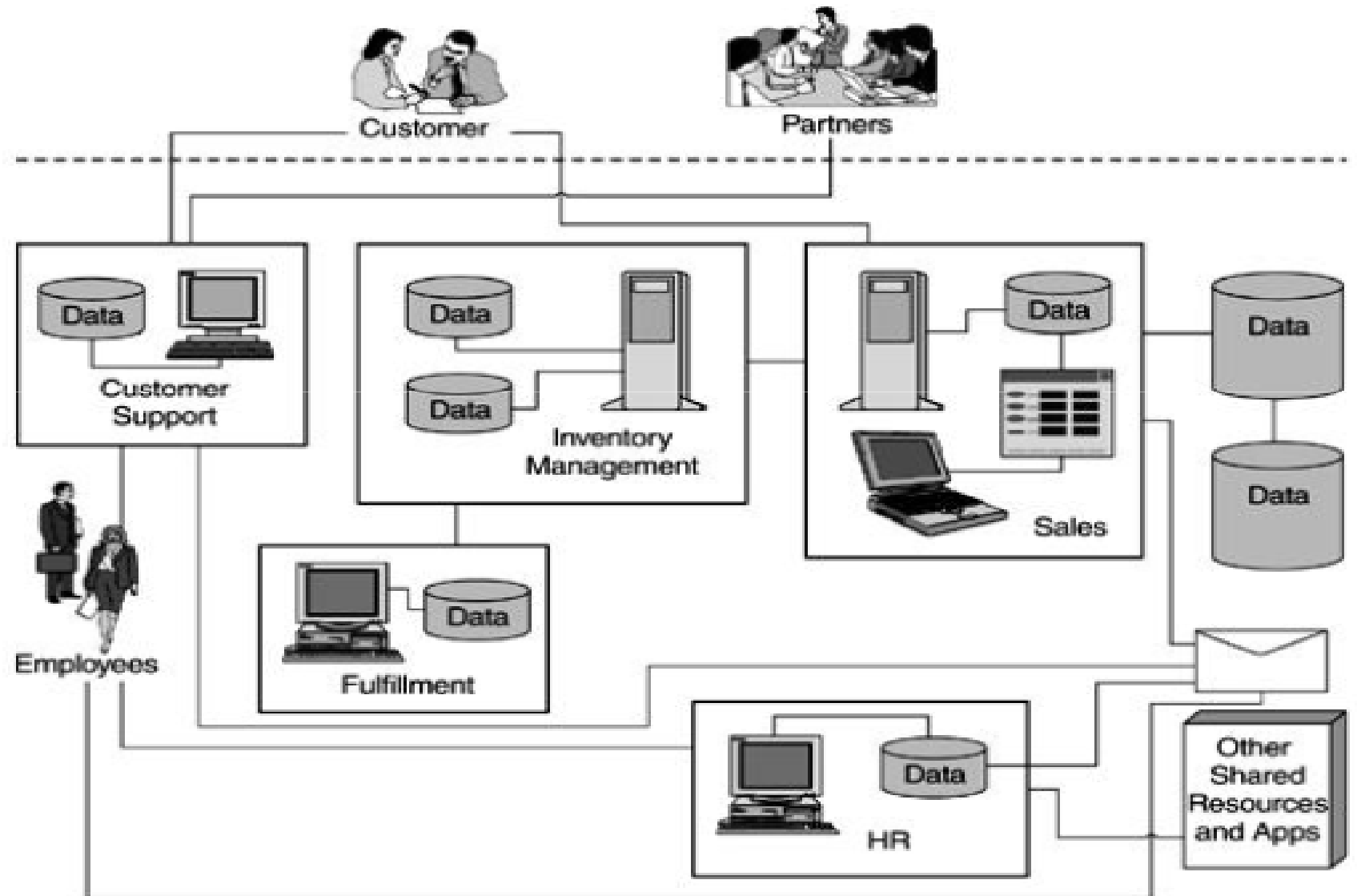
Enterprise Software

- Enterprise Software is software that solves an **enterprise problem (rather than a departmental problem)** and usually it is written using *Enterprise Software Architecture*.
-

Enterprise Software Architecture

- The organizing logic for a firm's core **business processes** and IT capabilities captured in a **set of principles, policies** and **technical choices** to achieve the **business standardization** and **integration requirements** of the firm's operating model.
 - Concerned with **cross** project/solution architecture and communication between **different** practices in architecture.
-

Enterprise Software Architecture



Kebutuhan pembangunan

Enterprise Software pada Perusahaan

- Meng**integrasikan** customer support dan product knowledge melalui WEB/Internet.
 - Dengan menghubungkan semua mesin-mesin server dan data melalui internet secara online, marketing perusahaan itu akan bertambah baik
 - Perusahaan akan menghemat biaya sales manajemen dan mempermudahnya, disamping itu dapat meraih pasar baru.
 - Enterprise software dapat membantu pekerjaan para pekerja di perusahaan sehingga mempermudah, mempercepat pekerjaan, mengefisiensikan pekerja, sekaligus memperkecil biaya pengeluaran perusahaan.
-

Enterprise Applications

- **Contoh Enterprise applications:** payroll (penggajian), patient records (daftar pasien rumah sakit), shipping tracking (data pengiriman barang), cost analysis (analisis keuangan), credit scoring (analisis penerimaan kredit), insurance (asuransi), supply chain (pemenuhan kebutuhan), accounting (akuntansi), customer service, and foreign exchange trading (valas).
 - **Enterprise applications bukan:** automobile fuel injection (pengisian bahan bakar otomatis), word processors, elevator controllers, telephone switches, operating systems, compilers, and games.
-

Karakteristik Enterprise Application

- Butuh **persistent data**, karena data digunakan secara bersama oleh banyak aplikasi, bahkan digunakan untuk jangka waktu yang lama.
 - Walaupun sangat mungkin terjadi perubahan sistem perusahaan, data **tidak boleh** berubah.
 - Enterprise application biasanya **menghandle data yang sangat besar**.
 - Dulu digunakan konsep file system (**standalone**) sekarang digunakan database relasional bahkan basis data berorientasi obyek yang **terdistribusi**
 - Pengguna enterprise application **banyak dan beraneka ragam**
 - Bagaimana cara menghandle concurrent access -> transaction management tool
 - Butuh log dan recovery
-

Karakteristik Enterprise Application

- Memiliki **banyak macam user interface** di masing-masing client
 - Bagaimana agar seluruh data yang ada dapat direpresentasikan ke seluruh user dengan semudah mungkin dengan bermacam-macam user interface tergantung kebutuhannya.
 - Butuh **terintegrasi** dengan aplikasi lain.
 - Mampu **memisahkan** business logic, data, dan presentasi
-

Kesulitan Pengembangan Enterprise Application

- Hardware/Software yang masih mahal, yang mungkin juga tidak (belum) berkembang sesuai keadaan dan kebutuhan sekarang.
 - Kesulitan mencari pekerja yang dapat memiliki kemampuan kerja yang baik, dan dapat mengikuti perkembangan teknologi.
 - Kesulitan dalam pembuatan software yang mampu mengintegrasikan seluruh sistem dan bersifat “**Distributed System Services**”.
-

Kesulitan Pengembangan Enterprise Application

- Kesulitan **perbedaan platform**.
- Adanya **multiple vendor** juga mempersulit pembuatan program.

Contohnya:

- Perusahaan A menggunakan SQL Server dan klien perusahaan A menggunakan Oracle.
 - Dalam hal ini dibutuhkan suatu software yang mampu mengatasi “lintas vendor”.
-

Kesulitan Enterprise Application

- Masih membutuhkan adanya **brainware**
 - 24x7 operation requires 24x7 staffing
 - Adanya kebutuhan **keamanan** sistem dan **integritas** data.
Misalnya:
 - Mampu handle “**system failure**” dengan “**system failure recovery**”
 - **Rollback** transaction untuk transaksi yang salah atau batal
 - **Transaction locking** yang mampu mengatasi keamanan data. Contohnya pada saat ada transaksi yang hampir bersamaan.
 - Mampu handle “**multi user situation**”
 - Menjaga **kekonsistensian** data walau ada error, delay, dan transaksi yang jauh.
-

Pengukuran Kesuksesan Enterprise Application

- **Response time:** adalah total waktu yang dibutuhkan sistem untuk memproses sebuah request dari luar ke sistem tersebut.

Mungkin sebuah aksi terhadap UI, atau sebuah pemanggilan API.

- **Min Response time**

- **Responsiveness:** adalah seberapa cepat sistem mengenali sebuah request sebagai sesuatu untuk diproses dan kemudian mengembalikan feedback walaupun proses belum selesai.

- User bisa frustrasi walaupun response time baik

- Walaupun belum selesai proses, sistem harus tetap memberikan respon.

- Misalnya gunakan timer atau progress bar, atau informasi lain.
-

Pengukuran Kesuksesan Enterprise Application

- **Latency**: adalah waktu minimum yang dibutuhkan untuk mendapatkan segala bentuk response.
 - **Throughput**: adalah berapa hasil yang diperoleh dalam suatu satuan waktu tertentu.
 - Jika kita mengukur copy file, throughput diukur dalam berapa **bytes per second**.
 - Untuk enterprise applications pengukuran berdasarkan **transactions per second** (tps), tapi masalahnya bergantung pada kompleksitas transaksi.
 - **Capacity** adalah indikasi seberapa maximum dari throughput
-

Pengukuran Kesuksesan Enterprise Application

- **Load** adalah tingkat tekanan / **stress** sebuah sistem
 - Misalnya diukur dengan berapa banyak user yang sedang terhubung saat itu
 - Misalnya response time untuk suatu request pada saat peak adalah 0.5 seconds dengan 10 users dan 2 seconds dengan 20 users.
 - **Load sensitivity** adalah bagaimana response time bervariasi dengan load.
 - Misalnya sistem A memiliki response time 0.5 seconds untuk 10 - 20 users dan sistem B memiliki response time 0.2 seconds untuk 10 users yang naik menjadi 2 seconds untuk 20 users.
 - Pada contoh di atas, sistem A memiliki load sensitivity yang lebih rendah daripada sistem B.
-

Pengukuran Kesuksesan Enterprise Application

- **Efficiency** adalah performa dibagi dengan resources.
 - Sebuah sistem yang memiliki 30 tps pada 2 CPU akan lebih efisien dibanding dengan sebuah sistem yang memiliki 40 tps pada 1 CPU.
 - **Scalability** adalah ukuran bagaimana penambahan resources (biasanya hardware) mempengaruhi performance.
 - **Vertical scalability** atau scaling up, berarti menambahkan lebih banyak tenaga terhadap single server, seperti penambahan memory.
 - **Horizontal scalability** atau scaling out, berarti menambahkan lebih banyak server.
-