

Sistem Terdistribusi [TIF304]  
Ringkasan Materi Bab I dan Bab II  
Characterization of Distributed Systems dan System Models



disusun oleh:

Dendy Setyowibowo (33123)

Fahim Jatmiko (33277)

Primanda Fauzi Pamungkas (33374)

Jurusan Teknik Elektro dan Teknologi Informasi  
Fakultas Teknik  
Universitas Gadjah Mada  
Jogjakarta  
2009

# Rangkuman Materi Sistem Terdistribusi Bab I dan Bab II

Dendy Setyowiwo, 33123-TE  
Fahim Jatmiko, 33277-TE  
Primanda Fauzi Pamungkas, 33374-TE  
Jurusan Teknik Elektro FT UGM,  
Yogyakarta

## Bab I

### Karakteristik Sistem Terdistribusi

#### A. Pendahuluan

Sistem terdistribusi adalah sebuah sistem yang memiliki komponen-komponen pendukung dan tergabung dalam sebuah jaringan komputer. Komputer-komputer tersebut berkomunikasi satu sama lain dengan jalan pengiriman pesan. Sistem terdistribusi tersebut meliputi internet, intranet, dan mobile computing.

Komputer-komputer dalam sebuah jaringan dapat terletak dalam lokasi yang berjauhan. Hubungan antara komputer-komputer tersebut dapat dijalin dengan komunikasi nirkabel. Sebuah jaringan pun juga dapat terdiri atas komputer-komputer dalam sebuah ruangan yang tentunya dapat berkomunikasi dengan lebih baik daripada yang berjauhan. Sistem terdistribusi tersebut memiliki konsekuensi berikut.

#### *Konkurensi*

Dalam jaringan komputer, proses eksekusi program harus berjalan secara konkuren. Setiap komputer dapat melakukan pekerjaannya masing-masing tanpa terjadi konflik di antaranya. Antarkomputer juga dapat melakukan *file sharing* dengan sistem ini. Sistem dapat melakukan hal ini dengan lebih baik lagi apabila sistem itu menambah sumber lagi, seperti penambahan komputer dalam jaringan. Penambahan komputer ini dapat meningkatkan kapasitas sistem dalam *file sharing*. Maka, diperlukan koordinasi konkurensi program yang mengeksekusi program *sharing* tersebut dalam sistem ini.

#### *Ti adanya clock global*

Dalam sebuah sistem terdistribusi, komputer memiliki clock masing-masing. Maka, diperlukan sebuah sistem koordinasi agar tidak terjadi konflik clock antarkomputer tersebut. Saat program membutuhkan koordinasi antarkomputer, dilakukan proses pertukaran pesan antarkomputer. Koordinasi antarkomputer yang dibutuhkan itu, tergantung pada waktu di saat program itu dijalankan.

#### *Kegagalan Independen*

Semua sistem komputer dapat mengalami kegagalan dan hal tersebut merupakan tanggung jawab pembuat sistem untuk memecahkannya. Begitu juga dengan sistem terdistribusi

yang dapat mengalami kegagalan dengan berbagai cara. Kesalahan dalam jaringan bisa saja membuat komputer terisolasi, meski hal tersebut tidak berarti bahwa komputer itu tidak dapat beroperasi. Kesalahan yang terjadi dalam sistem ini dapat menyebabkan sistem berjalan lebih lambat. Kesalahan dalam sistem itu pun bisa jadi kesalahan independen setiap komputer atau bahkan hanya sebagian kecil komponen komputer.

## B. Contoh-Contoh Sistem Terdistribusi

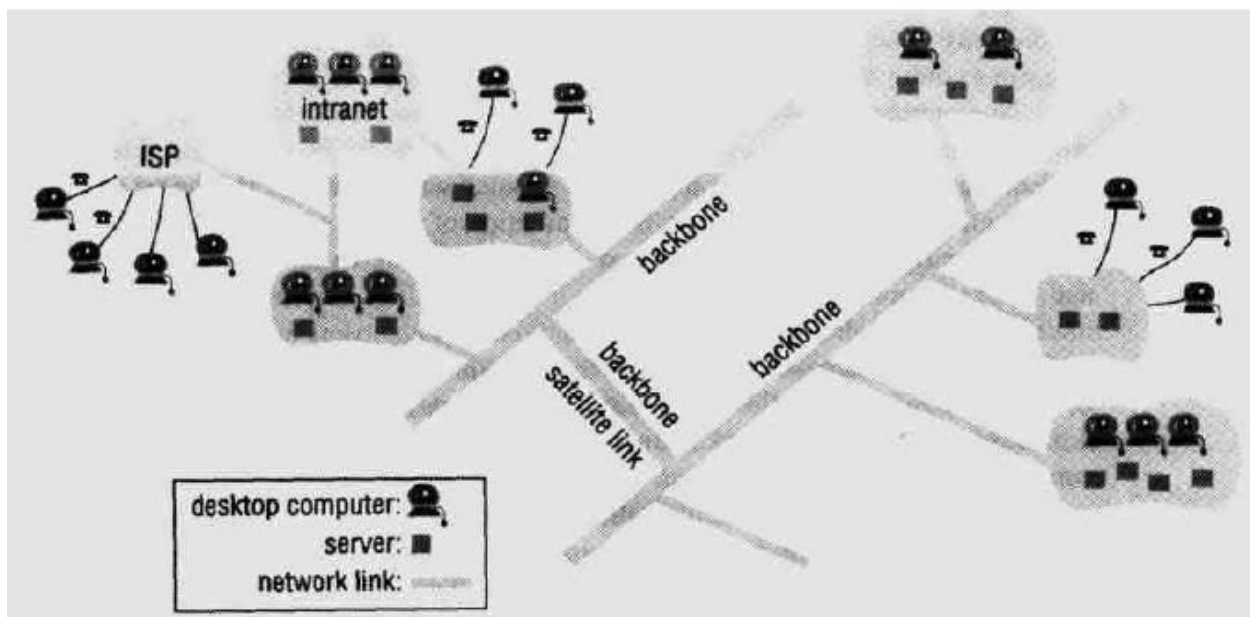
Sistem terdistribusi yang dibahas meliputi internet, intranet, dan mobile computing. Ketiganya merupakan sistem yang sangat terkenal di dunia saat ini. Dengan ketiganya, masyarakat mampu memanfaatkan banyak fasilitas dan aplikasi di berbagai tempat.

### B.1. Internet

Internet adalah sekumpulan komputer yang terhubung secara luas dan dalam bentuk yang bermacam-macam. Internet juga merupakan sebuah sistem terdistribusi yang ukurannya sangat besar karena dapat mencakup semua komputer di dunia. Internet dapat diakses dengan memanfaatkan fasilitas world wide web atau www, file transfer, dan email.

Layanan yang dapat dimanfaatkan pengguna tersebut dapat ditambah dengan fasilitas lain oleh subjaringan yang ukurannya jelas lebih kecil. Fasilitas tambahan tersebut biasanya digunakan untuk meningkatkan keamanan dan dilakukan oleh sebuah server lokal. Untuk mengakses internet, perlu adanya pihak yang menyediakan layanan tersebut. Pihak ini disebut Internet Service Provider atau ISP. Dalam internet, dapat dijumpai banyak fasilitas multimedia, seperti audio, video, dan data-data lainnya. Keberadaan internet pun kini telah membuat peradaban masyarakat dunia menjadi lebih maju.

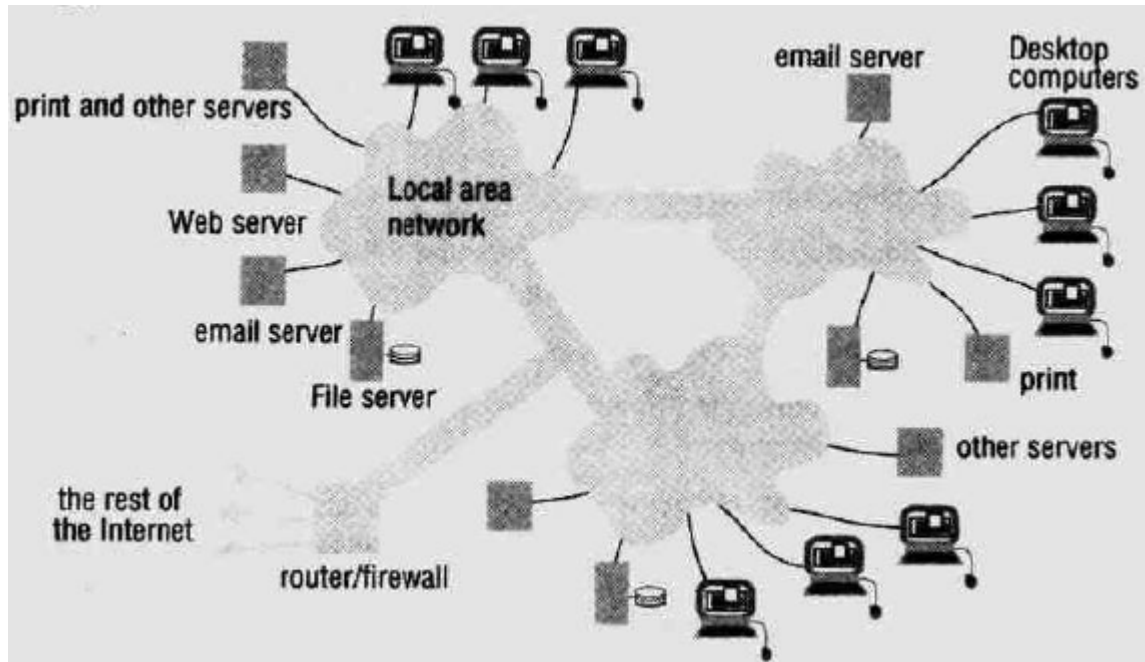
Berikut adalah gambaran umum dari internet.



Gambar1.1 Internet

## B.2. Intranet

Intranet adalah bagian kecil dari internet yang memiliki sistem administrasi sendiri dan dikelola oleh pihak tertentu. Intranet biasanya memiliki kebijakan keamanan sendiri. Berikut adalah contoh sederhana dari intranet.



Gambar1.2 Intranet

Intranet yang ada pada gambar tersebut terdiri atas beberapa jaringan Local Area Network (LAN) yang dihubungkan dengan jaringan backbone. Konfigurasi jaringan intranet menjadi tanggung jawab pihak yang membuatnya dan bukan lagi menjadi tanggung jawab penyedia internet.

Sebuah intranet dapat terhubung dengan jaringan internet melalui sebuah router. Router tersebut dapat membuat pengguna intranet memasuki dunia yang lebih luas, yakni internet. Pengguna di luar intranet pun bisa juga mengakses fasilitas intranet tersebut, tetapi tetap harus mematuhi aturan yang diberlakukan jaringan intranet tersebut. Hal ini dilakukan untuk menghindari masalah keamanan. Salah satu cara yang paling populer untuk itu adalah firewall. Firewall adalah cara untuk mengamankan sebuah intranet dengan tidak mengizinkan pesan yang tidak sah untuk masuk atau keluar. Firewall beroperasi dengan melakukan penyaringan pesan masuk dan keluar.

## B.3. Mobile dan Ubiquitous Computing

Dewasa ini, teknologi menuju pada bentuk yang lebih kecil supaya mudah dibawa ke mana-mana. Teknologi komputer juga mengalami hal demikian. Bentuk komputer yang portabel dapat dijumpai pada laptop, PDA, dan lain-lain. Dengan demikian, sistem terdistribusi dapat juga memiliki bentuk yang portabel. Pengguna pun dapat memanfaatkan layanan sistem ini di mana pun dia berada.

Secara umum, mobile dan ubiquitous computing berarti melakukan komputasi, dalam hal ini dengan komputer, di mana-mana. Dalam kaitannya dengan sistem terdistribusi, piranti-piranti portabel yang dapat digunakan tersebut dapat saling berkomunikasi, sehingga dapat tercipta sebuah sistem terdistribusi yang portabel, meski infrastruktur dasarnya tidak portabel.

Pemanfaatan *mobile computing* ini memang sangat berguna bagi kemajuan peradaban manusia. Kini, satu orang saja sudah bisa memanfaatkan teknologi jaringan internet dengan berbagai cara. Dengan laptop, seorang bisa terhubung ke jaringan itu melalui jaringan LAN. Dengan telepon selulernya, seorang bisa terhubung ke jaringan itu melalui jaringan Wireless Access Protocol (WAP). Sementara itu, dengan kamera digitalnya, seorang bisa melakukan transfer data dengan infrared. Dengan adanya *mobile computing* ini, seseorang dapat melakukan beberapa pekerjaan sekaligus dengan cepat.

### **C. Resource Sharing dan Web**

Kemampuan untuk berbagi sumber atau resource sharing menjadi motivasi utama adanya sistem terdistribusi. Satu buah komputer mungkin memiliki data yang tidak dimiliki komputer lainnya, sehingga perlu adanya fasilitas untuk berbagi sumber. Sumber yang dimaksud tidak hanya data, tetapi juga bisa berupa hardware, seperti printer. Sumber tersebut dikelola oleh server dengan sistem yang sesuai bidangnya, misalnya web server mengelola halaman web dan sumber web lainnya. Sumber tersebut dapat diakses klien, contohnya browser yang disebut sebagai klien dari web server. Browser sendiri masih dijalankan oleh pengguna (end user).

Dunia internet tentunya tidak dapat dipisahkan dari world wide web atau www. Hal tersebut merupakan sebuah sistem untuk menampilkan dan mengakses sumber yang ada di internet. Sistem www biasanya berada di awal alamat web tujuan.

Web adalah sebuah sistem yang terbuka dan bisa saja diperluas implementasinya tanpa mengganggu sistem yang lainnya. Operasi web berdasarkan standar komunikasi dan dokumen yang telah dipopulerkan dan diimplementasikan secara luas. Selain itu, web dapat bersifat terbuka terhadap berbagai jenis tipe berkas yang dapat dimunculkan dalam web itu sendiri.

Web sendiri terdiri atas tiga standar teknologi yang utama, yakni HyperText Markup Language (HTML), Uniform Resources Locators (URL), dan HyperText Transfer Protocol (HTTP).

#### *HTML*

HTML adalah sebuah bahasa untuk melakukan penetapan isi dan tampilan web yang diinginkan dalam browser. Halaman web sendiri terdiri atas heading, paragraf, tabel, dan gambar

dan itu semua dapat diatur dengan HTML. Pengaturan yang dimaksud meliputi pengaturan tata letak, bentuk, dan ukuran. HTML itu juga dapat digunakan untuk menetapkan *link* ke web lain dan sumber mana saja yang berhubungan dengan web mereka.

### *URL*

URL digunakan untuk mengidentifikasi dokumen dan sumber lainnya yang disimpan dalam web. Browser dapat mengecek URL untuk mengambil sumber dari server. URL bisa diketik dalam browser lalu browser akan mencarinya. Akan tetapi, kini lebih banyak pengguna yang menggunakan URL dengan melakukan klik saja di sebuah *link*.

### *HTTP*

HTTP adalah sebuah protokol atau pengatur yang dapat mendefinisikan jalan untuk browser dan jenis klien lainnya untuk mengakses sumber di server. HTTP sendiri memiliki beberapa fitur berikut.

#### *Request-reply interaction*

Dengan HTTP, klien dapat meminta pesan request ke server dengan URL. Lalu, server mengeceknya, bila tujuannya memang ada, server memberikan pesan reply ke klien. Jika tujuan klien memang tidak ada, server memberikan pesan error sebagai reply.

#### *Content types*

Browser tidak perlu memiliki kemampuan untuk menangani atau membuat penggunaan berbagai macam tipe berkas. Saat browser mengirim pesan request, hal tersebut sudah termasuk pengiriman tipe berkas yang mereka inginkan. Maka, server memberikan balasan sesuai.

#### *One resource per request*

Klien meminta satu buah sumber dalam satu setiap request HTTP. Jika sebuah halaman web terdiri atas tujuh gambar, browser akan memberikan total delapan request. Dengan demikian, setiap gambar tersebut memerlukan satu request HTTP dan satu lagi untuk mengakses sumber halaman web.

#### *Simple access control*

Setiap user dengan koneksi ke web server dapat dengan mudah mengakses semua sumber yang ditampilkan server. Untuk keperluan tertentu yang lebih privat, server biasanya melakukan autentikasi user dengan menggunakan password.

## **D. Tantangan yang Dihadapi dalam Sistem Terdistribusi**

Dalam sebuah sistem, tentunya ditemukan beberapa macam kendala. Hal tersebut juga terjadi dalam sistem terdistribusi. Meskipun dapat dijumpai di mana-mana, sistem ini tetap saja memiliki beberapa tantangan yang harus dihadapi untuk menghasilkan sebuah sistem yang jauh lebih baik dan memuaskan pengguna. Di masa depan, diharapkan sudah terdapat banyak

penyelesaian mengenai tantangan yang ada pada sistem ini. Berikut adalah beberapa contoh tantangan yang dihadapi dalam sistem terdistribusi.

### *Heterogenity*

Sistem terdistribusi tersusun atas beberapa jenis jaringan, yang tentunya memiliki perbedaan di antaranya. Selain itu, sistem ini juga terdiri atas sistem operasi, perangkat keras, dan bahasa pemrograman yang berbeda pula. Dengan internet protokol, diharapkan perbedaan-perbedaan itu dapat diatasi karena menggunakan protokol yang sama.

### *Openess*

Sistem terdistribusi harus bisa diperluas, yakni dengan langkah pertama berupa memperkenalkan antarmuka komponennya. Akan tetapi, kesatuan dari komponen-komponen tersebut menjadi tantangan yang dihadapi.

### *Security*

Untuk mengamankan informasi, dapat dilakukan dengan metode enkripsi. Enkripsi ini dapat membantu menyediakan perlindungan yang cukup terhadap sumber yang dapat diakses pengguna di mana saja. Enkripsi tersebut dapat menjaga data-data rahasia tetap aman saat dikirim melalui jaringan. Salah satu masalah utama yang ada mengenai hal itu adalah *denial of service*.

### *Scalability*

Sebuah sistem terdistribusi dapat diperbesar jika biaya penambahan user tidak terlalu besar atau masih berada dalam jumlah yang stabil. Algoritma untuk mengakses data harus menghindari efek bottleneck. Selain itu, data juga distrukturisasi secara hierarkis agar dapat diakses dalam waktu yang singkat saja.

### *Failure Handling*

Setiap proses dapat mengalami kegagalan. Kegagalan itu bisa terjadi secara bersama atau hanya sebagian komponen dari sebuah sistem. Maka, setiap komponen dalam sistem, dalam hal ini sistem terdistribusi, harus bisa mewaspadaai kemungkinan kegagalan itu dan bisa menyelesaikannya, bila kejadian itu terjadi.

### *Concurrency*

Adanya beberapa user sekaligus dalam sistem terdistribusi dapat menjadi peminta request ke sumber yang disediakan server. Dalam lingkungan yang konkuren, setiap sumber harus didesain menjadi aman untuk diakses.

### *Transparency*

Beberapa sudut dari sistem terdistribusi memang dibuat tidak terlihat agar programmer dapat fokus pada desain aplikasi yang mereka kembangkan. Bahkan, adanya kesalahan dalam

jaringan dan proses pun dapat dipresentasikan terhadap programmer dalam bentuk eksepsi saja. Akan tetapi, hal ini tetap harus dapat diatasi oleh programmer tersebut.



## **Bab II**

### **System Model**

Sebuah arsitektur model dari sebuah sistem terdistribusi yang terhubung dengan penempatan dari bagian-bagiannya dan hubungan di antara bagian-bagian tersebut. Misalnya model client-server dan model proses peer. Model client-server dapat dimodifikasi dengan :

- partisi dari data atau replikasi dalam cooperating server
- penyembunyian dari data dengan proxy server dan client
- penggunaan dari mobile code dan mobile agents
- persyaratan untuk menambah dan menghapus mobile devices dengan cara yang tepat

Model fundamental yang terhubung dengan beberapa deskripsi formal dari properties, dimana cukup umum di semua bentuk model arsitektur. Tidak ada pewaktuan secara umum dalam sebuah sistem terdistribusi, sehingga pewaktuan pada beberapa komputer yang berbeda tidak perlu adanya pemberian pewaktuan yang sama antara satu komputer dengan komputer lain. Semua komunikasi diantara proses-proses diterima dengan beberapa pesan. Komunikasi pesan di atas jaringan komputer dapat dipengaruhi dengan tunda, dapat dari berbagai kegagalan dan mudah diserang oleh beberapa resiko keamanan. Jaringan ini dialamatkan dengan tiga model :

- Model interaksi yang berurusan dengan performance dan kesulitan pengaturan batas waktu dalam sebuah sistem terdistribusi, misalnya untuk pengiriman pesan
- Model kegagalan yang berusaha untuk memberikan spesifikasi tepat dari kesalahan yang dapat ditunjukkan oleh proses-proses dan saluran komunikasi. Ini mendefinisikan keandalan komunikasi dan proses yang benar
- Model keamanan yang membahas kemungkinan ancaman terhadap proses dan jalur komunikasi. Ini memperkenalkan konsep jalur yang aman, aman terhadap ancaman-ancaman.

#### **A. Pendahuluan**

Sistem yang dimaksudkan untuk digunakan dalam lingkungan dunia nyata, harus dirancang untuk berfungsi dengan benar seluas mungkin dalam berbagai keadaan dan dalam menghadapi berbagai kemungkinan kesulitan dan ancaman. Setiap model dimaksudkan untuk memberikan abstrak, disederhanakan tetapi konsisten deskripsi dari sebuah aspek relevant desain sistem terdistribusi. Sebuah model arsitektur mendefinisikan suatu cara berinteraksi antara komponen-komponen dalam suatu sistem, dan memetakan komponen sistem ke dalam jaringan komputer.

Beberapa masalah dalam mendesain sistem terdistribusi :

- Beragam cara penggunaan : bagian komponen dari sistem merupakan subjek untuk melebarkan varian dalam beban kerja, misalnya beberapa halaman web yang diakses beberapa juta kali per hari. Beberapa bagian dari sebuah sistem mungkin akan terputus, atau kurang terhubung untuk waktu tertentu.
- Berbagai sistem lingkungan : sebuah sistem terdistribusi harus mengakomodasikan heterogen perangkat keras, sistem operasi dan jaringan. Jaringan tersebut dapat berbeda luas kinerjanya, jaringan nirkabel bekerja di sebagian kecil dari kecepatan lokal jaringan.

- Masalah internal : pewaktuan tanpa sinkronisasi, pembaruan data yang bertentangan, banyak mode dari kegagalan hardware dan software yang melibatkan komponen individu suatu sistem.
- Ancaman eksternal : serangan terhadap integritas data dan kerahasiaan, misalnya denial of service.

## **B. Architectural Models**

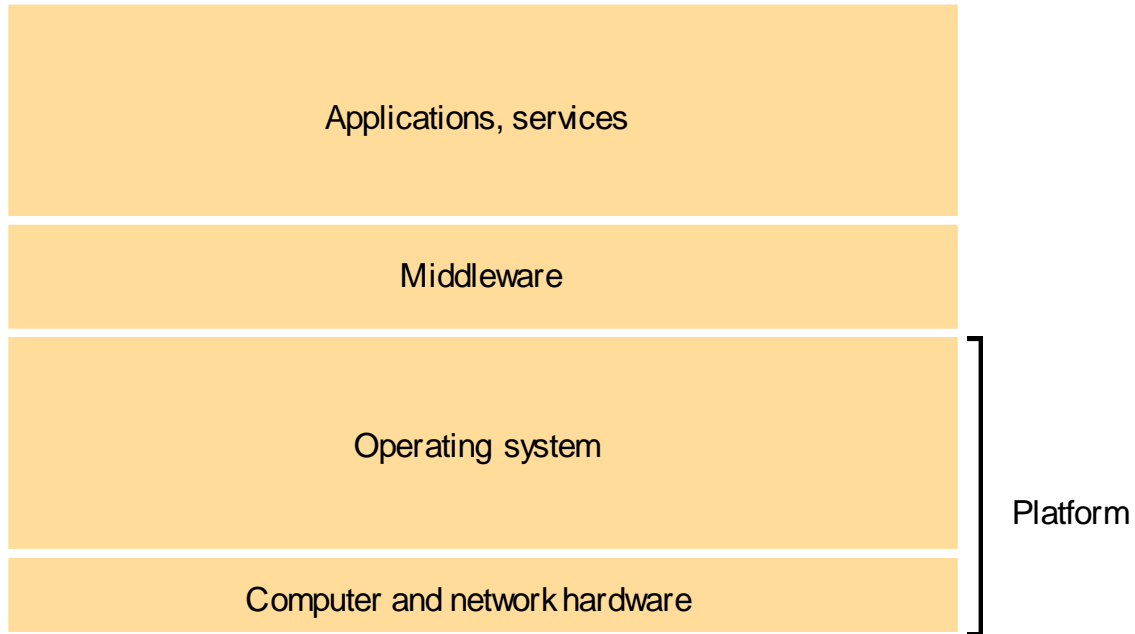
Arsitektur dari sebuah sistem, dimana strukturnya terdiri dari komponen-komponen yang terpisah secara spesifik. Keseluruhan tujuan adalah menyakinkan struktur akan bertemu sekarang dan dimungkinkan untuk tuntutan masa depan. Keprihatinan utama adalah untuk membuat sistem yang dapat diandalkan, dikelola, mudah beradaptasi dan biaya efektif. Sebuah model arsitektur dari sistem terdistribusi menyederhanakan dan mengabstraksikan fungsi dari komponen individual, dengan pertimbangan penempatan komponen melewati sebuah jaringan komputer dan hubungan antarkomponen.

Sebuah inisial penyederhanaan dapat diterima dengan pengklasifikasian proses; seperti proses server, proses client, dan proses peer. Klasifikasi dari proses mengidentifikasi tanggung jawab, dan oleh sebab itu membantu proses untuk mengakses luas kinerja dan menentukan pengaruh dari kegagalan setiap proses. Adapaun beberapa sistem dinamik yang dapat dibangun, dimana merupakan varian dari model client-server :

- Kemungkinan dari pergerakan code dari satu proses ke proses lain, yang mengijinkan sebuah proses untuk mengutus tugas ke proses lain.
- Beberapa sistem terdistribusi didesain untuk mengkatifkan komputer dan perangkat yang lain untuk menambah dan menghapus yang tidak terlihat. Penempatan yang sebenarnya dari proses, membentuk sebuah sistem terdistribusi dalam sebuah jaringan komputer, juga dipengaruhi beberapa performa jaringan, reliability, keamanan dan biaya. Model arsitektur dideskripsikan dapat menyediakan hanya beberapa tampilan penyerdahaan dari pola penting distribusi.

### ***B.1. Software Layers***

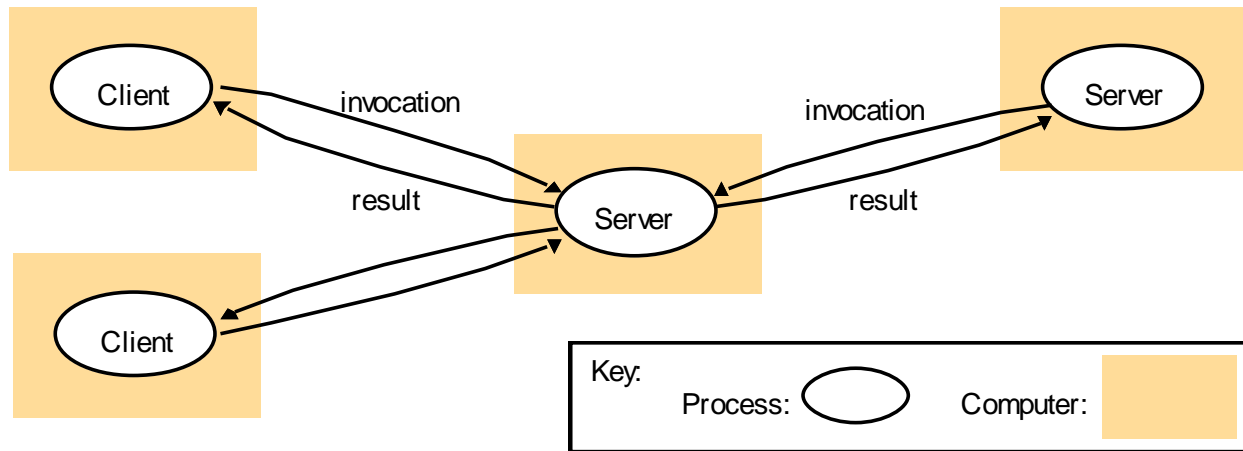
Pola software architecture menghubungkan keaslian penstrukturan dari software seperti layer atau modul dalam sebuah single computer dan baru-baru ini dalam pola pelayanan ditawarkan dan diminta diantara proses-proses yang terlokasi di satu komputer atau berbeda komputer.



Gambar2.1 Layer dalam sistem terdistribusi

Gambar di atas memperkenalkan pola dari “platform” dan ”middleware”. **Platform**, level hardware terendah dan software layer dapat dihubungkan seperti sebuah platform untuk sistem terdistribusi dan aplikasi. Layer dengan level rendah menyediakan pelayanan untuk layer yang berada di atasnya, yang diimplementasikan secara bebas dalam setiap komputer, dan memfasilitasi komunikasi dan koordinasi diantara proses. Misalnya: Intel x86/Windows, Intel x86/Linux, PowerPC/MacOS, Intel x86/Solaris, SunSPARC/SunOS. **Middleware**, sebuah layer software yang bertujuan menutupi sifat heterogen dan menyediakan sebuah model programming yang tepat untuk pengaplikasian programmer. Middleware direpresentasikan dengan proses atau objek dalam satu set komputer, dimana terdapat interaksi di antara setiap implementasi komunikasi dan pendukung untuk aplikasi terdistribusi. Middleware dapat juga menyediakan layanan untuk penggunaan dnegan program aplikasi, berupa servis infrastruktur dan menjamin model program terdistribusi dalam penyediaan middleware.

Beberapa komunikasi dapat disempurnakan dan diimplementasikan hanya dengan pengetahuan dan bantuan dari aplikasi yang berdiri pada perangkat akhir dari sistem komunikasi.

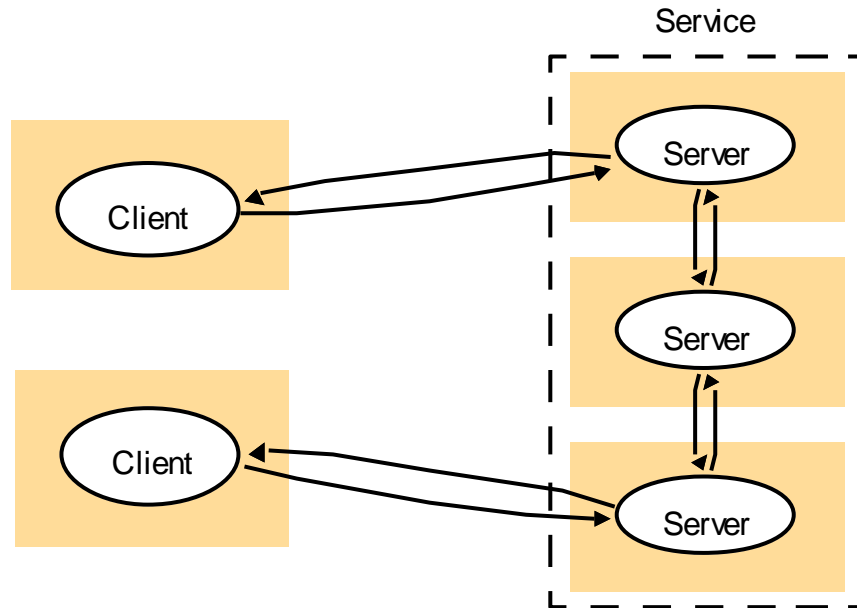


Gambar 2.2 Klien meminta request ke server

Gambar di atas merupakan arsitektur yang sering disebutkan sejak sistem terdistribusi didiskusikan, merupakan ilustrasi struktur yang simple, dimana proses client berinteraksi dengan individual server dalam komputer yang terpisah, pengaksesan resource dapat diatur. Server tidak menutup kemungkinan menjadi client dari server lainnya. Perlakuan yang tepat dalam program terdistribusi bergantung pada *upons check*, *error-correction mechanisms*, dan *security measures* pada beberapa level, beberapa akses yang disyaratkan ke data dengan ruang alamat dalam aplikasi. Pekerjaan yang sama seperti proses duplikasi dalam program aplikasi, berupa: *wasting programming effort*, dan lebih penting *adding unnecessary complexity*, dan *performing redundant computations*.

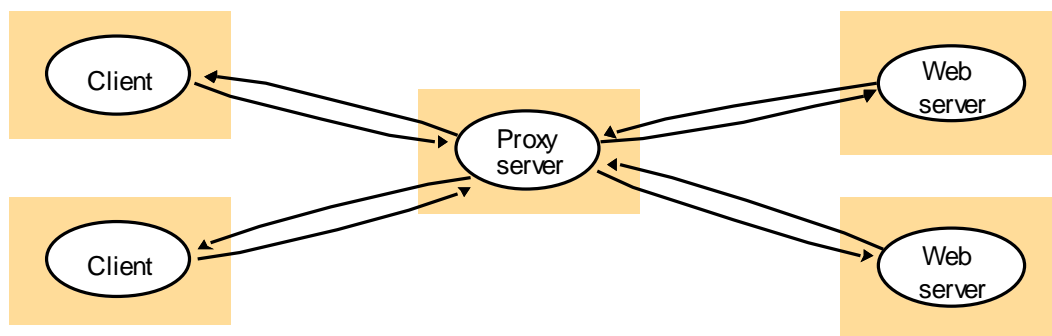
## B.2. System Architectures

Pembagian dari tanggung jawab di antara komponen-komponen sistem (aplikasi, server, dan proses-proses lain) dan penempatan dari komponen komputer dalam jaringan, dimungkinkan aspek yang lebih jelas dari desain sistem terdistribusi.



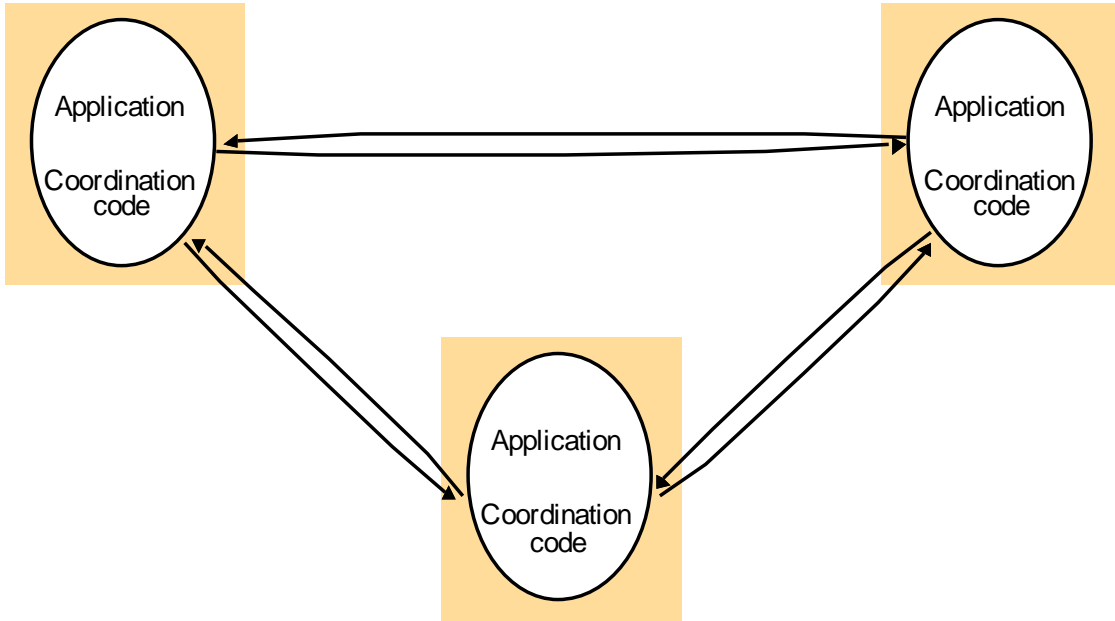
Gambar 2.3 Servis yang disediakan server jamak

Pelayanan dapat diimplementasikan seperti proses-proses server dalam komputer yang terpisah, yang berinteraksi untuk menyediakan suatu layanan ke proses client. Server dapat membagi suatu set dari objek dalam servis, yang menjadi pedoman dan mendistribusikan resource di antara client, atau server-server dapat memelihara copy replikasi dari resource ke beberapa host.



Gambar 2.4 Proxy server

Cache, merupakan sebuah penyimpanan dari data objek yang mula-mula digunakan setelah objek ditutup. Sejak sebuah objek baru diterima di komputer, akan dapat ditambahkan pada penyimpanan cache, penumpukkan beberapa objek data jika diperlukan. Web browser memelihara sebuah cache dari web pages yang mula-mula dikunjungi, dan resource web yang lain dalam lokal file system di client. Web proxy server menyediakan pembagian cache dari resource web untuk client pada sebuah website, atau melewati beberapa website. Tujuan proxy server, adalah meningkatkan kemampuan dan performa dari layanan oleh pengurangan dalam memuat daerah jaringan dan web server.



Gambar 2.5 Aplikasi terdistribusi berdasarkan proses peer

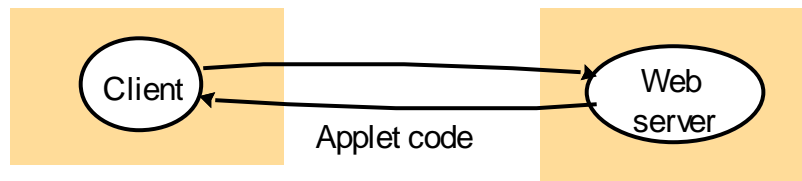
Dalam arsitektur ini, semua proses memainkan peran yang hampir mirip, proses interaksi secara kooperatif (peers), untuk menunjukkan sebuah aktivitas distribusi atau proses komputasi tanpa perbedaan di antara client dan server. Dalam model ini, code dalam proses peer memelihara konsistensi dari level aplikasi bersumber dan melakukan sinkronisasi level aplikasi saat diperlukan.

### B.3. Variasi dalam Model Client-Server

Beberapa varian dalam model client-server dapat berdasar pertimbangan dari faktor :

- Penggunaan mobile code dan mobile agents
- Penggunaan low-cost komputer
- Persyaratan untuk menambah dan menghapus mobile device

a) client request results in the downloading of applet code



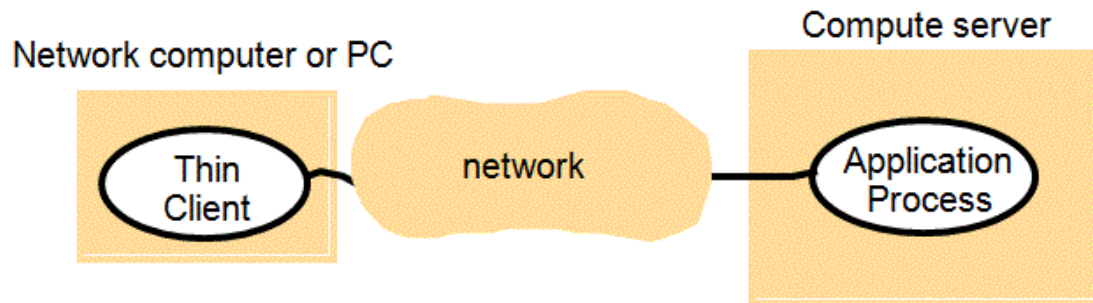
b) client interacts with the applet



Gambar 2.6 Applet Web

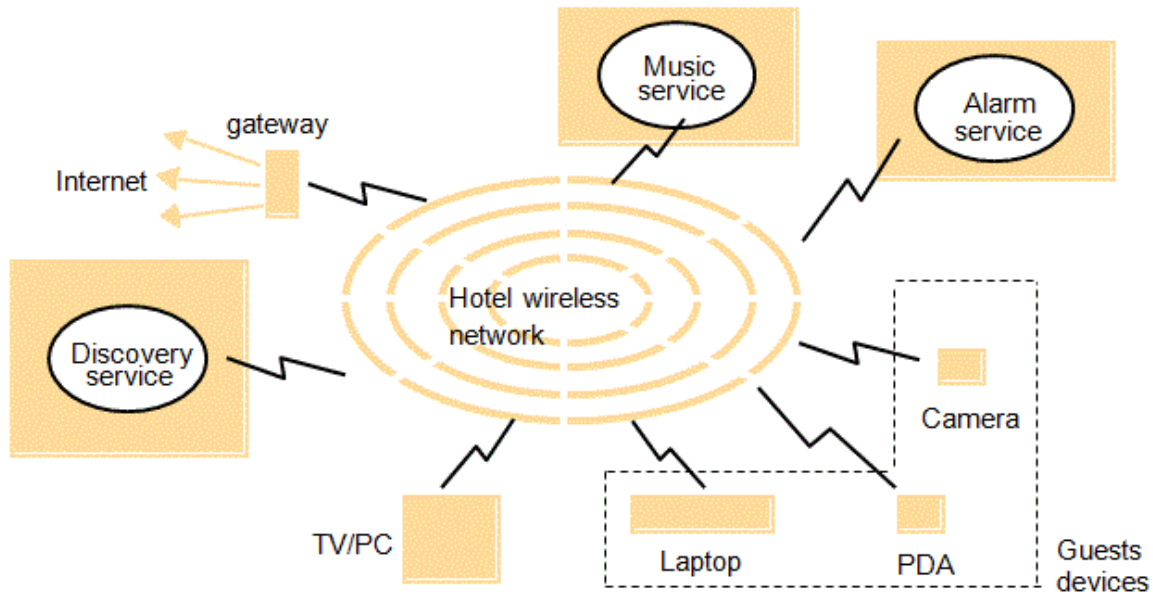
Applets, banyak digunakan pada mobile code, pengguna menjalankan browser dengan men-select suatu link ke sebuah applet yang code disimpan dalam sebuah web server. Code di-download ke browser dan menjalankan code tersebut.

Mobile agent, sebuah program yang sedang berjalan (termasuk code dan data), men-transfer dari satu komputer ke komputer lain dalam satu jaringan, melaksanakan sebuah tugas seseorang. Mobile agent dapat membuat beberapa invocation ke local resource pada setiap site yang dikunjungi.



Gambar 2.7 Klien thin dan server

Dalam arsitektur diilustrasikan aplikasi berjalan dalam sebuah komputer desktop pada pengguna. Operating system dan software aplikasi untuk komputer desktop, membutuhkan beberapa active code dan data untuk dilokasikan dalam sebuah local disk. Akan tetapi, pengaturan file aplikasi dan pemeliharaan dari sebuah base software lokal, mensyaratkan pertimbangan teknis dari sebuah pengguna yang tidak terqualifikasi. Saat thin client memilih sebuah software layer yang mendukung sebuah antarmuka berbasis window pada komputer pengguna, sambil mengeksekusi program aplikasi dalam sebuah remote komputer. Arsitektur ini mempunyai pengaturan yang lemah dan biaya hardware yang relatif rendah pada skema jaringan komputer. Proses aplikasi berjalan pada komputer server (sebuah komputer yang memiliki kemampuan menjalankan sejumlah aplikasi yang cukup besar secara bersamaan).



Gambar 2.8 Networking di dalam hotel

Gambar di atas menunjukkan sebuah jaringan nirkabel pada sebuah hotel, menyediakan beberapa fasilitas nirkabel (layanan music, sistem alarm, TV/PC, akses web, viewing service for images, beberapa device dapat mengakses jaringan nirkabel).

#### **B.4. Interfaces dan Objects**

Set dari beberapa fungsi tersedia untuk invocation dalam sebuah proses, yang dispesifikasikan dengan satu atau lebih *interface definition*. Interface definition, memiliki konsep yang hampir sama dengan beberapa bahasa (Modula, C++, Java). Dalam bentuk basic dari arsitektur client-server. Setiap proses server terlihat seperti sebuah entitas dengan interface yang tetap, disertai pendefinisian fungsi-fungsi yang dapat diminta.

Dalam bahasa berbasis objek, seperti bahasa C++ dan bahasa Java, dengan penambahan proses-proses terdistribusi yang dapat dikonstruksi dalam sebuah metode yang berbasis objek. Beberapa objek dapat dienkapsulasi dalam server atau proses peer, dan mereferensikan ke server melewati beberapa proses yang lain, sehingga metode-metode dapat diakses dengan remote invocation

#### **B.5. Design requirement for distributed architectures**

Performance issues, kesempatan penerimaan dari distribusi resource yang diperlukan dalam pengaturan update yang bersamaan. Performa jaringan dari proses yang terbatas dan kapasitas komunikasi komputer dan jaringan, yang dipertimbangkan. Quality of service, pengguna disediakan dengan functional, persyaratan dari sebuah layanan, misalnya file layanan dalam sistem terdistribusi. Properties non-functional yang utama dari sistem, mempengaruhi *quality of service* oleh client dan pengguna yang *reliability*, *security*, dan *performance*. Use of caching and replication, jaringan sering terlihat menjadi halangan utama pada penerapan sistem terdistribusi, tetapi beberapa proses telah dibuat dalam desain sistem, dimana pendapatan proses



dengan menggunakan data replication dan caching. Dependability issues, merupakan sebuah persyaratan dalam daerah aplikasi. Hal ini cukup penting tidak hanya dalam command dan aktivitas kontrol, dalam beberapa aplikasi komersial (termasuk perkembangan yang cepat dari internet commerce, dimana keamanan finansial bergantung pada kepercayaan sistem yang dioperasikan).

## **C. Fundamental Models**

Dalam sistem terdistribusi, model-model dasar yang dibahas adalah model interaksi, model kegagalan, dan model keamanan.

### ***C.1. Model Interaksi***

Sistem terdistribusi tidak dapat dilepaskan dari proses komunikasi yang tentunya merupakan bentuk interaksi. Model interaksi dalam sistem terdistribusi harus mampu menggambarkan komunikasi yang terjadi memiliki hambatan berupa tunda waktu. Sehingga, dapat dilakukan penggunaan proses yang terbaik untuk setiap kondisi. Ada dua faktor utama yang memengaruhi proses interaksi dalam sistem terdistribusi, yakni performa komunikasi dan tiadanya clock global.

Performa komunikasi diukur dengan latency, bandwidth, dan jitter. Latency berarti waktu tunda dalam pengiriman pesan. Bandwidth berarti jumlah informasi yang dapat dikirim dalam sebuah satuan waktu. Jitter adalah variasi waktu yang dibutuhkan untuk mengirim runtut bit pesan.

Dalam sistem terdistribusi, tidak ada clock global karena setiap komputer memiliki clocknya masing-masing, sehingga diperlukan cara untuk mengatasi hal tersebut. Salah satu cara yang digunakan untuk mengatasinya adalah dengan menggunakan agreement di antara dua clock.

Dalam model interaksi, terdapat dua jenis interaksi sistem terdistribusi, yakni sinkron dan asinkron. Sistem terdistribusi sinkron adalah sistem yang mengirimkan pesan dalam waktu yang dibatasi. Waktu eksekusi dalam sistem ini juga dibatasi. Sementara itu, sistem terdistribusi asinkron adalah sistem yang tidak terikat oleh waktu.

Dalam sistem ini, juga terdapat masalah event ordering yang membuat urutan pesan tidak sesuai kebenaran karena adanya delay dalam pengiriman pesan. Selain itu, juga terdapat agreement in Pepperland yang artinya menentukan siapa yang mengirim pesan dan kapan terjadi pengiriman pesan. Untuk mengatasi kesalahan urutan di dalamnya, digunakan logical times yang memungkinkan ordering dalam proses pengiriman pesan.

### ***C.2. Model Kegagalan***

Dalam sebuah sistem, selalu terdapat kemungkinan untuk gagal yang dapat dimodelkan. Untuk sistem terdistribusi, model kegagalannya meliputi hal-hal berikut.

### *Kegagalan Omission*

Kegagalan ini berarti proses gagal melakukan pengiriman pesan. Kegagalan yang terjadi dapat berupa kegagalan proses atau kegagalan komunikasi. Kegagalan ini dapat berupa crash. Crash tersebut menyebabkan proses terhenti. Crash dalam proses dapat disebut *fail-stop* jika proses-proses yang lain dapat mendeteksi crash yang terjadi. Fail-stop dapat dihasilkan dalam sebuah sistem yang sinkron, jika sebuah proses menggunakan timeouts untuk melakukan deteksi ketika proses-proses yang lain gagal merespon dan pesan dijamin dapat terkirim.

Kegagalan komunikasi juga termasuk bagian dari kegagalan omission. Secara umum, komunikasi terdiri atas proses pengiriman dan penerimaan pesan. Kegagalan dapat terjadi jika kanal komunikasi tidak dapat menyampaikan pesan dari pengirim ke penerima. Kegagalan tersebut terjadi dalam pengiriman pesan dari buffer pengirim ke buffer penerima atau disebut sebagai *dropping messages*. *Dropping messages* biasanya disebabkan oleh kurangnya ruang untuk buffer dalam penerima.

### *Kegagalan Arbitrary*

Kegagalan arbitrary atau Byzantine juga termasuk bagian model kegagalan sistem terdistribusi. Kegagalan arbitrary tersebut digunakan untuk menggambarkan kemungkinan terburuk dari kegagalan semantik.

### *Kegagalan Pewaktuan (Timing Failure)*

Timing failure terjadi dalam sistem yang sinkron dan memiliki batasan-batasan waktu yang terdiri atas waktu eksekusi proses, waktu pengiriman pesan, dan pesat *clock drift*. Kegagalan pewaktuan tersebut dapat mengakibatkan tidak adanya pesan respon dalam waktu tertentu. Kegagalan clock dapat mengakibatkan kesalahan dalam proses, yakni clock proses melebihi batas waktu yang ditentukan. Selain itu, juga terdapat kegagalan performa yang mengakibatkan kesalahan pada proses dan kanal. Kesalahan pada proses terjadi jika proses melebihi batas interval antara dua langkah yang ada, sementara itu, kesalahan pada kanal terjadi jika transmisi pesan memerlukan waktu yang lebih dari yang ditentukan.

### *Masking Failures*

Dalam sistem terdistribusi, setiap komponen tersusun atas beberapa jenis komponen yang memungkinkan membangun layanan, tetapi juga menimbulkan masalah. Contohnya, multiple servers yang menyimpan data-data dapat menyediakan layanan ketika salah satu di antara komponennya mengalami *crash*. Adanya crash tersebut dapat membuat sistem memiliki kemampuan untuk menutupinya. Sebuah layanan dapat menutupi kesalahannya dengan menyembunyikannya atau mengubah kesalahan tersebut ke dalam bentuk tipe kesalahan yang lebih dapat diterima sistem, misalnya sebuah checksums dapat digunakan untuk menutupi pesan yang corrupted dengan mengubah arbitrary failure ke dalam bentuk omission failure.

### *Reliable Communication*

Reliable communication berarti komunikasi yang handal dan terdiri atas dua unsur utama, yakni validity dan integrity. Validity berarti setiap pesan yang berada dalam buffernya pesan

pengirim harus benar-benar terkirim ke buffer penerima. Integrity berarti pesan yang diterima harus benar-benar sama dengan pesan yang dikirim.

Gangguan yang terjadi dalam hal integrity dapat berasal dari hal-hal berikut.

- Adanya aturan yang mengharuskan pesan dikirim ulang dan tetap membiarkan pesan tersebut terkirim dua kali.
- Adanya pengguna yang dapat memasukkan pesan palsu, mengulang pesan lama, atau merusakkan pesan yang ada.

### ***C.3. Model Keamanan***

Secara umum, model keamanan mengidentifikasi adanya kemungkinan ancaman terhadap proses dan komunikasi dalam sistem terdistribusi.

#### *Protecting Objects*

Object digunakan dalam beberapa jalan yang berbeda oleh pengguna yang berbeda. Contohnya, beberapa objek dapat menyimpan data privat pengguna. Untuk mendukungnya, terdapat hak akses yang memungkinkan hanya pemilik yang dapat mengaksesnya. Maka, pengguna harus diposisikan sebagai pihak yang memanfaatkan hak akses itu. Jadi, pengguna dapat mengirimkan permintaan dan server memberikan balasan atas permintaan tersebut. Server memiliki tanggung jawab untuk memilih siapa saja yang bisa melakukan akses. Jika pengguna diidentifikasi tidak memiliki hak akses, maka permintaannya ditolak. Hanya pengguna yang memiliki hak akses yang bisa diterima oleh server. Hal ini dapat dikatakan sebagai tindakan melindungi obyek atau protecting object.

#### *Securing Processes and Interactions*

Interaksi yang terjadi dalam sistem biasanya berupa pengiriman pesan yang bisa saja diserang oleh pihak luar. Penyerangan ini sangat mungkin terjadi karena pengiriman pesan dilakukan dalam jaringan yang terbuka. Penyerangan yang dilakukan pihak yang tidak bertanggung jawab ini biasanya meliputi aplikasi transaksi keuangan atau informasi-informasi yang penting.

Untuk memodelkan ancaman terhadap keamanan sistem, dibuat sebuah model enemy yang bisa mengirimkan pesan ke proses yang lain dan menyalin pesan antara dua pihak yang sedang berkomunikasi atau dapat disebut menyadap pesan. Tindakan ini dapat dilakukan dengan cara menghubungkan diri ke jaringan yang digunakan untuk komunikasi tersebut. Kemudian, enemy dapat melakukan penyerangan yang dia inginkan.

Ancaman yang dihadapi dapat terdiri atas ancaman terhadap prosesnya, ancaman terhadap kanal komunikasinya, dan layanan yang menolak.

Proses yang dibuat untuk menangani masalah permintaan yang masuk dapat menerima pesan dari proses lain, tetapi tidak dapat mengetahui identitas pengirim. Protokol komunikasi dapat memberikan informasi alamat pengirim, tetapi hal ini memudahkan enemy untuk menyerang. Maka, proses komunikasi yang terjadi mendapat ancaman dari pihak luar.

Pihak luar juga dapat melakukan penyalinan, pengubahan, atau menambahi pesan ketika mereka sedang berada dalam jaringan yang sama dengan proses komunikasi pihak lain. Contohnya, pesan hasil yang membawa pesan pengguna dapat dibuka oleh pengguna lain dan bisa saja diubah isinya. Hal tersebut merupakan ancaman terhadap kanal komunikasi.

Untuk mengatasi masalah keamanan tersebut, dapat dilakukan tindakan yang pengamanan, yakni memanfaatkan teknik kriptografi dan autentikasi. Kriptografi adalah ilmu yang bertujuan untuk menjaga agar pesan yang dikirim itu tetap dalam keadaan aman. Dalam kriptografi, terdapat proses enkripsi untuk membuat pesan yang dikirim itu hanya dapat dikenali oleh pihak-pihak yang berwenang. Sementara itu, autentikasi berarti pengecekan identitas yang akan mengakses pesan. Dengan metode kriptografi dan autentikasi, diharapkan dapat tercipta sebuah kanal komunikasi yang lebih aman.

Sementara itu, pihak-pihak yang tidak bertanggung jawab tetap saja masih mungkin memberikan ancaman lainnya, seperti denial of service dan mobile code. Denial of service adalah sebuah bentuk serangan di mana enemy mengganggu dengan jalan membuat permintaan yang berlebihan dalam layanan. Contohnya, operasi door locks dalam sebuah gedung dapat diaktifkan oleh sebuah serangan yang memenuhi kendalinya dengan request yang tidak jelas.

Bentuk lain ancaman adalah mobile code yang mengganggu proses yang menerima dan mengeksekusi kode dari pihak lain, seperti lampiran di e-mail. Metode ini mirip dengan metode yang dilakukan kuda Troya untuk mengelabui lawannya.

#### *Penggunaan Security Model*

Security model digunakan untuk mengatasi ancaman-ancaman yang mungkin dihadapi dalam jaringan komunikasi. Untuk mengatasi ancaman-ancaman tersebut, dapat digunakan metode kriptografi dan autentikasi. Dengan metode-metode tersebut, diharapkan proses dan kanal komunikasi benar-benar menjadi aman oleh pihak yang tidak bertanggung jawab.