

CSG3L3

SISTEM TERDISTRIBUSI

Topik 2 : Model Arsitektur Sistem Terdistribusi





Tantangan (Implementasi) Sistem Terdistribusi

<i>Distributed systems:</i>	<i>Early</i>	<i>Internet-scale</i>	<i>Contemporary</i>
<i>Scale</i>	Small	Large	Ultra-large
<i>Heterogeneity</i>	Limited (typically relatively homogenous configurations)	Significant in terms of platforms, languages and middleware	Added dimensions introduced including radically different styles of architecture
<i>Openness</i>	Not a priority	Significant priority with range of standards introduced	Major research challenge with existing standards not yet able to embrace complex systems
<i>Quality of service</i>	In its infancy	Significant priority with range of services introduced	Major research challenge with existing services not yet able to embrace complex systems



Pada Sistem Terdistribusi, apa saja yang berkomunikasi dan bagaimana cara mereka berkomunikasi?

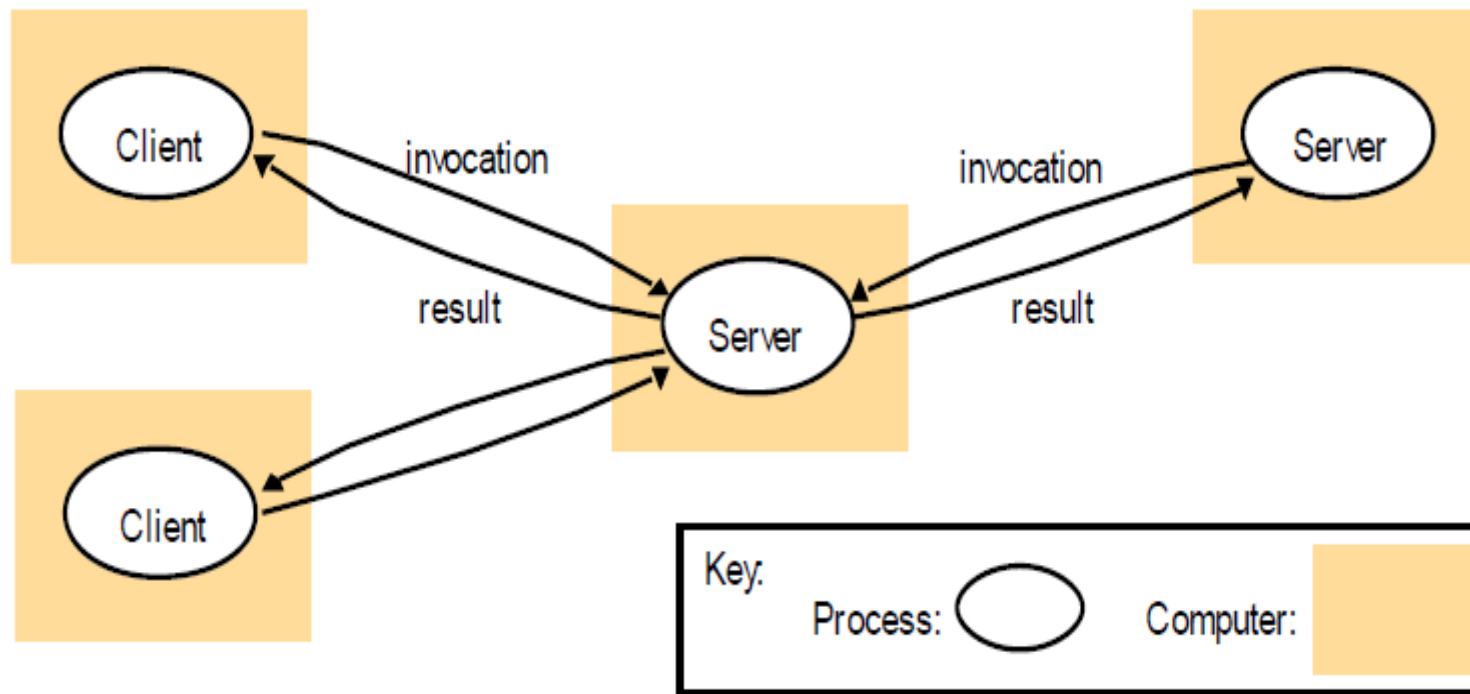
Ada yang disebut sebagai entitas dan paradigma komunikasi pada Sistem Terdistribusi yang dirangkum dan dikategorikan menjadi beberapa kelompok pada tabel berikut ini:



<i>Communicating entities (what is communicating)</i>		<i>Communication paradigms (how they communicate)</i>		
<i>System-oriented entities</i>	<i>Problem- oriented entities</i>	<i>Interprocess communication</i>	<i>Remote invocation</i>	<i>Indirect communication</i>
Nodes	Objects	Message passing	Request- reply	Group communication
Processes	Components	Sockets	RPC	Publish-subscribe
	Web services	Multicast	RMI	Message queues
				Tuple spaces



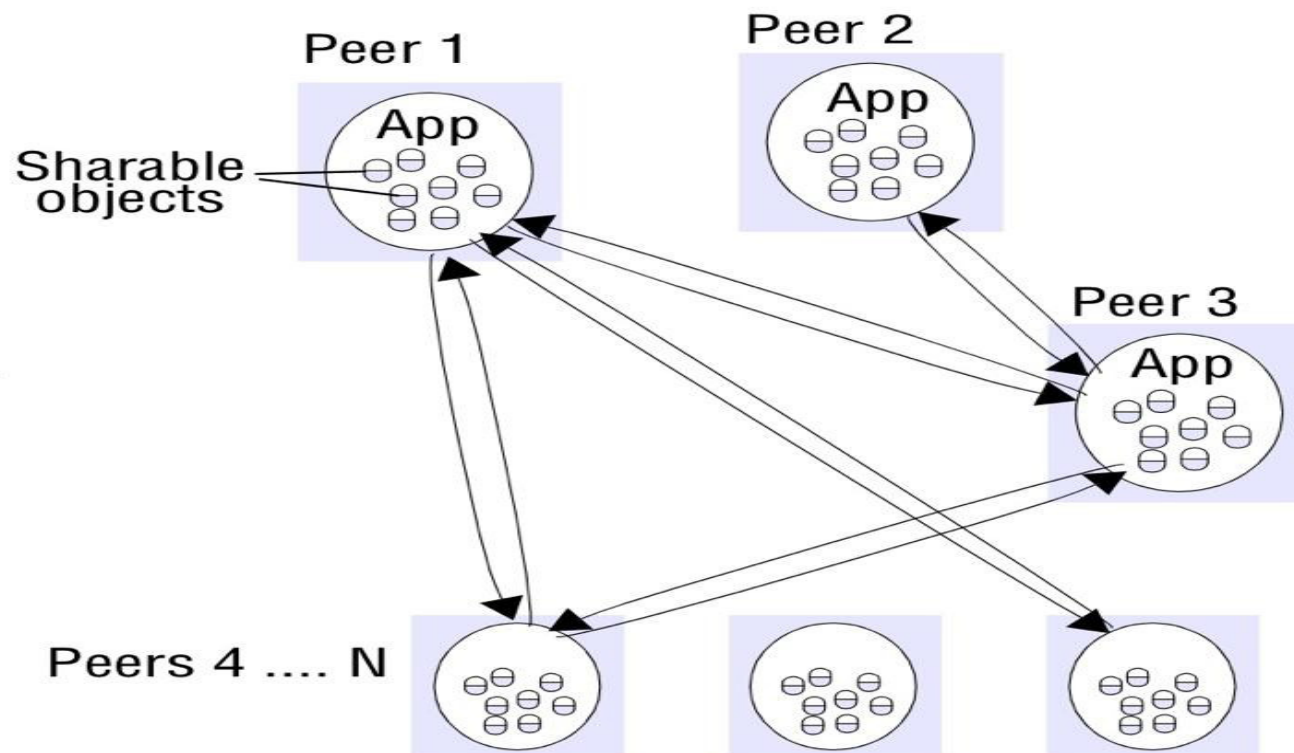
Hubungan *Client Server*



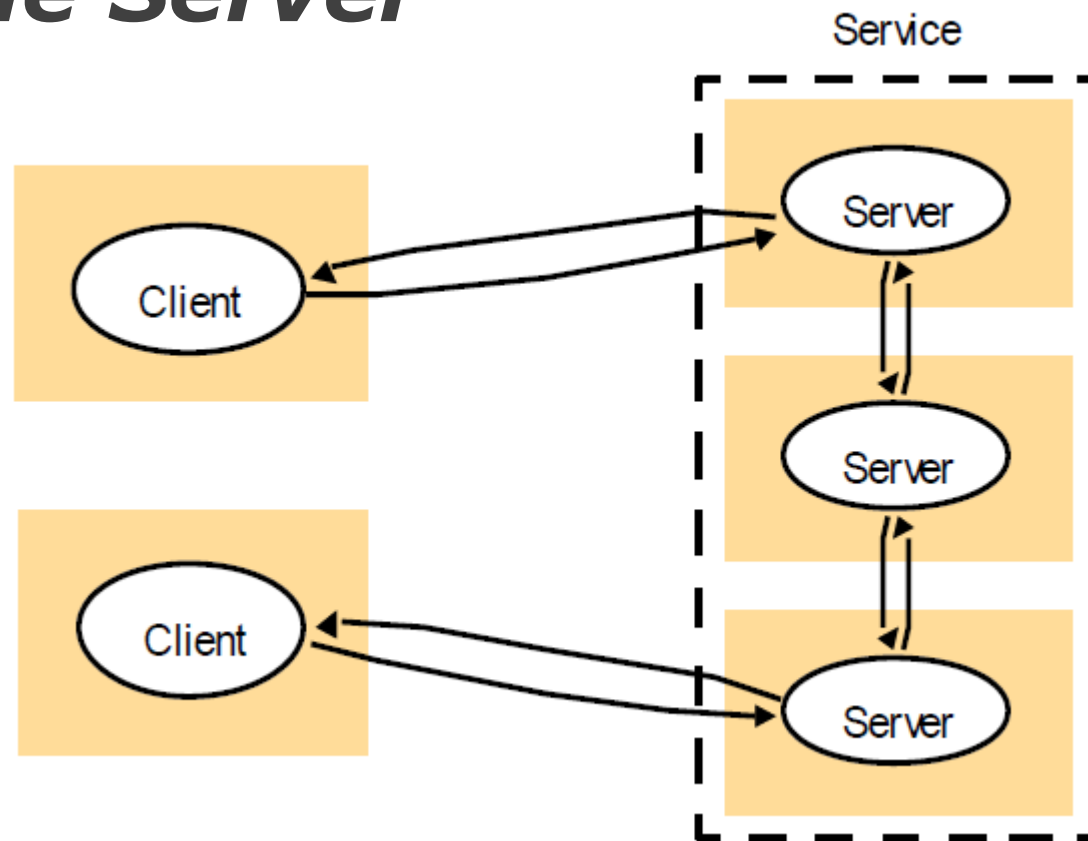
Bagaimana mekanismenya?



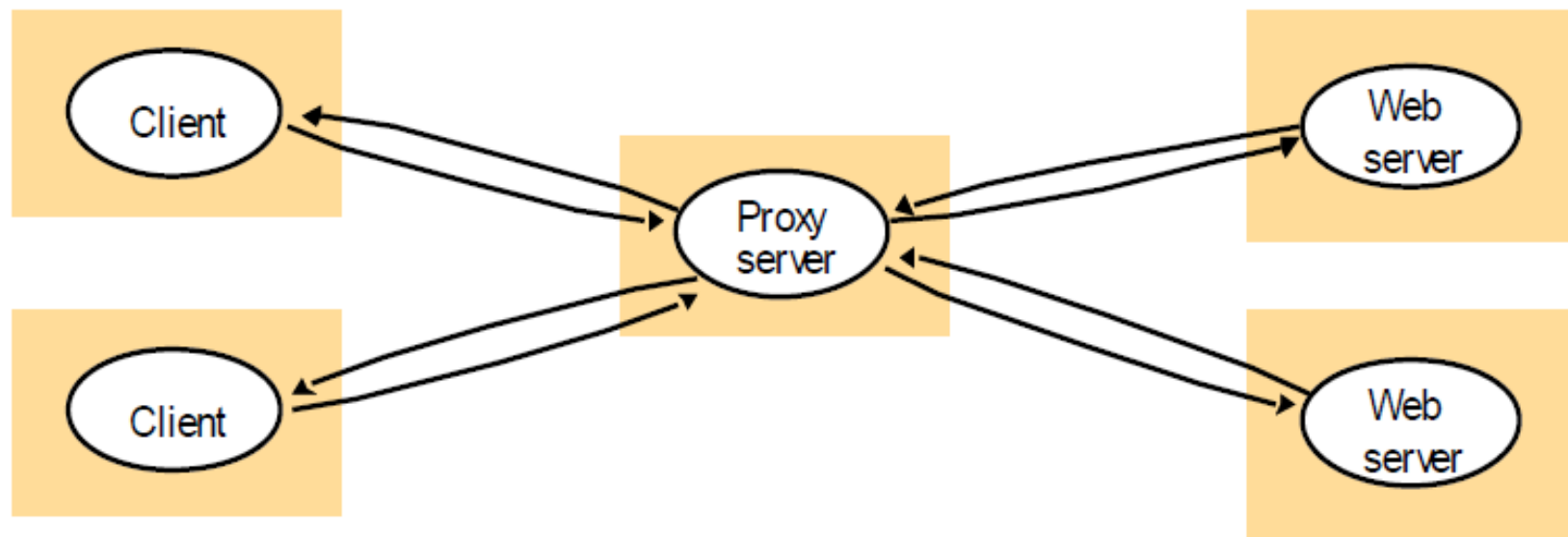
Peer-to-Peer (P2P)



Multiple Server

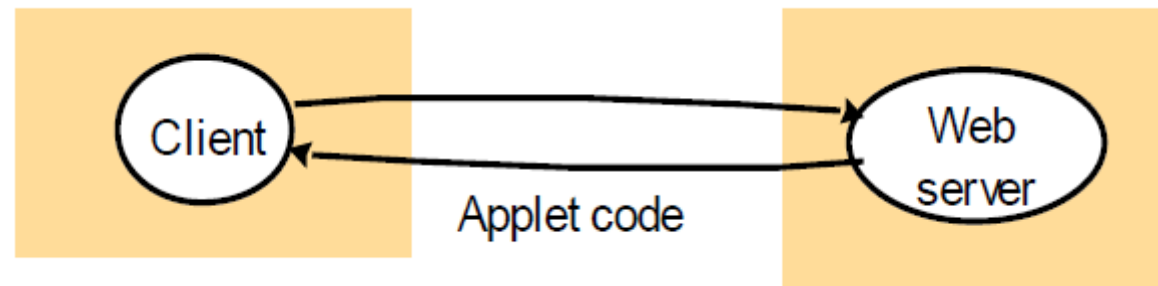


Web Proxy Server





Web Applets

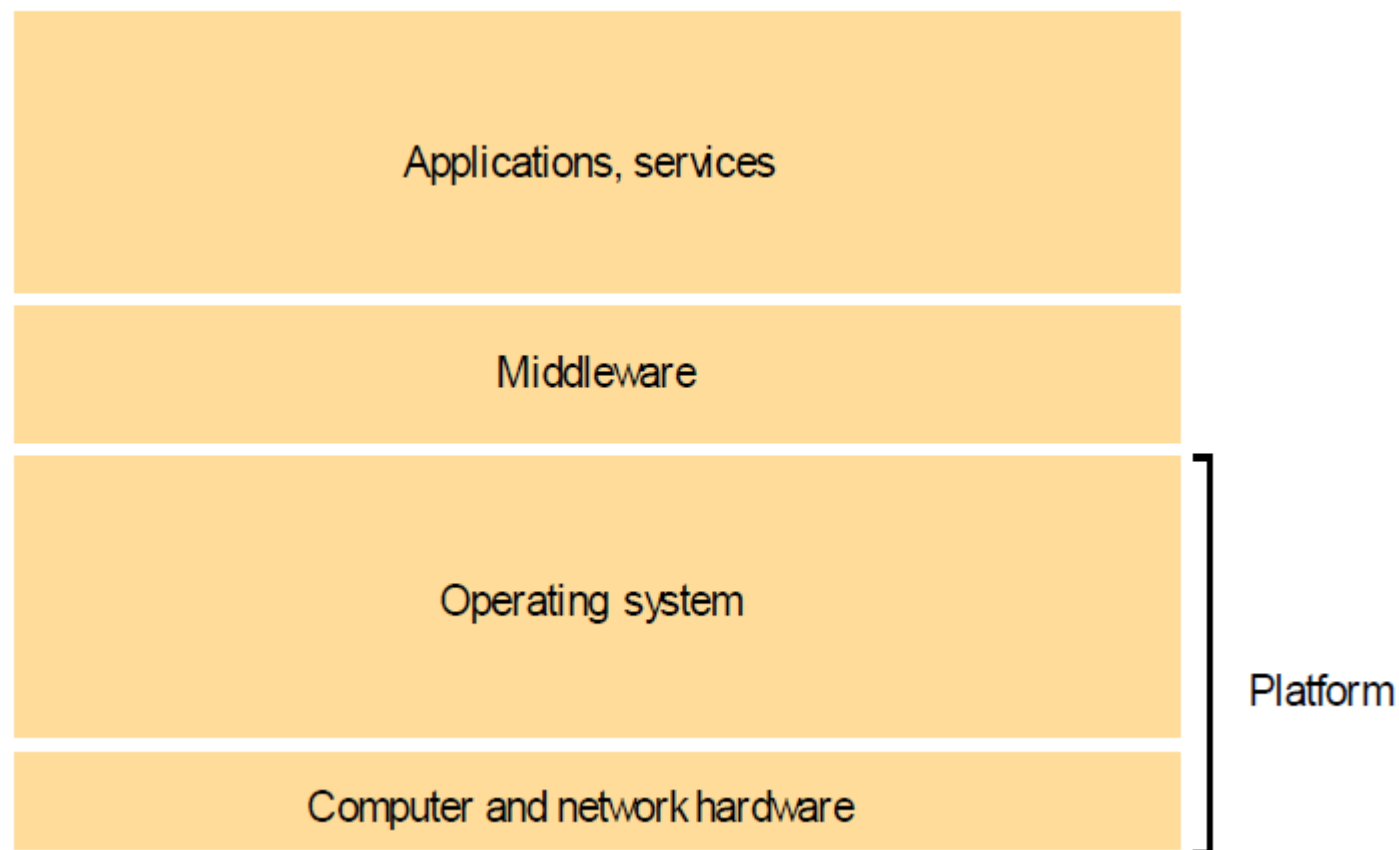


Klien mengirimkan *request* ke server untuk men-*download* kode *applet*.

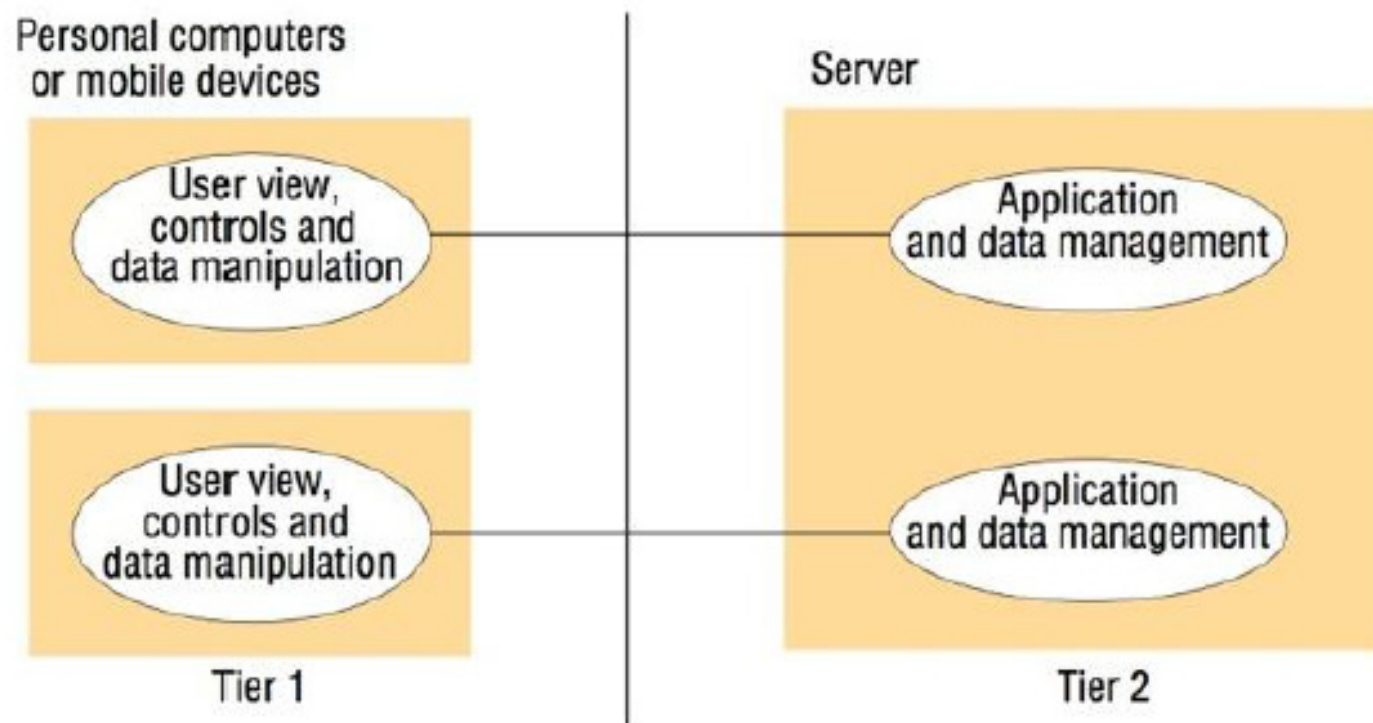


Klien berinteraksi dengan *applet*.

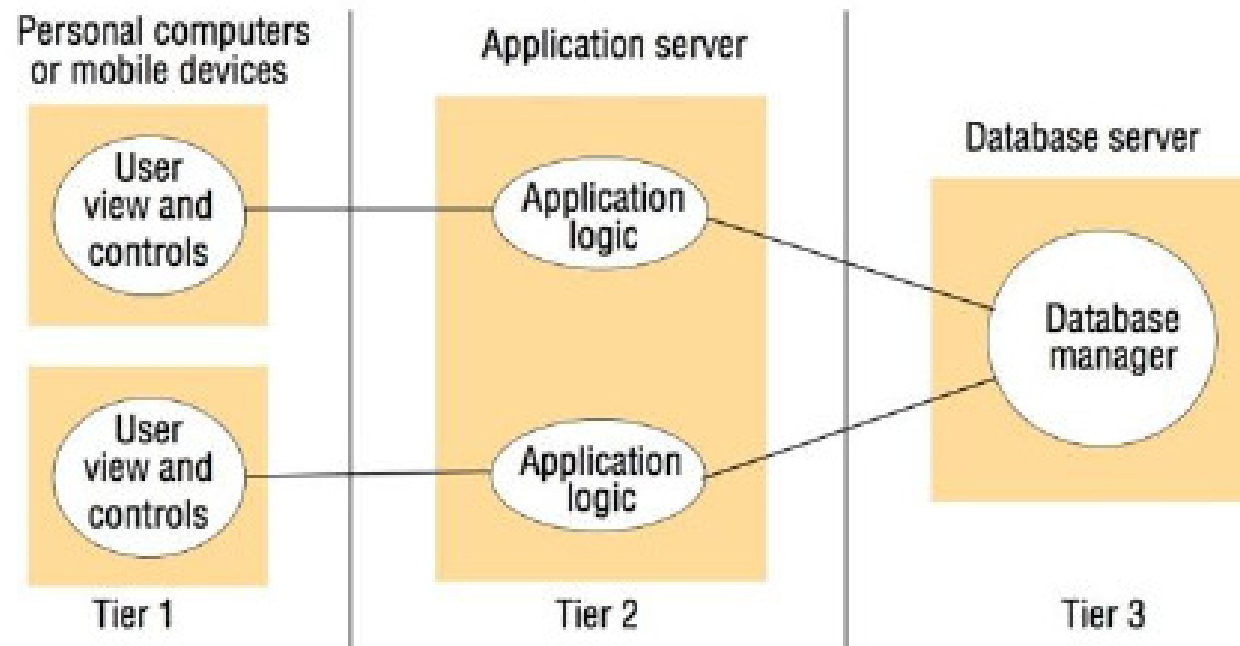
***Layer* pada Sistem Terdistribusi**



Arsitektur *Two-tier*



Arsitektur *Three-tier*

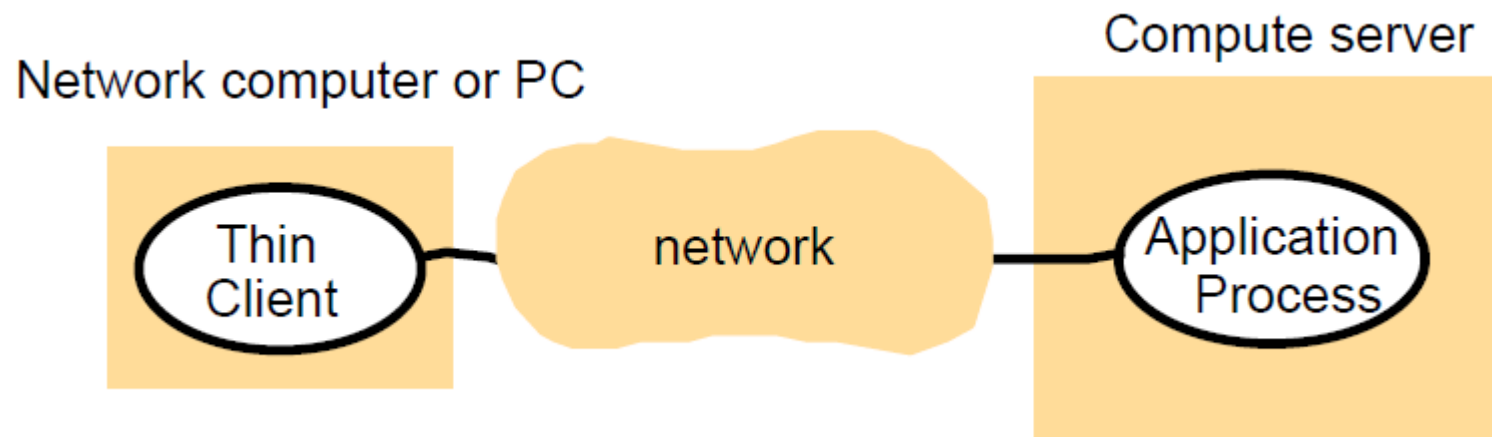


Kode AJAX: fungsi apakah ini?

```
$.ajax({  
    url: 'scores.php?game=Arsenal:Liverpool',  
    dataType: 'json',  
    onSuccess: function(score) {  
        updateScore(score);  
    }  
});  
  
function updateScore(score) { /* kode manipulasi elemen pada  
    halaman web */ }
```

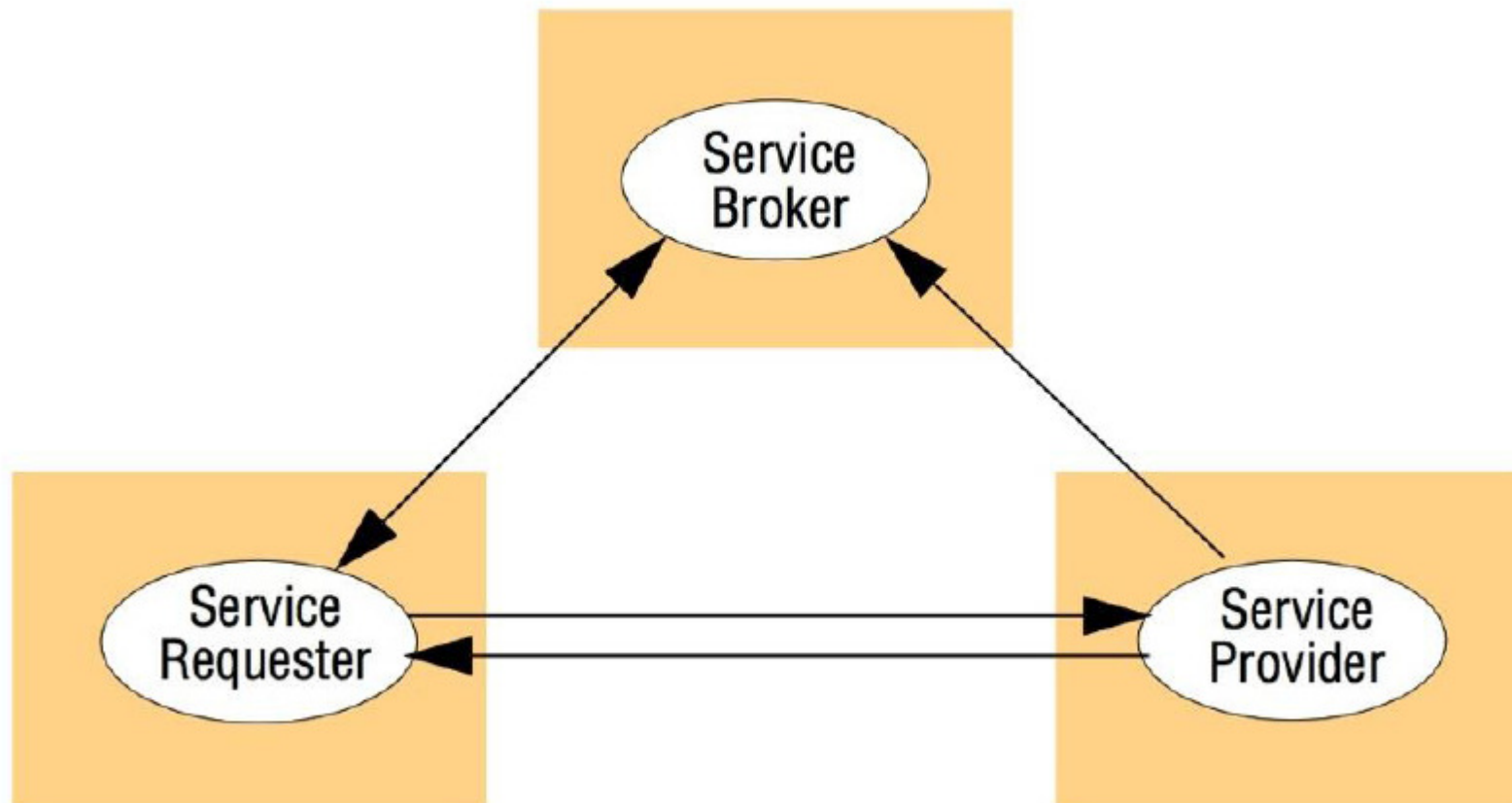


Thin Client



Dimana teman-teman menemukan arsitektur seperti gambar di atas?

Pola Arsitektur pada *Web Service*

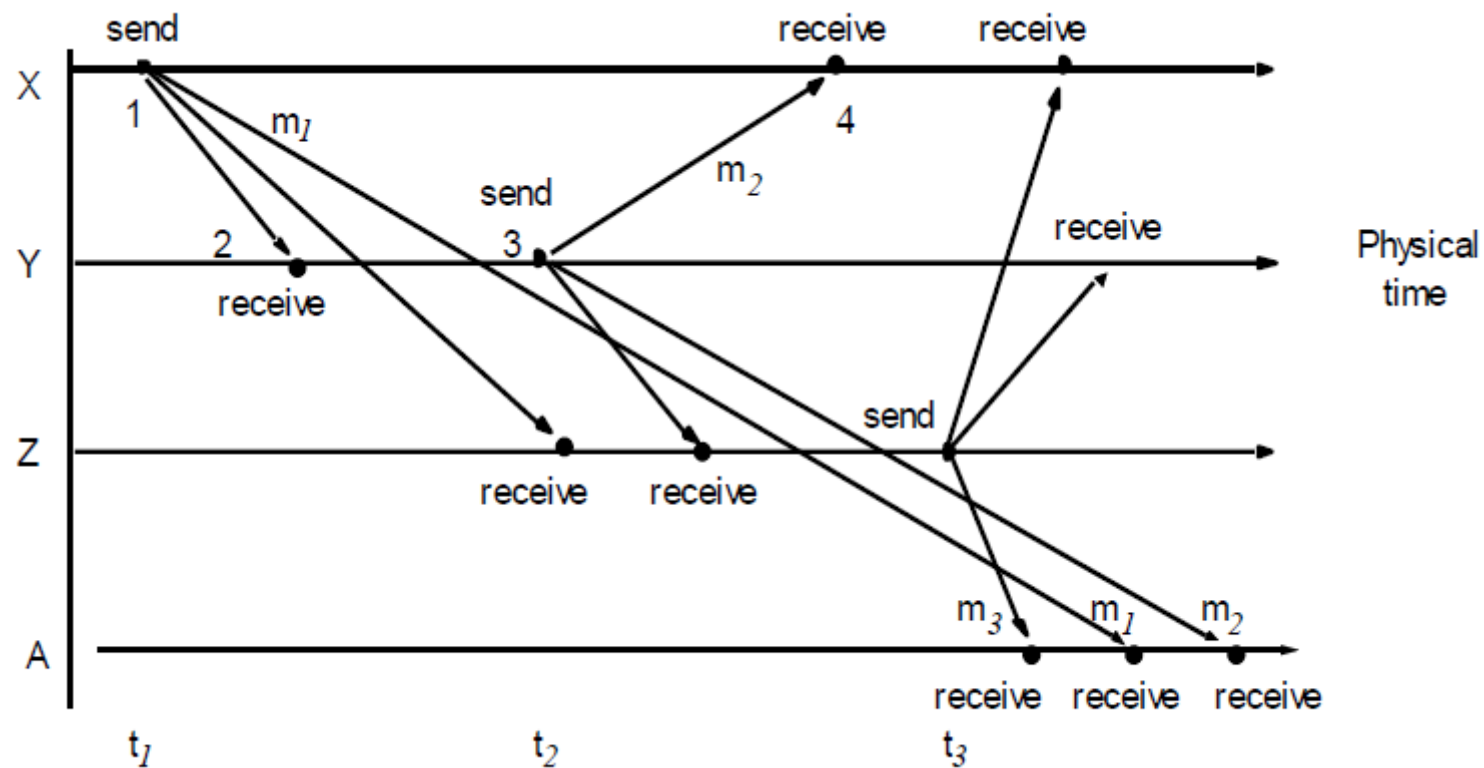


Kategori dari *Middleware*

<i>Major categories:</i>	<i>Subcategory</i>	<i>Example systems</i>
<i>Distributed objects (Chapters 5, 8)</i>	Standard	RM-ODP
	Platform	CORBA
	Platform	Java RMI
<i>Distributed components (Chapter 8)</i>	Lightweight components	Fractal
	Lightweight components	OpenCOM
	Application servers	SUN EJB
	Application servers	CORBA Component Model
	Application servers	JBoss
<i>Publish-subscribe systems (Chapter 6)</i>	-	CORBA Event Service
	-	Scribe
	-	JMS
<i>Message queues (Chapter 6)</i>	-	Websphere MQ
	-	JMS
<i>Web services (Chapter 9)</i>	Web services	Apache Axis
	Grid services	The Globus Toolkit
<i>Peer-to-peer (Chapter 10)</i>	Routing overlays	Pastry
	Routing overlays	Tapestry
	Application-specific	Squirrel
	Application-specific	OceanStore
	Application-specific	Ivy
	Application-specific	Gnutella

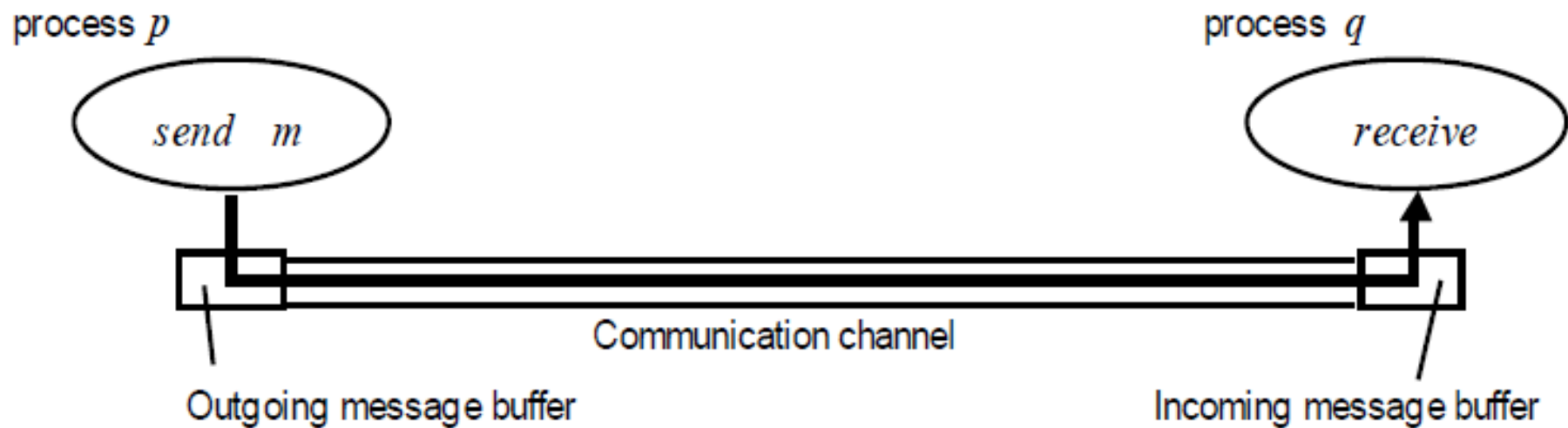


Skema urutan event pada *real-time system*





Proses dan Kanal (Saluran Komunikasi)





Omission dan Arbitrary Failures

Kelas <i>failure</i>	Aspek yang dipengaruhi	Deskripsi
<i>Fail-stop</i>	Proses	Sebuah proses <i>fail</i> (berhenti) begitu juga dengan proses lainnya. Proses lain dapat mendeteksi kondisi ini.
<i>Crash</i>	Proses	Sebuah proses <i>fail</i> begitu juga dengan proses lainnya. Proses lain tidak dapat mendeteksi kondisi ini.
<i>Omission</i>	Kanal	Sebuah pesan masuk ke <i>buffer outgoing message</i> tetapi sampai ke <i>node</i> yang bukan tujuan semestinya.
<i>Send-omission</i>	Proses	Proses pengiriman pesan berhasil tetapi pesan tidak ada di <i>buffer</i> pesan keluar (<i>outgoing message buffer</i>).



Omission and Arbitrary Failures (cont'd)

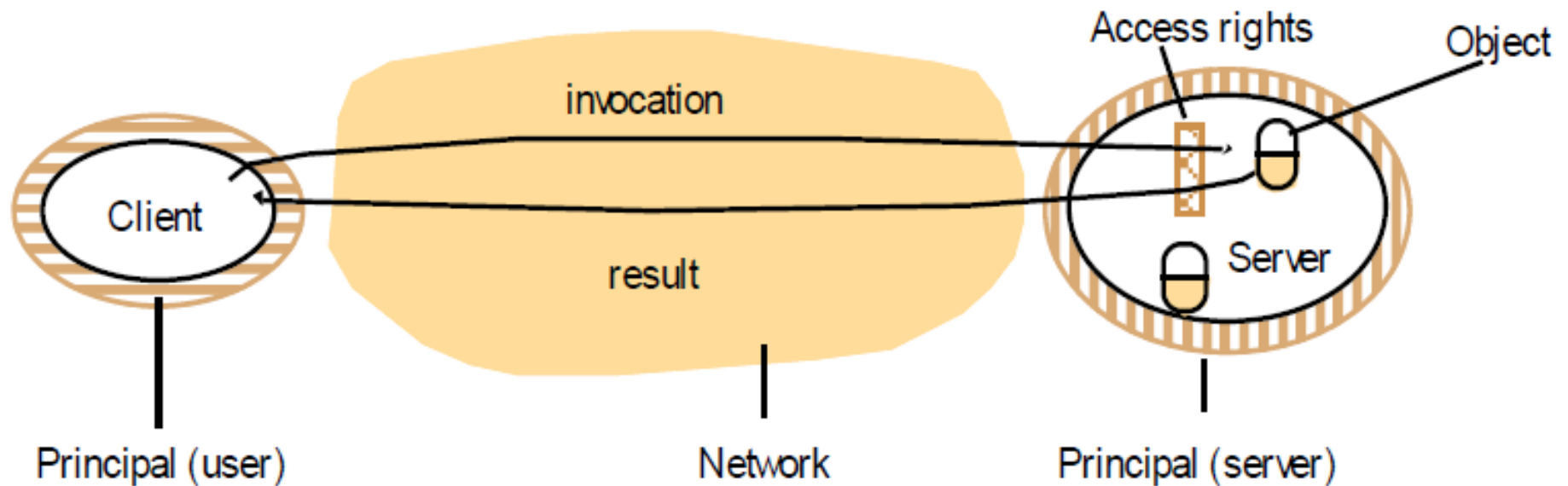
Kelas <i>failure</i>	Aspek yang dipengaruhi	Deskripsi
<i>Receive-omission</i>	Proses	Pesan sampai di <i>incoming message buffer</i> namun proses tidak menerima pesan tersebut.
<i>Arbitrary (Byzantine)</i>	Proses / kanal	Kelakuan aneh pada proses/kanal tertentu (<i>arbitrary behavior</i>). Bisa jadi mengirim pesan acak sewaktu-waktu, melakukan <i>commit omission</i> , proses berhenti secara tiba-tiba atau melakukan aksi yang tidak semestinya.



Timing Failures

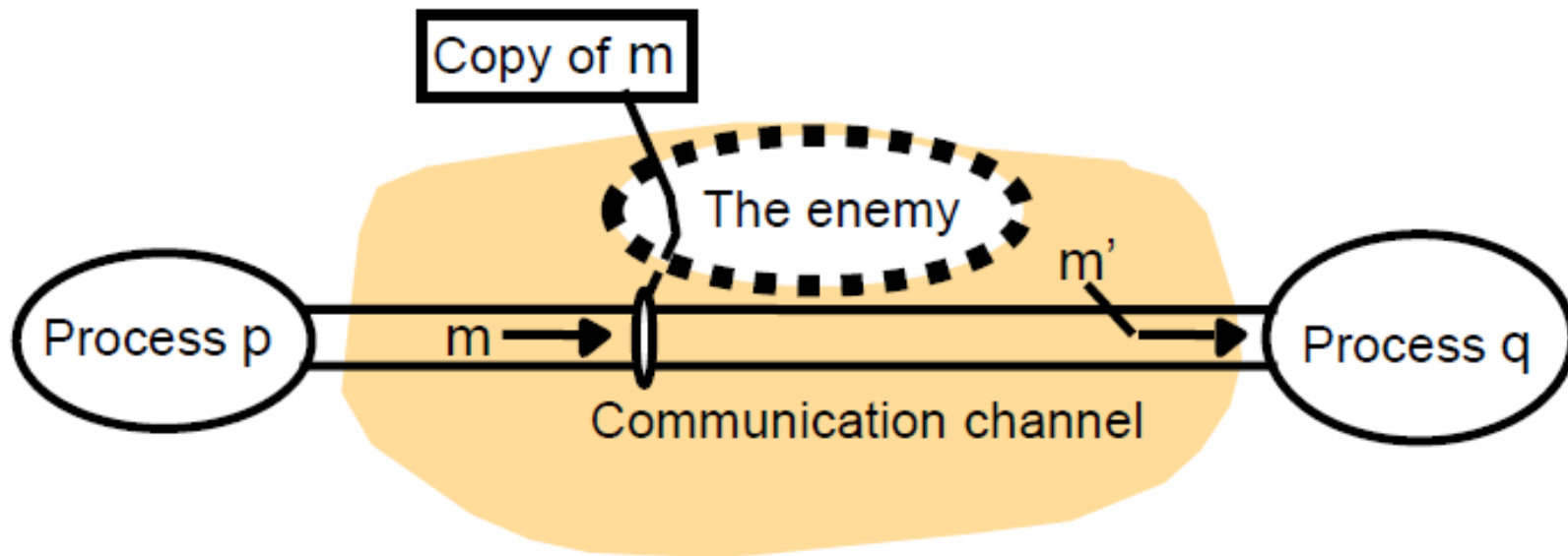
Kelas <i>failure</i>	Aspek yang dipengaruhi	Deskripsi
<i>Clock</i>	Proses	Waktu lokal sebuah proses melebihi batas atau melenceng dari sistem waktu nyata
<i>Performance</i>	Proses	Interval waktu antar dua aksi pada sebuah proses melebihi batas waktu yang semestinya
<i>Performance</i>	Kanal	Lama waktu untuk mentransmisikan sebuah pesan lebih dari batas yang telah ditentukan

Akses *Object* secara Prinsipal



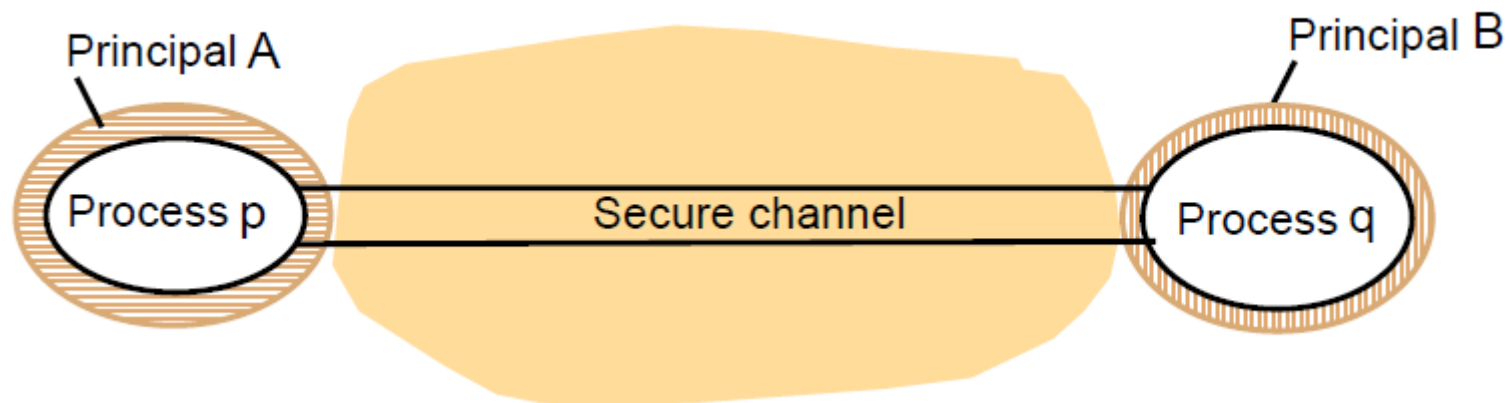


“Musuh” di Jaringan Komputer



**Apa serangan yang mungkin dilakukan?
Apa akibatnya? Bagaimana solusinya?**

Pengamanan Kanal Saat Transmisi Data





Ada Pertanyaan?



Referensi

- ▶ Coulouris, G. F., Dollimore, J., & Kindberg, T. (2012). *Distributed Systems: Concepts and Design 5th Edition*. London: Pearson Education.



Fakultas Informatika
School of Computing
Telkom University



THANK YOU