Sistem Terdistribusi
TIK-604

Pengantar Perkuliahan

Pertemuan 1: 11 s.d 13 Februari 2019

Husni

07
Februari
2019

4,143,377,949
Internet Users in the world

1,955,133,427
Total number of Websites

176,471,402,980
Emails sent today

4,437,054,056
Google searches today

4,205,967
Blog posts written today

512,797,323
Tweets sent today

4,744,165,031
Videos viewed today on YouTube

54,892,390
Photos uploaded today on Instagram

91,149,352
Tumblr posts today

07 Februari 2019

2,421,717,514
Facebook active users

676,715,672
Google+ active users

343,772,839
Twitter active users

275,580,013
Pinterest active users

222,416,621
Skype calls today

82,989
Websites hacked today

499,617
Computers sold today

2,960,104
Smartphones sold today

318,843
Tablets sold today

07 Februari 2019

4,172,381,398 GB

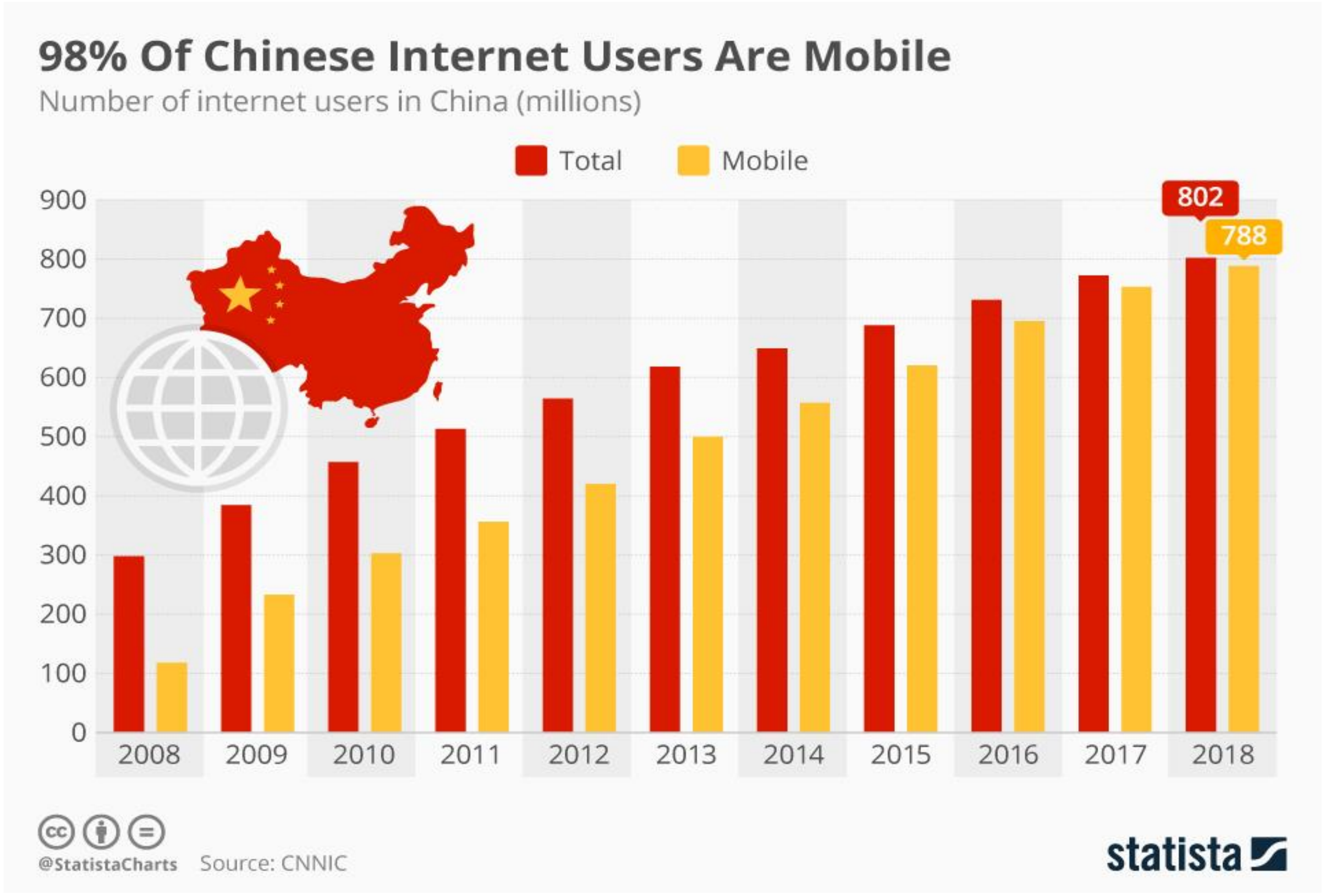
Internet traffic today

2,717,080 MWh

Electricity used today for the Internet

2,282,026 tons

CO_2 emissions today from the Internet

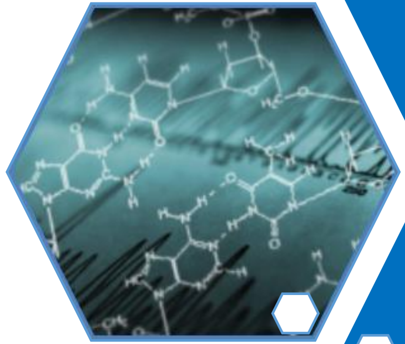


- Desember 2018: 4.1 milyar pengguna Internet
- Desember 2017: 3.7 milyar
- 49% Asia
- 20% = 802 juta dari China
- India: 500 juta
- 52.2% *Mobile Traffic*

Menuju Abad Puncak: Terobosan

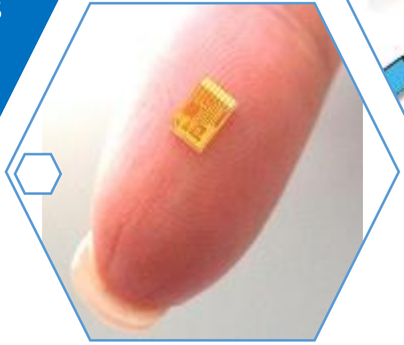
Astronomy

Ubiquitous Computing



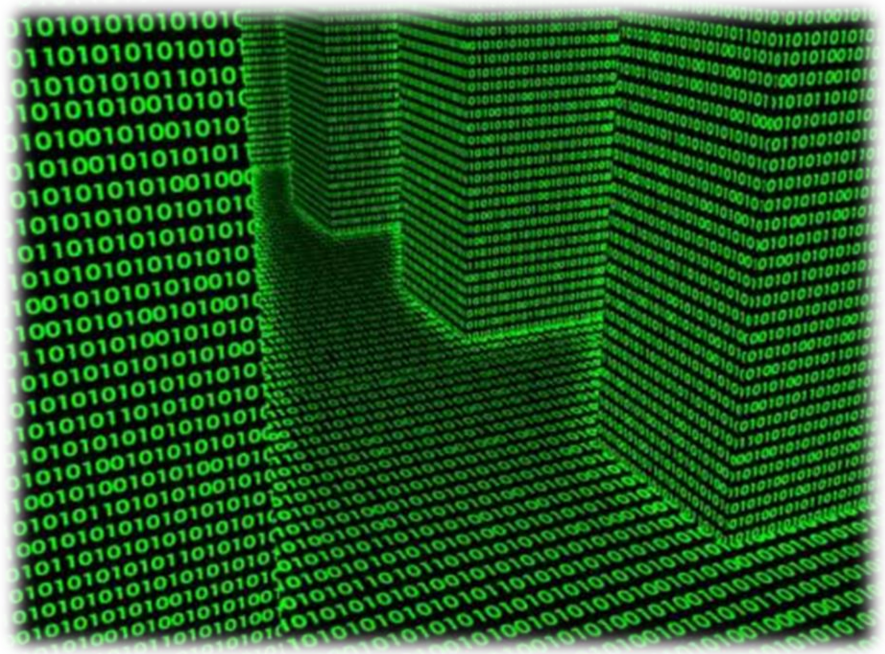
Smaller, Faster, Cheaper Sensors

Gene Sequencing and Biotechnology

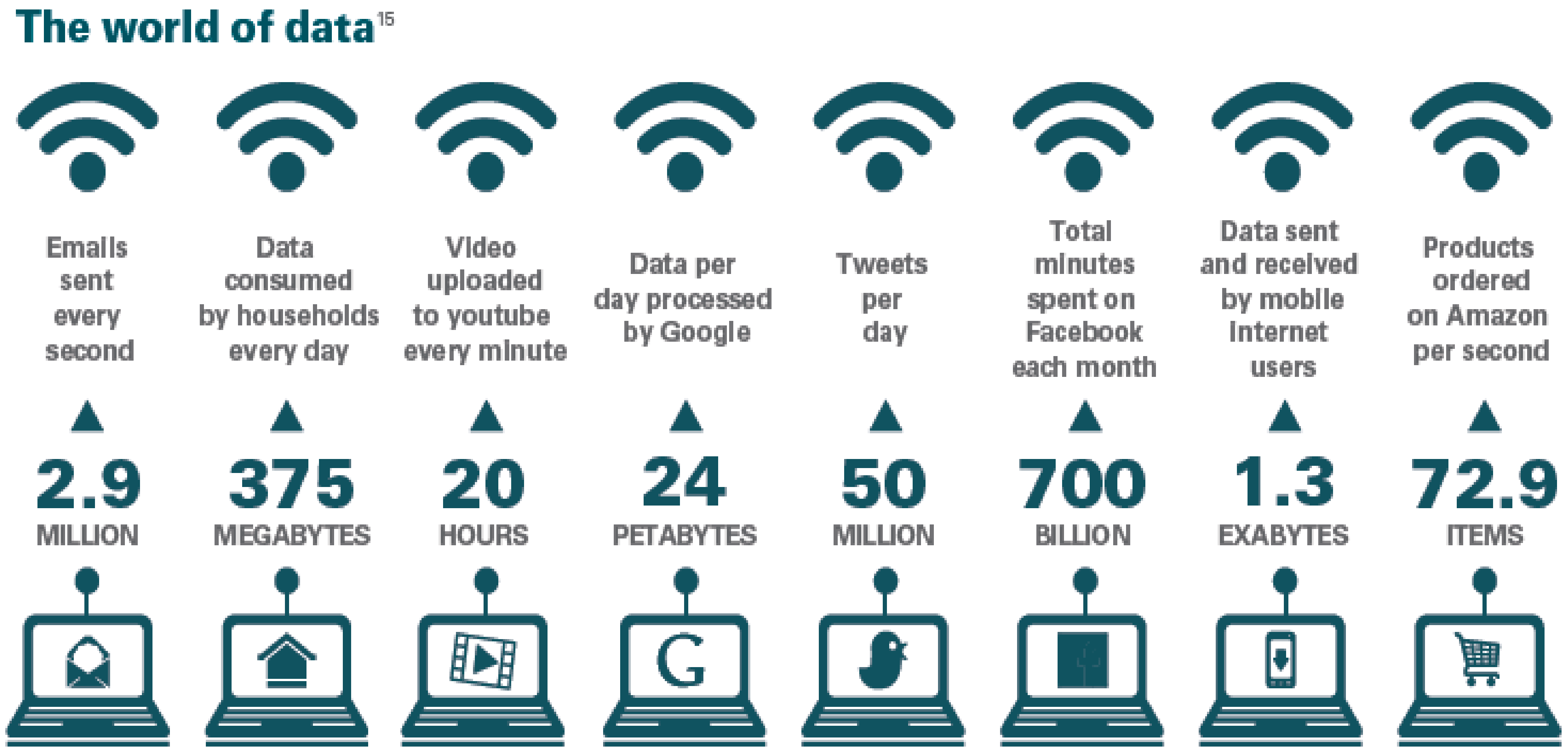


Tema Umum: Data

Jumlah data hanya terus tumbuh…
1.2 Zettabytes (10^{21} B atau lebih 1 Milyar TB)



Kita Hidup dalam Dunia Data…

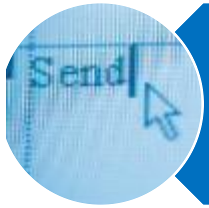


The world of data[15]

Emails sent every second	2.9	MILLION
Data consumed by households every day	375	MEGABYTES
Video uploaded to youtube every minute	20	HOURS
Data per day processed by Google	24	PETABYTES
Tweets per day	50	MILLION
Total minutes spent on Facebook each month	700	BILLION
Data sent and received by mobile Internet users	1.3	EXABYTES
Products ordered on Amazon per second	72.9	ITEMS

Apa yang Kita Lakukan Dengan Data?

Store



Share

Access

Process

Encrypt

... dan lainnya!

Kita ingin laksanakan ini dengan lancar...

Menggunakan Beragam Antarmuka & Perangkat

Computers

Mobile Devices

Consumer Electronics

Personal Monitors & Sensors

...dan bahkan *appliances*

Kita juga ingin mengakses, berbagi (*share*) dan memproses data kita dari semua perangkat, **kapan saja**, **dimana saja**!

Data Menjadi Sangat Penting untuk Hidup Kita

Domain
Data

- Health
- Education
- Environment
- Science
- Work
- Finance

... dan selanjutnya

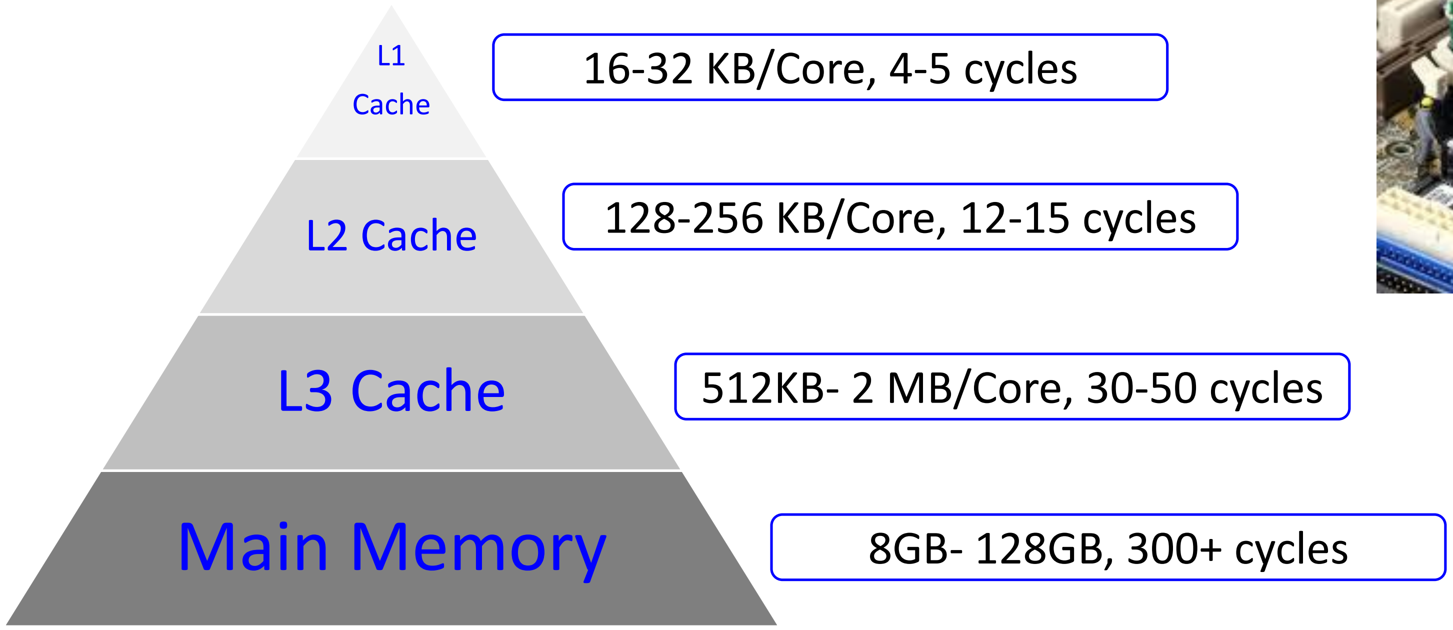
Menyimpan & Memroses Data *Scalable*?

- Suatu sistem dapat diskalakan: ✓
  - Pertama, secara *vertical* (atau up)
    - Dapat dicapai dengan upgrade hardware (misalnya CPU yang lebih cepat, memory lebih besar, disk lebih besar)
  - Kedua secara *horizontal* (atau *out*)
    - Dapat diperoleh dengan penambahan beberapa mesin

Vertical Scaling

- Peringatan: Komputer individu masih dapat menderita **keterbatasan sumber daya** sehubungan dengan skala masalah saat ini

1. Caches dan Memory:

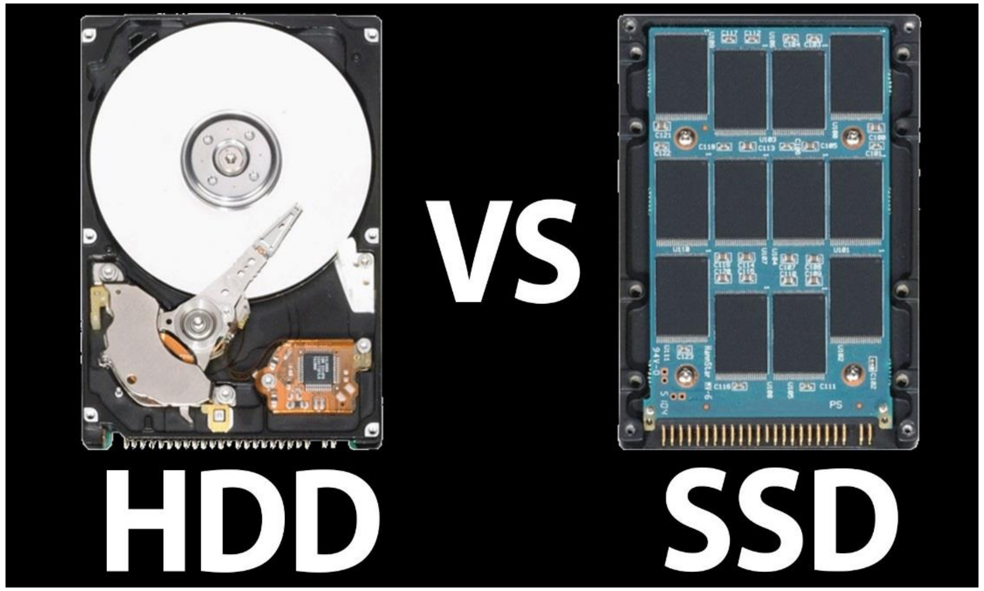

L1 Cache
16-32 KB/Core, 4-5 cycles
L2 Cache
128-256 KB/Core, 12-15 cycles
L3 Cache
512KB- 2 MB/Core, 30-50 cycles
Main Memory
8GB- 128GB, 300+ cycles

Vertical Scaling

- Peringatan: Komputer individu masih dapat menderita **keterbatasan sumber daya** sehubungan dengan skala masalah saat ini

2. Disks: beberapa kemajuan, tapi tetap saja:

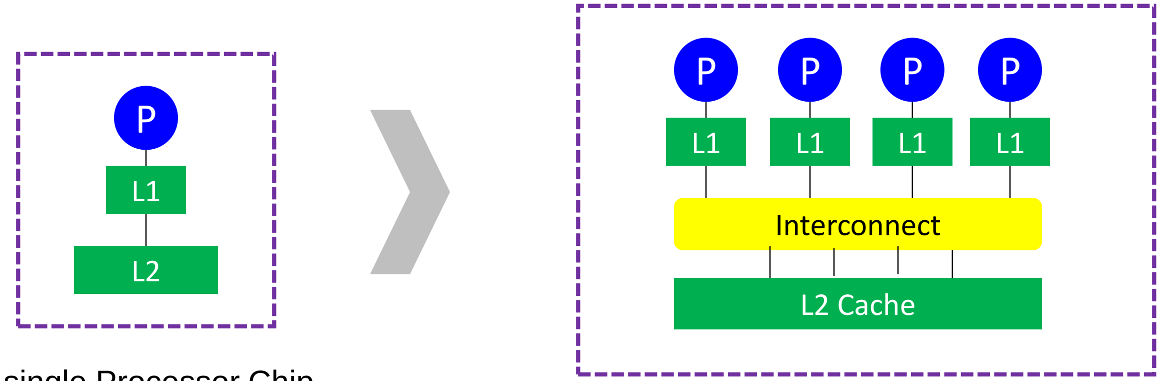
   - Kapasitas terbatas
   - Jumlah jalur terbatas
   - Bandwidth terbatas.



Vertical Scaling

- Peringatan: Komputer individu masih dapat menderita **keterbatasan sumber daya** sehubungan dengan skala masalah saat ini

  3. Processor (CPU):
     - Hukum Moore masih berlaku
     - Chip Multiprocessors (CMP) sudah tersedia



A single Processor Chip

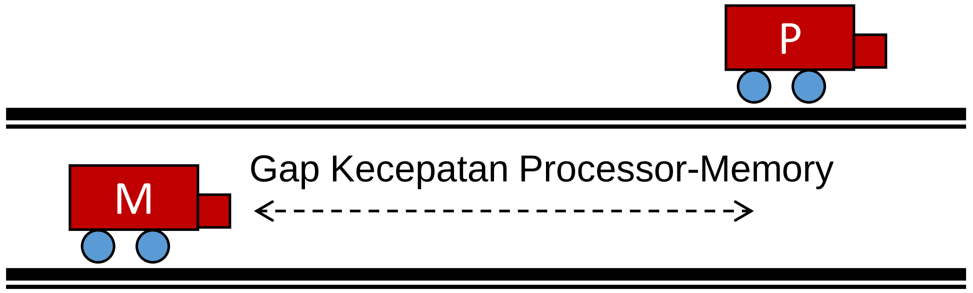
A CMP

Vertical Scaling

- Peringatan: Komputer individu masih dapat menderita **keterbatasan sumber daya** sehubungan dengan skala masalah saat ini

  3. Processor (CPU):

     - Tetapi sampai dengan beberapa tahun lalu, kecepatan CPU tumbuh 55% pertahun, sedangkan kecepatan memory tumbuh 7%.



Vertical Scaling

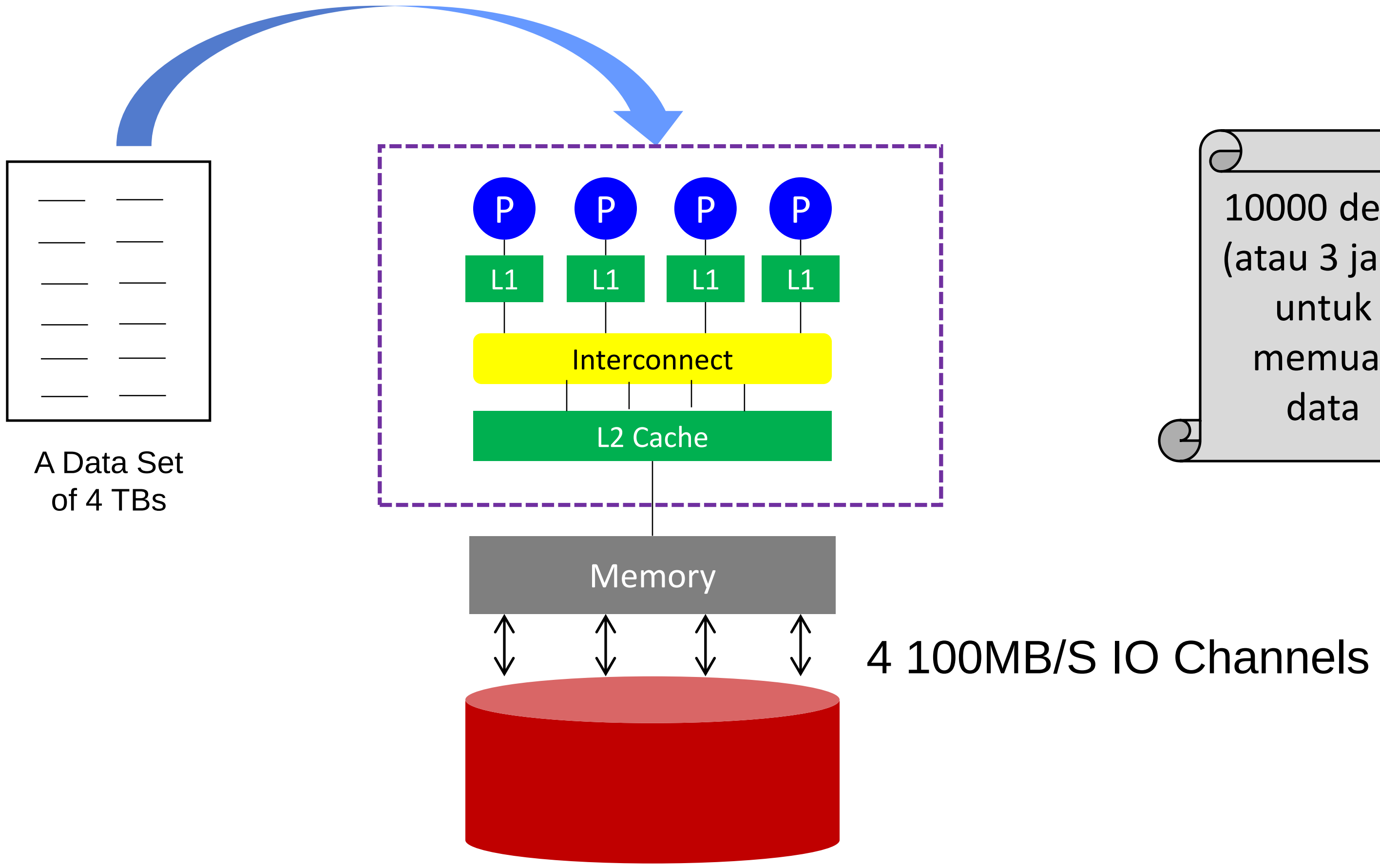
- Peringatan: Komputer individu masih dapat menderita **keterbatasan sumber daya** sehubungan dengan skala masalah saat ini

  3. Processors:

     - Tetapi sampai dengan beberapa tahun lalu, kecepatan CPU tumbuh 55% pertahun, sedangkan kecepatan memory tumbuh 7%.

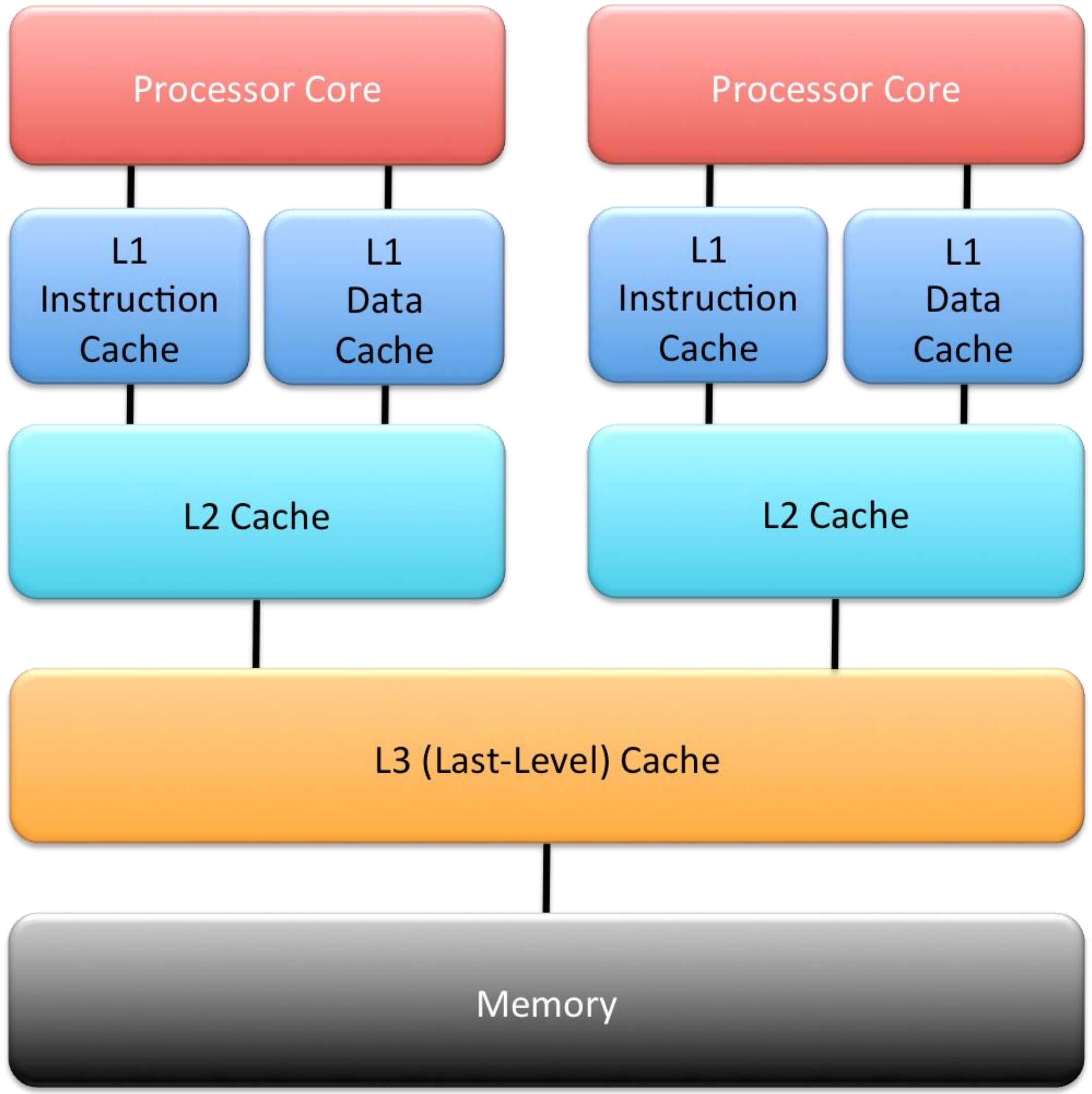
     - Bahkan jika ratusan atau ribuan core diletakan pada suatu CMP, merupakan tantangan untuk men-deliver data input ke core-core ini cukup cepat untuk pemrosesan.

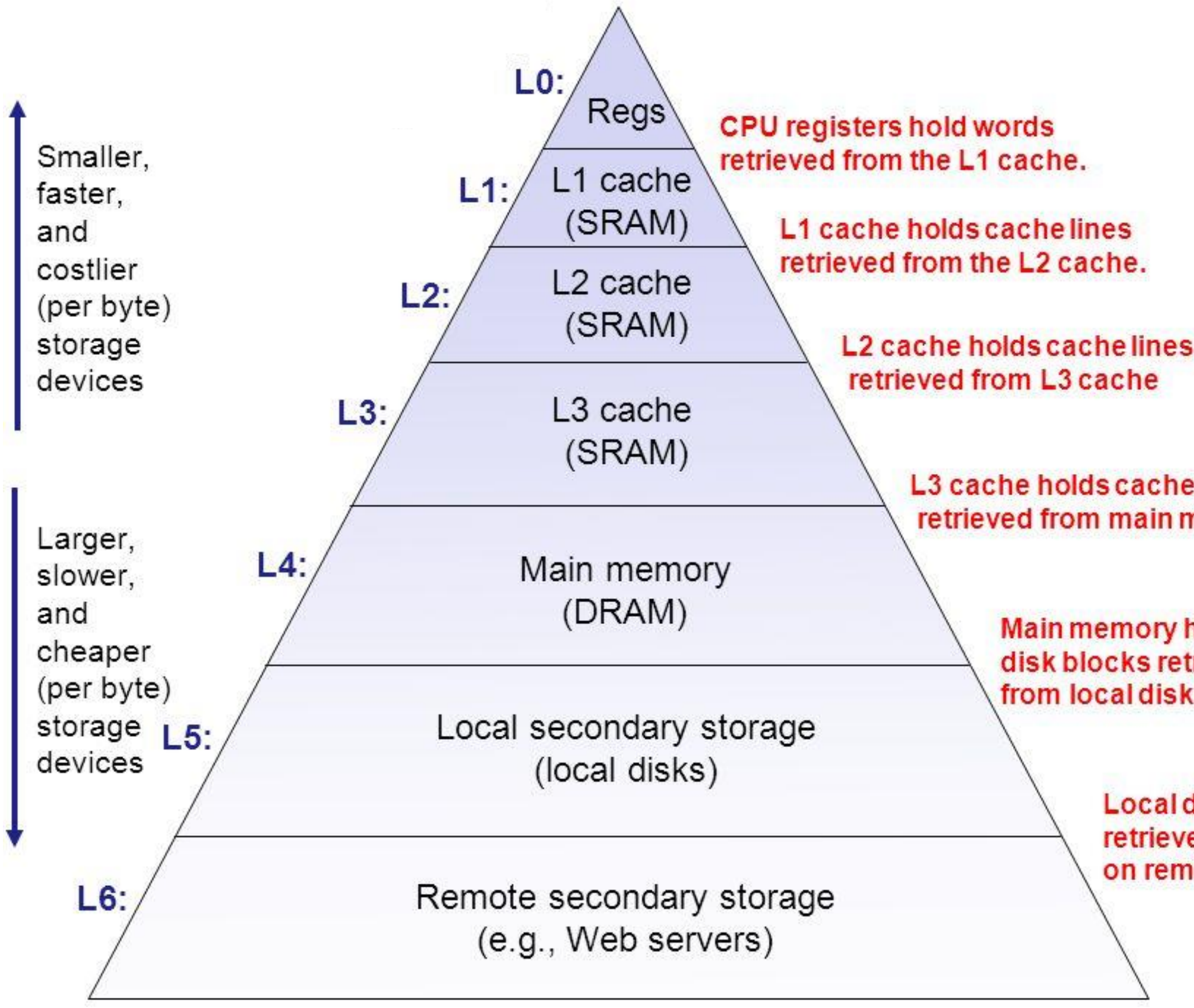
Vertical Scaling

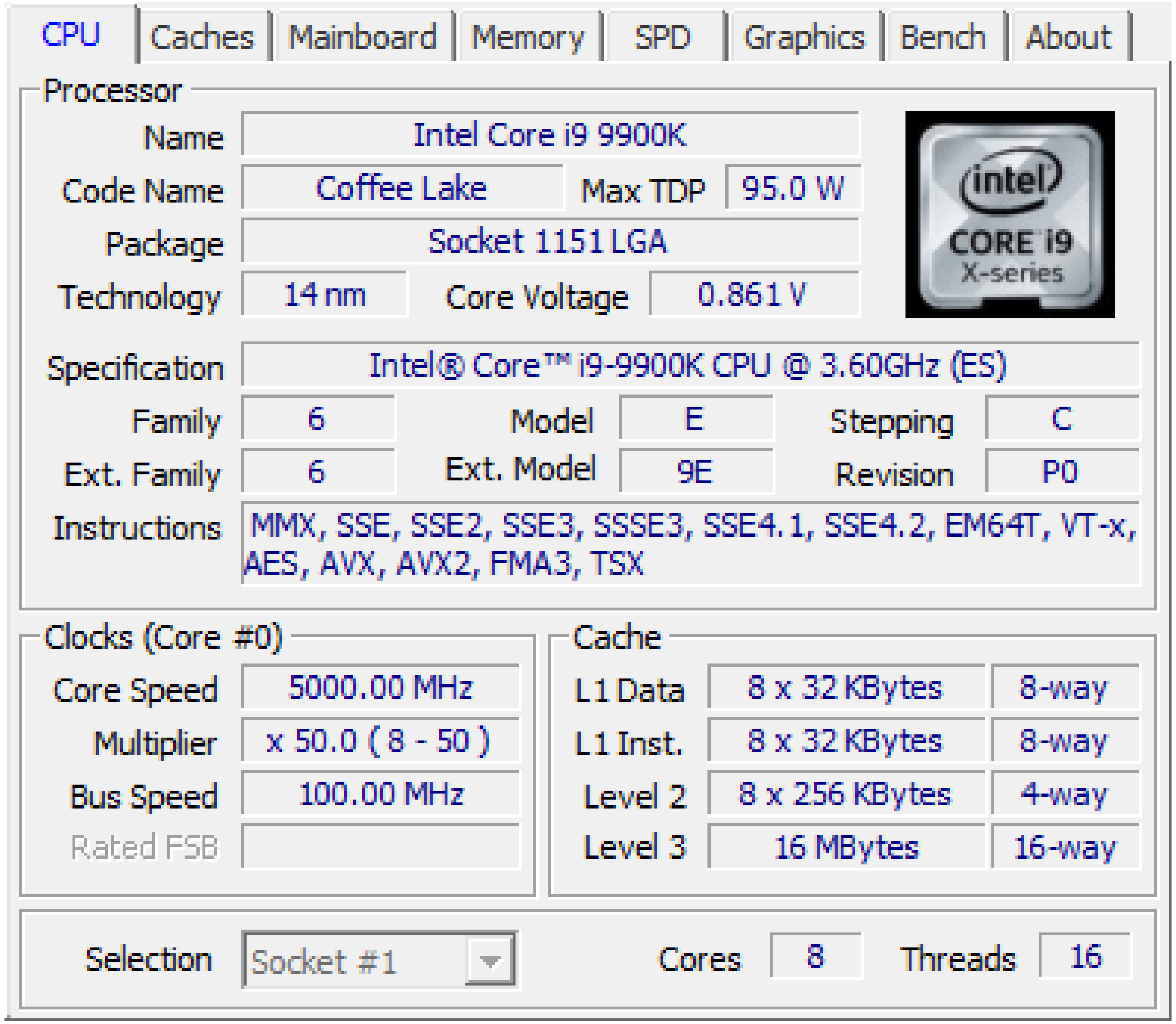


10000 detik
(atau 3 jam)
untuk
memuat
data

Arsitektur Processor Modern




Smaller, faster, and costlier (per byte) storage devices
Larger, slower, and cheaper (per byte) storage devices
L0:
Regs
L1:
L1 cache (SRAM)
L2:
L2 cache (SRAM)
L3:
L3 cache (SRAM)
L4:
Main memory (DRAM)
L5:
Local secondary storage (local disks)
L6:
Remote secondary storage (e.g., Web servers)
CPU registers hold words retrieved from the L1 cache.
L1 cache holds cache lines retrieved from the L2 cache.
L2 cache holds cache lines retrieved from L3 cache
L3 cache holds cache retrieved from main m
Main memory h disk blocks ret from local disk
Local d retrieve on rem



Processor

Name	Intel Core i9 9900K				
Code Name	Coffee Lake	Max TDP	95.0 W		
Package	Socket 1151 LGA				
Technology	14 nm	Core Voltage	0.861 V		
Specification	Intel® Core™ i9-9900K CPU @ 3.60GHz (ES)				
Family	6	Model	E	Stepping	C
Ext. Family	6	Ext. Model	9E	Revision	P0
Instructions	MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, EM64T, VT-x, AES, AVX, AVX2, FMA3, TSX				

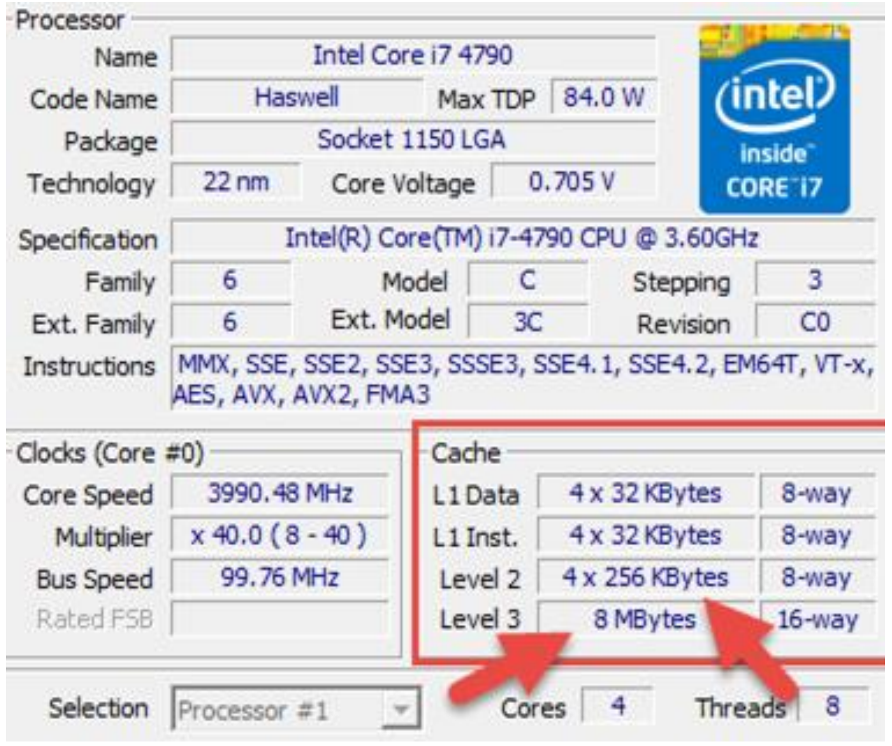
Clocks (Core #0)

Core Speed	5000.00 MHz
Multiplier	x 50.0 (8 - 50)
Bus Speed	100.00 MHz
Rated FSB	

Cache

L1 Data	8 x 32 KBytes	8-way
L1 Inst.	8 x 32 KBytes	8-way
Level 2	8 x 256 KBytes	4-way
Level 3	16 MBytes	16-way

Selection Socket #1 Cores 8 Threads 16



Processor

Name	Intel Core i7 4790				
Code Name	Haswell	Max TDP	84.0 W		
Package	Socket 1150 LGA				
Technology	22 nm	Core Voltage	0.705 V		
Specification	Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz				
Family	6	Model	C	Stepping	3
Ext. Family	6	Ext. Model	3C	Revision	C0
Instructions	MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, EM64T, VT-x, AES, AVX, AVX2, FMA3				

Clocks (Core #0)

Core Speed	3990.48 MHz
Multiplier	x 40.0 (8 - 40)
Bus Speed	99.76 MHz
Rated FSB	

Cache

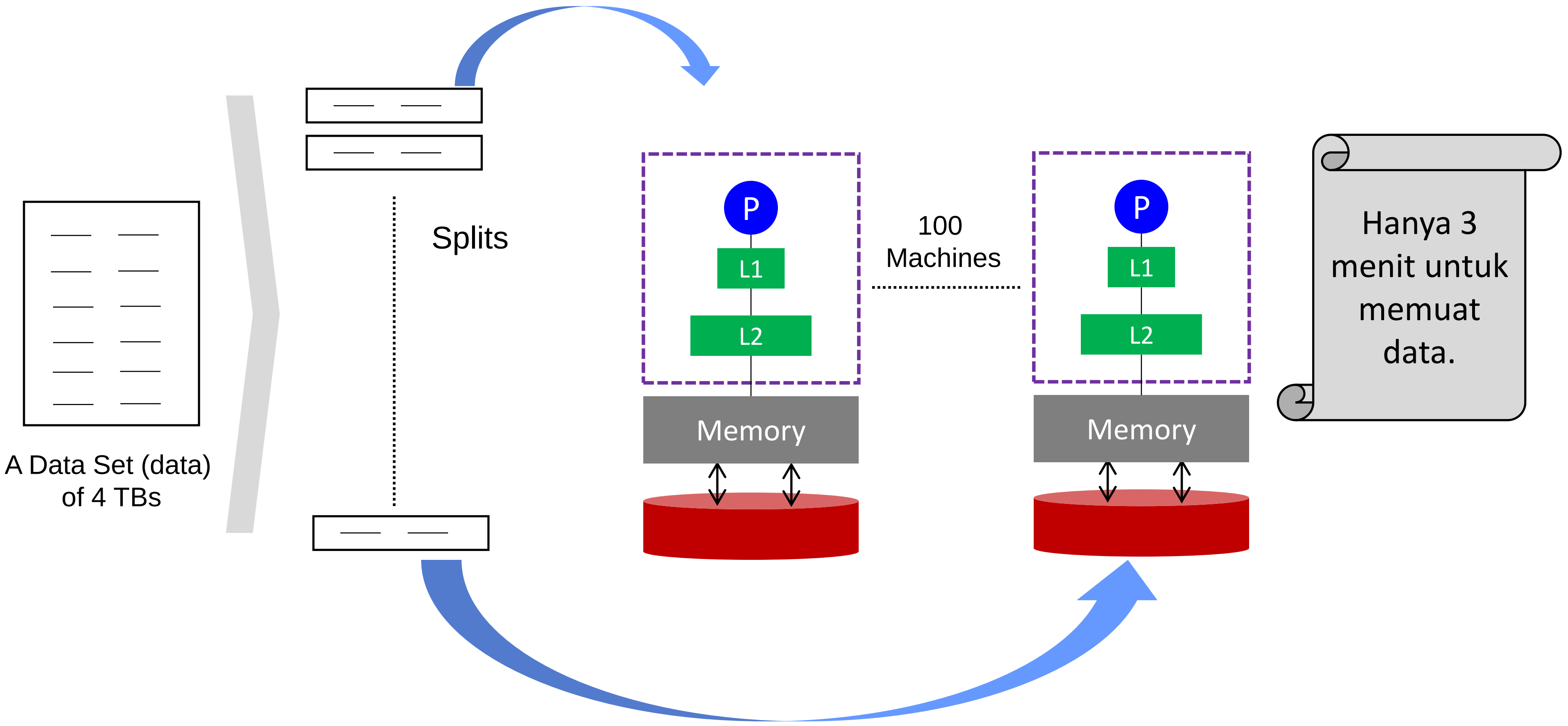
L1 Data	4 x 32 KBytes	8-way
L1 Inst.	4 x 32 KBytes	8-way
Level 2	4 x 256 KBytes	8-way
Level 3	8 MBytes	16-way

Selection Processor #1 Cores 4 Threads 8

Bagaimana Menyimpan dan Memroses Data pada Skala?

- Suatu sistem dapat diskalakan dengan:
  - Pertama, secara *vertical* (atau up)
    - Dapat dicapai dengan upgrade hardware (misal CPU yang lebih cepat, memory dan disk yang lebih besar)
  - Kedua, secara *horizontal* (atau *out*) ✓
    - Dapat diperoleh dengan penambahan beberapa mesin

Horizontal Scaling



Requirements

- Tapi, ini mengharuskan:
  - Cara mengekspresikan masalah sebagai proses paralel dan mengeksekusinya pada mesin-mesin berbeda (*Programming and Concurrency Models*)
  - Cara mengorganisasi proses-proses (*Architectures*)
  - Cara bagi proses-proses terdistribusi untuk bertukar informasi (*Communication Paradigms*)
  - Cara mencari, menempatkan dan berbagi sumber daya (*Naming Protocols*)
  - Cara bagi proses-proses terdistribusi bekerjasama dan saling men-sinkron-kan, dan menyepakati nilai-nilai yang dishare (*Synchronization*).

Requirements

- Tapi, ini mengharuskan:
  - Cara untuk mengurangi *latency*, meningkatkan *reliability*, dan memperbaiki kinerja (*Caching, Replication, dan Consistency*)
  - Cara untuk meningkatkan skalabilitas beban, mengurangi keragaman lintas sistem yang heterogen, dan menyediakan suatu derajat tinggi dari portabilitas dan fleksibilitas (*Virtualization*)
  - Cara untuk sembuh dari kegagalan persial (*Fault Tolerance*)
  - Cara mengintegrasikan data dari beberapa sumber berbeda dengan menggunakan *web service*.

Jadi, Apa Itu Sistem Terdistribusi?

Sistem Terdistribusi adalah

Kumpulan komputer independen yang nampak pada penggunanya sebagai satu sistem koheren

Sistem yang komponennya terletak di dalam jaringan komputer berkomunikasi dan mengkoordinasikan tindakannya hanya dengan menyampaikan pesan

Fitur-fitur

- Sistem Terdistribusi membawa empat fitur utama:

1. Pemisahan Geografis
2. Tidak ada jam fisik bersama
3. Tidak ada memory fisik bersama
4. Otonomi dan heterogenitas

Sistem Terdistribusi vs. Paralel

- Sistem terdistribusi berbeda dari sistem paralel, yang mengharuskan:

1. Kopling (sambungan) kuat
2. Jam fisik bersama
3. Memory fisik bersama
4. Homogenitas

Matakuliah

TIK-604 Sistem Terdistribusi

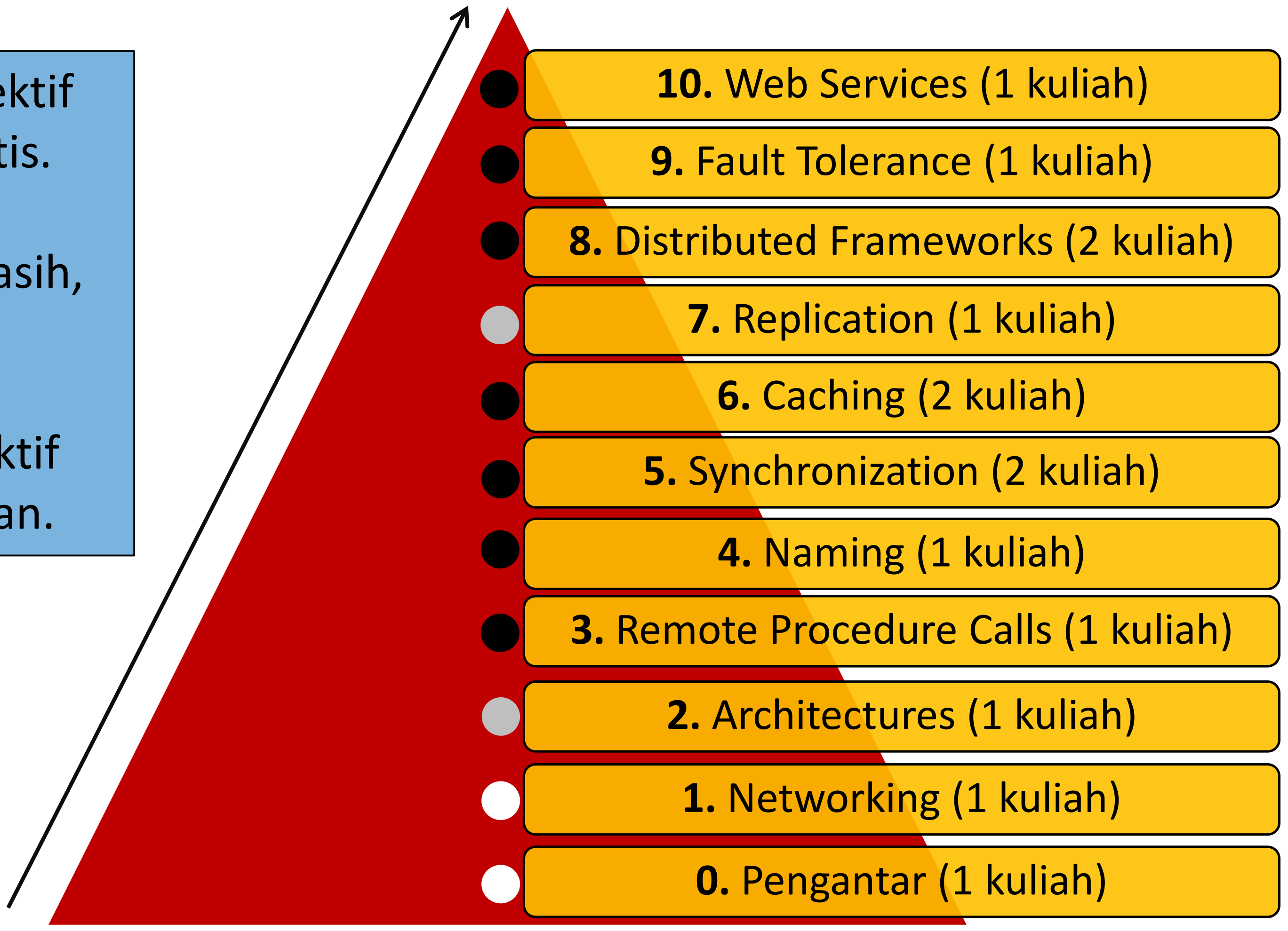
Semester Genap, 2018/2019

Pengantar Sistem Terdistribusi

○ **Prasyarat*:*** suatu perspektif komprehensif & cukup kritis.

◍ **Pemikiran*:*** Perspektif fasih, fleksibel dan efisien.

● **Harus dikuasai*:*** perspektif yang kuat dan mencerahkan.



● **10.** Web Services (1 kuliah)

● **9.** Fault Tolerance (1 kuliah)

● **8.** Distributed Frameworks (2 kuliah)

◍ **7.** Replication (1 kuliah)

● **6.** Caching (2 kuliah)

● **5.** Synchronization (2 kuliah)

● **4.** Naming (1 kuliah)

● **3.** Remote Procedure Calls (1 kuliah)

◍ **2.** Architectures (1 kuliah)

○ **1.** Networking (1 kuliah)

○ **0.** Pengantar (1 kuliah)

Tujuan Kuliah

Memberikan pemahaman komprehensif melalui studi kasus...

- Prinsip yang mendasari sistem terdistribusi
- Prinsip untuk mengoptimalkan sistem terdistribusi
- Model pemrograman dan mesin analitis sistem terdistribusi
- Bagaimana sistem terdistribusi modern memenuhi kebutuhan aplikasi terdistribusi kontemporer

Capaian Pembelajaran

- Memahami konsep inti dari sistem terdistribusi: Bagaimana beberapa mesin dapat digunakan untuk memecahkan masalah yang kompleks secara efisien, andal dan terukur (*scalable*).
- Menjelaskan bagaimana cara kerja dan kinerja dari sistem yang telah menerapkan konsep inti dari sistem terdistribusi
  - Menerapkan konsep semacam itu dalam merancang suatu aplikasi terdistribusi (sistem contoh).

Tim Pengajaran

Dosen: Husni
Asisten: ?
Tim Pengajaran TIK-604

Husni Jam Kerja

- Senin s.d Jumat, 07:30- 13:45
- Laboratorium Riset Informatika bidang Sistem Terdistribusi
- Silakan saat pintu terbuka atau dengan perjanjian.

Asisten? Jam Kerja

- ???

Metode Pengajaran

14 Kuliah (tatap muda di kelas)

- Memotivasi pembelajaran (*learning*)
- Memberikan framework atau roadmap untuk mengorganisasi informasi mengenai kuliah ini
- Menjelaskan subjek dan perkuat gagasan besar yang penting

Bimbingan dan Arahan

- Meminta mahasiswa mengungkapkan apa yang belum dimengerti, sehingga Dosen dapat membantunya
- Mempersilakan mahasiswa mempraktikkan keterampilan yang diperlukan untuk menguasai penerapannya

Tugas dan Proyek

Tugas Personal

- Khusus semester ini, tidak ada penugasan personal

Proyek

- Hanya ada dua proyek kelompok:
    - Pembuatan Rancangan Perangkat Lunak Terdistribusi
    - Mengkaji aplikasi terdistribusi

Proyek

- Untuk semua proyek, mohon ikuti aturan berikut:
  - Jika anda men-submit terlambat sehari, nilai akhir proyek akan dikurangi 25%
  - Jika terlambatnya dua hari, nilai akhir proyek berkurang 50%
  - Proyek tidak akan dinilai (dan otomatis mendapatkan nilai nol) jika terlambat lebih 2 hari.

Tugas Proyek: Kelompok

- Buat kelompok: 3 s.d 4 mahasiswa
- Buat rancangan aplikasi terdistribusi “sederhana”, ada komunikasi client-server (server, client $\geq$ 2), ada database untuk menyimpan transaksi
- Deadline: Rabu, 15 Mei 2019 jam 23:59:59
- Poin penilaian:
  - Makalah dan Presentasi
  - Rangkuman dan Presentasi

Metode Penilaian

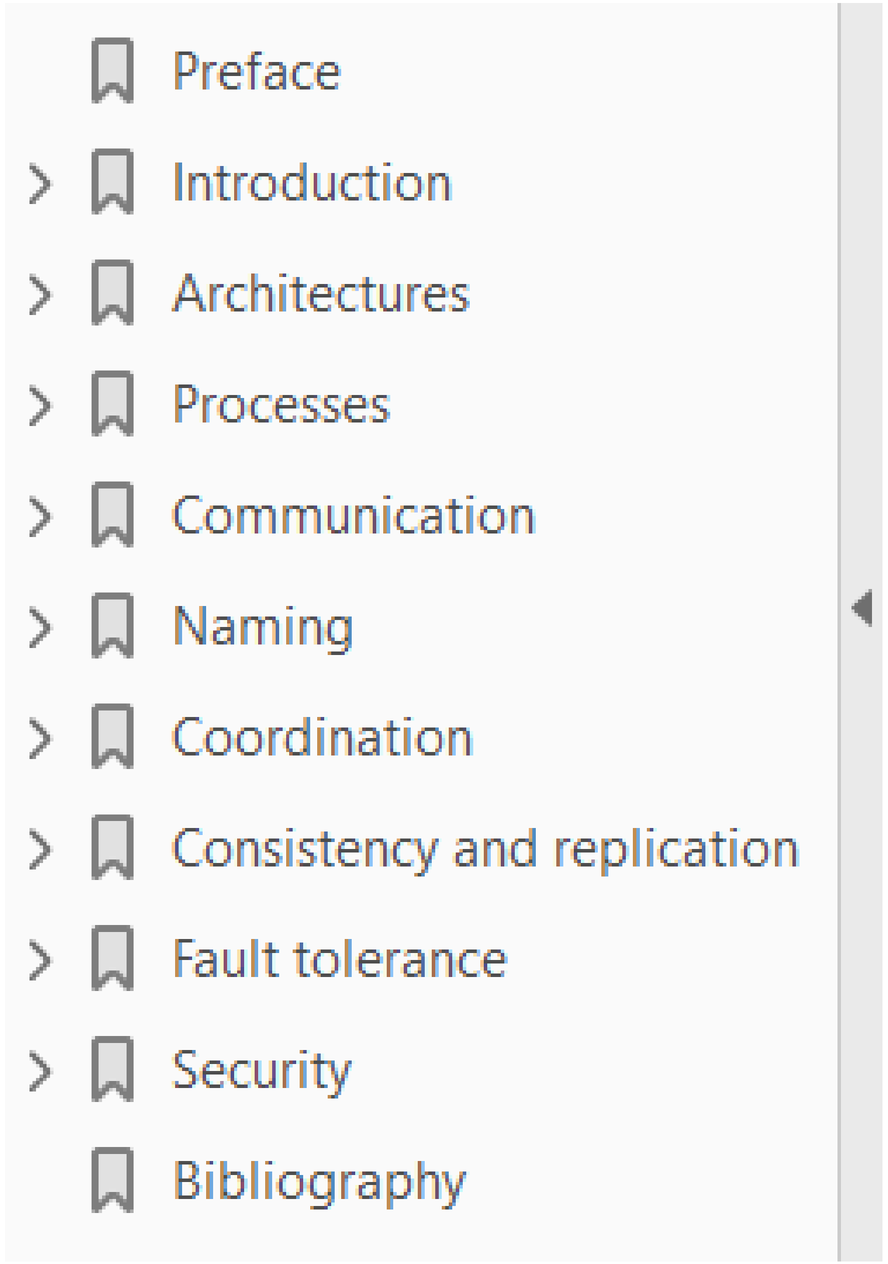
- Bagaimana proses pembelajaran diukur?

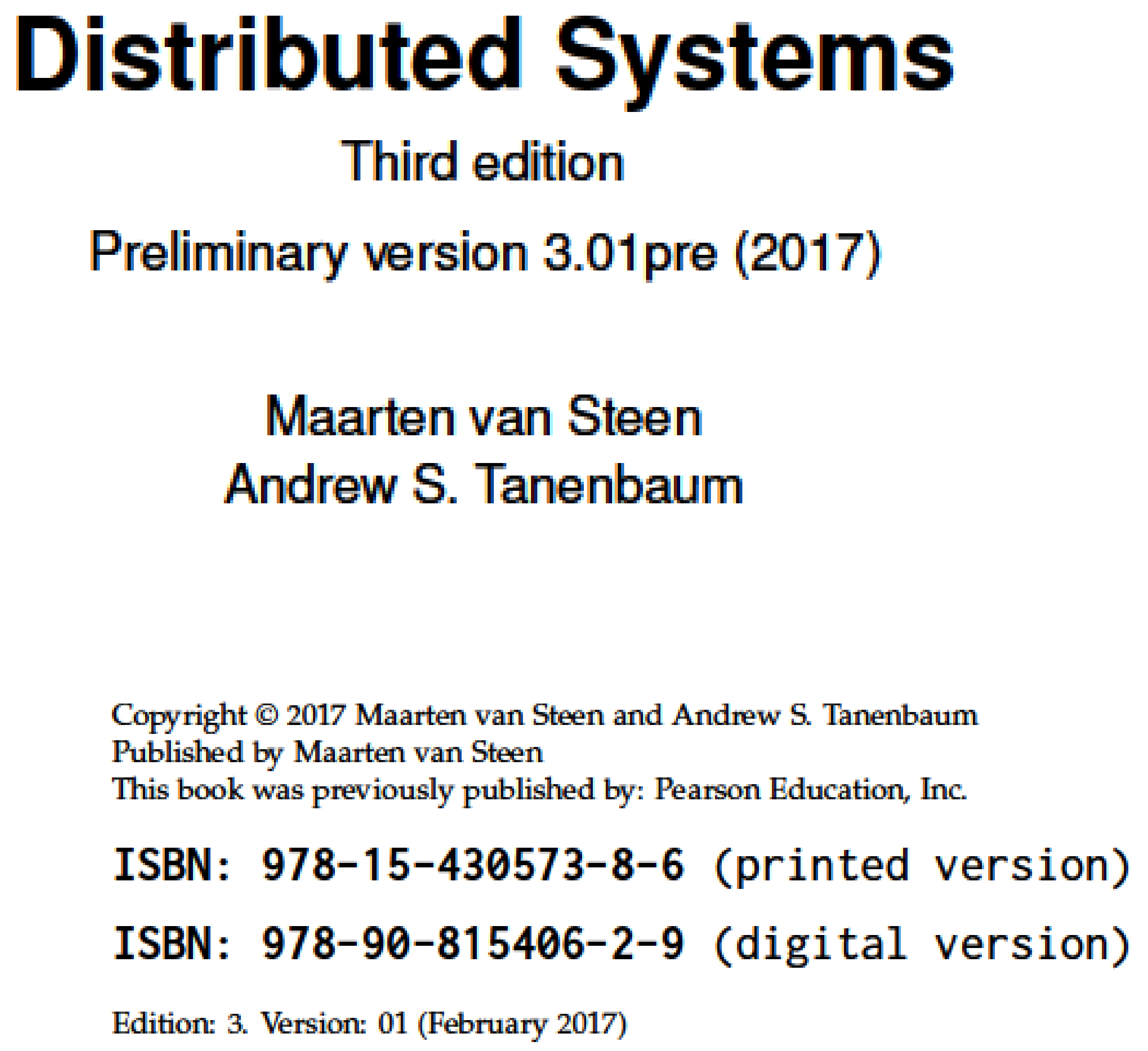
Tipe	#	Bobot
Proyek	2	60% (40% + 20%)
Ujian (UTS, UAS)	2	40% (20% + 20%)
Tugas Personal	0	0
Kuis/Tugas Khusus	0	0
Kehadiran & Keaktifan Diskusi	$\geq$ 80%	10%

Buku Teks

- Andrew S. Tannenbaum dan Maarten Van Steen (2007) ***Distributed Systems: Principles and Paradigms***, Edisi ke-2 atau ke-3 (2017).
  - 596 halaman
- George Coulouris, Jean Dollimore, Tim Kindberg, dan Gordon Blair (2012) ***Distributed Systems: Concepts and Design***, 5th Edition, Addison Wesley.
  - 1067 halaman

Buku pegangan utama Dosen, Mahasiswa dan Peneliti Sistem Terdistribusi, sejak 1990-an.





Distributed Systems

Third edition

Preliminary version 3.01pre (2017)

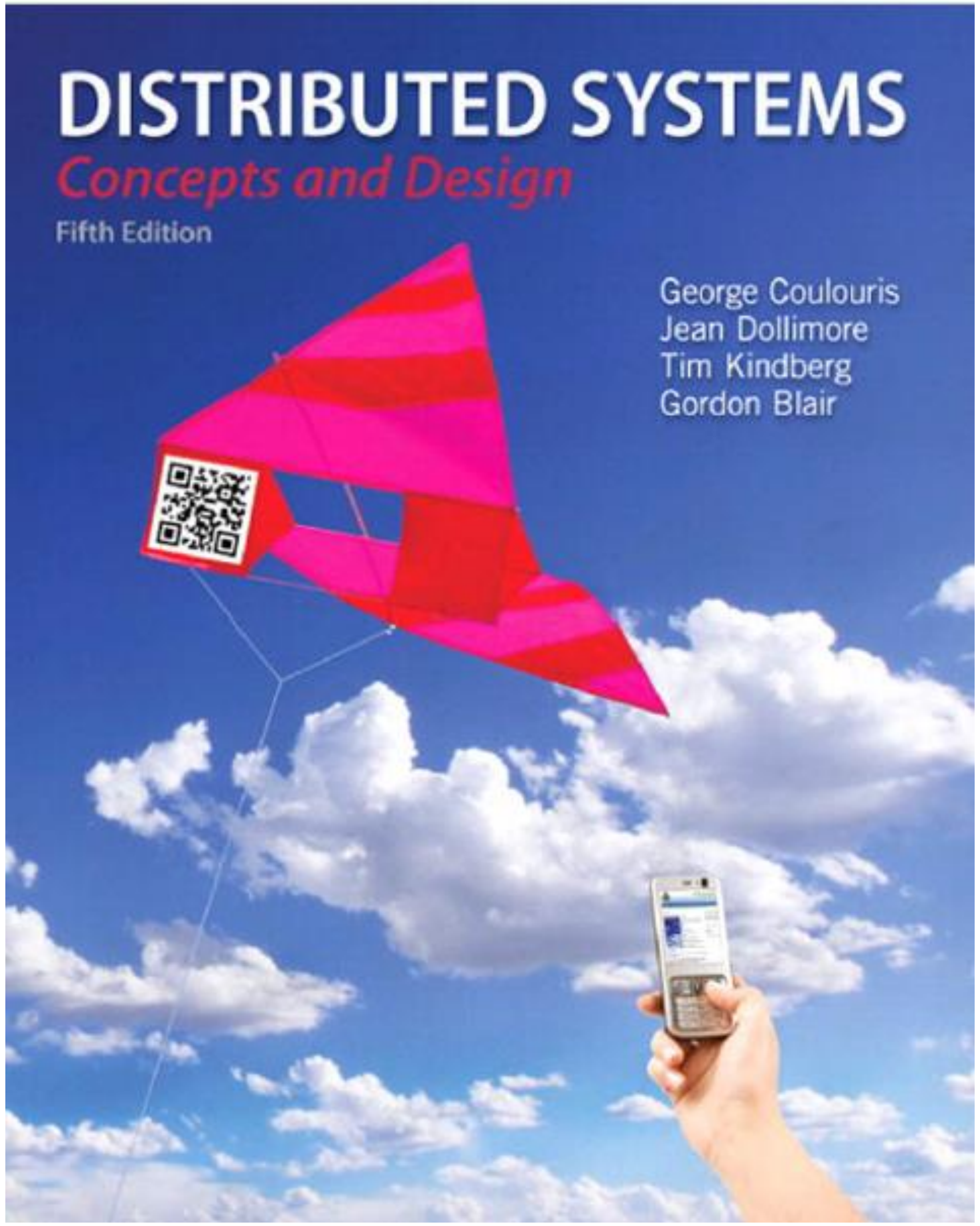
Maarten van Steen
Andrew S. Tanenbaum

Copyright © 2017 Maarten van Steen and Andrew S. Tanenbaum
Published by Maarten van Steen
This book was previously published by: Pearson Education, Inc.

ISBN: 978-15-430573-8-6 (printed version)

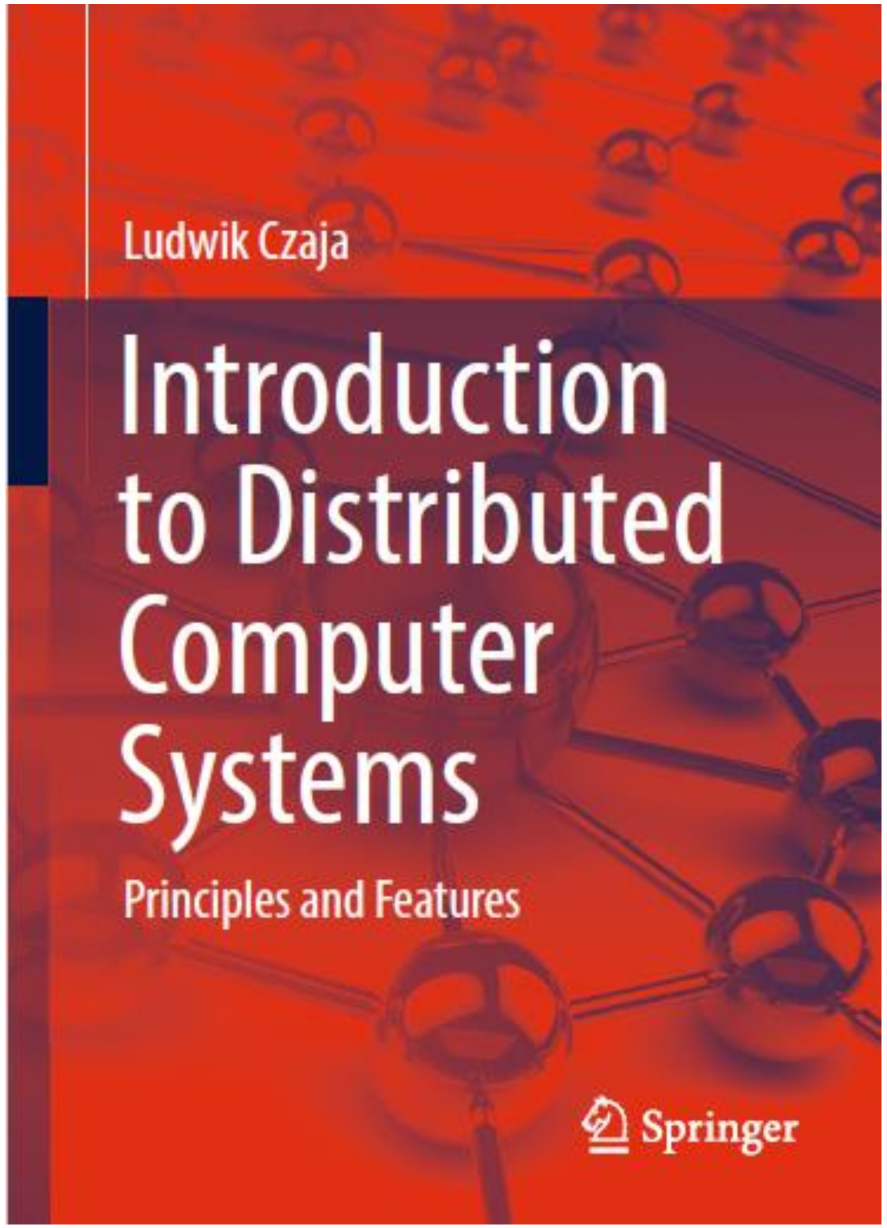
ISBN: 978-90-815406-2-9 (digital version)

Edition: 3. Version: 01 (February 2017)

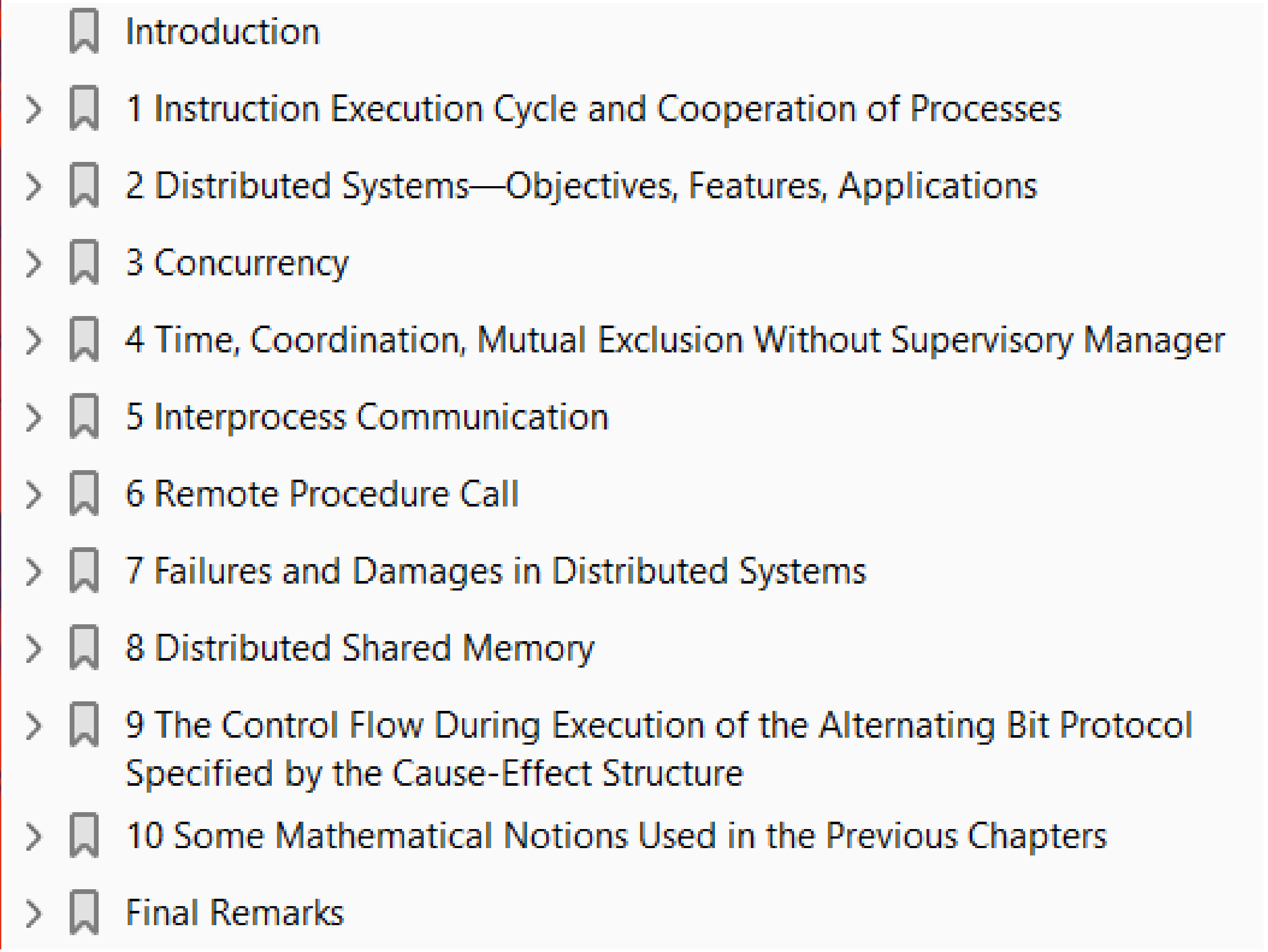

DISTRIBUTED SYSTEMS
Concepts and Design
Fifth Edition
George Coulouris
Jean Dollimore
Tim Kindberg
Gordon Blair

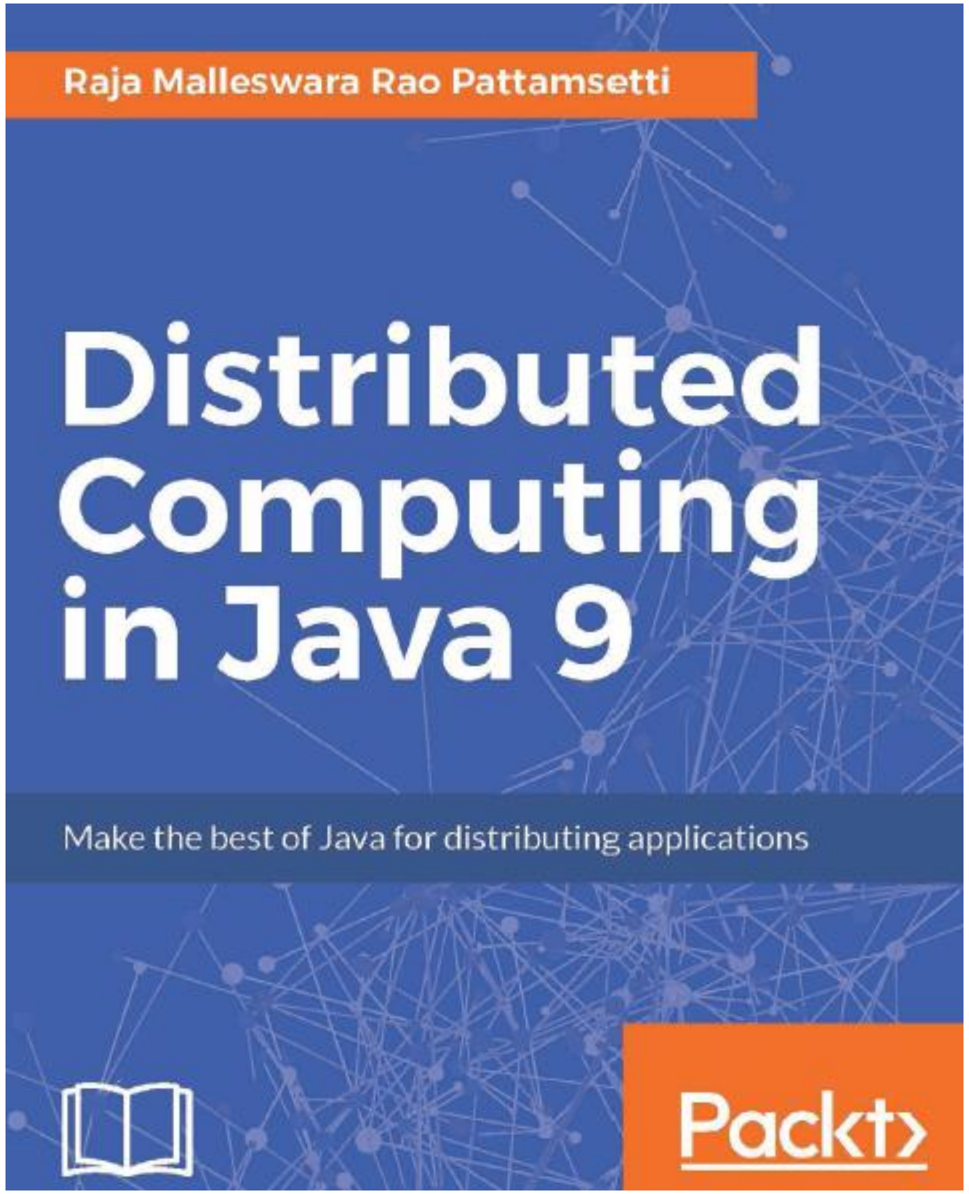
Buku Pelengkap

1. Ludwik Czaja (2018) ***Introduction to Distributed Computer Systems Principles and Features***, Springer International Publishing AG
2. Raja Malleswara dan Rao Pattamsetti (2017) ***Distributed Computing in Java 9***, Packt Publishing
3. Jan Graba (2013) ***An Introduction to Network Programming with Java, Java 7 Compatible***, 3rd Edition, Springer-Verlag London
4. Faruque Sarker dan Sam Washington (2015) ***Learning Python Network Programming***, Packt Publishing
5. Jobinesh Purushothaman (2015) ***RESTful Java Web Services***, *2nd Edition,* Packt Publishing
6. Eric Chou (2017) **Mastering Python Networking**,Packt Publishing



- 269 halaman, baru (2018)

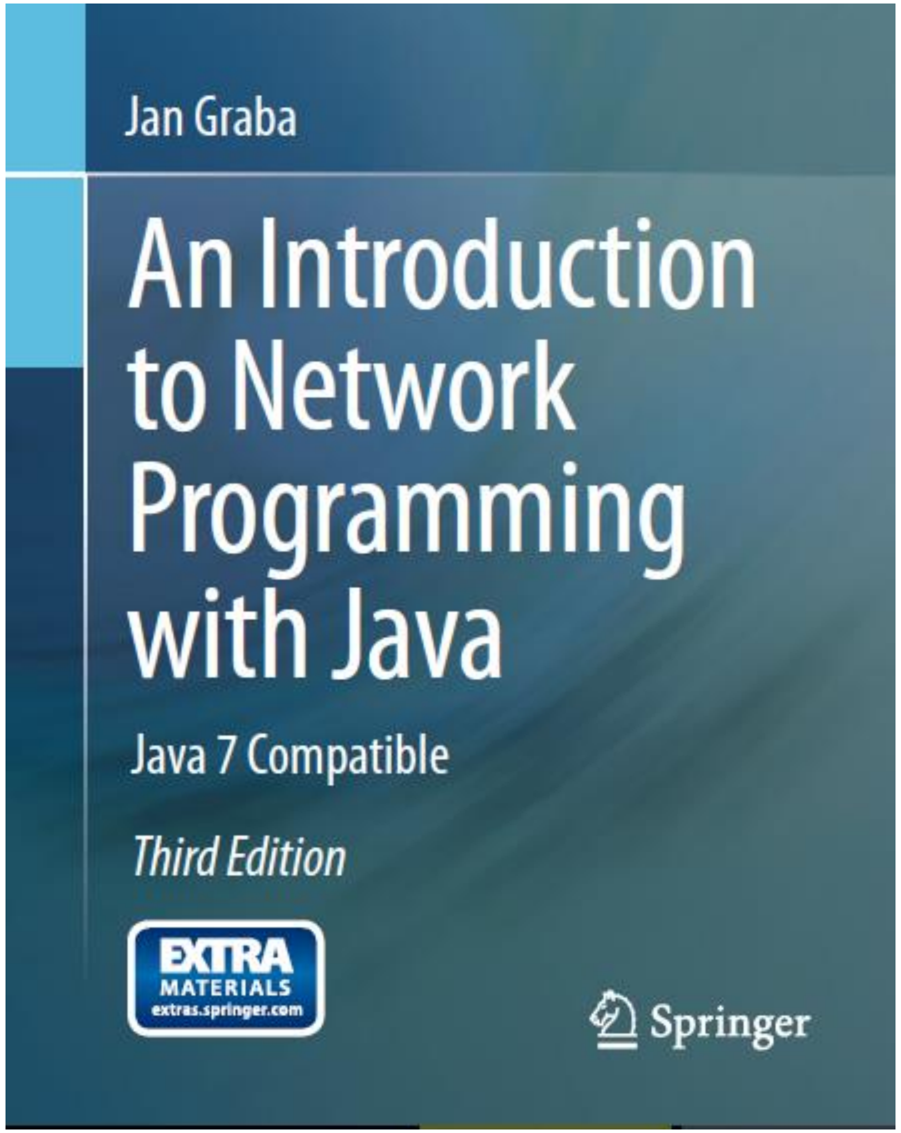


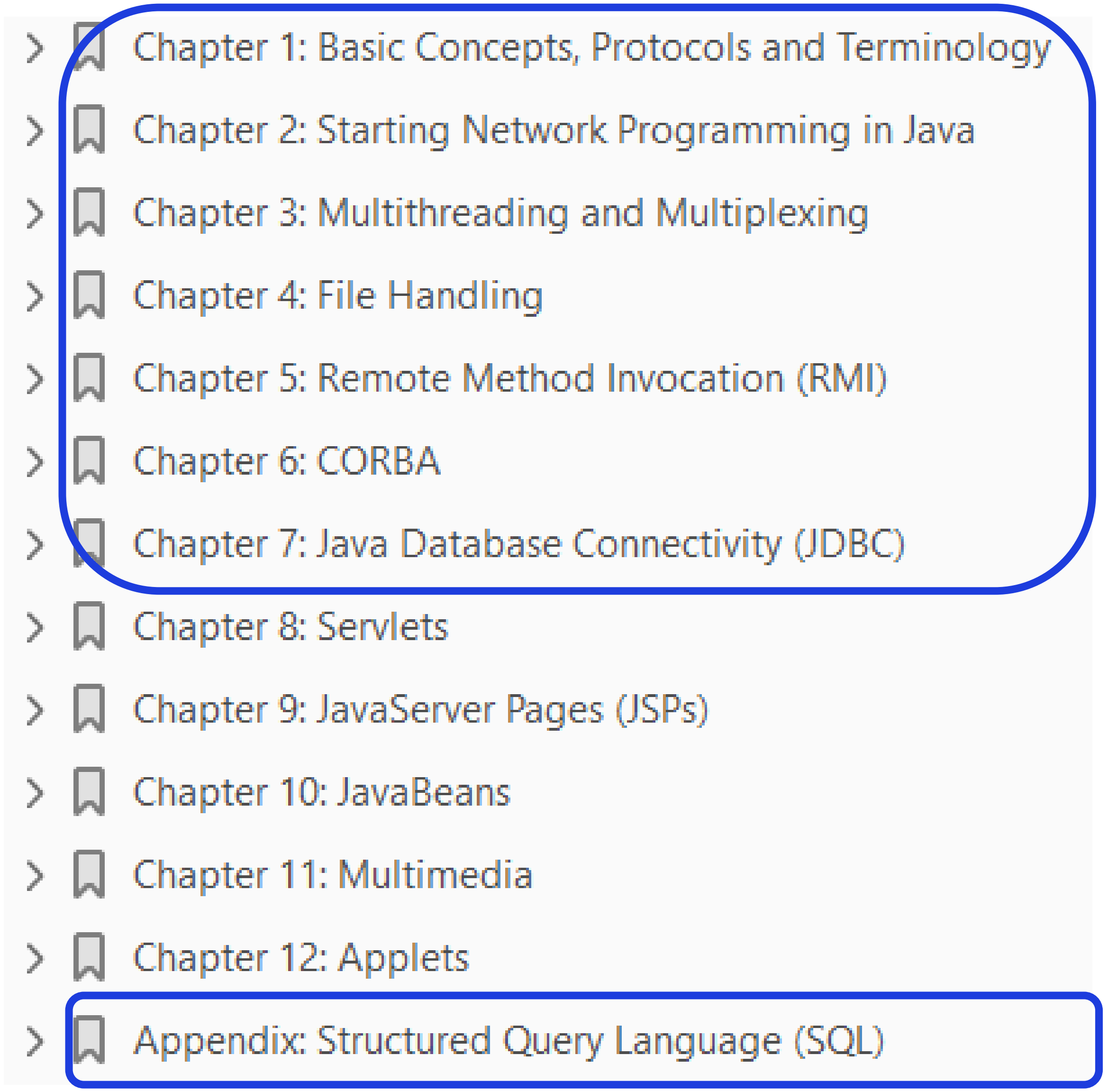

Raja Malleswara Rao Pattamsetti
Distributed Computing in Java 9
Make the best of Java for distributing applications
Packt>

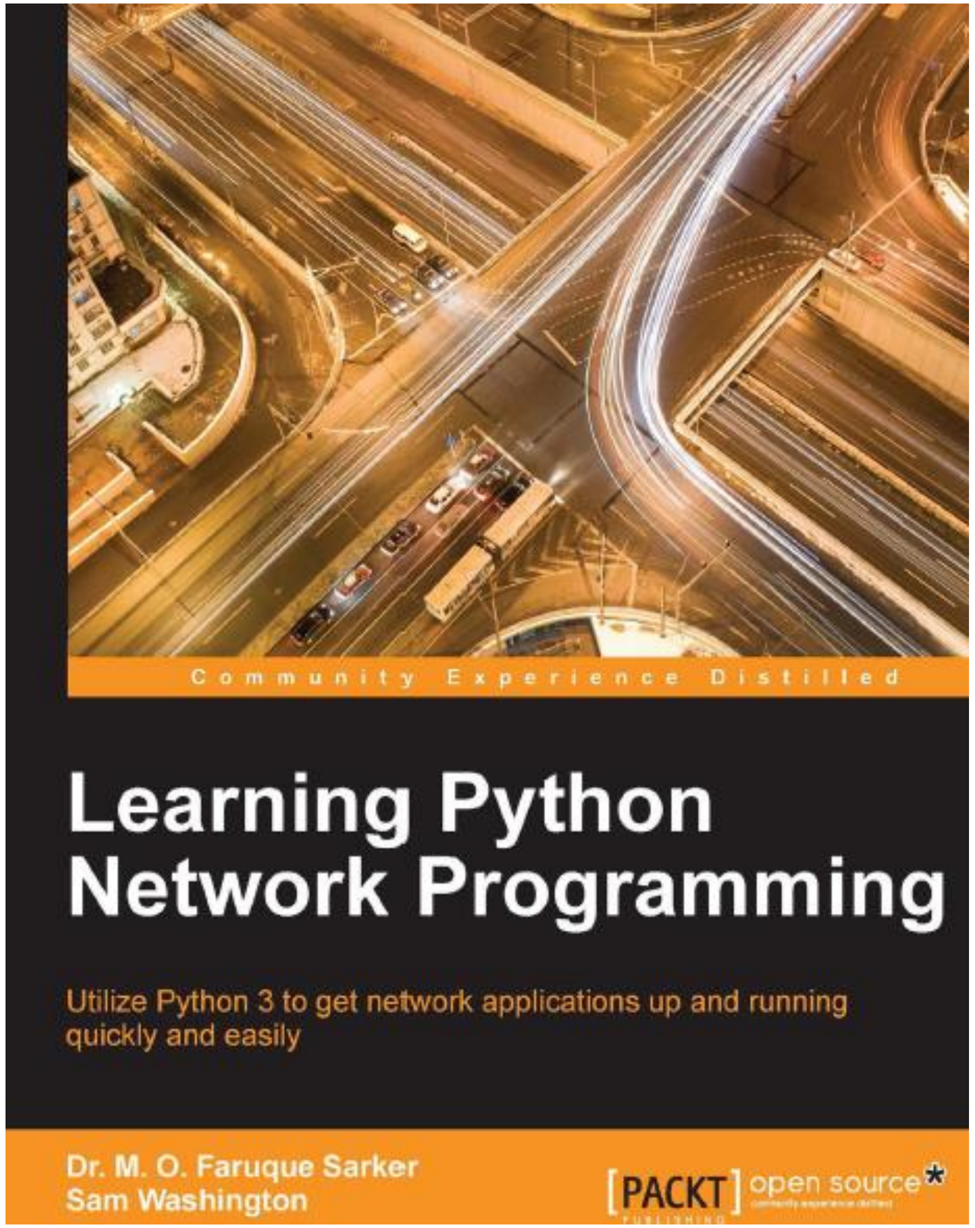
294 halaman

1. ***Quick Start to Distributed Computing***
2. ***Communication between Distributed Applications***
3. ***RMI, CORBA, and JavaSpace***
4. Enterprise Messaging
5. HPC Cluster Computing
6. **Distributed Databases**
7. Cloud and Distributed Computing
8. **Big Data Analytics**
9. Testing, Debugging, and Troubleshooting
10. Security

1. ***Quick Start to Distributed Computing***, reviews the basic concepts of distributed and parallel computing, what it is, why it is required, and how Java supports distributed computing, along with their architecture.
2. ***Communication between Distributed Applications***, covers different ways of communicating with remote systems in a distributed architecture, the concept of sockets, and stream programming with examples and URL-based communication.
3. ***RMI, CORBA, and JavaSpace***, teaches what the components of message-based systems are and how architectures such as RMI, CORBA, and Java Spaces complement it.
4. ***Enterprise Messaging***, explores the concept of enterprise integration patterns and the concepts of synchronous and asynchronous messaging, with technologies such as JMS and web services.
5. ***HPC Cluster Computing***, covers handling large amounts of data through parallel or distributed computing using HPC, cluster-computing architecture, and Java support for these implementations.
6. ***Distributed Databases***, covers the concepts and ways to set up a distributed database. It also explains how distributed databases help in performance improvisation with distributed transactions and XA transactions with examples.
7. ***Cloud and Distributed Computing***, explains how cloud and distributed computing go hand in hand. You will also learn the setup and procedure to configure your applications on market-leading cloud environments.
8. ***Big Data Analytics****,* discusses big data concepts and how big data helps in distributed computing. The chapter covers the implementations of big data, along with methods and applications, including Hadoop, MapReduce, and HDFS.
9. ***Testing, Debugging, and Troubleshooting***, explores how to test, debug, and troubleshoot distributed systems, and the different challenges in distributed computing.
10. ***Security***, discusses different security issues and constraints associated with distributed computing and how to address them.


Jan Graba
An Introduction to Network Programming with Java
Java 7 Compatible
Third Edition
EXTRA MATERIALS extras.springer.com
Springer




Community Experience Distilled
Learning Python Network Programming
Utilize Python 3 to get network applications up and running quickly and easily
Dr. M. O. Faruque Sarker
Sam Washington
PACKT open source

320 halaman

1. Network Programming and Python

2. HTTP and Working with the Web

3. APIs in Action

4. Engaging with E-mails

5. Interacting with Remote Systems

6. IP and DNS

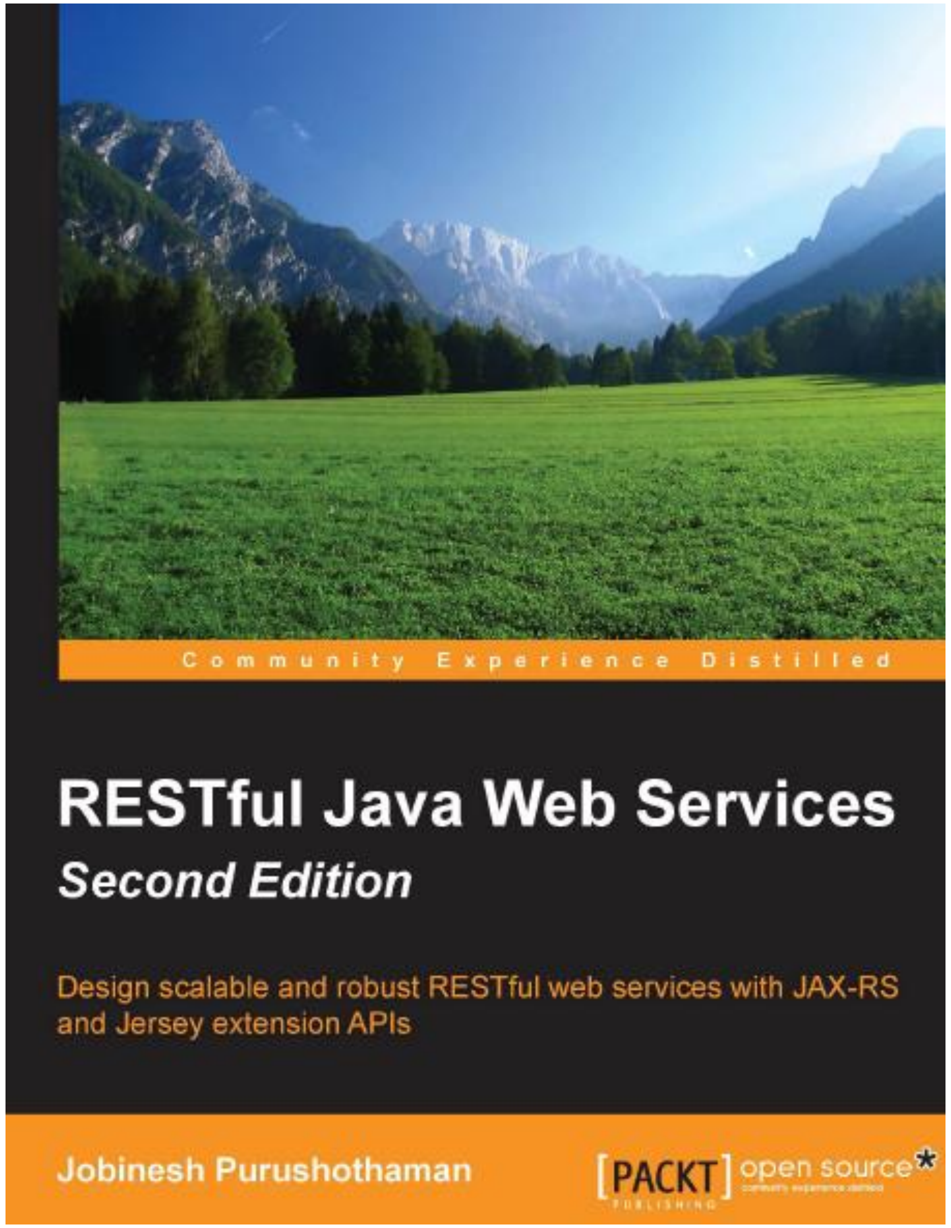
7. Programming with Sockets

8. Client and Server Applications

9. Applications for the Web

Lampiran: *Working with Wireshark*

1. ***Network Programming and Python***, introduces core networking concepts for readers that are new to networking, and also covers how network programming is approached in Python.
2. ***HTTP and Working with the Web***, introduces you to the HTTP protocol and covers how we can retrieve and manipulate web content using Python as an HTTP client. We also take a look at the standard library urllib and third-party Requests modules.
3. ***APIs in Action***, introduces you to working with web APIs using HTTP. We also cover the XML and JSON data formats, and walk you through developing applications using the Amazon Web Services Simple Storage Service (S3) and Twitter APIs.
4. ***Engaging with E-mails***, covers the principle protocols used in sending and receiving e-mails, such as SMTP, POP3, and IMAP, and how to work with them in Python 3.
5. ***Interacting with Remote Systems***, guides you through the ways of using Python to connect to servers and performing common administrative tasks, including the execution of shell commands through SSH, file transfers with FTP and SMB, authentication with LDAP, and to monitor systems with SNMP.
6. ***IP and DNS***, discusses the details of the Internet Protocol (IP), ways of working with IP in Python, and how to use DNS to resolve hostnames.
7. ***Programming with Sockets***, covers using TCP and UDP sockets from Python for writing low-level network applications. We also cover HTTPS and TLS for secure data transport.
8. ***Client and Server Applications***, looks at writing client and server programs for socket-based communication. By writing an echo application and a chat application we look at developing basic protocols, framing network data, and compare the multithreading and event-based server architectures.
9. ***Applications for the Web***, introduces you to writing web applications in Python. We cover the main approaches, methods of hosting Python web applications, and develop an example application in the Flask microframework.
10. ***Working with Wireshark***, covers packet sniffers, the installation of Wireshark, and how to capture and filter packets using the Wireshark application.


Community Experience Distilled

RESTful Java Web Services

Second Edition

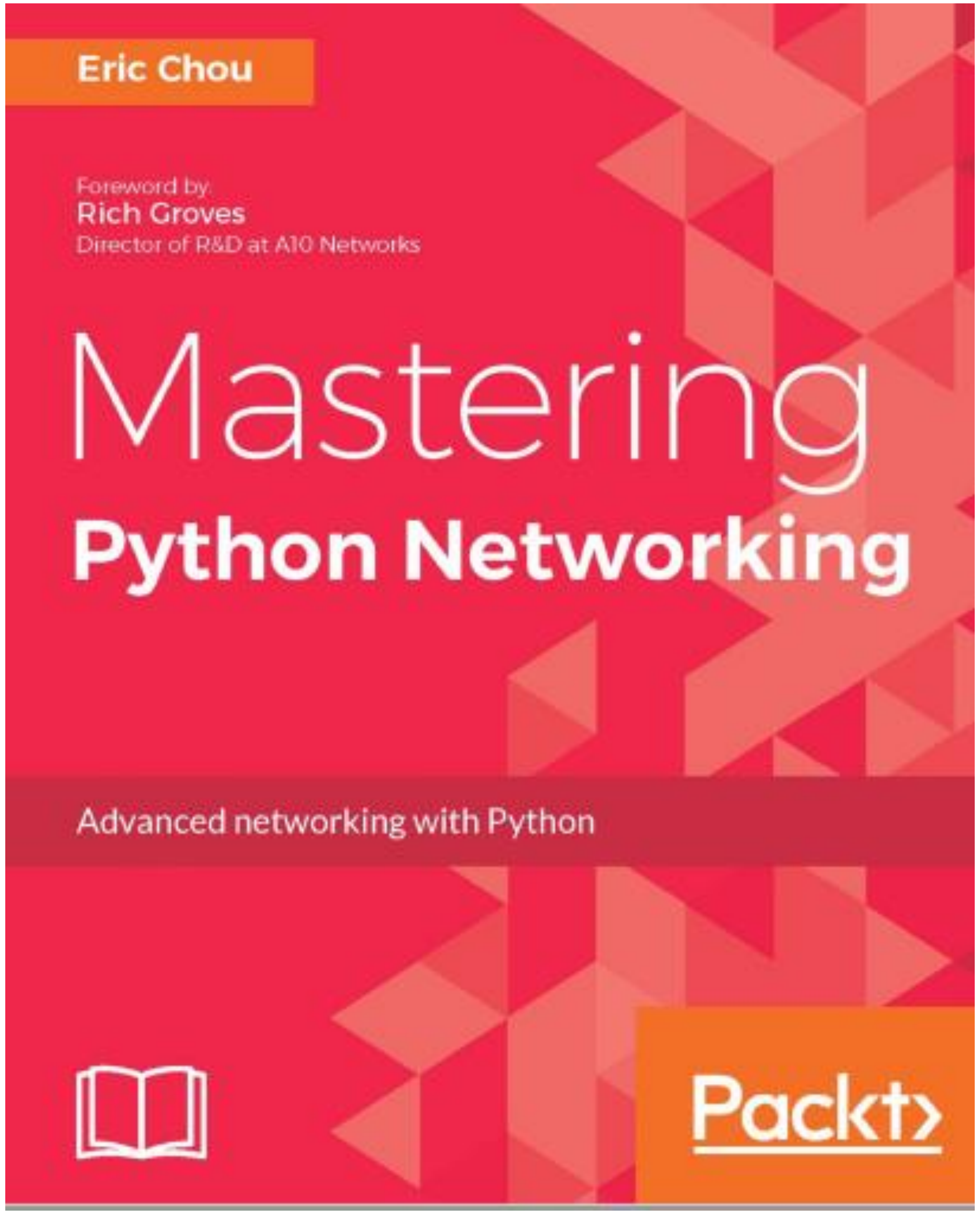
Design scalable and robust RESTful web services with JAX-RS and Jersey extension APIs

Jobinesh Purushothaman

[PACKT] open source* PUBLISHING

354 halaman

1. ***Introducing the REST Architectural Style***, covers the REST software architectural style and core architectural elements that form a RESTful system.
2. ***Java APIs for JSON Processing***, gives an overview of the JSON message format, and the popular tools and frameworks around JSON.
3. ***Introducing the JAX-RS API*, introduces JAX-RS APIs**. This chapter will explain how to build RESTful web services with JAX-RS APIs.
4. ***Advanced Features in the JAX-RS API***, takes a deeper look into the advanced JAX-RS APIs, along with many real life use cases and code samples.
5. ***Introducing the Jersey Framework Extensions***, discusses some of the very useful Jersey framework extension APIs that are not yet a part of the JAX-RS standard.
6. ***Securing RESTful Web Services***, explores how to secure RESTful web services using the HTTP basic authentication and OAuth protocols.
7. ***The Description and Discovery of RESTful Web Services***, describes popular solutions that are available today for describing, producing, consuming, and visualizing RESTful web services.
8. ***RESTful API Design Guidelines***, discusses the best practices and design guidelines that developers will find useful while building RESTful web services. Learning the best practices will help you avoid common pitfalls that others might have faced before.
9. ***Useful Features and Techniques***, covers various useful features and techniques that we had deferred while discussing specific topics in this book. This section explores tools and techniques for building, testing, extending, and packaging JAX-RS web applications.



1. Review of TCP/IP Protocol Suite and Python Language
2. Low-Level Network Device Interactions
3. API and Intent-Driven Networking
4. The Python Automation Framework - Ansible Basics
5. The Python Automation Framework - Ansible Advance Topics
6. Network Security with Python
7. Network Monitoring with Python - Part 1
8. Network Monitoring with Python - Part 2
9. Building Network Web Services with Python
10. OpenFlow Basics
11. Advanced OpenFlow Topics
12. OpenStack OpenDaylight and NFV
13. Hybrid SDN.

- Chapter 1, *Review of TCP/IP Protocol Suite and Python Language*, reviews the fundamental technologies that make up Internet communication today, from the OSI and client-server models to TCP, UDP, and IP protocol suites. It will also review the basics of the Python language in its types, operators, loops, functions, and packages.
- Chapter 2, *Low-Level Network Device Interactions*, uses practical examples to illustrate how to use Python to execute commands on a network device. It will discuss the challenges of having a CLI-only interface in automation. The chapter will use PExpect and Paramiko library examples.
- Chapter 3, *API and Intent-Driven Networking,* discusses the newer network devices that support Application Program Interfaces (APIs) and other high-level interaction methods. It also illustrates a tool that allows network engineers to abstract the low-level tasks when scripting in Python while focusing on the design and what you want the network to achieve. A discussion of Cisco NX-API, Juniper PyEZ, and Arista PyEAPI among other technologies is also included.
- Chapter 4, *The Python Automation Framework - Ansible Basics*, discusses the basics of Ansible, an open source, Python-based automation framework. Ansible goes one step further from APIs and focuses on network intents and device interaction. In this chapter, we will cover the advantages of using Ansible, its architecture, and practical examples of Ansible with Cisco, Juniper, and Arista devices.
- Chapter 5, *The Python Automation Framework - Ansible Advance Topics*, builds on the knowledge obtained from the previous chapter and covers the more advanced Ansible concepts such as conditionals, loops, templates, variables, vaults, and roles. It will also introduce how to write your own Ansible module that fits in your network environment.
- Chapter 6, *Network Security with Python*, introduces several Python tools to help you secure your network. It will discuss using Scapy for security testing, using Ansible to quickly implement access lists, and forensic analysis with syslog and UFW using Python.
- Chapter 7, *Network Monitoring with Python - Part 1*, covers monitoring the network using various tools. It will use SNMP and PySNMP for queries to obtain device information. From the results, we will use Matplotlib and Pygal to visualize the results. The chapter will end with Cacti examples and how to use Python scripts as input source.
- Chapter 8, *Network Monitoring with Python - Part 2*, covers more network-monitoring tools. It will start with using Graphviz to graph network graphs automatically from LLDP information. It will move to introducing push-based network monitoring using NetFlow and other similar technologies. We will use Python to decode flow packets as well as use ntop to visualize flow information. We will also introduce hosted Elasticsearch as a way to complement network monitoring.
- Chapter 9, *Building Network Web Services with Python*, shows you how to use the Python web framework, Flask, to create your own API on the network level. The network-level API offers benefits such as abstracting the requester away from network details, consolidating and customizing operations, and better security by limiting the exposure of available operations.
- Chapter 10, *OpenFlow Basics*, covers the basics of OpenFlow, a protocol that many credit as the technology that stared the software-defined networking movement. The protocol separates the control and data plane of network devices, which allows network operators to quickly prototype and innovate new features and functions. We will use the Python-based controller Ryu as well as Mininet to simulate an OpenFlow network. We will introduce examples of OpenFlow layer 2 switches and firewalls.
- Chapter 11, *Advanced OpenFlow Topics*, introduces advanced OpenFlow topics by building additional network applications and features using OpenFlow. We will start with building an OpenFlow router with static flows and then enhance the application with the REST API and build BGP speaking capabilities to interact with traditional networks. The chapter will end with using the firewall applications example as a way to virtualize traditional network functions.
- Chapter 12, *OpenStack, OpenDaylight, and NFV*, covers other software-defined networking projects: OpenStack, OpenDaylight, and Network Function Virtualization. We will focus on the OpenStack network project, Neutron, in the chapter to discuss the service architecture and how to try out OpenStack with TryStack and DevStack. The chapter will also cover a basic OpenDaylight controller example for a simple hub with Mininet.
- Chapter 13, *Hybrid SDN*, uses the knowledge from previous chapters and discusses various considerations and methods for implementing a software-driven network. We will discuss preparing your network for SDN and OpenFlow and considerations for greenfield eployment, controller redundancy, BGP interoperability, monitoring integration, controller secure TLS connection, and physical switch selection for your network.

Pekan	Tanggal	Topik	Bentuk	Keterangan
01.		Pengantar	Kuliah	
02.		Penjelasan Proyek Prinsip Jaringan	Kuliah	
03.		Prinsip Jaringan, lanjutan Remote Procedure Calls	Kuliah	
04.		Remote Procedure Calls, lanjutan Arsitektur	Kuliah	
05.		Arsitektur, lanjutan	Kuliah	
06.		Pemrograman Multithread, Socket	Kuliah	
07.		Penamaan	Kuliah	
08.		Ujian Tengah Semester (UTS)	Ujian	Open Book

Pekan	Tanggal	Topik	Bentuk	Keterangan
09.		Sinkronisasi	Kuliah	
10.		*Web Service*	Kuliah	
11		*Web Service (lanjutan)*	Kuliah	
12.		*Message Passing Interface*	Kuliah	
13.		Caching	Kuliah	
14.		Replikasi	Kuliah	
15.		Toleransi Kegagalan	Kuliah	
16.		Ujian Akhir Semester (UAS)	Ujian	Open Book

Kuliah Selanjutnya…

- Prinsip-prinsip Jaringan [Komputer]

Pertanyaan?