

# Sistem Terdistribusi

## TIK-604

### **Prinsip-prinsip Jaringan**

Kuliah 02: 18 s.d 20 Februari 2019

Husni

# Hari ini...

- Sebelumnya:

- Pengantar kuliah

- Kuliah hari ini:

- Jenis-jenis jaringan (*Network*)
- Prinsip-prinsip *Networking*:
  - *Encapsulation*
  - *Layering*
  - *Routing*
  - *Kendali Congestion*

- Pengumuman:

- Silakan belajar mandiri dan kelompok topik pemrograman multithread dengan Java, Python atau Bahasa pemrograman lain pilihan anda!

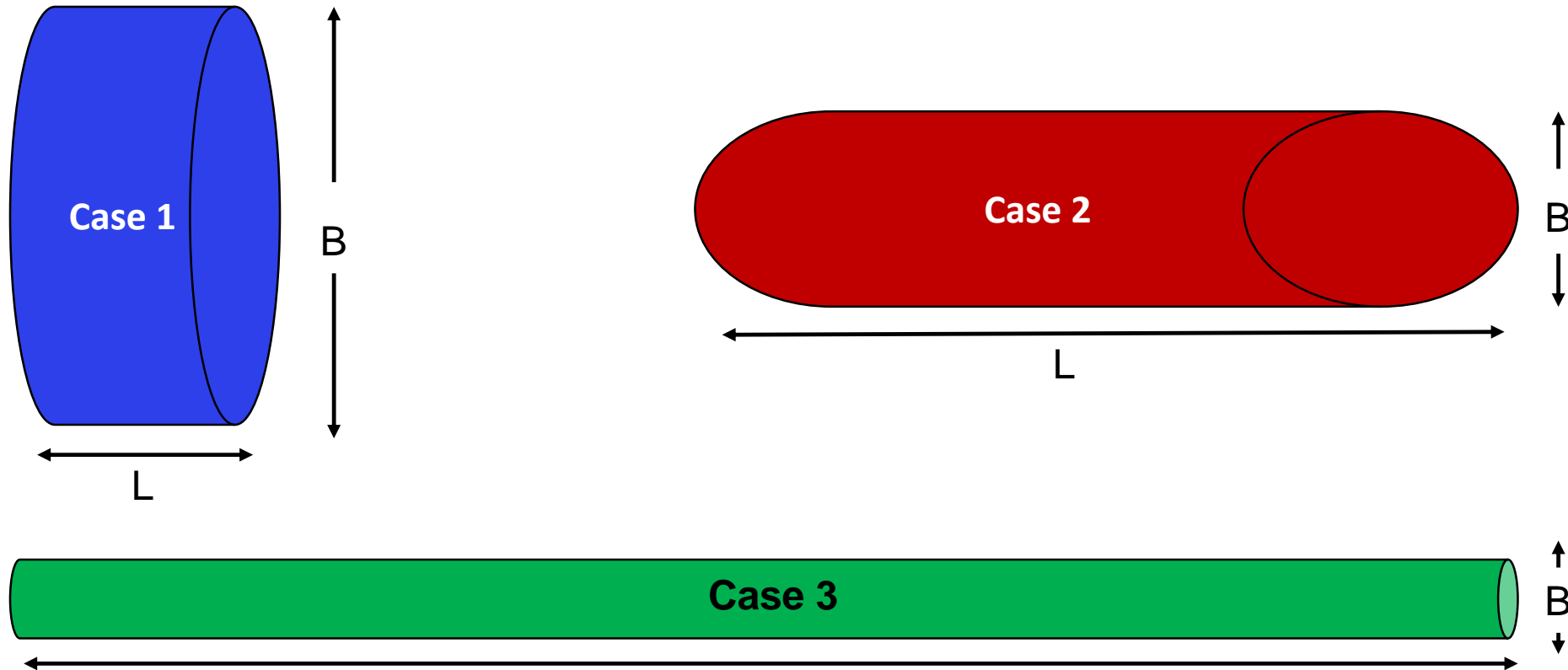
# Pengantar Networking: Tujuan Pembelajaran

- Memahami bagaimana komputer di Internet berkomunikasi
- Secara rinci, setelah kuliah topik ini, anda akan mampu:
  - Mengetahui jenis-jenis jaringan modern
  - Mendeskripsikan prinsip *networking* seperti *layering*, *encapsulation*, dan *packet-switching*
  - Menguraikan bagaimana paket dirutekan dan bagaimana kemacetan dikendalikan.
  - Menganalisis *scalability*, *reliability*, dan *fault-tolerance* di Internet

# Jaringan dalam Sistem Terdistribusi

- Sistem terdistribusi hanyalah koleksi komponen-komponen yang berkomunikasi untuk menyelesaikan suatu masalah
- Mengapa *programmer* sistem terdistribusi harus memahami jaringan komputer?
  - Isu *networking* sangat mempengaruhi *performance*, *fault-tolerance*, dan *security* dari sistem terdistribusi
  - Misal, *Gmail outage* pada Sep 1, 2010 – Jurubicara Google berkata “*we had slightly underestimated the load which some recent changes placed on the request routers. ... . few of the request routers became overloaded... causing a few more of them to also become overloaded, and within minutes nearly all of the request routers were overloaded.*”

# *Latency* dan *Bandwidth*



- $B$  = Bandwidth (atau *Capacity*) dan  $L$  = Latency (atau *Delay*)
- $B \times L$  memberikan perkiraan jumlah bit dalam perjalanan
- Saat  $B \times L$  meningkat, *ketidak-tentuan* meningkat (lebih banyak bit menghilang)
- Nilai tinggi dari  $B \times L$  mengakibatkan “*Buffer Bloat*”.

# Jaringan dalam Sistem Terdistribusi

Isu <i>Networking</i>	Ulasan Terhadap Rancangan Sistem Terdistribusi
<b>Performance</b>	Mempengaruhi <i>latency</i> dan <i>data-transfer-rate</i> dari <i>messages</i> .
<b>Scalability</b>	Ukuran Internet meningkat. Perlu manajemen lalu-lintas yang lebih baik di masa depan.
<b>Reliability</b>	Mendeteksi <i>error</i> komunikasi dan melaksanakan <i>error-checks</i> pada lapisan aplikasi.
<b>Security</b>	Install <i>firewall</i> . Memasang modul otentikasi, privasi dan keamanan <i>end-to-end</i> .
<b>Mobility</b>	Mengharapkan koneksi <i>intermittent</i> untuk perangkat bergerak.
<b>Quality-of-service</b>	Internet adalah karya terbaik. Sulit untuk memastikan jaminan QoS sempurna, misalnya untuk pesan multimedia.

# Klasifikasi Jaringan

Cara penting untuk mengelompokkan jaringan:

1. Berdasarkan pada ukuran

- *Body Area Networks (BAN)*
- *Personal Area Networks (PAN)*
- *Local Area Networks (LAN)*
- *Wide Area Networks (WAN)*

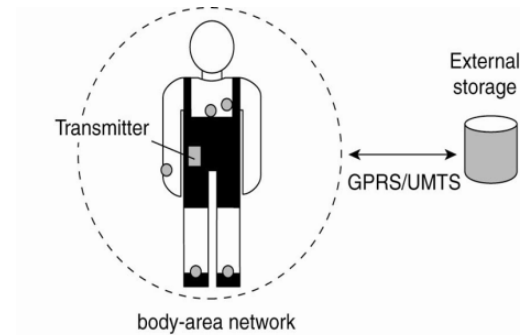
2. Berdasarkan pada teknologi

- *Ethernet Networks*
- *Wireless Networks*
- *Cellular Networks*

# Klasifikasi Jaringan: BAN dan PAN

- *Body Area Networks (BAN):*

- Perangkat yang ada membentuk unit komputasi *wearable*
- Beberapa *Body Sensor Unit (BSU)* berkomunikasi dengan *Body Central Unit (BCU)*
- Khususnya menggunakan jaringan *low-cost* dan *low-energy*



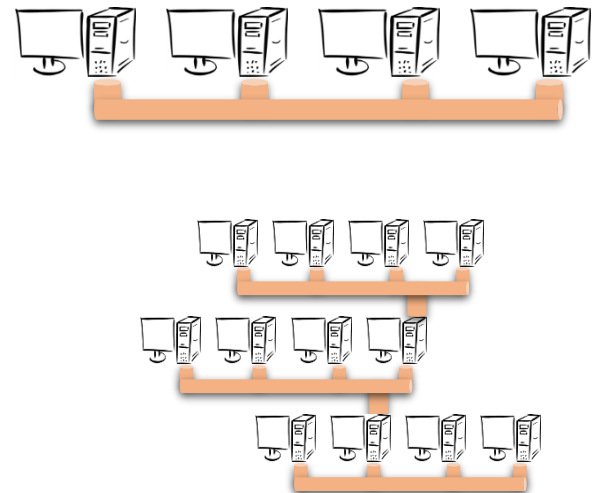
- *Personal Area Networks (PAN):*

- PAN menghubungkan berbagai perangkat digital yang dipakai oleh pengguna (*mobile phone, tablet, camera*)
- Jaringan *low-cost* dan *low-energy*
- Misalnya *Bluetooth*



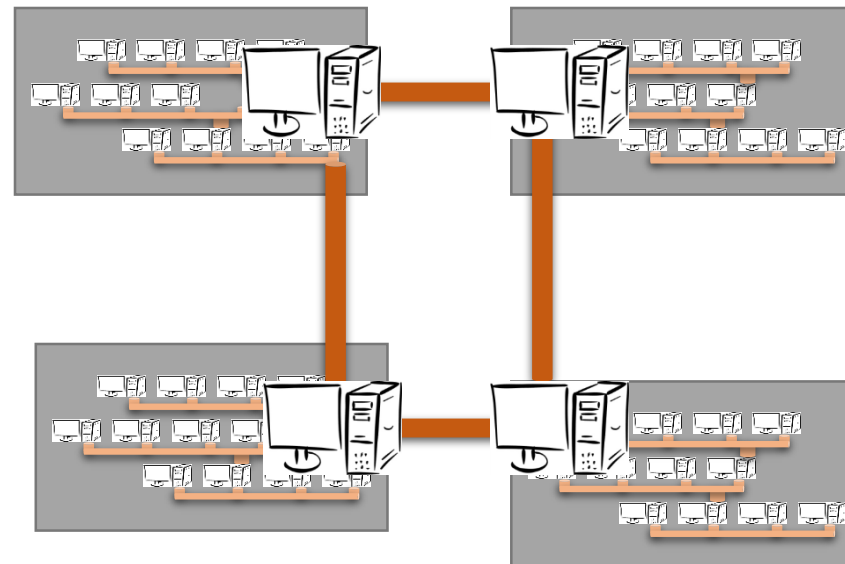
# Klasifikasi Jaringan: LAN

- Beberapa komputer yang dikoneksikan oleh suatu media komunikasi
  - Misalnya kabel *twisted copper*, *optical fiber*
- *Data-transfer-rate* tinggi dan *latency* rendah
- LAN terdiri dari:
  - Segmen
    - Biasanya dalam suatu departmen/lantai suatu gedung
    - *Bandwidth* dishare, tidak perlu *routing*
  - Jaringan lokal
    - Melayani kampus/gedung kantor
    - Banyak segmen dihubungkan dengan switch/hub
    - Khususnya, mewakili jaringan dalam suatu organisasi.

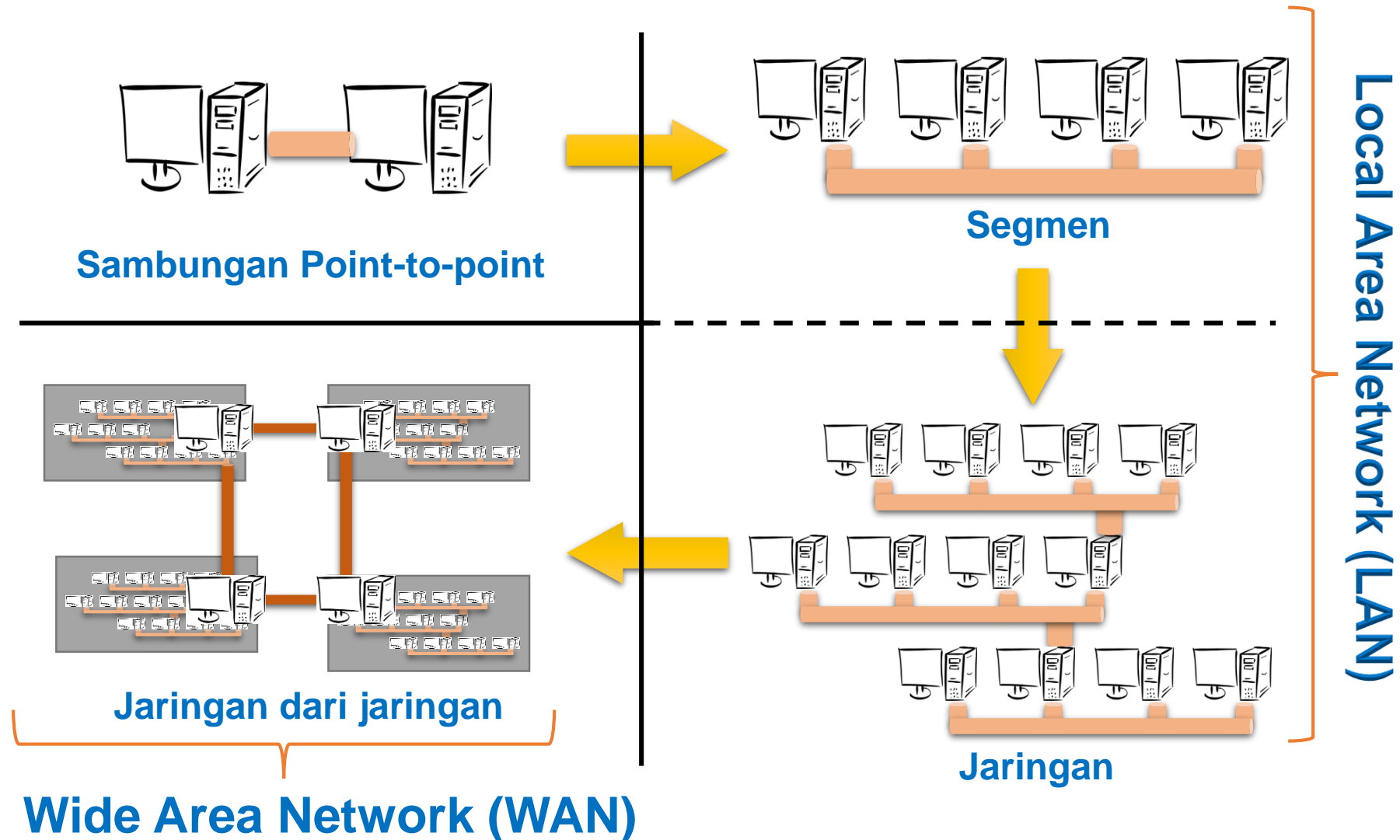


# Klasifikasi Jaringan: **WAN**

- Biasanya mencakup area yang lebih luas (kota, propinsi, negara,...)
- Terdiri dari jaringan-jaringan berbagai organisasi
- Lalu-lintas dirutekan dari satu organisasi ke lainnya
  - Routers
- *Bandwidth* dan *latency*
  - Bervariasi
  - Lebih buruk daripada LAN
- WAN Terbesar = Internet

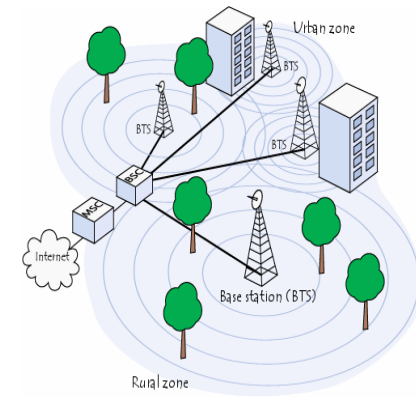


# Rangkuman Jaringan Berdasarkan Ukuran



# Jenis Jaringan: Berdasarkan Teknologi

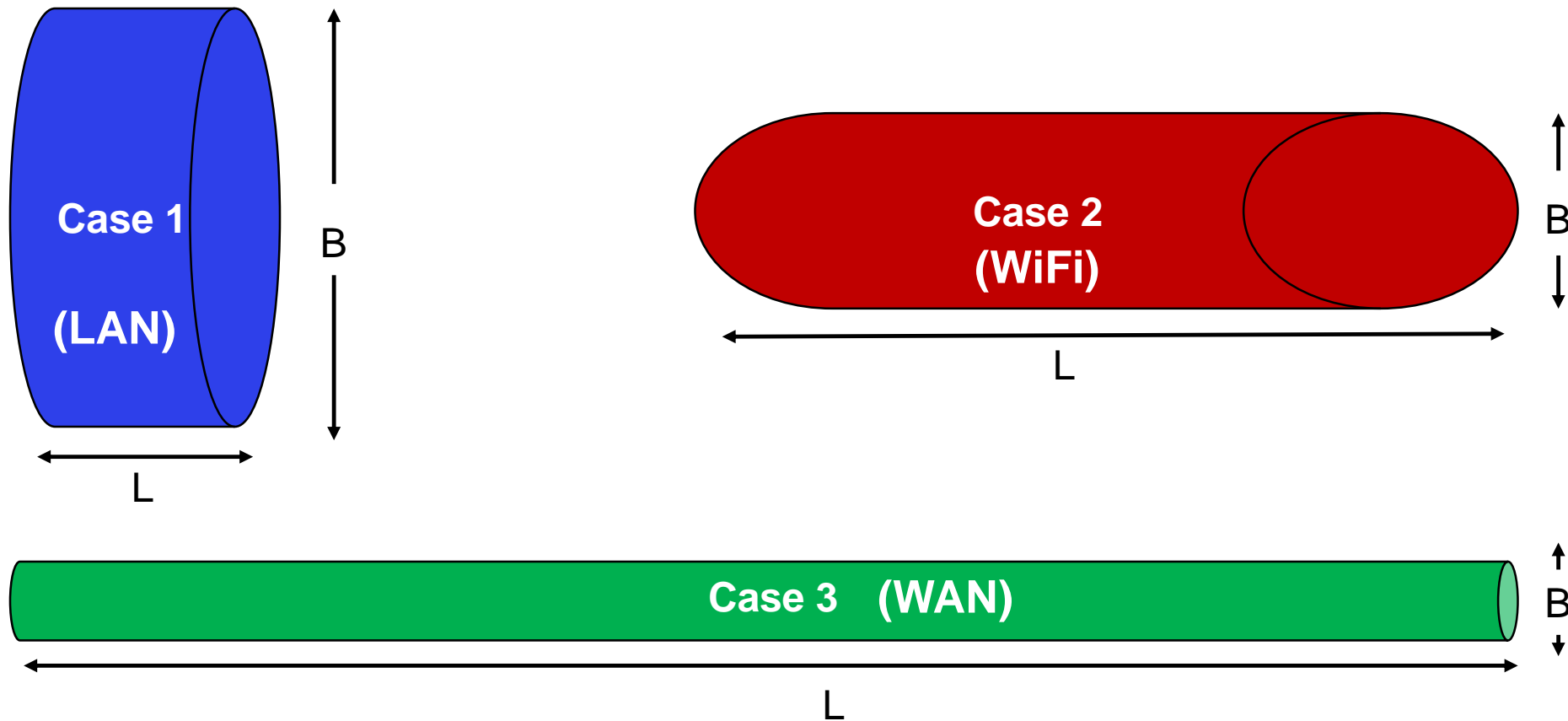
- Jaringan Ethernet
  - Sebagian besar digunakan dalam Internet berkabel
- LAN Nirkabel (*wireless*)
  - Terutama dirancang untuk menyediakan akses nirkabel ke Internet
  - Rentang rendah (100an meter), *bandwidth* tinggi
- Jaringan seluler (2G/3G/4G)
  - Awalnya dirancang untuk membawa suara
  - Rentang cakupan besar (beberapa kilometer)
  - *Bandwidth* rendah.



# Kinerja Khas Beberapa Jenis Jaringan

Jaringan	Contoh	Rentang	<i>Bandwidth</i> (Mbps)	<i>Latency</i> (ms)
Wired LAN	Ethernet	1-2 km	10 – 10,000	1 – 10
Wired WAN	Internet	Worldwide	0.5 – 600	100 – 500
Wireless PAN	Bluetooth	10 – 30 m	0.5 – 2	5 – 20
Wireless LAN	WiFi	0.15 – 1.5 km	11 – 108	5 – 20
Cellular	2G – GSM	100m – 20 km	0.270 – 1.5	5
Modern Cellular	3G	1 – 5 km	348 – 14.4	100 – 500

# *Latency* dan *Bandwidth*



- $B$  = Bandwidth (atau *Capacity*) dan  $L$  = Latency (atau *Delay*)
- $B \times L$  memberikan perkiraan jumlah bit di perjalanan
- Jika  $B \times L$  meningkat, *ketidakpastian* meningkat (lebih banyak bit yang dapat hilang)
- Nilai tinggi dari  $B \times L$  mengakibatkan “*Buffer Bloat*”.

# Prinsip-prinsip *Networking*

- Protokol-protokol jaringan
- Transmisi paket
- Jaringan berlapis (*layered*)
  - *Physical layer*
  - *Data-link layer*
  - *Network layer* dan *routing*
  - *Transport layer* dan kendali kemacetan (*congestion*)

# Protokol *Networking*

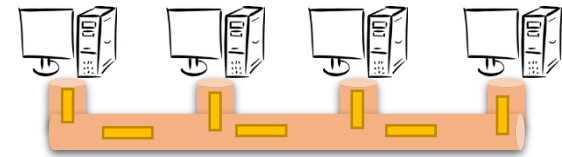
- Jika dua entitas ingin berkomunikasi pada suatu *network*, diperlukan suatu kesepakatan yang telah didefinisikan
  - Berapa banyak bit harus digunakan untuk mensinyalkan bit 0 atau bit 1?
  - Bagaimana penerima (*receiver*) tahu bit terakhir dari pesan (*message*)?
  - Bagaimana *receiver* dapat mendeteksi jika pesannya rusak?
- Protokol adalah suatu himpunan aturan dan format yang *well-known* untuk komunikasi antar entitas
- Menstandarkan himpunan protokol yang *well-known* akan mendukung komunikasi antar entitas heterogen (lintas vendor)



# Transmisi Paket

- *Messages* dipecah ke dalam paket-paket (*packets*)

- Paket merupakan satuan data yang ditransmisikan antara asal (*source*) dan tujuan (*destination*)
- Paket dapat berubah-ubah panjangnya



- Ukuran maksimum dari paket dikenal sebagai *Maximum Transmission Unit* (MTU)

- MTU mencegah suatu host mengirimkan *message* yang terlalu panjang

- Setiap paket mempunyai dua *field* utama:

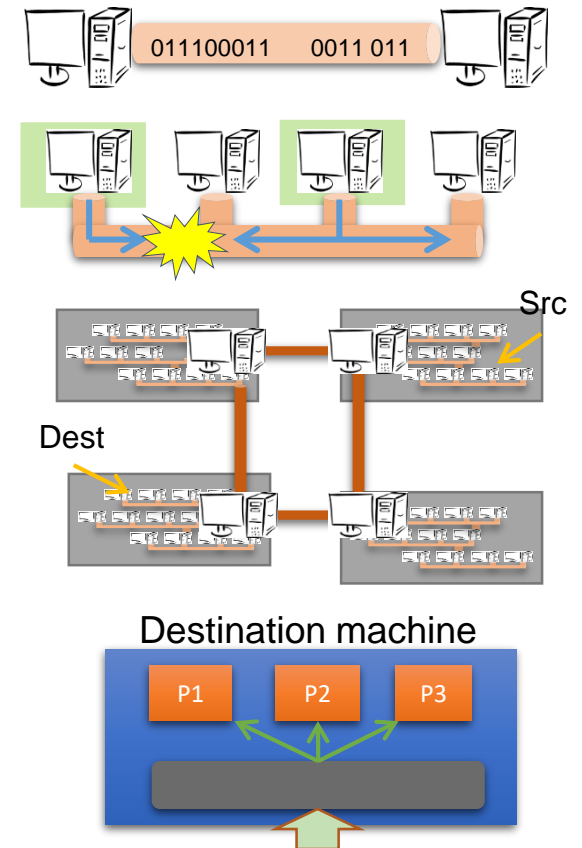
- **Header:** berisi *meta-information* mengenai paket
  - Misal Panjang paket, ID penerima
- **Data**



# Lapisan dalam Jaringan

- *Software* jaringan disusun ke dalam suatu hirarki lapisan (*layer*)
  - Protokol dalam satu layer melaksanakan satu fungsi tertentu
  - Layering merupakan rancangan *scalable* dan *modular* untuk *software* kompleks
- Fungsi khusus dalam software jaringan:

Fungsi (Kemampuan)	Layer (Lapisan)
Mengirimkan bit-bit pada media transmisi	Physical
Mengkoordinasikan transmisi dari banyak host yang secara langsung terkoneksi pada media umum	Data link
Merutekan paket-paket melalui jaringan lanjutan	Network
Menangani pesan – bukan paket – antara proses pengirim dan penerima ( <i>sender &amp; receiver</i> )	Transport
Memenuhi <i>requirement</i> komunikasi bagi aplikasi tertentu.	Application



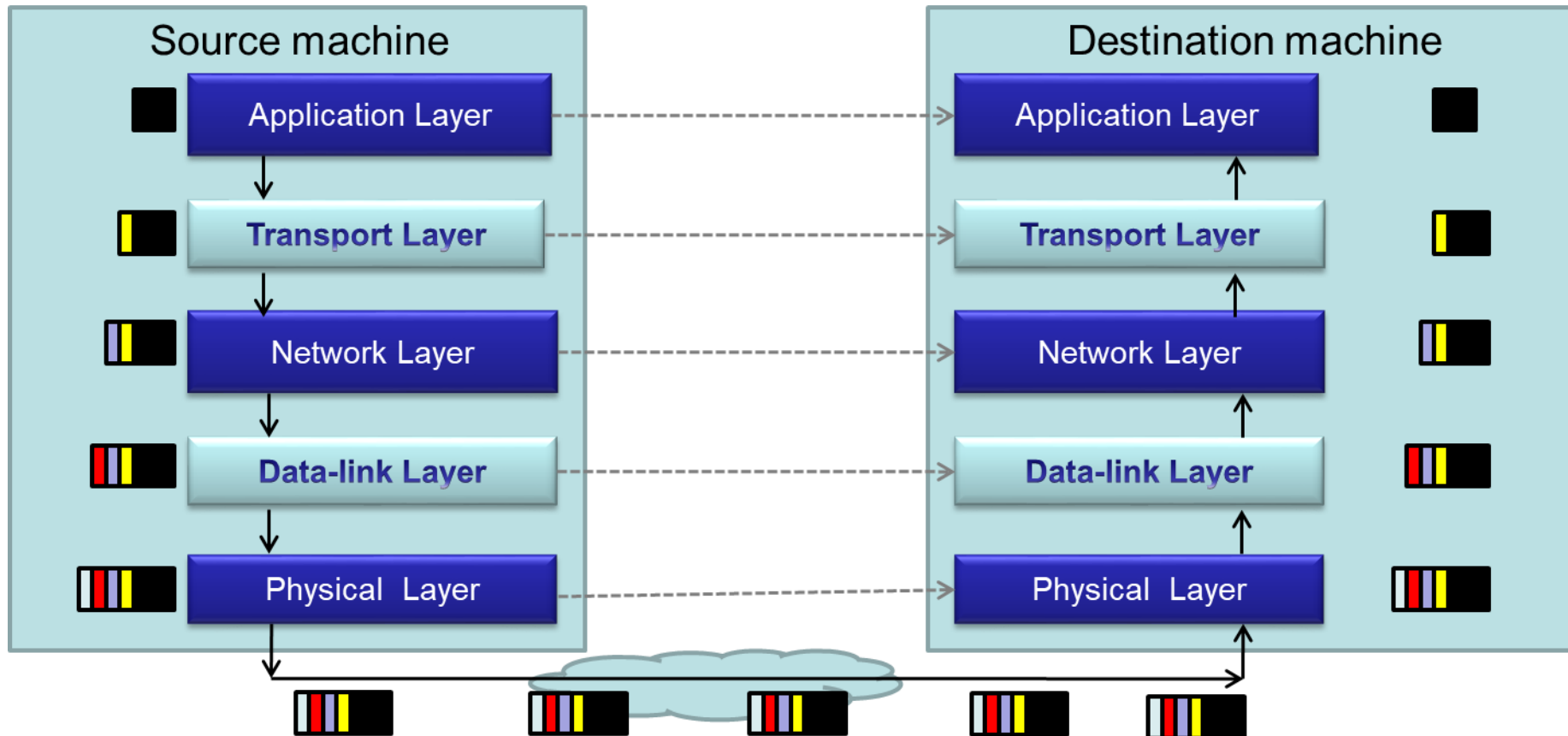
# Model Referensi OSI

- Model Referensi *Open Systems Interconnection* (OSI)
  - Suatu model *networking* berlapis yang distandardkan oleh ISO
  - Model ini mengidentifikasi berbagai *layer* dan fungsinya

Fungsi	Layer	Contoh Protokol
Memenuhi kebutuhan komunikasi bagi aplikasi spesifik	Application	HTTP, FTP
Mengirimkan data dalam representasi jaringan yang independent dari representasi komputer perorangan	Presentation	Representasi Data CORBA
Mendukung <i>reliability</i> dan adaptasi, seperti deteksi kegagalan dan penyembuhan otomatis.	Session	SIP
Menangani message, bukan paket, antara proses <i>sender</i> dan <i>receiver</i>	Transport	TCP, UDP
Menentukan rute paket melalui jaringan berikutnya.	Network	IP, ATM
Mengkoordinasikan transmisi dari banyak host yang secara langsung terkoneksi pada suatu medium umum	Data-link	Ethernet MAC
Mentransmisi bit-bit melalui media transmisi.	Physical	Ethernet

# Enkapsulasi Paket

- *Encapsulation* merupakan Teknik untuk membungkus dan membuka bungkusan paket data dalam arsitektur berlapis (*layered*)



# Layer yang akan kita pelajari Hari ini

1. *Physical layer*
2. *Data-link layer*
3. *Network layer*
4. *Transport layer*

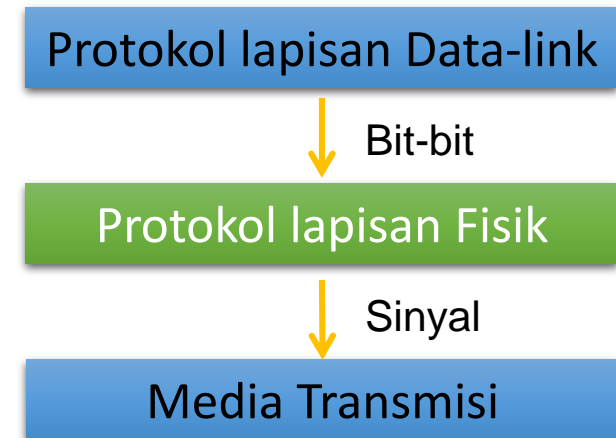
# Layer yang akan kita pelajari hari ini

1. *Physical layer*
2. *Data-link layer*
3. *Network layer*
4. *Transport layer*

# Lapisan *Physical*

- Protokol layer Fisik mengirimkan serangkaian bit pada media transmisi
  - Memodulasi bit-bit ke dalam sinyal yang dapat dilewatkan di atas media

Media Transmisi	Jenis Sinyal
Twisted-pair (kabel Ethernet)	Sinyal listrik
Fiber Optic Circuits	Sinyal cahaya ( <i>light</i> )
Wireless channel	Sinyal <i>Electro-magnetic</i>



# Layer yang akan kita pelajari hari ini

1. *Physical layer*
2. *Data-link layer*
3. *Network layer*
4. *Transport layer*

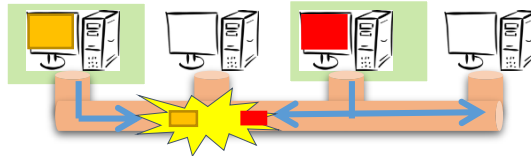


# Lapisan Data-link

- Protokol dalam layer *data-link* memastikan bahwa paket-paket dihantarkan dari satu host ke lainnya di dalam satu jaringan lokal
- Protokol pada layer *Data-link* menyediakan dua fungsi utama:
  - Bagaimana mengkoordinasi antar *transmitter* sehingga paket-paket diterima dengan baik?
    - Koordinasi
  - Bagaimana mengidentifikasi host lain pada jaringan lokal tersebut?
    - Pengalamatan (*addressing*) pada jaringan lokal.

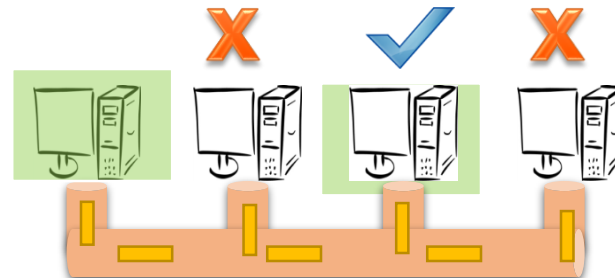
# Koordinasi pada layer Data-link

- Paket tidak berhasil diterima oleh *receiver* jika *sender* mengirimkan (*transmit*) data tersebut pada saat transmisi *sender* lain sedang aktif
  - Paket tersebut dikatakan mengalami tabrakan (*collision*) jika tidak berhasil sampai pada *receiver*
- *Collision* dihindari dengan memeriksa media sebelum transmisi



# Pengalamatan Pada Jaringan Lokal

- Setiap perangkat yang terkoneksi ke jaringan mempunyai alamat unik bernama alamat *Medium Access Control (MAC)*
  - *MAC address* panjangnya 6 byte
    - Misal: 2A:D4:AB:FD:EF:8D
- Pendekatan:
  - Layer Data-link mem-*broadcasts* paket pada media
  - *Receiver* membaca *header* paket dan memeriksa jika paket tersebut dialamatkan ke dirinya

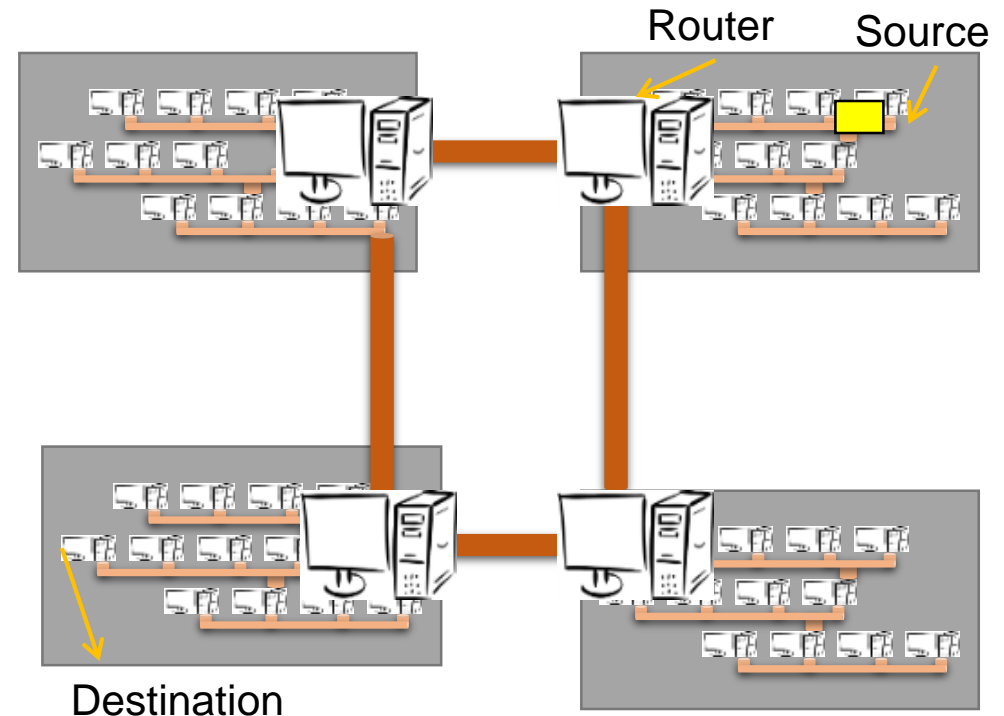


# Layer yang akan kita pelajari hari ini

1. *Physical layer*
2. *Data-link layer*
3. *Network layer*
4. *Transport layer*

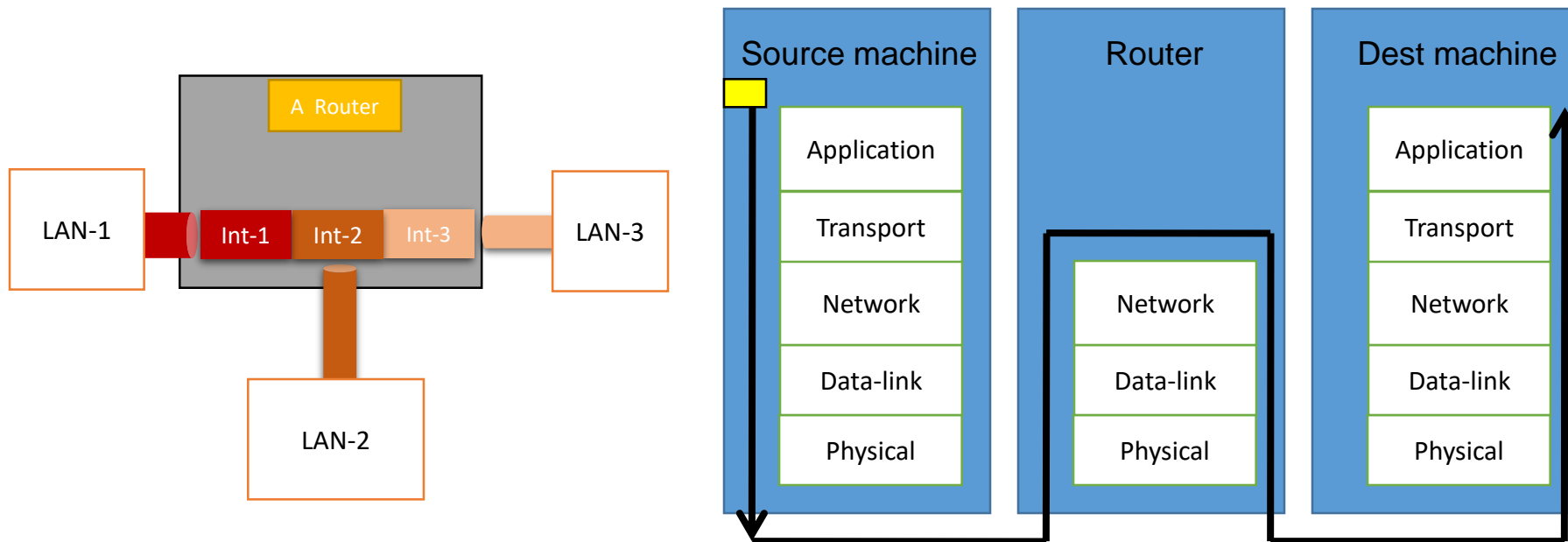
# Lapisan Network

- Protokol pada layer Network melaksanakan peran dari *routing*
  - Protokol memastikan bahwa suatu paket dirutekan dari mesin asal (*source*) ke mesin tujuan (*destination*)
  - Paket-paket boleh melintasi LAN berbeda untuk mencapai tujuan tersebut
- *Internet Protocol* (IP) adalah protokol layer network yang paling luas digunakan
  - Khususnya, IP address digunakan untuk mengidentifikasi mesin



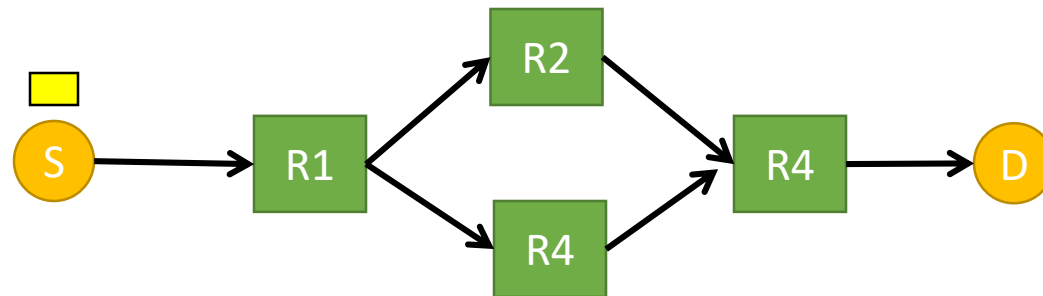
# Router

- **Router**: perangkat yang meneruskan paket melalui banyak jaringan
- Router menghubungkan dua atau lebih jaringan
  - Setiap *network interface* dikoneksikan ke LAN atau host (lokal)
- Paket berjalan sampai layer network pada router



# Algoritma *Routing*

- Paket harus ditransmisikan dalam serangkaian *hop* melalui router-router
  - Rentetan *hop* yang ditempuh oleh paket dikenal sebagai rute (*route*)
- **Algoritma *Routing*** bertanggung-jawab menentukan rute bagi transmisi paket
- Tantangan untuk perancangan *routing algorithm* di Internet:
  - Kinerja: Lalulintas lintas berbagai jaringan bervariasi
  - Kegagalan router: Router di Internet mungkin mengalami kerusakan



# Algoritma *Routing* (Lanj.)

- *Routing algorithm* mempunyai dua aktifitas
  1. Memutuskan *next-hop* yang diambil oleh setiap paket
    - Algoritmanya harus cepat dan efisien
  2. Secara dinamis mengupdate informasi konektifitas
    - Memelihara pengetahuan jaringan dengan memonitor router dan lalu-lintas
- Aktifitas di atas didistribusikan di seluruh jaringan
  - Keputusan *routing* dibuat mengikuti suatu *hop-by-hop basis*
  - Informasi mengenai router *next-hop* yang mungkin disimpan secara lokal
  - Informasi diupdate secara berkala
- Kita akan membahas algoritma routing sederhana bernama *Distance Vector Algorithm*.



# *Distance Vector Algorithm*

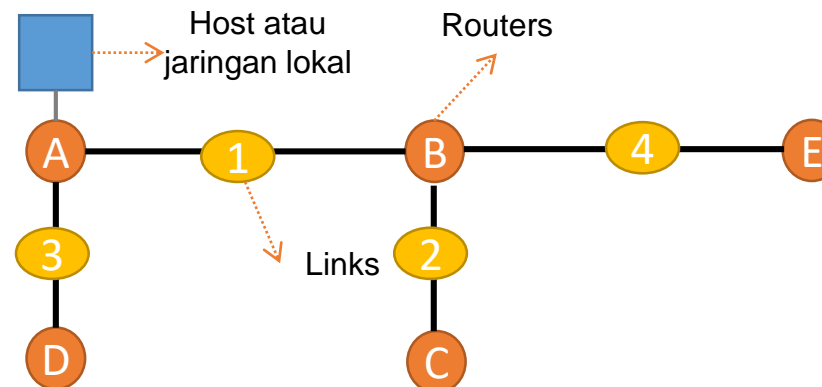
- *Distance Vector* (DV) menggunakan algoritma teori graf untuk menemukan rute terbaik dalam jaringan
  - DV menggunakan algoritma *path* terpendek terkenal bernama **Bellman-Ford**
- Dua aktifitas utama dari algoritma *routing* DV:
  1. Menentukan *next-hop* terbaik pada setiap router
  2. Secara dinamis *mengupdate* informasi konektifitas pada semua router.

# Distance Vector Algorithm:

## Penentuan *Next-hop*

- Setiap *router* memelihara suatu *tabel routing* yang terdiri dari:
  - **Destination**: IP tujuan dari paket tersebut
  - **Link**: Sambungan keluar (*outgoing*) kemana paket seharusnya diteruskan
  - **Cost**: Jarak antara *router* tersebut dengan tujuan
    - Misal, *Cost* dapat diperkirakan sebagai *delay* bagi paket untuk mencapai tujuan
- *Router* melihat tabel tersebut untuk menentukan *next-hop* terbaik

Tabel routing pada router A		
To	Link	Cost
A	local	0
B	1	1
C	1	2
D	3	1
E	1	2



# Tabel Routing untuk Skenario Contoh

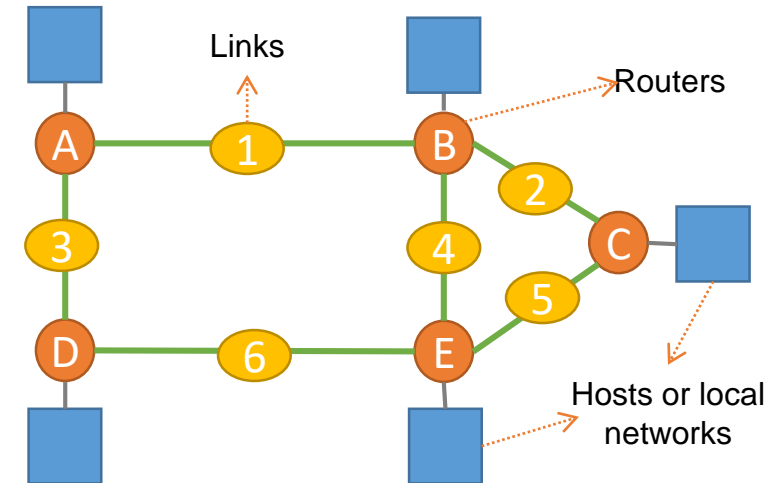
<i>Routings from A</i>		
<i>To</i>	<i>Link</i>	<i>Cost</i>
A	local	0
B	1	1
C	1	2
D	3	1
E	1	2

<i>Routings from B</i>		
<i>To</i>	<i>Link</i>	<i>Cost</i>
A	1	1
B	local	0
C	2	1
D	1	2
E	4	1

<i>Routings from C</i>		
<i>To</i>	<i>Link</i>	<i>Cost</i>
A	2	2
B	2	1
C	local	0
D	5	2
E	5	1

<i>Routings from D</i>		
<i>To</i>	<i>Link</i>	<i>Cost</i>
A	3	1
B	3	2
C	6	2
D	local	0
E	6	1

<i>Routings from E</i>		
<i>To</i>	<i>Link</i>	<i>Cost</i>
A	4	2
B	4	1
C	5	1
D	6	1
E	local	0



# *Distance Vector Algorithm:*

## Mengupdate Informasi Konektifitas

- Konektifitas diupdate melalui pertukaran tabel *routing*
- **Router Information Protocol** (RIP) digunakan untuk pengiriman pesan (*message*) update
  1. Kirimkan tabel *routing* ke router-router tetangga
    - Secara berkala, atau saat tabel lokal berubah
  2. Saat tabel *routing* tetangga diterima:

Kasus	Jika tabel routing yang diterima ...	Update terhadap tabel routing lokal
1.	Mempunyai tujuan baru yang tidak ada di dalam tabel routing lokal	Update Cost dan Link
2.	Mempunyai rute dengan cost lebih baik ke tujuan di dalam tabel routing lokal	Update Cost saja
3.	Mempunyai informasi yang lebih baru	Update Cost dan Link

# Pseudocode dari RIP

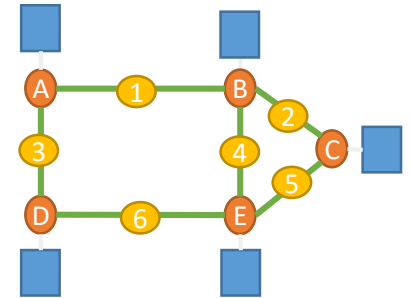
**Send:** Each  $t$  seconds or when  $TL$  changes, send  $TL$  on each non-faulty outgoing link

**Receive:** Whenever a routing table  $Tr$  is received on link  $n$ :

```

for all rows  $Rr$  in  $Tr$  {
  if ( $Rr.Link \neq n$ ) {
     $Rr.cost = Rr.cost + 1$ ; // Update cost
     $Rr.Link = n$ ; // Update next-hop
    if ( $Rr.destination$  is not in  $TL$ ) {
      add  $Rr$  to  $TL$ ; // add new destination to  $TL$  Kasus 1
    }
  }
  else for all rows  $RL$  in  $TL$  {
    if ( $Rr.destination = RL.destination$ ) {
      Kasus 2 //  $Rr.cost < RL.cost$  : remote node has better route
      Kasus 3 //  $RL.Link = n$  : information is more recent
      if ( $Rr.cost < RL.cost$  OR  $RL.Link = n$ ) {
         $RL = Rr$ ;
      }
    }
  }
}

```



$TL$  at A

To	Link	Cost
A	local	0
D	3	1
C	3	3

$Tr$  recvd @ A from B on link  $n=1$

To	Link	Cost
A	1	1
B	local	0
C	2	1

# Rangkuman: Routing di Internet

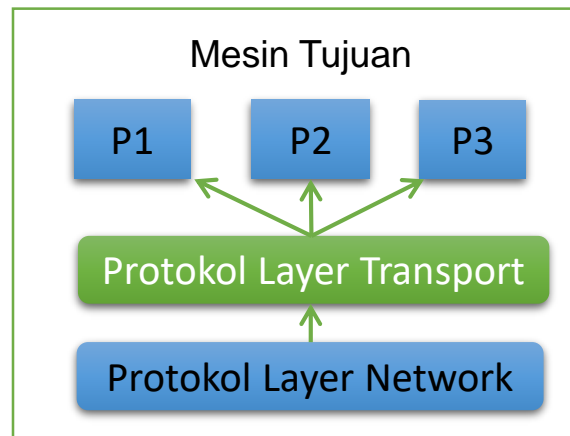
- Setiap mesin di Internet diidentifikasi oleh suatu IP Address
- Mesin asal (*source*) mengirimkan paket pada jaringan lokalnya
- Router lanjutan memeriksa paket tersebut, dan meneruskannya ke router dengan *next-hop* terbaik
- Jika tujuan (*destination*) secara langsung tersambung ke jaringan lokal dari router, router meneruskan paket pada jaringan lokal masing-masing.
- Router-router bertukar informasi untuk menjadi informasi yang *up-to-date* mengenai jaringan.

# Lapisan yang akan dibahas...

1. *Physical layer*
2. *Data-link layer*
3. *Network layer*
4. *Transport layer*

# Lapisan Transport

- Protokol pada layer Transport menyediakan aplikasi komunikasi *end-to-end*
- Ini merupakan layer paling rendah dimana *message* (bukan *packet*) ditangani
- *Message* dialamatkan ke *port* komunikasi yang tersambung ke proses
  - Layer Transport *me-multiplexes* setiap paket yang diterima ke portnya masing-masing





# Protokol Layer Transport Sederhana

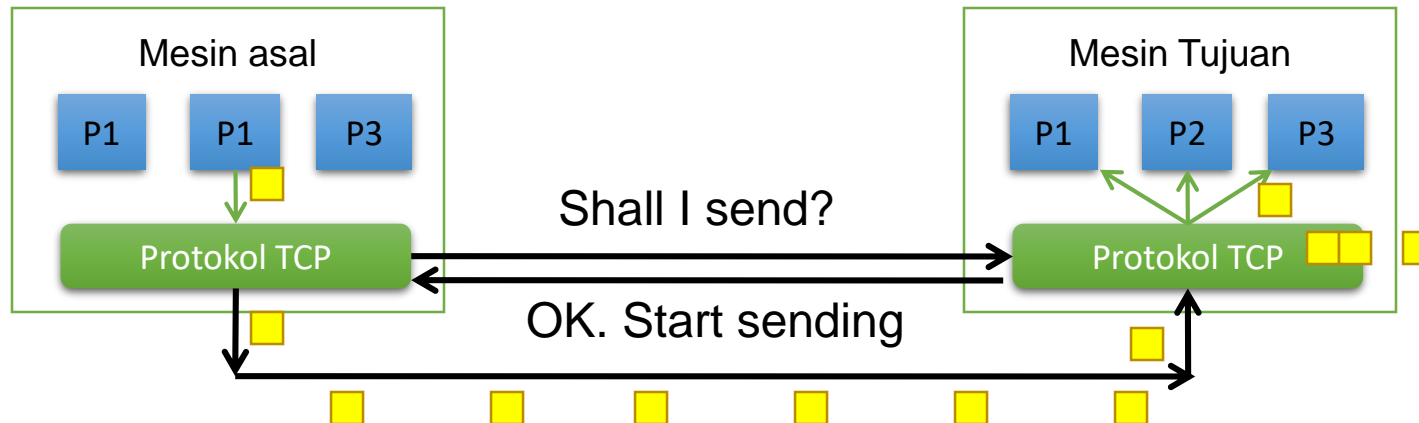
- Protokol Transport (sederhana) menyediakan layanan berikut:
  1. *Multiplexing Service*
  2. *Connection-less Communication*: Proses *sender* dan *receiver* tidak menginisiasi suatu koneksi sebelum pengiriman *message*
    - Setiap *message* dienkapsulasi dalam paket (disebut pula *datagram*)
    - *Message* pada *receiver* dapat berbeda urutannya dengan yang dikirimkan oleh *sender*
    - Contoh: *User Datagram Protocol (UDP)*

# Protokol Layer Transport Lanjutan

- Protokol layer transport yang lebih baik biasanya menyediakan layanan lebih dari sekedar *multiplexing* sederhana
- Transmission Control Protocol (TCP) merupakan protocol yang luas digunakan yang menyediakan tiga layanan tambahan:
  1. Komunikasi berorientasi koneksi (*connection-oriented*)
  2. Reliabilitas (*reliability*)
  3. Kendali Kemacetan (*congestion control*)

# 1. Komunikasi Berorientasi Koneksi

- *Sender* dan *receiver* akan bersalaman sebelum pengiriman *messages*
  - Salaman (*handshake*) membantu mensetup parameter-parameter koneksi, dan mengalokasikan sumber daya pada tujuan (*destination*) untuk menerima paket-paket.
- Destination menyediakan suatu *in-order delivery* dari messages untuk proses yang dimaksud
  - *Destination* akan mem-*buffer* paket tersebut sampai paket sebelumnya diterima
  - Ia kemudian akan menghantarkan paket-paket dengan urutan yang sama seperti dari *sender*.



## 2. Reliabilitas

- Paket-paket mungkin hilang dalam jaringan dikarenakan *buffer overflows* pada router atau error transmisi
- Dalam TCP, destination mengirimkan suatu ACK kepada sender
  - Jika ACK tidak diterima oleh sender, sender akan menganggap adanya *packet error*, dan melakukan pengiriman ulang paket tersebut.

### 3. Kendali Kemacetan

- Kapasitas dari jaringan dibatasi oleh setiap sambungan (link) komunikasi dan router
  - Ruang *buffer* dan *link-bandwidth* terbatas
- Apa yang terjadi jika suatu *Sender* mentransmit paket pada kecepatan yang lebih besar dari pada kapasitas jaringan?
  - Paket jatuh pada router lanjutan (di perjalanan)
  - ACK bersesuaian tidak akan diterima oleh Sender (asal)
  - Sender melakukan pengiriman ulang (*retransmit*)
  - Lebih banyak paket dibangun pada antrian router
  - Jaringannya runtuh (*down*).

# 3. Kendali Kemacetan (Lanj.)

- Untuk menghindari kemacetan, *dua* fungsi dapat diadopsi:
  1. Deteksi kemacetan pada router
    - Jika suatu router mengira akan *buffer overflow*, ia mengikuti satu dari dua strategi:
      - Menjatuhkan paket dan membiarkan *sources* mengatur ulang dengan mengamati *packet loss*
      - Mengirimkan paket “*Explicit Congestion Notification (ECN)*” ke *source*
  2. Atur input pada Sender (*source*)
    - Jika TCP-sender menyimpulkan macet (misal karena menerima paket ECN), maka ia mengurangi kecepatan pengiriman paketnya.

# Rekap: Prinsip-prinsip Jaringan

## Tujuan Pembelajaran

- Memahami bagaimana komputer-komputer di Internet berkomunikasi
- Secara rinci, setelah mempelajari topik ini, anda akan mampu untuk:
  - Mengetahui berbagai jenis jaringan (berbeda)
  - Menggambarkan prinsip-prinsip networking seperti *layering*, *encapsulation*, dan *packet-switching*
  - Menjelaskan bagaimana paket-paket data dirutekan dan bagaimana kemacetan dikendalikan.
  - Menguraikan konsep *scalability*, *reliability*, dan *fault-tolerance* di Internet.

# Kuliah berikutnya...

- *Remote Procedure Calls (RPC)*