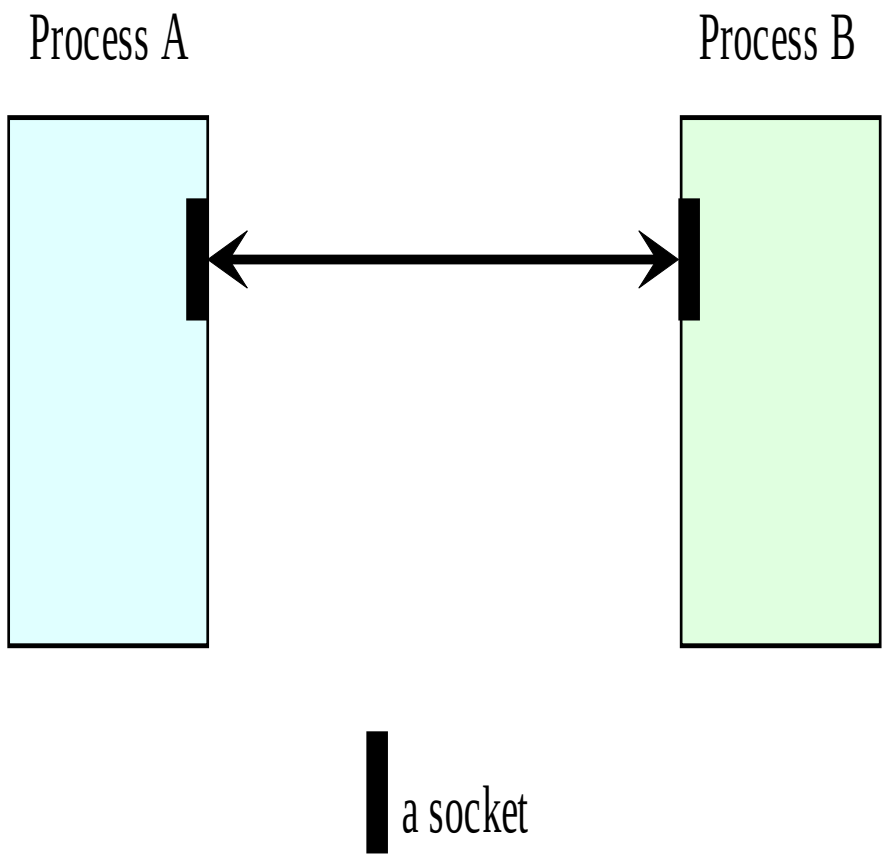
Pemrograman Jaringan 6

anton@ukdw.ac.id

Socket

- Socket adalah sebuah abstraksi perangkat lunak yang digunakan sebagai suatu "terminal" dari suatu hubungan antara dua mesin atau proses yang saling berinterkoneksi.



Operasi Socket

- Socket dapat melakukan operasi:
  - Koneksi ke mesin remote
  - Mengirim data
  - Menerima data
  - Mentutup koneksi
  - Bind to a port
  - Listen pada data yang masuk
  - Menerima koneksi dari mesin remote pada port tertentu
- Di tiap mesin yang saling berinterkoneksi, harus terpasang socket.

Socket API in Java

- Pada J2SE telah disediakan paket java.net yang berisi kelas- kelas dan interface yang menyediakan API (Application Programming Interface):
  - level rendah (Socket, ServerSocket, DatagramSocket)
  - level tinggi (URL, URLConnection).
- Disimpan pada package java.net.*

InetAddress class

- Kelas ini digunakan untuk mengambil informasi IP suatu komputer. Kelas ini bersifat static dan tidak memiliki konstruktor.

  Method-methodnya adalah:

  – getByName(namahost) yang akan menerima sebuah string nama host dan mengembalikan alamat IP berdasarkan DNS, berupa object InetAddress.
    - Untuk menampilkannya: gunakan method toString()
  – getLocalHost() yang akan mengembalikan alamat IP dan nama host pada komputer lokal.
  – getAllByName(namahost) mengembalikan array InetAddress
- Kemungkinan error: UnknownHostException

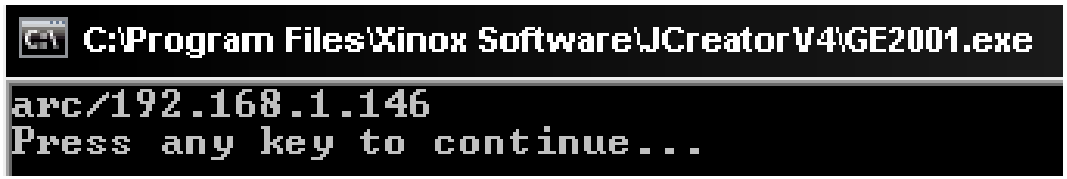
Contoh getByName

```
Enter host name: www.google.com
IP address: www.google.com/66.249.89.99
Press any key to continue...
```

```
import java.net.*;
import java.util.*;

public class IPFinder
{
    public static void main(String[] args)
    {
        String host;
        Scanner input = new Scanner(System.in);

        System.out.print("\n\nEnter host name: ");
        host = input.next();
        try
        {
            InetAddress address =
                            InetAddress.getByName(host);
            System.out.println("IP address: "
                                    + address.toString());
        }
        catch (UnknownHostException uhEx)
        {
            System.out.println("Could not find " + host);
        }
    }
}
```

Contoh getLocalHost

```
import java.net.*;

public class MyLocalIPAddress
{
    public static void main(String[] args)
    {
        try
        {
            InetAddress address =
                                InetAddress.getLocalHost();
            System.out.println(address);
        }
        catch (UnknownHostException uhEx)
        {
            System.out.println(
                        "Could not find local address!");
        }
    }
}
```



Contoh getAllByName

```
import java.net.*;
public class Jaringan {
    public static void main(String[] args) {
        try{
            InetAddress address = InetAddress.getByName("antonie.com");
            System.out.println(address);
            InetAddress[] addresses = InetAddress.getAllByName("localhost");
            for(int i=0;i<addresses.length;i++)
                System.out.println(addresses[i]);
            InetAddress mesin = InetAddress.getLocalHost();
            System.out.println(mesin);
            String lokal = mesin.getHostName();
            String ip = mesin.getHostAddress();
            System.out.println(lokal);
            System.out.println(ip);
        } catch (UnknownHostException uhe){ }
    }
}
```

```
localhost/127.0.0.1
arc/192.168.1.146
arc
192.168.1.146
Press any key to continue...
```

NSLookup Clone

- Menggunakan InetAddress
- Lihat HostLookup.java

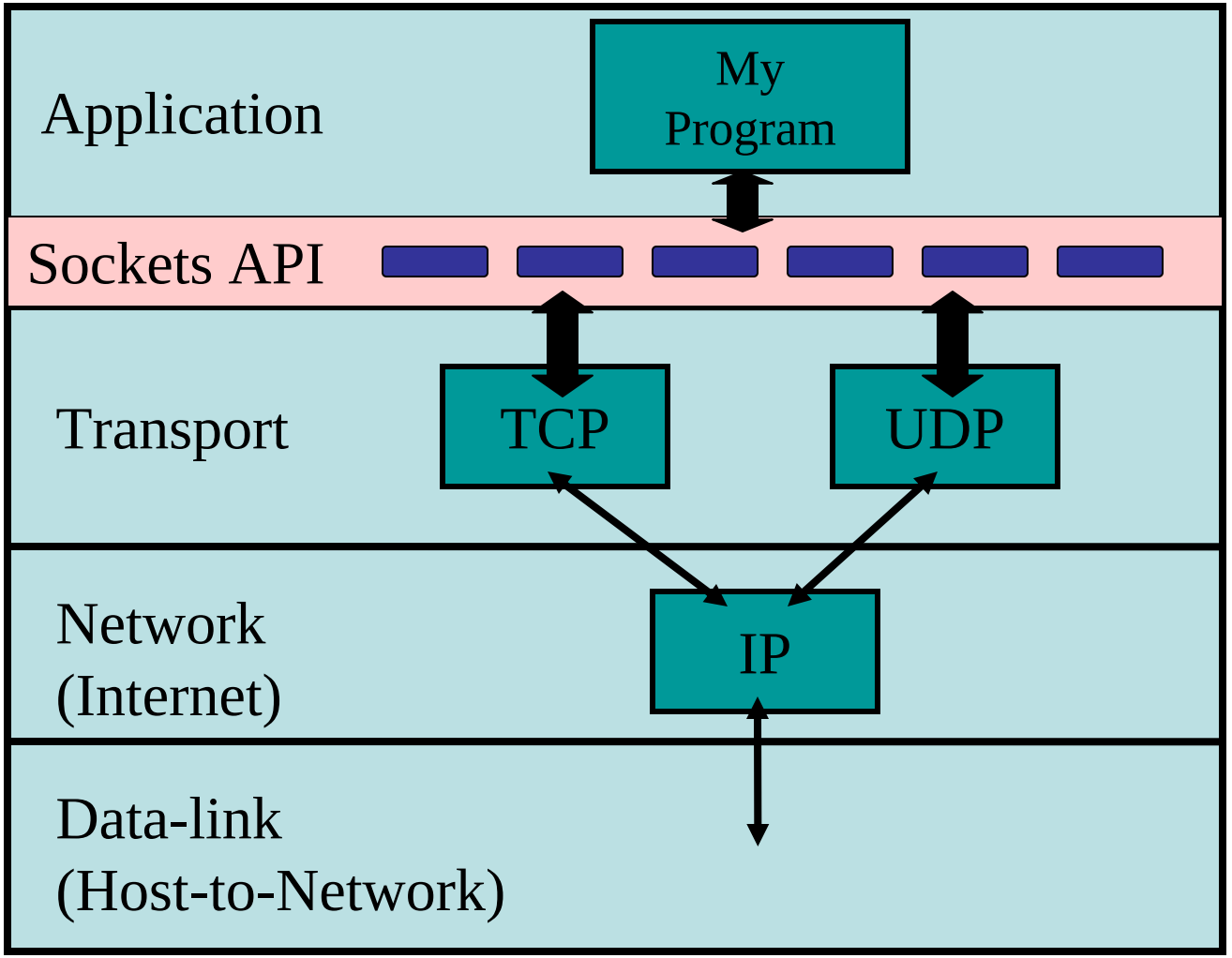
Informasi Antar muka Jaringan

- Untuk mendapatkan informasi network interface pada Java telah terdapat kelas NetworkInterface yang mampu mendapatkan informasi tentang antar muka jaringan, nama device, dan IP yang ter-bind. Nama device misalnya eth0, lo0
- Contoh: DisplayNet.java

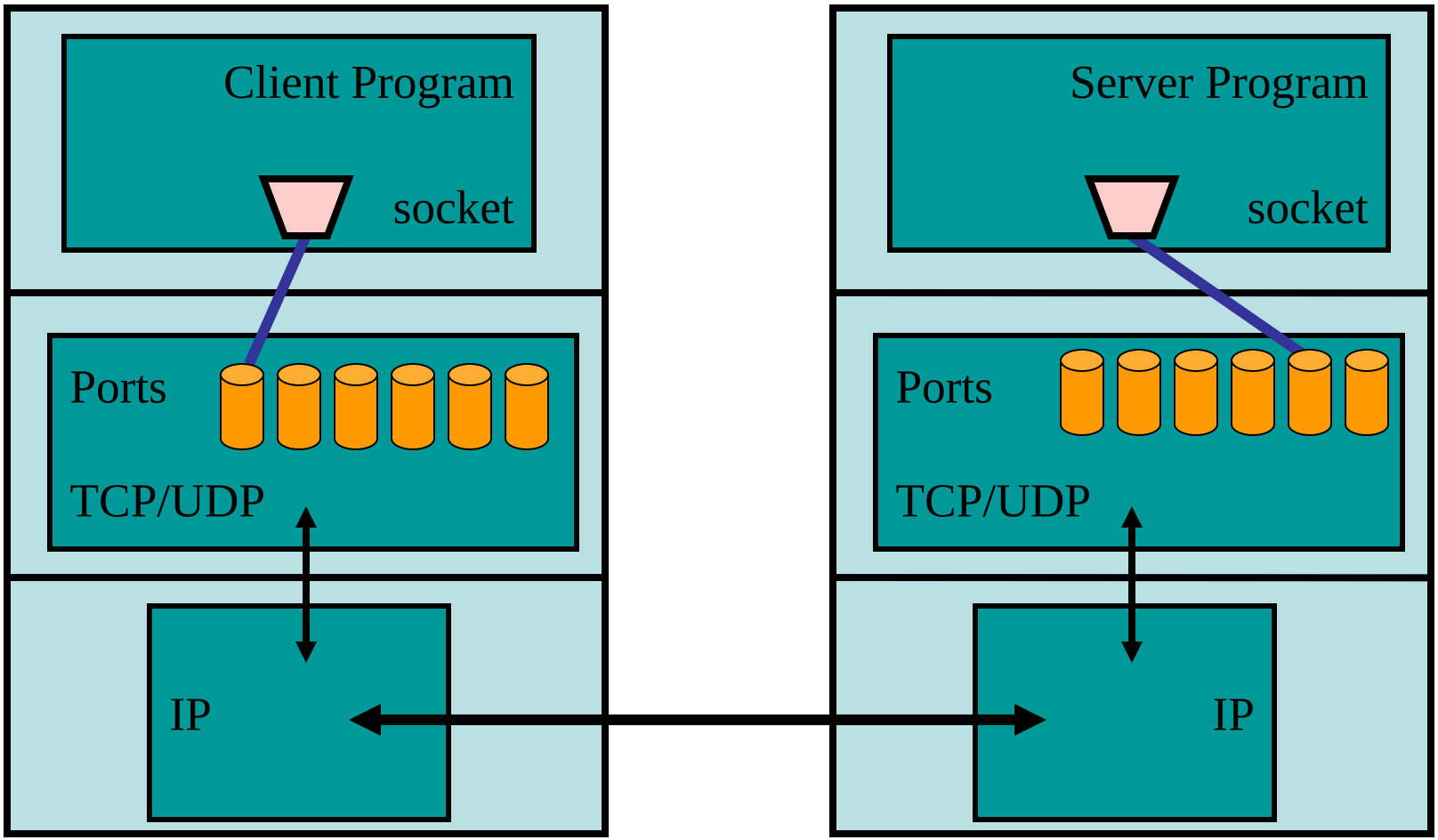
Connection-oriented & Connectionless Socket

- A socket programming construct can make use of either the UDP (User Datagram Protocol) or TCP (Transmission Control Protocol).
- Sockets that use UDP for transport are known as ***datagram sockets***, while sockets that use TCP are termed ***stream sockets/TCP sockets***.

IP Protocol Suite



Socket to Socket Communication



Review: TCP vs. UDP

- TCP
  - Connection-oriented
  - Reliable
  - Stateful
- UDP
  - Connectionless
  - Unreliable
  - Stateless

Review: Ports

- Used to differentiate applications running on same host (address)
- Represented as 16-bit integer
  - Well-known ports: 0 – 1023
  - Registered: 1024 – 49151
  - Dynamic Ports: 49152 - 65535

Java Socket & ServerSocket class

- Untuk Socket Connection Oriented Programming
  - This means that the connection between server and client remains open throughout the duration of the dialogue between the two and is only broken (under normal circumstances) when one end of the dialogue formally terminates the exchanges (via an agreed protocol)
- Kelas java.net.ServerSocket digunakan oleh Server untuk listen koneksi
- Kelas java.net.Socket digunakan oleh Client untuk inisialisasi koneksi.
- Setelah client terkoneksi ke server dengan menggunakan Socket, maka ServerSocket akan mengembalikan status server ke client melalui koneksi yang terbentuk sebelumnya.

The Java.net.Socket Class

- Connection is accomplished through the constructors. Each Socket object is associated with exactly one remote host. To connect to a different host, you must create a new Socket object.

  public Socket(String host, int port) throws UnknownHostException, IOException

  public Socket(InetAddress address, int port) throws IOException

  public Socket(String host, int port, InetAddress localAddress, int localPort)
  throws IOException

  public Socket(InetAddress address, int port, InetAddress localAddress, int localPort) throws IOException

- **Sending and receiving data is accomplished with output and input streams. There are methods to get an input stream for a socket and an output stream for the socket**.

  public InputStream getInputStream() throws IOException

  public OutputStream getOutputStream() throws IOException

- **There's a method to close a socket:**

  public void close() throws IOException

The Java.net.ServerSocket Class

- The *java.net.ServerSocket* class represents a server socket. It is constructed on a particular port. Then it calls accept() to listen for incoming connections.
  - accept() blocks until a connection is detected.
  - Then accept() returns a java.net.Socket object that is used to perform the actual communication with the client.

public ServerSocket(int port) throws IOException
public ServerSocket(int port, int backlog) throws IOException
public ServerSocket(int port, int backlog, InetAddress bindAddr) throws IOException

public Socket accept() throws IOException
public void close() throws IOException

Prinsip ServerSocket

- Create a ServerSocket object.
  - ServerSocket servSock = new ServerSocket(1234);
- Put the server into a waiting state.
  - Socket link = servSock.accept();
- Set up input and output streams.
  - Scanner input = new Scanner(link.getInputStream());
  - PrintWriter output = new PrintWriter(link.getOutputStream(),true);
- Send and receive data.

  output.println("Awaiting data...");

  String input = input.nextLine();
- Close the connection (after completion of the dialogue).
  - link.close();

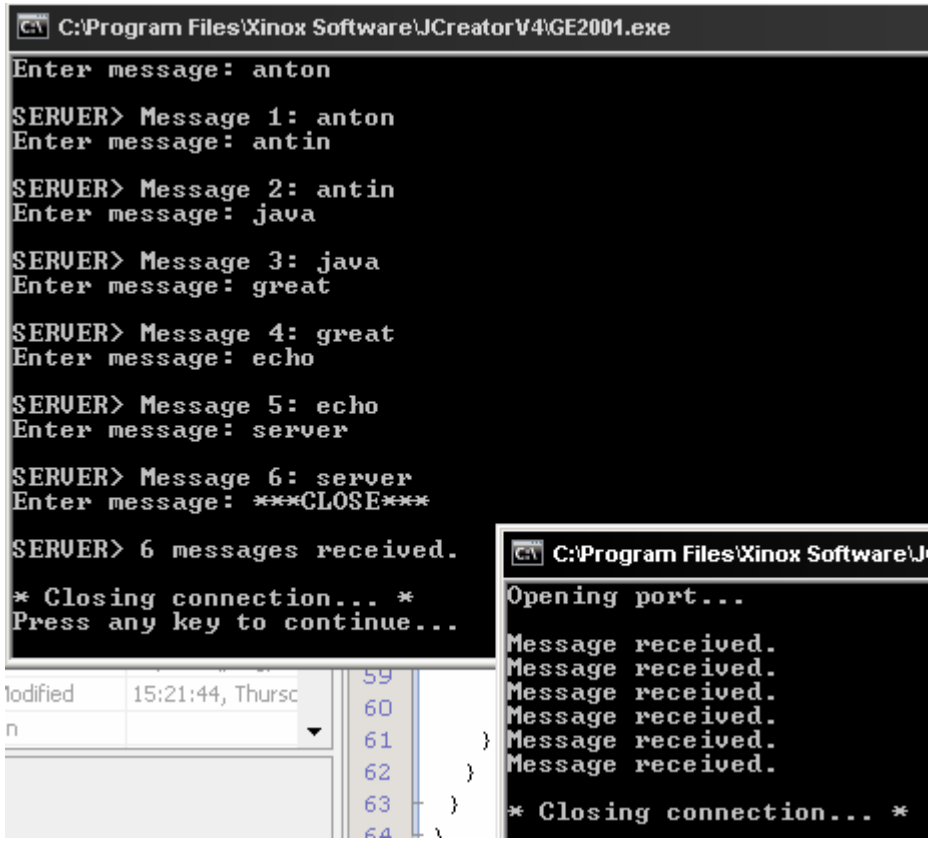
Prinsip Socket (client)

- Establish a connection to the server.
  - the server's IP address (of type InetAddress);
  - the appropriate port number for the service.

    Socket link = new Socket(InetAddress.getLocalHost(),1234);
- Set up input and output streams.
  - Scanner input = new Scanner(link.getInputStream());
  - PrintWriter output = new PrintWriter(link.getOutputStream(),true);
- Send and receive data.
  - The Scanner object at the client will receive messages sent by the PrintWriter object at the server,
  - while the PrintWriter object at the client will send messages that are received by the Scanner object at the server (using methods nextLine and println respectively).
- Close the connection.

Contoh

- TCPEchoServer
- TCPEchoClient



Contoh InfoClient dan InfoServer

- InfoClient.java
- InfoServer.java

Contoh Socket untuk port Scanner

- PortScanner.java

See u next week

- HTTP Socket
- Multithreading ServerSocket dan Socket
- JDBC