

# **JOBSHEET 1**



Introduction  
Programming  
Socket

## **I. PREFACE**

Welcome to this learning guide on Socket Programming! This guide has been designed to help you understand the fundamentals of Socket Programming and how to develop networked applications using sockets. Whether you are a beginner or an experienced programmer, this guide will provide you with valuable insights, practical tips, and resources to enhance your learning experience. By following this guide, you will be able to develop networked applications and explore the vast possibilities of Socket Programming.

## **II. LEARNING OBJECTIVES**

Upon completing this learning guide, you will be able to:

- Understand the basics of Socket Programming
- Develop client-server applications using sockets
- Implement different types of sockets and their functionalities
- Handle errors and exceptions in Socket Programming
- Explore advanced Socket Programming concepts such as multi-threading and asynchronous I/O

## **III. TOOLS AND MATERIALS**

To follow this learning guide, you will need the following tools and materials:

- A computer with a programming language such as Python, Java, or C/C++ installed
- A text editor or an Integrated Development Environment (IDE)
- A basic understanding of programming concepts such as variables, data types, functions, and control structures
- Access to the internet to download and install necessary libraries and modules

## **IV. INTRODUCTION**

Socket Programming is a fundamental concept in networked communication that enables computers to exchange data over a network. It provides a way for applications to communicate with each other, regardless of their location and operating systems. Socket Programming allows developers to build client-server applications, where clients can request data from servers or send data to them. In this learning guide, we will explore the basics of Socket Programming and its various components. We will also develop client-server applications using sockets and learn how to handle errors and exceptions.

## V. HANDLING COMPUTER NAME AND IP WITH JAVA

In dealing with computer names and IP addresses using Java, a distributed system consists of a group of standalone computers that work together, possibly sharing computational resources such as processors, memory, and storage to provide services to users. To communicate with other computers, each computer is assigned an IP address, such as 192.168.29.251, at the application level. However, recognizing computer addresses based on this numerical notation is not easy for humans (programmers!).

Therefore, network infrastructure provides naming services, such as DNS (Domain Naming Service) on the internet or intranet, which store a dictionary of user-friendly computer names and their IP addresses. The task of such naming services is to translate a user-friendly computer address, such as `www.google.com`, into an IP address, such as `66.102.7.104`.

```
# Copyright (c) 1993-1999 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
# 102.54.94.97 rhino.acme.com # source server
# 38.25.63.10 x.acme.com # x client host

127.0.0.1 localhost
192.168.29.21 jarkom-01
```

A similar service exists on local computers, called a resolver, which stores translation dictionaries in a special file (on Windows XP, this file is usually located at `C:\Windows\system32\drivers\etc\hosts`). If a local computer is asked to contact a computer name, it will attempt to translate the computer name into the actual IP address using the resolver. If the computer name entry is not found in the hosts file, the local computer will then attempt to contact the DNS service (based on the DNS Server setting on the computer). Java provides an `InetAddress` object that utilizes the operating system and network services to perform computer name and IP address translations. The demo below shows how to use the `InetAddress` object to retrieve the local computer's IP address (D1-1) and name (D1-2), perform IP to computer name translation (D1-3), and name to IP translation (D1-4) (similar to the `NSLookup` shell command).

## PRACTICUM

- **Practicum 1 - Retrieve your computer's IP automatically**

➤ Create the program below, save it with the name getIP.java

```
import java.net.*;

public class getIP {
    public static void main(String args[]) throws Exception {
        InetAddress host = null;
        host = InetAddress.getLocalHost();
        byte ip[] = host.getAddress();

        for (int i=0; i<ip.length; i++) {
            if (i > 0) {
                System.out.print(".");
            }
            System.out.print(ip[i] & 0xff);
        }
        System.out.println();
    }
}
```

<i>import java.net.*;</i>	This line imports the necessary classes from the java.net package, including the InetAddress class, which is used to represent IP addresses.
<i>host = InetAddress.getLocalHost();</i>	This line retrieves the IP address of the local host and assigns it to the "host" variable.
<i>byte ip[] = host.getAddress();</i>	This line retrieves the IP address as an array of bytes and assigns it to the "ip" variable.
<i>for (int i=0; i&lt;ip.length; i++) {</i>	This line starts a for loop that iterates over each byte in the "ip" array.
<i>if (i &gt; 0) { System.out.print("."); }</i>	This conditional statement checks if it is not the first byte in the IP address and prints a dot (".") as a separator before printing the byte.
<i>System.out.print(ip[i] &amp; 0xff);</i>	This line prints the value of the current byte in the IP address. The bitwise AND operation with 0xff is used to ensure that the byte is treated as an unsigned value.

Compile the above program, run it and observe the results

```
javac getIP.java
java getIP
```

- **Practicum 2 - Get the hostname of the computer**

➤ Create the program below, save it with the name getName.java

```
import java.net.*;

public class getName {
    public static void main(String args[]) throws Exception {
        InetAddress host = null;
        host = InetAddress.getLocalHost();
        System.out.println("Nama komputer Anda: " +
            host.getHostName());
    }
}
```

*InetAddress host =  
InetAddress.getLocalHost();*

Membuat objek InetAddress yang mewakili alamat IP lokal atau localhost menggunakan metode statis getLocalHost().

*return host.getHostName();*

Mengembalikan nama host yang diperoleh dari objek InetAddress menggunakan metode getHostName(). Metode ini mengambil nama host yang sesuai dengan alamat IP yang diberikan.

➤ Compile and run the above program (see D1-1) and observe the results

```
javac getName.java
java getName[]
```

- **Practicum 3 - Converts the IP address to the associated hostname**

➤ Create the program below with the name IPtoName.java

```
import java.net.*;

public class IPtoName {
    public static void main(String args[]) {
        if (args.length == 0) {
            System.out.println("Pemakaian: java IPtoName <IP address>");
            System.exit(0);
        }
        String host = args[0];
        InetAddress address = null;

        try {
            address = InetAddress.getByName(host);
        }
        catch (UnknownHostException e) {
            System.out.println("invalid IP - malformed IP");
            System.exit(0);
        }
    }
}
```

```

    }
    System.out.println(address.getHostName());
}
}

```

<i>if (args.length == 0) {     System.out.println("Pemakaian: java IPtoName &lt;IP address&gt;");     System.exit(0);</i>	Jika tidak ada argumen yang diberikan, pesan ini akan dicetak ke konsol untuk memberi tahu pengguna cara menggunakan program dengan benar.
<i>String host = args[0];</i>	Menyimpan alamat IP yang diberikan sebagai argumen baris perintah ke dalam variabel host.
<i>try {</i>	Memulai blok try, yang digunakan untuk menangkap dan menangani pengecualian yang mungkin terjadi.
<i>address = InetAddress.getByNome(host);</i>	Menggunakan metode statis getByNome() dari kelas InetAddress untuk mendapatkan objek InetAddress yang mewakili alamat IP yang diberikan.
<i>catch (UnknownHostException e) {:</i>	Jika terjadi pengecualian UnknownHostException selama mencari alamat IP yang sesuai dengan nama host, program akan melompat ke blok catch yang sesuai.
<i>System.out.println("invalid IP - malformed IP");</i>	Pesan ini akan dicetak ke konsol jika terjadi pengecualian UnknownHostException, menunjukkan bahwa alamat IP yang diberikan tidak valid atau salah format.
<i>System.out.println(address.getHostName());</i>	Jika tidak ada pengecualian yang terjadi, metode getHostName() dari objek address akan digunakan untuk mendapatkan nama host yang sesuai dengan alamat IP yang diberikan, dan kemudian nama host tersebut akan dicetak ke konsol.

- Compile and run with local computer IP arguments and other computers

```

javac IPtoName.java
java IPtoName 192.168.1.8

```

- **Practicum 4 - Retrieve IP address associated with the provided hostname**

- Create this program, save it with the name NsLookup.java

```

import java.net.*;

public class NsLookup {
    public static void main(String args[]) {

        if (args.length == 0) {
            System.out.println("Pemakaian java NsLookup <hostname>");
            System.exit(0);
        }

        String host = args[0];
        InetAddress address = null;
    }
}

```

```

    try {
        address = InetAddress.getByName (host);
    }
    catch(UnknownHostException exception){
        System.out.println("Unknown host");
        System.exit(0);
    }
    byte[] ip = address.getAddress();
    for (int i=0; i<ip.length; i++){
        if (i>0) System.out.print(".");
        System.out.print((ip[i] & 0xff));
    }
    System.out.println();
}
}

```

<pre> if (args.length == 0) {     System.out.println("Pemakaian java NsLookup &lt;hostname&gt;");     System.exit(0); } </pre>	<p>Jika tidak ada argumen yang diberikan, pesan ini akan dicetak ke konsol untuk memberi tahu pengguna cara menggunakan program dengan benar.</p>
<pre>String host = args[0];</pre>	<p>Menyimpan nama host yang diberikan sebagai argumen baris perintah ke dalam variabel host.</p>
<pre>address = InetAddress.getByName(host);</pre>	<p>Menggunakan metode statis getByName() dari kelas InetAddress untuk mendapatkan objek InetAddress yang mewakili alamat IP yang sesuai dengan nama host yang diberikan.</p>
<pre>byte[] ip = address.getAddress();</pre>	<p>Mengambil alamat IP dalam bentuk array byte dari objek address menggunakan metode getAddress() dan menyimpannya dalam variabel ip.</p>
<pre>for (int i=0; i&lt;ip.length; i++){</pre>	<p>Melakukan iterasi melalui setiap byte dalam array alamat IP.</p>
<pre>if (i&gt;0) System.out.print(".");</pre>	<p>Mencetak tanda titik setelah byte pertama dalam alamat IP.</p>
<pre>System.out.print((ip[i] &amp; 0xff));</pre>	<p>Mencetak nilai setiap byte dalam alamat IP setelah mengkonversinya menjadi bilangan tak bertanda dengan menggunakan operator bitwise &amp; dan mask 0xFF.</p>
<pre>System.out.println();</pre>	<p>Mencetak baris baru setelah mencetak alamat IP secara lengkap.</p>

➤ Compile and run (try different hostnames)

```

javac NsLookup.java
java NsLookup DESKTOP-8LPT4D5

```

- Enter for example www.detik.com then the address will be displayed IP from www.youtube.com. Enter the host name of any other website, what is the IP address?