CHAPTER 4
ANALYSIS AND DESIGN

4.1 Analysis

In the communication network, the data transmission process using reference model that has been standardized by a standardization body that is ISO (International Organization for Standardization). Reference model is OSI (Open Systems Interconnection). According to (Bora et al. 2014) An Open System Interconnection model commonly known as an OSI model ratified in 1984, is like an INTERFACE between two parties i.e., one sender and other receiver . The purpose is that all the learning and development process of communication networks around the world have the same guidelines. Computers can exchange data by traversing the seven OSI layers. The process of packet delivery based on the OSI model can be illustrated as follows:

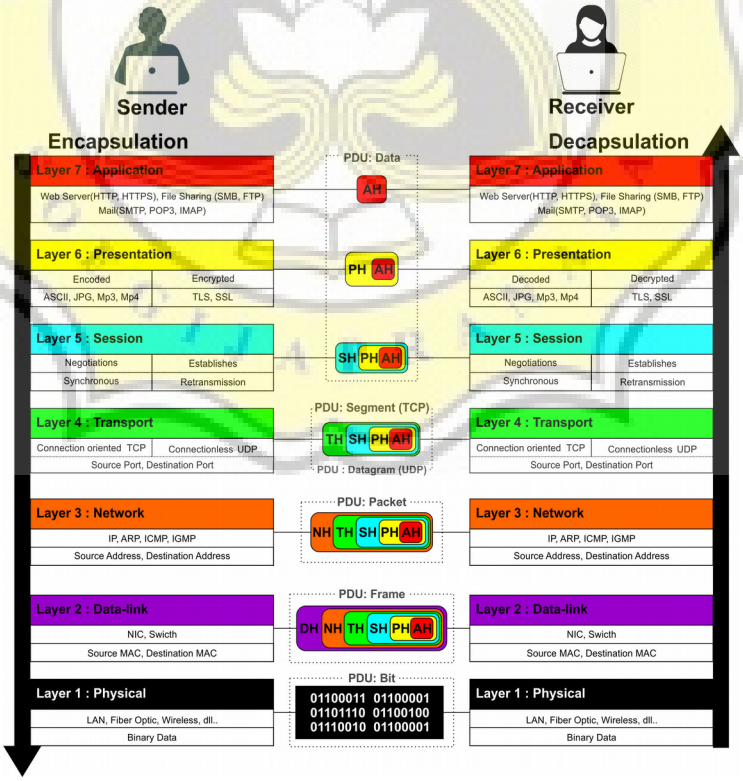

Illustration 4.1: Packet data delivery process in the OSI model.

Based on the above illustration, Can be explained as follows :

- **Application Layer**

  The process begins with the user interacting with applications that implement communication components such as web browsers, email, file sharing or other network applications to send data.

  Example: HTTP, HTTPS, FTP, SMB, SMTP.

- **Presentation Layer**

  In this layer, data from the sender application layer can be translated into a common format. Then the general format is translated to a format known by the receiving application layer.

  Example: ASCII, JPG, GIF,MIDI.

- **Session Layer**

  Session layer is a bridge for presentation layer and transport layer. Responsible in building, maintaining and synchronizing the interaction between communication systems.

  Example: NetBIOS.

- **Transport Layer**

  Entering the transport layer, the data has additional information about the protocol used. This information is often referred to as Header. Then the data is encapsulated by adding headers in the source port and destination port. This layer uses two protocols namely TCP (transmission control protocol) and UDP (user datagram protocol). UDP is a Connectionless protocol (the process of sending data without any responsibility if a data error occurs). While TCP is a Connection oriented protocol (a relationship

in charge of data transmitted). The data unit protocol for TCP is called Segment whereas UDP is called Datagram.

Example: TCP(80),TCP(21),TCP(445), UDP(53).

- **Network Layer**

  In the Network layer, there is the addition of headers in the form of IP Address source and Destination IP Address. Data protocol of the blessed unit becomes a Packet.

  Example: Router

- **Data-link Layer**

  Part of datalink layer occurs the addition of headers like MAC source, destination MAC and protocol field. Finally the packet turns into a data unit called a frame. The data-link layer can also detect errors and retransmit for failed frames, encode the packets into multiple bits and decode again in the recipient's data-link layer.

  Example : Swicth.

- **Physical Layer**

  In the Physical Layer, data packets are sent in the form of electrical voltage. The electrical voltage itself is binary data that has been translated into a set of numbers 0 and 1. Through transmission media, the packet begins its journey to the Physical Layer at the receiving end and the whole process will reverse and the packet returns to the application layer of the receiving end of the data.

  Example : HUB, LAN, Fiber Optik dan Nirkabel atau Wireless.

In the OSI seven-layer network model, the term used to group information added or removed by each layer of the OSI model is the Data Protocol Unit (PDU). In Layer 1, a PDU is called a bit, in Layer 2 called a frame, in Layer 3 is called a packet. In Layer 4 the PDU is referred to as a segment for the TCP protocol and datagram for the UDP protocol. In layer 5 upwards, the PDU is called data. Here is an encapsulation PDU illustration for TCP / IP or UDP / IP communications via Ethernet:

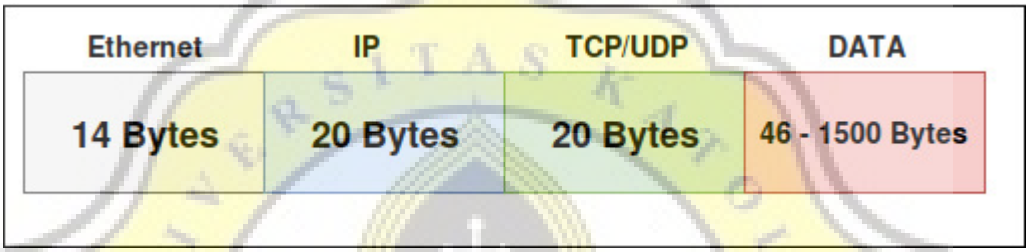

Illustration 4.2: PDU Encapsulation

An Ethernet frame is a part of the network packet at the data-link layer with a maximum size of 1514 bytes. Ethernet frame starts with the Ethernet header which contains the 6 byte for destination MAC address (Media Access Control Address) and the 6 byte for source MAC address. Resumed EtherType field size 2 byte serves to indicate the type of protocol on the network layer used. The center of the frame is the payload data for IP headers wrapped in Frame. EtherType values for some notable protocols are defined by IETF (Internet Engineering Task Force) such as: IPv4 (Internet Protocol Version 4) value is 0x0800, IPv6 (Internet Protocol Version 6) value it is 0x86DD and ARP (Address Resolution Protocol) value it is 0x0806. This project focuses on IPv4 protocol type. The reason is because IPv4 is the most widely used today. IPv4 is described in IETF publication RFC 791 (September 1981), replacing an earlier definition (RFC 760, January 1980). The IPv4 format is a header information consisting of several fields as follows:

Table 4.1: IPv4 Header Fields

Field	Size (bits)	Description
Version	4	Used to indicate which IP version to use.
Internet Header Length (IHL)	4	Used to indicate IP header size.
Type Of Service (TOS)	8	Used to implement a fairly simple QoS (Quality of Service).
Total Length	16	Length of Packet IP (including IP and IP Payload header). Packet size max = 2 ^ 16 = 65,535 Bytes
Identification (Fragmen ID)	16	Identify the original packet of these fragments, to help reassemble fragmented packets.
Flag	3	The first bit is reserved and must be zero. The second bit is the DF flag (Do not Fragments). If DF is set, the packet can not be fragmented. The third bit is Flag MF (More Fragments). If MF is set, there are more fragments to come. The unfragmented packet of the MF flag is set to zero.
Fragment Offset	13	Used in reassembly of fragmented packets.
Time to Live (TTL)	8	To prevent circling packets indefinitely across networks with routing loops. The maximum TTL value is 255.
Protocol	8	Defines the next header with protocol number such as Number 6 defines TCP and number 17 defines UDP Header.
Checksum	16	To protect data packets from corrupt.
Source Address	32	It contains the IPv4 address of the sender.
Destination Address	32	Contains an IPv4 address from the recipient.
Options (0-40 Byte)	0-40	This is an optional field that can be used or not. If there is an option in the header then the first byte is represented.

The next header of the IPv4 header is determined by the protocol number. The protocol number is a protocol that has been managed and assigned by IANA (Internet Assigned Numbers Authority). Some of the most commonly used protocol numbers are ICMP (Internet Control Message Protocol) value is 1, TCP (Transmission Control Protocol) value is 6 and UDP (User Datagram Protocol) value is 17. The TCP header consists of several fields as follows:

Table 4.2: TCP Header Fields

Field	Size (bits)	Description
Source port	16	Identifies the Source Port number (Sender Port).
Destination Port	16	Identify the Destination Port number (Receiver Port).
Sequence number	32	Sequence is responsible for ensuring that all IP packets sent are actually received.
Acknowledgment	32	Shows the next sequence number expected of the shipping device from another device.
Offset Data	4	The offset data field stores the total TCP header size in multiples of four bytes. Headers that do not use TCP optional fields have offset 5 data (representing 20 bytes), whereas headers that use optional fields of maximum size have offset data 15 (represents 60 bytes).
Reserverd	6	Reserverd always has a zero value. This field serves to align the total header size as a multiple of byte (essential for computer data processing efficiency).
Flags	6	TCP headers use six control flags and each bit represents both on and off values or 1 and 0. The flags are: URG (Urgent), ACK (Acknowledgment), PSH (Push), RST (Reset), SYN (Synchronize), And FIN (Finish).
Window Size	16	This field shows how many octets can be received at once. This value represents how much host recipient buffer is available for

		temporary storage of this IP packet.
Checksum	16	Used to verify data integrity.
Urgent Pointer	16	Can be used to mark messages that require priority processing. However this field is usually set to zero or ignored.
Optional	32	Serves as a container of some additional TCP options. Each TCP option will take up 32 bits of space, so the TCP header size can be indicated by using the Data offset field.

UDP header consists of several fields as follows:

Table 4.3: UDP Header Fields

Field	Size (bits)	Description
Source port	16	Identifies the Source Port number (Sender Port).
Destination port	16	Identify the Destination Port number (Receiver Port).
Length	16	Indicates the length of UDP messages
Checksum	16	Contains integrity checking information from UDP messages

The last part of the header is the data carried by UDP and TCP. The TCP field contains segments of data from user applications, such as email sections or web pages. While UDP contains datagrams of applications such as DNS (Domain Name System), SNMP (Simple Network Management Protocol) and SNMP (Simple Network Management Protocol).

4.2 Design

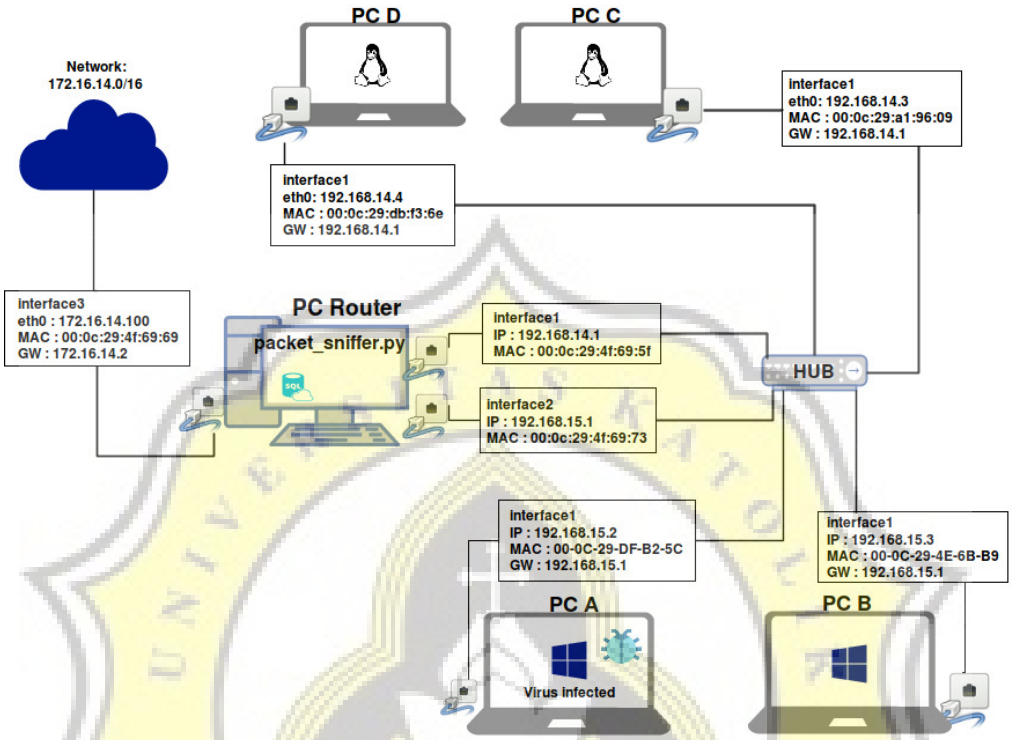

Illustration 4.3: Network topology for running Packet Sniffer program

In the above network topology illustration, there are five PC (Personal Computer) and one HUB. HUB serves to connect the five computers using an Ethernet cable. Two pc with OS (Operting System) Windows logo named "PC A" and "PC B". Both PCs are on the network 192.168.15.0/24. While the two pc with OS logo (Operting System) Linux which is named "PC C" and "PC D" is in network 192.168.14.0/24. The two different networks can not communicate without the help of a router. PC is located in the middle between the two networks is the Router. The PC is named "PC Router". PC Router connects two different networks and allow all PCs on this topology accsess to the internet

Packet sniffer works by intercepting and recording network traffic that passes through the network interface. If all of PCs wants to access the internet,

them must go through the network interface on "PC Router". The network interface to be passed "PC A" and "PC B" is named "interface2". The network interface to be passed "PC C" and "PC D" is named "interface1". This topology suitable for packet sniffer programs, as it can capture network traffic passing through the network interface. So Packet sniffer will be installed on "PC Router".

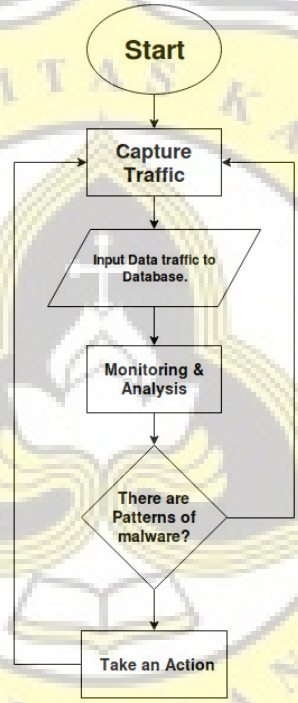

Illustration 4.4: Packet Sniffer Flowchart

The flowchart above illustrates how the packet sniffer program works. It starts with capturing network packet traffic consisting of Ethernet Header, IP header, TCP Header and UDP Header. Then the network packets are saved to the database. Data stored in the database is used on web-based programs for monitoring and analysis.

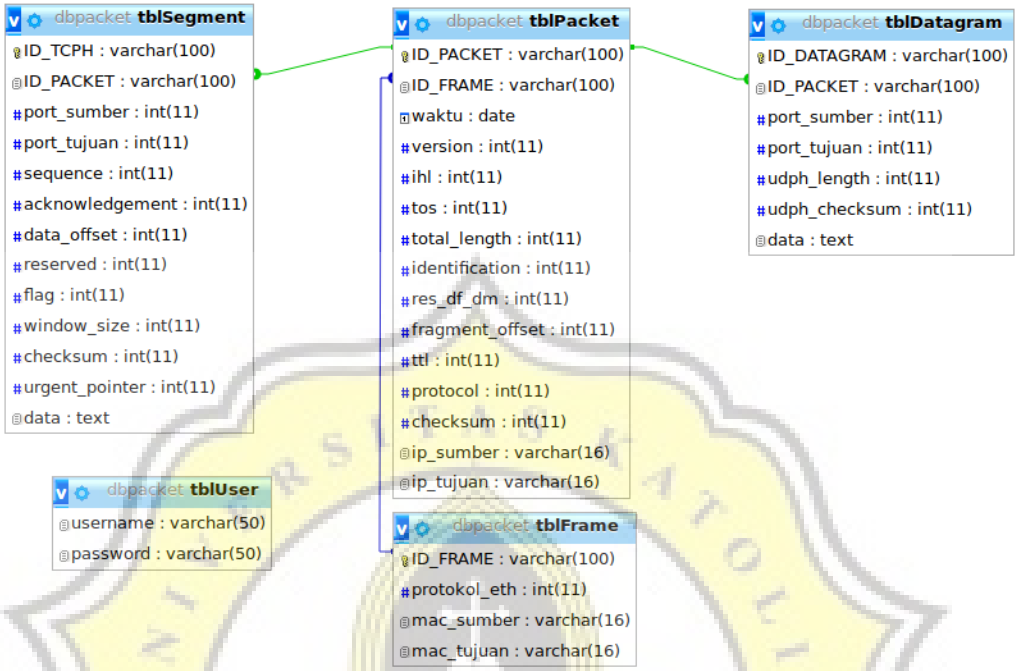

Illustration 4.5: Packet Sniffer Database

The packet sniffer database has five tables in it: tblUser to login as administrator, tblFrame to fill data from ethernet frame header, tblPacket to fill data from IP header, tblSegment to fill data from TCP header and tblDatagram to fill data from UDP header.