



## Efektivitas Sniffer Menggunakan Natural Language dalam Pembelajaran Lalu Lintas Jaringan Komputer

Putu Adhika Dharmesta<sup>1</sup>, I Made Agus Dwi Suarjaya<sup>2</sup>, I Made Sunia Raharja<sup>3</sup>

<sup>1,2,3</sup>Program Studi Teknologi Informasi, Fakultas Teknik, Universitas Udayana

<sup>1</sup>adikadarmesta8@gmail.com, <sup>2</sup> agussuarjaya@it.unud.ac.id, <sup>3</sup> sunia.raharja@unud.ac.id

### Abstract

Computer networks are currently very active in the development of technology that is around us. Seeing this, of course knowledge of the network will be needed if there is a problem on the network. Scapy is a Python module that allows for sending, sniffing and dissecting a packet on a network. This capability allows users to create an application that can dissect how the workings of a network packet. Researchers will create a protocol traffic learning application on a computer network using Scapy and natural language to convey the results of the ongoing sniffing process. The application uses natural language to convey the translation of the sniffing process. The translation result of the sniffing process by using the natural language of this application is expected to be effective and can facilitate and make users understand and learn about the work process of a network packet. To measure the effectiveness of the use of natural language for the translation of the sniffing process a questionnaire was distributed to students of the SMKN 1 Denpasar school majoring in Computer and Network Engineering. The results of the distribution of the questionnaire were then calculated using a Likert scale and then the results obtained that the original results of the sniffing process got a Likert scale value of 37%. While the results of sniffing that have been translated get a value of 73%. This shows respondent better understands the results that have been translated compared to the original results that have not been translated.

**Keywords:** Computer Network, Sniffing, Natural Language, Scapy, Python

### Abstrak

Jaringan komputer saat ini sangat berperan aktif dalam perkembangan teknologi yang ada di sekitar kita. Semua perangkat teknologi saat ini selalu terkoneksi pada suatu jaringan internet. Melihat hal tersebut tentunya pengetahuan mengenai jaringan akan sangat dibutuhkan apabila terjadi sebuah permasalahan pada jaringan. Scapy adalah sebuah modul Python yang memungkinkan untuk mengirim, *sniffing* dan membedah sebuah paket pada sebuah jaringan. Kemampuan ini memungkinkan pengguna untuk membuat sebuah aplikasi yang dapat membedah bagaimana cara kerja dari suatu paket jaringan. Peneliti membuat sebuah aplikasi pembelajaran lalu lintas protokol pada sebuah jaringan komputer dengan menggunakan Scapy dan *natural language* untuk menyampaikan hasil dari proses *sniffing* yang berlangsung. Aplikasi tersebut menggunakan *natural language* untuk menyampaikan hasil translasi proses *sniffing*. Hasil translasi proses *sniffing* dengan menggunakan *natural language* dari aplikasi ini efektif dan dapat memudahkan dan membuat pengguna mengerti dan belajar mengenai proses kerja sebuah paket jaringan. Untuk mengukur efektivitas dari penggunaan *natural language* terhadap hasil translasi proses *sniffing* dilakukan penyebaran kuisioner terhadap siswa sekolah SMKN 1 Denpasar jurusan Teknik Komputer dan Jaringan. Hasil penyebaran kuisioner kemudian dihitung dengan menggunakan skala likert yang kemudian didapatkan hasil bahwa hasil asli proses *sniffing* mendapatkan nilai skala likert sebesar 37%. Sedangkan hasil *sniffing* yang sudah ditranslasi mendapatkan nilai 73%. Hal ini menunjukkan responden lebih memahami hasil yang sudah ditranslasi dibandingkan dengan hasil asli yang belum ditranslasi sehingga terbukti penerapan *natural language* dalam *sniffer* efektif dalam meningkatkan pemahaman jaringan komputer.

**Kata kunci:** Jaringan Komputer, *Sniffing*, *Natural Language*, Scapy, Python

## 1. Pendahuluan

Terjadinya arus globalisasi yang sangat besar saat ini membuat serangkaian perkembangan ilmu pengetahuan dan teknologi kian pesat setiap harinya. Untuk itu dibutuhkan sebuah sikap untuk selalu ingin belajar oleh setiap individu untuk dapat tetap relevan di zaman yang bergerak serba cepat seperti saat ini. Apabila dipelajari lebih dalam lagi, belajar dan pembelajaran merupakan dua hal yang sangat berbeda secara konseptual. Apabila mengambil pengertian dari jurnal yang dituliskan oleh Aprida Pane dan Muhammad Darwis Dasopang [1], belajar merupakan suatu kegiatan yang dilaksanakan oleh suatu individu ataupun kelompok yang dilakukan dengan penuh kesadaran untuk meningkatkan kompetensi diri mereka. Berbeda dengan belajar, pembelajaran merupakan suatu proses yang mengatur dan mengorganisir lingkungan yang ada di sekitar peserta didik untuk lebih mengoptimalkan proses belajar dari peserta didik tersebut.

Penyampaian materi pembelajaran tentunya harus menggunakan bahasa sehari-hari untuk memudahkan dalam mengartikan apa yang sedang dipelajari. Peran *natural language* sangat penting untuk diimplementasikan kedalam sebuah aplikasi pembelajaran. Penggunaan *natural language* tidak hanya terbatas pada antarmuka dari aplikasi namun juga dalam hasil yang dikeluarkan oleh aplikasi pembelajaran tersebut. Unit kosakata dan tata bahasa yang sudah diatur sedemikian rupa dapat digunakan untuk antarmuka dan juga hasil yang diberikan oleh sebuah aplikasi komputer.

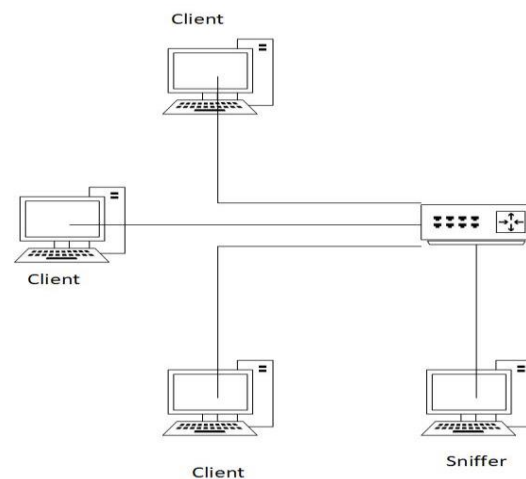
*Natural language processing* adalah sebuah proses pembuatan model komputasi dari bahasa sehari-hari yang biasa digunakan sehingga interaksi antara manusia dengan komputer dapat terjadi dengan lebih natural. Sebuah sistem yang memanfaatkan *natural language processing* ini harus dapat memperhatikan pengetahuan terhadap bahasa yang akan digunakan. Jenis aplikasi yang dibuat dalam pendekatan *natural language* biasanya dalam bentuk *text-based application* dan *dialogue-based application*. Pembentukan kalimat oleh sebuah sistem yang memanfaatkan *natural language processing* harus memperhatikan aturan-aturan penyusunan kalimat suatu bahasa yang akan digunakan. [2].

Pembentukan sebuah kalimat dalam bahasa Indonesia memiliki unsur pembentuk yang saling berangkai dan membentuk suatu makna. Unsur pembentuk tersebut adalah subjek, predikat, objek, pelengkap dan keterangan. Untuk membentuk sebuah kalimat bahasa Indonesia setidaknya ada unsur subjek dan predikat [3].

Jaringan komputer adalah sebuah sistem yang mana terdiri atas komputer-komputer yang didesain untuk dapat

berbagi sumber daya (printer, CPU), berkomunikasi dan dapat mengakses informasi yang diperlukan oleh pengguna (peramban web) [4]. Penggunaan jaringan komputer ditujukan agar suatu perangkat komputer dapat berbagi sumber daya, berkomunikasi dan mengakses informasi antara satu perangkat dengan perangkat lainnya.

Ada banyak cara untuk dapat memahami protokol yang terdapat di dalam jaringan. Salah satunya adalah dengan melihat secara langsung saat protokol pada jaringan tersebut bekerja. Paket *sniffing* adalah salah satu proses membaca paket data yang dikirimkan melalui jaringan [5]. Hal ini dapat dilakukan dengan *software* ataupun perangkat khusus. Mekanisme kerja *sniffer* dapat dilihat pada Gambar 1. Saat paket ditransmisikan dari sumber menuju tujuannya, Paket tersebut akan melalui banyak media perantara [6]. Setiap NIC pada sebuah komputer akan memiliki *physical address* yang unik antara satu perangkat dengan perangkat lainnya. Saat sebuah paket sampai pada suatu NIC, *hardware address* yang berada pada *frame* paket data akan dicocokkan dengan *physical address* yang terdapat pada NIC. Apabila alamatnya sama maka paket akan diterima oleh NIC. Namun apabila berbeda paket tersebut akan dibuang. Tapi apabila NIC dari perangkat diubah menjadi *promiscuous mode* maka NIC nantinya akan dapat menangkap semua paket data yang sampai.



Gambar 1. Cara Kerja Sniffer

Ada beberapa studi yang sudah memanfaatkan paket *sniffer* untuk berbagai kebutuhan penelitian mereka. Apri Siswanto beserta kawan [7] melakukan analisis terhadap lalu lintas aktivitas data pengguna pada jaringan internet di Sekolah Vokasi Telkom Pekanbaru. Data yang didapat oleh pengguna kemudian ditampilkan dalam bentuk grafik sehingga memudahkan pengguna untuk mengetahui penggunaan *bandwidth* dan *user* yang ada pada suatu waktu di dalam jaringan. Kelebihan dari penelitian ini adalah

peneliti mampu menggunakan proses *sniffing* dengan *tool* Wireshark untuk mengetahui beban penggunaan jaringan pada 14 hari penelitian dan dapat memberikan rekomendasi penambahan *bandwith* jaringan untuk meningkatkan layanan. Namun kekurangan dari penelitian ini adalah tidak dijelaskan kisaran waktu akses paling ramai selama periode penelitian

Elamarian [8] dalam penelitiannya melakukan pengujian response time terhadap protokol DNS, HTTP dan ICMP untuk mengetahui waktu yang dibutuhkan untuk mengakses *database brain signal/images* dari berbagai *server* di penjuru dunia. Kelebihan dari penelitian ini adalah penelitian ini menggunakan studi kasus yang cukup umum terjadi. Namun kelemahan dari penelitian ini adalah kesimpulan yang dibuat hanya seputar prosedur penelitian dengan manfaat yang kurang dipaparkan dengan baik.

Penelitian yang dilakukan Piyush Goyal [9] melakukan komparasi dua buah paket *sniffing tool* yang sering digunakan yaitu wireshark dan Tcpcdump untuk melihat yang terbaik diantara keduanya. Kelebihan dari penelitian ini adalah metode pembandingan yang beragam dimulai dari penggunaan power, memori, prosesor, dan kemampuan menganalisis paket. Peneliti juga memberikan kesimpulan yang dapat membantu dalam menentukan penggunaan *tool* untuk melakukan *packet sniffer*. Namun kekurangan dari penelitian ini adalah kurangnya referensi yang memadai.

Praful Saxena dalam penelitiannya [5] melakukan analisa terhadap lalu lintas jaringan dengan menggunakan sebuah *tool* bernama Wireshark untuk membantu seorang *network administrator* dalam mencari kelemahan jaringan yang sedang diawasi. Kelebihan dari penelitian ini adalah pembahasan yang dilakukan terhadap *tool* Wireshark sangat mendalam dan mendasar sehingga dapat membantu pengguna yang baru pertama kali menggunakan Wireshark. Kekurangan dari penelitian ini adalah tidak ditampilkannya hasil *generate report* dari Wireshark seperti yang dituliskan penulis pada bagian abstraknya.

Penelitian yang dilakukan oleh Mohsin Khan dan teman-temannya [10] membahas mengenai bagaimana cara kerja dari protokol DHCP dengan melakukan *capture* Paket DHCP menggunakan Wireshark untuk dianalisa lebih dalam proses serta parameter apa yang terlibat di dalamnya. Kelebihan dari penelitian ini adalah peneliti mampu memberikan contoh mengenai pemanfaatan sebuah *packet sniffer* seperti Wireshark untuk memahami sebuah protokol jaringan seperti DHCP dengan lebih mendalam. Kekurangan dari penelitian ini adalah kurang rapinya penulisan jurnal.

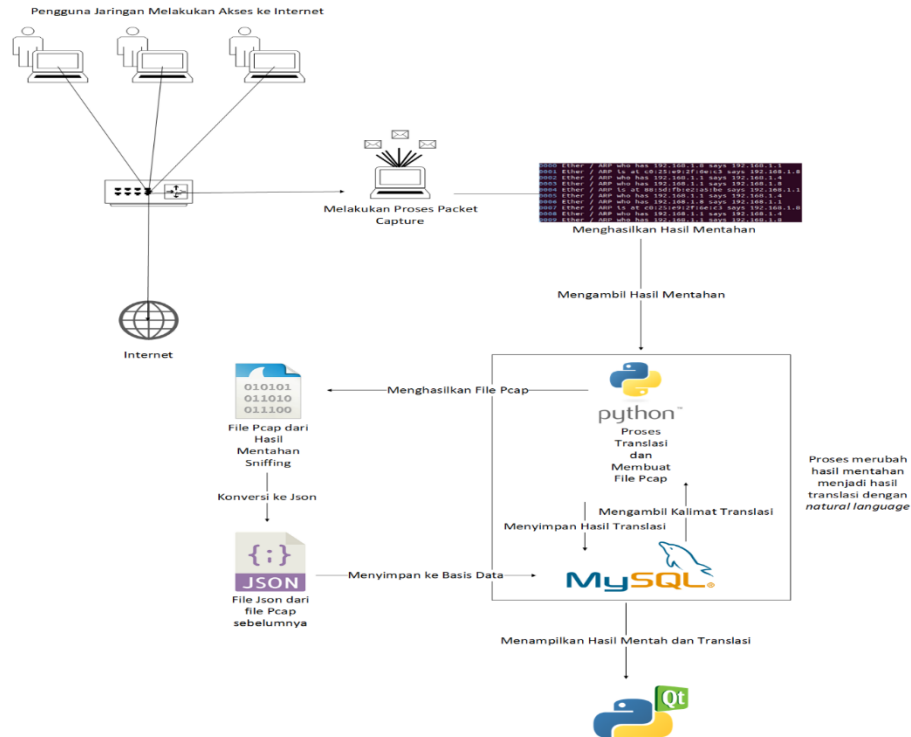
Penelitian lainnya yang dilakukan oleh Mahesh Kumar beserta Rakhi Yadav [11] membahas mengenai alur paket

dari protokol TCP dan UDP dengan studi kasus mengirimkan sebuah email. Kelebihan dari penelitian ini adalah penjelasan mengenai mekanisme protokol TCP dan UDP yang dilengkapi dengan studi kasus mengirimkan *e-mail* menggunakan kedua protokol tersebut cukup baik. Kekurangan dari penelitian ini terletak pada kerapian penulisan jurnal dan peletakan gambar yang masih perlu untuk diperbaiki lagi.

Setelah berbagai *state of the art* yang sudah dipaparkan sebelumnya, dapat diperhatikan bahwa secara umum peneliti lebih melakukan pemanfaatan sebuah *packet sniffer* untuk meneliti sebuah protokol ataupun kinerja dari sebuah jaringan komputer. Hanya penelitian yang dilakukan oleh Piyush Goyal [9] yang membahas mengenai perbandingan kinerja dari *packet sniffer* yang umum digunakan. Masih belum banyak peneliti yang melakukan penelitian mengenai *packet sniffer* serta pengaplikasiannya dalam sebuah aplikasi pembelajaran serta melihat efektivitas dari hasil yang diberikan terhadap pengguna yang akan belajar protokol sebuah jaringan komputer. Berbeda dengan penelitian yang dilakukan oleh Piyush Goyal [9], yang membandingkan efektivitas kinerja dari dua *tool* untuk melakukan proses *sniffing* yaitu TCPDUMP dan Wireshark, penelitian ini membuat sebuah aplikasi yang dapat melakukan translasi dari hasil proses *sniffing* dengan menggunakan *tool* Scapy yang kemudian dinilai efektivitas proses translasi tersebut dalam membuat responden lebih memahami proses dari sebuah jaringan komputer berlangsung. Penelitian dengan melakukan *sniffing* pada sebuah jaringan komputer dan kemudian ditranslasikan dengan konsep *Natural Language Processing* ini merupakan suatu inovasi terobosan baru (*novelty*). Penyimpanan kalimat translasi ke dalam basis data memungkinkan dilakukannya perubahan bahasa sesuai dengan kebutuhan untuk dapat membuat pengguna lebih memahami mengenai apa yang ditampilkan oleh aplikasi. Penelitian ini bertujuan untuk menghasilkan sebuah aplikasi pembelajaran yang mampu digunakan untuk memahami mekanisme sebuah protokol jaringan serta secara efektif mampu meningkatkan pemahaman pengguna terhadap suatu protokol jaringan.

## 2. Metode Penelitian

Desain sistem dari aplikasi pembelajaran ini menggunakan Python sebagai Bahasa pemrograman dasarnya. Python dipilih dikarenakan adanya sebuah modul khusus yang dapat membantu dalam melakukan paket *capture* atau *sniffing* di dalam aplikasi ini. Konsep dari proses *sniffing* yang akan dilakukan oleh aplikasi dapat dilihat pada Gambar 1. Terlihat pada gambar bahwa sebuah komputer yang memiliki aplikasi *sniffer* di dalamnya pertama harus terkoneksi dahulu di dalam aplikasi



Gambar 2 Gambaran Umum Sistem

Scapy adalah sebuah program Python yang mampu untuk mengirim, menerima dan menganalisis paket data di dalam sebuah jaringan [12]. Scapy memungkinkan pengguna untuk dapat menangkap sebuah paket data yang berada pada sebuah jaringan. Kegiatan mengambil paket data pada sebuah jaringan ini sering disebut sebagai *sniffing*. Penelitian kali ini akan mencoba untuk menggunakan Scapy untuk melakukan *sniffing* terhadap jaringan komputer wifi yang digunakan. Metode penelitian yang akan dilakukan pada penelitian kali ini dapat dilihat pada Gambar 2.

Terlihat pada gambar 2 bahwa pengguna aplikasi harus terhubung terlebih dahulu ke dalam sebuah jaringan untuk dapat menggunakan aplikasi ini. Hal pertama yang dilakukan oleh aplikasi yang dibuat ini adalah meminta berbagai inputan parameter dari pengguna seperti nama *interface*, jumlah paket yang akan ingin didapatkan, lama waktu pelaksanaan proses *sniffing*, protokol yang ingin untuk ditranslasikan, kemudian nama *file* dari hasil translasi nantinya. Proses *sniffing* kemudian berjalan berdasarkan atas parameter yang sudah diberikan sebelumnya. Sistem kemudian akan melakukan validasi terhadap protokol yang dipilih. Apabila protokol tertentu dipilih maka sistem akan mengambil kalimat translasi dari basis data yang sudah disiapkan sebelumnya. Sistem kemudian akan menyusun kalimat berdasarkan protokol

yang dipilih dan menyimpan hasil translasi tersebut ke dalam sebuah basis data.

#### Kode Program *Sniffing* Dengan Scapy

```
pkt = sniff(iface=net_iface,
            filter=proto_sniff,
            count=int(pkt_to_sniff),
            timeout=int(time_to_sniff),
            prn=paket_log)
wrpcap(file_name + ".cap", pkt)
```

*Sniffing* yang dilakukan terhadap jaringan dilakukan dengan menggunakan modul Scapy seperti pada kode program diatas. Modul ini dapat diaplikasikan ke dalam aplikasi desktop berbasis Python. Untuk melakukan *sniffing* digunakan sebuah fungsi yaitu `sniff()`. Fungsi ini akan menjalankan proses *sniffing* dengan memperhatikan beberapa parameter yang sudah dimasukan seperti *interface* jaringan, filter protokol, jumlah paket, dan waktu pelaksanaan. Parameter `prn` digunakan untuk mentranslasi hasil paket *sniffing* kedalam format yang diinginkan. Fungsi tersebut di simpan ke dalam variabel `pkt` yang nantinya akan digunakan untuk membuat *file pcap* sebagai data *raw* untuk pembanding dengan hasil translasi yang kemudian akan di konversi ke dalam *file json* untuk memudahkan dipindahkan ke dalam basis data.

Proses translasi antara hasil *sniffing* paket ke dalam sebuah kalimat yang sudah dibahasakan sedemikian rupa dengan

mempertahankan *natural language* dilakukan dengan mengidentifikasi terlebih dahulu paket yang masuk. Setelah paket didapatkan oleh sistem, paket kemudian akan diperiksa protokol yang terkandung di dalamnya. Apabila protokol tertentu terdeteksi di dalamnya maka sistem aplikasi akan memberikan kalimat tertentu sesuai dengan status protokol tersebut bekerja.

---

#### Kode Translasi Paket Data Jaringan

---

```
if ARP in packet:
    if packet[ARP].op == 1:
        # Save ke database
        query_id = "select id_tb_hasil from
        tb_hasil where nama_hasil = '%s'" %
        (file_name)
        id_hasil = run_sql_int(query_id)

        translate = "[ARP]\t "
        +run_sql("select kalimat1 from
        arp_packet where flag='op_1'") \
        + str(now) + run_sql("select
        kalimat2 from arp_packet where
        flag='op_1'") + str(
        packet[ARP].psrc) + run_sql("select
        kalimat3 from arp_packet where
        flag='op_1'") +
        str(packet[ARP].pdst)

        query = "INSERT INTO
        `packet_sniff`.`
        tb_det_hasil` (`id_tb_hasil`,
        `hasil_jadi`) VALUES ('%s', '%s')" %
        (id_hasil[0], translate)

        id_row = insert_sql(query)
        rows_id.append(id_row)

        # Save ke File TXT
        print("[ARP]\t " +run_sql("select
        kalimat1 from arp_packet where
        flag='op_1'") + str(now)
        +run_sql("select kalimat2 from
        arp_packet where flag='op_1'") +
        str(packet[ARP].psrc)
        +run_sql("select kalimat3 from
        arp_packet where flag='op_1'") +
        str(packet[ARP].pdst),
        file=sniffer_log)

    elif packet[ARP].op == 2:
        # Save ke database
        query_id = "select id_tb_hasil from
        tb_hasil where nama_hasil = '%s'" %
        (file_name)
        id_hasil = run_sql_int(query_id)

        translate = "[ARP]\t
        " +run_sql("select kalimat1 from
        arp_packet where flag='op_2'") +
        str(now) + run_sql("select kalimat2
        from arp_packet where flag='op_2'")
        +str(packet[ARP].psrc)
        +run_sql("select kalimat3 from
        arp_packet where flag='op_2'") +
        str(packet[ARP].hwsr)

        query = "INSERT INTO
        `packet_sniff`.`
        tb_det_hasil`
        (`id_tb_hasil`,`hasil_jadi`) VALUES
        ('%s', '%s')" % (id_hasil[0],
        translate)

        id_row = insert_sql(query)
```

---

---

```
rows_id.append(id_row)
```

---

```
# Save ke File TXT
print("[ARP]\t "
+run_sql("select kalimat1 from
arp_packet where flag='op_2'") +
str(now) + run_sql("select kalimat2
from arp_packet where flag='op_2'")
+str(packet[ARP].psrc) +
run_sql("select kalimat3 from
arp_packet where flag='op_2'") +
str(packet[ARP].hwsr),
file=sniffer_log)
```

---

Proses translasi dari hasil asli paket menjadi sebuah kalimat yang sudah menerapkan *natural language* dimulai saat suatu paket sudah terdeteksi jenis protokol yang digunakan beserta proses yang terkandung di dalamnya. Seperti terlihat pada kode program diatas, aplikasi akan mengambil kalimat yang dibutuhkan dari basis data berdasarkan protokol dan proses yang terkandung dengan memperhatikan *flag* yang dibawa oleh setiap paket. Kode program memperlihatkan translasi terhadap protokol ARP. Secara umum protokol lain juga menggunakan pola yang sama seperti ARP yaitu identifikasi jenis protokol kemudian dilanjutkan dengan identifikasi *flag* baru selanjutnya proses translasi berupa penulisan kalimat.

Pengambilan kalimat dilakukan dengan menggunakan fungsi `run_sql()` yang nantinya akan menjalankan perintah `sql` dan memasukan *flag* protokol beserta *flag* kedalam perintah `sql`. Hasil pengambilan perintah `sql` biasanya akan berupa *tuple*. Untuk ditampilkan pada aplikasi kalimat tersebut dirubah ke dalam bentuk *string*. Susunan tabel dalam basis data yang digunakan cukup sederhana. Tabel yang terdapat pada basis data diantaranya adalah `arp_paket`, `icmp_paket`, `dns_paket`, `tcp_paket`, `udp_paket`, `dhcp_paket`, `http_paket`. Kalimat yang akan dimasukan ke dalam sistem dimasukan ke dalam masing-masing tabel sesuai dengan nama paket yang merepresentasikan kalimat tersebut. Setiap tabelnya juga tidak memiliki relasi satu sama lain.

Kalimat dari basis data kemudian disusun sedemikian rupa sehingga menjadi kalimat utuh yang dapat merepresentasikan apa yang sedang terjadi pada paket tersebut. Desain *natural language* dari aplikasi ini dapat dilihat dari salah satu contoh dari hasil translasi pada Gambar 3 dimana gambar tersebut memperlihatkan hasil translasi dari protokol ARP. Satu baris dari hasil translasi tersebut sebenarnya memiliki berbagai macam *flag* yang terkandung dalam sebuah paket seperti pada kode program Hasil *Sniffing* Belum Ditranslasi berikut. Untuk membuat sebuah kalimat translasi dari satu paket protokol ARP maka diambil beberapa *flag* penting didalamnya seperti `op`, `psrc` dan `pdst`. *Flag* tersebut kemudian dijadikan dasar untuk membuat kalimat translasi. Kalimat translasi lalu dibuat dengan memperhatikan unsur kalimat dalam bahasa Indonesia dimana setidaknya harus terkandung

subjek dan predikat. Terlihat pada kalimat translasi yaitu “IP 192.168.10.38 bertanya siapa pemilik perangkat dengan IP 192.168.10.1” bahwa kalimat tersebut memiliki unsur subjek pada bagian “IP 192.168.10.38”, predikat pada bagian “bertanya”, objek pada bagian “siapa pemilik perangkat” dan pelengkap pada bagian “dengan IP 192.168.10.1”. Apabila dibandingkan maka terlihat jelas bahwa hasil translasi lebih mudah dimengerti jika dibandingkan dengan hasil yang belum ditranslasi.

Tabel 1. Tabel Skala Penilaian

Skala	Bobot Penilaian
Sangat Mengerti	5
Mengerti	4
Netral	3
Tidak Mengerti	2
Sangat Tidak Mengerti	1

Penetapan skor penilaian pada kuisioner akan dilakukan

```
Waktu: 11/26/2019, 02:16:05 IP 192.168.10.38 bertanya siapa pemilik perangkat dengan IP 192.168.10.1
Waktu: 11/26/2019, 02:16:05 IP 192.168.10.1 merupakan pemilik dari perangkat 6c:3b:6b:e3:8a:23
Waktu: 11/26/2019, 02:16:06 IP 192.168.10.1 bertanya siapa pemilik perangkat dengan IP 192.168.10.39
Waktu: 11/26/2019, 02:16:35 IP 192.168.10.1 bertanya siapa pemilik perangkat dengan IP 192.168.10.39
```

Gambar 3. Hasil Translasi Protokol ARP

#### Hasil *Sniffing* Belum Ditranslasi

```
<Ether dst=ff:ff:ff:ff:ff:ff
src=c0:87:eb:40:f2:2f type=ARP | <ARP
hwtype=0x1 ptype=IPv4 hwlen=6 plen=4 op=who-
has hwsrc=c0:87:eb:40:f2:2f psrc=192.168.1.4
hwdst=00:00:00:00:00:00 pdst=192.168.1.1 |>>
```

Hasil dari kalimat translasi tersebut kemudian disimpan didalam sebuah basis data dengan nama induk yang sudah dibuat sebelumnya untuk memudahkan dalam melihat kembali hasil *sniffing* yang sudah dilakukan sebelumnya. Basis data ini kemudian akan ditampilkan menggunakan sebuah tampilan GUI dari PyQT5 untuk dapat dilihat oleh pengguna untuk dipelajari lebih lanjut mengenai peran dari setiap protokol yang ada.

Untuk mengukur efektivitas dari penggunaan *natural language* dalam aplikasi untuk memahami protokol jaringan komputer maka dilakukan metode kuisioner kepada murid SMK Negeri 1 Denpasar. Metode ini dilakukan dengan cara mengumpulkan data berupa nilai pemahaman responden terhadap hasil dari sebelum dilakukan translasi dan sesudah dilakukan translasi pada sebuah paket dengan protokol tertentu. Kuisioner dibuat dengan menampilkan hasil dari proses *sniffing* asli dan yang sudah dilakukan translasi dari aplikasi dengan menggunakan protokol ARP, ICMP, DNS, TCP, UDP, DHCP, dan HTTP. Penulisan hasil pada kuisioner kemudian dikelompokkan menjadi dua kelompok yaitu hasil *sniffing* asli dan hasil *sniffing* translasi. Responden kemudian diminta untuk memberikan nilai pemahaman terhadap hasil yang diberikan. Skala penilaian yang diberikan pada kuisioner dapat dilihat pada Tabel 1.

dengan menggunakan skala likert. Skala Likert adalah sebuah skala yang umum digunakan dalam mengetahui bagaimana pemahaman seseorang terhadap sesuatu hal yang diujikan [13]. Skala Likert memiliki sebuah persamaan yang digunakan untuk menghitung hasil kuisioner yang didapatkan untuk menyimpulkan pendapat responden secara umum. Persamaan Skala Likert dapat dilihat pada rumus 1.

$$\text{Skor Penilaian (Hasil Asli / Hasil Translasi)} = \frac{\text{Total Bobot}}{Y \times 100} \quad (1)$$

Total bobot diperoleh dari bobot nilai pada setiap skala nilai pemahaman yang kemudian dikalikan dengan jumlah responden. Nilai tersebut lalu dijumlahkan dengan nilai dari setiap skala pada hasil yang masih sama. Sebagai contoh apabila sebanyak 8 orang memilih pemahaman “sangat tidak mengerti” pada hasil asli protokol ARP maka jumlah 8 tersebut dikalikan dengan bobot nilai dari skala pemahaman “Sangat Tidak Mengerti”. Hal tersebut dilakukan pula pada hasil asli dari protokol lain dengan skala yang sama. Semua hasil tersebut kemudian dijumlahkan untuk mendapatkan total bobot.

Hasil persamaan ini nantinya akan memberikan persentase yang menggambarkan pemahaman responden terhadap hasil yang diberikan pada kuisioner. *Interval* skor penilaian yang dihasilkan dapat dilihat pada Tabel 2.

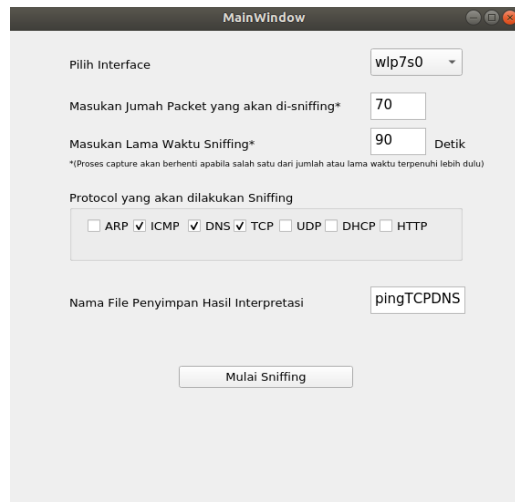
Tabel 2. Tabel Interval Skor Penilaian

Nilai	Keterangan
0% - 19,99%	Sangat Tidak Mengerti
20% - 39,99%	Tidak Mengerti
40% - 59,99%	Netral
60% - 79,99%	Mengerti
80% - 100%	Sangat Mengerti



### 3. Hasil dan Pembahasan

Hasil pengembangan dari sistem aplikasi yang sudah dibuat sebelumnya adalah berupa sebuah aplikasi dengan menggunakan pola kerja seperti pada Gambar 2. Aplikasi dijalankan dan dikembangkan di dalam sistem operasi Ubuntu. Pengujian aplikasi dilakukan pada jaringan Wi-Fi Program Studi Teknologi Informasi Universitas Udayana untuk mengetahui apakah aplikasi sudah bekerja sebagaimana mestinya.



Gambar 4. Tampilan Input Aplikasi *Sniffing*

Aplikasi akan dimulai dengan sebuah tampilan (Gambar 4) yang akan menampilkan berbagai parameter yang akan diperlukan oleh aplikasi untuk melakukan kegiatan *sniffing*-nya. Proses kemudian akan berjalan apabila setiap parameter sudah terisi semuanya. Hasil dari proses *sniffing* yang dijalankan adalah sebuah kalimat yang berisi hasil translasi dari paket yang berhasil ditangkap.

Untuk menilai efektivitas dari penggunaan bahasa yang digunakan dalam melakukan proses translasi paket data, maka dilakukan penyebaran kuisioner dengan responden merupakan siswa yang sedang mendalami bidang teknologi informasi dan turunannya. Penyebaran dilakukan terhadap siswa Teknik Komputer dan Jaringan yang sedang menempuh Pendidikan di SMK Negeri 1 Denpasar. Jumlah responden yang ada adalah sebanyak 29 orang.

Responden diberikan perbandingan hasil *sniffing* yang dilakukan terhadap sebuah protokol. Hasil pertama merupakan hasil yang didapatkan apabila menggunakan aplikasi *sniffing* pada umumnya yang disimpan ke dalam sebuah file *pcap* seperti yang terlihat pada kode program diatas. Hasil kedua merupakan hasil translasi seperti pada Gambar 4 yang didapatkan apabila menggunakan aplikasi pembelajaran yang sudah dibuat. Dua hasil tersebut kemudian dibandingkan dan diberikan nilai seberapa

paham responden terhadap masing-masing hasil yang diberikan.

Indikator efektivitas dari penggunaan *natural language* dalam hasil proses *sniffing* untuk meningkatkan pemahaman pengguna terhadap suatu paket protokol jaringan dibagi menjadi dua. Pertama skor penilaian hasil translasi *sniffing* lebih besar jika dibandingkan dengan skor penilaian hasil asli *sniffing*. Kedua responden lebih banyak memilih skala penilaian “Mengerti” dan “Sangat Mengerti” pada hasil translasi setiap protokol di kuisioner yang diberikan.

Untuk menghitung skor penilaian terhadap hasil *sniffing* asli dan translasi maka perlu untuk dilakukan pemetaan terhadap jawaban responden pada kuisioner yang diberikan. Pemetaan dilakukan dengan membuat sebuah tabel dengan protokol, skala pemahaman dan jumlah responden yang memilih skala pemahaman pada responden tersebut.

Tabel 3. Pemahaman Responden Terhadap Hasil Asli *Sniffing*

	STM	TM	Netral	M	SM	Total Responden
ARP	8	16	4	1	0	29
ICMP	9	14	6	0	0	29
DNS	8	19	1	1	0	29
TCP	9	18	2	0	0	29
UDP	7	15	4	3	0	29
DHCP	13	13	2	1	0	29
HTTP	7	16	2	4	0	29

Tabel 3 merupakan sebuah tabel yang menampilkan jumlah responden yang memilih pemahaman tertentu terhadap hasil asli *sniffing* protokol jaringan tertentu pada skala pemahaman yang sudah ditentukan dalam bentuk STM (Sangat Tidak Mengerti), TM (Tidak Mengerti), Netral, M (Mengerti), dan SM (Sangat Mengerti). Tersedia juga tabel yang menampilkan pemahaman responden terhadap hasil translasi *sniffing* yang dapat dilihat pada tabel 4. Untuk mengukur efektivitas dari penggunaan *natural language* terhadap hasil *sniffing* protokol jaringan untuk meningkatkan pemahaman pengguna terhadap hasil *sniffing* tersebut maka digunakan persamaan dari skala likert yang dapat dilihat pada metode penelitian.

Tabel 4. Pemahaman Responden Terhadap Hasil Translasi *Sniffing*

	STM	TM	Netral	M	SM	Total Responden
ARP	1	3	8	11	6	29
ICMP	0	3	5	12	9	29
DNS	0	2	3	18	6	29
TCP	1	6	12	6	4	29
UDP	1	3	7	10	8	29
DHCP	1	3	8	10	7	29
HTTP	1	3	9	11	5	29

Setiap skala pemahaman pada protokol tertentu memiliki jumlah responden yang bervariasi. Jumlah orang yang memilih pada setiap skala nilai pemahaman ini kemudian dikalikan dengan bobot penilaian yang sudah ditentukan pada setiap skala.

$$\text{Total bobot (Sangat Tidak Mengerti (ARP))} = JO \times BP \quad (2)$$

Sebagai contoh dapat dilihat pada persamaan 2 untuk melakukan perhitungan terhadap total bobot dari skala “Sangat Tidak Mengerti” pada protokol ARP. Jumlah orang (JO) yang memilih “Sangat Tidak Mengerti” adalah sebanyak 8 orang. Sedangkan bobot penilaian (BP) dari skala “Sangat Tidak Mengerti” adalah 1. Maka akan didapatkan total bobot dari skala “Sangat Tidak Mengerti” pada protokol ARP adalah sebesar 8. Perhitungan seperti persamaan diatas kemudian dilakukan juga pada skala dan protokol yang berbeda. Nantinya perhitungan tersebut akan menghasilkan tabel 5.

Tabel 5. Perhitungan Total Bobot Hasil Asli Sniffing

	STM	TM	Netral	M	SM	Total Responden
ARP	8	32	12	4	0	29
ICMP	9	28	18	0	0	29
DNS	8	32	3	4	0	29
TCP	9	32	6	0	0	29
UDP	7	30	12	12	0	29
DHCP	13	26	6	4	0	29
HTTP	7	32	6	16	0	29

Hasil perhitungan total bobot dari setiap protokol seperti yang terlihat pada tabel 5 kemudian dijumlahkan untuk mendapatkan total bobot secara keseluruhan.

$$T_{\text{keseluruhan}} = \sum_{i=1}^{35} T_i = T_1 + T_2 + T_3 + \dots + T_{35} \quad (3)$$

Persamaan diatas digunakan untuk menghitung Total bobot keseluruhan dari kuisioner hasil asli sniffing. Hasil yang didapatkan adalah sebesar 376. Nilai total bobot (T) keseluruhan dari skala pemahaman terhadap seluruh protokol jaringan yang ada ini nantinya akan digunakan sebagai salah satu variabel dalam persamaan untuk mencari skor indeks penilaian yang dipaparkan pada metode penelitian.

$$Y = TR \times JP \times \text{Nilai Max BO} \quad (4)$$

Persamaan diatas digunakan untuk mencari nilai dari variabel Y. Jumlah total responden (TR) yang memberikan penilaian pada setiap protokol adalah 29. Jumlah protokol (JP) yang ada adalah sebesar 7. Nilai 7 diperlukan untuk dikalikan dengan nilai 29 untuk dapat mengetahui total protokol yang diberikan nilai. Nilai maksimum dari bobot skala penilaian (BO) yang diberikan adalah sebesar 5. Skala penilaian dengan bobot nilai 5 adalah skala penilaian “Sangat Mengerti”. Setelah semua variabel sudah dimiliki maka dilanjutkan dengan mencari

nilai dari skor penilaian hasil asli sniffing. Persamaan untuk mencari skor penilaian dapat dilihat pada persamaan 1 dalam metode penelitian.

$$\text{Skor Penilaian (Hasil Asli Sniffing)} = \frac{T_{\text{keseluruhan}}}{Y} \times 100 \quad (5)$$

Setelah semua variabel dimasukan kedalam persamaan 5 diatas bahwa skor penilaian dari hasil asli sniffing adalah sebesar 37%. Apabila dilihat kembali dengan indeks penilaian pada Tabel 2 maka didapatkan bahwa responden tidak mengerti dengan hasil asli sniffing yang dikeluarkan aplikasi.

Tabel 6. Perhitungan Total Bobot Hasil Translasi Sniffing

	STM	TM	Netral	M	SM	Total Responden
ARP	1	6	24	44	30	29
ICMP	0	6	15	48	45	29
DNS	0	6	9	72	30	29
TCP	1	12	36	24	20	29
UDP	1	6	21	40	40	29
DHCP	1	6	24	40	35	29
HTTP	1	6	27	44	25	29

Untuk mendapatkan skor penilaian dengan hasil translasi cukup dengan melakukan apa yang sudah dilakukan sebelumnya untuk mencari skor penilaian hasil asli sniffing diantaranya adalah menghitung total bobot yang sudah tertera pada tabel 6, menghitung Tkeseluruhan, mencari nilai y dan memasukan semuanya ke dalam persamaan 6. Hanya saja perlu disesuaikan pada saat menghitung total bobot dengan jawaban responden terhadap hasil translasi sniffing yang berada di kuisioner.

$$\text{Skor Penilaian (Hasil Translasi Sniffing)} = \frac{T_{\text{keseluruhan}}}{Y} \times 100 \quad (6)$$

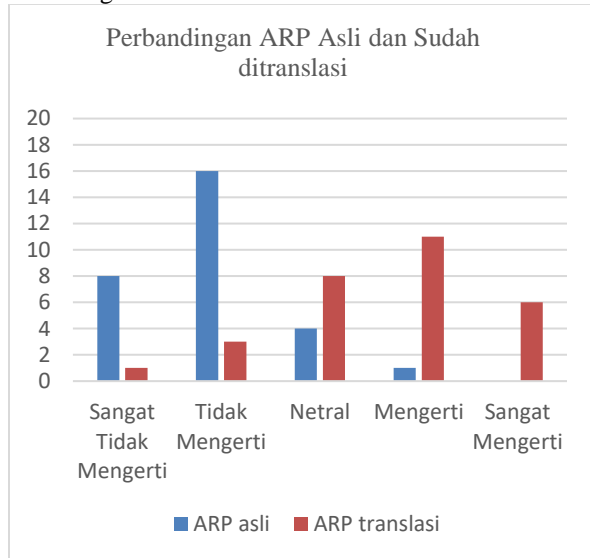
Setelah semua variabel dimasukan kedalam persamaan 6 maka akan didapatkan skor penilaian hasil translasi sniffing sebesar 73%. Apabila diperhatikan dengan indeks penilaian dari Tabel 2 maka didapatkan kesimpulan bahwa responden mengerti dengan hasil translasi yang dikeluarkan aplikasi. Hasil translasi ini menggunakan konsep *natural language* sehingga dapat dikatakan bahwa penyampaian cara kerja protokol jaringan kepada pengguna lebih mudah dipahami dengan menggunakan konsep *natural language* dibandingkan dengan yang tidak menggunakan.

Untuk melihat perbandingan pemahaman setiap protokol dengan lebih mendetail maka akan diberikan grafik pada setiap protokol yang menampilkan antara jumlah responden yang memahami antara hasil asli dan translasi.

Apabila diperhatikan pada grafik yang dihasilkan setelah melakukan survei terhadap sejumlah responden terkait tingkat perbandingan pemahaman responden dengan hasil sniffing asli ARP dengan hasil sniffing yang sudah



ditranslasikan menggunakan aplikasi ini maka terlihat pada Gambar 5 bahwa responden memiliki kecenderungan untuk tidak mengerti dengan apa yang dimaksud dari hasil *sniffing* protokol ARP asli dengan 83% responden memilih antara sangat tidak mengerti dan tidak mengerti. Sedangkan 17% sisanya memilih netral dan mengerti.



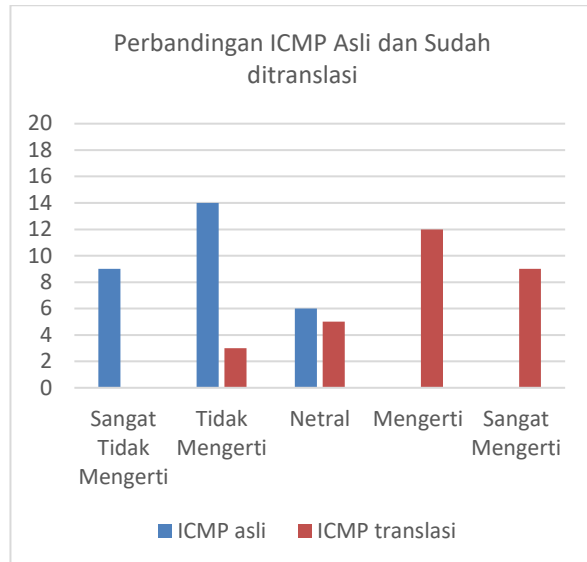
Gambar 5. Hasil Survei terhadap Protokol ARP Asli dan Translasi

Saat responden diberikan hasil *sniffing* yang sudah dilakukan translasi oleh aplikasi dapat diperhatikan bahwa pemahaman responden terhadap paket data sebuah protokol ARP memiliki kecenderungan lebih baik dengan paket data yang sudah ditranslasi apabila dibandingkan dengan paket data yang asli tanpa ada proses translasi dengan 59% responden memilih pilihan “Mengerti” dan “Sangat Mengerti”. Sedangkan hanya sebanyak 14% responden memilih “Sangat Tidak Mengerti” dan “Tidak Mengerti”.

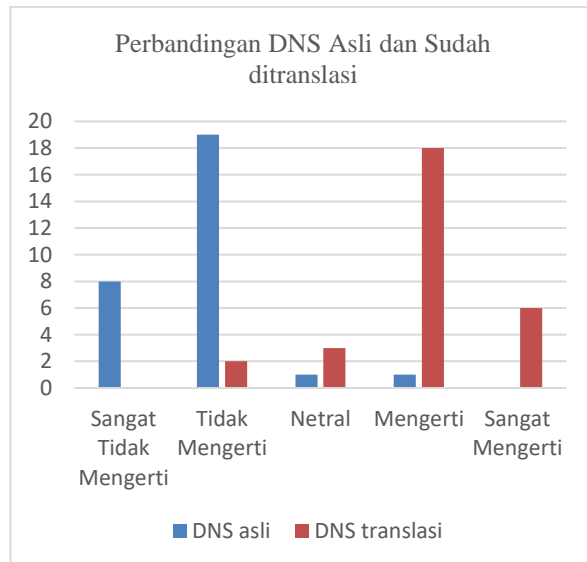
Apabila memperhatikan hasil survei terhadap hasil *sniffing* asli dari protokol ICMP maka dapat dilihat bahwa sebanyak sebanyak 23 orang atau 79% dari keseluruhan responden menyatakan sangat tidak mengerti serta sebanyak 6 orang atau 21% dari keseluruhan responden menyatakan netral. Hal ini memberikan gambaran bahwa Masih banyak responden yang belum terlalu memahami mengenai hasil asli dari proses *sniffing* terhadap protokol ICMP.

Translasi yang dilakukan oleh aplikasi ini terhadap protokol ICMP juga memberikan respon yang baik terhadap para responden. Nilai pemahaman “Sangat Mengerti” dan “Mengerti” pada hasil *sniffing* protokol ICMP yang sudah di-translasi mengalami peningkatan jika dibandingkan dengan hasil *sniffing* protokol ICMP Asli yang mana sebanyak 21 orang atau 72% dari

keseluruhan responden. Sedangkan sebanyak 8 orang atau 28% dari keseluruhan responden memilih “Netral” dan “Sangat Tidak Mengerti.”



Gambar 6. Hasil Survei Terhadap Protokol ICMP Asli dan Translasi

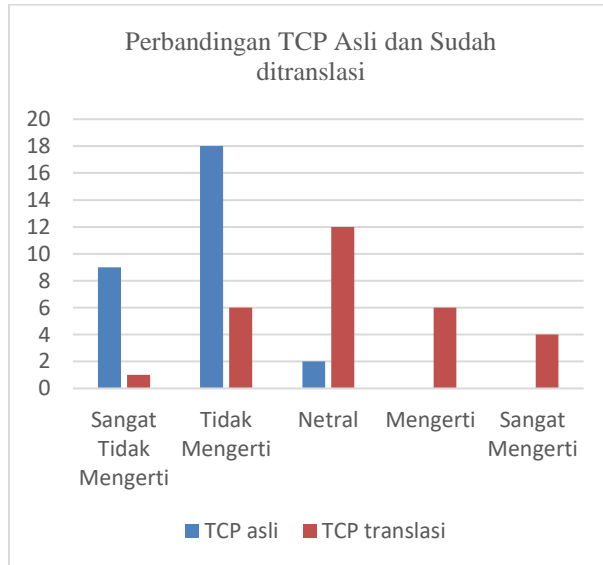


Gambar 7. Hasil Survei Terhadap Protokol DNS Asli dan Sudah ditranslasi

Protokol DNS atau yang biasa disebut sebagai *Domain Name Server* adalah sebuah protokol yang menyediakan kemampuan untuk mencari alamat dari suatu komputer atau *server* yang berada pada suatu jaringan. Hasil survei dari DNS Asli menunjukkan bahwa terdapat 27 orang atau 93% dari keseluruhan responden yang menyatakan “Sangat Tidak Mengerti” dan “Tidak Mengerti”, serta 2 orang atau 7% dari keseluruhan responden menyatakan “Netral” dan “Mengerti”. Hal ini menunjukkan bahwa

responden sebagian besar tidak mengerti dengan hasil asli dari proses *sniffing* protokol DNS.

Berbeda dengan versi aslinya, respon dari responden terhadap hasil *sniffing* yang sudah dilakukan translasi oleh aplikasi ini cenderung lebih memahami terhadap maksud dari paket data yang diberikan. Hal ini dapat dilihat dari jumlah responden yang memilih “Sangat Tidak Mengerti” kini tidak ada sedangkan jumlah orang yang memilih “Mengerti” dan “Sangat Mengerti” kini menjadi total sebanyak 24 orang atau 83% dari jumlah responden keseluruhan.

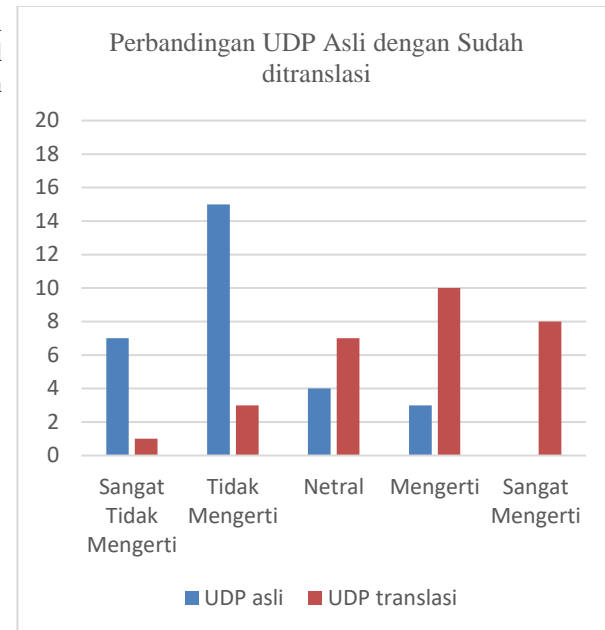


Gambar 8. Hasil Survei Terhadap Protokol TCP Asli dan Sudah ditranslasi

Protokol TCP adalah sebuah protokol yang sangat umum sekali digunakan dalam jaringan Internet saat ini. Protokol ini lebih mengutamakan untuk menjamin setiap data yang ingin dikirimkan sampai pada tujuan yang diinginkan dan bukan untuk mengutamakan kecepatan penyampaian data [14].. Apabila diperhatikan dari hasil survei TCP asli yang dilakukan maka dapat terlihat bahwa responden kesulitan dalam memahami kegiatan yang dilakukan oleh protokol TCP. Hal ini terlihat dari jumlah responden yang memilih “Sangat Tidak Mengerti” dan “Tidak Mengerti” adalah sejumlah 27 orang atau 93% dari keseluruhan responden.

Setelah aplikasi ini melakukan translasi terhadap paket data dengan protokol TCP maka kita bisa lihat perbedaannya dari pemahaman hasil TCP Asli dengan yang sudah ditranslasi adalah adanya pergeseran nilai pemahaman yang mana sebelumnya nilai “Mengerti” dan “Sangat Mengerti” adalah tidak ada seperti yang terlihat pada Gambar 8. Kini dengan dilakukan translasi nilai pemahaman responden apabila dilihat dari Gambar 8 adalah sebanyak 10 orang atau 34% dari jumlah

keseluruhan responden yang memilih nilai “Mengerti” dan “Sangat Mengerti” serta 12 orang memilih “Netral” atau 41% dari jumlah keseluruhan responden. Hal ini menunjukkan keberhasilan dari hasil translasi aplikasi ini yang mampu membuat pengguna menjadi lebih memahami mekanisme sebuah protokol bekerja pada jaringan.

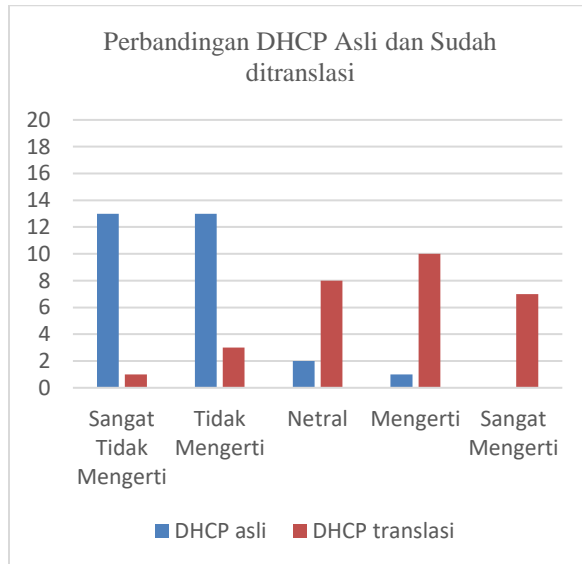


Gambar 9. Hasil Survei Terhadap Protokol UDP Asli dan Sudah ditranslasi

Protokol UDP adalah sebuah protokol yang mengirimkan datanya tanpa adanya jaminan terhadap integritas data tersebut dapat sampai ke tujuan atau tidak. Protokol ini lebih mengutamakan kecepatan dalam mengirimkan datanya. *Reliability* penyampaian sebuah data masih lebih baik dilakukan oleh protokol TCP [15]. Apabila diperhatikan dari hasil survei yang dilakukan terhadap hasil *sniffing* protokol UDP Asli seperti yang terlihat pada Gambar 9 maka dapat disimpulkan bahwa sebagian besar responden masih belum terlalu memahami apa yang dimaksud dari hasil *sniffing* protokol UDP Asli. Hal ini terlihat dari jumlah responden yang memilih “Sangat Tidak Mengerti” dan “Tidak Mengerti” adalah sebanyak total 22 orang atau 76% dari jumlah total responden.

Setelah dilakukan translasi terhadap hasil *sniffing* UDP asli menggunakan aplikasi ini maka dapat terlihat pada Gambar 9 bahwa tingkat pemahaman responden terhadap hasil *sniffing* protokol UDP yang sudah di-translasi menunjukkan adanya peningkatan pemahaman responden terhadap paket protokol UDP yang diberikan. Terlihat bahwa nilai dari responden yang memilih “Sangat Tidak Mengerti” kini menjadi 1. Sebaliknya nilai responden yang memilih “Mengerti” dan “Sangat Mengerti” kini

meningkat menjadi total 18 orang atau 62% dari jumlah keseluruhan responden. Hal ini menunjukkan adanya peningkatan pemahaman setelah dilakukan translasi.



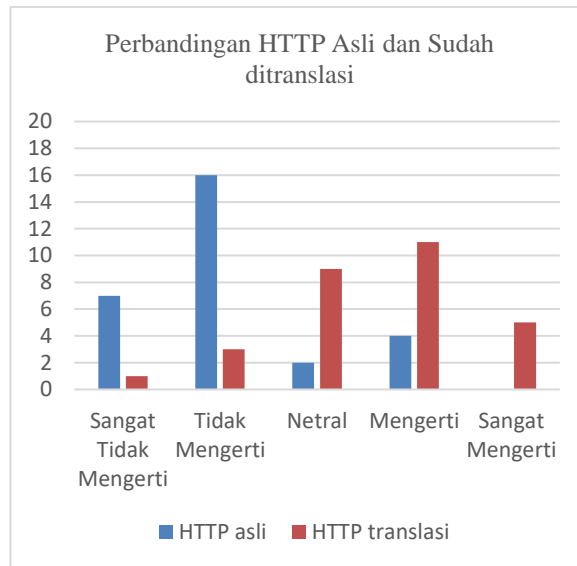
Gambar 10. Hasil Survei Terhadap Protokol DHCP Asli dan Sudah ditranslasi

Protokol DHCP adalah sebuah protokol yang umumnya digunakan pada saat sebuah perangkat baru saja terhubung ke dalam sebuah jaringan dan membutuhkan IP Address untuk dapat berkomunikasi [16]. DHCP bertugas untuk memberikan IP tersebut kepada perangkat yang baru saja terkoneksi ke dalamnya. Apabila diperhatikan pada Gambar 10, responden umumnya memilih nilai pemahaman “Sangat Tidak Mengerti” dan “Tidak Mengerti” pada saat diberikan hasil dari DHCP Asli. Jumlah total yang memilih dua nilai tersebut mencapai 24 orang atau 83% dari jumlah total responden. Hal ini menunjukkan bahwa secara garis besar responden tidak dapat memahami paket data DHCP yang dihasilkan secara langsung oleh proses *sniffing*.

Sedangkan saat responden memberikan penilaian terhadap hasil *sniffing* DHCP yang sudah di-translasi, nilai pemahaman yang diberikan oleh responden lebih cenderung mengarah pada nilai “Mengerti” dan “Sangat Mengerti” dengan jumlah sebanyak 17 orang atau 59% dari jumlah keseluruhan responden. Hal ini menunjukkan bahwa hasil translasi dari protokol DHCP lebih dimengerti dibandingkan hasil asli yang dihasilkan sebelumnya.

Protokol HTTP adalah sebuah protokol yang membantu *client* dan *web server* untuk dapat saling berkomunikasi satu sama lain. Komunikasi yang dilakukan biasanya dapat berupa pemberian asset website kepada *client* serta data dari *client* kepada *web server*. Apabila memperhatikan hasil survei yang dilakukan pada Gambar

11 maka dapat kita lihat bahwa secara mayoritas, responden masih memilih nilai “Sangat Tidak Mengerti” dan “Tidak Mengerti” pada hasil HTTP asli sebanyak 23 orang atau 79% dari keseluruhan responden.



Gambar 11. Hasil Survei Terhadap Protokol HTTP Asli dan Sudah ditranslasi

Begitu responden diberikan hasil *sniffing* paket dengan protokol HTTP yang sudah ditranslasi terlihat bahwa nilai pemahaman yang diberikan oleh responden meningkat. Terlihat pada Gambar 11 bahwa responden yang memilih nilai “Mengerti” dan “Sangat Mengerti” kini mengalami peningkatan dengan nilai total yang mencapai 16 orang atau 55% dari keseluruhan jumlah responden. Hal ini menunjukkan bahwa orang-orang yang sebelumnya memiliki pemahaman yang lebih rendah terhadap paket yang ada kini bisa lebih memahami paket yang ada dengan adanya translasi tersebut.

#### 4. Kesimpulan

Penggunaan Scapy dalam pembuatan aplikasi pembelajaran lalu lintas protokol ini sangat mengambil peran penting terutama sangat membantu mengenali setiap protokol yang ada. Implementasinya yang mudah diterapkan ke dalam Python membuat pembuatan aplikasi untuk kebutuhan pembelajaran protokol dapat lebih mudah dilakukan. Penerapan basis data untuk menyimpan setiap kalimat yang ada membuat penggunaan *natural language* dalam hasil proses *sniffing* dapat dilakukan dengan baik. Pengujian yang dilakukan dengan menyebarkan kuisioner kepada berbagai kalangan pelajar yang sedang mendalami ilmu jaringan komputer juga menunjukkan efektivitas aplikasi ini dalam membantu mereka dalam memahami sebuah paket data dengan protokol tertentu pada sebuah jaringan dengan

menggunakan hasil yang sudah ditranslasi dengan *natural language*. Skor penilaian hasil *sniffing* translasi yang menunjukkan hasil 73% dapat dijadikan sebagai indikator efektivitas penggunaan *natural language* dalam aplikasi pembelajaran lalu lintas protokol jaringan.

### Daftar Rujukan

- [1] A. Pane and M. Darwis Dasopang, "Belajar Dan Pembelajaran," *FITRAH Jurnal Kaji. Ilmu-ilmu Keislam.*, vol. 3, no. 2, p. 333, 2017.
- [2] Yuliana, *Bab 3 : Pengolahan Bahasa Alami*. [Online] Tersedia di : <http://yuliana.lecturer.pens.ac.id/Kecerdasan%20Buatan/Buku/Bab%205%20Natural%20Language%20Processing.pdf> [Accessed 13 April 2020].
- [3] Y. Icha, "Pembentuk Serta Pola Kalimat : Pengertian, Unsur Dan Contohnya Lengkap," *Ruangguru.Co.Id*, 2019. [Online]. Available: <https://www.ruangguru.co.id/pengertian-dan-contoh-unsur-pembentuk-serta-pola-kalimat-lengkap/>. [Accessed: 05-May-2020].
- [4] M. J. N. Yudianto, "Jaringan Komputer dan Pengertiannya," *Ilmukomputer.Com*, pp. 1–10, 2014.
- [5] P. Saxena, "Analysis of Network Traffic by using Packet Sniffing Tool : Wireshark," *Int. J. Adv. Res. Ideas Innov. Technol.*, vol. 3, no. 6, pp. 804–808, 2017.
- [6] C. Gandhi, G. Suri, R. P. Golyan, P. Saxena, and B. K. Saxena, "Packet Sniffer – A Comparative Study," *Int. J. Comput. Networks Commun. Secur.*, vol. 2, no. 5, pp. 179–187, 2014.
- [7] A. Siswanto, A. Syukur, E. A. Kadir, and Suratin, "Network traffic monitoring and analysis using packet sniffer," *Proc. - 2019 Int. Conf. Adv. Commun. Technol. Networking, CommNet 2019*, pp. 1–4, 2019.
- [8] V. Elamaram *et al.*, "Exploring DNS, HTTP, and ICMP Response Time Computations on Brain Signal/Image Databases using a Packet Sniffer Tool," *IEEE Access*, vol. 6, pp. 59672–59678, 2018.
- [9] P. Goyal and A. Goyal, "Comparative study of two most popular packet sniffing tools-Tcpdump and Wireshark," in *Proceedings - 9th International Conference on Computational Intelligence and Communication Networks, CICN 2017*, 2018, vol. 2018-Janua, pp. 77–81.
- [10] M. Khan, S. Alshomrani, and S. Qamar, "Investigation of DHCP Packets using Wireshark," *Int. J. Comput. Appl.*, vol. 63, no. 4, pp. 1–9, 2013.
- [11] M. Kumar and M. T. Scholar, "Tcp & Udp Packets Analysis Using Wireshark," *Int. J. Sci. Eng. Technol. Res.*, vol. 4, no. 7, pp. 2470–2474, 2015.
- [12] P. Biondi, "Scapy Documentation," vol. 469, no. 4, pp. 155–203, 2017 [Online] Tersedia di : [https://scapy.readthedocs.io/\\_/downloads/en/latest/pdf/](https://scapy.readthedocs.io/_/downloads/en/latest/pdf/) [Accessed 2 Februari 2020].
- [13] Editor, "Pengertian Skala Likert dan Contoh Cara Hitung Kuesionernya," *diedit.com*, 2020. [Online]. Available: <https://www.diedit.com/skala-likert/> [Accessed 2 Februari 2020].
- [14] S. Jonnalagadda, "Introduction to TCP / IP Protocol Suite Srinivas Jonnalagadda , Ph . D .," no. April, 2019.
- [15] F. T. Al-Dhief *et al.*, "Performance Comparison between TCP and UDP Protocols in Different Simulation Scenarios," *Int. J. Eng. Technol.*, vol. 7, no. 4.36, pp. 172–176, 2018.
- [16] W. Goralski, "Dynamic Host Configuration Protocol," in *The Illustrated Network*, Elsevier, 2017, pp. 563–586.