WORKSHOP PEMROGRAMAN JARINGAN
MODUL 9
(NETWORK INTERFACES)

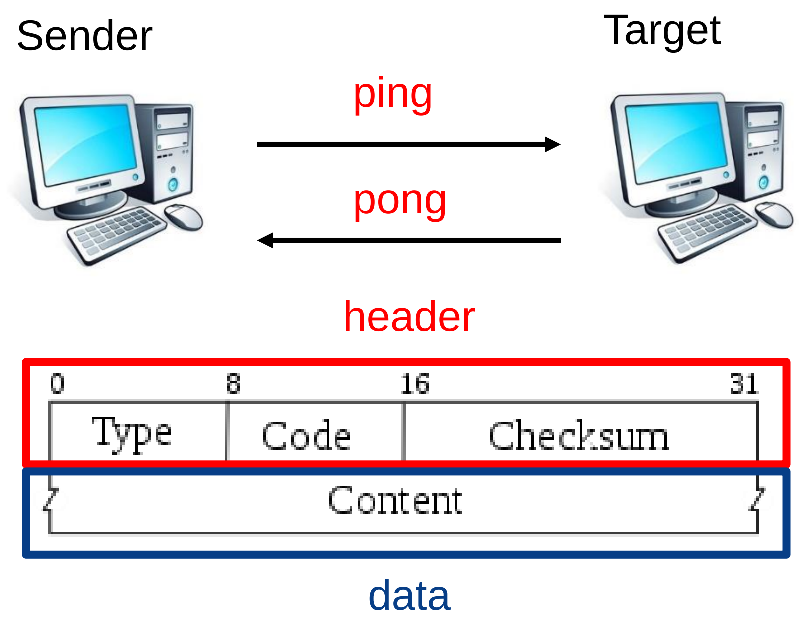
Mochammad Zen Samsono Hadi, ST. MSc. Ph.D

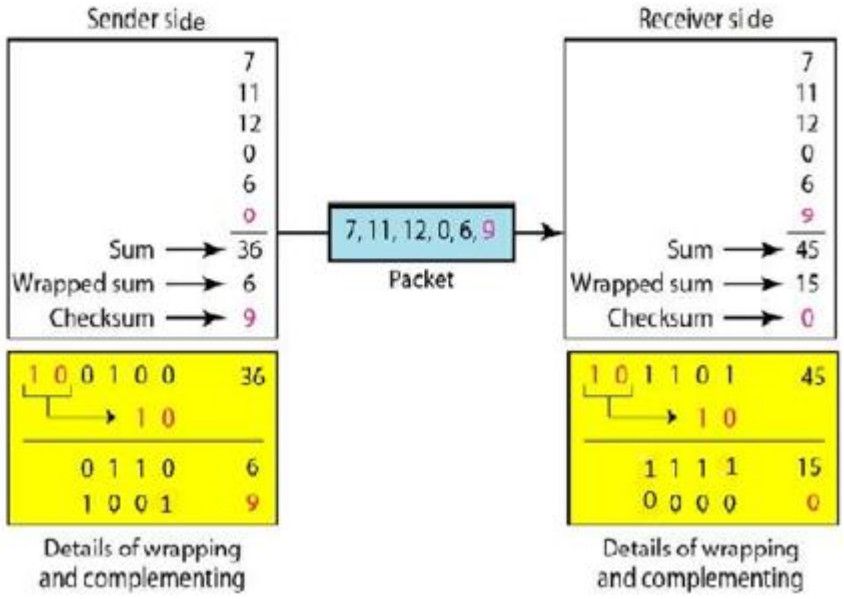
Teknik Telekomunikasi PENS

TOPIK PEMBAHASAN

- Aplikasi ICMP (ping)
- List Interface
- Menampilkan alamat IP

Format Pesan ICMP





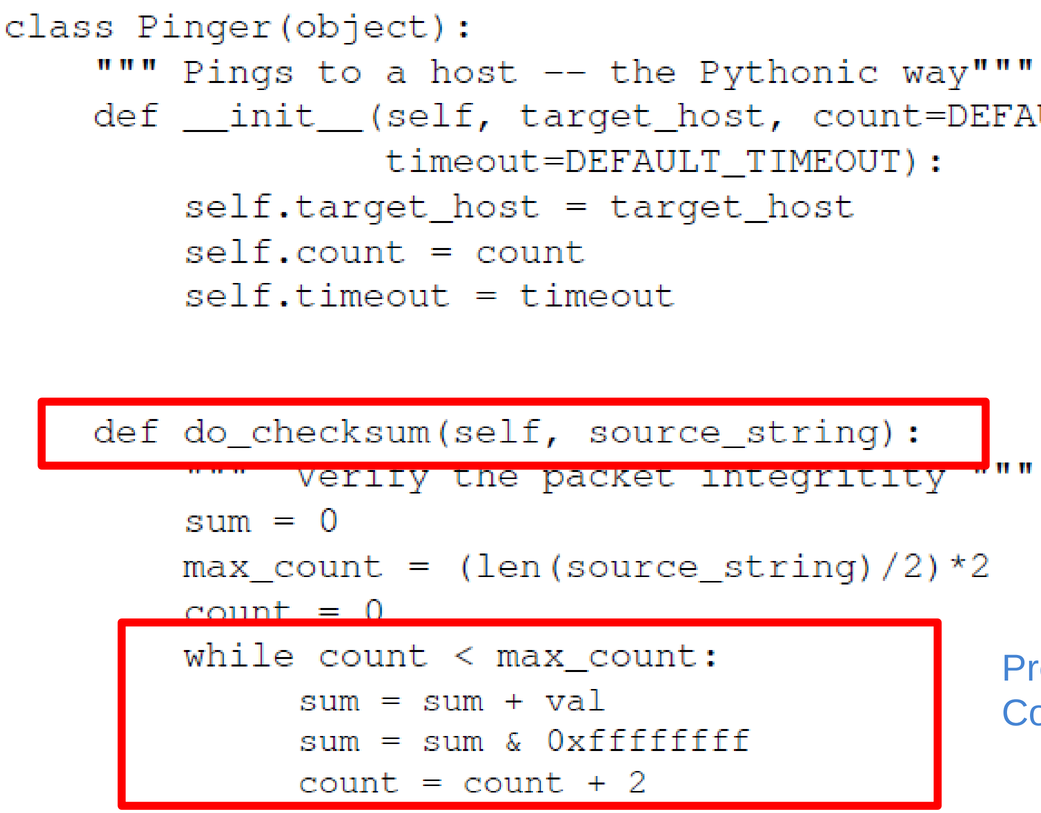
- Type:

  a. Echo reply dan echo request (type 0 dan 8)

  b. Destination unreachable (type 3)

  c. Time exceeded, dll
- Code: dest. Net (0), desired host (1), targeted port (3)
- Checksum: perhitungan pengecekan error
- Content: ID, sequence number, operational data

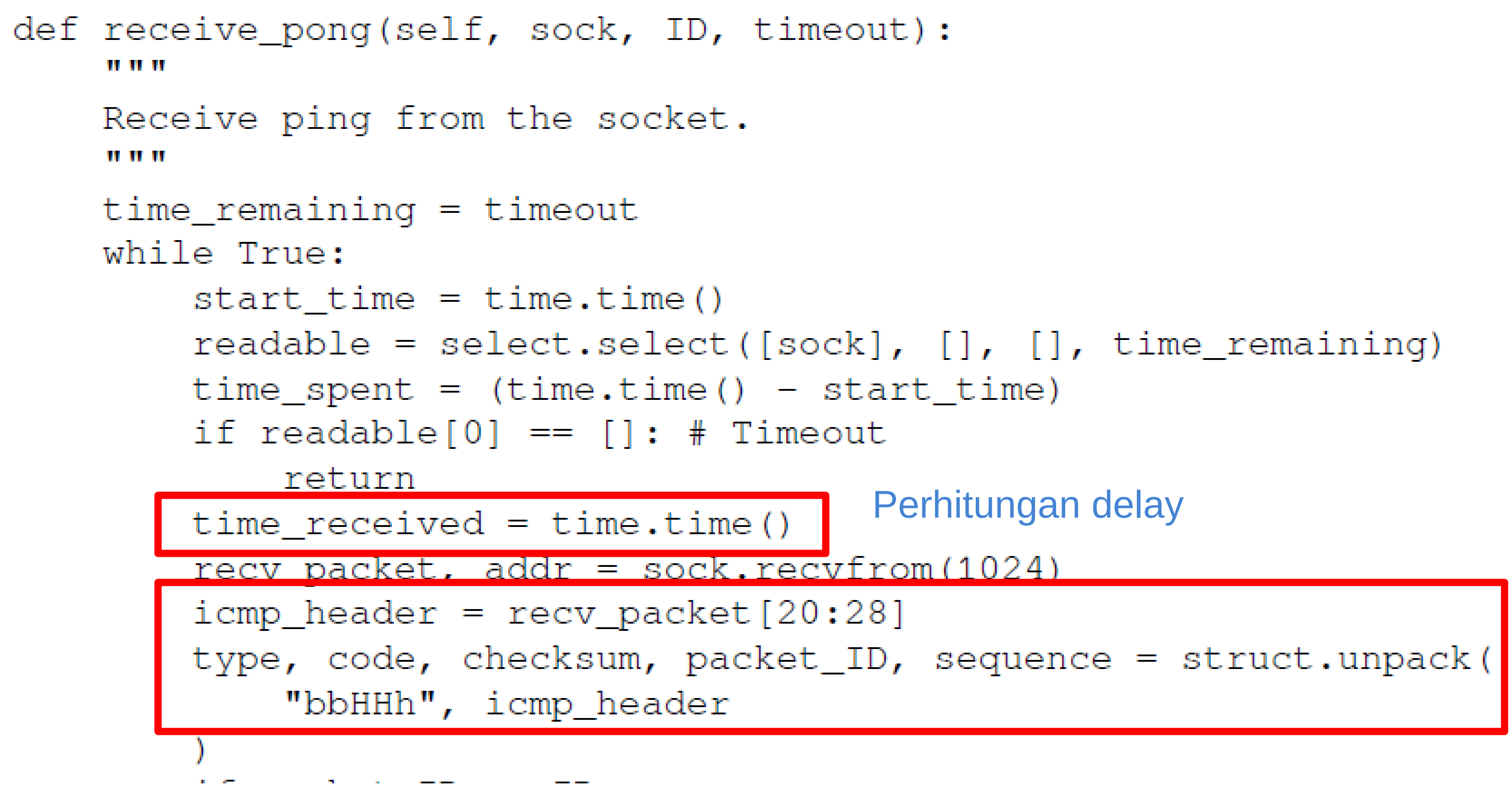
Pengecekan Checksum



```
class Pinger(object):
    """ Pings to a host -- the Pythonic way"""
    def __init__(self, target_host, count=DEFAULT_COUNT,
                 timeout=DEFAULT_TIMEOUT):
        self.target_host = target_host
        self.count = count
        self.timeout = timeout


    def do_checksum(self, source_string):
        """  verify the packet integritity """
        sum = 0
        max_count = (len(source_string)/2)*2
        count = 0
        while count < max_count:
             sum = sum + val
             sum = sum & 0xffffffff
             count = count + 2
```

Proses perhitungan checksum
Coba ganti: 0xffffffff dengan 0xfffffff3

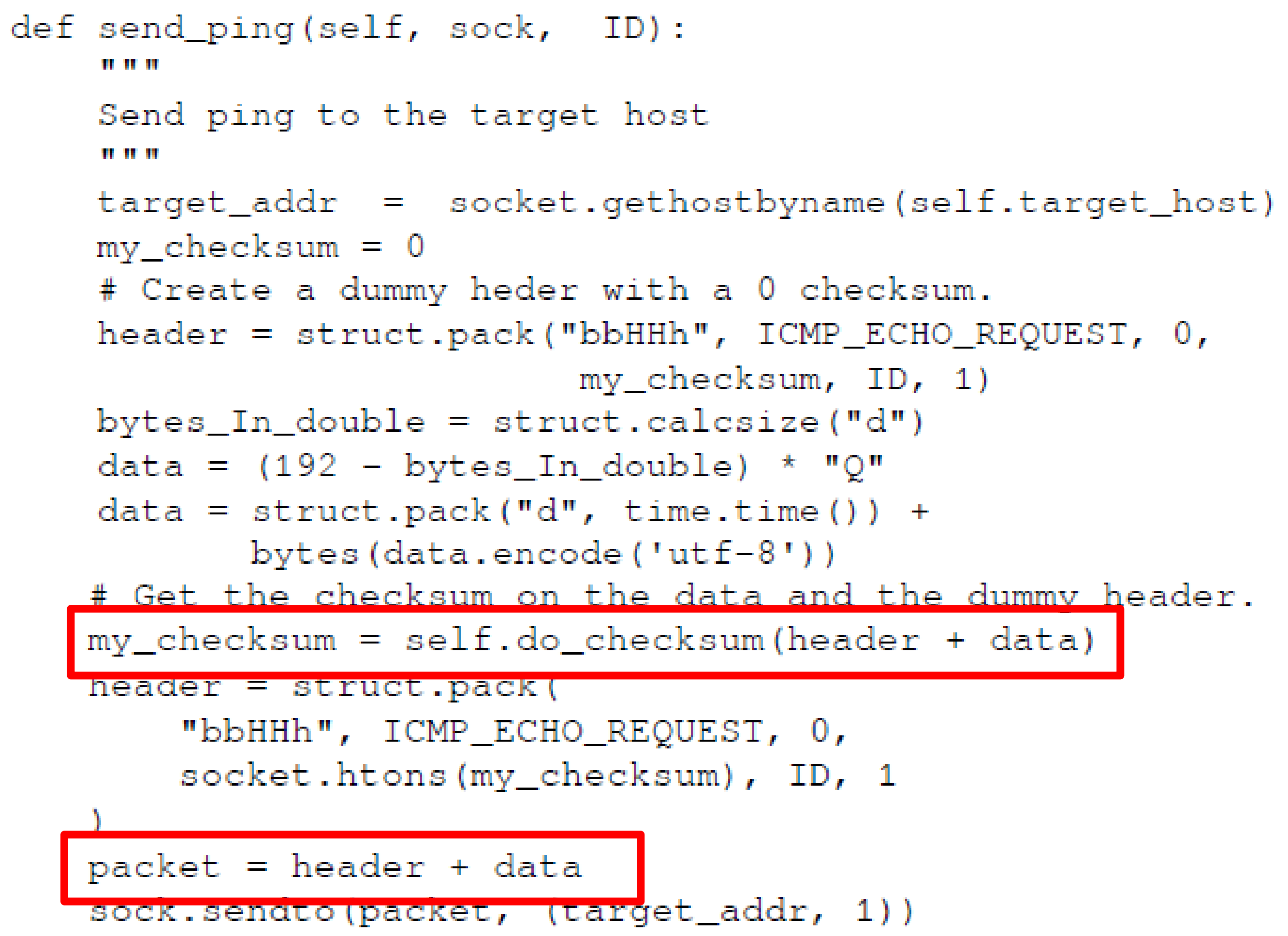
Menerima paket: pong



```
def receive_pong(self, sock, ID, timeout):
    """
    Receive ping from the socket.
    """
    time_remaining = timeout
    while True:
        start_time = time.time()
        readable = select.select([sock], [], [], time_remaining)
        time_spent = (time.time() - start_time)
        if readable[0] == []: # Timeout
            return
        time_received = time.time()
        recv_packet, addr = sock.recvfrom(1024)
        icmp_header = recv_packet[20:28]
        type, code, checksum, packet_ID, sequence = struct.unpack(
            "bbHHh", icmp_header
        )
```

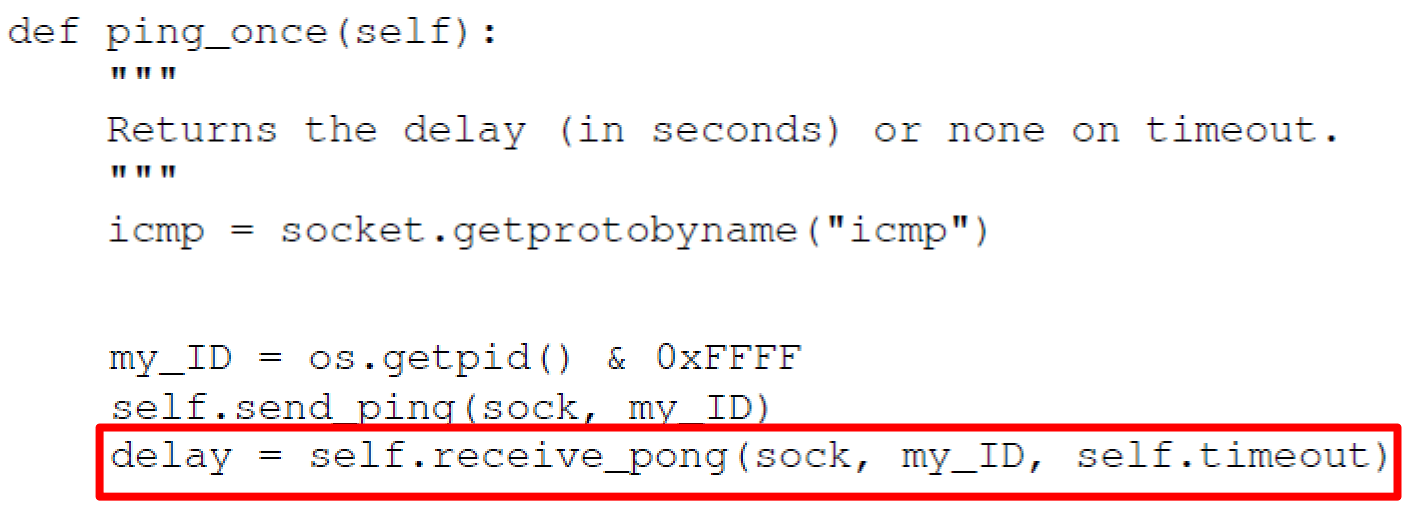
Perhitungan delay

Mengirim paket: ping

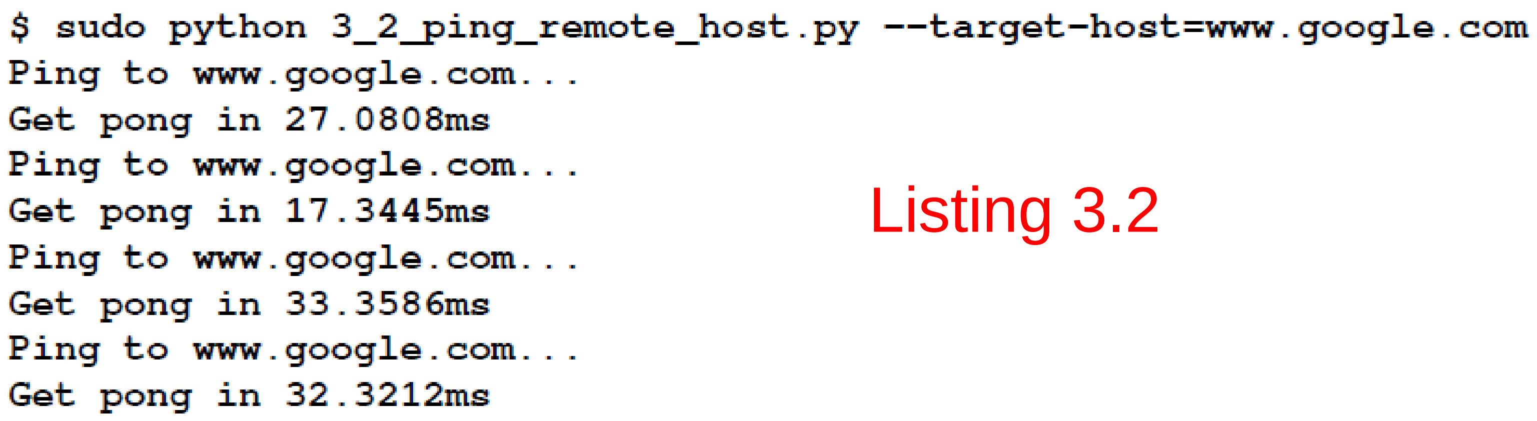


```
def send_ping(self, sock,  ID):
    """
    Send ping to the target host
    """
    target_addr  =  socket.gethostbyname(self.target_host)
    my_checksum = 0
    # Create a dummy heder with a 0 checksum.
    header = struct.pack("bbHHh", ICMP_ECHO_REQUEST, 0,
                          my_checksum, ID, 1)
    bytes_In_double = struct.calcsize("d")
    data = (192 - bytes_In_double) * "Q"
    data = struct.pack("d", time.time()) +
           bytes(data.encode('utf-8'))
    # Get the checksum on the data and the dummy header.
    my_checksum = self.do_checksum(header + data)
    header = struct.pack(
        "bbHHh", ICMP_ECHO_REQUEST, 0,
        socket.htons(my_checksum), ID, 1
    )
    packet  =  header + data
    sock.sendto(packet,  (target_addr, 1))
```

Perhitungan delay & Hasil



```
def ping_once(self):
    """
    Returns the delay (in seconds) or none on timeout.
    """
    icmp = socket.getprotobyname("icmp")

    my_ID = os.getpid() & 0xFFFF
    self.send_ping(sock, my_ID)
    delay = self.receive_pong(sock, my_ID, self.timeout)
```

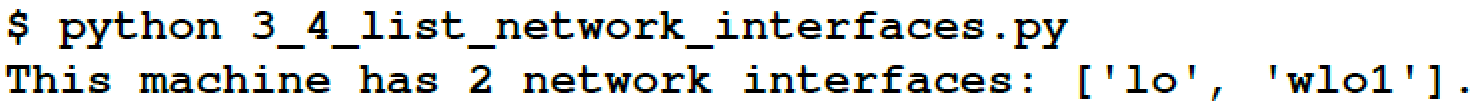


```
$ sudo python 3_2_ping_remote_host.py --target-host=www.google.com
Ping to www.google.com...
Get pong in 27.0808ms
Ping to www.google.com...
Get pong in 17.3445ms
Ping to www.google.com...
Get pong in 33.3586ms
Ping to www.google.com...
Get pong in 32.3212ms
```

Listing 3.2

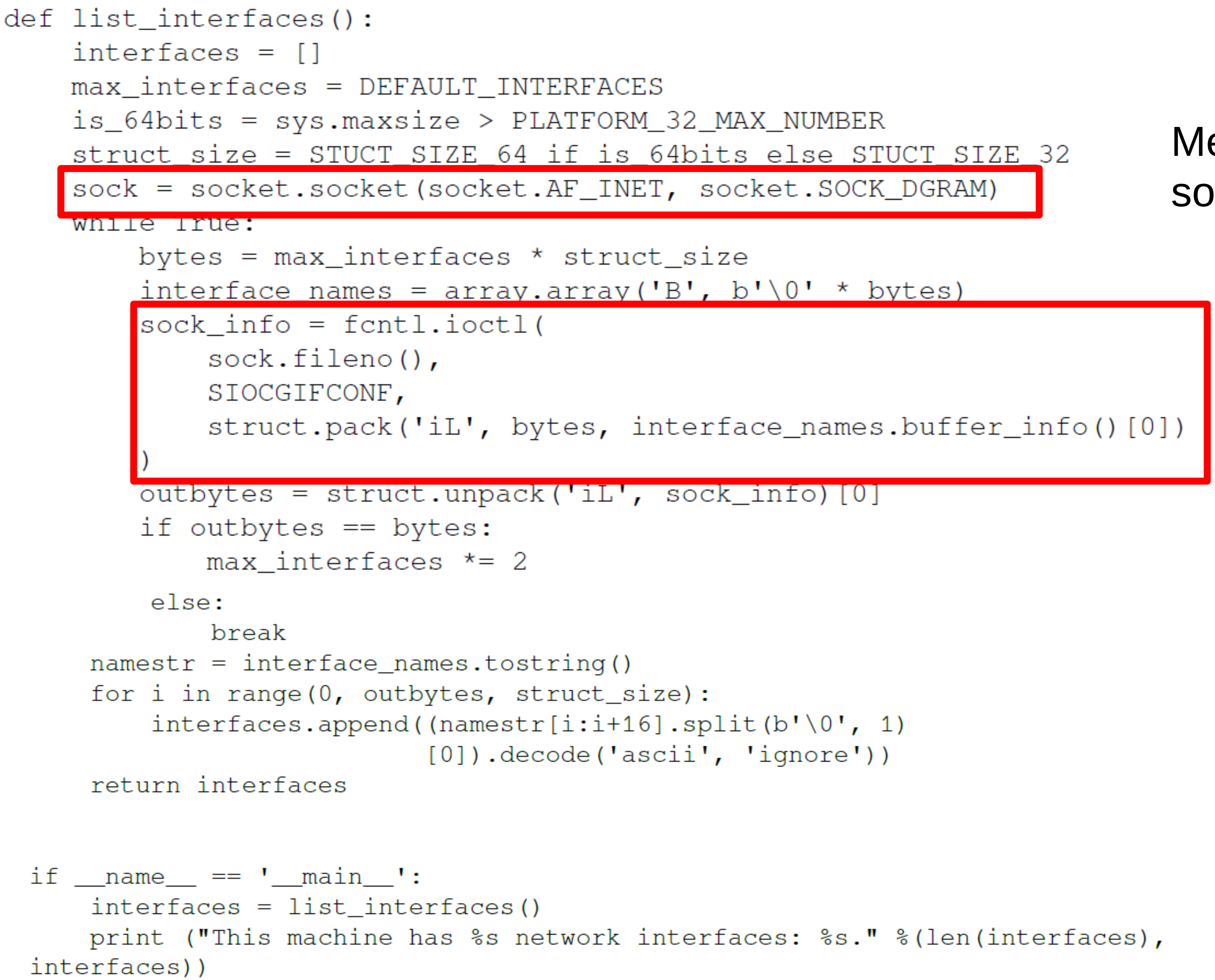
List interface di komputer

- Pengganti perintah:
  - # ifconfig		di linux versi lama
  - # ip address		di linux versi baru
- Menggunakan fcntl module untuk menangani I/O di linux
- Hasil listing 3.4



```
$ python 3_4_list_network_interfaces.py
This machine has 2 network interfaces: ['lo', 'wlo1'].
```

Program



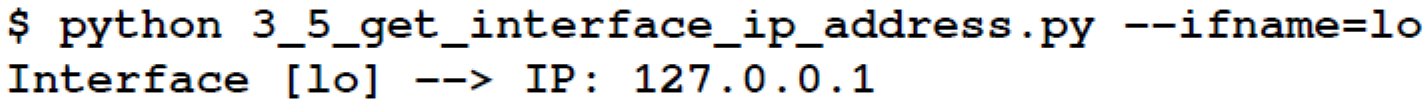
```
def list_interfaces():
    interfaces = []
    max_interfaces = DEFAULT_INTERFACES
    is_64bits = sys.maxsize > PLATFORM_32_MAX_NUMBER
    struct_size = STUCT_SIZE_64 if is_64bits else STUCT_SIZE_32
    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    while True:
        bytes = max_interfaces * struct_size
        interface_names = array.array('B', b'\0' * bytes)
        sock_info = fcntl.ioctl(
            sock.fileno(),
            SIOCGIFCONF,
            struct.pack('iL', bytes, interface_names.buffer_info()[0])
        )
        outbytes = struct.unpack('iL', sock_info)[0]
        if outbytes == bytes:
            max_interfaces *= 2

        else:
            break
    namestr = interface_names.tostring()
    for i in range(0, outbytes, struct_size):
        interfaces.append((namestr[i:i+16].split(b'\0', 1)
                          [0]).decode('ascii', 'ignore'))
    return interfaces


if __name__ == '__main__':
    interfaces = list_interfaces()
    print ("This machine has %s network interfaces: %s." %(len(interfaces),
interfaces))
```

Memodifikasi socket UDP

Proses mengambil list interface

Menampilkan alamat IP

- Pengganti perintah:
  - # ifconfig lo   di linux versi lama
  - # ip address show lo   di linux versi baru
- Menggunakan object socket dan fcntl.ioctl() function untuk mendapatkan informasi IP
  - Fungsi socket.inet_ntoa() utk merubah format IP ke yg mudah dibaca
- Hasil



```
$ python 3_5_get_interface_ip_address.py --ifname=lo
Interface [lo] --> IP: 127.0.0.1
```

Program

```
def get_ip_address(ifname):
    s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    return socket.inet_ntoa(fcntl.ioctl(
        s.fileno(),
        0x8915,  # SIOCGIFADDR
        struct.pack(b'256s', bytes(ifname[:15], 'utf-8'))
    )[20:24])

if __name__ == '__main__':
  parser = argparse.ArgumentParser(description='Python
                                     networking utils')
  parser.add_argument('--ifname', action="store",
                       dest="ifname", required=True)
  given_args = parser.parse_args()
  ifname = given_args.ifname
  print ("Interface [%s] --> IP: %s" %(ifname, get_ip_address(ifname)))
```

Percobaan

- Cobalah listing 3.2 untuk aplikasi ICMP dan rubahlah nilai di perhitungan checksum, analisa hasilnya.
- Cobalah listing 3.4 untuk mengetahui list interface
- Cobalah listing 3.5 untuk mengetahui alamat IP
- Buatlah menu pilihan seperti berikut:

MENU PILIHAN

1. Mengetahui delay ke suatu alamat
2. Mengetahui list interface
3. Mengetahui alamat IP

Masukkan pilihan anda:

Jika di tekan 1:

Input:

Masukkan alamat web: www.pens.ac.id

Output:

ping to www.pens.ac.id
Get pong in 27ms
….

Percobaan

- Buatlah menu pilihan seperti berikut:

MENU PILIHAN
1. Mengetahui delay ke suatu alamat
2. Mengetahui list interface
3. Mengetahui alamat IP

Masukkan pilihan anda:

Jika di tekan 1:

Input:

Masukkan alamat web: www.pens.ac.id

Output:

ping to www.pens.ac.id
Get pong in 27ms
….

Jika ditekan 2:

List interface akan muncul

Jika ditekan 3:

Input:

Masukkan interface: lo

Ouput:

lo: 127.0.0.1