Tugas: Jelaskan apa yang dimaksud dengan socket pada python network programming ?

Tuliskan jawaban anda pada doc ms word kemudia upload pada assingment ini.

Nama : Nurul Amalina Setyorini
NIM : 202420005
Jurusan : Magister Teknik Informatika
Kelas : Regular B

Tugas: Jelaskan apa yang dimaksud dengan socket pada python network programming ?

Tuliskan jawaban anda pada doc ms word kemudian upload pada assingment ini.

Socket adalah penghubung antara dua aplikasi yang dapat berkomunikasi satu sama lain (baik secara lokal pada satu mesin atau secara jarak jauh antara dua mesin di lokasi terpisah).

Pada dasarnya, socket berfungsi sebagai tautan komunikasi antara dua entitas, yaitu server dan klien. Server akan memberikan informasi yang diminta oleh klien. Misalnya, ketika Anda mengunjungi halaman ini, browser membuat socket dan terhubung ke server.

Modul Socket

Untuk membuat socket, Anda menggunakan fungsi socket.socket(), dan sintaksnya sesederhana:

```
1  import socket
2  s= socket.socket (socket_family, socket_type, protocol=0)
```

Berikut uraian argumennya:

- **socket_family**: Mewakili keluarga alamat (dan protokol). Ini bisa berupa AF_UNIX atau AF_INET.
- **socket_type**: Mewakili jenis soket, dan dapat berupa SOCK_STREAM atau SOCK_DGRAM.
- **protocol**: Ini adalah argumen opsional, dan biasanya default ke 0.

Setelah mendapatkan objek socket Anda, Anda kemudian dapat membuat server atau klien sesuai keinginan menggunakan metode yang tersedia di modul socket.

Nama : Oktariansyah

Nim : 202420006

A. Network Socket

Network socket merupakan alamat yang mengandung data alamat ip address dan nomor port. Singkatnya, socket merupakan cara yang mudah untuk berkomunikasi dengan komputer lain. Oleh karena itu, socket merupakan suatu proses yang dapat berkomunikasi dengan proses yang lain melalui jaringan.

Pada bahasa pemrograman python, untuk membuat socket menggunakan fungsi socket.socket() yang tersedia pada modul socket. Sintaks standar dari fungsi socket adalah:

s = socket.socket(socket_family, socket_type, protocol=0)

Deskripsi parameter dari fungsi socket diatas adalah sebagai berikut:

socket_family: socket.AF_INET, PF_PACKET

- AF_INET merupakan alamat untuk IPv4.
- PF_PACKET merupakan device driver layer. Umumnya merupakan library pcap yang digunakan pada linux.

B. Cara Kerja Method Socket Server

Dalam konsep arsitektur client-server, terdapat dua layanan yang berbeda dari masing-masing perangkat. Server bertugas secara terpusat untuk memberikan service/layanan yang diminta oleh client. Sedangkan client bertugas untuk mengirimkan permintaan dan menerima layanan dari server.

Beberapa metode pada fungsi socket di python, yaitu:

- socket.bind(address): Method ini digunakan untuk menghubungkan alamat ip dengan nomor port ke socket. Socket harus dibuka dahulu sebelum terhubung dengan alamat tersebut.
- socket.listen(q): Method ini akan memulai fase mendengarkan koneksi TCP. Argumen q mendefinisikan jumlah koneksi maksimum yang dapat ditangani server.
- socket.accept(): Penggunaan method ini adalah untuk menerima koneksi yang dikirim dari client. Sebelum menggunakan method ini, method socket.bind(address) dan socket.listen(q) harus digunakan terlebih dahulu. Method socket.accept() akan mengembalikan dua nilai yaitu: client_socket dan address, dimana client_socket adalah objek socket baru yang digunakan untuk mengirim dan menerima data selama terhubung, dan address adalah alamat client.

C. Method Socket Client

Method yang terdapat untuk fungsi di socket client adalah:

socket.connect(address): Method ini untuk menghubungkan client ke server. Argumen address adalah alamat servernya.

D. Method Socket

Beberapa fungsi yang terdapat pada method socket adalah sebagai berikut:

- socket.recv(bufsize): Method ini menerima pesan TCP dari socket. Argumen bufsize mendefinisikan jumlah data maksimum yang dapat diterima dalam suatu waktu.
- socket.recvfrom(bufsize): Method ini menerima data dari socket. Method ini akan mengembalikan sepasang nilai, nilai pertama akan memberikan informasi penerimaan data, nilai kedua akan memberikan alamat socket untuk melakukan pengiriman data
- socket.recv_into(buffer): Method ini menerima data kurang dari atau sama dengan argumen buffer. Parameter buffer dibuat oleh method bytearray()
- socket.recvfrom_into(buffer): Method ini mempunyai data dari socket dan mengirimkan melalui buffer. Nilai kembalian adalah nbytes dan address, dimana nbytes adalah jumlah bytes yang diterima, dan address adalah alamat socket pada saat mengirim data.
- socket.send(bytes): Method ini digunakan untuk mengirimkan data ke socket. Sebelum mengirim data, pastikan bahwa socket sudah terhubung ke mesin. Method ini akan mengembalikan jumlah byte yang terkirim.
- socket.sendto(data, address): Method ini digunakan untuk mengirim data ke socket. Secara umum, method ini menggunakan UDP. UDP merupakan protocol yang bersifat connectionless (tidak memperdulikan apakah paket sudah terkirim atau belum yang penting sudah dikirimkan oleh si pengirim (server/client)).
- socket.sendall(data): Method ini akan mengirimkan semua data ke socke

Nama : Oman Arrohman
NPM : 202420042
Kelas : MTI Regular A

Tugas Python Programming

Tugas: Jelaskan apa yang dimaksud dengan socket pada python network programming ?

Jawaban :

Soket adalah titik akhir dari saluran komunikasi dua arah. Soket dapat berkomunikasi dalam suatu proses, antara proses pada mesin yang sama, atau antara proses dari jarak jauh, selain itu juga soket dapat diimplementasikan melalui sejumlah jenis saluran yang berbeda : soket domain Unix, TCP, UDP, dan sebagainya.

Python menyediakan dua tingkat akses ke layanan jaringan. Pada tingkat rendah, Anda dapat mengakses dukungan soket dasar dalam sistem operasi yang mendasarinya, yang memungkinkan Anda untuk mengimplementasikan klien dan server untuk kedua protokol berorientasi koneksi dan tanpa sambungan.

Nama : Puspita Dewi Setyadi

Nim : 202420011

Kelas : REG. A 23

Socket adalah mekanisme komunikasi yang memungkinkan terjadinya pertukaran data antar program atau proses baik dalam satu mesin maupun antar mesin.

Soket dapat diimplementasikan melalui sejumlah jenis saluran yang berbeda: soket domain Unix, TCP, UDP, dan sebagainya. Pustaka socket menyediakan kelas khusus untuk menangani transportasi umum serta antarmuka umum untuk menangani sisanya.

Modul Socket

Untuk membuat soket, Anda harus menggunakan fungsi socket.socket () yang tersedia dalam modul soket, yang memiliki sintaks umum

s = socket.socket (socket_family, socket_type, protocol=0)

Server Socket Method

Method	Penjelasan
s.bind()	This method binds address (hostname, port number pair) to socket.
s.listen()	This method sets up and start TCP listener.
s.accept()	This passively accept TCP client connection, waiting until connection arrives (blocking).

Client Socket Method

Method	Penjelasan
s.connect()	This method actively initiates TCP server connection.

General Method Socket

Method	Penjelasan
s.recv()	This method receives TCP message
s.send()	This method transmits TCP message
s.recvfrom()	This method receives UDP message
s.sendto()	This method transmits UDP message
s.close()	This method closes socket
socket.gethostname()	Returns the hostname.

```
#!/usr/bin/python          # This is server.py file

import socket              # Import socket module

s = socket.socket()        # Create a socket object
host = socket.gethostname() # Get local machine name
port = 12345               # Reserve a port for your service.
s.bind((host, port))       # Bind to the port

s.listen(5)                # Now wait for client connection.
while True:
  c, addr = s.accept()     # Establish connection with client.
  print 'Got connection from', addr
  c.send('Thank you for connecting')
  c.close()                # Close the connection
```

Server Sederhana

Untuk menulis server Internet, kami menggunakan fungsi soket yang tersedia di modul soket untuk membuat objek soket. Objek soket kemudian digunakan untuk memanggil fungsi lain untuk menyiapkan server soket.

Sekarang sebut bind(hostname,port) berfungsi untuk menentukan port untuk layanan Anda pada host yang diberikan. Selanjutnya, panggil metode penerimaan objek yang

dikembalikan. Metode ini menunggu sampai klien terhubung ke port yang Anda tentukan, dan kemudian mengembalikan objek koneksi yang mewakili koneksi ke klien itu.

Client Sederhana

Mari kita menulis program klien yang sangat sederhana yang membuka koneksi ke port yang diberikan 12345 dan host yang diberikan. Ini sangat sederhana untuk membuat klien soket menggunakan fungsi modul soket Python.

Socket.connect (hosname, port) membuka koneksi TCP ke hostname pada port. Setelah Anda memiliki soket terbuka, Anda dapat membaca darinya seperti objek IO apa pun. Setelah selesai, jangan lupa untuk menutupnya, karena Anda akan menutup file.

Kode berikut adalah klien yang sangat sederhana yang terhubung ke host dan port yang diberikan, membaca data yang tersedia dari soket, dan kemudian keluar

```
#!/usr/bin/python          # This is client.py file

import socket              # Import socket module

s = socket.socket()        # Create a socket object
host = socket.gethostname() # Get local machine name
port = 12345               # Reserve a port for your service.

s.connect((host, port))
print s.recv(1024)
s.close                    # Close the socket when done
```

Sekarang jalankan server.py ini di latar belakang dan kemudian jalankan di atas client.py untuk melihat hasilnya.

Jalankan server. python server.py & Setelah server berjalan lanjutkan

Jalankan client: python client.py

Hasilnya akan seperti ini : Got connection from ('127.0.0.1', 48437) Thank you for connecting

Modul Internet pada Python

Protocol	**Common function**	**Port No**	**Python module**
HTTP	Web pages	80	httplib, urllib, xmlrpclib
NNTP	Usenet news	119	Nntplib
FTP	Transfer file	20	ftplib, urllib
SMTP	Mengirim email	25	Smtplib
POP3	Fetching email	110	Poplib
IMAP4	Fetching email	143	Imaplib
Telnet	Command lines	23	telnetlib
Gopher	Document transfers	70	gopherlib, urllib

Nama : Rachmad Iqbal

Nim : 202420002

A. Network Socket

Network socket merupakan alamat yang mengandung data alamat ip address dan nomor port. Singkatnya, socket merupakan cara yang mudah untuk berkomunikasi dengan komputer lain. Oleh karena itu, socket merupakan suatu proses yang dapat berkomunikasi dengan proses yang lain melalui jaringan.

Pada bahasa pemrograman python, untuk membuat socket menggunakan fungsi socket.socket() yang tersedia pada modul socket. Sintaks standar dari fungsi socket adalah:

s = socket.socket(socket_family, socket_type, protocol=0)

Deskripsi parameter dari fungsi socket diatas adalah sebagai berikut:

socket_family: socket.AF_INET, PF_PACKET

- AF_INET merupakan alamat untuk IPv4.
- PF_PACKET merupakan device driver layer. Umumnya merupakan library pcap yang digunakan pada linux.

B. Cara Kerja Method Socket Server

Dalam konsep arsitektur client-server, terdapat dua layanan yang berbeda dari masing-masing perangkat. Server bertugas secara terpusat untuk memberikan service/layanan yang diminta oleh client. Sedangkan client bertugas untuk mengirimkan permintaan dan menerima layanan dari server.

Beberapa metode pada fungsi socket di python, yaitu:

- socket.bind(address): Method ini digunakan untuk menghubungkan alamat ip dengan nomor port ke socket. Socket harus dibuka dahulu sebelum terhubung dengan alamat tersebut.
- socket.listen(q): Method ini akan memulai fase mendengarkan koneksi TCP. Argumen q mendefinisikan jumlah koneksi maksimum yang dapat ditangani server.
- socket.accept(): Penggunaan method ini adalah untuk menerima koneksi yang dikirim dari client. Sebelum menggunakan method ini, method socket.bind(address) dan socket.listen(q) harus digunakan terlebih dahulu. Method socket.accept() akan mengembalikan dua nilai yaitu: client_socket dan address, dimana client_socket adalah objek socket baru yang digunakan untuk mengirim dan menerima data selama terhubung, dan address adalah alamat client.

C. Method Socket Client

Method yang terdapat untuk fungsi di socket client adalah:

socket.connect(address): Method ini untuk menghubungkan client ke server. Argumen address adalah alamat servernya.

D. Method Socket

Beberapa fungsi yang terdapat pada method socket adalah sebagai berikut:

- socket.recv(bufsize): Method ini menerima pesan TCP dari socket. Argumen bufsize mendefinisikan jumlah data maksimum yang dapat diterima dalam suatu waktu.
- socket.recvfrom(bufsize): Method ini menerima data dari socket. Method ini akan mengembalikan sepasang nilai, nilai pertama akan memberikan informasi penerimaan data, nilai kedua akan memberikan alamat socket untuk melakukan pengiriman data
- socket.recv_into(buffer): Method ini menerima data kurang dari atau sama dengan argumen buffer. Parameter buffer dibuat oleh method bytearray()
- socket.recvfrom_into(buffer): Method ini mempunyai data dari socket dan mengirimkan melalui buffer. Nilai kembalian adalah nbytes dan address, dimana nbytes adalah jumlah bytes yang diterima, dan address adalah alamat socket pada saat mengirim data.
- socket.send(bytes): Method ini digunakan untuk mengirimkan data ke socket. Sebelum mengirim data, pastikan bahwa socket sudah terhubung ke mesin. Method ini akan mengembalikan jumlah byte yang terkirim.
- socket.sendto(data, address): Method ini digunakan untuk mengirim data ke socket. Secara umum, method ini menggunakan UDP. UDP merupakan protocol yang bersifat connectionless (tidak memperdulikan apakah paket sudah terkirim atau belum yang penting sudah dikirimkan oleh si pengirim (server/client)).
- socket.sendall(data): Method ini akan mengirimkan semua data ke socke

Nama : Ribhan Mandala

Nim : 202420035

A. Network Socket

Network socket merupakan alamat yang mengandung data alamat ip address dan nomor port. Singkatnya, socket merupakan cara yang mudah untuk berkomunikasi dengan komputer lain. Oleh karena itu, socket merupakan suatu proses yang dapat berkomunikasi dengan proses yang lain melalui jaringan.

Pada bahasa pemrograman python, untuk membuat socket menggunakan fungsi socket.socket() yang tersedia pada modul socket. Sintaks standar dari fungsi socket adalah:

s = socket.socket(socket_family, socket_type, protocol=0)

Deskripsi parameter dari fungsi socket diatas adalah sebagai berikut:

socket_family: socket.AF_INET, PF_PACKET

- AF_INET merupakan alamat untuk IPv4.
- PF_PACKET merupakan device driver layer. Umumnya merupakan library pcap yang digunakan pada linux.

B. Cara Kerja Method Socket Server

Dalam konsep arsitektur client-server, terdapat dua layanan yang berbeda dari masing-masing perangkat. Server bertugas secara terpusat untuk memberikan service/layanan yang diminta oleh client. Sedangkan client bertugas untuk mengirimkan permintaan dan menerima layanan dari server.

Beberapa metode pada fungsi socket di python, yaitu:

- socket.bind(address): Method ini digunakan untuk menghubungkan alamat ip dengan nomor port ke socket. Socket harus dibuka dahulu sebelum terhubung dengan alamat tersebut.
- socket.listen(q): Method ini akan memulai fase mendengarkan koneksi TCP. Argumen q mendefinisikan jumlah koneksi maksimum yang dapat ditangani server.
- socket.accept(): Penggunaan method ini adalah untuk menerima koneksi yang dikirim dari client. Sebelum menggunakan method ini, method socket.bind(address) dan socket.listen(q) harus digunakan terlebih dahulu. Method socket.accept() akan mengembalikan dua nilai yaitu: client_socket dan address, dimana client_socket adalah objek socket baru yang digunakan untuk mengirim dan menerima data selama terhubung, dan address adalah alamat client.

C. Method Socket Client

Method yang terdapat untuk fungsi di socket client adalah:

socket.connect(address): Method ini untuk menghubungkan client ke server. Argumen address adalah alamat servernya.

D. Method Socket

Beberapa fungsi yang terdapat pada method socket adalah sebagai berikut:

- socket.recv(bufsize): Method ini menerima pesan TCP dari socket. Argumen bufsize mendefinisikan jumlah data maksimum yang dapat diterima dalam suatu waktu.
- socket.recvfrom(bufsize): Method ini menerima data dari socket. Method ini akan mengembalikan sepasang nilai, nilai pertama akan memberikan informasi penerimaan data, nilai kedua akan memberikan alamat socket untuk melakukan pengiriman data
- socket.recv_into(buffer): Method ini menerima data kurang dari atau sama dengan argumen buffer. Parameter buffer dibuat oleh method bytearray()
- socket.recvfrom_into(buffer): Method ini mempunyai data dari socket dan mengirimkan melalui buffer. Nilai kembalian adalah nbytes dan address, dimana nbytes adalah jumlah bytes yang diterima, dan address adalah alamat socket pada saat mengirim data.
- socket.send(bytes): Method ini digunakan untuk mengirimkan data ke socket. Sebelum mengirim data, pastikan bahwa socket sudah terhubung ke mesin. Method ini akan mengembalikan jumlah byte yang terkirim.
- socket.sendto(data, address): Method ini digunakan untuk mengirim data ke socket. Secara umum, method ini menggunakan UDP. UDP merupakan protocol yang bersifat connectionless (tidak memperdulikan apakah paket sudah terkirim atau belum yang penting sudah dikirimkan oleh si pengirim (server/client)).
- socket.sendall(data): Method ini akan mengirimkan semua data ke socke

Nama : Yudistira Sira Permana
NIM : 182420104
Kelas : MTI 23 A

Socket pada Phyton Network Programming

Socket pada Phyton Network Programming merupakan titik akhir dari saluran komunikasi dua arah yang memungkinkan melakukan implementasi klien dan server untuk protokol berorientasi koneksi dan tanpa koneksi. Soket dapat berkomunikasi dalam suatu proses, antar proses pada mesin yang sama, atau antar proses di benua yang berbeda dengan memiliki pustaka yang menyediakan akses tingkat tinggi ke protokol jaringan tingkat aplikasi tertentu, seperti FTP, HTTP, dan sebagainya.

Pada bahasa pemrograman python, untuk membuat socket menggunakan fungsi socket.socket() yang tersedia pada modul socket. Sintaks standar dari fungsi socket adalah:

s = socket.socket(socket_family, socket_type, protocol=0)

Deskripsi parameter dari fungsi socket diatas adalah sebagai berikut:

socket_family: socket.AF_INET, PF_PACKET

- AF_INET merupakan alamat untuk IPv4.
- PF_PACKET merupakan device driver layer. Umumnya merupakan library pcap yang digunakan pada linux.

Dalam konsep arsitektur client-server, terdapat dua layanan yang berbeda dari masing-masing perangkat. Server bertugas secara terpusat untuk memberikan service/layanan yang diminta oleh client. Sedangkan client bertugas untuk mengirimkan permintaan dan menerima layanan dari server.

Nama : Yusria Lenitasari

NIM : 202420003

Jurusan : Magister Teknologi Informatika

Tugas 03 : *Computer Network and Communication*

---

1. Jelaskan apa yang dimaksud dengan socket pada python network programming ?

   Jawab :

   Soket adalah titik akhir dari saluran komunikasi dua arah. Soket dapat berkomunikasi dalam suatu proses, antara proses pada mesin yang sama, atau antara proses di berbagai benua.

   Soket dapat diimplementasikan melalui sejumlah jenis saluran yang berbeda seperti soket domain Unix, TCP, UDP, dan sebagainya. Pustaka socket menyediakan kelas khusus untuk menangani transportasi umum serta antarmuka umum untuk menangani sisanya.

   Python memiliki dua tingkat layanan jaringan akses. :

   1. Layanan jaringan tingkat rendah

      Layanan pada tingkat ini, kita dapat mengakses dukungan soket dasar dalam system operasi yang memungkinkan untuk mengimplementasikan klien dan server untuk kedua protocol berorientasi koneksi tanpa sambungan. Python memberikan standar BSD Socket API, Anda dapat mengakses semua metode yang mendasari sistem operasi Socket antarmuka

   2. Layanan jaringan tingkat tinggi

      Pada layanan tingkat tinggi modul layanan jaringan SocketServer menyediakan kelas server pusat seperti FTP, HTTP dan seterusya

   Network socket merupakan alamat yang mengandung data alamat IP address dan nomor port. Singkatnya, socket merupakan cara yang mudah untuk

berkomunikasi dengan komputer lain. Oleh karena itu, socket merupakan suatu proses yang dapat berkomunikasi dengan proses yang lain melalui jaringan.

Pada bahasa pemrograman python, untuk membuat socket menggunakan fungsi socket. socket() yang tersedia pada modul socket memiliki sintaks umum

```
s = socket.socket(socket_family, socket_type, protocol=0)
```

Deskripsi parameter dari fungsi socket diatas adalah sebagai berikut:

socket_family: socket.AF_INET, PF_PACKET

Keterangan :

AF_INET merupakan alamat untuk IPv4.

PF_PACKET merupakan device driver layer. Umumnya merupakan library pcap yang digunakan pada linux.

1. Socket Server

   Cara Kerja Metode Socket Server

   Dalam konsep arsitektur client-server, terdapat dua layanan yang berbeda dari masing-masing perangkat. Server bertugas secara terpusat untuk memberikan service/layanan yang diminta oleh client. Sedangkan client bertugas untuk mengirimkan permintaan dan menerima layanan dari server.

   Beberapa metode pada fungsi socket di python, yaitu:

   1. Socket.bind(address)

      Metode ini digunakan untuk menghubungkan alamat ip dengan nomor port ke socket. Socket harus dibuka dahulu sebelum terhubung dengan alamat tersebut.
   2. socket.listen(q)

      Metode ini akan memulai fase mendengarkan koneksi TCP. Argumen q mendefinisikan jumlah koneksi maksimum yang dapat ditangani server.

3. socket.accept()

   Penggunaan metode ini adalah untuk menerima koneksi yang dikirim dari client. Sebelum menggunakan metode ini, metode socket.bind(address) dan socket.listen(q) harus digunakan terlebih dahulu. Metode socket.accept() akan mengembalikan dua nilai yaitu: client_socket dan address, dimana client_socket adalah objek socket baru yang digunakan untuk mengirim dan menerima data selama terhubung, dan address adalah alamat client.

2. Socket Client

   Metode Socket Client

   Metode yang terdapat untuk fungsi di socket client adalah

   1. socket.connect(address): Metode ini untuk menghubungkan client ke server. Argumen address adalah alamat servernya.

Metode Socket Umum

Beberapa fungsi umumyang terdapat pada metode socket adalah sebagai berikut :

No	Metode	Penjelasan	Keterangan
1	socket.recv(bufsize)\ s.recv()	Metode ini menerima pesan TCP dari socket	Argumen bufsize mendefinisikan jumlah data maksimum yang dapat diterima dalam suatu waktu
2	socket.recvfrom(bufsize)\ s.recvfrom()	Metode ini menerima data dari socket	Metode ini akan mengembalikan sepasang nilai, nilai pertama akan memberikan informasi penerimaan data, nilai kedua akan memberikan alamat socket untuk melakukan pengiriman data
3	socket.recv_into(buffer)\	Metode ini menerima data kurang dari atau sama dengan argumen buffer	Parameter buffer dibuat oleh metode bytearray()

4	socket.recvfrom_into(buffer)	Metode ini mempunyai data dari socket dan mengirimkan melalui buffer	Nilai kembalian adalah nbytes dan address, dimana nbytes adalah jumlah bytes yang diterima, dan address adalah alamat socket pada saat mengirim data.
5	socket.send(bytes)\ s.send()	Metode ini digunakan untuk mengirimkan data ke socket	Sebelum mengirim data, pastikan bahwa socket sudah terhubung ke mesin. Metode ini akan mengembalikan jumlah byte yang terkirim.
6	socket.sendto(data, address)\ s.sendto()	Metode ini digunakan untuk mengirim data ke socket	Secara umum, metode ini menggunakan UDP. UDP merupakan protocol yang bersifat connectionless (tidak memperdulikan apakah paket sudah terkirim atau belum yang penting sudah dikirimkan oleh si pengirim (server/client)).
7	socket.sendall(data)	Metode ini akan mengirimkan semua data ke socket	

Berikut ini terdapat kode program client server sederhana:

Nama file : serverku.py

```
import socket

host = "192.168.0.1"
port = 12345
```

```
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind((host,port))
s.listen(2)
conn, addr = s.accept()
print addr, "Selamat Anda Sudah Terhubung dengan Serverku.py"
conn.send("Terima Kasih karena telah berkomunikasi dengan Serverku.py")
conn.close()
```

Nama file : clientku.py

```
import socket

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
host = "192.168.0.1"
port =12345
s.connect((host,port))
print s.recv(1024)
s.send("Hai Serverku.py, Clientku.py ingin berkomunikasi")
s.close()
```

Output dari kode program diatas adalah:

```
mylabs:latihan triawan$ python server1.py
('127.0.0.1', 50632) Selamat Anda Sudah Terhubung dengan Serverku.py
mylabs:latihan triawan$
```

Output Serverku.py

```
mylabs:latihan triawan$ python client1.py
Terima Kasih karena telah berkomunikasi dengan Serverku.py
mylabs:latihan triawan$
```

Metode konektivitas client server diatas hanya untuk menangani satu permintaan yang dikirim oleh client. Apabila menginginkan server socket menangani lebih dari satu service, maka tinggal tambahkan looping setelah statement listen.

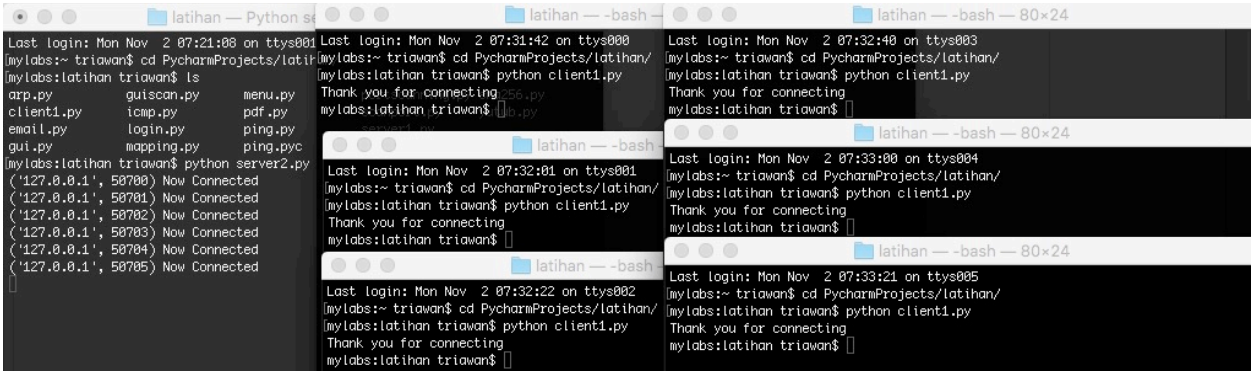
Berikut ini kode program lengkapnya :

Nama file : serverku2.py

```
import socket
        host = "0.0.0.0"

        port = 12345

        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

```
s.bind((host, port))
s.listen(2)
while True:
    conn, addr = s.accept()
    print addr, "Now Connected"
    conn.send("Thank you for connecting")
    conn.close()
```

Output dari program diatas adalah sebagai berikut:



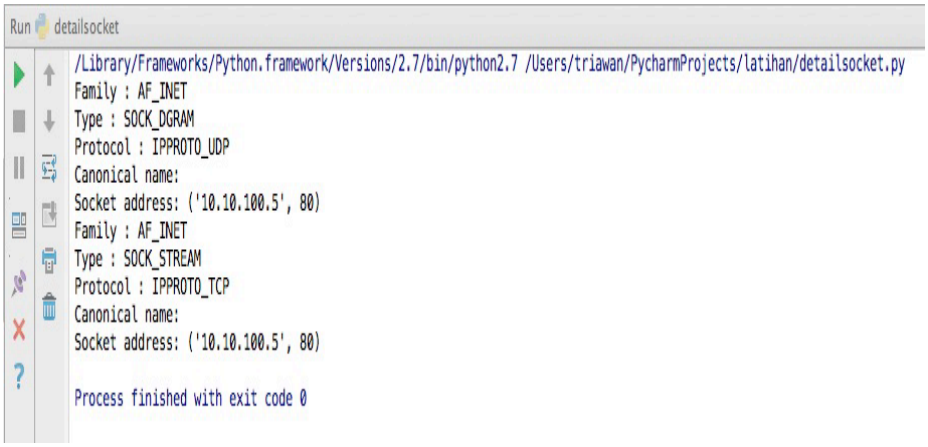
Kode program client server sederhana sudah selesai. Berikut ini terdapat kode program terakhir untuk mengetahui secara detail alamat ip dan port beserta tipenya.

Nama file : detailsocket.py

```
import socket
def get_protnumber(prefix):
    return dict((getattr(socket, a), a)
                for a in dir(socket)
                if a.startswith(prefix))
```

```
proto_fam = get_protnumber('AF_')

types = get_protnumber('SOCK_')

protocols = get_protnumber('IPPROTO_')

for res in socket.getaddrinfo('www.unmuhjember.ac.id', 'http'):

    family, socktype, proto, canonname, sockaddr = res


    print 'Family :', proto_fam[family]

    print 'Type :', types[socktype]

    print 'Protocol :', protocols[proto]

    print 'Canonical name:', canonname

    print 'Socket address:', sockaddr
```

Output Program tersebut adalah :



Modul Internet pada Python

Berikut tabel daftar beberapa modul penting dalam pemrograman Jaringan / Internet Python.

Protocol	Common function	Port No	Python module
HTTP	Web pages	80	httplib, urllib, xmlrpclib
NNTP	Usenet news	119	nntplib
FTP	Transfer file	20	ftplib, urllib
SMTP	Mengirim email	25	smtplib
POP3	Fetching email	110	poplib
IMAP4	Fetching email	143	imaplib
Telnet	Command lines	23	telnetlib
Gopher	Document transfers	70	gopherlib, urllib

Aldo Fajarino
202420004
MTI Reg B
Tugas Python Programming

Socket adalah penghubung antara dua aplikasi yang dapat berkomunikasi satu sama lain (baik secara lokal pada satu mesin atau secara jarak jauh antara dua mesin di lokasi terpisah).

Titik akhir dari saluran komunikasi dua arah. Soket dapat berkomunikasi dalam suatu proses, antara proses pada mesin yang sama, atau antara proses di berbagai benua.

Soket dapat diimplementasikan melalui sejumlah jenis saluran yang berbeda: soket domain Unix, TCP, UDP, dan sebagainya. Pustaka socket menyediakan kelas khusus untuk menangani transportasi umum serta antarmuka umum untuk menangani sisanya.

Pada dasarnya, socket berfungsi sebagai tautan komunikasi antara dua entitas, yaitu server dan klien. Server akan memberikan informasi yang diminta oleh klien. Misalnya, ketika Anda mengunjungi halaman ini, browser membuat socket dan terhubung ke server.

Modul Socket
Untuk membuat socket, Anda menggunakan fungsi socket.socket(), dan sintaksnya sesederhana:

```
import socket
s= socket.socket (socket_family, socket_type, protocol=0)
```
Berikut uraian argumennya:

socket_family: Mewakili keluarga alamat (dan protokol). Ini bisa berupa AF_UNIX atau AF_INET.
socket_type: Mewakili jenis soket, dan dapat berupa SOCK_STREAM atau SOCK_DGRAM.
protocol: Ini adalah argumen opsional, dan biasanya default ke 0.
Setelah mendapatkan objek socket Anda, Anda kemudian dapat membuat server atau klien sesuai keinginan menggunakan metode yang tersedia di modul socket.

Server Sederhana
Untuk menulis server Internet, kami menggunakan fungsi soket yang tersedia di modul soket untuk membuat objek soket. Objek soket kemudian digunakan untuk memanggil fungsi lain untuk menyiapkan server soket.

Sekarang sebut bind(hostname,port) berfungsi untuk menentukan port untuk layanan Anda pada host yang diberikan.

Selanjutnya, panggil metode penerimaan objek yang dikembalikan. Metode ini menunggu sampai klien terhubung ke port yang Anda tentukan, dan kemudian mengembalikan objek koneksi yang mewakili koneksi ke klien itu.

Client Sederhana
Mari kita menulis program klien yang sangat sederhana yang membuka koneksi ke port yang diberikan 12345 dan host yang diberikan. Ini sangat sederhana untuk membuat klien soket menggunakan fungsi modul soket Python.

Socket.connect (hosname, port) membuka koneksi TCP ke hostname pada port. Setelah Anda memiliki soket terbuka, Anda dapat membaca darinya seperti objek IO apa pun. Setelah selesai, jangan lupa untuk menutupnya, karena Anda akan menutup file.

Kode berikut adalah klien yang sangat sederhana yang terhubung ke host dan port yang diberikan, membaca data yang tersedia dari soket, dan kemudian keluar

```
#!/usr/bin/python          # This is client.py file

import socket              # Import socket module

s = socket.socket()        # Create a socket object
host = socket.gethostname() # Get local machine name
port = 12345               # Reserve a port for your service.

s.connect((host, port))
print s.recv(1024)
s.close                    # Close the socket when done
```

Sekarang jalankan server.py ini di latar belakang dan kemudian jalankan di atas client.py untuk melihat hasilnya.

Jalankan server.
python server.py &

Setelah server berjalan lanjutkan

Jalankan client:
python client.py

Hasilnya akan seperti ini : Got connection from ('127.0.0.1', 48437) Thank you for connecting

Modul Internet pada Python
Berikut tabel daftar beberapa modul penting dalam pemrograman Jaringan / Internet Python.

Protocol	Common function	Port No	Python module
HTTP	Web pages	80	httplib, urllib, xmlrpclib
NNTP	Usenet news	119	nntplib
FTP	Transfer file	20	ftplib, urllib
SMTP	Mengirim email	25	smtplib
POP3	Fetching email	110	poplib
IMAP4	Fetching email	143	imaplib
Telnet	Command lines	23	telnetlib
Gopher	Document transfers	70	gopherlib, urllib

Jelaskan apa yang dimaksud dengan socket pada python network programming ?

Jawab :

Socket adalah titik akhir dari saluran komunikasi dua arah. Socket dapat berkomunikasi dalam suatu proses, antara proses pada mesin yang sama, atau antara proses di berbagai benua.

Nama : Cornelia Tri Wahyuni

NPM : 202420044

Kelas : MTI Regular A

Tugas: Jelaskan apa yang dimaksud dengan socket pada python network programming ?

Jawaban :

Socket merupakan alamat yang mengandung data alamat ip address dan nomor port. Singkatnya, socket merupakan cara yang mudah untuk berkomunikasi dengan komputer lain. Oleh karena itu, socket merupakan suatu proses yang dapat berkomunikasi dengan proses yang lain melalui jaringan.

Pada bahasa pemrograman python, untuk membuat socket menggunakan fungsi socket.socket() yang tersedia pada modul socket.

Python menyediakan dua tingkat akses ke layanan jaringan. Pada tingkat rendah, Anda dapat mengakses dukungan soket dasar dalam sistem operasi yang mendasarinya, yang memungkinkan Anda untuk mengimplementasikan klien dan server untuk kedua protokol berorientasi koneksi dan tanpa sambungan.

Nama : Dhea Noranita Putri
Nim : 182420112

SOAL :

Jelaskan apa yang dimaksud dengan socket pada python network programming ?

JAWAB :

Python menyediakan dua tingkat akses ke layanan jaringan. Pada tingkat rendah, yang dapat mengakses dukungan soket dasar dalam sistem operasi yang mendasarinya, yang memungkinkan untuk mengimplementasikan klien dan server untuk kedua protokol berorientasi koneksi dan tanpa sambungan. Python juga memiliki pustaka yang menyediakan akses tingkat lebih tinggi ke protokol jaringan tingkat aplikasi tertentu, seperti FTP, HTTP, dan seterusnya. Soket adalah titik akhir dari saluran komunikasi dua arah. Soket dapat berkomunikasi dalam suatu proses, antara proses pada mesin yang sama, atau antara proses di berbagai benua. Soket dapat diimplementasikan melalui sejumlah jenis saluran yang berbeda: soket domain Unix, TCP, UDP, dan sebagainya. Pustaka socket menyediakan kelas khusus untuk menangani transportasi umum serta antarmuka umum untuk menangani sisanya.

Penggunaan Dasar Socket pada Python, Phython hanya menggunakan dua domain komunikasi, yaitu : UNIX (AF_UNIX) danInternet (AF_INET) domain. Pengalamatan pada UNIX domain direpresentasikan sebagaistring, dinamakan dalam lokal path: contoh/tmp/sock. Sedangkan pengalamatan Internetdomain direpresentasikan sebagai tuple(host,port), dimana host merupakanstring yangmerepresentasikan namahost internet yang sah (hostname), misalnya :darkstar.drslump.net atau berupa IP address dalam notasi dotted decimal, misalnya :192.168.1.1. Dan port merupakan nomor port yang sah antara 1 sampai 65535. Tetapidalam keluarga UNIX penggunaan port di bawah 1024 memerlukan akses root privileges.Sebelum menggunakan modul socket dalam Python, maka modul socket harus terlebih dahuludiimport.

Berikut Proses contohnya :

1. Membuat Socket (Creating)Socket dibuat melalui pemanggilan socket(family, type[, Proto]).
2. Menghubungkan Socket (Connecting). Sebuahserver dari sudut pandang kita adalah sebuah proses yang mendengarkan (listen) padaport tertentu. Ketika proses lain ingin berhubungan dengan server atau menggunakan layananserver, maka proses harus

terhubung dengan alamat dan nomor port tertentu yangdispesifikasikan oleh server. Ini dilakukan dengan memanggil metode socketconnect(address), dimana address adalah sebuah tuple(host, port) untuk Internetdomain dan pathname untuk UNIX domain.

3. Mengikatkan Socket ke Port (Binding). Setelah socket berhasil dibuat, maka Python akan mengembalikan sebuah socket descriptor.Sebelum digunakan, maka socket harus diikatkan (binding) ke alamat dan nomor port yangsesuai agar proses lain dapat ditujukan ke socket.
4. Mendengarkan Koneksi (Listening). telah socket diikatkan (bind), langkah selanjutnya adalah memanggil methodlisten(queue). Perintah ini mengistruksikan socket untuk listen pada port-port yang telahdiikatkan (bind), dan queue merupakan sebuah integer yang merepresentasikan maksimumantrian koneksi.
5. Menerima Koneksi (Accepting). Untuk menerima koneksi dari permintaan (request) client pada koneksi yang menggunakansocket stream (TCP). Method yang digunakan accept()
6. Mengirim Data ke Koneksi (Sending) Menerima koneksi tidak akan berarti tanpa digunakan untuk mengirim dan menerima data.Oleh karena itu digunakan method send(string) untuk socket stream (TCP) dansendto(string,address) untuk socket datagram (UDP).
7. Menerima Data Dari Koneksi (Receiving). Untuk menerima data yang dikirim dari server digunakan method recv(bufsize) untuksocket stream dan recvfrom(bufsize).
8. Menutup Koneksi (Closing) Untuk menutup koneksi yang telah dibuat digunakan method close(s).

Nama : Dhea Noranita Putri
Nim : 182420112

MK : COMPUTER NETWORK AND COMUNICATION

Nama : Enggi Ardius
Nim : 202420007

Pemrograman Socket dengan Python

Python merupakan salah satu bahasa pemrograman yang populer terutama di bidang keamanan komputer. Dalam modul ini, beberapa percobaan pemrograman untuk mendukung proses keamanan komputer ditulis menggunakan python versi 2. Jika tidak ingin ribet, maka gunakan sistem operasi linux varian terbaru, misalnya: ubuntu, kali linux, dan begitu seterusnya. Bahasa pemrograman python dan modulnya secara otomatis terinstal di sistem operasi linux. Soket jaringan adalah alamat yang berisi data untuk alamat ip dan nomor port. Singkatnya, soket adalah cara mudah untuk berkomunikasi dengan komputer lain. Oleh karena itu, soket merupakan suatu proses yang dapat berkomunikasi dengan proses lain melalui jaringan. Dalam bahasa pemrograman python, untuk membuat socket menggunakan fungsi socket.socket () yang tersedia pada modul socket. Sintaks standar dari fungsi soket adalah: s = socket.socket (socket_family, socket_type, protocol = 0) Uraian parameter fungsi socket di atas adalah sebagai berikut: socket_family: socket.AF_INET, PF_PACKET AF_INET adalah alamat untuk IPv4. PF_PACKET adalah lapisan driver perangkat. Umumnya ini adalah pustaka pcap yang digunakan di linux.

Penggunaan Dasar Socket

Python hanya menggunakan dua domain komunikasi, yaitu : UNIX (AF_UNIX) dan Internet (AF_INET) domain. Pengalamatan pada UNIX domain direpresentasikan sebagai *string*, dinamakan dalam lokal path: contoh /tmp/sock. Sedangkan pengalamatan Internet domain direpresentasikan sebagai *tuple*(host,port), dimana *host* merupakan *string* yang merepresentasikan nama *host* internet yang sah (*hostname*), misalnya : darkstar.drslump.net atau berupa IP address dalam notasi *dotted decimal*, misalnya : 192.168.1.1. Dan port merupakan nomor port yang sah antara 1 sampai 65535. Tetapi dalam keluarga UNIX penggunaan port di bawah 1024 memerlukan akses *root privileges*. Sebelum menggunakan modul socket dalam Python, maka modul socket harus terlebih dahulu diimport. Berikut contohnya :

```
#!/usr/bin/env python
#Mengimport modul socket
import socket
# Mengimport seluruh konstanta, data, dan method
from socket import *
# Mengimport konstanta
from socket import AF_INET, SOCK_STREAM
```

Membuat Socket (*Creating*)

Socket dibuat melalui pemanggilan socket(family, type[, Proto]). Untuk lebih jelasnya dapat dilihat pada tabel 1 dan tabel 2 berikut ini :

Tabel 1 Konstanta Keluarga (Family) Protokol

Family	Penjelasan
AF_UNIX	Unix Domain Protocol
AF_INET	IPv4 Protocol
AF_INET6	IPv6 Protocol

Tabel 2 Konstanta Type Socket

Type	Penjelasan
SOCK_STREAM	Stream Socket (TCP)
SOCK_DGRAM	Datagram Socket (UDP)
SOCK_RAW	Raw Socket
SOCK_RDM	-
SOCK_SEQPACKET	-

Untuk proto bersifat opsional dan biasanya bernilai 0. Untuk membuat socket stream (TCP) internet domain digunakan *statement* berikut :

```
sock = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
```

Jika SOCK_STREAM diganti dengan SOCK_DGRAM berarti membuat socket datagram (UDP). Kemudian untuk membuat socket stream dalam UNIX domain :

```
sock = socket.socket(socket.AF_UNIX,socket.SOCK_STREAM)
```

Menghubungkan Socket (*Connecting*)

Sebuah *server* dari sudut pandang kita adalah sebuah proses yang mendengarkan (*listen*) pada port tertentu. Ketika proses lain ingin berhubungan dengan *server* atau menggunakan layanan *server*, maka proses harus terhubung dengan alamat dan nomor port tertentu yang dispesifikasikan

```
sock.connect (('localhost',12345)) atau
sock.connect (('192.168.1.1',12345))
```

oleh *server*. Ini dilakukan dengan memanggil metode socket connect(address), dimana address adalah sebuah *tuple* (host, port) untuk Internet domain dan pathname untuk UNIX domain. Berikut contohnya :

Sedangkan untuk UNIX domain,

```
sock.connect ('/tmp/sock') #Koneksi ke filesocket
```

Mengikatkan Socket ke Port (*Binding*)

Setelah socket berhasil dibuat, maka Python akan mengembalikan sebuah *socket descriptor*. Sebelum digunakan, maka socket harus diikatkan (*binding*) ke alamat dan nomor port yang sesuai agar proses lain dapat ditujukan ke socket. Berikut ini contoh untuk *binding* socket pada internet domain :

```
sock.bind(('localhost',12345)) atau
sock.bind(('192.168.1.1',12345))
```

Sedangkan untuk mengikatkan (*binding)* socket pada UNIX domain digunakan :

```
sock.bind('/tmp/sock')                    #/tmp/sock merupakan file socket
```

Perintah di atas akan membuat file pipe /tmp/sock yang dapat digunakan untuk berkomunikasi antara *server* dan *client*.

Mendengarkan Koneksi (*Listening*)

Setelah socket diikatkan (*bind*), langkah selanjutnya adalah memanggil method listen(queue). Perintah ini mengistruksikan socket untuk *listen* pada port-port yang telah diikatkan (*bind*), dan queue merupakan sebuah integer yang merepresentasikan maksimum antrian koneksi, berikut contoh penggunaannya :

```
sock.listen(5) #Mendengarkan koneksi dengan maksimum
                antrian sebanyak 5
```

Menerima Koneksi (*Accepting*)

Untuk menerima koneksi dari permintaan (*request*) *client* pada koneksi yang menggunakan socket stream (TCP). Method yang digunakan accept(), berikut contoh penggunaannya :

```
sock.accept() #Menerima koneksi
```

Statement di atas akan mengembalikan sebuah *tuple* (conn, address) dimana conn adalah objek socket baru yang berguna untuk mengirim dan menerima data dari koneksi, dan *address* merupakan alamat dari *client*.

Mengirim Data ke Koneksi (*Sending*)

Menerima koneksi tidak akan berarti tanpa digunakan untuk mengirim dan menerima data. Oleh karena itu digunakan method send(string) untuk socket stream (TCP) dan sendto(string,address) untuk socket datagram (UDP). Berikut ini penggunaannya untuk socket stream.

```
sock.send('ini pesan dari server')
```

Sedangkan untuk socket datagram digunakan :

```
sock.sendto('pesan dari server' , ('192.168.1.1' ,12345))
```

Menerima Data Dari Koneksi (*Receiving*)

Untuk menerima data yang dikirim dari *server* digunakan method recv(bufsize)untuk socket stream dan recvfrom(bufsize). Berikut ini penggunaannya untuk socket stream :

```
sock.recv(1024)          #Menerima data sebesar 1024 byte
```

Statement di atas akan mengembalikan data yang dikirimkan oleh *client*. Sedangkan untuk socket datagram :

```
sock.recvfrom(1024) #Menerima data sebesar 1024byte
```

Statement di atas akan mengembalikan dua buah field yaitu *data, address*.

Menutup Koneksi (*Closing*)

Untuk menutup koneksi yang telah dibuat digunakan method close(s). Berikut penggunaanya :

```
sock.close() #Menutup koneksi
```

COMPUTER NETWORK AND COMMUNICATIONS

PEMROGRAMAN SOCKET

Socket adalah mekanisme komunikasi yang memungkinkan terjadinya pertukaran data antar program atau proses baik dalam satu mesin maupun antar mesin. Gaya pemrograman socket sendiri berawal dari sistem Unix BSD yang terkenal dengan kepeloporannya pada bidang penanganan jaringan, sehingga sering disebut BSD Socket. Socket pertama kali diperkenalkan di sistem Unix BSD versi 4.2 tahun 1983 sebagai kelanjutan dari implementasi protokol TCP/IP yang muncul pertama kali pada sistem Unix BSD 4.1 pada akhir 1981. Hampir setiap variant Unix dan Linux mengadopsi BSD Socket. Pada lingkungan Unix, socket memberikan keleluasaan pemrograman gaya Unix yang terkenal dengan ideologinya, *Semua di Unix/Linux adalah file*. Komunikasi antar program dapat berlangsung lewat penggunaan deskriptor file standar Unix dengan bantuan socket.

Keunggulan dari penggunaan socket adalah anda dapat melakukan komunikasi antar proses/program melalui jaringan berbasis yang TCP/IP tentunya, bahkan dengan program lain yang berjalan pada platform non-unix seperti Microsoft Windows, sepanjang program tersebut berbicara dalam protokol transfer yang sama. Fasilitas-fasilitas yang disediakan oleh mesin unix seperti rlogin, ssh, ftp, dan lain-lain menggunakan socket sebagai sarana komunikasi mereka. Socket dibentuk dan digunakan dengan cara yang berbeda. Komunikasi socket terutama diciptakan untuk tujuan menjembatani komunikasi antara dua buah program yang dijalankan pada mesin yang berbeda. Jangan khawatir, ini tentu saja berarti dua program pada mesin yang sama dapat juga saling berkomunikasi. Kelebihan lain dari komunikasi socket adalah mampu menangani banyak klien sekaligus (multiple clients).

B. Jenis Socket

Ada dua golongan socket di Unix yang paling umum dipakai yaitu:

1. Socket Lokal atau AF_UNIX

Socket Lokal adalah socket yang melakukan komunikasi dengan perantaraan sebuah file yang biasanya diletakkan pada direktori /tmp atau /usr/tmp ataupun /var/tmp. Socket semacam ini digunakan umumnya terbatas untuk komunikasi antar aplikasi dalam satu mesin.

2. Socket Networking atau AF_INET

Socket Networking ditujukan untuk komunikasi antar aplikasi antar mesin dalam lingkungan jaringan TCP/IP. Identifikasi socket dilakukan dengan sebuah service identifier yaitu berupa nomor port TCP/IP yang dapat di sambung oleh client.

Socket Networking memiliki beberapa jenis, yang paling umum digunakan yaitu:

a. Socket Stream atau SOCK_STREAM

Socket Stream adalah socket komunikasi full-duplex berbasis aliran (stream) data. Pada model komunikasi Socket Stream, koneksi dua aplikasi harus dalam kondisi tersambung dengan benar untuk dapat bertukar data. Ini dapat dianalogikan seperti komunikasi telepon. Jika sambungan telepon di salah satu titik putus, maka komunikasi tidak dapat terjadi. Koneksi model seperti ini akan menjamin data dapat dipertukarkan

dengan baik, namun memiliki kelemahan dalam hal penggunaan jalur data yang relatif besar dan tidak boleh terputus.

b. Socket Datagram atau SOCK_DGRAM

Socket Datagram berkomunikasi dengan cara yang berbeda. Socket ini tidak membutuhkan koneksi yang tersambung dengan benar untuk mengirimkan dan menerima data. Model koneksi semacam ini tidak dapat menjamin data dapat dipertukarkan dengan baik, namun memiliki keunggulan dalam hal penggunaan jalur data yang minimal. Socket Datagram dapat dianalogikan dengan komunikasi yang terjadi pada kelas, misalnya pada saat guru melakukan broadcasting materi pelajaran untuk diterima oleh setiap murid. Tidak ada yang dapat menjamin materi pelajaran dapat diterima oleh semua murid dengan baik, kecuali diterapkan metoda rechecking. Rechecking ini dapat dilakukan baik oleh guru maupun murid. Guru bertanya untuk memastikan jawaban dari murid benar, atau murid bertanya untuk memastikan kebenaran materi yang diterimanya. Socket Datagram pun menggunakan metoda ini untuk menjamin pengiriman data dapat dilakukan dengan baik.

C. Sekilas Tentang Socket, TCP Dan UDP

1. Mengenal Socket

Pengertian socket adalah interface pada jaringan yang menjadi titik komunikasi antarmesin pada Internet Protocol, dan tentunya tanpa komunikasi ini, tidak akan ada pertukaran data dan informasi jaringan.

Socket terdiri dari elemen-elemen utama sebagai berikut:

1. Protokol.
2. Local IP.
3. Local Port.
4. Remote IP.
5. Remote Port.

Dalam komunikasi antara dua pihak, tentunya harus digunakan kesepakatan aturan dan format yang sama agar komunikasi dapat dimengerti. Seperti halnya dua orang yang menggunakan bahasa yang sama, maka bahasa di sini berfungsi sebagai protokol. Protokol yang digunakan dalam socket dapat menggunakan TCP ataupun UDP.

Contoh komunikasi sederhana adalah komunikasi antara komputer A dan komputer B. Baik komputer A maupun komputer B harus memiliki identitas unik, yang direpresentasikan oleh IP masing-masing. Komunikasi yang terjadi melalui port, sehingga baik komputer A maupun komputer B harus memiliki port yang dapat diakses satu sama lain.

2. TCP dan UDP

Pemrograman socket adalah cara untuk menggunakan komponen/API (Application Programming Interface) socket untuk membuat sebuah aplikasi.

Aplikasi socket umumnya terdiri dari dua kategori berdasarkan pengiriman datanya, yaitu:

a. Datagram socket (menggunakan UDP).

b. Stream socket (menggunakan TCP).

Terdapat perlakuan yang berbeda antara UDP dan TCP, walaupun sama-sama berfungsi sebagai protokol pertukaran data.

UDP tidak memerlukan proses koneksi terlebih dahulu untuk dapat mengirimkan data, paket-paket data yang dikirimkan UDP bisa jadi melalui rute yang berbeda-beda, sehingga hasil yang diterima bisa jadi tidak berurutan.

Contohnya jika aplikasi socket pengirim mengirimkan berturut-turut pesan 1, pesan 2, dan pesan 3, maka aplikasi socket penerima belum tentu mendapatkan pesan yang berurutan dimulai dari pesan 1, pesan 2, dan terakhir pesan 3. Bisa saja pesan 2 terlebih dulu diterima, menyusul pesan-pesan yang lain, atau berbagai kemungkinan lainnya. Bahkan, dapat terjadi pesan yang dikirimkan tidak sampai ke penerima karena kegagalan pengiriman paket data.

Tidak demikian halnya dengan stream socket yang menggunakan TCP. Jenis ini mengharuskan terjadinya koneksi terlebih dahulu, kemudian mengirimkan paket-paket data secara berurutan, penerima juga dijamin akan menerima data dengan urutan yang benar, dimulai dari data pertama yang dikirimkan hingga data terakhir. TCP dapat menangani data yang hilang, rusak, terpecah, ataupun terduplikasi.

Dari sekilas perbedaan ini, kita dapat menarik kesimpulan bahwa aplikasi socket yang menggunakan TCP memerlukan pertukaran data dua arah yang valid. Sedangkan, aplikasi socket yang menggunakan UDP lebih memprioritaskan pada pengumpulan data.

Karena itu aplikasi socket dengan TCP sering diterapkan untuk aplikasi chat, transfer file, ataupun transaksi-transaksi penting. Sedangkan aplikasi socket dengan UDP cocok diterapkan untuk aplikasi monitoring jaringan, game online, dan aplikasi-aplikasi broadcast.

D. Port dan Winsock

1. Port

Salah satu elemen penting yang digunakan dalam aplikasi socket adalah port. Port merupakan sebuah koneksi data virtual yang digunakan aplikasi untuk bertukar data secara langsung.

Terdapat banyak port di dalam sebuah sistem komputer dengan fungsinya masing-masing. Sebagai contoh, dalam mengirim e-mail digunakan service SMTP yang umumnya menggunakan port 25. Sementara service POP3 untuk menerima e-mail menggunakan port 110, port 80 digunakan untuk HTTP, port 443 digunakan untuk HTTPS, dan seterusnya.

Nomor-nomor port dikategorikan dalam tiga jenis sebagai berikut:

a. Well-known ports.

Merupakan port yang telah digunakan secara internal oleh sistem Windows, misalnya port untuk koneksi Internet, service FTP, dan seterusnya. Port yang telah digunakan ini adalah port 0 sampai dengan port 1023.

b. Registered ports.

Port ini dapat digunakan dalam aplikasi Anda, range-nya adalah port 1024 hingga port 49151, cukup banyak port yang tersedia yang bebas Anda pilih sehingga Anda tidak perlu kuatir kekurangan port untuk aplikasi Anda.

c. Dynamic/Private ports.

Dari port 49152 sampai dengan port 65535.

2. Winsock

Untuk pemrograman aplikasi socket berbasis Windows, maka komponen API yang sering digunakan adalah Winsock (Win-dows Socket API) yang mendukung interface standar TCP/IP, yang merupakan protokol jaringan paling popular saat ini (contoh protokol jaringan yang lain adalah NetBIOS, IPX dari Novell, AppleTalk dari Apple, dan sebagainya).

Pengertian TCP/IP (TCP over IP) mungkin dapat menjadi sedikit rancu jika diartikan TCP/IP hanya mengizinkan pengiriman TCP (dan tidak UDP), padahal seperti yang telah kita bahas, pengiriman socket dapat melalui TCP maupun UDP.

Pengertian TCP/IP di sini sebenarnya digunakan untuk menunjukkan teknologi jaringan/Internet, termasuk di dalamnya adalah UDP. Jika Anda menggunakan UDP, dapat juga disebut sebagai UDP/IP (UDP over IP), tetapi umumnya istilah ini jarang digunakan dan istilah TCP/IP telah mencakup, baik TCP maupun UDP.

Pada bahasa pemrograman visual seperti Visual Basic/Delphi, Anda dapat menggunakan control Winsock yang telah disediakan untuk mengembangkan aplikasi socket.

Walaupun kita akan mencontohkan aplikasi socket dalam environment Windows, Anda tidak perlu khawatir jika aplikasi socket yang menggunakan Winsock tidak dapat berkomunikasi dengan aplikasi socket berbasis Unix/Linux, karena komunikasi tetap dapat terjadi selama aplikasi tersebut menggunakan protokol jaringan yang sama.

Bagi Anda yang terpaksa hanya menggunakan satu komputer, dapat memanfaatkan alamat localhost atau 127.0.0.1 yang mengizinkan dua aplikasi berjalan pada satu mesin komputer dan berkomunikasi satu sama lain.

3. Tools Tambahan

Aplikasi socket merupakan aplikasi jaringan dan jika Anda mendalami seluk-beluk jaringan, tentu akan familiar dengan tools tambahan yang umumnya digunakan dalam jaringan. Tools ini kemungkinan dapat berguna untuk diimplementasikan ke dalam aplikasi socket Anda.

Tools yang dimaksud, antara lain:

a. Ping.

Ping digunakan untuk memeriksa keberadaan remote host dengan jalan mengirimkan sinyal kepada remote host. Keberadaan remote host dapat ditentukan dengan melihat response yang diterima. Ping juga dapat digunakan untuk mengukur kecepatan transfer data. Salah satu contoh penggunaan ping dalam aplikasi socket adalah memeriksa server yang tersedia sebelum mengirimkan data (dengan asumsi tersedia lebih dari 1 server).

b. Telnet.

Telnet merupakan singkatan dari TELecommunication NET-work. Umumnya istilah telnet saat ini merujuk pada aplikasi telnet client yang tersedia pada kebanyakan operating sys-tem. Telnet mengizinkan Anda mengakses remote host dan menggunakan service-nya. Sebagai contoh, Anda dapat mengirimkan e-mail melalui telnet yang menggunakan port 25 (service SMTP) pada remote host tertentu. Jika Anda telah masuk ke dalam environment telnet, command line yang digunakan adalah command berbasis Unix/Linux. Aplikasi socket dapat dimodifikasi bekerja seperti telnet dengan mengakses remote host dan port tertentu. Di dalam aplikasi socket, Anda dapat mengambil dan mengolah response yang didapat dari remote host.

c. Netstat.

Netstat menampilkan status jaringan yang terjadi. Dapat menampilkan port yang sedang terkoneksi, atau dalam kondisi menunggu/listening, juga menampilkan protokol yang digunakan, apakah TCP atau UDP. Dengan Netstat, Anda dapat mengetahui koneksi jaringan yang terjadi, hal ini dapat dimanfaatkan di dalam aplikasi socket, misalnya untuk melihat port yang sedang aktif dan digunakan.

Ada kalanya Anda perlu menjalankan tools jaringan yang telah disebutkan di atas melalui aplikasi Anda. Untuk keperluan ini, Anda dapat menggunakan shell command yang disediakan oleh bahasa pemrograman yang Anda gunakan.

Misalnya pada Visual Basic, dapat digunakan perintah Shell diikuti parameter yang diperlukan.

Jika ingin mengolah response yang dihasilkan oleh tools tertentu, Anda dapat menuliskan hasilnya pada sebuah file teks, contohnya jika Anda menjalankan perintah netstat –an > hasil.txt pada Command Prompt Windows, maka informasi mengenai koneksi yang aktif akan tersimpan dalam file hasil.txt, di mana Anda dapat mengolah file hasil.txt tersebut lebih lanjut di dalam aplikasi Anda.

E. Socket Programming

Socket adalah sebuah cara untuk berkomunikasi dengan program atau node lain menggunakan file deskriptor. Di UNIX (dimana socket diciptakan) kita sering mendengar slogan: “everything is a file”, jadi untuk berkomunikasi dengan program atau node lain semudah kita membaca dan menulis file deskriptor. Antarmuka socket dan file adalah mirip, jika pada file kita membukanya dengan open() sedangkan pada socket kita manggunakan socket(). Pada file deskriptor yang menjadi tujuan adalah sebuah file, sedangkan pada socket adalah komputer atau node lain. Intinya ketika kita telah terhubung dengan socket(), maka antarmukanya sama saja dengan sebuah file. Sebuah abstraksi perangkat lunak yang digunakan sebagai suatu “terminal” dari suatu hubungan antara dua mesin atau proses yang saling berinterkoneksi

Penggunaan socket programming memungkinkan adanya komunikasi antara client dan server. Salah satu contoh sederhana penggunaan socket programming adalah pembuatan program untuk chatting. Program tersebut sebenarnya merupakan bentuk aplikasi berupa komunikasi antara client dan server. Ketika seorang user (client) melakukan koneksi ke chat server, program akan membuka koneksi ke port yang diberikan, sehingga server perlu membuka socket pada port tersebut dan “mendengarkan” koneksi yang datang. Socket sendiri merupakan gabungan antara host-adress dan port adress. Dalam hal ini socket digunakan untuk komunikasi antara client dan server.

Socket merupakan fasilitas IPC (Inter Proses Communication) untuk aplikasi jaringan. Agar suatu socket dapat berkomunikasi dengan socket lainnya, maka socket butuh diberi suatu alamat unik sebagai identifikasi. Alamat socket terdiri atas Alamat IP dan Nomer Port. Contoh alamat socket adalah **192.168.29.30**: **3000**, dimana nomer 3000 adalah nomer portnya. Alamat IP dapat menggunakan alamat Jaringan Lokal (LAN) maupun alamat internet. Jadi socket dapat digunakan untuk IPC pada LAN maupun Internet.

F. Model Aplikasi Client Server

Untuk membuat aplikasi socket yang sederhana diperlukan dua aplikasi. Yaitu, pertama adalah aplikasi server yang akan menerima data, sedangkan aplikasi kedua adalah aplikasi client yang mengirimkan data pada server. Baik aplikasi server dan aplikasi client mendefinisikan port yang sama sebagai jalur komunikasi.

Obyek socket pada sisi client dan server berbeda sedikit. Pada sisi aplikasi server, suatu socket server dibentuk dan melakukan operasi listen/menunggu. Operasi ini pada intinya menunggu permintaan koneksi dari sisi client. Sedangkan pada sisi client, dibentuk suatu socket biasa.

Pada saat socket client, informasi alamat socket server dilewatkan sebagai argumen dan socket client akan otomatis mencoba meminta koneksi ke socket server. Pada saat permintaan koneksi client sampai pada server, maka server akan membuat suatu socket biasa. Socket ini yang nantinya akan berkomunikasi dengan socket pada sisi client. Setelah itu socket server dapat kembali melakukan listen untuk menunggu permintaan koneksi dari client lainnya. Langkah ini umumnya hanya dilakukan jika aplikasi server mengimplementasikan multithreading.

Setelah tercipta koneksi antara client dan server, maka keduanya dapat saling bertukar pesan. Salah satu atau keduanya kemudian dapat mengakhiri komunikasi dengan menutup socket.

Untuk protokol UDP, perbedaanya adalah socket di sisi server sama dengan socket di sisi client, dan tidak ada operasi listen pada sisi server. Kemudian saat paket data dikirimkan, alamat socket penerima harus disertakan sebagai argumen.

Program Aplikasi Client Server bisa dibuat dengan menggunakan Visual Basic 6, NET, Delphi, dan lain sebagainya. Sebagai contoh yaitu Aplikasi Client Server sederhana dengan Delphi dimana tugas aplikasi server cukup sederhana, yaitu hanya siap sedia menerima data yang masuk pada sebuah port.

a. Dasar Pemrograman Socket Menggunakan Delphi

Apa yang harus dipersiapkan oleh seorang programmer dalam membuat program mengunakan socket ini? Yaitu:

1. Komponen Pengirim dan Penerima

   Pertama kali anda harus tahu terlebih dahulu mekanisme dari socket ini. Sebagai contoh logikanya, jika seseorang mengirim suatu barang pasti ada penerimanya. Begitu juga di socket, jika mengirimkan data harus ada yang menerima. Kalau anda sudah mengerti apa yang sudah dijelaskan tadi, maka yang mengirim adalah komponen **TClientSocket** dan yang menerima adalah **TServerSocket**.
2. Tujuan Pengiriman

   Selanjutnya jika seseorang akan mengirim suatu surat maka harus ada alamat tujuan pengiriman. Jika di socket ini disebut sebagai host atau IP Address dan Port. Jadi pada saat **TClientSocket** akan mengirim pesan harus diisi terlebih dahulu IP Address dan port tadi.
3. Status Penerimaan

Setelah **TClientSocket** mengirim sinyal untuk terhubung ke **TServerSocket** maka perlu adanya status bahwa keduanya telah terhubung. Atau bahkan jika diantara keduanya sudah tidak terhubung lagi. Ini sangat penting mengingat pada saat akan mengirim data, harus mengetahui terlebih dahulu status keterhubungan ini.

b. Pengolahan Data

Komunikasi data antara server dan client di atas merupakan bentuk komunikasi satu arah sederhana. Data yang dikirimkan dari client pun merupakan data mentah yang tidak memerlukan pengolahan data lebih lanjut.

Anda dapat membuat sendiri function dan rutin untuk mengolah data yang dikirim dan diterima sesuai dengan kebutuhan aplikasi, karena data yang dikirimkan antarmesin bisa jadi sangat bervariasi. Misalnya saja aplikasi server/client Anda memerlukan pertukaran data identitas mesin, tanggal, jam, header pesan, isi pesan, dan lain sebagainya. Anda dapat mengirimkannya dalam format tertentu, misalnya bentuk string dengan karakter pemisah untuk membedakan masing-masing field.

Dalam komunikasi data di dalam jaringan, Anda perlu mempertimbangkan besarnya data yang lalu-lalang pada jaringan, baik dengan menggunakan TCP maupun UDP. Keduanya harus dipersiapkan untuk mampu menangani data yang besar jika memang pengguna aplikasi socket Anda sangat luas. Pastinya tidak terdapat masalah yang berarti jika Anda mencobanya dengan dua atau beberapa komputer dalam sebuah jaringan lokal, tetapi coba bayangkan seberapa besar total data yang harus dikirim dan diterima pada sebuah aplikasi, misalnya game online. Pada contoh game online, sebuah server harus dipersiapkan untuk mampu melayani sedemikian banyak client, dan jaringan yang digunakan bukan lagi jaringan lokal, tetapi sudah merupakan jaringan Internet, di mana siapapun dapat menggunakan aplikasi Anda selama ia memiliki koneksi Internet.

Mungkin Anda bertanya, jika data yang keluar-masuk memerlukan pengolahan lebih lanjut, mengapa tidak digunakan database, sehingga Anda tidak perlu pusing membuat rutin atau modul untuk mengolah data yang dikirim/diterima melalui komunikasi socket? Pada umumnya, aplikasi socket client/server memang menggunakan database pada sisi server, tetapi jika aplikasi socket mengharuskan sisi client menggunakan database tertentu, maka akan membatasi penggunaan aplikasi itu sendiri.

Selain itu, kegunaan komunikasi socket adalah agar dapat berjalan lintas platform. Tidak peduli operating system apa yang digunakan pengguna aplikasi, komunikasi socket tetap berjalan selama digunakan protokol yang sama. Sebenarnya jika Anda melihat software database seperti SQL Server, intinya juga merupakan aplikasi socket, di mana menggunakan port tertentu sebagai jalur komunikasi, tetapi software tersebut telah dikemas menjadi produk database yang spesifik.

G. Trouble Shooting Jaringan

Kadang sebagai admin jaringan, kita kerap kali menemukan masalah-masalah, untuk itu diperlukan ilmu yang mempelajari tentang trouble shooting jaringan. Di sini akan dijabarkan beberapa trouble shooting dan penanganannya :

1. Antar komputer 1 dan lainnya tidak bisa berkomunikasi :
   a. Cek apakah pembagian IP pada class yang sama
   b. Cek apakah komputer dalam 1 workgroup yang sama
   c. Restart Komputer
2. Komputer sudah diset dengan ip calss yang benar dan dalam 1 workgroup tetapi tidak dapat berkomunikasi

a. Cek apakah LAN card rusak, yaitu dengan menjalankan perintah ping 127.0.0.1 atau ping localhost.
b. Cek Apakah Kabel LAN sudah benar pembuatannya, untuk komputer yang melewati Hub tidak perlu crossing pada kabelnya, tetapi untuk komputer yang tidak melewati Hub harus memakai metode crossing 1-3, 2-6.

SELESAI

Network Socket Programming

Python menyediakan dua tingkat akses ke layanan jaringan. Pada tingkat rendah, Anda dapat mengakses dukungan soket dasar dalam sistem operasi yang mendasarinya, yang memungkinkan Anda untuk mengimplementasikan klien dan server untuk kedua protokol berorientasi koneksi dan tanpa sambungan.

Python juga memiliki pustaka yang menyediakan akses tingkat lebih tinggi ke protokol jaringan tingkat aplikasi tertentu, seperti FTP, HTTP, dan seterusnya.

Bab ini memberi Anda pemahaman tentang konsep paling terkenal dalam Networking - Socket Programming.

Apa itu Socket?

Soket adalah titik akhir dari saluran komunikasi dua arah. Soket dapat berkomunikasi dalam suatu proses, antara proses pada mesin yang sama, atau antara proses di berbagai benua.

Soket dapat diimplementasikan melalui sejumlah jenis saluran yang berbeda: soket domain Unix, TCP, UDP, dan sebagainya. Pustaka socket menyediakan kelas khusus untuk menangani transportasi umum serta antarmuka umum untuk menangani sisanya.

Modul Socket

Untuk membuat soket, Anda harus menggunakan fungsi socket.socket () yang tersedia dalam modul soket, yang memiliki sintaks umum

s = socket.socket (socket_family, socket_type, protocol=0)

Server Socket Method

Method	Penjelasan
s.bind()	This method binds address (hostname, port number pair) to socket.
s.listen()	This method sets up and start TCP listener.
s.accept()	This passively accept TCP client connection, waiting until connection arrives (blocking).

Client Socket Method

Method	Penjelasan
s.connect()	This method actively initiates TCP server connection.

General Method Socket

Method	Penjelasan
s.recv()	This method receives TCP message
s.send()	This method transmits TCP message
s.recvfrom()	This method receives UDP message
s.sendto()	This method transmits UDP message
s.close()	This method closes socket
socket.gethostname()	Returns the hostname.

```
#!/usr/bin/python          # This is server.py file

import socket              # Import socket module

s = socket.socket()        # Create a socket object
host = socket.gethostname() # Get local machine name
port = 12345               # Reserve a port for your service.
s.bind((host, port))       # Bind to the port

s.listen(5)                # Now wait for client connection.
while True:
  c, addr = s.accept()     # Establish connection with client.
  print 'Got connection from', addr
  c.send('Thank you for connecting')
  c.close()                # Close the connection
```

Server Sederhana

Untuk menulis server Internet, kami menggunakan fungsi soket yang tersedia di modul soket untuk membuat objek soket. Objek soket kemudian digunakan untuk memanggil fungsi lain untuk menyiapkan server soket.

Sekarang sebut bind(hostname,port) berfungsi untuk menentukan port untuk layanan Anda pada host yang diberikan.

Selanjutnya, panggil metode penerimaan objek yang dikembalikan. Metode ini menunggu sampai klien terhubung ke port yang Anda tentukan, dan kemudian mengembalikan objek koneksi yang mewakili koneksi ke klien itu.

Client Sederhana

Mari kita menulis program klien yang sangat sederhana yang membuka koneksi ke port yang diberikan 12345 dan host yang diberikan. Ini sangat sederhana untuk membuat klien soket menggunakan fungsi modul soket Python.

Socket.connect (hosname, port) membuka koneksi TCP ke hostname pada port. Setelah Anda memiliki soket terbuka, Anda dapat membaca darinya seperti objek IO apa pun. Setelah selesai, jangan lupa untuk menutupnya, karena Anda akan menutup file.

Kode berikut adalah klien yang sangat sederhana yang terhubung ke host dan port yang diberikan, membaca data yang tersedia dari soket, dan kemudian keluar

```
#!/usr/bin/python          # This is client.py file


import socket              # Import socket module


s = socket.socket()        # Create a socket object

host = socket.gethostname() # Get local machine name

port = 12345               # Reserve a port for your service.


s.connect((host, port))

print s.recv(1024)

s.close                    # Close the socket when done
```

Sekarang jalankan server.py ini di latar belakang dan kemudian jalankan di atas client.py untuk melihat hasilnya.

Jalankan server.

python server.py &

Setelah server berjalan lanjutkan

Jalankan client:

python client.py

Hasilnya akan seperti ini : Got connection from ('127.0.0.1', 48437) Thank you for connecting

Modul Internet pada Python

Berikut tabel daftar beberapa modul penting dalam pemrograman Jaringan / Internet Python.

Protocol	Common function	Port No	Python module
HTTP	Web pages	80	httplib, urllib, xmlrpclib
NNTP	Usenet news	119	nntplib
FTP	Transfer file	20	ftplib, urllib
SMTP	Mengirim email	25	smtplib
POP3	Fetching email	110	poplib
IMAP4	Fetching email	143	imaplib
Telnet	Command lines	23	telnetlib
Gopher	Document transfers	70	gopherlib, urllib

Socket pada Pemerograman Phyton

Mirza Eka Putra

202420034

Definisi socket pada Phyton

Soket adalah titik akhir dari saluran komunikasi dua arah. Soket dapat berkomunikasi dalam suatu proses, antara proses pada mesin yang sama, atau antara proses di berbagai benua.

Soket dapat diimplementasikan melalui sejumlah jenis saluran yang berbeda: soket domain Unix, TCP, UDP, dan sebagainya. Pustaka socket menyediakan kelas khusus untuk menangani transportasi umum serta antarmuka umum untuk menangani sisanya.

Modul Socket

Untuk membuat soket, Anda harus menggunakan fungsi socket.socket () yang tersedia dalam modul soket, yang memiliki sintaks umum

```
s = socket.socket (socket_family, socket_type, protocol=0)
```

Server Socket Method

Method	Penjelasan
s.bind()	This method binds address (hostname, port number pair) to socket.
s.listen()	This method sets up and start TCP listener.
s.accept()	This passively accept TCP client connection, waiting until connection arrives (blocking).

Client Socket Method

Method	Penjelasan
s.connect()	This method actively initiates TCP server connection.

General Method Socket

Method	Penjelasan
s.recv()	This method receives TCP message
s.send()	This method transmits TCP message
s.recvfrom()	This method receives UDP message
s.sendto()	This method transmits UDP message
s.close()	This method closes socket
socket.gethostname()	Returns the hostname.

```
#!/usr/bin/python          # This is server.py file

import socket              # Import socket module

s = socket.socket()        # Create a socket object
host = socket.gethostname() # Get local machine name
port = 12345               # Reserve a port for your service.
s.bind((host, port))       # Bind to the port
```

```
s.listen(5)                # Now wait for client connection.
while True:
   c, addr = s.accept()     # Establish connection with client.
   print 'Got connection from', addr
   c.send('Thank you for connecting')
   c.close()                # Close the connection
```

Server Sederhana

Untuk menulis server Internet, kami menggunakan fungsi soket yang tersedia di modul soket untuk membuat objek soket. Objek soket kemudian digunakan untuk memanggil fungsi lain untuk menyiapkan server soket.

Sekarang sebut `bind(hostname,port)` berfungsi untuk menentukan port untuk layanan Anda pada host yang diberikan.

Selanjutnya, panggil metode penerimaan objek yang dikembalikan. Metode ini menunggu sampai klien terhubung ke port yang Anda tentukan, dan kemudian mengembalikan objek koneksi yang mewakili koneksi ke klien itu.

Client Sederhana

Mari kita menulis program klien yang sangat sederhana yang membuka koneksi ke port yang diberikan 12345 dan host yang diberikan. Ini sangat sederhana untuk membuat klien soket menggunakan fungsi modul soket Python.

Socket.connect (hosname, port) membuka koneksi TCP ke hostname pada port. Setelah Anda memiliki soket terbuka, Anda dapat membaca darinya seperti objek IO apa pun. Setelah selesai, jangan lupa untuk menutupnya, karena Anda akan menutup file.

Kode berikut adalah klien yang sangat sederhana yang terhubung ke host dan port yang diberikan, membaca data yang tersedia dari soket, dan kemudian keluar

Jelaskan apa yang dimaksud dengan socket pada python network programming ?

Jawab :

Socket adalah titik akhir dari saluran komunikasi dua arah. Socket dapat berkomunikasi dalam suatu proses, antara proses pada mesin yang sama, atau antara proses di berbagai benua.