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

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

TUGAS COMPUTER NETWORK AND COMMUNICATION

TUGAS Pyton Programming

Jelaskan apa yang dimaksud dengan socket pada python network programming ?

NAMA : Nanda S. Prawira
NIM : 192420056

MAGISTER TEKNIK INFORMATIKA

UNIVERITAS BINA DARMA

PALEMBANG

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.

Networking Python

Networking PythonPython 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.

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 fileimport socket          # Import socket modules = socket.socket()       # Create a socket objecthost = socket.gethostname() # Get local machine nameport = 12345              # Reserve a port for your service.s.bind((host, port))       # Bind to the ports.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 fileimport socket          # Import socket modules = socket.socket()       # Create a socket objecthost = socket.gethostname() # Get local machine nameport = 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

Nama : Rahmi
NIM : 192420046

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

   Jawaban :

   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.

SOCKET PADA PHYTON NETWORK PROGRAMMING

Rani Okta Felani

192420048

PROGRAM PASCASARJANA

MEGISTER TEHNIK INFORMATIKA

UNIVERSITAS BINA DARMA PALEMBANG

TAHUN 2020

SOCKET PADA PHYTON NETWORK PROGRAMMING

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.

Networking Python

Networking PythonPython 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.

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 fileimport socket # Import socket modules = socket.socket() # Create a socket objecthost = socket.gethostname() # Get local machine nameport = 12345 # Reserve a port for your service.s.bind((host, port)) # Bind to the ports.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 fileimport socket          # Import socket modules = socket.socket()          # Create a socket objecthost = socket.gethostname() # Get local machine nameport = 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

Nama : Suwani
Nim : 192420049
MK : COMPUTER NETWORK AND COMMUNICATION

SOCKET PADA PHYTON NETWORK PROGRAMMING

Socket 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).

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.

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.

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.

Networking Python

Networking 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.

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 berbedasoket 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 fileimportsocket# Import socket modules=socket.socket()# Create a socket objecthost=socket.gethostname()# Get local machine nameport=12345# Reserve a port for your service.s.bind((host,port))# Bind to the ports.listen(5)# Now wait for client connection.whileTrue:

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*importsocket*# Import socket modules*=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

Protocol	Common function	Port No	Python module
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

TUGAS PYTHON PROGRAMMING

NAMA : **THEO VHALDINO**

NIM : **192420058**

ANGKATAN : **MTI22**

MAGISTER TEKNIK INFORMATIKA

UNIVERITAS BINA DARMA

PALEMBANG

Soal :

Jelaskan apa yang dimaksud dengan socket padaPython Network Programming ?

Penyelesaian :

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

Untuk membuat socket, harus menggunakan fungsi socket.socket() yang tersedia dalam modul socket, yang memiliki sintaks umum. fungsi socket.socket ini buat membentuk socket, nilai kembalinya berupa deskripsi socket (objek) yang nantinya kita bisa menggunakan method-method lain dari objek ini.

TUGAS UTS

TUGAS PYTHON PROGRAMMING

NAMA : YAYAN CANDRA SUBIDIN
NIM : 192420054

MAGISTER TEKNIK INFORMATIKA

UNIVERITAS BINA DARMA

PALEMBANG

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

   Jawaban :

   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.

TUGAS

TUGAS NETWORK ACCESS

NAMA : AL ADRI NOFA GUSANDI
NIM : 192420053

MAGISTER TEKNIK INFORMATIKA

UNIVERITAS BINA DARMA

PALEMBANG

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

Jawaban :

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.

Nama : Arpa Pauziah
NIM : 192420055
Ruangan : U 705
Mata Kuliah : Computer Network And Communication

Pertanyaan : Jelaskan apa yang dimaksud dengan socket pada python network programing?

Penjelasan:

Network Socket pada python

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.

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.

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.

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 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()
```

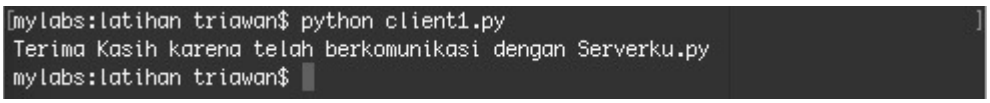
Nama file : clientku.py

```
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:

Gambar Output Serverku.py



Gambar Output Clientku.py

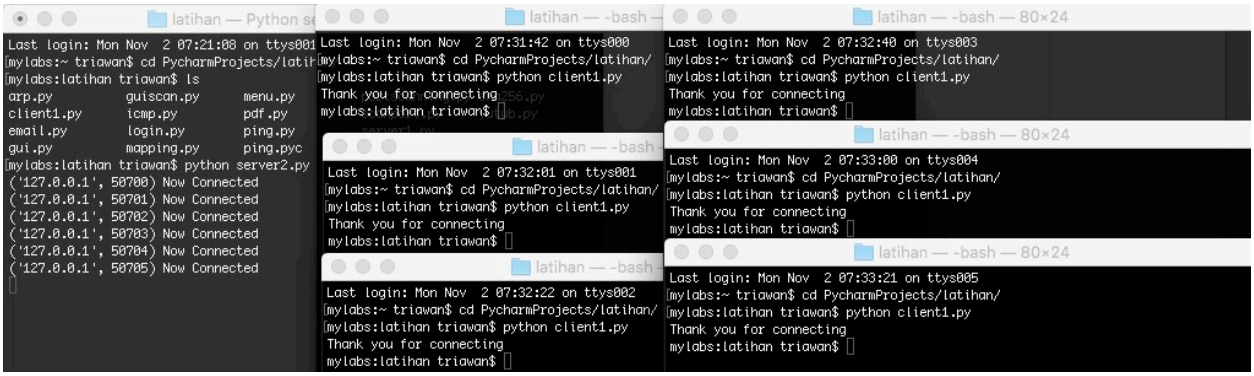
Metode konektivitas client server diatas hanya untuk menangani satu permintaan yang dikirim oleh client. Apabila menginginkan server socket menangani lebih dari satu service, maka tinggal tambahkan looping setelah statement listen.

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:



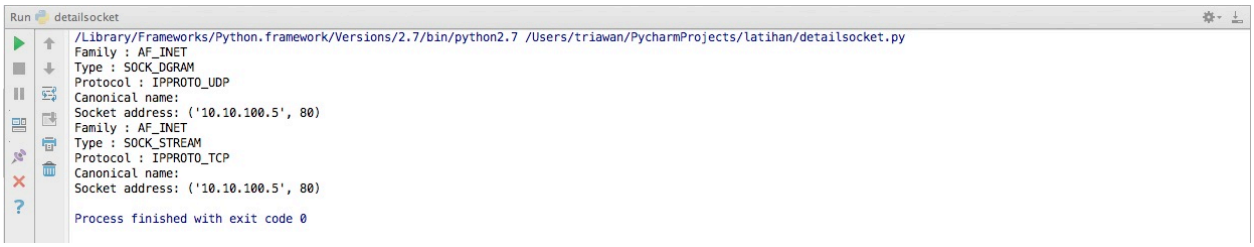
Gambar Output serverku2.py dan clientku.py ketika dijalankan

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 :



Gambar Detail Socket

SOCKET PYTHON NETWORK PROGRAMMING

ELPINA SARI

ENTERPRISE IT INFRASTRUCTURE

192420050

PROGRAM STUDI TEKNIK INFORMATIKA – S2

PROGRAM PASCASARJANA

UNIVERSITAS BINA DARMA

PALEMBANG

2020

Jelaskan apa yang dimaksud dengan socket pada python network programming ?

Jawaban :

1. Penggunaan Dasar Socket

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.

2. Networking Python

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.

Networking 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.

- **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

Method	Penjelasan
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.

- **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 Konstanta Keluarga (Family) Protokol

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

Tabel 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 oleh *server*. Ini dilakukan dengan memanggil metode socket connect(address), dimana address adalah sebuah *tuple*

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

  (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 fil

  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
  ```

- **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
```

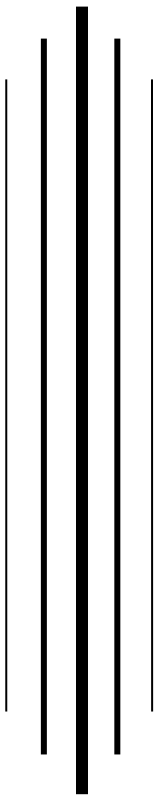
- **Modul Internet pada Python**

  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

TUGAS COMPUTER NETWORK AND COMMUNICATION

KELAS MTI 22A



DOSEN PENGASUH

Dr. Edi Surya Negara, S.Kom, M.Kom

DISUSUN OLEH:

FADEL MUHAMMAD MADJID

192420052

PROGRAM PASCA SARJANA MAGISTER TEKNIK INFORMATIKA

UNIVERSITAS BINA DARMA

SOCKET PADA PYTHON PROGRAMMING

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.

Networking Python

Networking 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.

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 fileimport socket          # Import socket modules = socket.socket()          # Create a socket objecthost = socket.gethostname() # Get local machine nameport = 12345          # Reserve a port for your service.s.bind((host, port))       # Bind to the ports.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

DAFTAR PUSTAKA

https://realpython.com/python-sockets/

https://belajarpython.com/tutorial/networking-python

Nama : Isti Maátun Nasichah

NPM : 192420051

Program : Magister Teknik Informatika

TUGAS

Jelaskan apa yang dimaksud dengan socket pada python network programming ?

Jawab :

Pengenalan Phyton

Python merupakan bahasa pemrograman yang populer khususnya pada bidang keamanan komputer. Pada modul ini, beberapa eksperimen dalam pembuatan program untuk mendukung proses keamanan komputer, ditulis menggunakan python versi 2. Apabila tidak ingin rumit, maka pakailah sistem operasi linux varian terbaru, misalnya ubuntu, kali linux, dan lain sebagainya. Bahasa pemrograman python beserta modul-modulnya sudah terinstall otomatis di sistem operasi linux

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.

Pengertian Socket

Socket 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.

Socket 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.

Pengenalan 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.

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).

Cara Kerja Server Socket Method

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.

Client Socket Method

Method yang terdapat untuk fungsi di socket client adalah:

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

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

General 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 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.

Menjalankan Server

python server.py &

Setelah server berjalan lanjutkan

Menjalankan 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

TUGAS PYTON PROGRAMMING

D
I
S
U
S
U
N

Oleh

NAMA : M. Iqbal Rivana

NIM : 192420057

MAGISTER TEKNIK INFORMATIKA

UNIVERITAS BINA DARMA

PALEMBANG

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

JAWABAN :

Python memainkan peran penting dalam pemrograman jaringan. Pustaka standar Python memiliki dukungan penuh untuk protokol jaringan, encoding, dan decoding data dan konsep jaringan lainnya, dan lebih sederhana untuk menulis program jaringan dengan Python daripada C ++.

Ada dua tingkat akses layanan jaringan dengan Python. Ini adalah:

1. Akses Tingkat Rendah
2. Akses Tingkat Tinggi

Dalam kasus pertama, pemrogram dapat menggunakan dan mengakses dukungan soket dasar untuk sistem operasi menggunakan pustaka Python, dan pemrogram dapat mengimplementasikan protokol tanpa koneksi dan berorientasi koneksi untuk pemrograman. Protokol jaringan tingkat aplikasi juga dapat diakses menggunakan akses tingkat tinggi yang disediakan oleh pustaka Python. Protokol ini adalah HTTP, FTP, dll.

Apa Sockets?

Soket adalah titik akhir dalam aliran komunikasi antara dua program atau saluran komunikasi yang beroperasi melalui jaringan. Mereka dibuat menggunakan satu set permintaan pemrograman yang disebut socket API (Application Programming Interface). Pustaka soket Python menawarkan kelas untuk menangani angkutan umum sebagai antarmuka generik.

Soket dan API soket digunakan untuk mengirim pesan melalui jaringan. Mereka menyediakan bentuk komunikasi antar-proses (IPC). Jaringan dapat berupa jaringan lokal yang logis ke komputer, atau yang secara fisik terhubung ke jaringan eksternal, dengan sambungannya sendiri ke jaringan lain. Contoh yang jelas adalah Internet, yang Anda sambungkan melalui ISP Anda.

Soket dapat diimplementasikan pada sejumlah jenis saluran yang berbeda: soket domain Unix, TCP, UDP, dan sebagainya. Pustaka soket menyediakan kelas khusus untuk menangani pengangkutan umum serta antarmuka umum untuk menangani sisanya. Soket menggunakan protokol untuk menentukan jenis koneksi untuk komunikasi port-ke-port antara mesin klien dan server. Protokol digunakan untuk:

1. Domain Name Servers (DNS)
2. IP addressing
3. E-mail
4. FTP (File Transfer Protocol)

Python memiliki metode soket yang memungkinkan pemrogram mengatur berbagai jenis soket secara virtual. Sintaks untuk metode soket adalah:

- **listen()**: is used to establish and start TCP listener.
- **bind()**: is used to bind-address (host-name, port number) to the socket.
- **accept()**: is used to TCP client connection until the connection arrives.
- **connect()**: is used to initiate TCP server connection.
- **send()**: is used to send TCP messages.
- **recv()**: is used to receive TCP messages.
- **sendto()**: is used to send UDP messages
- **close()**: is used to close a socket.

Setelah Anda menentukan soket, Anda dapat menggunakan beberapa metode untuk mengelola koneksi. Beberapa metode soket server yang penting adalah:

Syntax:

```
g = socket.socket (socket_family, type_of_socket, protocol=value)
```

For example, if we want to establish a TCP socket, we can write the following code snippet:

Example:

```
# imports everything from 'socket'

from socket import *

# use socket.socket() - function

tcp1=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

Here's another example to establish a UDP socket. The code is:

```
udp1=socket.socket (socket.AF_INET, socket.SOCK_DGRAM)
```

https://www.w3schools.in/python-tutorial/network-programming/
https://www.tutorialspoint.com/python/python_networking.htm