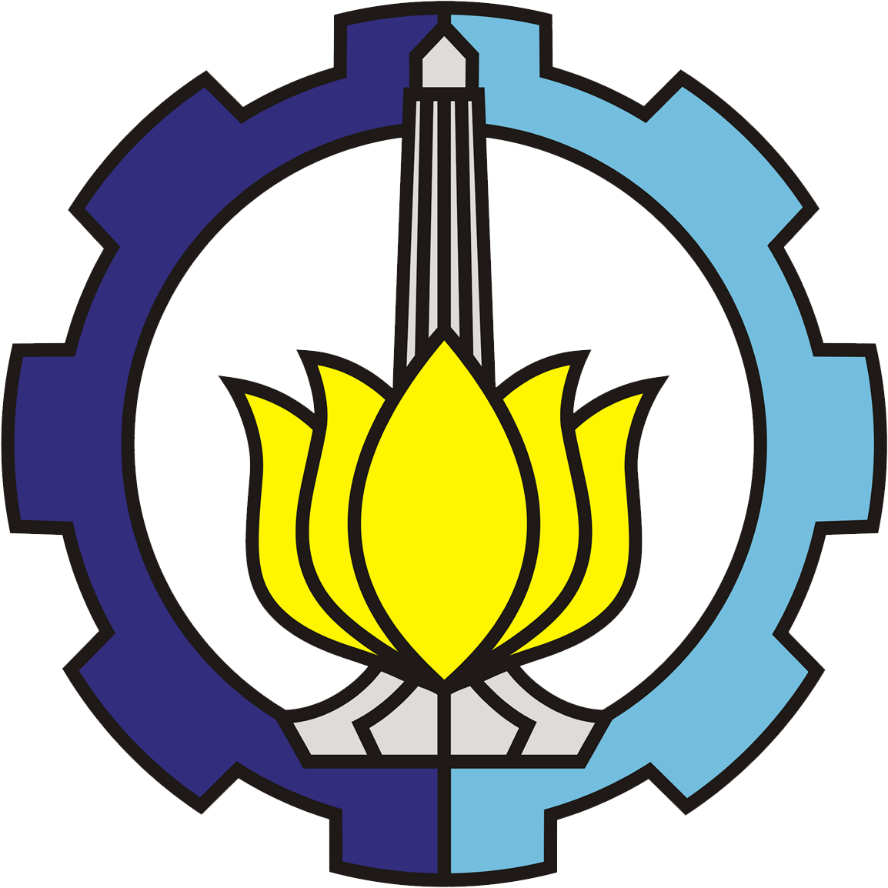
**LAPORAN TUGAS**

**APLIKASI CHATTING**



Oleh

**Kelompok 4 Pemrograman Jaringan E**

FARHAN RAMADHANA (5114100078)MUHAMMAD FAISHAL ILHAM (5114100076)KHARISMA MONIKA (5114100092)WILDAN LUTFI (5114100080)ANTONIUS KEVIN (5114100132)

A. LATAR BELAKANG

Salah satu mata kuliah semester 5 adalah Pemrograman Jaringan. Pada mata kuliah ini, penulis mendapatkan tugas untuk membuat server aplikasi chat dan membuat client dari server kelompok lain.

B. RANCANGAN SERVER

1. DESKRIPSI

Server ini digunakan untuk aplikasi yang mampu digunakan untuk chatting oleh beberapa user. Sistem ini menggunakan 1 server. Sistem ini memiliki database yang menyimpan user-user yang telah/pernah registrasi dalam bentuk .txt. Sistem ini akan mengembalikan nilai berupa kode-kode yang memiliki arti masing-masing. Setiap command dan return diakhiri dengan “\n”.

1. FITUR

2.1. REGISTRASI

Fitur registrasi merupakan fitur pertama kali yang harus dilakukan dalam aplikasi ini yaitu mendaftarkan diri sebagai anggota aplikasi chat server ini, ketika mendaftar maka user akan memiliki username dan password yang berguna untuk dapat masuk ke dalam aplikasi ini.

2.1.1. COMMAND

REGIST<spasi>USERNAME<spasi>PASSWORD

2.1.2. PARAMETER

* USERNAME : alfanumerik (tanpa spasi)
* PASSWORD : alfanumerik (tanpa spasi)

2.1.3. RETURN

* 101 : registrasi berhasil
* 102 : registrasi gagal username sudah ada
* 103 : registrasi gagal kesalahan format
* 101 : REGIST [USERNAME] OK (untuk informasi di server)
* 001 : syntax error
* 003 : failed karena sudah login

2.2. LOGIN

Fitur login merupakan fitur kedua yang wajib dijalankan dalam aplikasi ini sebelum masuk ke dalam sistem. User yang telah melakukan registrasi harus melakukan login dengan menggunakan username dan password yang dimilikinya.

2.2.1. COMMAND

LOGIN<spasi>USERNAME<spasi>PASSWORD

2.2.2. PARAMETER

* USERNAME : alfanumerik (tanpa spasi)
* PASSWORD : alfanumerik (tanpa spasi)

2.2.3. RETURN

* 201 : berhasil masuk
* 202 : login gagal username/password salah
* 201 LOGIN [USERNAME] OK (informasi dalam sisi server)
* 001 : syntax error
* 003 : anda sudah login

2.3. CHAT PERSONAL

Fitur ini merupakan salah satu fitur yang dapat digunakan untuk mengirimkan pesan personal kepada user yang sedang aktif di dalam aplikasi chat ini. Fitur ini hanya dapat dijalankan ketika user telah masuk ke dalam sistem.

2.3.1. COMMAND

SENDP<spasi>USERNAME\_PENERIMA<spasi>MESSAGE

2.3.2. PARAMETER

* USERNAME\_PENERIMA : alfanumerik (tanpa spasi)
* MESSAGE : pesan (256 character)

2.3.3. RETURN

* 301 : pesan personal terkirim
* 301 SENDP [USERNAME] OK (tampilan dalam sisi server)
* 302 : gagal karena username penerima tidak ada
* 303 : gagal karena username sedang tidak online
* 304 : gagal karena message kosong
* 001 : command salah
* 002 : belum login

2.4. CREATE GROUP

Fitur ini merupakan fitur untuk membuat group chat di dalam aplikasi chat ini. Ketika grup dibuat oleh salah satu user maka otomatis user yang membuat grup tersebut akan langsung tergabung ke dalam group chat yang dibuatnya.

2.4.1. COMMAND

CREATE<spasi>NAMA\_GROUP

2.4.2. PARAMETER

NAMA\_GROUP : alfanumerik (tanpa spasi)

2.4.3. RETURN

* 401 : grup berhasil dibuat
* 401 [GROUP] BY [USERNAME] CREATED (tampilan dalam sisi server)
* 402 : gagal karena nama group sudah ada
* 403 : gagal karena kesalahan format parameter
* 001 : command salah
* 002 : anda belum login

2.5. JOIN GROUP

Fitur ini merupakan fitur lanjutan ketika group chat sudah ada di dalam sistem. User baru yang melakukan login dapat bergabung ke dalam group chat yang ada dengan menggunakan command seperti di bawah ini.

2.5.1. COMMAND

JOIN<spasi>NAMA\_GROUP

2.5.2. PARAMETER

NAMA\_GROUP : alfanumerik (tanpa spasi)

2.5.3. RETURN

* 501 : berhasil gabung group
* 501 [USERNAME] JOIN [GROUP] (informasi di tampilan server)
* 502 : gagal karena group tidak ada
* 503 : gagal join karena sudah join grup yang dituju
* 001 : command salah
* 002 : anda belum login

2.6. LEAVE GROUP

User yang telah tergabung ke dalam suatu group chat dapat meninggalkan group tersebut dengan menggunakan fitur leave group. Nantinya ketika user telah meninggalkan group tersebut maka setiap pesan yang ada dalam group tersebut tidak akan dikirimkan ke user yang sudah meninggalkan group tersebut.

2.6.1. COMMAND

LEAVE<spasi>NAMA\_GROUP

2.6.2. PARAMETER

NAMA\_GROUP : alfanumerik (tanpa spasi)

2.6.3. RETURN

* 601 : berhasil keluar group
* 601 [USERNAME] LEAVE [GROUP] (tampilan dalam sisi server)
* 602 : gagal karena anda belum masuk group
* 603 : gagal karena group tidak ada
* 001 : command salah
* 002 : anda belum login

2.7. CHAT GROUP

Ketika group chat sudah ada dan memiliki beberapa member maka, member yang tergabung dalam suatu group chat dapat mengirimkan pesan kepada group tersebut yang nantinya pesan tersebut akan diteruskan ke tiap user yang tergabung dalam suatu group chat tersebut.

2.7.1. COMMAND

SENDG<spasi>NAMA\_GROUP<spasi>MESSAGE

2.7.2. PARAMETER

* NAMA\_GROUP : alfanumerik (tanpa spasi)
* MESSAGE : pesan (256 character)

2.7.3. RETURN

* 701 : SENDG OK pesan terkirim
* 701 [NAMAGRUP] [PENGIRIM] OK
* 702 : SENDG FAILED group tidak ada
* 703 : SENDG FAILED message kosong
* 704 : SENDG FAILED user belum tergabung
* 001 : FAILED command salah
* 002 : FAILED anda belum login

2.8. LIST USER AKTIF

Fitur ini berfungsi untuk menampilkan daftar user yang sedang login di dalam sistem yang nantinya fitur ini berhubungan untuk mengirimkan pesan personal. Hal ini karena tanpa adanya fitur ini maka user akan sulit untuk mengetahui user mana saja yang sedang aktif dan dapat dikirimi pesan.

2.8.1. COMMAND

LISTUS

2.8.2. PARAMETER

-

2.8.3. RETURN

* 003 : menampilkan daftar user yang sedang aktif
* 001 : command salah
* 002 : anda belum login

2.9. LIST GROUP

Fitur ini berfungsi untuk menampilkan daftar grup yang telah dibuat di dalam sistem ini. Fitur ini memudahkan user untuk bergabung ke dalam suatu grup chat di dalam sistem.

2.9.1. COMMAND

LISTGR

2.9.2. PARAMETER

-

2.9.3. RETURN

* 004 : menampilkan list group yang ada
* 001 : command salah
* 002 : belum login

2.10. CHAT PUBLIC/ BROADCAST

Chat public merupakan fitur yang berfungsi untuk mengirimkan pesan seperti broadcast message ke semua user yang sedang login ke dalam sistem.

2.10.1. COMMAND

SEND<spasi>PUBLIC<spasi>MESSAGE

2.10.2. PARAMETER

MESSAGE : pesan (256 character)

2.10.3. RETURN

* 801 : pesan public/broadcast terkirim
* 801 BC BY [USERNAME] SENT (informasi tampilan di sisi server)
* 803 : gagal karena message kosong
* 001 : command salah
* 002 : anda belum login

2.11. LOGOUT

Fitur ini merupakan fitur terakhir di dalam aplikasi chat ini dimana ketika pengguna berada di dalam sistem maka ia dapat keluar atau logout dari aplikasi chat tersebut dan koneksinya akan terputus.

2.11.1. COMMAND

LOGOUT

2.11.2. PARAMETER

-

2.11.3. RETURN

* 203 : berhasil logout
* 203 [USERNAME] LOGOUT (tampilan dalam sisi server)
* 001 : command salah
* 002 : belum login

1. FORMAT TAMPILAN PESAN

Format tampilan pesan ini merupakan tampilan pesan yang diterima setiap user di dalam aplikasi chat ini yang meliputi pesan personal, public (broadcast) chat, dan pesan groupBC(USERNAME) : pesan -> jika pesan berasal dari pesan SEND PUBLICNAMAGRUP(USERNAME) : pesan -> jika pesan berasal dari groupUSERNAME : pesan -> jika pesan berasal dari personal chat

1. PEMBAGIAN KERJA RANCANGAN

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **NRP** | **NAMA** | **RANCANGAN YANG DISELESAIKAN** |
| 1 | 5114100078 | FARHAN RAMADHANA | Fitur Create Group |
| Fitur Leave Group |
| 2 | 5114100076 | MUHAMMAD FAISHAL ILHAM | Fitur Login |
| Fitur List Group yang ada |
| 3 | 5114100092 | KHARISMA MONIKA | Deskripsi rancangan protocol |
| Fitur List User yang Aktif |
| 4 | 5114100080 | WILDAN LUTFI | Fitur Chat Group |
| Fitur Join Group |
| 5 | 5114100132 | ANTONIUS KEVIN | Fitur chat Personal |
| Fitur Chat Public |

C. SERVER

1. CODE SERVER

|  |
| --- |
| *#python version: Python 3.5 for Windows* **import** os **import** socket **import** sys **from** thread **import** \*  sock=socket.socket(socket.AF\_INET,socket.SOCK\_STREAM)  server\_address=(**'10.151.36.250'**,10040) online\_list={} group\_list={} *#def send\_to\_all\_klien(message): # for koneksi in online\_list.itervalues(): # koneksi.sendall(message.encode('utf-8'))  #Kevin* **def** registrasi(uName, uPass):  check = uName.isalnum()  check2 = uPass.isalnum()  **if not** check :  **return** 103  **elif not** check2:  **return** 103  f = open(**'daftarAnggota.txt'**, **'r'**)  data = f.read()  temp = data.split(**'\n'**)  length = len(temp)  **for** x **in** range(0, length-1):  temp[x] = temp[x].split(**' '**)  **for** y **in** range(0, length-1):  **if** temp[y][0]==uName:   **return** 102  f.close()  f = open(**'daftarAnggota.txt'**, **'a+'**)  ID = **"%s %s\n"** % (uName, uPass)  f.write(ID)  f.close()  **return** 101  *#Ical* **def** login(uName, uPass):  f = open(**'daftarAnggota.txt'**, **'r'**)  data = f.read()  temp = data.split(**'\n'**)  length = len(temp)  **for** x **in** range(0, length-1):  temp[x] = temp[x].split(**' '**)  **for** y **in** range(0, length-1):  **if** temp[y][0] == uName:  **if** temp[y][1] == uPass:  **return** 201  f.close()  **return** 202  *#Wildan* **def** check\_nonexist(uName):  f = open(**'daftarAnggota.txt'**, **'r'**)  data = f.read()  temp = data.split(**'\n'**)  length = len(temp)  **for** x **in** range(0, length - 1):  temp[x] = temp[x].split(**' '**)  **for** y **in** range(0,length-1):  **if** temp[y][0]==uName:  **return** False  f.close()  **return** True  *#Risma* **def** check\_this\_client\_online(conn):  **for** con **in** online\_list.itervalues():  **if**(conn == con):  **return** True  **return** False  *#Farhan* **def** check\_group\_member(grup,conn):  nama = username(conn)  **for** user **in** group\_list[grup]:  **if** user == nama:  **return** True  **return** False  *#Kevin* **def** username(conn):  **for** user, con **in** online\_list.iteritems():  **if** con == conn:  **return** user *#def createGroup():  #def joinGroup():  #def chatGroup():  #def leaveGroup():  #def listUser():  #def listGroup():* **def** perintah(command,conn):  temp=command.strip().split()  **if** (check\_this\_client\_online(conn)):  *#WILDAN* **if** temp[0]==**'SEND'**:  **if** len(temp) < 3:  message=**"803\n"** conn.sendall(message.encode(**'utf-8'**))  **else**:  **if** temp[1]==**'PUBLIC'**:  nama = username(conn)  **for** koneksi **in** online\_list.itervalues():  i = 2  koneksi.send(**'BC('**+nama+**'): '**)  **while** i <= len(temp) - 1:  koneksi.send(temp[i] + **' '**)  i += 1  koneksi.send(**'\n'**)  message = **"801\n"** conn.sendall(message.encode(**'utf-8'**))  **print "801 BC BY "**+nama+**" SENT\n"  else**:  message=**"001\n"** conn.sendall(message.encode(**'utf-8'**))  *#KEVIN* **elif** temp[0]==**'SENDP'**:  nama=username(conn)  **if** (len(temp) < 3):  conn.sendall(**'304\n'**)  **elif** (check\_nonexist(temp[1])):  conn.sendall(**'302\n'**)  **elif**(temp[1] **not in** online\_list):  conn.sendall(**'303\n'**)  **else**:  i=2  nama = username(conn)  online\_list[temp[1]].send(nama+**":"**)  **while** i <=len(temp)-1:  online\_list[temp[1]].send(temp[i]+**' '**)  i+=1  online\_list[temp[1]].send(**"\n"**)  **print '301 SENDP '**+nama+**' OK\n'** conn.sendall(**'301\n'**)  *#RISMA* **elif** temp[0]==**'LOGOUT'**:  conn.sendall(**'203\n'**)  nama = username(conn)  **del** online\_list[nama]  **print '203 '** + nama + **' LOGOUT\n'** conn.close()  *#FARHAN* **elif** temp[0]==**'LISTUS'**:  message = **"005\n"** conn.sendall(message.encode(**'utf-8'**))  **for** user **in** online\_list.iterkeys():  conn.sendall(user+**"\n"**)  *#ICAL* **elif** temp[0]==**'CREATE'**:  check = temp[1].isalnum()  **if** len(temp)<2:  message = **"001\n"** conn.sendall(message.encode(**'utf-8'**))  **elif** temp[1] **in** group\_list:  message = **"402\n"** conn.sendall(message.encode(**'utf-8'**))  **elif not** check:  message = **"403\n"** conn.sendall(message.encode(**'utf-8'**))  **else**:  nmGroup = temp[1]  nama = username(conn)  group\_list[nmGroup]=[nama,]  **print "401 "**+temp[1]+**" BY "**+nama+**" CREATED\n"** message = **"401\n"** conn.sendall(message.encode(**'utf-8'**))  *#RISMA* **elif** temp[0]==**'JOIN'**:  nama = username(conn)  **if** len(temp) < 2:  message = **"001\n"** conn.sendall(message.encode(**'utf-8'**))  **elif** temp[1] **not in** group\_list:  message = **"502\n"** conn.sendall(message.encode(**'utf-8'**))  **elif** nama **in** group\_list[temp[1]]:  message = **"503\n"** conn.sendall(message.encode(**'utf-8'**))  **else**:  group\_list[temp[1]].append(nama)  message = **"501\n"  print "501 "**+nama+**" JOIN "**+temp[1]+**"\n"** conn.sendall(message.encode(**'utf-8'**))  *#ICAL* **elif** temp[0] == **'LISTGR'**:  message = **"004\n"** conn.sendall(message.encode(**'utf-8'**))  **for** group\_name **in** group\_list.iterkeys():  conn.sendall(group\_name+**" \n"**)  *#FARHAN* **elif** temp[0] == **'SENDG'**:  **if** len(temp) < 3:  message = **"703\n"** conn.sendall(message.encode(**'utf-8'**))  **elif** temp[1] **not in** group\_list:  message = **"702\n"** conn.sendall(message.encode(**'utf-8'**))  **elif** check\_group\_member(temp[1],conn)==False:  message = **"704\n"** conn.sendall(message.encode(**'utf-8'**))  **else**:  nama = username(conn)  **for** user **in** group\_list[temp[1]]:  i=2  online\_list[user].send(temp[1]+**'('**+nama+**'):'**)  **while** i <=len(temp)-1:  online\_list[user].send(temp[i]+**' '**)  i+=1  online\_list[user].send(**'\n'**)  conn.sendall(**'701\n'**)  **print "701 "**+temp[1]+**" "**+nama+**" OK\n"** *#KEVIN* **elif** temp[0] == **'LEAVE'**:  **if** len(temp) < 2:  message = **"001\n"** conn.sendall(message.encode(**'utf-8'**))  **elif** temp[1] **not in** group\_list:  message = **"603\n"** conn.sendall(message.encode(**'utf-8'**))  **elif** check\_group\_member(temp[1],conn)==False:  message = **"602\n"** conn.sendall(message.encode(**'utf-8'**))  **else**:  nama = username(conn)  **for** user **in** group\_list[temp[1]]:  **if** nama == user:  group\_list[temp[1]].remove(nama)  conn.sendall(**'601\n'**)  **print '601 '**+nama+**' LEAVE '**+temp[1]+**'\n'  elif** temp[0]==**'REGIST' or** temp[0]==**'LOGIN'**:  message = **"003\n"** conn.sendall(message.encode(**'utf-8'**))  **else**:  message=**"001\n"** conn.sendall(message.encode(**'utf-8'**))  **elif** temp[0] == **'REGIST'**:  **if** len(temp) < 3:  message = **"001\n"** conn.sendall(message.encode(**'utf-8'**))  **else**:  regist = registrasi(temp[1], temp[2])  **if** regist == 103:  message = **"103\n"** conn.sendall(message.encode(**'utf-8'**))  **elif** regist == 102:  message = **"102\n"** conn.sendall(message.encode(**'utf-8'**))  **elif** regist == 101:  message = **"101\n"  print "101 REGIST "**+temp[1]+**" OK\n"** conn.sendall(message.encode(**'utf-8'**))  **elif** temp[0] == **'LOGIN'**:  **if** len(temp) < 3:  message = **"001\n"** conn.sendall(message.encode(**'utf-8'**))  **else**:  loginn = login(temp[1], temp[2])  **if** loginn == 201:  uName = temp[1]  message = **"201\n"  print "201 LOGIN "** + temp[1] + **" OK\n"** conn.sendall(message.encode(**'utf-8'**))  online\_list[uName] = conn  *# disini kirim data status login, nama usernya ke list online* **elif** loginn == 202:  message = **"202\n"** conn.sendall(message.encode(**'utf-8'**))  **elif not** check\_this\_client\_online(conn):  message = **"002\n"** conn.sendall(message.encode(**'utf-8'**))  **else**:  message = **"001\n"** conn.sendall(message.encode(**'utf-8'**))   **def** klien(conn):  **while** True:  **try**:  data = conn.recv(1024)  **if** data:  perintah(data.decode(**'utf-8'**),conn)  **else**:  **break  except**:  conn.close()  **break** sock.bind(server\_address) sock.listen(1)   **while** True:  koneksi,addr=sock.accept()  *#client\_list.append(koneksi)* start\_new\_thread(klien,(koneksi,))  koneksi.close() sock.close() |

1. PEMBAGIAN KERJA SERVER

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **NRP** | **NAMA** | **FITUR YANG DISELESAIKAN** |
| 1 | 5114100078 | FARHAN RAMADHANA | Fungsi check group member |
| Fitur list user yang sedang aktif |
| Fitur mengirim pesan di group |
| Fungsi untuk handling syntax error |
| 2 | 5114100076 | MUHAMMAD FAISHAL ILHAM | Membuat fungsi untuk menangani fitur login |
| Fitur create group |
| Fitur list group yang ada |
| Fitur login |
| 3 | 5114100092 | KHARISMA MONIKA | Fungsi mengecek user yang online |
| Fitur logout |
| Fitur join group |
| Fitur Resgistrasi |
| 4 | 5114100080 | WILDAN LUTFI | Fungsi untuk mengecek username yang ingin dikirim pesan ada atau tidak |
| Fitur broadcast message |
| Fungsi untuk handling ketika sudah login, maka tidak bisa login lagi. |
| Fungsi untuk menangani handling ketika belum login |
| 5 | 5114100132 | ANTONIUS KEVIN | Fungsi untuk menangani registrasi |
| Fungsi untuk mencari username dengan koneksi tertentu |
| Fitur private message |
| Fitur leave group |
| Fungsi untuk menghubungkan koneksi server-client |

D. APLIKASI CHAT

1. DESKRIPSI SERVER

Server kelompok 5.

1. FITUR SERVER

2.1. REGISTER

* PARAMETER
* userid -> alfanumerik
* pass -> alfanumerik
* FORMAT

REGISTER(spasi)userid(spasi)password

* RETURN
* SUKSES : kondisi ini menyatakan bahwa clien berhasil ter-registrasi +REG\_SUCCESS
* GAGAL :kondisi ini menyatakan bahwa user tidak berhasil teregister karena userid sudah terpakai -USR\_REGISTERED

2.2. LOGIN

* PARAMETER
* userid -> alfanumerik
* pass -> alfanumerik
* FORMAT

LOGIN(spasi)userid(spasi)password

* RETURN
* SUKSES: +LOGIN\_SUCCESS
* GAGAL :kondisi yang menyatakan jika userid yang dimasukkan belum terdaftar -USER\_NOT\_FOUNDkondisi yang menyatakan jika password yang dimasukkan salah -WRONG\_PASS

2.3. LOGOUT

* PARAMETER

-

* FORMAT

LOGOUT

* RETURN
* SUKSES : return yang menyatakan bahwa client berhasil logout +LOGOUT

2.4. CREATE GROUP

* PARAMETER
* FORMAT

CREATEGROUP(spasi)namagrup(spasi)password

* RETURN
* SUKSES : +GROUP\_CREATED
* GAGAL :kondisi jika nama grup yang akan dibuat sudah terdaftar -GROUP\_EXISTED

2.5. LIST GROUP

* PARAMETER

-

* FORMAT

LISTGROUP

* RETURN
* SUKSES: <<menampilkan list grup yang sudah ada>>

2.6. JOIN GROUP

* PARAMETER
* FORMAT

JOIN(spasi)nama\_grup(spasi)password

* RETURN
* SUKSES: +JOINED\_SUCCESS
* GAGAL:kondisi yang menyatakan gagal untuk join grup karena jumlah anggota sudah penuh -GROUP\_FULL kondisi yang menyatakan bahwa grup ingin diikuti tidak ada -GROUP\_NOT\_FOUNDkondisi yang menyatakan bahwa password yang dimasukka salah -WRONG\_PASS

2.7. LIST USER

* PARAMETER
* FORMAT

LISTUSER(spasi)nama\_grup

* RETURN
* SUKSES : <<Muncul list user dalam grup>>
* GAGAL:kondisi yang menyatakan bahwa grup tidak ada/belum terdaftar -NO\_GROUP

2.8. PRIVATE MESSAGE

* PARAMETER
* FORMAT

PM (spasi) userid\_penerima (spasi) pesan

* RETURN
* SUKSES: +MESSAGE\_SENT
* GAGAL:kondisi yang menyatakan bahwa userid\_penerima tidak ada/belum terdaftar -USER\_NOT\_FOUND

2.9. GROUP MESSAGE

* PARAMETER
* FORMAT

GM(spasi)namagrup(spasi)pesan

* RETURN
* SUKSES : +MESSAGE\_SENT
* GAGAL: kondisi yang menyatakan bahwa grup belum terdaftar -GROUP\_NOT\_FOUND kondisi yang menyatakan bahwa user belum join grup sehingga tidak bisa berkirim pesan -GROUP\_NOT\_JOINED

1. RECEIVE PRIVATE MESSAGE FORMAT

RPM(spasi)userID(spasi)pesan

1. RECEIVE GROUP MESSAGE FORMAT

GM(spasi)nama\_grup(spasi)userID(spasi)pesan

1. SYNTAX ERROR

Jika client memasukkan format yang tidak ada/salah maka akan tampil“-SYNTAX\_ERROR”. Tanpa tanda petikTanda (+) (-) pada return juga merupakan syntax

1. CODE CLIENT

|  |
| --- |
| **import** Tkinter **from** Tkinter **import** \* **import** tkMessageBox  **import** socket **import** sys **from** thread **import** \*  *# Create a TCP/IP socket* sock = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)  server\_address = (**'10.151.36.250'**, 10050)  sock.connect(server\_address) mains = 1  *#KEVIN* **def** terima\_pesan\_personal(text1):   **while** True:  status=sock.recv(1024)  temp = status.split()  i = 2  message = []  message.append(temp[1] + **":"**)  **while** i != len(temp):  message.append(temp[i])  i += 1  message.append(**"\n"**)  strings = **" "**.join(message)  **print** strings  text1.insert(END, strings) *#WILDAN* **def** terima\_pesan\_grup(text1):  **while** True:  status = sock.recv(1024)  **print** status  temp = status.split()  i = 3  message = []  message.append(temp[1] + **": "** + temp[2] + **": "**)  **while** i != len(temp):  message.append(temp[i])  i += 1  strings = **" "**.join(message)  **print** strings  text1.insert(END, strings) *#RISMA* **def** terima\_list\_grup(text1):  sock.sendall(**"LISTGROUP"**)  **while** True:  status=sock.recv(1024)  temp = status.split()  i = 0  **while** i != len(temp) - 1:  text1.insert(END, temp[i] + **"\n"**)  i += 1 *#KEVIN* **def** terima\_pesan():  **while** True:  status = sock.recv(1024)  **if** status == **"+REG\_SUCCESS"**:  tkMessageBox.showinfo(**"Success!"**,**"Registration success!"**)  **elif** status == **"-USR\_REGISTERED"**:  tkMessageBox.showinfo(**"Failed!"**, **"Username already used!"**)  **elif** status == **"+LOGIN\_SUCCESS"**:  response = tkMessageBox.showinfo(**"Success!"**, **"Login success"**)  **if** response == **"ok"**:  mainWindow()  **elif** status == **"-USER\_NOT\_FOUND"**:  tkMessageBox.showinfo(**"Failed!"**, **"Username not found!"**)  **elif** status == **"-WRONG\_PASS"**:  tkMessageBox.showinfo(**"Failed!"**, **"Wrong password"**)  **elif** status == **"+ALREADY\_LOGGED\_IN"**:  tkMessageBox.showinfo(**"Failed!"**, **"Already logged in"**)  **elif** status == **"+GROUP\_CREATED"**:  tkMessageBox.showinfo(**"Success!"**, **"Group successfully created!"**)  **elif** status == **"-GROUP\_EXISTED" or** status == **"-GROUP\_EXISTS"**:  tkMessageBox.showinfo(**"Failed!"**, **"Group already exists!"**)  **elif** status == **"+JOINED\_SUCCESS"**:  tkMessageBox.showinfo(**"Success!"**, **"Successfully joined group!"**)  **elif** status == **"-GROUP\_FULL"**:  tkMessageBox.showinfo(**"Failed!"**, **"Group already full!"**)  **elif** status == **"-GROUP\_NOT\_FOUND"**:  tkMessageBox.showinfo(**"Failed!"**, **"Group not found!"**)  **elif** status == **"-WRONG\_PASS"**:  tkMessageBox.showinfo(**"Failed!"**, **"Wrong group password!"**) *#FARHAN* **def** terima\_list\_grup\_member(text1):  **while** True:  status = sock.recv(1024)  temp = status.split()  i = 0  **while** i != len(temp) - 1:  text1.insert(END, temp[i] + **"\n"**)  i += 1 *#FARHAN* **def** membersListWindowShow():  top1 = Tkinter.Tk()  frame0 = Frame(top1)  frame0.pack()  W1 = Text(frame0)  W1.pack()  start\_new\_thread(terima\_list\_grup\_member, (W1,))  top1.mainloop() *#FARHAN* **def** membersListWindowHandler(entry1):  grup = entry1.get()  sock.sendall(**"LISTUSER "** + grup)  membersListWindowShow() *#FARHAN* **def** membersListWindow():  top1 = Tkinter.Tk()  top1.title(**"Group Members"**)  L0 = Label(top1, text=**"Group Members\n"**)  L0.pack()  frame1 = Frame(top1)  frame1.pack()  frame3 = Frame(top1)  frame3.pack()  L1 = Label(frame1, text=**"Group Name"**)  L1.pack(side=LEFT)  E1 = Entry(frame1, bd=5)  E1.pack(side=RIGHT)  Z = Button(frame3, text=**"View Members"**, command=**lambda**: membersListWindowHandler(E1))  Z.pack()  top1.mainloop() *#ICAL* **def** personalWindowHandler(entry1, entry2,text1):  recipient = entry1.get()  message = entry2.get()  text1.insert(END,**"You -> "**+recipient+**" : "**+message+**"\n"**)  sock.sendall(**"PM "** + recipient + **" "** + message) *#ICAL* **def** groupListWindow():  top = Tkinter.Tk()  top.title(**"Group List"**)  frame0 = Frame(top)  frame0.pack()  W1 = Text(frame0)  W1.pack()  start\_new\_thread(terima\_list\_grup, (W1,))  top.mainloop() *#KEVIN* **def** groupMessageWindowHandler(entry1,entry2,text1):  grup = entry1.get()  message = entry2.get()  text1.insert(END, **"You -> "** + grup + **" : "** + message + **"\n"**)  sock.sendall(**"GM "** + grup + **" "** + message) *#KEVIN* **def** groupMessageWindow():  top = Tkinter.Tk()  top.title(**"Group Chat"**)  frame0 = Frame(top)  frame0.pack()  W1 = Text(frame0)  W1.pack()  frame1 = Frame(top)  frame1.pack()  frame2 = Frame(top)  frame2.pack()  frame3 = Frame(top)  frame3.pack()  L1 = Label(frame1, text=**"Group"**)  L1.pack(side=LEFT)  E1 = Entry(frame1, bd=5)  E1.pack(side=RIGHT)  L2 = Label(frame2, text=**"Message"**)  L2.pack(side=LEFT)  E2 = Entry(frame2, bd=5)  E2.pack(side=RIGHT)  Z = Button(frame3, text=**"Send"**, command=**lambda**: groupMessageWindowHandler(E1, E2,W1))  Z.pack()  start\_new\_thread(terima\_pesan\_grup,(W1,))  top.mainloop() *#RISMA* **def** personalWindow():  top = Tkinter.Tk()  top.title(**"Personal Chat"**)  frame0 = Frame(top)  frame0.pack()  W1 = Text(frame0)  W1.pack()  frame1 = Frame(top)  frame1.pack()  frame2 = Frame(top)  frame2.pack()  frame3 = Frame(top)  frame3.pack()  L1 = Label(frame1, text=**"Recipient"**)  L1.pack(side=LEFT)  E1 = Entry(frame1, bd=5)  E1.pack(side=RIGHT)  L2 = Label(frame2, text=**"Message"**)  L2.pack(side=LEFT)  E2 = Entry(frame2, bd=5)  E2.pack(side=RIGHT)  Z = Button(frame3, text=**"Send"**, command=**lambda**: personalWindowHandler(E1, E2,W1))  Z.pack()  start\_new\_thread(terima\_pesan\_personal,(W1,))  top.mainloop() *#KEVIN* **def** createWindowHandler(entry1,entry2):  groupname = entry1.get()  password = entry2.get()  sock.sendall(**"CREATEGROUP "** + groupname + **" "** + password) *#ICAL* **def** joinWindowHandler(entry1,entry2):  groupname = entry1.get()  password = entry2.get()  sock.sendall(**"JOIN "** + groupname + **" "** + password) *#KEVIN* **def** createWindow():  top1 = Tkinter.Tk()  top1.title(**"Sign Up"**)  L0 = Label(top1, text=**"Create New Group\n"**)  L0.pack()  frame1 = Frame(top1)  frame1.pack()  frame2 = Frame(top1)  frame2.pack()  frame3 = Frame(top1)  frame3.pack()  L1 = Label(frame1, text=**"Group Name (alphanumeric)"**)  L1.pack(side=LEFT)  E1 = Entry(frame1, bd=5)  E1.pack(side=RIGHT)  L2 = Label(frame2, text=**"Password (alphanumeric)"**)  L2.pack(side=LEFT)  E2 = Entry(frame2, bd=5, show=**"\*"**)  E2.pack(side=RIGHT)  Z = Button(frame3, text=**"Create Group!"**, command=**lambda**: createWindowHandler(E1, E2))  Z.pack()  top1.mainloop() *#RISMA* **def** joinWindow():  top1 = Tkinter.Tk()  top1.title(**"Join Group"**)  L0 = Label(top1, text=**"Join A Group\n"**)  L0.pack()  frame1 = Frame(top1)  frame1.pack()  frame2 = Frame(top1)  frame2.pack()  frame3 = Frame(top1)  frame3.pack()  L1 = Label(frame1, text=**"Group Name"**)  L1.pack(side=LEFT)  E1 = Entry(frame1, bd=5)  E1.pack(side=RIGHT)  L2 = Label(frame2, text=**"Password"**)  L2.pack(side=LEFT)  E2 = Entry(frame2, bd=5, show=**"\*"**)  E2.pack(side=RIGHT)  Z = Button(frame3, text=**"Join Group!"**, command=**lambda**: joinWindowHandler(E1, E2))  Z.pack()  top1.mainloop() *#WILDAN* **def** groupWindowHandler(entry1):  choice = entry1.get()  **if** choice == **"1"**:  createWindow()  **elif** choice == **"2"**:  joinWindow()  **elif** choice == **"3"**:  groupListWindow()  **elif** choice == **"4"**:  membersListWindow()  **elif** choice == **"5"**:  groupMessageWindow()  **else**:  tkMessageBox.showinfo(**"Failed!"**, **"Wrong number!"**) *#FARHAN* **def** logoutHandler(top):  sock.sendall(**"LOGOUT"**)  response = tkMessageBox.showinfo(**"Success!"**, **"Logout success"**)  **if** response == **"ok"**:  **global** mains  mains.destroy() *#KEVIN* **def** groupWindow():  top = Tkinter.Tk()  top.title(**"Groups"**)  frame1 = Frame(top)  frame1.pack()  frame2 = Frame(top)  frame2.pack()  L1 = Label(frame1, text=**"Choose a number"**)  L1.pack()  L2 = Label(frame1, text=**"1: Create group"**)  L2.pack()  L3 = Label(frame1, text=**"2: Join group"**)  L3.pack()  L3 = Label(frame1, text=**"3: Group List"**)  L3.pack()  L4 = Label(frame1, text=**"4: Group's Members List"**)  L4.pack()  L5 = Label(frame1, text=**"5: Send Group Message"**)  L5.pack()  E1 = Entry(frame2, bd=5)  E1.pack(side=RIGHT)  Z = Button(frame2, text=**"Submit"**, command=**lambda**: groupWindowHandler(E1))  Z.pack(side=LEFT)  start\_new\_thread(terima\_pesan, ())  top.mainloop() *#ICAL* **def** mainWindow():  top = Tkinter.Tk()  top.title(**"Home"**)  A = Button(top,text=**"Personals"**, command=personalWindow)  A.pack()  B = Button(top, text=**"Groups"**, command=groupWindow)  B.pack()  C = Button(top, text=**"Logout"**, command=**lambda**: logoutHandler(top))  C.pack()  top.mainloop() *#WILDAN* **def** signupWindowHandler(entry1,entry2):  username = entry1.get()  password = entry2.get()  sock.sendall(**"REGISTER "** + username + **" "** + password) *#FARHAN* **def** firstWindowHandler(entry1,entry2):  username = entry1.get()  password = entry2.get()  sock.sendall(**"LOGIN "** + username + **" "** + password) *#WILDAN* **def** signupWindow():  top1 = Tkinter.Tk()  top1.title(**"Sign Up"**)  L0 = Label(top1, text=**"Sign Up\n"**)  L0.pack()  frame1 = Frame(top1)  frame1.pack()  frame2 = Frame(top1)  frame2.pack()  frame3 = Frame(top1)  frame3.pack()  L1 = Label(frame1, text=**"User Name (alphanumeric)"**)  L1.pack(side=LEFT)  E1 = Entry(frame1, bd=5)  E1.pack(side=RIGHT)  L2 = Label(frame2, text=**"Password (alphanumeric)"**)  L2.pack(side=LEFT)  E2 = Entry(frame2, bd=5,show=**"\*"**)  E2.pack(side=RIGHT)  Z = Button(frame3, text=**"Sign Up!"**, command= **lambda**: signupWindowHandler(E1,E2))  Z.pack()  top1.mainloop() *#RISMA* **def** firstWindow():  top = Tkinter.Tk()  **global** mains  mains = top  top.title(**"Welcome!"**)  L0 = Label(top, text=**"Welcome to Wumbo Chat!\n"**)  L0.pack()  frame1 = Frame(top)  frame1.pack()  frame2 = Frame(top)  frame2.pack()  frame3 = Frame(top)  frame3.pack()  L1 = Label(frame1, text=**"User Name"**)  L1.pack( side = LEFT)  E1 = Entry(frame1, bd =5)  E1.pack(side = RIGHT)  L2 = Label(frame2, text=**"Password"**)  L2.pack(side = LEFT)  E2 = Entry(frame2, bd =5,show=**"\*"**)  E2.pack(side = RIGHT, )  A = Button(frame3, text =**"Log In!"**, command = **lambda**: firstWindowHandler(E1,E2) )  A.pack()  L3 = Label(frame3, text=**"\nDon't have an account?\n"**)  L3.pack()  B = Button(frame3, text =**"Sign Up"**, command = signupWindow)  B.pack()  top.mainloop()   start\_new\_thread(terima\_pesan,())  **try**:  firstWindow() **finally**:  sock.close() |

1. PEMBAGIAN KERJA CLIENT

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **NRP** | **NAMA** | **FITUR YANG DISELESAIKAN** |
| 1 | 5114100078 | FARHAN RAMADHANA | Membuat fungsi- fungsi untuk menampilkan list group member |
| Membuat fungsi untuk menghandler ketika user melakukan login di first window |
| Fungsi untuk menangani ketika melakukan logout |
| 2 | 5114100076 | MUHAMMAD FAISHAL ILHAM | Fungsi untuk menampilkan grouplist pada window |
| Membuat fungsi untuk menghandle ketika user telah login ke dalam sistem (menampilkan main window) |
| Fungsi untuk menghandle join group dari window join group |
| Fungsi untuk menghandle pesan personal yang akan dikirimkan |
| 3 | 5114100092 | KHARISMA MONIKA | Fungsi untuk menghandle return list group |
| Fungsi untuk membuat tampilan first window ketika pertama kali aplikasi dijalankan |
| Fungsi untuk menampilkan window untuk melakukan join group |
| Fungsi untuk menampilkan window untuk mengirimkan pesan personal |
| 4 | 5114100080 | WILDAN LUTFI | Fungsi untuk menangani terima pesan group |
| Fungsi untuk menghandle ketika user melakukan sign up |
| Fungsi untuk menampilkan sign up window |
| Fungsi untuk menghandle ketika user memilih pilihan pada window group |
| 5 | 5114100132 | ANTONIUS KEVIN | Fungsi untuk menghandling terima pesan personal |
| Fungsi untuk menhandle terima kode return dari server |
| Fungsi untuk menampilkan pilihan menu pada group window |
| Fungsi untuk menampilkan window untuk membuat group dan fungsi untuk menghandle pembuatan group tersebut |
| Fungsi untuk menampilkan window terkait group message dan fungsi untuk menghandler pengiriman pesan group |