

ISSN 2087-3336 (Print) | 2721-4729 (Online)

TEKNOSAINS: Jurnal Sains, Teknologi dan Informatika

Volume 9, Nomor 2, Juli 2022, hlm 121-135

<http://jurnal.sttmcileungsi.ac.id/index.php/tekno>

DOI: 10.37373

Perancangan aplikasi chat-room dengan prinsip *threading* melalui pemrograman dengan bahasa java

Design of chat-room applications with threading principles through programming in the Java language

Mohamad Firdaus*

*Jurusan Teknik Industri, Fakultas Teknologi Informasi, Universitas Indraprasta PGRI

*Jl. Nangka No. 58 C (TB. Simatupang), Kel. Tanjung Barat, Kec. Jagakarsa, Jakarta Selatan 12530

*Koresponden Email: mfirdausmumu@gmail.com

Artikel dikirim: 30/04/2022

Artikel direvisi: 8/05/2022

Artikel diterima: 23/05/2022

ABSTRAK

Aplikasi komputer untuk berkomunikasi dalam sistem jaringan dibagi menjadi dua yaitu aplikasi yang mendukung komunikasi secara langsung, seperti *video* (gambar), *audio* (suara), *text* (tulisan). Sedangkan yang tidak langsung contohnya adalah aplikasi Email. Aplikasi yang mendukung korespondensi atau saling bertukar pesan dalam komputer disebut dengan aplikasi *chatting*. *Multitasking* adalah kemampuan suatu program yang dapat mengerjakan banyak pekerjaan dalam satu waktu. Sedangkan prinsip *multithreading* adalah kelanjutan dari prinsip *multitasking*. *Multithreading* mencakup banyak *thread* (urutan kerja) dari alur urutan kontrol yang terdapat dalam satu program. Dalam Pembuatan Aplikasi chat-room ini, bahasa pemrograman yang dipakai adalah bahasa Java (*Java Language*). Bahasa Java adalah sebuah bahasa yang berorientasi objek sederhana yang memiliki banyak kesamaan komponen dengan bahasa C dan C++. Kelebihan dalam pemrograman bahasa Java ini antara lain mempunyai sifat *multi-OS platform*, juga dapat digunakan prinsip *threading* dapat meningkatkan kinerja dari program aplikasi yang dibuat. Perancangan aplikasi dalam penelitian ini dibangun dengan metode *Unified Software Development Process* dan perancangan aplikasi menggunakan *Unified Modeling Language* (UML). Mekanisme pengerjaan penelitian ini dibuat dari proses pembuatan Spesifikasi Kebutuhan Perangkat Lunak (SKPL), kemudian dilanjutkan dengan pembuatan Desain Perancangan Perangkat Lunak (DPPL), kemudian diikuti dengan pembuatan aplikasi dan ditutup dengan pengujian. Proses ini dilakukan terus menerus sesuai dengan jangka waktu perencanaan. Hasil dari penelitian ini adalah dokumen perancangan dan aplikasi chat-room sebagai alternatif *software* yang dapat berjalan di berbagai *OS platform*

Kata Kunci: *Chatting, Unified Modeling Language, Unified Software Development Process.*

ABSTRACT

Computer applications to communicate in network systems are divided into two, namely applications that support direct communication, such as video (images), audio (sound), and text (writing). While the indirect example is the Email application. Applications that support correspondence or exchange messages on a computer are called chat applications. Multitasking is the ability of a program to do many jobs at one time. While the principle of multithreading is a continuation of the principle of multitasking. Multithreading includes multiple threads (work sequences) of the control sequence flow contained in a single program. In making this chat-room application, the programming language used is the Java language (Java Language). The Java language is a simple object-oriented language that has many components in common with the C and C++ languages. The advantages of programming the Java language include having a multi-OS platform nature, threading principles can also be used to



TEKNOSAINS: Jurnal Sains, Teknologi & Informatika is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

improve the performance of the application programs created. The application design in this study was built using the Unified Software Development Process method and the application design used the Unified Modeling Language (UML). The working mechanism of this research is made from the process of making Software Requirements Specifications (SKPL), then followed by making a Software Design Design (DPPL), then followed by making applications and closing with testing. This process is carried out continuously according to the planning period. The results of this research are design documents and chat-room applications as alternative software that can run on various OS platforms

Keywords: Chat, Unified Modeling Language, Unified Software Development Process.

1. PENDAHULUAN

Pada saat ini komunikasi berkembang dengan pesat dan telah digunakan pada berbagai bidang kehidupan. Setelah era telepon pada awal abad 19, saat ini dunia komunikasi telah diperkaya dengan diperkenalkannya Internet. Internet adalah sistem komunikasi komputer global yang memungkinkan semua layanan komputer terjadi, singkatnya, internet telah memungkinkan revolusi yang telah mengubah cara kita hidup, bekerja dan bermain[1]. Disamping itu, komunikasi dengan jaringan komputer ini mempunyai banyak kelebihan antara lain, komunikasi dengan media ini tidak hanya melalui suara, tetapi juga dengan visualisasi objek[2]. Karena berbagai kelebihan dan manfaatnya, ditunjang dengan biaya murah dan pengoperasian yang mudah, minat pemakaian sistem jaringan komputer ini semakin lama semakin tinggi. Manfaat-manfaat internet tersebut antara lain:

- a. Mencari informasi yang aktual.
- b. Berkomunikasi.
- c. Hiburan interaktif.

Adapun aplikasi komputer untuk berkomunikasi dalam sistem jaringan dibagi menjadi dua yaitu:

- a. Aplikasi yang mendukung komunikasi secara langsung, seperti
 - Video (gambar).
 - *Telephone* (suara).
 - *Text* (tulisan).
- b. Aplikasi yang mendukung komunikasi secara tak langsung, seperti
 - E-mail (tulisan).

Di antara komunikasi langsung pada sistem jaringan komputer di atas, terdapat pula jenis aplikasi yang mendukung korespondensi atau saling bertukar pesan dalam komputer, disebut dengan chatting. Aplikasi yang mendukung komunikasi ini adalah aplikasi chat. Contoh aplikasi chat ini antara lain: MIRC, Microsoft netmeeting dan lain-lain[3]. Pembahasan aplikasi jaringan untuk berkomunikasi melalui Jaringan komputer, dalam bentuk korespondensi atau saling berkirim tulisan.

2. METODE

2.1 Unified modeling language

Metode perancangan perangkat lunak ini adalah dengan UML (*Unified Modeling Language*), hasil perancangan perangkat lunak, implementasi perangkat lunak dan analisa kebutuhan sistem. Tujuan dibuatnya tahapan-tahapan tersebut adalah agar pembuatan perangkat lunak semakin optimal dan lebih terstruktur, sehingga apabila terjadi kesalahan akan mudah dideteksi.

2.2. Unified software development process

Pada pengembangan atau rekayasa perangkat lunak diperlukan suatu proses berkelanjutan. Hal ini diperlukan untuk menyempurnakan[4] perangkat lunak tersebut sehingga diharapkan dapat mendekati kesempurnaan. Untuk mengembangkan perangkat lunak secara berkelanjutan, diperlukan suatu kerangka kerja. Kerangka kerja pengembangan perangkat lunak ini memandang perangkat lunak sebagai suatu produk yang dihasilkan melalui suatu proses logis yang berurutan dengan masukan-

masukannya yang tepat[5]. Inilah yang dimaksudkan dengan pendekatan process framework [6]. *System Development Life Cycle* (disingkat SDLC)[7] atau disebut daur hidup pengembangan sistem adalah kerangka kerja proses pengembangan perangkat lunak yang berkelanjutan. SDLC membagi tahapan pengembangan perangkat lunak menjadi.

- a. Tahap Komunikasi
- b. Tahap Perencanaan
- c. Tahap Pemodelan
- d. Tahap Konstruksi
- e. Tahap Implementasi

USDP merupakan salah satu kerangka kerja pengembangan perangkat lunak berorientasi objek. USDP, yang kemudian lebih dikenal dengan *Unified Process* (disingkat UP), dikembangkan oleh Grady Booch, Ivar Jacobson dan James Rumbaugh[8], di dalam perkembangannya UP memiliki fase-fase sebagai berikut [9]:

- a. Inception
- b. Elaboration
- c. Construction
- d. Transition

Ditujukan untuk secara konsisten beradaptasi dengan tren pengembangan perangkat lunak yang semakin besar dan semakin kompleksnya. USDP merupakan kerangka kerja pengembangan yang berbasis komponen; ini berarti bahwa perangkat lunak yang dihasilkan akan terdiri atas komponen-komponen perangkat lunak yang saling terhubung melalui antarmuka yang terdefinisi dengan baik[10]. Karakteristik kerangka kerja USDP dalam mengembangkan perangkat lunak disebut *use case driven proces*. Ini mengandung pengertian bahwa USDP terus berpedoman pada diagram *Use-Case*, yang mengilustrasikan fungsionalitas sistem dalam gambar yang mudah dipahami oleh *user*. *Diagram Use Case* merupakan diagram sentral yang menjadi pedoman model pengembangan USDP lainnya. Model pengembangan USDP lainnya adalah[11]:

- a. Model Analisis; yang bertujuan untuk memperhalus dan merinci definisi-definisi masing-masing *use-case*.
- b. Model Perancangan, yang digunakan untuk mendefinisikan struktur statis aplikasi seperti kelas antarmuka dan hubungannya masing-masing dalam kerangka perangkat lunak yang dibangun.
- c. Model Konstruksi, yang memuat komponen-komponen (merekpresentasikan kode-kode dalam bahasa pemrograman tertentu yang dipilih) dan melakukan pemetaan kelas-kelas ke komponen-komponen.
- d. Model Implementasi, yang berguna untuk mendefinisikan simpul-simpul komputer secara fisik dan melakukan pemetaan masing-masing komponen ke setiap simpul komputer yang ada.
- e. Model Pengujian, yang ditujukan untuk mendeskripsikan skenario testing untuk melakukan verifikasi dan validasi terhadap perangkat lunak yang dikembangkan. USDP menggunakan UML (*Unified Modelling Language*) sebagai alat pemodelan perangkat lunak yang dikembangkan. UML merupakan suatu metode modelling generasi ketiga dan bahasa spesifikasi yang sifatnya non-proprietary. UML adalah suatu metode terbuka yang digunakan untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan artifak-artifak dari suatu pengembangan sistem software yang berbasis pada objek[12].

UML merupakan hasil kompilasi best engineering practice yang sudah terbukti sukses dalam pemodelan sistem yang besar, sistem yang kompleks, khususnya pada level arsitektur. Ada tiga aspek utama dalam pemodelan sistem yang mampu didukung oleh UML[10]:

- a. Functional Model, untuk menunjukkan fungsionalitas dari suatu sistem dari sudut pandang user atau pengguna. Ini dicapai dengan menggunakan *Use Case Diagram*.

- b. Object Model, untuk menunjukkan struktur dan sub struktur dari suatu sistem dengan menggunakan object, atribut, operasi dan juga asosiasi. Ini dicapai dengan menggunakan class diagram.
- c. Dynamic Model, menunjukkan internal behavior dan suatu sistem. Ini dicapai dengan menggunakan sequence diagram, activity diagram dan juga statechart diagram.

2.3 Bahasa JAVA

Bahasa pemrograman JAVA adalah bahasa pemrograman berorientasi objek tingkat tinggi, berbasis kelas, yang dirancang untuk memiliki dependensi implementasi sesedikit mungkin. Ini adalah bahasa pemrograman tujuan umum yang dimaksudkan untuk membiarkan pemrogram menulis sekali, berjalan di mana saja yang berarti bahwa kode Java yang dikompilasi dapat berjalan di semua platform yang mendukung Java tanpa perlu dikompilasi ulang [13]. Aplikasi Java biasanya dikompilasi ke bytecode yang dapat berjalan di mesin virtual Java (JVM) apa pun terlepas dari arsitektur komputer yang mendasarinya. Sintaks Java mirip dengan C dan C++, tetapi memiliki fasilitas tingkat rendah yang lebih sedikit daripada keduanya. Java runtime menyediakan kapabilitas dinamis (seperti refleksi dan modifikasi kode runtime) yang biasanya tidak tersedia dalam bahasa kompilasi tradisional. Java awalnya dikembangkan oleh James Gosling di Sun Microsystems dan dirilis pada Mei 1995 sebagai komponen inti dari platform Java Sun Microsystems. Implementasi asli dan referensi kompiler Java, mesin virtual, dan perpustakaan kelas awalnya dirilis oleh Sun di bawah lisensi hak milik.

3. HASIL DAN PEMBAHASAN

Pada pada proses analisis dan perancangan aplikasi. Tahapan analisis dan perancangan yang dilakukan adalah sebagai berikut:

- a. Menentukan kebutuhan sistem dengan menggunakan metode wawancara.
- b. Pemodelan dengan UML (*Unified Modeling Language*)
- c. Pembuatan software dengan bahasa JAVA.
- d. Pengujian software yang sudah dibuat
- e. Pendokumentasian Source code dan keterangan lain.

3.1. Perancangan perangkat lunak dengan UML (*Unified Modeling Language*)

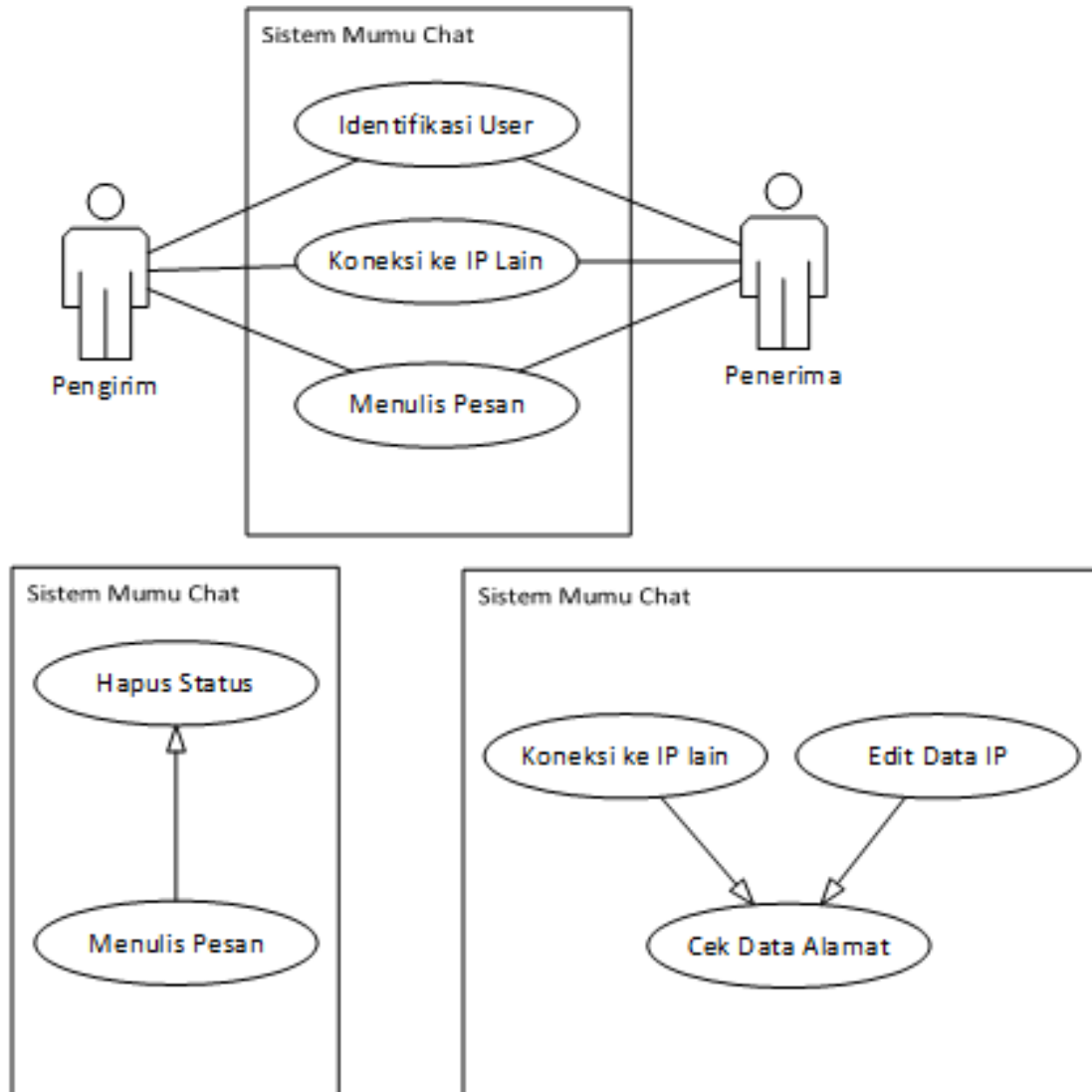
Aplikasi chat dirancang dengan menggunakan diagram UML (*Unified Modeling Language*) dapat dibagi menjadi beberapa bagian diagram. Diagram-diagram UML itu akan dibagi menjadi [4] [8]:

- a. *UML Use Case Diagram*. Merupakan diagram yang menggambarkan hubungan aktivitas sistem aplikasi dengan pengguna, juga desain sistem aplikasi secara garis besar.
- b. *UML Static Structure Diagram*. Merupakan diagram yang menggambarkan struktur program secara lebih terperinci yang ditunjukkan dengan memasukan class, jenis koneksi apa saja yang digunakan dalam sistem aplikasi yang dibuat.
- c. *UML StateChart Diagram*. Merupakan diagram yang menunjukkan urutan aktivitas mesin selama aplikasi berjalan.
- d. *UML Sequence Diagram*. Merupakan diagram yang menggambarkan hubungan pengguna dengan aplikasi yang dihubungkan dengan urutan waktu untuk menjalankan *event* atau proses tertentu.
- e. *UML Deployment Diagram*. Merupakan diagram yang menggambarkan struktur bagaimana perangkat keras dan perangkat lunak menjalankan aplikasi dilengkapi dengan konfigurasi dan pengembangan.
- f. *UML Component Diagram*. Merupakan suatu diagram yang menggambarkan level implementasi dari aplikasi. Terdiri dari class yang dihubungkan dengan ketergantungan antar sesamanya.

- g. *UML Collaboration Diagram*. Merupakan diagram yang menggambarkan kerjasama dari aplikasi yang meliputi hubungan antara objek dan interaksi antar objek dalam aplikasi.
- h. *UML Activity Diagram*. Merupakan diagram yang menggambarkan aliran kontrol dari aplikasi yang akan dibuat.

3.1.1. Diagram UML use case

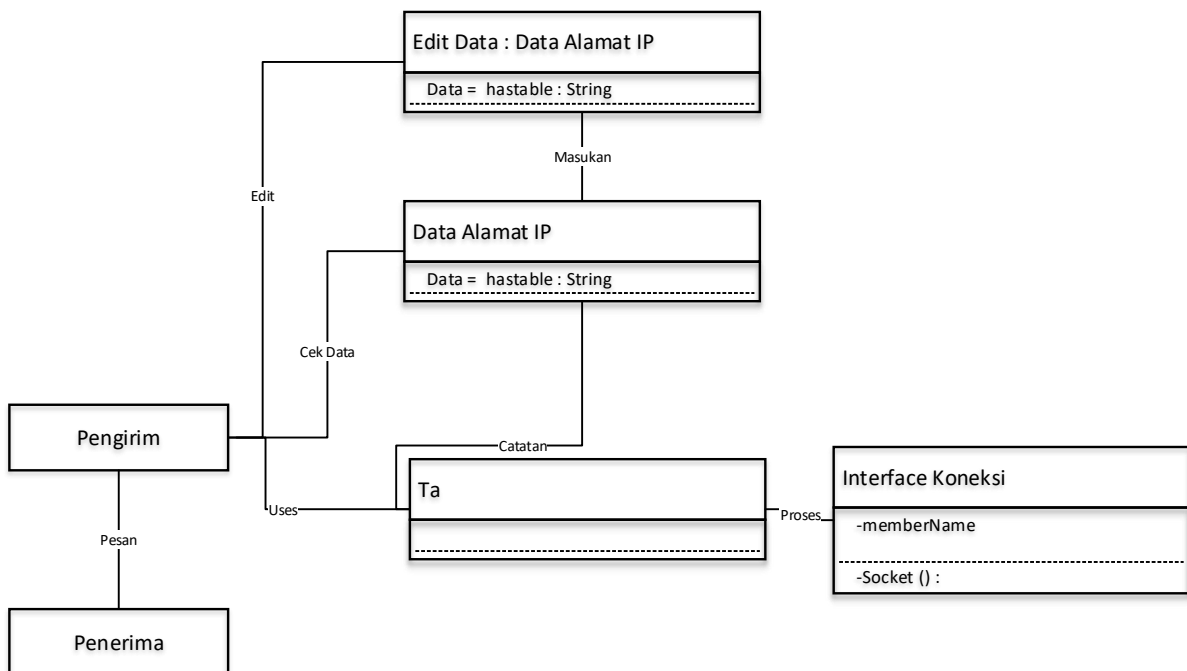
Berikut ini alir perancangan UML use case mumu chat



Gambar 1. UML use case mumu chat

Pada diagram ini terlihat sistem aplikasi ini terdiri dari dua faktor yaitu pengirim dan penerima yang keduanya masuk kedalam batasan saling berkomunikasi dengan aplikasi chat. Urutan proses yang terjadi antara dua pengguna tersebut adalah mengidentifikasi user, menghubungi IP lain, baru menuliskan pesan yang dimaksud. Sedangkan dalam sistem aplikasi chat sendiri terdapat dua buah objek yaitu menghubungi pengguna lain dan status koneksi. Batasan dalam menghubungi pengguna lain terdapat proses koneksi ke IP lain dan mengedit data IP yang keduanya berhubungan dengan proses mencetak alamat. Sedangkan pada batasan status koneksi terdapat proses menulis pesan yang dapat dihilangkan dengan proses menghapus status.

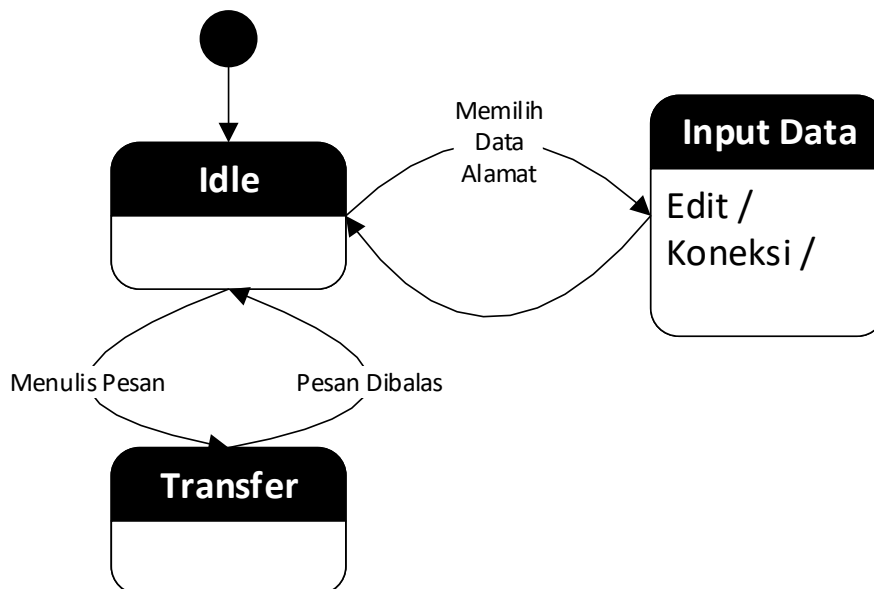
3.1.2. Diagram UML *static structure*



Gambar 2. Diagram static structure mumu chat

Pada diagram *Static Structure* ini dapat terlihat struktur dari program pengiriman pesan pada sistem chat. *Class-class* pada program Java yang terlibat antara lain pada data alamat IP terdapat jenis data string, Class data yang merupakan memanggil dari class server dan class client pada aplikasi, dan terakhir koneksi yang dibentuk adalah koneksi antar socket.

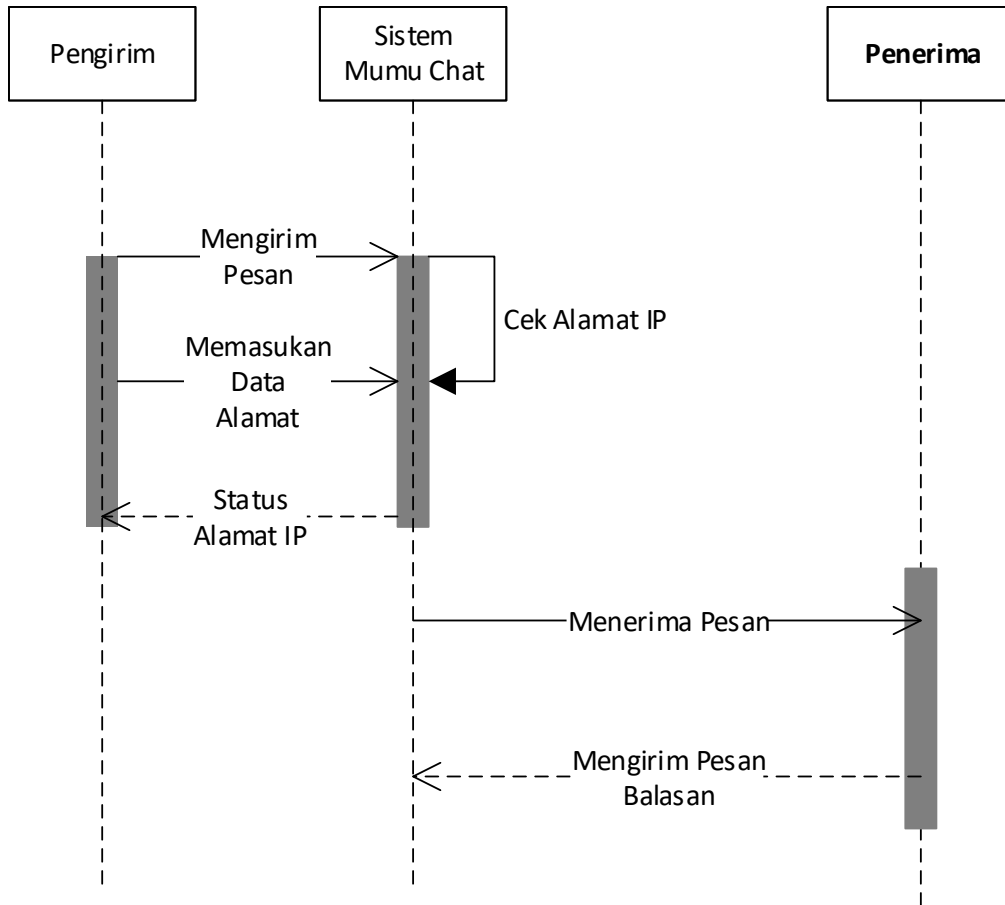
3.1.3. Diagram UML statechart



Gambar 3. UML statechart mumu chat

Pada diagram *Statechart* ini terlihat aktivitas dari mesin yang akan menjalankan aplikasi sistem chat. Mula-mula sistem dalam keadaan *idle* (menganggur), kemudian sistem dimasukkan proses memilih alamat yang didalamnya terdapat proses edit dan koneksi yang dapat dipilih. Bila salah satu proses ini dipilih maka aktivitas mesin akan kembali berjalan. Aktivitas mesin kembali berjalan setelah proses transfer dikerjakan.

3.1.4. Diagram UML sequence

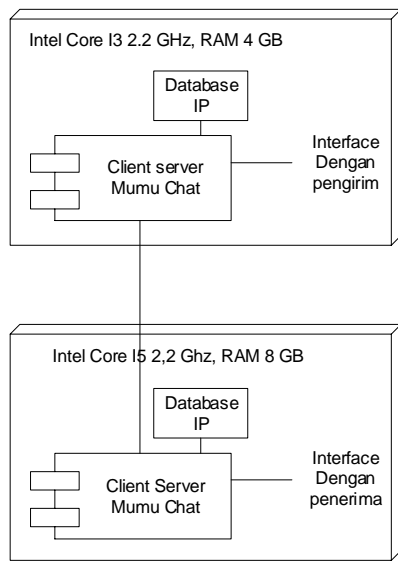


Gambar 4. UML sequence mumu chat

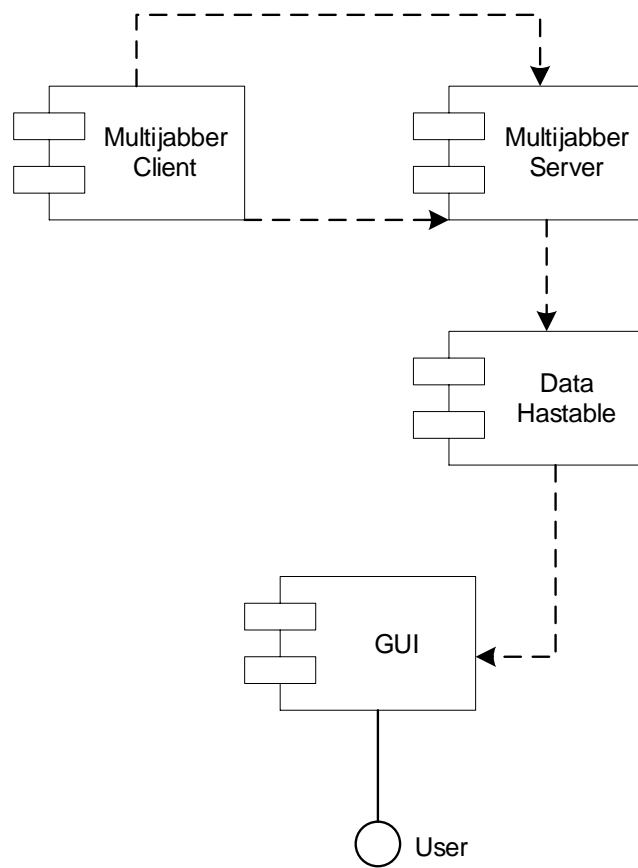
Pada diagram ini dapat dilihat urutan kerja antara objek yang terlibat pada penggunaan aplikasi ini. Pengirim akan mengirimkan pesan, memasukan data alamat dalam satu waktu, sekaligus memperoleh status alamat IP. Sedangkan sistem chat akan mengecek alamat IP, mengirimkan pesan dan menerima pesan dari penerima hal ini juga pada satu tahap. Penerima mendapat pesan sesudah itu mengirimkan pesan kembali.

3.1.5. Diagram UML *deployment*

Pada diagram *deployment* gambar 5 terlihat pengembangan sistem ini pada sistem perangkat lunak dan perangkat keras yang akan mengeksekusi aplikasi. Konfigurasi komputer ke 1 adalah: Intel Core I3 2.2 GHz, RAM 4 GB, yang didalamnya diinstal program chat yang terdiri dari data IP dan sistem client server program chat. Spesifikasi komputer yang ke 2 adalah: Intel Core I5 2,2 Ghz, RAM 8 GB, yang didalamnya diinstal program chat yang terdiri dari data IP dan sistem client server program chat.



Gambar 5. UML deployment mumu chat.

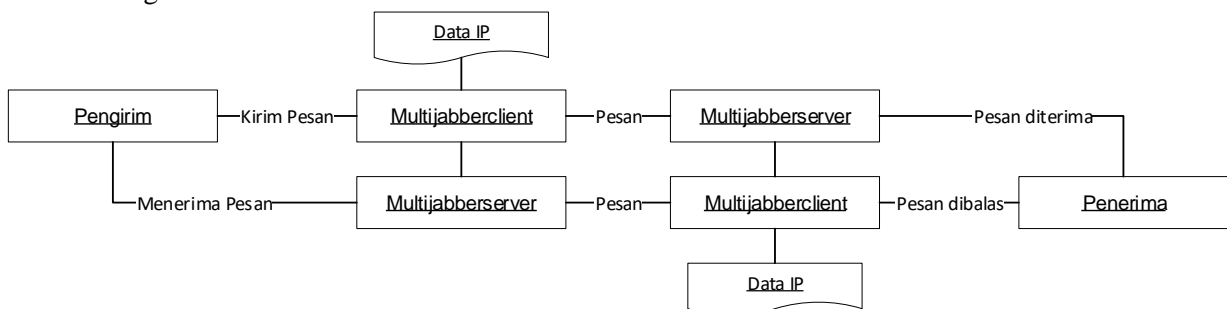


Gambar 6. UML component mumu chat

3.1.6. Diagram UML component

Pada gambar 6 terlihat hubungan antar komponen dalam sistem chat yang menyangkut hubungan antara klien, server, basis data, dan antarmuka grafis dari aplikasi[14]. Disini komponen grafis merupakan interaksi pertama dari user untuk menggunakan aplikasi, lalu user memilih data alamat IP dalam data hastable, kemudian interaksi dengan server yang mengatur socket lalu baru dengan klien.

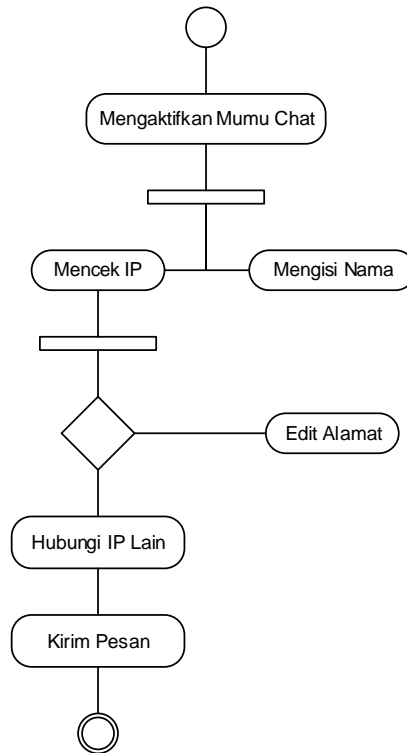
3.1.7. Diagram UML collaboration



Gambar 7. UML collaboration mumu chat

Pada gambar 7 diagram ini terlihat bila pengirim ingin mengirimkan pesan maka harus terlebih dahulu mengecek data alamat IP pada klien kemudian mengirim pesan pada server penerima. Begitu pula kebalikan dari penerima, pesan diterima kemudian membalas dari klien penerima lalu diterima server pengirim.

3.1.8. Diagram UML activity

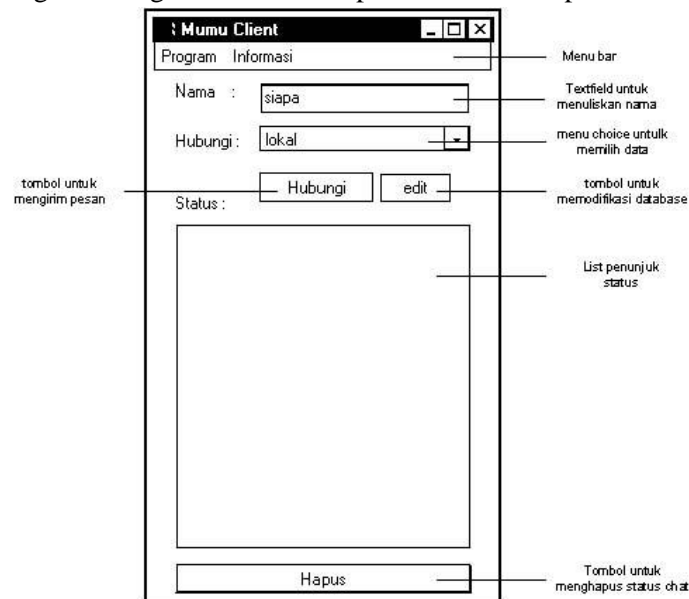


Gambar 8. UML activity mumu chat

Pada gambar 8 diagram ini terlihat aktivitas yang terjadi dalam menggunakan aplikasi chat. Yang pertama-tama adalah mengaktifkan sistem aplikasi chat, kemudian mengecek alamat IP dan mengisi nama. Bila alamat sudah tersedia maka bisa langsung menghubungi alamat tersebut, dan jika alamat belum tersedia dalam mengedit alamat tersebut pada data hash table. Bila koneksi telah terjadi yang terakhir adalah menuliskan pesan lalu dikirim.

3.2. Hasil perancangan perangkat lunak

Hasil dari Perancangan Perangkat Lunak dari aplikasi chat ini dapat terlihat pada Gambar 9.



Gambar 9. Antarmuka program aplikasi mumu chat

Pada antarmuka aplikasi mumu chat ini, jendela mumu client seperti gambar 9 dapat dibagi menjadi empat bagian yaitu [15]:

- a. *Menubar*, merupakan menu yang terdapat pada bagian atas jendela program yang digunakan sebagai menu untuk mengaktifkan antarmuka yang lain
- b. yang berbentuk huruf yang apabila diklik akan memunculkan tabel menu ke bawah.
- c. Tombol, digunakan untuk mengakses antarmuka yang lain yang berbentuk tombol segi empat.
- d. *List*, suatu tempat berbentuk segi empat untuk sarana menulis dengan adanya perpanjangan dengan *scrollbar* (batang yang bisa di naikkan dan di turunkan) agar tulisan di atasnya dapat dilihat kembali apabila sampai batas kotak tersebut, kelebihan list adalah kita tidak dapat merubah tulisan yang ada didalamnya, tidak seperti *textarea* dan *textfield*.
- e. *TextField*, suatu kotak untuk sarana menulis tanpa adanya perpanjangan dengan *scrollbar*, sehingga tulisan akan terus bergerak ke samping kanan.
- f. *Choice*, Suatu kotak pilihan yang dapat muncul dari sebelah kanan kotak pilihan tersebut.

3.2.1. Tombol

Tombol yang ada pada antarmuka Mumu Client antara lain:

- a. Tombol “Hubungi” digunakan untuk menghubungi pengguna yang lain.
- b. Tombol “Hapus” digunakan untuk menghapus informasi pada status *textarea*.
- c. Tombol “edit” digunakan untuk memodifikasi data pada *hash table*.

3.2.2. List

List yang ada pada antarmuka mumu client hanya ada satu yang berfungsi sebagai penunjuk status.

3.2.3. TextField

TextField yang ada pada antarmuka mumu client hanya terdapat satu yang berfungsi sebagai tempat mengisi nama.

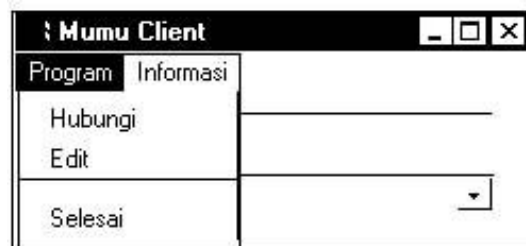
3.2.4. Choice

Choice yang ada pada antarmuka mumu client hanya terdapat satu yang berfungsi sebagai tempat memilih data nama dan IP komputer yang terdapat dalam *hash table*.

3.2.5. Menubar

Menubar yang dibuat terdiri dari 2 menu yaitu (Gambar 10.):

1. Menu program, menu untuk memilih aktivitas aplikasi pada mumu chat.

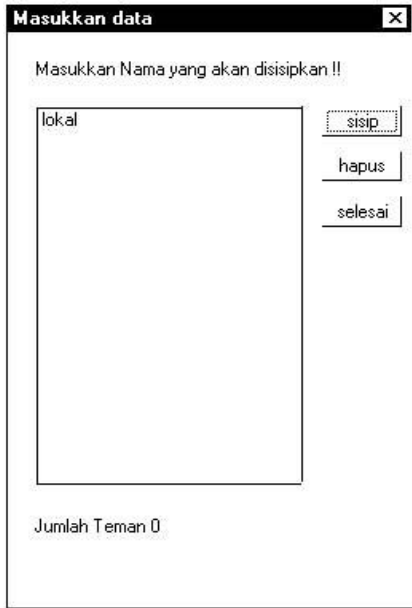


Gambar 10. Menu program

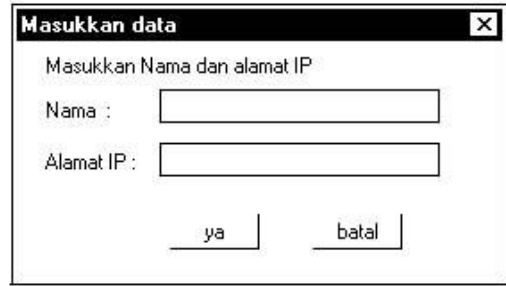
Dalam menu program terdapat beberapa sub menu masing-masing adalah:

- a. Sub menu “Hubungi”, submenu ini digunakan untuk menghubungi klien yang lain apabila telah mengisi nama dan alamat IP yang akan dituju.
- b. Sub menu “Edit”, sub menu ini digunakan untuk memodifikasi basis data yang ada.

Pada Gambar 11 terdapat grafik antarmuka yang lain apabila menu edit dipilih. Pada menu grafik yang diberi label “masukkan data” ini terdapat tiga tombol yaitu: sisip, hapus dan selesai. Apabila menu sisip ditekan akan muncul grafik antarmuka seperti gambar 12.

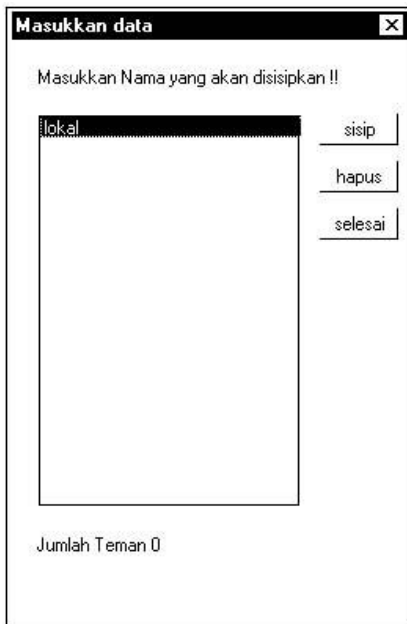


Gambar 11. Dialog edit



Gambar 12. menu dialog untuk memasukkan data

Pada menu ini terdapat dua textfield sebagai tempat untuk memasukan data nama dan alamat komputer atau IP, kemudian terdapat dua tombol pilihan yaitu tombol ya dan tombol batal, bila data dipilih sudah benar dan siap dimasukan maka tombol yang dipilih, sedangkan apabila data yang dimasukkan belum sesuai maka dipilih tombol batal. Sedangkan tombol pada kedua pada menu “masukkan data” adalah tombol hapus tombol ini berguna untuk menghilangkan nama atau alamat pada basis data yang sudah tak dikehendaki lagi. Grafik antarmukanya seperti terlihat pada Gambar 13. Pada menu ini, apabila ingin menghapus data arahkan kursor atau mouse ke nama yang akan dihapus lalu tekan tombol hapus, secara otomatis tombol data akan hilang.



Gambar 13. Menu dialog untuk menghapus data



Gambar 14. Menu informasi

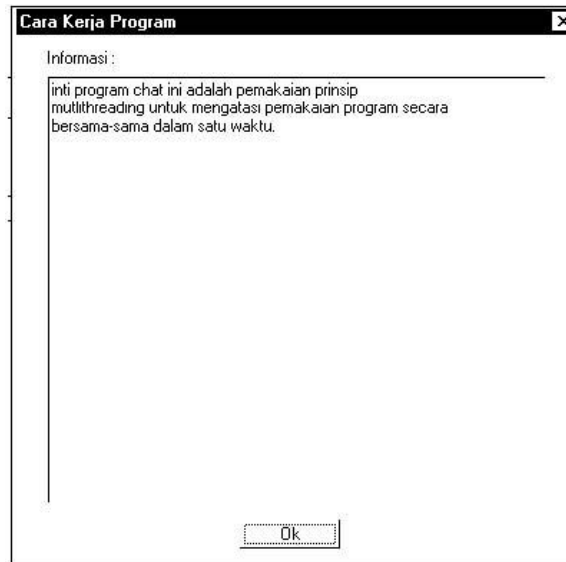
Sedangkan tombol yang terakhir yaitu tombol selesai, digunakan untuk membatalkan masuk ke menu dialog ini atau dapat pula sebagai tombol kembali ke grafik antarmuka awal apabila sudah selesai memasukan data[16].

a. Menu Informasi seperti terlihat pada Gambar 14, terdapat menu untuk memberikan informasi kepada pengguna.

Dalam menu informasi ini terdapat dua sub menu yaitu:

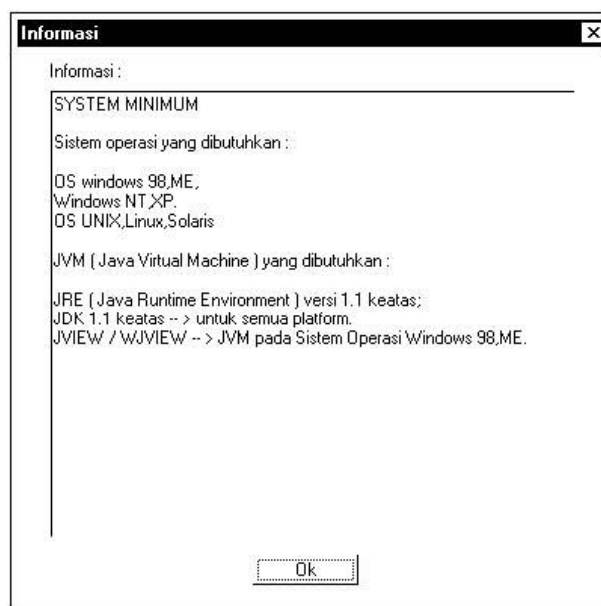
- a. Sub menu “cara kerja”.
- b. Sub menu “info”.

Kedua sub menu ini pada dasarnya akan menghubungkan dengan sebuah grafik antarmuka dialog yang akan memberikan sejumlah informasi yang mungkin akan berguna bagi pengguna. Menu di atas digunakan untuk memberikan informasi tentang cara kerja dari program yang dibuat.



Gambar 15. Dialog cara kerja

Sedangkan apabila menu info dipilih maka yang akan keluar adalah dialog seperti Gambar 15. Dialog ini digunakan untuk memberikan informasi tentang program sistem minimum untuk menjalankan program aplikasi yang dibuat ini. Dan program apa saja yang mendukung sekaligus diperlukan untuk pembuatan dan pengoperasian program aplikasi ini (Gambar 16).



Gambar 16. Dialog informasi

d. Pengujian dan analisa perangkat lunak

Bab ini membahas tentang pengujian dari aplikasi chat. Pengujian ini dilakukan apakah program dapat berjalan dengan baik dan sesuai dengan tujuan yang diharapkan, juga melihat apa saja kelemahan dari aplikasi yang telah dibuat ini.

3.3 Lingkungan uji coba

Pengujian aplikasi chat ini dilakukan dengan dua cara yaitu pada sebuah komputer dan pada dua buah komputer yang terhubung dengan kabel UTP.

3.4 Pelaksanaan uji coba

Berikut ini langkah-langkah yang dilakukan dalam pengujian aplikasi chat:

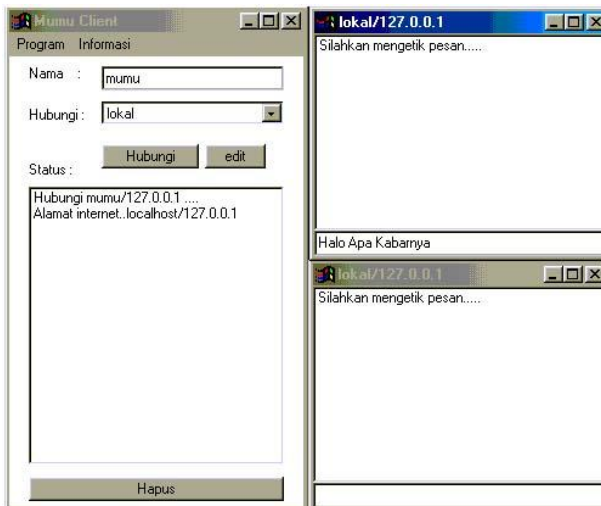
- Komputer yang akan digunakan dinyalakan, sampai windows aktif.
- Menjalankan perintah Jview atau Wjview pada menu run pada menu scrollbar Start untuk menjalankan aplikasi chat, dengan format: `wjview /cp: c:\"direktori tempat program java berada"\Ta.class`.
- Memasukkan nama pada program chat.
- Memilih daftar nama dan alamat komputer yang dituju.
- Mengaktifkan hubungan dengan komputer yang lain.
- Setelah muncul window chatting maka proses chat dapat dilakukan.

3.4.1 Hasil uji coba

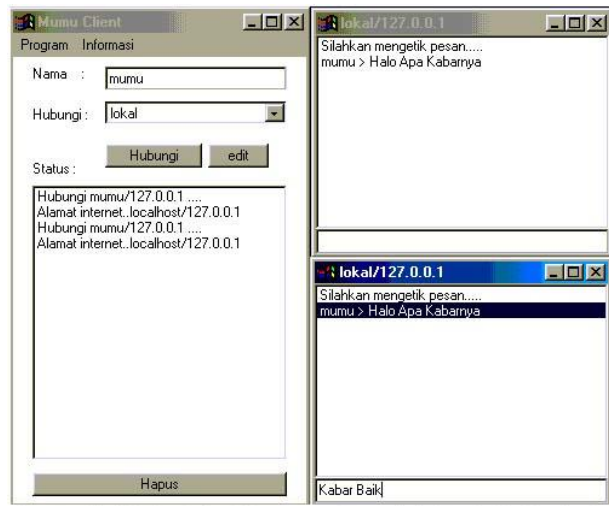
Pada uji coba program aplikasi ini dilakukan dengan dua jenis uji coba yaitu satu komputer dengan alamat lokal dan dengan dua komputer.

3.4.2 Pengujian menggunakan satu komputer

Pada pengujian ini alamat komputer yang dipakai adalah 127.0.0.1 alamat ini digunakan untuk memberikan umpan balik ke alamat IP sendiri (*looping*) [17].

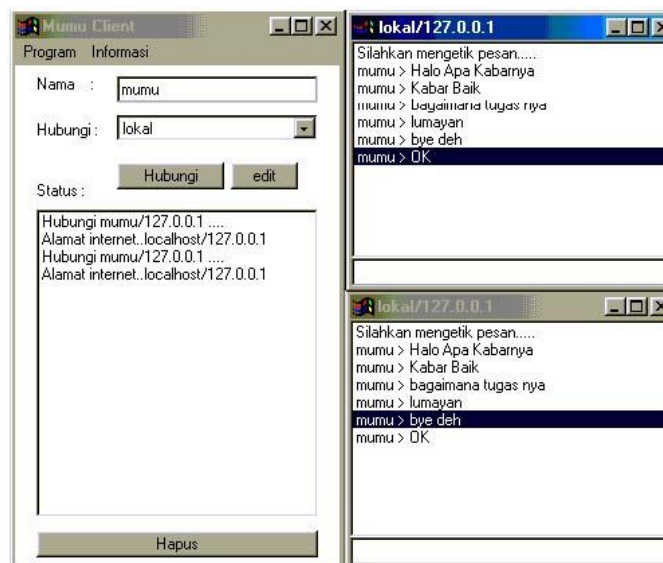


Gambar 17. Uji coba aplikasi chat ketika terjadi koneksi



Gambar 18. Uji coba aplikasi chat dengan umpan balik (*looping*) ke port sendiri

Dari Gambar 17 terlihat ketika memulai koneksi atau hubungan antar socket. Pada Gambar 18. terlihat pada sisi penerima pesan yang telah dikirim dibalas kembali. Gambar 19. memperlihatkan aktivitas tukar-menukar pesan berjalan dengan lancar.



Gambar 19. Uji coba aplikasi chat setelah berjalan beberapa saat.

5. SIMPULAN

Dari hasil pengujian terhadap program aplikasi maka dapat ditarik kesimpulan sebagai berikut: 1. Aplikasi chat ini berjalan baik pada LAN. 2. Jenis koneksi yang dipakai pada aplikasi chat ini adalah peer-to-peer (point-to-point). 3. Aplikasi ini mempunyai dapat berjalan dengan sistem operasi yang berbeda sehingga memudahkan pengguna sistem yang berbeda seperti contoh OS berbasis windows dengan OS berbasis Linux. 4. Karena penggunaan multithreading pada pembuatan program aplikasi ini maka setiap window chatting mempunyai alur kerja sendiri dan berbeda dengan alur window chatting yang lain.

DAFTAR PUSTAKA

- [1] D. E. Comer, *The Internet Book: Everything You Need to Know about Computer Networking and How the Internet Works*. CRC Press, 2018.
- [2] A. Saputra, A. Imamuddin, and P. Sukamto, "Rancang Bangun Aplikasi Sistem Penjualan Case Study: Pt. X," *INFOTECH J. Inform. Teknol.*, vol. 1, no. 2, pp. 78–86, 2020, doi: 10.37373/infotech.v1i2.67.
- [3] E. Lestari Pratiwi, "Konsep Dasar Algoritma Dan Pemrograman Dengan Bahasa Java," *Poliban Press*, 2020.
- [4] S. Karouw, "ANALISA DAN PERANCANGAN APLIKASI DORMITORY MANAGEMENT MENGGUNAKAN UNIFIED SOFTWARE DEVELOPMENT PROCESS," *ejournal unsrat*, vol. 1, p. 3, 2013.
- [5] Erma Kurniasari Nurhasanah, Slamet Abadi, and Pria Sukamto, "SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN MAHASISWA BERPRESTASI DENGAN METODE SIMPLE ADDITIVE WEIGHTING," *TEKNOSAINS J. Sains, Teknol. dan Inform.*, vol. 7, no. 2, 2020, doi: 10.37373/tekno.v7i2.18.
- [6] Roger Pressman, *Software Engineering, A Practitioner's Approach*, 6th ed. McGrawHill, Singapore, 2005.
- [7] P. Adi and G. Permana, "Rancang Bangun Sistem Informasi KIPEM (Studi Kasus : Banjar Delodpasar , Desa Blahkiuh Bali)," *J. Speed - Sentra Penelit. Eng. dan Edukasi*, vol. 10, no. 3, 2018.
- [8] I. Booch, Grady, Rumbaugh, James; Jacobson, *The Unified Modeling Language User Guide*. 1999.

- [9] Henderi, *Analysis and Design System with Unified Modeling Language (UML)*. Tangerang, 2007.
- [10] S. Schach, *Object Oriented Software Engineering*, 8th ed. McGrawHill, 2008.
- [11] Adi Nugroho, *Rekayasa Perangkat Lunak berorientasi Objek dengan Metode USDP*. Yogyakarta: Penerbit Andi, 2010.
- [12] M. Fowler, *UML Distilled; Panduan Singkat Bahasa Pemodelan Obyek Standar*, 3rd ed. Yogyakarta: Penerbit Andi, 2005.
- [13] J. Sanchez and M. P. Canton, *Java Programming for Engineers*. CRC Press, 2002.
- [14] Sumardiono, "PERANCANGAN SISTEM PENILAIAN (E-RESULT) PEGAWAI DENGAN MODEL WATERFALL DI UNIVERSITAS XYZ," *TEKNOSAINS J. Sains, Teknol. dan Inform.*, vol. 8, no. 1, 2021, doi: 10.37373/tekno.v8i1.76.
- [15] B. Eckel, *Thinking in JAVA*. 1999.
- [16] Setio Pamuji, Pria Sukamto, Iskandar, and Haryanto, "SISTEM PAKAR BERBASIS DESKTOP DIAGNOSA PENYAKIT KULIT MENGGUNAKAN METODE BRUTE FORCE," *INFOTECH J. Inform. Teknol.*, vol. 1, no. 2, 2020, doi: 10.37373/infotech.v1i2.68.
- [17] A. . Tanenbaum, *Computer Networks*. Prentice-Hall, 1997.