

Implementasi Komunikasi *Peer-to-Peer* dengan Mengembangkan Aplikasi Menggambar Kolaboratif

Verrel Novendra Sulu¹, David Tulus Halomoan Haryanto², Anantaujas Cipta Adinata³, Sibgah Rabbani Kusuma⁴

Universitas Pradita, Jl. Gading Serpong Boulevard No.1 Tower 1 Banten 15810

^{*1}email: verrel.novendra@student.pradita.ac.id

²email: david.tulus@student.pradita.ac.id

³email: anantaujas.cipta@student.pradita.ac.id

⁴email: sibgah.rabbani@student.pradita.ac.id

ABSTRAK – Dengan berkembangnya teknologi dan internet yang semakin pesat, kolaborasi jarak jauh telah menjadi kebutuhan yang sangat penting, terutama dalam bidang seni dan kreativitas. Pengembangan aplikasi yang mendukung kolaborasi secara real time memungkinkan individu untuk berinteraksi secara efektif meskipun terpisah oleh jarak. Untuk itu, penelitian dilakukan dengan tujuan untuk mengembangkan aplikasi menggambar kolaboratif yang berbasis *Peer-to-Peer* (P2P). Aplikasi dikembangkan menggunakan bahasa pemrograman Python, dengan modul Tkinter untuk antarmuka grafis yang user-friendly, dan socket programming untuk komunikasi antar pengguna melalui jaringan. Dalam penelitian ini, metodologi yang digunakan mencakup studi literatur, implementasi yang dipelajari dalam pembangunan aplikasi, dan mengevaluasi hasil akhir program. Hasil penelitian menunjukkan bahwa aplikasi berhasil mengimplementasikan komunikasi P2P dengan baik dan memungkinkan pengguna untuk menggambar secara bersama-sama dengan pengguna lainnya secara real time tanpa adanya gangguan seperti tabrakan atau masalah yang signifikan. Namun, terdapat beberapa keterbatasan yang telah diidentifikasi, antara lain seperti proses penemuan peer yang masih dilakukan secara manual dan tantangan terkait skalabilitas pada jaringan ketika jumlah pengguna melebihi dari tiga orang atau pengguna. Meskipun demikian, aplikasi ini memberikan wawasan yang bermanfaat mengenai potensi teknologi P2P dalam aplikasi kolaborasi kreatif, serta membuka peluang untuk pengembangan lebih lanjut untuk meningkatkan performa dan memperluas kapasitas aplikasi agar dapat digunakan oleh lebih banyak pengguna.

Kata Kunci – *Peer-to-Peer*, Python, Gambar Kolaboratif

Implementation of *Peer-to-Peer* Communication by Developing Collaborative Drawing Application

ABSTRACT – With the rapid advancement of technology and the internet, remote collaboration has become a crucial need, particularly in the fields of art and creativity. The development of applications that support real-time collaboration enables individuals to interact effectively, even when separated by distance. This research aims to develop a collaborative drawing application based on *Peer-to-Peer* (P2P) technology. The application was developed using the Python programming language, with the Tkinter module for a user-friendly graphical interface and socket programming for communication between users over the network. The methodology used in this study includes literature review, implementation learned during the application development process, and evaluation of the final results. The research findings indicate that the application successfully implemented P2P communication, allowing users to draw together with others in real time without significant issues such as conflicts or other disruptions. However, several limitations have been identified, including the manual peer discovery process and challenges related to scalability when the number of users exceeds three. Nevertheless, this application provides valuable insights into the potential of P2P technology for creative collaboration applications and opens opportunities for further development to enhance performance and expand the application's capacity to accommodate more users.

Keywords - *Peer-to-Peer*, Python, Collaborative Drawing

1. PENDAHULUAN

Dengan berkembangnya internet dan teknologi secara pesat, berbagai aspek pada kehidupan manusia semakin dipengaruhi termasuk bidang seni dan kreativitas. Pada era digital yang semakin maju ini, kolaborasi jarak jauh merupakan hal yang sangat umum. Seperti dalam pola kehidupan sehari-hari maupun bisnis, kebutuhan untuk bekerja bersama-sama namun berada di lokasi yang berbeda semakin meningkat. Hal ini menciptakan permintaan yang besar akan aplikasi-aplikasi yang dapat mendukung kolaborasi jarak jauh yang memungkinkan individu untuk berbagi ide, informasi dan kreativitas secara efisien.

Aplikasi seperti *Google Docs*, *Trello*, *Microsoft Teams* merupakan sebuah aplikasi kolaboratif yang memungkinkan banyak orang untuk menggunakan aplikasi tersebut secara bersamaan, dan terbukti dapat meningkatkan produktivitas dalam berbagi informasi, ide atau dokumentasi. Aplikasi tersebut juga mengubah cara orang berinteraksi dalam lingkungan kerja atau pendidikan, dengan kemajuan teknologi yang pesat, inovasi dalam aplikasi kolaboratif akan lebih beragam dengan berbagai macam fungsionalitas.

Namun, meskipun terdapat berbagai macam aplikasi kolaboratif yang tersedia terdapat sebuah kebutuhan yang belum terpenuhi, seperti aplikasi model menggambar kolaboratif. Aplikasi seperti ini memungkinkan untuk pengguna menggambar dan berkreasi secara bersama-sama. Aplikasi ini memiliki banyak potensi dan dapat dimanfaatkan dalam berbagai macam bidang seperti desain grafis hingga pendidikan seni. Namun, untuk membuat aplikasi tersebut, terdapat tantangan yang harus dihadapi, seperti masalah *latency* pada jaringan, sinkronisasi gambar, dan masalah-masalah lainnya. Beberapa aplikasi membutuhkan server terpusat yang dimana dapat menghadirkan hambatan-hambatan seperti biaya operasional, kecepatan atau privasi data.

Untuk mengatasi masalah seperti latensi tinggi serta privasi dan keamanan data dalam aplikasi kolaboratif, salah satu solusinya adalah menerapkan arsitektur *Peer-to-Peer* (P2P). Dalam sistem P2P, setiap pengguna berperan sebagai "peer" yang dapat berkomunikasi langsung dengan pengguna lain tanpa memerlukan server terpusat. Pendekatan ini berpotensi mengurangi latensi dan meningkatkan privasi serta keamanan data, karena tidak ada server terpusat yang menyimpan informasi pengguna [1].

Beberapa penelitian yang telah mengeksplorasi implementasi teknik P2P dalam aplikasi kolaboratif. Misalnya, studi oleh Bandara dan Jayasumana (2012) membahas sebuah tantangan dan solusi dalam

pengembangan aplikasi kolaboratif berbasis P2P, menyoroti pentingnya sebuah penemuan sumber daya, pemilihan, pencocokan, dan pengikatan dalam sistem semacam itu.

Selain itu, peneliti Bhagatkar et al. (2019) mengembangkan kerangka kerja P2P untuk e-learning yang mencakup streaming kuliah secara langsung dan papan tulis bersama, menunjukkan potensi arsitektur P2P dalam mendukung aplikasi kolaboratif [2].

Aplikasi kolaboratif dalam bidang teknologi informasi dirancang untuk memfasilitasi kerja sama dan komunikasi antara pengguna secara daring. Fungsi utama dari aplikasi ini meliputi pengeditan bersama, penyimpanan file terpusat, komunikasi waktu nyata, dan pelacakan proyek. Dengan memanfaatkan arsitektur P2P, aplikasi kolaboratif dapat meningkatkan efisiensi dan produktivitas tim, serta memastikan bahwa data pengguna lebih aman dan privasi lebih terjaga.

Penelitian ini bertujuan untuk mengembangkan dan mengimplementasikan komunikasi P2P pada aplikasi menggambar kolaboratif, dengan harapan dapat mengatasi permasalahan yang ada pada arsitektur terpusat dan meningkatkan pengalaman pengguna dalam berkolaborasi secara daring.

2. TINJAUAN PUSTAKA

Pembangunan aplikasi menggambar kolaboratif berbasis *Peer-to-Peer* memerlukan penelitian atas cara kerjanya arsitektur dan alat apa saja yang dibutuhkan dalam pembuatannya. Untuk itu, peneliti melakukan studi literatur dan mengkaji berbagai artikel ilmiah dan jurnal yang relevan untuk pembangunan program.

2.1. Arsitektur *Peer-to-Peer*

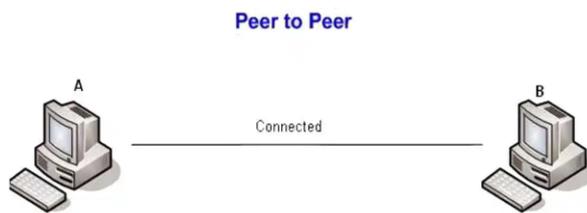
Peer-to-Peer (P2P) merupakan sebuah arsitektur jaringan yang menempatkan klien dan server pada posisi yang setara, di mana setiap "peer" dapat berfungsi sebagai klien maupun server secara bersamaan. Arsitektur ini berbeda dengan arsitektur yang umum digunakan, seperti klien-server, di mana satu pihak berperan sebagai server, sementara pihak lainnya berperan sebagai klien. Arsitektur *Peer-to-Peer* memberikan solusi terhadap masalah 'overload' yang sering dialami oleh sistem terpusat (centralized) seperti arsitektur klien-server [3].

Pada sistem P2P, setiap node atau perangkat dalam jaringan dapat berfungsi untuk menyediakan layanan dan menerima layanan dari perangkat lainnya, yang menciptakan komunikasi secara langsung antar pengguna. Hal ini mengurangi ketergantungan terhadap server pusat. Selain itu keamanan, privasi, dan skalabilitas yang menjadi

keuntungan penggunaan P2P dalam aplikasi, dikarenakan data tidak terpusat pada satu server [2].

Sekarang P2P sering digunakan dalam berbagai sistem aplikasi, contohnya untuk berbagi *file*, bertukar informasi, *messaging*, streaming, dan lain sebagainya. Contohnya seperti Skype, BitTorrent, Gnutella, dan Kaaza [4][5].

Selain itu P2P juga digunakan dalam sistem aplikasi yang lebih kompleks seperti sistem pembayaran terdesentralisasi (Bitcoin) dan blockchain yang memungkinkan transaksi dan data disimpan secara aman tanpa adanya otoritas pusat [6].



Gambar 1. Peer-to-Peer

Berdasarkan Gambar 1 *Peer-to-Peer*, ilustrasi tersebut menggambarkan arsitektur jaringan di mana setiap komputer atau perangkat (*peer*) dalam jaringan memiliki peran yang setara. Setiap *peer* dapat berfungsi sebagai klien maupun server secara bersamaan, memungkinkan komunikasi langsung antar perangkat tanpa memerlukan server pusat. Hal ini menciptakan jaringan yang lebih terdistribusi, mengurangi ketergantungan pada satu titik pusat, dan meningkatkan fleksibilitas serta skalabilitas sistem.

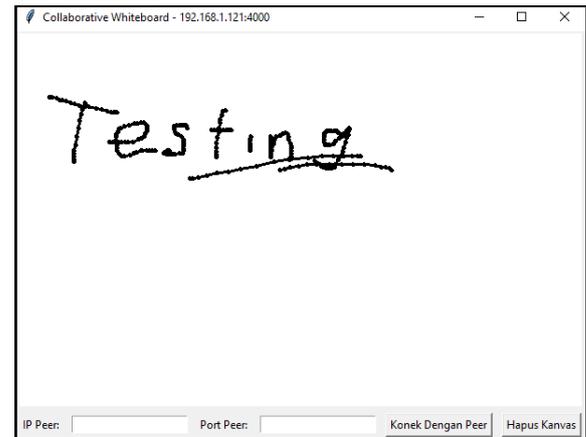


Gambar 2. Python

Berdasarkan Gambar 2, Python merupakan bahasa pemrograman yang umum digunakan untuk pembuatan situs web dan software, machine learning, data analysis, dan networking. Python dapat digunakan untuk variasi program yang berbeda dan tidak hanya untuk masalah tertentu saja.

Yang membuat Python kuat dibanding bahasa pemrograman lainnya adalah aksesnya yang luas terhadap library dan modul bawaan, maupun eksternal. Dalam Python, pada dasarnya tidak dapat membuat sebuah GUI yang interaktif. Untuk itu, diperlukan modul seperti Tkinter, kivy atau wxPython[7].

2.3. TKinter

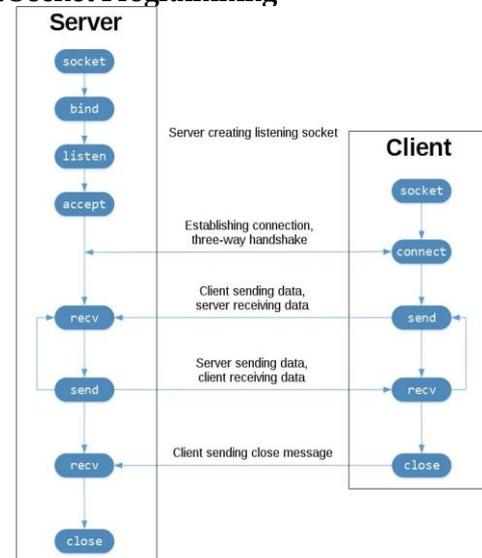


Gambar 3. Aplikasi GUI

Pada Gambar 3, Tkinter adalah modul bawaan Python yang menyediakan antarmuka untuk pengembangan aplikasi GUI (Graphical User Interface). Dengan Tkinter, pengguna dapat membuat jendela, tombol, kotak teks, kanvas, menu, dan elemen GUI lain dengan cukup mudah. Tkinter dikembangkan dengan TCL (Tool Command Language) dan memiliki support untuk Windows, Linux, dan Mac OS[8].

Tkinter memiliki berbagai fungsi yang dapat digunakan dalam pembuatan aplikasi, contohnya seperti fungsi `create_oval`, yang akan membuat sebuah lingkaran pada titik tertentu pada kanvas yang telah dibuat[9].

2.4. Socket Programming



Gambar 4. Socket Programming

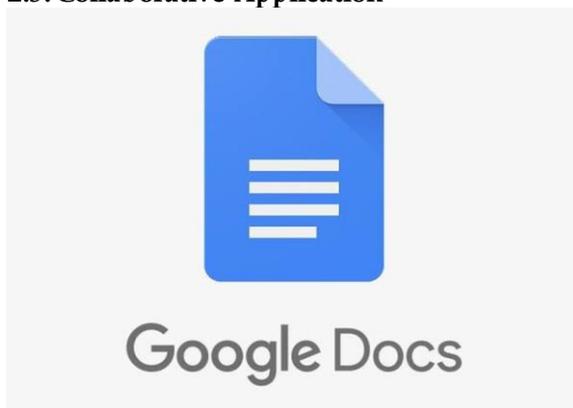
Pada Gambar 4, metode atau teknik socket programming merupakan dasar dari komunikasi antar perangkat dalam jaringan, yang sangat relevan untuk membangun aplikasi berbasis real-time seperti whiteboard kolaboratif. Pada aplikasi menggambar kolaboratif, socket digunakan untuk mengelola

transfer data, seperti koordinat gambar, warna, dan alat menggambar. Dengan menggunakan modul *socket* dalam *Python*, setiap perangkat dapat bertindak sebagai sebuah server maupun client, memungkinkan komunikasi *Peer-to-Peer* secara langsung.

Dalam implementasi berbasis TCP (*Transmission Control Protocol*), *socket* memastikan data seperti gerakan kuas atau titik gambar dikirimkan secara andal dari satu peer ke peer lain, menjaga sinkronisasi di antara semua pengguna pada kanvas bersama. Sementara itu, protokol UDP dapat digunakan jika prioritasnya adalah kecepatan pengiriman data, meskipun kurang andal dibanding TCP.

Pendekatan ini memungkinkan semua node dalam jaringan untuk saling berkomunikasi tanpa memerlukan server terpusat, yang merupakan karakteristik utama dari aplikasi berbasis P2P. Dengan adanya kemampuan ini, setiap pengguna dapat menggambar di kanvas secara bersamaan, dan perubahan yang dilakukan akan langsung terlihat oleh pengguna lain secara real-time. Implementasi ini mencerminkan manfaat utama *socket programming* dalam menciptakan solusi komunikasi yang fleksibel, sinkron, dan efisien.

2.5. Collaborative Application



Gambar 5. Aplikasi Kolaboratif

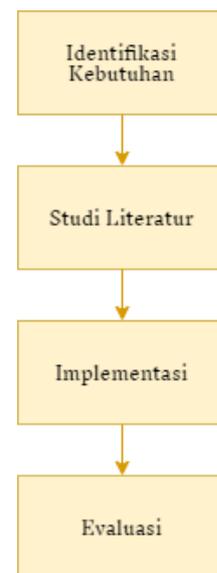
Pada Gambar 5 disajikan sebuah aplikasi kolaboratif memungkinkan pengguna untuk bekerja secara bersama-sama dalam waktu nyata. Salah satu contoh populer adalah Google Docs pada gambar 2.4.1, dimana beberapa pengguna dapat mengedit dokumen secara bersamaan. Untuk mendukung fitur kolaborasi ini, aplikasi harus memiliki mekanisme sinkronisasi data yang kuat dan menangani konflik perubahan dengan efisien[10][11].

Dalam konteks menggambar kolaboratif, aplikasi ini memungkinkan pengguna untuk berinteraksi secara langsung melalui antarmuka visual. Pengguna dapat menggambar, berbagi ide, dan bekerja secara kolektif pada kanvas yang sama dengan pembaruan real-time[12]. Implementasi

aplikasi kolaboratif sering kali memanfaatkan teknologi *Peer-to-Peer* untuk meningkatkan responsivitas dan keandalan komunikasi [13].

3. METODOLOGI PENELITIAN

Pada metodologi penelitian ini mengadopsi sebuah pendekatan terstruktur untuk merancang, mengembangkan, hasil penelitian diharapkan dapat berkontribusi pada proses pengembangan, dan mengevaluasi aplikasi menggambar kolaboratif yang memanfaatkan teknologi *Peer-to-Peer*. Metode yang digunakan meliputi studi literatur, perancangan sistem, implementasi prototipe, pengujian kinerja, dan pengujian kengan aplikasi kolaboratif yang tidak hanya efisien secara teknis, tetapi juga memberikan pengalaman pengguna yang menyenangkan. Alur kegiatan dapat dibuat berdasarkan metodologi penelitian yang digunakan seperti pada Gambar 6.



Gambar 6. Alur Penelitian

1. Identifikasi Kebutuhan

1. Koneksi *Peer-to-Peer* (P2P):

- Aplikasi harus memungkinkan koneksi langsung antara pengguna melalui alamat IP dan port tanpa memerlukan server terpusat.
- Sistem harus dapat mendeteksi dan menangani permintaan koneksi baru secara dinamis.
- Setiap peer harus dapat mengirim dan menerima data secara real time, termasuk perubahan pada kanvas gambar.

2. Kanvas Kolaboratif:

- Menyediakan kanvas digital yang dapat dibagikan oleh semua

pengguna yang terhubung.

- Perubahan yang dilakukan oleh satu pengguna (seperti menggambar, menghapus, atau mengubah warna) harus disinkronkan secara real time ke semua peer.

3. Manajemen Peer:

- Sistem harus memungkinkan penambahan dan penghapusan peer secara dinamis dari daftar koneksi.
- Informasi tentang peer yang terhubung, seperti alamat IP dan status koneksi, harus disimpan dan diperbarui secara otomatis.

4. Antarmuka Pengguna (UI):

- Menyediakan formulir untuk memasukkan alamat IP dan port peer lain.
- Memiliki pemberitahuan visual untuk menunjukkan status koneksi antara peer. o Menampilkan kanvas interaktif dengan alat gambar yang intuitif.

5. Pengiriman dan Penerimaan Data:

- Pesan data yang dikirim harus dikodekan untuk transmisi yang efisien dan didekodekan di sisi penerima.
- Semua perubahan pada kanvas harus disiarkan ke semua rekan yang terhubung.

6. Performa dan Efisiensi:

- Sistem harus memiliki latensi rendah untuk memastikan pengalaman menggambar yang responsif.
- Sinkronisasi data antara rekan harus efisien dan tidak membebani bandwidth jaringan.

2. Studi Literatur

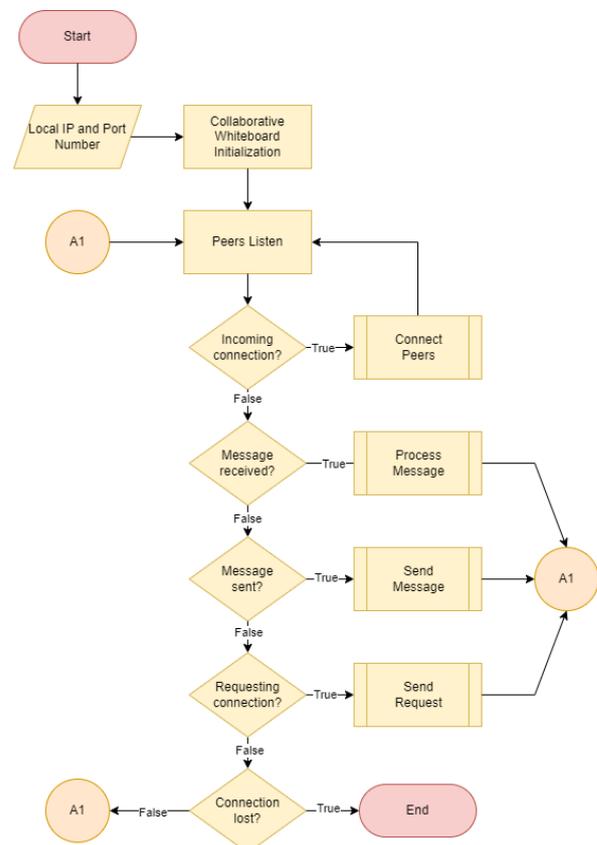
Bauwens dkk. (2019) menyoroti bahwa kolaborasi dalam jaringan P2P sering kali tidak memerlukan izin, sehingga memungkinkan partisipasi tanpa memerlukan izin dari pihak lain. Sistem P2P menawarkan fleksibilitas untuk berbagi dan berkolaborasi secara real-time melalui infrastruktur teknologi yang mendukung koneksi langsung antar perangkat. Hal ini relevan untuk aplikasi menggambar kolaboratif, di mana sinkronisasi data yang *real-time* antarpengguna merupakan persyaratan utama.

Dalam konteks teknologi informasi, jaringan P2P

digunakan untuk meningkatkan sebuah efisiensi komunikasi dan memungkinkan pengolahan data secara terdesentralisasi. Durbin dan Brooks (2019) dalam penelitian mereka tentang model permutasi P2P untuk DNS menunjukkan bahwa pendekatan ini dapat mengurangi latensi dan meningkatkan skalabilitas sistem, yang sangat penting untuk aplikasi kolaboratif berbasis jaringan [14].

3. Implementasi

Pada tahap Implementasi, peneliti akan mulai merancang pembuatan program dari hal yang telah dipelajari. Program memiliki alur sebagai berikut:



Gambar 7. Alur Program

Pada bagian pertama, diperlukan IP dan *port* dari pengguna aplikasi. Karena aplikasi tidak berskala besar, peneliti memilih pendekatan yang lebih sederhana, yaitu menggunakan IP *private* atau lokal dengan *port* yang dapat dipilih sesuai kemauan pengguna. *Port* yang diberikan pengguna diwajibkan berada di atas angka 1,024 dan di bawah angka 65,535.

UI (*User Interface*) dari aplikasi akan ditampilkan, dengan sebuah kanvas kosong dan sebuah *form* untuk mengisi IP dan juga *Port peer* selain pengguna.

Lalu, setiap pengguna akan mendengar apabila terdapat permintaan dari pengguna lainnya. Bila terdapat permintaan untuk terhubung, pengguna, atau *peer* yang mengirim permintaan akan membuka socket untuk terhubung dengan *peer* yang dikirim

permintaan tersebut. Setelah terhubung, kedua *peer* dan pengguna akan menambahkan informasi koneksi tersebut ke daftar *peers* pada program mereka.

Proses selanjutnya berjalan apabila terdapat sebuah pesan yang dikirimkan oleh salah satu *peers* pengguna. Pesan tersebut akan melalui sebuah proses untuk mendecode informasi yang diberikan. Dalam proses ini, data mentah dari pesan yang diterima diubah menjadi informasi yang dapat dipahami oleh sistem. Setelah proses decoding berhasil, informasi tersebut akan dibroadcast ke semua *peer* yang terhubung dengan pengguna. Setiap *peer* yang menerima broadcast akan menjalani proses yang mirip, yaitu menerima dan mendecode. Dalam aplikasinya,

Proses berikutnya berjalan apabila pengguna ingin mengirimkan sebuah pesan pada *peer*. Pesan pertama akan melalui proses *encoding* agar data dapat dikirim dan diterima oleh *peer*. Pada bagian *peer*, terjadi juga hal yang sama seperti proses sebelumnya, yaitu *decoding*, dan *broadcasting* pada semua *peer*.

Berikutnya adalah proses yang dijalankan apabila pengguna ingin terhubung dengan *peer* lain. Pengguna pertama memasukkan alamat IP dan *port* dari *peer* yang ingin dihubungkan melalui UI pengguna. Kemudian sistem akan mencoba melakukan koneksi kepada *peer* tersebut dengan membuat sebuah socket dan menginisialisasi komunikasi. Apabila koneksi berhasil, alamat IP dan *port* dari masing-masing *peer* akan disimpan dalam daftar *peers*.

Program akan berakhir apabila semua *peers* telah memutuskan hubungan dengan pengguna.

4. Evaluasi

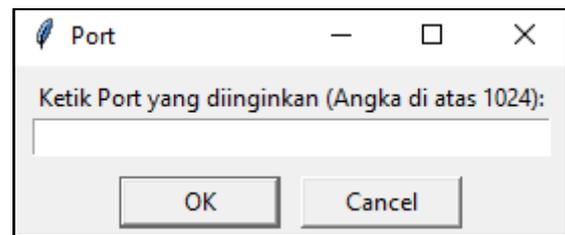
Pada tahap evaluasi, penelitian ini bertujuan untuk menilai sejauh mana aplikasi menggambar kolaboratif *Peer-to-Peer* (P2P) memenuhi tujuan yang ditetapkan. Evaluasi pertama akan mengukur kinerja teknis, terutama latensi dan kecepatan sinkronisasi antar perangkat dalam jaringan P2P, untuk memastikan aplikasi berjalan dengan responsivitas tinggi dan tanpa penundaan yang berarti. Selanjutnya, aspek sinkronisasi data akan diuji untuk memastikan perubahan yang dilakukan oleh satu pengguna, seperti menggambar bentuk bulat, diterima dengan benar oleh pengguna lain tanpa kesalahan atau perbedaan tampilan, mendukung kolaborasi real-time yang akurat.

4. HASIL DAN PEMBAHASAN

4.1 Visual Aplikasi

Pada saat aplikasi diluncurkan, akan tampil sebuah dialog yang meminta *pengguna* untuk

mengetik *port* yang ingin mereka gunakan.



Gambar 8. Dialog pemilihan port

Aplikasi memerlukan angka port di atas 1,024 dan di bawah 65,535. Apabila pengguna menekan tombol *cancel*, pengguna akan ditampilkan error dan keluar dari aplikasi.

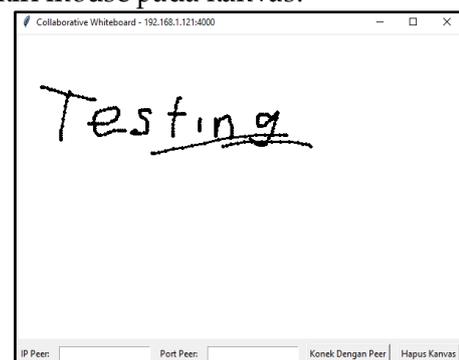
Setelah memilih angka port yang diinginkan, pengguna akan diarahkan pada kanvas kosong dengan UI untuk membuat hubungan dengan *peer* atau pengguna lainnya dalam satu jaringan.



Gambar 9. Kanvas kosong

Pada bagian atas dari kanvas, dapat terlihat judul dari jendela, yaitu 'Collaborative Whiteboard - 192.168.1.121:4000', yang berarti pengguna memilih *port* 4000 pada pilihan Gambar 9.

Dengan kanvas tersebut, pengguna dapat menggambar yang diinginkan dengan menekan tombol kiri mouse pada kanvas.



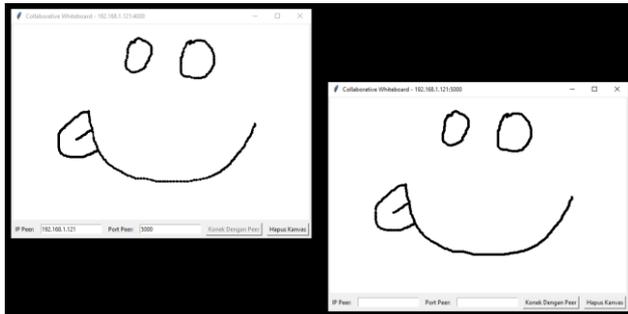
Gambar 10. Kanvas dengan gambar 'Testing'

Pengguna dapat membuat sebuah hubungan dengan pengguna lainnya dengan mengisi *input* pada bagian bawah dari jendela. Gambar 11 menunjukkan sebuah koneksi yang sukses antara dua *peer*.

```
Alamat IP Lokal Anda: 192.168.1.121
Mendengar pada alamat 192.168.1.121:4000
Terhubung pada peer 192.168.1.121:5000
```

Gambar 11. Koneksi sukses dalam jaringan lokal

Setelah hubungan berhasil dilakukan, kedua pengguna atau *peers* dapat menggambar dan menghapus Kanvas secara *real-time* dan perubahan yang dibuat oleh setiap *peer* akan juga tercermin di aplikasi *peer* lainnya, seperti halnya pada Gambar 12.



Gambar 12. Penggambaran Kolaboratif

4.2 Pembahasan

Implementasi jaringan *Peer-to-Peer* (P2P) dalam aplikasi menggambar kolaboratif telah menunjukkan hasil yang memuaskan. Arsitektur P2P memungkinkan setiap perangkat berfungsi sebagai klien dan server sekaligus, sehingga meningkatkan fleksibilitas dan efisiensi dalam berbagi sumber daya.

Meskipun konsep P2P telah ada sejak lama, teknik ini tetap relevan dan modern, terutama dalam konteks jaringan nirkabel. Teknologi nirkabel seperti Wi-Fi dan Bluetooth memungkinkan perangkat terhubung tanpa kabel, mendukung mobilitas yang lebih besar bagi pengguna.

Unsur *Peer-to-Peer* dalam aplikasi ini terlihat dari kemampuan setiap pengguna untuk bertindak baik sebagai server maupun sebagai client. Setiap pengguna juga dapat menerima sebuah koneksi dari pengguna lain dan juga melakukan inisiasi koneksi ke pengguna lain. Dengan demikian, arsitektur jaringan yang dihasilkan bersifat lebih terdistribusi dibandingkan pada arsitektur klien-server yang tradisional.

Implementasi arsitektur *Peer-to-Peer* (P2P) dalam aplikasi menggambar kolaboratif *real-time* telah mencapai sebuah tujuan utamanya, yaitu memungkinkan pengguna untuk menggambar bersama secara langsung tanpa memerlukan server terpusat. Namun, sistem yang dikembangkan masih memiliki beberapa keterbatasan, terutama dalam mekanisme penemuan *peer*. Saat ini, pengguna harus secara manual memasukkan alamat IP dan port *peer* yang ingin dihubungkan, yang tidak efisien dan kurang praktis. Selain itu, sistem mengalami

kesulitan ketika lebih dari tiga *peer* terhubung secara bersamaan, menunjukkan bahwa skala jaringan yang didukung masih terbatas. Mekanisme *broadcast* data juga menggunakan metode sederhana, di mana setiap data yang diterima oleh satu *peer* langsung disebar ke *peer* lainnya. Pendekatan ini dapat menyebabkan masalah seperti latensi dalam pengiriman dan penerimaan data, serta potensi terjadinya *loop* yang tidak terkontrol. Untuk mengatasi masalah ini, implementasi algoritma gosip dapat menjadi solusi yang efektif [15].

5. SIMPULAN

Kemajuan teknologi dan internet yang pesat telah menghasilkan lahirnya berbagai aplikasi kolaboratif yang memfasilitasi kerjasama jarak jauh. Persyaratan khusus, seperti aplikasi sketsa kolaboratif, masih memiliki potensi yang belum terealisasi. Dengan berbagai kesulitan seperti latensi, sinkronisasi, dan privasi data, arsitektur *Peer-to-Peer* (P2P) berkembang sebagai solusi atas kendala desain terpusat. Implementasi P2P memungkinkan komunikasi langsung antara pengguna tanpa memerlukan server pusat, yang meningkatkan efisiensi, privasi, dan keamanan data. Pembuatan aplikasi menggambar kolaboratif berbasis P2P memerlukan sebuah pendekatan multidisiplin yang mencakup analisis literatur, penggunaan teknologi seperti *socket programming*, dan integrasi modul antarmuka pengguna seperti Tkinter. Solusi ini berupaya menghasilkan aplikasi yang responsif dan efisien yang memberikan pengalaman pengguna sebaik mungkin. Saran peneliti bagi pembaca, ataupun peneliti lain yang memiliki tujuan yang sama, yaitu mengimplementasi arsitektur *Peer-to-Peer*, untuk meningkatkan kemampuan menampung *peer* dalam jumlah yang banyak, membuat antarmuka yang lebih interaktif, mengoptimisasi fungsi *broadcast* dan untuk membuat fitur untuk mempermudah *discovery peer* lain.

DAFTAR PUSTAKA

- [1] F. O. Ehiagwina, N. A. Iromini, I. S. Olatinwo, K. Raheem, and K. Mustapha, "A State-of-the-Art Survey of *Peer-to-Peer* Networks: Research Directions, Applications and Challenges," *Journal of Engineering Research and Sciences*, vol. 1, no. 1, pp. 19-38, Feb. 2022, doi: 10.55708/js0101003.
- [2] N. Bhagatkar, K. Dolas, and R. K. Ghosh, "An Integrated P2P Framework for E-Learning," Mar. 2019.
- [3] M. Bauwens, V. Kostakis, and A. Pazaitis, *Peer to Peer: The Commons Manifesto*. University of Westminster Press, 2019. doi: 10.16997/book33.

- [4] Augustine John, Cruciani Antonio, and Gillani Altaf Iqra, "Maintaining Distributed Data Structures in Dynamic Peer-to-Peer Networks," *Distributed, Parallel, and Cluster Computing*, 2024.
- [5] Y. Faodiansyah, K. Amron, and E. S. Pramukantoro, "Analisis dan Perbandingan Performansi File Sharing Peer-to-Peer Menggunakan Framework JXTA dan Gnutella," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 10, pp. 3771-3778, Feb. 2018, [Online]. Available: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/2732>
- [6] G. A. Luhkito, A. Kusyanti, and R. A. Siregar, "Implementasi Blockchain pada Peer-to-Peer Transaction menggunakan Algoritma U-Quark," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 4, pp. 1267-1272, Mar. 2021, [Online]. Available: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/8807>
- [7] Rawat Ajay, "A Review on Python Programming," *International Journal of Research in Engineering, Science and Management*, vol. 3, no. 12, Dec. 2020.
- [8] Manalu Dudes, Saragih Surbakti Reagan, and Sitorus Sam Putra Peniel, *PEMPROGRAMAN PYTHON DENGAN GUI TKINTER*. Borneo Novelty Publishing, 2024.
- [9] G. J. Sumual, B. Pinontoan, and L. A. Latumakulita, "GUI Application to Setup Simple Graph on the Plane using Tkinter of Python," *d\ 'Cartesian: Jurnal Matematika dan Aplikasi*, vol. 10, no. 1, pp. 8-14, Jul. 2021, doi: 10.35799/dc.10.1.2021.32138.
- [10] Made Pradnyana Ambara, Pande Ketut Widiartana, and Yohanes Priyo Atmojo, "Implementasi Socket Programming Sebagai Media Sinkronisasi Database Terdistribusi dengan Teknik Multi Master Replication," *Jurnal Sistem dan Informatika (JSI)*, vol. 14, no. 2, pp. 113-124, Aug. 2020, doi: 10.30864/jsi.v14i2.271.
- [11] D. Iovescu and C. Tudose, "Real-Time Document Collaboration—System Architecture and Design," *Applied Sciences*, vol. 14, no. 18, p. 8356, Sep. 2024, doi: 10.3390/app14188356.
- [12] Nasrulloh Fazrin Sofhian and Heriyana Tio, "DESAIN DAN PENGEMBANGAN APLIKASI SUMBER BELAJAR KOLABORATIF MELALUI KNOWLEDGE MANAGEMENT SYSTEM UNTUK PERGURUAN TINGGI," *NUANSA INFORMATIKA : JURNAL TEKNOLOGY DAN INFORMASI*, vol. 17, no. 1, 2023.
- [13] D. Yuliana, A. Baijuri, A. A. Suparto, S. Seituni, and S. Syukria, "PEMANFAATAN APLIKASI CANVA SEBAGAI MEDIA VIDEO PEMBELAJARAN KREATIF, INOVATIF, DAN KOLABORATIF," *Jurnal Pendidikan Teknologi Informasi (JUKANTI)*, vol. 6, no. 2, pp. 247-257, Nov. 2023, doi: 10.37792/jukanti.v6i2.1025.
- [14] J. Durbin and J. Brooks, "Peer-to-Peer, Permutable Models for DNS," *Journal of Computer Science and Software Engineering*, vol. 10, no. 2, 2018.
- [15] A. N. Tama, S. Nugroho, and H. K. Wardana, "Implementasi Algoritma Gosip Dalam Jaringan Peer-to-Peer," *Techné : Jurnal Ilmiah Elektroteknika*, vol. 17, no. 01, pp. 7-11, Apr. 2018, doi: 10.31358/techne.v17i01.153.