

1. Pendahuluan

1.1 Latar belakang

Sharing file yang berisikan audio, video, data, atau apapun dalam format digital sudah merupakan hal yang biasa di dalam aplikasi jaringan. Untuk dapat melakukan *file sharing*, maka harus terdapat *resource* yang di-*share* pada satu atau beberapa komputer. Pada awalnya perkembangannya, hanya menggunakan satu komputer untuk menyimpan semua *file* atau *resource* dan biasa disebut *file server*. Seiring dengan meningkatnya kebutuhan akan file, maka *resource* yang disimpan di *server* juga bertambah besar sehingga diperlukan pengaturan untuk resource tersebut. Jika *client* yang mengakses server semakin banyak, maka dapat terjadi *bottle neck*. Selain masalah tersebut, file server memiliki kelemahan *single point of failure*, yaitu jika server rusak atau *down* maka semua client tidak dapat melakukan aktivitas yang berhubungan dengan file sharing.

Terdapat solusi untuk mengatasi masalah di atas, yaitu dengan menyimpan resource di beberapa komputer. Contohnya adalah peer-to-peer file sharing, yaitu teknologi file sharing di mana setiap node memiliki fungsi dan kedudukan yang sama atau tanpa central server. Setiap *client* di dalam *P2P network* harus memiliki *P2P software*. Jika suatu *host* akan melakukan *file sharing*, maka *host* tersebut harus menjalankan *P2P software* untuk dapat bergabung di dalam jaringan P2P. *Host* yang telah bergabung di jaringan P2P, yang disebut *client*, harus mempunyai *resource* yang akan di-*share* dengan cara membuka direktorinya. Contoh aplikasinya adalah Napster, Kazaa, Gnutella, Bittorent, dll.

Permasalahan dari *P2P file sharing* adalah apabila suatu host tidak memiliki *P2P software*, maka *host* tersebut tidak dapat mengakses *file sharing*. Solusinya adalah dengan cara membangun *P2P file sharing* dengan *moderated server* yang dapat diakses melalui web oleh *client* yang tidak memiliki *P2P software*. Hal ini yang menjadi dasar pemikiran penulis untuk mencoba membuat aplikasi *P2P network* untuk *file sharing* yang diakses melalui web.

Client cukup *request* file yang akan dicari, kemudian *moderated server* akan mencari file di index file. Jika file yang dimaksud telah ditemukan, maka hasil pencarian akan ditampilkan di halaman web *client*. Selanjutnya *client* dapat *download* file yang diinginkan melalui perantaraan *web server*.

1.2 Perumusan masalah

Permasalahan yang dijadikan objek penelitian tugas akhir ini menitikberatkan pada :

1. Bagaimana mekanisme untuk membangun *P2P network* untuk keperluan *file sharing*.
2. Bagaimana mekanisme untuk melakukan *file searching* dan *file downloading* dalam *P2P network* di atas.

1.3 Tujuan

Tujuan dari penyusunan tugas akhir ini adalah:

- Membangun aplikasi untuk *file sharing* pada P2P network melalui suatu *moderated server*.
- Menganalisa performansi protokol P2P dan aplikasi yang dibuat.

1.4 Metodologi penyelesaian masalah

Masalah yang telah penulis rumuskan di atas akan dipecahkan dengan metode studi literatur terhadap bahasan-bahasan yang berkaitan dengan masalah utama, yakni:

1. Studi literatur mengenai terhadap bahasan-bahasan yang berkaitan dengan masalah utama, antara lain:
 - Konsep dasar dan algoritma P2P.
 - *File searching*
 - *File sharing*.
2. Pendekatan yang digunakan sebagai tindak lanjut dari literature review adalah :
 - a. Menggunakan *centralized Peer-to-Peer* (P2P) sebagai teknologi P2P karena aplikasi yang dibuat menggunakan perantara web server, sehingga satu server tersebut berfungsi sebagai web server dan server di jaringan P2P.
 - b. Menggunakan protokol HTTP untuk transfer file karena protokol ini sudah umum digunakan dan mudah diimplementasikan.
 - c. Menggunakan metode *centralized index and repositories* sebagai metode *file searching* untuk dapat mempercepat pencarian file.
 - d. Aplikasi digunakan pada intranet.
3. Selanjutnya untuk membangun aplikasi, maka penulis menggunakan metode waterfall yang meliputi:
 - a. *Requirements analysis*
Mengumpulkan dan menganalisa apa saja kebutuhan sistem.
 - b. *Design*
Desain digambarkan dengan menggunakan UML.
 - c. *Implementation*
Implementasi dilakukan dengan menggunakan konsep *Object Oriented Programming* dan bahasa Java.
 - d. *Testing*
Pengujian perangkat lunak dan analisa hasil pengujian yang meliputi fungsionalitas perangkat lunak yang terdiri dari *file searching* dan *file sharing*.