

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Dari penelitian yang dilakukan oleh Dwi Putri Irianingsih Nina Henrarini (2014) mengenai “Upload download dan sinkronisasi berkas pada OwnCloud ubuntu 12.04” perbedaan dengan sistem yang akan dibuat adalah penelitian tersebut masih menggunakan sistem monolithic sedangkan sistem yang akan dibuat menerapkan sistem mircroservices dengan docker container.

Dari Agung Purnomo (2015) “Upload download backup dan sinkronisasi file pada Network Attachtmen Storages (NAS) menggunakan FreeNAS 9.2.17” Perbedaan dengan sistem yang akan dibuat adalah penelitian tersebut mengunduh, mengupload dan sinkronisasi semua file, sedangkan sistem yang akan dibuat khusus untuk berkas multimedia, sehingga menawarkan fitur antarmuka ke klien yang lebih menawan karena plex media server akan secara otomatis mendownload berkas metadata dari internet seperti poster film, rating film, dan synopsis dari film yang disimpan pada server.

Penelitian lain juga dilakukan oleh Sutriono (2015) tentang object storage menggunakan owncloud pada ubuntu 12.04 perbedaan dengan sistem yang akan dibuat adalah penelitian tersebut ditujukan untuk manajemen *object storage* yang berarti dapat menampung segala jenis berkas, sedangkan penelitian yang akan dibangun kusus manajemen berkas multimedia yang menitik beratkan pada fitur antar muka yang lebih menarik.

Untuk perbandingan selanjutnya akan disajikan dalam bentuk tabel 2.1

berikut ini :

Tabel 2.1 Data Penelitian Tentang Sistem Manajemen berkas

Peneliti	Tahun	Layanan	Sistem Operasi	Perangkat Keras (Server)	Perangkat Lunak
Dwi Putri Irianingsih Nina Hendrarini, Anang Sularsa	2014	Upload, Download, sinkronisasi data	Ubuntu 12.04	Notebook Processor Intel Core I3 RAM 1 GB HDD 500GB	owncloud
Agung Purnomo	2015	Upload,download, backup, sinkronisasi dan file sharing	FreeNAS 9.2.1.7x64	Processor Intel CORE I3 3.0 Ghz Motherboard Chipset Biostar RAM 8 GB DDR 2 HDD 320GB	owncloud
Sutriono	2015	Upload, download, backup, sinkronisasi dan file sharing	Ubuntu 12.04	Processor Intel Pentium 4 2.4 Ghz Motherboard Chipset Amptron RAM 2 GB DDR2	owncloud
Usulan Peneliti	2017	Multi Media Server Cloud Computing	Ubuntu 14.04	Notebook Intel Core I3 RAM 2 GB	Plex, Docker, Web Browser

2.2 Dasar Teori

2.2.1 SAMBA

Samba adalah himpunan aplikasi yang bertujuan agar komputer dengan sistem operasi Linux, BSD (Berkeley Software Distribution) atau Unix dapat bertindak sebagai file dan print server yang berbasis protokol SMB (Session Message Block). Jaringan yang semacam ini biasa dijumpai pada Windows Workgroup atau Windows NT Domain. Samba juga dilengkapi dengan beberapa program bantu sehingga Sistem Operasi Linux (dan Unix lainnya) bisa mengakses sumber daya yang ada pada jaringan Windows yang telah ada. Bisa dikatakan, Samba adalah jembatan penghubung antara Windows dan Unix. Samba terdiri atas dua program yang berjalan di background: SMBD (Server Message Block Daemon) dan NMBD (NetBIOS Name Block Daemon). Secara singkat dapat disebutkan bahwa SMBD (Server Message Block Daemon) adalah program yang akan menghasilkan proses baru untuk setiap klien yang aktif, sementara NMBD (NetBIOS Name Block Daemon) bertugas mengkonversi nama komputer (NetBIOS) menjadi alamat IP (Internet Protocol) sekaligus juga memantau share yang ada di jaringan. Kerja SMBD (Server Message Block Daemon) sendiri diatur melalui sebuah file konfigurasi. Dengan membuat file konfigurasi yang tepat, Samba dapat dijadikan file server, print server, Domain Controller, dan banyak fungsi lainnya. Keuntungan utama dari layanan samba sebagai domain controller adalah harganya yang relative murah jika dibandingkan dengan windows NT/200x. harga lisensi windows NT/200x sendiri sudah cukup tinggi, belum lagi harga akses per client. Dengan linux dan samba, semuanya dapat diperoleh

dengan nyaris gratis. Namun demikian itu tidak berarti samba adalah perangkat lunak yang murahan. Suatu penelitian membuktikan bahwa kecepatan server samba dua setengah kali lebih cepat dibandingkan windows NT 2003 server, bukti penelitian bisa dilihat di situs ” <http://www.kegel.com/nt-linux-benchmarks.html>”

. (Azikin. A. 2017)

2.2.2 Jaringan Komputer

Network atau jaringan, dalam bidang komputer dapat diartikan sebagai dua atau lebih komputer yang dihubungkan sehingga dapat berhubungan dan dapat berkomunikasi, sehingga akan menimbulkan suatu efisiensi, sentralisasi dan optimasi kerja. Pada jaringan komputer yang dikomunikasikan adalah data, satu komputer dapat berhubungan dengan komputer lain dan saling berkomunikasi (salah satunya bertukar data) tanpa harus membawa disket ke satu komputer ke komputer lainnya seperti yang biasa kita lakukan. Ada beberapa jenis jaringan komputer dilihat dari cara pemrosesan data dan pengaksesannya.

1. **Host-Terminal.** Dimana terdapat sebuah atau lebih server yang dihubungkan dalam suatu dumb terminal. Karena Dumb Terminal hanyalah sebuah monitor yang dihubungkan dengan menggunakan kabel RS-232, maka pemrosesan data dilakukan di dalam server, oleh karena itu maka suatu server haruslah sebuah sistem computer yang memiliki kemampuan pemrosesan data yang tinggi dan penyimpanan data yang sangat besar.

2. **Client - Server.** Dimana sebuah server atau lebih yang dihubungkan dengan beberapa client. Server bertugas menyediakan layanan, bermacam-macam jenis layanan yang dapat diberikan oleh server, misalnya adalah pengaksesan berkas, peripheral, database, dan lain sebagainya. Sedangkan client adalah sebuah terminal yang menggunakan layanan tersebut. Perbedaannya dengan hubungan dumb terminal, sebuah terminal client melakukan pemrosesan data di terminalnya sendiri dan hal itu menyebabkan spesifikasi dari server tidaklah harus memiliki performansi yang tinggi, dan kapasitas penyimpanan data yang besar karena semua pemrosesan data yang merupakan permintaan dari client dilakukan di terminal client.
3. **Peer to Peer.** Dimana terdapat beberapa terminal komputer yang dihubungkan dengan media kabel. Secara prinsip, hubungan peer to peer ini adalah bahwa setiap komputer dapat berfungsi sebagai server (penyedia layanan) dan client, keduanya dapat difungsikan dalam suatu waktu yang bersamaan. (Sofana. I . 2010)

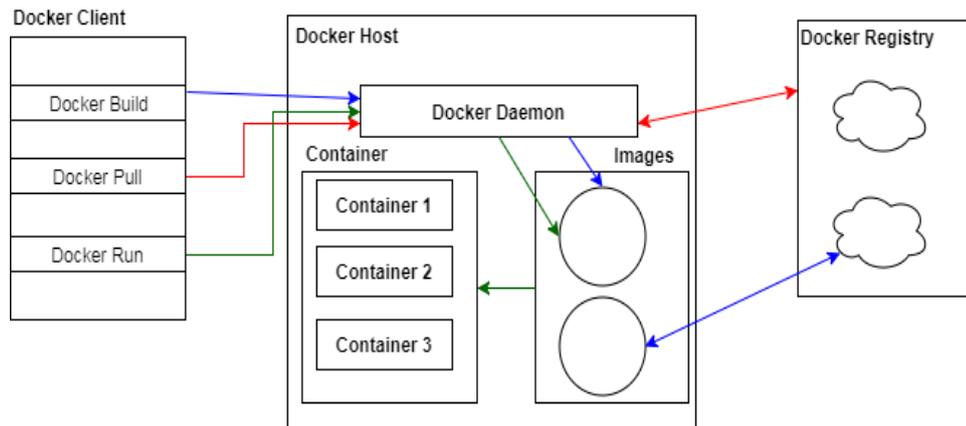
2.2.3 Cloud Computing

Menurut Purbo (2012:1), "*Cloud Computing* adalah sebuah model komputasi / *computing*, dimana sumber daya seperti *processor/computing power*, *storage*, *network*, dan *software* menjadi abstrak dan diberikan sebagai layanan di jaringan/*internet* menggunakan pola akses *remote*."

2.2.4 Docker

Docker adalah sebuah projek open-source yang mengotomasi pekerjaan sebuah aplikasi di dalam software container, demikian itu mendukung lapisan tambahan dari abstraksi dan otomasi sebuah operating system-level virtualization dalam linux. Docker menggunakan fitur isolasi sumber daya dari kernel linux seperti cgroups dan namespaces untuk mengizinkan container secara independen untuk berjalan di dalam satu contoh linux, menghindari overhead ketika menjalankan mesin virtual. Namespaces Linux kernel sepenuhnya mengisolasi lingkungan operasi sebuah proses aplikasi, termasuk proses trees, network, user IDs dan mounted file systems, sementara cgroups mendukung isolasi sumber daya, termasuk CPU, memory, block I/O dan jaringan. Docker berisi libcontainer library sebagai acuan implementasi untuk container, dan dibangun di atas libvirt, LXC (Linux Container) dan system-nspawn, yang mendukung antar muka pada dukungan fasilitas dengan Linux kernel. (__, <https://docs.docker.com>)

Menurut penelitian industry analyst firm 451, “Docker adalah sebuah tool yang bisa mem-package sebuah aplikasi bersama dependency-nya dalam sebuah virtual container dan dapat berjalan di server Linux apapun. Hal ini membantu fleksibilitas dan portabilitas dimana aplikasi bisa dijalankan, apakah di public/private cloud, bare metal, dan lain-lain.



Gambar 2.1 Arsitektur docker

Gambar 2.1 diatas memperlihatkan arsitektur klien server docker, mengacu pada anak panah biru menunjukkan proses docker build yang merequest ke docker daemon kemudian docker daemon akan mengecek apakah *images* yang diminta sudah ada di mesin host, jika belum maka akan mencari ke docker hub (<https://hub.docker.com/>) jika *images* yang dituju ada pada docker hub maka akan mengunduh *images* tersebut kemudian menyimpannya di host. Anak panah merah menunjukkan proses docker pull, yaitu docker client meminta request ke docker daemon kemudian docker daemon meneruskan langsung ke docker hub.

Mengacu pada anak panah hijau menunjukkan proses docker run yaitu adalah proses pembuatan kontainer, pertama docker client akan meminta *images* yang akan digunakan untuk base container kepada docker daemon, jika *images* yang diminta sudah ada pada host maka akan menggunakan *images* tersebut untuk base containernya, jika tidak, maka akan mengarah ke docker hub, lalu menyimpan *images* tersebut di host dan membuat container menggunakan *images* tersebut.

- **Kelebihan Docker dari Mesin Virtualisasi Traditional**

Metode yang biasa digunakan untuk mendistribusikan aplikasi dan mengisolasi eksekusinya adalah menggunakan virtual mesin atau VMs. Format virtual mesin

biasanya adalah VMWare's vmdk, Oracle Virtualbox's vdi, dan Amazon EC2's ami. Secara teori format tersebut akan memungkinkan setiap developer secara otomatis mem-package aplikasinya ke dalam sebuah "mesin" untuk kemudahan distribusi dan deployment. Namun praktiknya, hal itu hampir tidak terjadi karena beberapa alasan :

1. Size : Mesin virtual sangat besar yang membuatnya tidak praktis untuk disimpan dan dipindahkan.
2. Performance : Menjalankan mesin virtual mengkonsumsi CPU dan memory secara signifikan, yang membuatnya tidak praktis di banyak scenario, contohnya pengembangan aplikasi lokal multi-tier, dan deployment CPU dan aplikasi intensif memory skala besar dalam jumlah mesin yang banyak.
3. Portability : Membandingkan lingkungan virtual mesin tidak berjalan baik satu sama lain, walaupun tool konversi ada, tapi mereka terbatas dan menambah lebih banyak overhead
4. Hardware-centric : Virtual mesin didesain dengan operator mesin, bukan pengembang software. Hasilnya, mereka menawarkan peralatan yang sangat terbatas untuk apa yang sering dibutuhkan oleh pengembang; membangun, mengetest dan menjalankan software-nya. Sebagai contoh, penawaran virtual mesin tanpa fasilitas untuk versioning aplikasi, monitoring, konfigurasi, logging atau layanan discovery.

Sebaliknya, Docker memakai metode sandboxing berbeda, yang dikenal sebagai containerization. Tidak seperti virtualisasi tradisional, containerization mengambil tempat pada level kernel. Banyak sistem kernel

sistem operasi modern saat ini yang mendukung kebutuhan sederhana untuk containerization, termasuk Linux dengan OpenVZ, vserver, LXC, Solaris dengan Zones dan FreeBSD dengan Jails.

Docker dibangun di atas primitif low-level tersebut untuk menawarkan developer sebuah format portable dan lingkungan runtime yang memecahkan 4 masalah di atas. Docker container sangat kecil (dan perpindahannya bisa dioptimalkan dengan layers), Docker pada dasarnya memiliki nol overhead memory dan CPU, Docker benar-benar portable dan didisain dari dasar dengan disain application-centric. Virtual mesin baik digunakan untuk mengalokasikan sumber daya hardware, container beroperasi pada level proses, yang membuatnya sangat ringan dan sempurna sebagai unit pendistribusi software. Bagian terbaik adalah, karena Docker beroperasi pada level OS, maka Docker masih bisa berjalan di dalam mesin virtual.

1. Portable Deployment Across Machines

Docker menentukan format untuk mem-bundle sebuah aplikasi beserta semua dependency-nya kedalam sebuah objek yang bisa didistribusikan ke mesin yang menjalankan Docker (Docker-enable machine), dan mengeksekusinya dengan jaminan bahwa lingkungan tempat mengeksekusi aplikasi akan sama. LXC mengimplementasikan proses sandboxing, yang merupakan prasyarat penting untuk deployment yang portable, namun tidak cukup hanya deployment yang portable. Jika kita mengirimkan sebuah salinan aplikasi yang diinstall pada sebuah LXC dengan konfigurasi custom, itu hampir tidak akan berjalan pada mesin yang lain seperti berjalan pada mesin sebelumnya, karena itu terikat dengan

mesin sebelumnya dimana aplikasi itu dibangun dengan konfigurasi tertentu; jaringan, penyimpanan, logging, distro, etc. Docker mempunyai sebuah abstraksi untuk pengaturan mesin secara spesifik, jadi Docker container yang sama persis bisa berjalan, tanpa harus dirubah pada banyak mesin yang berbeda, dengan banyak konfigurasi yang berbeda.

2. Versioning

Docker mempunyai kemampuan untuk melacak urutan versi dari sebuah container, memeriksa diff diantara versi, membuat versi baru, *rolling back*, dan lain-lain. Fitur *history* juga terdapat pada container, bagaimana sebuah container dirakit dan oleh siapa, jadi kita mempunyai kemampuan melacak secara penuh dari server produksi sepanjang perjalanan kembali ke *upstream* developer. Docker juga mengimplementasikan incremental upload dan download, mirip seperti git pull, jadi versi baru sebuah container bisa ditransfer dengan hanya mengirimkan diffs.

3. Component re-use

Semua container bisa digunakan sebagai “*base image*” untuk membuat komponen yang lebih spesifik. Hal ini bisa dilakukan manual atau sebagai sebuah pembangunan secara otomatis. Contohnya, kita bisa menyiapkan lingkungan Python yang ideal, dan menggunakannya sebagai dasar dari 10 aplikasi yang berbeda. Atau konfigurasi *Postgresql* untuk dapat digunakan kembali (*re-use*) untuk semua project di masa yang akan datang, dan seterusnya.

4. Sharing

Docker mempunyai akses ke sebuah public registry dimana ribuan orang telah meng-upload *images* yang berbeda, dari Redis, CouchDB, untuk beberapa distro Linux. Registry juga memiliki sebuah “*standard library*” untuk container yang bermanfaat dan dipantau secara resmi oleh team Docker. Registry itu sendiri bersifat *open-source*, jadi setiap orang bisa membangun *registry* mereka sendiri untuk menyimpan dan mendistribusikan ke *private container*, untuk deployment server internal contohnya.

5. Tool Ecosystem

Docker memiliki sebuah API untuk mengotomasi dan meng-custom pembuatan dan *deployment* sebuah container. Ada banyak tool yang berintegrasi dengan Docker untuk megembangkan kemampuannya. Deployment PaaS (*Dokku, Deis, Flynn*), *multi-node orchestration* (Maestro, Salt, Mesos, Openstack Nova), management dashboards (docker-ui, Openstack Horizon, Shipyard, portainer.io), manajemen konfigurasi (Chef, Puppet), continuous integration (Jenkins, Strider, Travis) dan lain-lain. Docker dengan cepat menetapkan diri sebagai standar untuk tool berbasis container. (Protnoy Matthew, 2012)

2.2.5 Docker Compose

Docker Compose adalah sebuah program untuk mendefinisikan dan menjalankan lebih dari satu kontainer, berdasarkan rules, atau aturan yang di deklarasikan menggunakan yaml, dengan nama *default* docker-compose.yml. sehingga dengan satu perintah dapat menjalankan beberapa service / secara

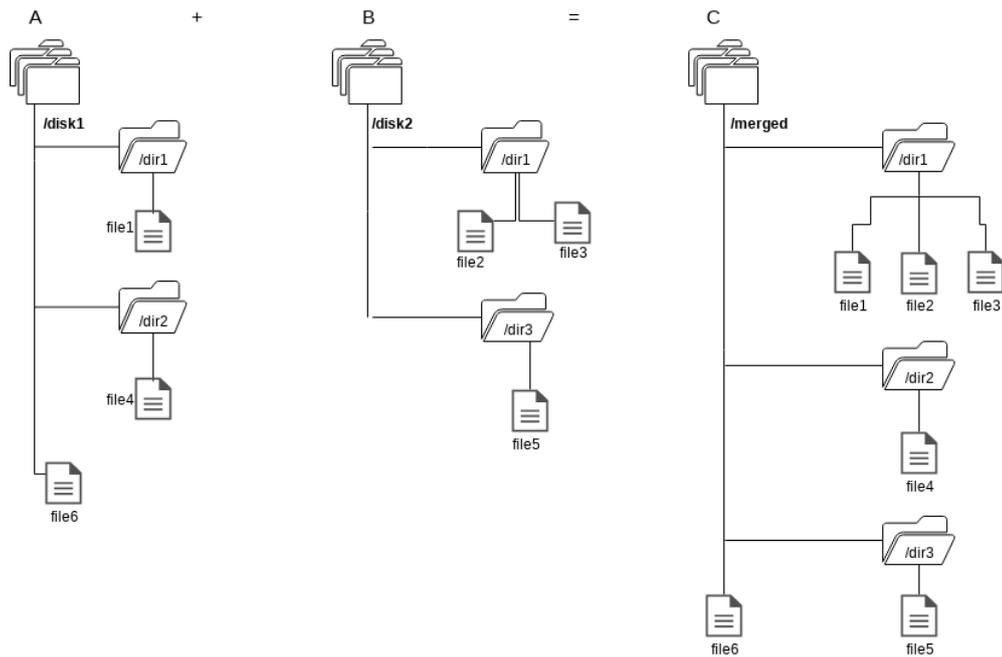
bersamaan. Misalnya, untuk menjalankan wordpress container, container tersebut tidak dapat berjalan hanya dengan satu container wordpress saja, karena wordpress membutuhkan webserver untuk menerima request dari klien dan juga database untuk menyimpan data, maka dalam kasus ini dibutuhkan container database dan webserver dengan total 3 container, untuk mempermudah pengelolaan container ini, docker menyediakana aplikasi docker-compose.

(__, <https://docs.docker.com/compose>)

2.2.6 MergerFS

Mergerfs adalah tool berbasis *union filesystem* yang digunakan untuk kebutuhan manajemen penyimpanan untuk menggabungkan beberapa *storages device* (seperti *hardisk* atau *flashdrive*) dengan berbagai macam kapasitas dan jenis yang berbeda dengan cara menggabungkan beberapa *mountpoint* menjadi satu *mountpoint*. (__, <https://github.com/trapexit/mergerfs>)

Untuk penjelasan lebih detail mengenai cara kerja mergerfs dapat dilihat pada gambar 2.2 berikut:



Gambar 2.2 MergerFS

Penjelasan gambar diatas adalah terdapat 2 *mountpoint* yaitu `/disk1` dan `/disk2`, cara kerja mergerfs adalah menggabungkan 2 *mountpoint* tersebut menjadi satu *mountpoint* yaitu `/merged`, dengan hasil isi dari directory dan berkas pada `/disk1` dan `/disk2` dijadikan 1, dan kapasitas `/merged` sendiri merupakan hasil perpaduan kapasitas `/disk1` dan `/disk2`. Misalnya `/disk1` berkapasitas 80GB dan `/disk2` berkapasitas 320GB maka `/merged` memiliki kapasitas $320\text{GB} + 80\text{GB} = 400\text{GB}$

2.2.7 Scalability

Scalability atau skalabilitas adalah kemampuan sebuah sistem untuk menangani pertumbuhan jumlah data tanpa memberikan dampak pada sistem yang telah berjalan, dengan definisi tersebut maka sistem ini nantinya akan mengimplementasikan fitur dari MergerFS yang dapat mengenali dan mengelola berbagai jenis media penyimpanan seperti hardisk external ataupun USB Flash

Drive, dan menggabungkan penyimpanan baru tersebut ke *mountpoint* tempat dimana berkas media disimpan, sehingga ukuran penyimpanan pada sistem akan bertambah sesuai dengan penambahan kapasitas yang ditambahkan.

2.2.8 PLEX

Plex adalah aplikasi media center dan media server paling populer saat ini. Dengan menggunakan Plex kita bisa mengakses berbagai file media seperti musik, video, dan foto dari satu perangkat ke perangkat lainnya. Aplikasi Plex untuk desktop bisa kita dapatkan secara gratis dan tersedia untuk Windows, Mac, maupun Linux. Salah satu kelebihan Plex adalah adanya fitur transcoding secara real-time. Dengan fitur ini kita tidak perlu memusingkan apakah perangkat kita bisa memutar file media tersebut atau tidak. Semua file media akan di transcoding secara real-time agar bisa dinikmati di seluruh perangkat yang kita miliki. Kelebihan lainnya adalah adanya fitur MyPlex yang bisa kita gunakan untuk menghubungkan file media di rumah kita agar bisa diakses saat kita sedang di luar rumah

Berikut adalah beberapa kelebihan plex media server dibandingkan dengan provider streaming lain :

- Mobile sync adalah fitur yang memungkinkan pengguna mobile untuk menonton media secara offline

- Parental control adalah salah satu fitur yang mengontrol content yang ada pada plex media server yang membatasi content dewasa dan content kekerasan
- Music extras yang merupakan fitur untuk men-*sync* musik dari internet yang berupa lyrics, album art, artis bios, dan album extras.
- Cloud media sync adalah fitur untuk *sync account* cloud seperti dropbox, google drive dan provider cloud storage lain.

Tv streaming yang merupakan streaming tv secara online dengan menggunakan plex media server

“[https://support.plex.tv/articles/200288286/what is plex](https://support.plex.tv/articles/200288286/what-is-plex)”

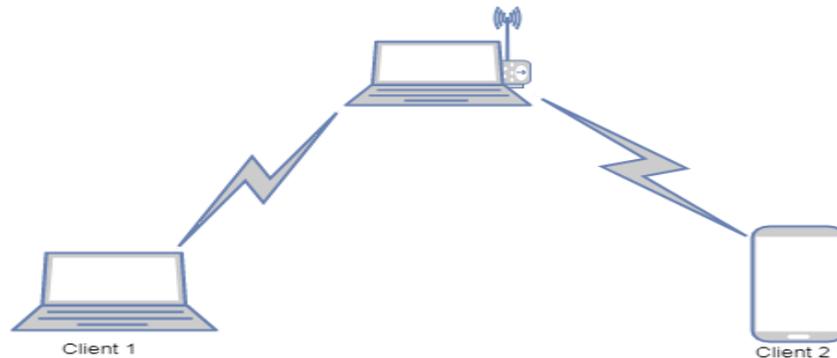
(support.plex.tv 2019)

2.2.9 X-plore

X-plore File Manager adalah sebuah peralatan pengaturan berkas yang memungkinkan Anda untuk menjelajahi semua folder Anda di Android. Anda bisa menyalin, merekatkan dan memindahkan berkas apa pun, dan umumnya mengelola seluruh konten yang ada di memori ponsel atau tablet Anda. Dengan X-plore File Manager, pengguna tidak hanya bisa menjelajahi folder, berkas, atau berkas terkompresi (misalnya dalam format ZIP, 7ZIP, RAR), tapi juga berkas dalam format APK, mirip halnya dengan konten ZIP.

2.2.10 Struktur Jaringan

Pada bagian struktur jaringan ini menggambarkan relasi antara klien dan server, untuk membuktikan bahwa aplikasi berbasis web ini bersifat *cross platform* maka klien menggunakan *smartphone android* dan laptop atau PC



Gambar 2.3 Struktur Jaringan

Pada Gambar 2.3 menjelaskan mengenai relasi antara klien dan server melalui jaringan nirkabel, klien mengakses alamat ip dan port server pada web browser, kemudian server merespond dengan antarmuka *login page* atau jika klien sudah login sebelumnya maka server akan langsung merespond dengan menampilkan antarmuka berkas multimedia yang tersedia di server.