SISTEM MOBILE CLOUD STORAGE DAN DNS AD-BLOCKER UNTUK PERLINDUNGAN PRIVASI DATA PRIBADI

Disusun Oleh:

N a m a : Hilman Maulana

NIM : 15523184

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM INDONESIA

2020

HALAMAN PENGESAHAN DOSEN PEMBIMBING

SISTEM MOBILE CLOUD STORAGE DAN DNS AD-BLOCKER UNTUK PERLINDUNGAN PRIVASI DATA PRIBADI

TUGAS AKHIR

Disusun Oleh:

N a m a : Hilman Maulana

NIM : 15523184

Yogyakarta, 20 September 2020

Pembimbing,

(Dr. Syarif Hidayat, S.Kom., M.I.T)

HALAMAN PENGESAHAN DOSEN PENGUJI

SISTEM MOBILE CLOUD STORAGE DAN DNS AD-BLOCKER UNTUK PERLINDUNGAN PRIVASI DATA PRIBADI

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 12 Oktober 2020

TIM PENGUJI

Dr. Syarif Hidayat, S.Kom., M.I.T. _____

Anggota 1

Arie Sujarwo, S.Kom., M.I.T (Hons). _____

Anggota 2

Hanson Prihantoro Putro, S.T., M.T. _____

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia

(Dr. Raden Teduh Dirgahayu, S.T., M.Sc.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Hilman Maulana

NIM : 15523184

Tugas akhir dengan judul:

SISTEM MOBILE CLOUD STORAGE DAN DNS AD-BLOCKER UNTUK PERLINDUNGAN PRIVASI DATA PRIBADI

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apa pun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 12 Oktober2020

(Hilman Maulana)

HALAMAN PERSEMBAHAN

Dengan mengucapkan syukur Alhamdulillah, penulis mempersembahkan Tugas Akhir ini kepada:

1. Bapak Muslim Mukhdar dan Ibunda Aprida atas dukungannya selama ini dan senantiasa memanjatkan doanya yang tak henti-hentinya sehingga penulis dapat menyelesaikan Tugas Akhir ini.
2. Adik Nudya Maulana yang selalu memberikan dukungan agar Tugas Akhir ini dapat terselesaikan.
3. Keluarga besar Ibu dan Bapak yang telah mendukung dan mendoakan hingga Tugas Akhir ini dapat terselesaikan.
4. Teman belajar, diskusi dan teman dalam segala hal sejak kecil Fengkie Junis yang telah memberikan ilmunya serta motivasinya selama ini.
5. Teman-teman dari Tembilahan di Yogyakarta Herdy, Sofyan, Novri, Imel dan seluruh Keluarga Tembilahan Gemilang Yogyakarta yang sudah membuat daerah perantauan ini serasa di rumah.
6. Faishal, Tahmid, Icung, Mbah, Sopi, Ula, Fadhil, Savana dan seluruh teman-teman Metamorf yang sudah banyak membantu selama masa perkuliahan sampai selesainya Tugas Akhir ini.
7. Teman-teman pengurus HMTF Nurul, Harits, Nayya, Afifah, Luwak, Mas Bambang, Ajoy, Adrian, Irfan dan semua anggota HMTF yang sudah sangat banyak membantu selama ini.
8. Mas Fitra, Mba Nelly, Pipiet, Dira, Tya, Harry, Evan dan teman-teman KOSMIK lainnya yang sudah membantu belajar dalam komunitas.
9. Lur, Palek, Hanif, Madi, Tapir, Quddus dan teman-teman Luput *Adventure* yang sudah menemani perjalanan-perjalanan di masa kuliah.
10. Tim *DevOps* Infosys Mas Hanif, Danu, Mas Rahmad, Mas Indra yang sudah memberi banyak ilmu yang membantu memudahkan pengerjaan Tugas Akhir ini.
11. Alm. Pak Maryanto, Ibu Nang, Pak Kusno yang sudah banyak membantu selama tinggal di Yogyakarta.
12. Seluruh keluarga besar Informatika UII yang sudah membantu selama masa perkuliahan hingga saat ini.

13. Kepada semua pihak yang tidak dapat penulis sebutkan satu persatu yang telah membantu dalam pengerjaan tugas akhir ini. Terima kasih atas dukungan dan doanya.
14. Terakhir terima kasih kepada Hilman Maulana yang sudah melewati berbagai macam proses hingga akhirnya dapat menyelesaikan Tugas Akhir ini.

HALAMAN MOTO

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

Assalamu'alaikum Warahmatullahi Wabarakatuh

Puji syukur penulis panjatkan atas kehadirat Allah Subhanhu Wa Ta'alla yang telah melimpahkan rahmat, hidayah, dan karunia-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul "Sistem Mobile Cloud Storage Dan Dns Ad-Blocker Untuk Perlindungan Privasi Data Pribadi.". Tak lupa shalawat dan salam kami haturkan kepada junjungan kita Nabi Muhammad Sallahu Alaihi Wassalam, yang telah membawa kita dari zaman jahiliyah menuju zaman terang benderang.

Tugas Akhir ini dibuat sebagai salah satu syarat yang harus dipenuhi untuk memperoleh gelar Sarjana Strata-1 (S1) di Jurusan Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia. Selain itu tugas akhir ini juga penulis dapat mengimplementasikan ilmu dan pengetahuan yang telah diperoleh selama berada di bangku perkuliahan.

Selama proses pengerjaan tugas akhir ini, penulis menyadari banyak kesulitan yang telah dilalui dan adanya bantuan, dukungan, bimbingan serta doa dari berbagai pihak. Untuk itu dengan segala kerendahan hati izinkanlah penulis menyampaikan rasa terima kasih dan penghargaan tersebut kepada:

1. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc Selaku Ketua Jurusan Infotmatika Fakultas Teknologi Industri Universitas Islam Indonesia.
2. Bapak Dr. Syarif Hidayat, S.Kom., M.I.T selaku Dosen Pembimbing tugas akhir yang tak kenal lelah di dalam memberikan masukkan, motivasi, dan membimbing dalam proses pengerjaan sistem yang berjudul Sistem Mobile Cloud Storage Dan Dns Ad-Blocker Untuk Perlindungan Privasi Data Pribadi..
3. Ibu Chanifah Indah Ratnasari, S.Kom., M.Kom selaku dosen pembimbing akademik penulis yang telah membimbing penulis selama masa perkuliahan di Informatika UII.
4. Kedua orang tua yang selalu mendukung, mendoakan dan selalu percaya kepada anaknya sehingga tugas akhir ini dapat terselesaikan.
5. Seluruh teman-teman yang terlibat mulai dari pengerjaan, sumbang pikiran dan bantuan moral sehingga penulis dapat menyelesaikan tugas akhir ini.

Semoga segala bimbingan, motivasi, dan dukungannya mendapat imbalan dari Allah SWT. Penulis menyadari bahwa laporan Tugas Akhir ini masih banyak kekurangan. oleh karena itu, penulis mengharapkan masukan yang konstruktif dan saran serta kritik yang positif yang dapat membuat laporan ini menjadi lebih bermanfaat bagi semua pihak.

Yogyakarta, 12 Oktober2020

(Hilman Maulana)

SARI

Pekerjaan menggunakan komputer tidak lepas dengan adanya data yang diproses dan disimpan. Dengan banyaknya pekerjaan yang dikerjakan dengan komputer maka banyak pula data yang harus disimpan dan data yang disimpan pun idealnya dapat diakses dari mana saja. Media penyimpanan yang dapat diakses di mana saja yang tersedia saat ini adalah layanan *cloud storage* yang memiliki masalah pada mahalnya biaya yang dibutuhkan untuk berlangganan dan privasi dari data yang disimpan. Masalah privasi juga terdapat pada iklan *website* yang berbahaya dan mengganggu konten utama serta aplikasi *Ad-Blocker* gratis tidak dapat dipercaya dalam menjaga privasi pengguna. Untuk itu dibutuhkan sebuah sistem yang dapat mengakomodasi kebutuhan penyimpanan data dan pemblokiran iklan tertarget yang aman dan memberi keleluasaan pengguna dalam pengaturan penggunaannya. Pada penelitian ini, metodologi yang diusulkan untuk menyelesaikan masalah adalah sistem *cloud server* dan DNS *ad-blocker* untuk perlindungan privasi data pribadi Raspberry Pi. Sistem yang dikembangkan menggunakan mekanisme *backup* RAID 1 *mirroring* pada konfigurasi *disk* yang terpasang. DNS *ad-blokcer* berjalan pada level *network* dimana setiap perangkat yang terkoneksi pada jaringannya otomatis akan menggunakan *ad-blocker* tersebut. Untuk pengaksesan dari internet dilakukan menggunakan VPN untuk keamanan agar data tetap tersimpan di jaringan privat. Sistem yang dikembangkan memberikan keamanan pada penggunanya dengan pengaksesan hanya pada jaringan privat yang tidak dapat diakses bebas di internet. Dalam sistem ini juga diberikan kebebasan dalam konfigurasi sesuai kebutuhan pengguna seperti kapasitas penyimpanan, akses terhadap jaringan privat dan iklan apa saja yang ingin diblokir.

kunci: Raspberry Pi, *cloud storage, ad-blocker, vpn*

GLOSARIUM

Access Point	Perangkat jaringan yang berisi penerima dan pengirim sinyal ke dan *client* atau pengguna
Open Source	Langkah untuk menelusuri kesalahan kode program.
Server	Sebuah sistem komputer yang menyediakan layanan, bauk untuk menyimpan atau mengolah data tertentu dalam sebuah jaringan komputer.
Software	Istilah khusus untuk data yang diformat, dan disimpan secara digital, termasuk program komputer, dokumentasinya, dan berbagai informasi yang bisa dibaca dan ditulis oleh komputer.
Hardware	Semua bagian fisik komputer yang bekerja saling mendukung sehingga komputer dapat berjalan.
Alamat IP	Label numerik yang ditetapkan untuk setiap perangkat yang terhubung ke jaringan komputer yang menggunakan Protokol Internet untuk komunikasi.
Terminal	Jenis khusus *file* perangkat yang mengimplementasikan sejumlah perintah tambahan (ioctls) di luar baca dan tulis.
Cloud	Awan (*cloud*) adalah metafora dari internet, sebagaimana awan yang sering digambarkan di diagram jaringan komputer.
Mini Computer	Mini Computer adalah perangkat computer yang berukuran kecil yang menggunakan konsep *Single Board Computer* atau computer yang seluruh perangkatnya berada dalam satu *board* (papan PCB)
Storage	Storage adalah tempat penyimpanan data digital seperti *Harddisk* atau *flash disk.*

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I
PENDAHULUAN

1.1. Latar Belakang

Setiap pekerjaan yang melibatkan komputer pasti tidak lepas dengan adanya data yang diproses seperti teks, gambar dan suara. Dengan banyaknya pekerjaan menggunakan komputer banyak juga data yang harus diproses dalam pengerjaannya. Di pasaran saat ini sangat banyak ditemukan perangkat penyimpanan tambahan yang sangat beragam. Tapi dalam penggunaan sehari-hari sangat tidak efektif jika harus membawa perangkat setiap saat karena adanya resiko tertinggal, dan hilang. Di pasaran juga sudah banyak layanan *cloud storage* dengan bermacam variasi, yang menjadi masalah adalah harga yang ditawarkan dalam layanannya. Rata-rata harga bulanan untuk *cloud storage* yang paling murah adalah $1.50 per-*Gigabyte* per bulan tanpa manajemen *backup* (Networks, 2015). *Cloud computing* juga memunculkan masalah baru dalam privasi data di mana hal ini adalah aspek yang penting(Shrivastava, 2017). Privasi data menjadi masalah karena dalam layanan *cloud storage* yang ada di pasaran kebanyakan mempunyai kebijakan privasi yang memperbolehkan permintaan data pengguna oleh pihak tertentu seperti pemerintah. Dalam hal ini contohnya Google berdasarkan *Transparency Report Help Center*-nya menyebutkan Google dapat memberikan data pengguna dalam beberapa kondisi seperti permintaan untuk kepentingan kasus administratif yang cakupannya cukup luas. Dalam laporannya Google telah memberikan informasi data pengguna kepada pihak tertentu khususnya di Indonesia pada periode 2014-2019 sebanyak 27 kali (Google, 2019).

Selain itu juga dalam pengaksesan internet pada saat ini terdapat masalah di mana sangat mengganggunya iklan-iklan yang ditampilkan pada halaman *website*. Iklan *online* sudah banyak beredar di internet dan pendapatan dari penayangan menjadi pemasukan dalam model bisnis akses gratis namun terdapat konsekuensi dari iklan *online* tersebut di mana perantara iklan dapat mengakses jutaan data pengguna (Parra-arnau, Rodr, Parra-arnau, & Rodr, 2016).

Permasalahan dari iklan pada *website* adalah jenis dan penempatan iklan yang sangat menggagu. Contohnya pada akses *website* berita saja contohnya kita bisa mendapati iklan yang muncul menutupi konten utama. Dalam praktiknya pada saat ini terdapat banyak pilihan *ad-blocker* yang hadir sebagai ekstensi gratis pada *web browser*. Namun terdapat permasalahan lagi pada beberapa *ad-blocker* yang bisnis modelnya adalah mengumpulkan data dari penggunanya untuk dijual. Masalah juga terjadi karena kepopuleran layanan blok iklan ini menjadi sasaran peretas untuk mengambil keuntungan melalui layanannya seperti yang terjadi

pada ekstensi populer Adblock Plus pada tahun 2019. Seorang peneliti di bidang keamanan menemukan celah keamanan pada Adblock Plus di mana peretas dapat membaca email Google korban dan layanan Google lainnya (Abel, 2019). Pada layanan blok iklan gratis ini juga kita tidak leluasa dalam penyesuaian pengaturan yang ada dan tidak ditampilkan pula catatan iklan apa saja yang diblokir dan yang diperbolehkan ditampilkan.

Dalam permasalahan *cloud storage* dan iklan *online* pada *website* ini dapat dilihat kesamaan permasalahannya adalah masalah privasi dan keleluasaan dalam penggunaan layanan bagi para pengguna. Untuk itu dibutuhkan sebuah sistem yang dapat mengakomodasi kebutuhan penyimpanan data dan pemblokiran iklan tertarget yang aman, dan memberi keleluasaan pengguna dalam pengaturan penggunaannya.

Untuk pengaplikasian sistem ini dibutuhkan *server* yang dapat diakses selama 24 jam. Dalam hal ini digunakan *Mini Computer* sebagai perangkat *server*, *mini computer* yang digunakan yaitu Raspberry Pi yang murah dari segi harga perangkat dan daya yang dibutuhkan, memiliki bentuk yang kecil dan memiliki fitur yang *powerfull* sebagai sebuah *server*. *Mini Computer server* ini berperan sebagai perangkat NAS (*Network Attached Storage)* yang menjadi media penyimpanan data secara daring dan sebagai server untuk sistem *adblock* yang bekerja pada DNS *level* untuk menghilangkan iklan dalam semua halaman *website* yang diakses.

Dalam penyimpanan *file* terdapat risiko terjadinya kerusakan pada perangkat penyimpanan seperti *harddisk*. Dengan adanya risiko tersebut sangat dibutuhkan penanganan terhadap kehilangan data yang terjadi karena terjadi kerusakan pada media penyimpanan. Dengan kekurangan tersebut dapat disolusikan dengan teknologi RAID 1 (*Redundant Array of Independent Disk Level 1)*(M. Nourman Hadi Satrio H, Nugroho Suharto, Martono Dwi Atmadja, 2019). Untuk sistem blokir iklan dan penyimpanan data ini hanya dapat diakses dari dalam jaringan privat seperti jaringan internet di rumah. Untuk itu dibutuhkan VPN(*Virtual Private Network)* yang dapat memungkinkan kita mengakses jaringan privat rumah dari jaringan internet publik. Sehingga layanan penyimpanan data dan blokir iklan dapat digunakan di mana saja.

1.2. Rumusan Masalah

Berdasarkan latar belakang masalah di atas maka dapat di rumuskan, belum adanya sistem *cloud storage* dan blokir iklan berbasis *mini computer* Raspberry Pi yang aman dan dapat memberi keleluasaan bagi pengguna untuk melakukan konfigurasi pada sistem tersebut.

1.3. Batasan Masalah

Batasan masalah diperlukan untuk menghindari meluasnya ruang lingkup yang dibahas dalam penelitian agar tidak menyimpang dari pokok permasalahan. Batasan masalah dalam penelitian ini adalah sebagai berikut:

a. Perangkat yang digunakan adalah Raspberry Pi 3 Model B RAM 1GB.
b. Teknologi RAID yang digunakan adalah RAID1 untuk 2-3 *harddisk*.
c. Perangkat penyimpanan yang digunakan atau *disk* adalah 2 buah *Flash Disk* 16 GB.
d. DNS *ad-blocking* hanya dapat digunakan pada jaringan yang sama.
e. Pengaksesan sistem dari luar jaringan privat hanya dapat dilakukan menggunakan VPN.

1.4. Tujuan Penelitian

Tujuan penelitian ini adalah untuk membuat *Cloud Storage server* yang aman dari sisi pengaksesan dan penyimpanan data serta sistem blokir iklan pada *DNS level* yang aman. Sistem ini dapat diakses pada jaringan privat rumah maupun dalam jaringan internet publik sehingga pengguna dapat mengakses sistem ini kapan pun dan di mana pun.

1.5. Manfaat Penelitian

Penelitian ini bermanfaat untuk membuat sebuah *cloud storage server* dan sistem blokir iklan yang aman dan memberi keleluasaan dalam pengaturan sehingga pengguna tidak perlu mengkhawatirkan biaya layanan yang mahal dan keamanan data pribadi pada *cloud storage* dan layanan blokir iklan yang ada di pasaran.

1.6. Metode Penelitian

Adapun metode penelitian yang digunakan dalam penelitian ini adalah sebagai berikut:

a. Merumuskan masalah penelitian

Pada tahap ini penulis mencari permasalahan yang sebelumnya sudah ada dan akan dikembangkan dengan solusi baru agar dapat lebih bermanfaat.

b. Konsultasi dengan dosen pembimbing untuk pengenalan alat dan sistem yang akan digunakan pada penelitian.

Pada tahap ini penulis berdiskusi dengan dosen pembimbing mengenai topik dan permasalahan yang akan dijadikan penelitian.

c. Menganalisis permasalahan pada sistem yang sudah ada untuk dicari solusinya dalam penelitian.

Pada tahap ini penulis menentukan topik penelitian kemudian mencari solusi atas masalah yang diangkat pada penelitian tersebut.

d. Mempersiapkan kebutuhan alat dan sistem yang digunakan pada penelitian.

Pada tahap ini penulis mempersiapkan kebutuhan alat maupun sistem yang akan digunakan, seperti: Raspberry Pi 3 Model B, *flash disk, power adapter*, docker, pihole, nextcloud dan pivpn.

e. Melakukan perancangan sistem sesuai dengan tujuan penelitian.

Pada tahap ini, penulis mulai membangun rancangan arsitektur sistem yang akan dibuat. Membuat rancangan sistem dan gambaran implementasi pada penelitian.

f. Mengimplementasikan sistem sesuai dengan rancangan.

Pada tahap ini, penulis mulai membangun dan mengimplementasikan rancangan sistem seperti mengatur konfigurasi Raspberry Pi agar dapat diakses pada jaringan dan konfigurasi *disk* pada Raspberry Pi serta konfigurasi kebutuhan sistem lainnya.

g. Melakukan pengujian sistem dengan melakukan percobaan pada sistem yang telah dibuat serta menganalisis kelebihan dan kekurangan.

Pada tahap ini, penulis melakukan pengujian terhadap sistem yang telah dibuat sehingga dapat mengevaluasi apakah sistem dapat berjalan sesuai dengan kebutuhan.

1.7. Sistematika Penulisan

Sistematika penulisan penelitian ini disusun untuk mempermudah dalam memahami tantang penelitian yang dijalankan. Dalam penulisan tugas akhir ini dibagi menjadi lima bab. Berikut merupakan sistematika penulisan tugas akhir:

BAB I PENDAHULUAN

Membahas tentang masalah yang akan dibahas dalam laporan tugas akhir ini, di mana dalam laporan tugas akhir ini akan membahas mengenai sistem *cloud storage* dengan konfigurasi *disk* RAID 1 dan DNS *ad-blocker* untuk perlindungan privasi data pribadi. Bab ini berisi rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian metodologi penelitian dan sistematika penulisan.

BAB II LANDASAN TEORI

Membahas mengenai teori-teori yang dikumpulkan oleh penulis dan digunakan sebagai landasan teori untuk menyelesaikan tugas akhir ini.

BAB III METODOLOGI PENELITIAN

Membahas mengenai metode yang digunakan dalam penyelesaian tugas akhir ini. Metode ini meliputi uraian tentang kebutuhan perangkat keras, kebutuhan perangkat lunak, kebutuhan masukan, kebutuhan keluaran dan perancangan sistem *cloud storage* dan *adblocking.*

BAB IV IMPLEMENTASI DAN PENGUJIAN

Membahas mengenai hasil dan pembahasan dari tugas akhir ini. Implementasi mengenai sistem *cloud storage* dan DNS *ad-blocker* yang telah dibuat berdasarkan bab Metodologi. Implementasi tentang kebutuhan perangkat keras, kebutuhan perangkat lunak, kebutuhan masukan, kebutuhan keluaran dan implementasi sistem. Pada bab ini juga memuat konfigurasi dari sistem yang dibuat.

BAB V KESIMPULAN DAN SARAN

Membahas mengenai rangkuman dari awal hingga akhir dari pengerjaan tugas akhir ini. Selain itu juga memuat saran-saran yang dapat berguna bagi pengembang selanjutnya untuk melanjutkan dan mengembangkan penelitian ini.

BAB II
LANDASAN TEORI

2.1. Tinjauan Pustaka

Pada penelitian (M. Nourman Hadi Satrio H, Nugroho Suharto, Martono Dwi Atmadja, 2019) berjudul "Implementasi Protokol Jaringan Openvpn Dan Teknologi Redundant Array Of Independent Disk Level 1 (Raid-1) Pada File" peneliti melakukan penelitian dengan membuat sistem *File Server* yang dirancang menggunakan perangkat dengan biaya yang terjangkau yaitu Raspberry Pi. Sistem yang dibangun peneliti menggunakan teknologi RAID 1 dan protokol jaringan OpenVPN. Manajemen perangkat RAID yang terpasang di sistem dilakukan menggunakan *software* mdadm dan protokol OpenVPN menggunakan PiVPN. Untuk sistem *File Server* digunakan Nextcloud sebagai layanan penyimpanan data dengan *web server* Apache2 dan MariaDB sebagai *database* dari sistem yang berjalan. Sistem *file server* ditampilkan pada web browser di mana pengguna dapat melakukan *upload* dan *download* file pada sistem. Pada penelitian ini peneliti berfokus kepada kecepatan *upload* dan *download* dari sistem serta *load* perangkat yang dihasilkan selama terjadi aktivitas pada sistem. Berikut beberapa penelitian sebelumnya yang ditunjukkan pada Tabel 2.1

Tabel 2.1 Tinjauan Pustaka

No	Judul	Pencapaian	Saran
1	Securing Network Using Raspberry Pi by Implementing VPN, Pi-Hole, and IPS (VPiSec) (Taib et al., 2020)	Sistem dapat menyamarkan alamat dari pengguna yang mengakses internet menggunakan OpenVPN dan berhasil memblokir iklan yang tampil pada halaman website saat terkoneksi pada jaringan yang sama dengan sistem.	Sistem dapat me-*update* *database* P-Hole secara otomatis untuk data *domain* yang diblokir dan juga merubah konfigurasi *router* agar semua IP yang terdaftar masuk ke dalam konfigurasi sistem.
2	Implementasi Protokol Jaringan	Sistem dapat berjalan dengan perangkat	Sistem dapat menggunakan protokol keamanan yang

	Openvpn Dan Teknologi Redundant Array Of Independent Disk Level 1 (Raid-1) Pada File (M. Nourman Hadi Satrio H, Nugroho Suharto, Martono Dwi Atmadja, 2019)	Raspberry Pi dengan implementasi teknologi RAID 1 dan OpenVPN yang dapat diakses di luar jaringan kampus. Pengaksesan *file server* terbatas pada pengguna tertentu dengan performa yang cukup baik.	lebih baik dalam pengaksesan perangkat, dapat mencoba menggunakan *single board computer* lainnya dan menggunakan perangkat penyimpanan yang lebih besar.
3	Remotely Accessible, Low Power Network Attached Storage Device (Lanka, 2018)	Sistem yang dibuat menunjukkan pengurangan dalam penggunaan daya dibandingkan dengan NAS server yang ada di pasaran.	Pada penelitian ini sistem yang dikembangkan memiliki *write speed* yang lebih kecil dari perangkat yang menjadi perbandingan.
4	Rancang Bangun Personal Cloud Storage Berbasis Raspberry Pi (Suharta, 2018.)	Sistem dapat digunakan untuk pengoperasian aplikasi penyimpanan file melalui jaringan internet dengan konfigurasi yang mudah tanpa terjadi kesalahan ataupun *error* dalam sistem.	Sistem dapat menggunakan aplikasi web server yang lebih optimal, menambahkan sistem keamanan dan menggunakan konfigurasi RAID untuk dapat menggunakan beberapa *harddisk.*

Data pada Tabel 2.1 Tinjauan Pustaka adalah referensi yang digunakan oleh penulis sebagai penunjang dalam pengimplementasian sistem *cloud server* dan DNS *adblock* untuk membuat sistem *cloud storage* yang murah dan aman. Sistem *cloud storage* ini menggunakan perangkat keras Raspberry Pi 3 model B yang terpasang dua *disk* yang terkonfigurasi RAID 1,

sistem juga dapat melakukan *blocking ads* menggunakan Pi-Hole. Sistem nantinya akan berjalan pada jaringan privat rumah yang jika ingin diakses dari jaringan internet luar rumah harus menggunakan *Virtual Private Network* yang juga sudah terkonfigurasi pada perangkat. Pengaksesan menggunakan VPN adalah salah satu instrumen keamanan agar data tidak terbuka bebas di dalam internet. Penerapan VPN ini dapat meminimalisir penyadapan dari sisi pengguna yang sangat lemah (Harsapranata, 2015). Sistem *cloud storage* dan DNS *adblock* ini dapat diakses menggunakan *web browser* untuk akses dan konfigurasinya.

Setelah mempelajari tinjauan pustaka yang terdapat pada Tabel 2.1, maka selanjutnya penulis akan membuat sistem *cloud storage* dengan konfigurasi *disk* RAID 1 dan DNS *ad-blocker* untuk perlindungan privasi data pribadi. Sistem ini bertujuan untuk membuat *Cloud Storage server* yang aman dari sisi pengaksesan dan penyimpanan data serta sistem blokir iklan pada *DNS level* yang aman dan transparan. Sistem ini dapat diakses pada jaringan privat rumah maupun dalam jaringan internet publik sehingga pengguna dapat mengakses sistem ini kapan pun dan dimana pun.

2.2. Dasar Teori

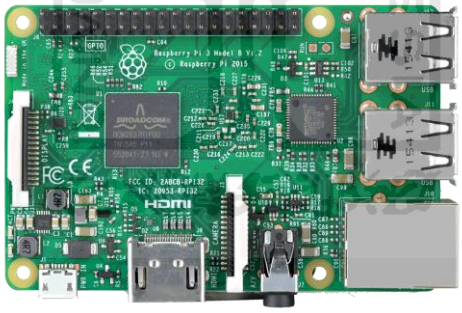
Dasar Teori dalam sistem *cloud storage* dengan konfigurasi *disk* RAID 1 dan DNS *ad-blocker* untuk perlindungan privasi data pribadi menjadi acuan dalam penelitian yang dilakukan. Dasar teori yang digunakan disebutkan dalam penjelasan di bawah ini:

2.2.1. Single Board Computer

Single Board Computer (SBC) adalah sebuah komputer yang terbuat dari sebuah papan sirkuit cetak tunggal dengan *microprocessor*, memori dan perankat *input/output* serta fitur-fitur fungsional komputer lainnya. *Single Board Computer* pada umumnya digunakan untuk tujuan edukasi atau keperluan sebagai *controller* perangkat mesin mekanik. *Single board computer* adalah sebuah terobosan dalam dunia manufaktur *board* komputer yang dibuat menggunakan media yang kecil dengan kemampuan yang serupa dan dengan daya yang lebih kecil (Atmojo, 2015). Dengan desain yang *simple* dan memiliki fitur yang cukup menyerupai komputer *desktop* namun tidak memiliki daya ekspansi perangkat yang banyak dan kekuatan komputasi yang lebih kecil.

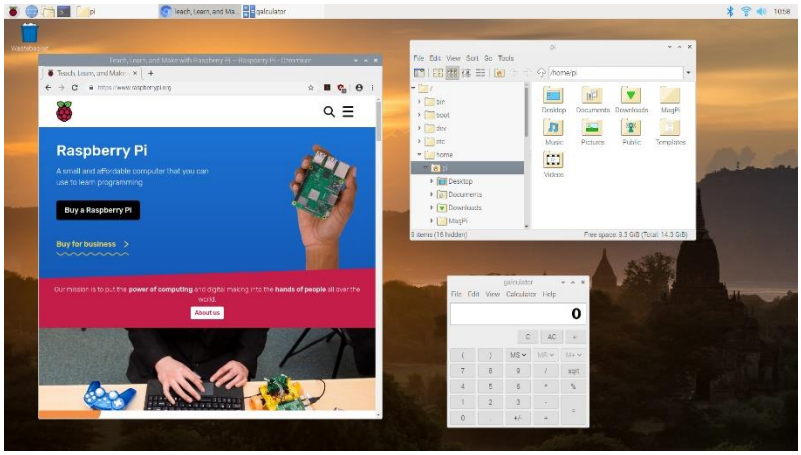
2.2.2. Raspberry Pi

Raspberry Pi adalah salah satu *Single Board Computer* yang dikembangkan salah satu perusahaan komputer dari Inggris yaitu Raspberry Pi Foundation. Pada awalnya tujuan dari Raspberry Pi dibuat adalah untuk meningkatkan keinginan orang untuk mempelajari ilmu komputer (Upton, 2016). Namun, setelah perilisannya Raspberry Pi banyak digunakan untuk kegunaan yang lebih luas seperti pada bidang *robotic* dan pegiat hobi lainnya. Raspberry Pi beberapa generasi, generasi pertama adalah Raspberry Pi 1 Model B yang menggunakan arsitektur ARMv6Z (32-bit) dengan *processor* terpasang adalah ARM1176JZF-S 700 Mhz, menggunakan 256/512 MiB RAM. Sedangkan generasi terakhir yang dirilis saat ini adalah Raspberry Pi 4 Model B dengan 4 *core processor* Cortex-A72 1.5 GHz dengan arsitektur ARMv8-A (64/32-bit) dan pilihan RAM sampai dengan 8 GB. Raspberry Pi berjalan menggunakan sistem operasi yang ter-*install* dalam MicroSD yang terpasang, USB *port* dengan internal USB-hub, 26 pin *General purpose input-output* (GPIO) *connector*, HDMI *port*, RJ45 *port* untuk *networking*, *wifi* dan *bluetooth* serta micro-USB *input* untuk *power* yang terlihat pada Gambar 2.1. Raspberry Pi juga dapat dipasangkan dengan berbagai aksesoris dan modul seperti kamera, layar dan banyak modul yang tersedia di pasaran.



Gambar 2.1 Raspberry Pi

Raspberry Pi dapat dijalankan menggunakan sistem operasi yang disediakan oleh Raspberry Pi Foundation yaitu Raspberry Pi OS, sebuah sistem operasi berbasis Debian(32-bit). Selain itu Raspberry Pi juga dapat dijalankan menggunakan sistem operasi pihak ketiga seperti Ubuntu, Arch Linux ARM dan banyak lagi. Kebanyakan dari sistem operasi yang dapat digunakan pada Raspberry Pi adalah sistem operasi yang berbasis Linux, namun ada juga beberapa sistem operasi yang tidak berbasis Linux yang dapat digunakan seperti FreeBSD dan Windows 10 IoT Core.



Gambar 2.2 Raspberry Pi OS

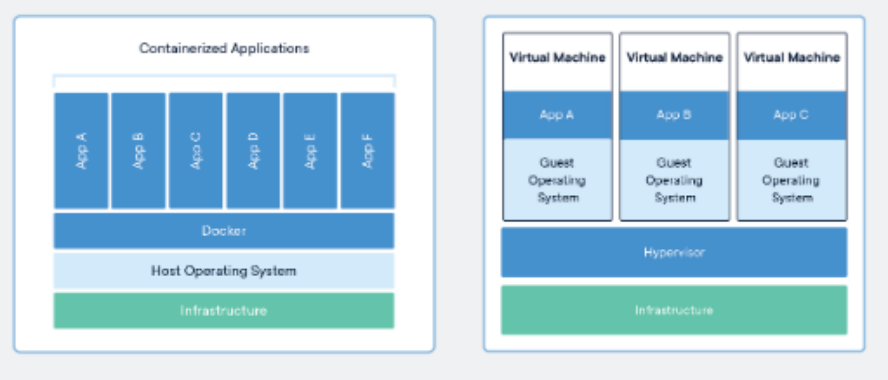
2.2.3. Penyimpanan Data Komputer

Penyimpanan data pada komputer adalah media yang digunakan untuk menyimpan data digital yang terdapat pada komputer. Media penyimpanan data ini dapat memungkinkan komputer untuk membaca, menulis dan mengeksekusi data pada media penyimpanan tersebut. Media penyimpanan data pada komputer ada tiga dan salah satunya adalah media dari *magnetic disk* contohnya adalah *flashdisk. Flashdisk* adalah piranti penyimpanan yang kecil dengan *interface* jenis USB dan berukuran kecil tetapi mempunyai kapasitas penyimpanan yang besar hingga 1 TB.

2.2.4. Docker

Docker adalah perangkat lunak yang memungkinkan penggunanya untuk menjalankan sebuah aplikasi dalam bentuk *container*. *Container* sendiri adalah sebuah *standard unit* dari sebuah perangkat lunak yang menyatukan suatu kode dan semua *dependencies* dari sebuah program dalam satu *environment* yang terpisah dari satu dan yang lain. Docker adalah daemon yang mampu mengelola *container* Linux Sebagai *image* tersendiri (Adiputra, 2015). Docker adalah salah satu *platform container* yang banyak digunakan pada saat ini.

Container memiliki kesamaan dengan teknologi pendahulunya *Virtual Machine* karena sama-sama melakukan isolasi terhadap sebuah sumber daya pada komputer. Namun perbedaannya adalah *container* fungsinya memvirtualisasi sebuah sistem operasi bukan perangkat keras seperti yang dilakukan oleh *Virtual Machine.* Oleh sebab itu *container* jauh lebih ringkas dan efisien. Perbedaan arsitektur *container* dan *Virtual Machine* dapat dilihat pada Gambar 2.2.

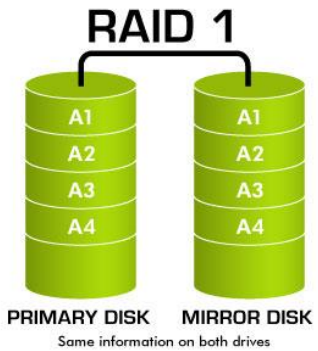


Gambar 2.3 Perbandingan arsitektur *Container* dan *Virtual Machine*

2.2.5. RAID (Redundant Array of Inexpensive Disk)

RAID adalah teknologi virtualisasi penyimpanan data yang menggabungkan beberapa perangkat penyimpanan fisik menjadi satu komponen ataupun lebih untuk kepentingan redundansi data, peningkatan performa atau keduanya sekaligus. Tujuan dari penggunaan teknologi RAID adalah untuk menjaga keandalan data agar tetap terjaga (Santi, Rumani, & Purwanto, 2013). Pada teknologi ini data didistribusikan ke beberapa perangkat berdasarkan konfigurasi yang diinginkan. Dalam RAID konfigurasi dibagi menjadi beberapa level contohnya RAID 0 dan RAID 1 yang dapat dilihat pada Gambar 2.3. Level dalam konfigurasi pada RAID memiliki memiliki tujuan masing-masing seperti keandalan, ketersediaan, performa dan kapasitas.

Dalam implementasinya konfigurasi RAID atau pendistribusian data dalam beberapa perangkat dapat diatur baik menggunakan perangkat keras maupun perangkat lunak. Solusi untuk pengaturan menggunakan perangkat lunak biasanya sudah termasuk dalam bagian dari sistem operasi atau perangkat lunak tambahan, sedangkan solusi pengaturan menggunakan perangkat keras dapat menggunakan perangkat yang khusus dibuat untuk manajemen RAID.



Gambar 2.4 Implementasi RAID 1

2.2.6. MDADM (Multiple Disk and Device Administration)

MDADM adalah Linux *software* RAID *configuration* yaitu sebuah *software* yang digunakan untuk mengatur dan memonitor perangkat RAID. Linux *software* RAID *configuration* dapat membaca semua yang terbaca pada Linux kernel sebagai *block device* seperti sebuah *harddisk* maupun partisinya. MDADM dapat digunakan untuk membuat *array disk* pada konfigurasi RAID. Pada *software* ini dapat digunakan konfigurasi RAID 0 sampai dengan RAID 10.

2.2.7. NextCloud

Nextloud merupakan sebuah *platform open-source* yang dapat digunakan untuk layanan *file hosting*. Pada fungsinya Nextcloud hampir sama seperti Google Drive dan Dropbox, Nextcloud juga mempunyai fungsi sebagai server perangkat lunak *office* seperti Microsoft Office 365. Namun untuk fungsionalitas *office-nya* Nextcloud terbatas pada arsitektur sistem x86/x64 dan tidak mendukung arsitektur ARM yang umumnya digunakan *single board computer*.

Nextcloud adalah *platform* yang bersifat *open-source* yang artinya dapat dengan mudah dan gratis dipasang pada *server* sendiri tanpa ada biaya. Pengguna juga dapat berkontribusi pada sistem yang dibangun. Nextcloud sendiri dibangun menggunakan bahasa pemrograman PHP dan Javascript. Nextcloud dapat diitegrasikan dengan banyak database seperti SQLlite, MariaDB dan PostgreSQL.

Fitur yang ditawarkan oleh Nextcloud sangat banyak dan *platform* ini bersifat modular dimana pengguna dapat menambahkan ekstensi dengan fungsionalitasnya masing-masing sesuai dengan kebutuhan. Pada umumnya fiturnya adalah *upload data, download data*, melihat data pada server dan juga manajemen *user* serta sistem yang sudah terpasang. Namun, juga dapat ditambahkan ekstensi seperti yang sudah disebutkan tadi.

Pada penelitian ini Nextcloud digunakan sebagain *interface cloud storage*. Untuk pengaksesan data seperti *upload* dan *download* dilakukan pada *platform* ini yang dapat dilakukan pada *web browser* atau *mobile client* Nextcloud.

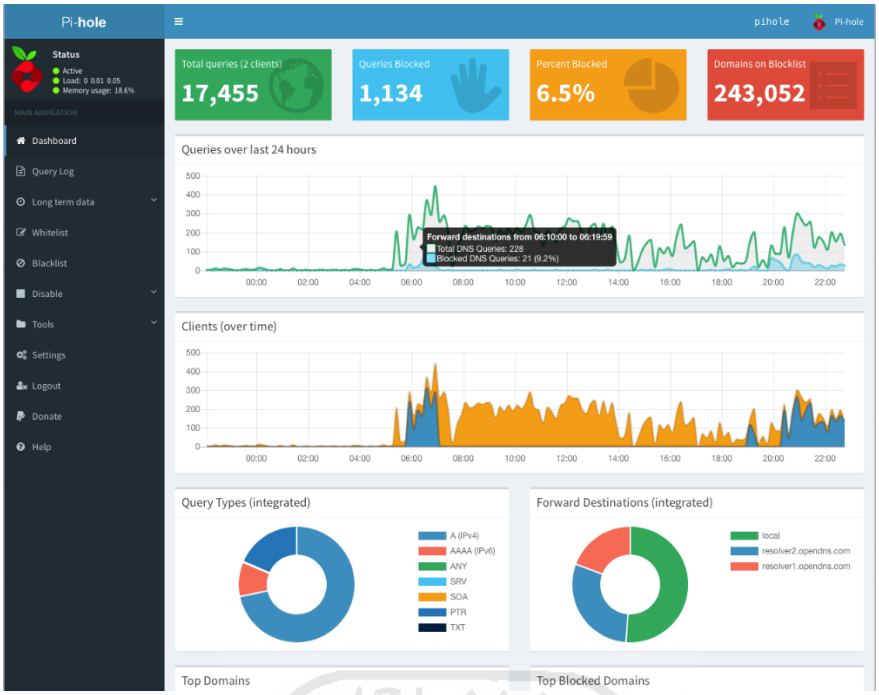
2.2.8. MariaDB

MariaDB merupakan *database* yang dikembangkan oleh komunitas *open-source* yang merupakan proyek yang berasal dari *fork* MySQL *relational database management system* untuk membuat proyek *database* ini tetap gratis dan *open-source* dibawah lisensi GNU *(General Public License)* setelah MySQL diakuisisi oleh Oracle pada tahun 2009. MariaDB bertujuan untuk menjaga kompatibilitas terhadap MySQL sebagai alternatif *database* yang *open-source* tetapi tetap memiliki fitur yang setara. Namun, pada pengembangannya MariaDB mengembangkan fitur yang berbeda atau tidak berpatokan pada MySQL lagi.

2.2.9. Pi-Hole

Pi-Hole merupakan senuah software *open-source* DNS *sinkhole* yang berjalan pada platform berbasis UNIX dan berfungsi pada level *network level* (Rolon & Background, 2019). Fitur utama Pi-Hole adalah pemblokiran iklan melalui DNS *query* yang sudah terdaftar pada *blocklist* yang ada di Pi-Hole. Pi-Hole digunakan pada jaringan privat seperti pada jaringan rumah ataupun jaringan lokal pada komputer. Pi-Hole didesain untuk berjalan pada perangkat *embedded* seperti Raspberry Pi, namun tidak menutup kemungkinan untuk dapat dijalankan pada perangkat komputer biasa.

Pi-Hole didukung dengan tampilan admin yang dapat dilihat pada Gambar 2.3. Pi-Hole dibangun dengan modifikasi dnsmasq yang disebut FTLDNS, cURL, lighttpd dan PHP. Bekerja sebagai DNS server Pi-Hole dapat menggantikan DNS server yang sebelumnya sudah ada seperti yang disediakan oleh ISP. Fitur utama Pi-Hole adalah domain *blocklist* dan *whitelist*, setiap domain yang terdaftar dalam *blocklist* Pi-Hole akan menolak untuk *resolve* domain tersebut ke alamatnya. Karena berjalan pada *network level* juga Pi-Hole dapat memblokir iklan *banner* pada website bahkan iklan bawaan dalam aplikasi android bahkan pada smart TV.

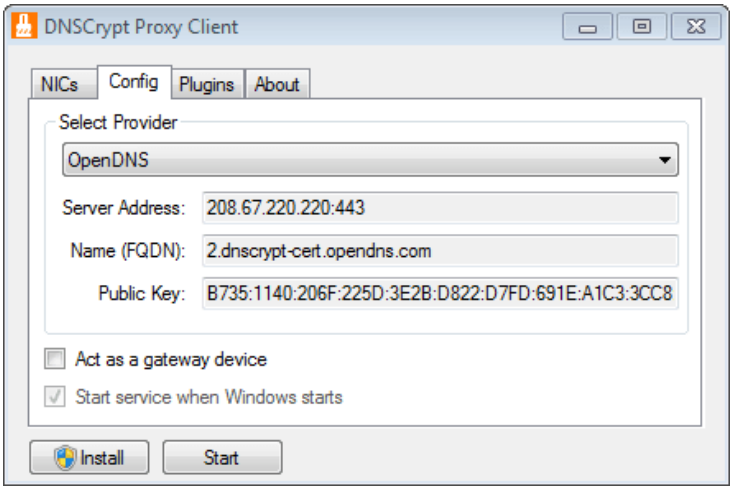


Gambar 2.5 Admin page Pi-Hole

2.2.10. DNSCrypt

DNSCrypt adalah sebuah protokol yang mengotentikasi komunikasi antara DNS *client* dan DNS *resolver*. Protokol ini menggunakan *cryptographic signatures* untuk memverifikasi bahwa respons-respons DNS yang masuk adalah berasal dari DNS *resolver* yang dipilih dan belum dirusak. Dengan metode ini akan menghindarkan pengguna dari DNS *spoofing.*

DNSCrypt sendiri bukanlah produk melainkan sebuah protokol yang siapapun dapat mengimplementasikan. Salah satu contoh implementasinya adalah dnscrypt-proxy. Dnscrypt-proxy mengimplementasikan versi terbaru dari protokol, dapat digunakan dalam banyak platform seperti Windows, Linux, MacOS dan dapat ditambahkan dengan ekstensi lainnya.



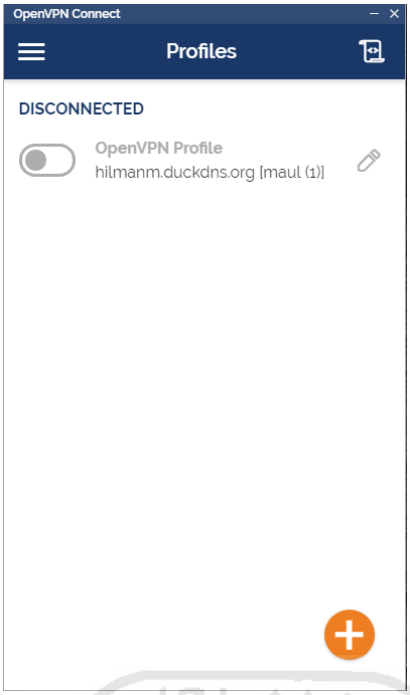
Gambar 2.6 DNSCrypt Client

Pada saat ini banyak organisasi dan individu yang menjalankan DNS *resolver* yang mendukung implementasi protokol DNSCrypt. Jadi pengguna hanya perlu untuk menjalankan DNSCrypt *client* seperti dnscrypt-proxy. Salah satu DNS *server* yang mendukung implementasinya adalah doh.tiar.app, sebuah DNS *resolver* yang mengutamakan *privacy* dan dapat memblokir lebih dari 250 ribu iklan, *ad-tracking*, *malware* dan *domain phising*.

2.2.11. PiVPN

PiVPN adalah sebuah proyek yang VPN server yang terkhusus untuk *platform single board computer* Raspberry Pi dengan menyajikan instalasi yang mudah dan cepat. Sedangkan VPN (*Virtual Private Network*) sendiri adalah teknologi yang memungkinkan pengguna mengakses jaringan privat dari jaringan publik untuk tujuan keamanan. PiVPN sendiri sebenarnya adalah penyederhanaan dalam instalasi OpenVPN yang dibangun oleh komunitas untuk kemudahan pengguna dalam melakukan instalasi VPN server pada perangkat Raspberry Pi.

Pada pengembangannya PiVPN menambahkan fitur tambahan untuk penambahan fungsionalitas dan untuk kemudahan manajemen. PiVPN mendukung dua VPN client yang dapat digunakan yaitu OpenVPN *client* dan WireGuard. Contoh penggunaan OpenVPN *client* dapat dilihat pada Gambar 2.7. OpenVPN *client* dapat digunakan pada berbagai *platform* seperti Android, IOS dan Windows.



Gambar 2.7 OpenVPN Client

BAB III
ANALISI DAN PERANCANGAN SISTEM

3.1. Analisis Kebutuhan

Analisis kebutuhan adalah proses dalam pencarian informasi untuk kebutuhan sistem yang akan dibangun. Informasi yang dikumpulkan antara lain spesifikasi dari perangkat yang digunakan, kebutuhan perangkat lunak yang akan digunakan dan pemodelan sistem saat berjalan. Informasi ini berguna untuk menunjang pengembangan sistem agar sesuai dengan apa yang diinginkan. Dalam penelitian ini penulis menggunakan metode studi Pustaka untuk mengumpulkan informasi-informasi yang diinginkan. Metode studi pustaka adalah metode pengumpulan informasi yang relevan dengan topik atau masalah dari penelitian dengan mengumpulkan informasi dari buku-buku, karya ilmiah, ensiklopedia dan internet (transiskom.com, 2016). Pada tahap ini penulis mengumpulkan informasi tentang sistem yang akan dirancang dalam tugas akhir ini yaitu sistem *cloud storage server* dan DNS *ad-block* untuk perlindungan privasi data pribadi. Informasi yang dikumpulkan mengenai sistem yang akan dibuat adalah informasi perangkat keras dan perangkat lunak yang sesuai dengan sistem yang akan digunakan pada perancangan sistem. Pengumpulan informasi tentang sistem ini dilakukan guna mendapat gambaran tentang kebutuhan dari sistem agar dapat berjalan dan berfungsi sesuai dengan yang diinginkan.

3.1.1. Analisis Kebutuhan Fungsi

Analisis kebutuhan fungsi adalah tahapan yang dilakukan untuk menetapkan fungsi-fungsi yang dapat dilakukan oleh sistem. Analisis kebutuhan fungsi pada sistem sebagai berikut:

a. Mengunduh, mengunggah dan melihat data (teks, gambar, video, dokumen) pada tampilan sistem di *web browser*.
b. Melakukan manajemen pengguna pada sistem *cloud storage.*
c. Melakukan *mirroring* sebagai mekanisme *backup* pada *disk* yang terpasang pada perangkat sistem.
d. Memblokir iklan pada situs web maupun aplikasi pada setiap perangkat yang terhubung pada sistem sesuai dengan *block list* yang sudah ditentukan.
e. Menambah dan mengurangi *block list* pada sistem.

f. Mengakses sistem dari luar jaringan privat rumah menggunakan VPN.
g. Manajemen pengguna VPN pada sistem.

3.1.2. Analisis Kebutuhan *Input*

Pada tahap ini dilakukan analisis terhadap kebutuhan input apa saja yang dibutuhkan dalam pembuatan sistem. Input yang dibutuhkan dalam pengembangan sistem ini adalah sebagai berikut:

a. *Input* jaringan internet. Jaringan internet yang terkoneksi melalui WIFI yang memungkinkan sistem dapat diakses melalui internet. Untuk *input* ini dibutuhkan data SSID dan *password* dari jaringan jaringan yang tersedia agar dapat terhubung ke internet.
b. Data dokumen, teks, gambar atau video. Data ini yang akan disimpan dalam sistem *cloud storage* sebagai media penyimpanan.
c. Data DNS *query*. DNS *query* adalah istilah teknis dalam permintaan pada suatu server. Data ini adalah data pengaksesan domain situs web yang dilakukan pada *web browser* ataupun yang dilakukan di belakang layar oleh aplikasi.

3.1.3. Analisis Kebutuhan *Output*

Pada tahap ini dilakukan analisa kebutuhan *output* informasi apa saja yang akan diberikan kepada pengguna. *Output* informasi yang diberikan pada pengguna dalam sistem ini adalah sebagai berikut:

a. *User interface* untuk sistem *cloud storage* dimana pengguna dapat mengunduh, mengunggah dan melihat data (teks, gambar, video, dokumen) pada tampilan sistem di web browser.
b. Iklan pada halaman situs web sudah tidak terlihat lagi.
c. *Query* apa saja yang diblokir dan diizinkan untuk diteruskan pada jaringan.
d. Informasi grafis tentang pengguna, *query* dan jumlah domain yang diblokir.

3.1.4. Analisis Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak merupakan sekumpulan perangkat lunak yang dibutuhkan untuk pengembangan sistem agar dapat berjalan sesuai dengan yang diinginkan. Analisis kebutuhan perangkat lunak pada sistem ini adalah sebagai berikut:

a. Sistem Operasi Raspberry Pi OS sebagai sistem operasi *server* yang digunakan.
b. Terminal sebagai media konfigurasi sistem.
c. MDADM sebagai *software* untuk konfigurasi RAID pada disk.
d. Docker sebagai sistem *container* yang mengisolasi sistem *cloud storage.*
e. Nextcloud, platform *cloud storage* yang digunakan untuk antarmuka yang diakses pengguna dalam melakukan akses terhadap data pada *server.*
f. MariaDB, *platform database* yang digunakan untuk menyimpan data pada *cloud storage.*
g. Pi-Hole, *software* yang digunakan sebagai *ad-blocker* pada sistem.
h. DNSCrypt, protocol yang digunakan sebagai DNS *resolver* pada Pi-Hole.
i. PiVPN, VPN *server* yang digunakan untuk pengaksesan sistem dari luar jaringan rumah.

3.1.5. Analisis Kebutuhan Perangkat Keras

Dalam pengembangan sistem digunakan beberapa perangkat keras sebagai media di mana sistem akan berjalan. Perangkat keras yang digunakan pada sistem ini adalah sebagai berikut:

a. Raspberry Pi 3 Model B yang digunakan sebagai perangkat *server* pada sistem.
b. *MicroSD card* 16 GB digunakan sebagai *bootable disk* pada Raspberry Pi.
c. 2 *Flash disk* 16GB sebagai media penyimpanan data pada sistem *cloud storage.*
d. Router *Access Point* sebagai penyedia internet untuk server.
e. Kabel USB untuk menghubungkan perangkat Raspberry Pi dengan *power adaptor.*
f. Adaptor power 5V sebagai penyedia daya pada perangkat Raspberry Pi.

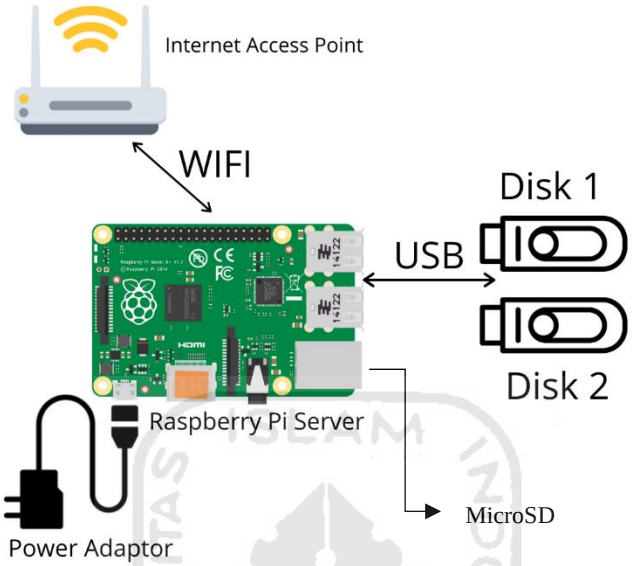
3.2. Metode Perancangan

Dalam pengembangan dan perancangan sistem *cloud storage* dan DNS *adblock* untuk perlindungan privasi data pribadi ini membutuhkan beberapa tahap perancangan. Perancangan yang dilakukan dalam pembangunan sistem ini antara lain adalah perancangan perangkat keras, perancangan perangkat lunak dan perancangan sistem dan jaringan.

3.2.1. Perancangan Perangkat Keras

Dalam tahapan perancangan perangkat keras dilakukan perancangan terhadap perangkat keras yang digunakan dalam pengembangan sistem *cloud storage* dan DNS *adblock*. Perangkat keras yang digunakan dalam pengembangan sistem ini telah disebutkan pada bagian analisis kebutuhan perangkat keras. Pada analisis kebutuhan *input* dibutuhkan koneksi internet yang

membuat sistem dapat diakses melalui jaringan internet. Hubungan tiap perangkat pada sistem ini dapat dilihat pada Gambar 3.1



Gambar 3.1 Diagram Blok Perangkat Keras

Pada Gambar 3.1 terlihat hubungan antar perangkat keras yang sudah disebutkan pada analisis kebutuhan terhubung satu sama lain pada perancangan sistem *cloud storage* dengan konfigurasi *disk* RAID 1dan DNS *adblock* untuk perlindungan privasi data pribadi. Alur kerja antar tiap perangkat keras yang terhubung pada sistem dapat dijelaskan sebagai berikut:

Raspberry Pi sebagai server di mana semua proses dan aktivitas sistem dilakukan pada perangkat ini. Model Raspberry Pi yang digunakan adalah Raspberry Pi 3 Model B dengan CPU Quad Core 1.2GHz Broadcom BCM2837 64bit dan RAM 1GB, perangkat ini juga memiliki modul WIFI dan *bluetooth* yang sudah tertanam pada perangkat, sehingga memungkinkan perangkat dapat terkoneksi ke jaringan internet melalui WIFI dari *access point*. Raspberry Pi sebagai server yang akan melakukan proses dan aktivitas pada sistem dimana semua perangkat lunak terpasang seperti *cloud storage server*, DNS *adblock server* dan PiVPN *server*.

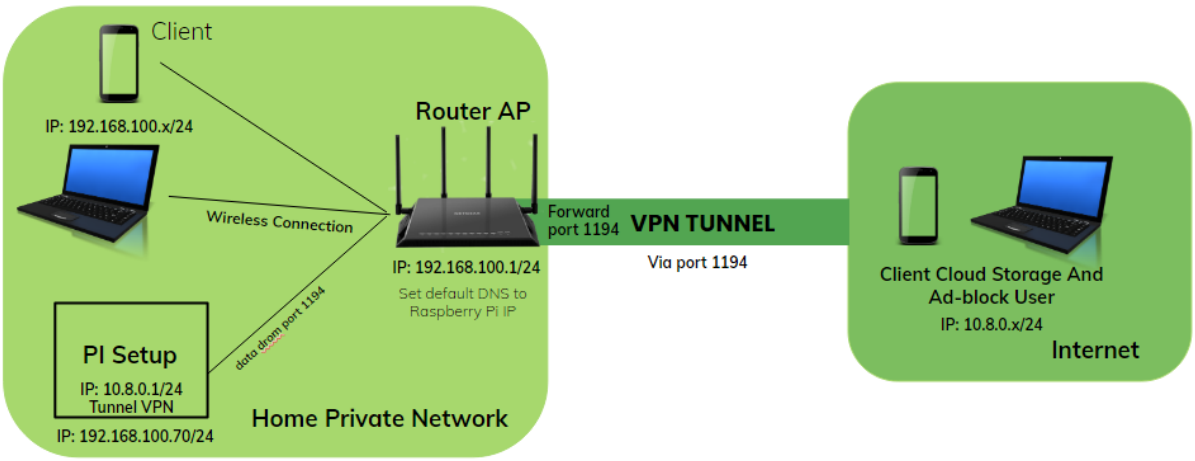
Dua buah *disk* terpasang pada Raspberry Pi melalui *port* USB yang tersedia pada perangkat. Dua buah *disk* ini berfungsi sebagai media penyimpanan pada *cloud storage server* yang terpasang pada perangkat Raspberry Pi. Disk yang terpasang pada perangkat

dikonfigurasi mekanisme *backup mirroring* menggunakan teknologi RAID 1. Pada teknologi RAID 1 yang dikonfigurasi menggunakan perangkat lunak MDADM menjadikan dua *disk* yang terpasang memiliki konfigurasi dan isi yang sama antara satu dan lain. Mekanisme *backup mirroring* yang dilakukan RAID 1 ini adalah dengan menuliskan blok yang sama di setiap *disk* yang terpasang. Keuntungannya adalah jika terjadi kerusakan pada *blok* salah satu *disk* ataupun kerusakan perangkat *disk* yang digunakan tidak merusak atau menghilangkan data yang ada di dalamnya. Selama masih ada minimal satu *disk* yang masih berfungsi dengan normal maka data akan tetap aman berada dalam sistem.

Internet *Access Point* di sini berfungsi sebagai penyedia internet bagi perangkat Raspberry Pi server. Perangkat ini memberikan akses internet dari ISP (*Internet Service Provider*) ke Raspberry Pi melalui koneksi WIFI. Perangkat ini juga sudah mempunyai DHCP (*Dynamic Host Configuration Protocol*) *server* yang bertugas memberi alamat IP pada perangkat yang terkoneksi secara acak dan berjangka waktu. Agar Raspberry Pi tetap memiliki alamat IP yang sama setiap saat perlu dilakukan DHCP *reservation* pada perangkat *access point* dengan mendaftarkan MAC *address* dari Raspberry Pi pada pengaturan *access point.*

3.2.2. Perancangan Desain Arsitektur Jaringan

Perancangan desain arsitektur jaringan adalah tahap perancangan sistem yang akan dikembangkan dan implementasinya pada jaringan yang digunakan. Pada tahap ini dibuat rancangan alamat IP yang digunakan dan di *port* berapa sistem berjalan. Perancangan tahap ini dapat dilihat pada diagram blok pada Gambar 3.2



Gambar 3.2 Diagram blok perancangan sistem dan jaringan

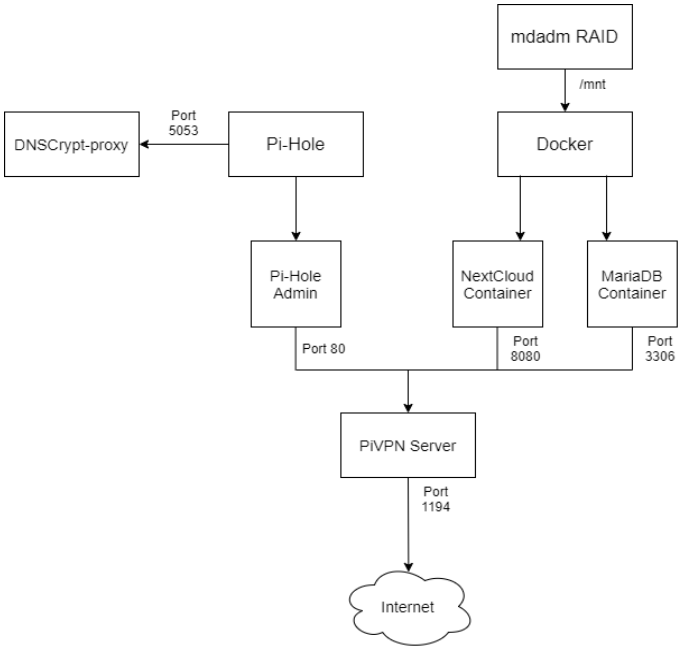
Pada Gambar 3.2 Diagram blok perancangan sistem dan jaringan setiap perangkat sudah memiliki alamat IP masing-masing. Perangkat Raspberry Pi diberikan IP 192.168.100.70/24 pada jaringan. Setiap *service* yang berjalan pada sistem dapat diakses melalui IP Raspberry Pi diikuti dengan *port* dimana service berjalan seperti Nextcloud berjalan pada *port* 8080 maka dapat diakses melalui alamat http://192.168.100.70:8080. PiVPN yang berjalan dalam perangkat Raspberry berjalan di port 1194 sebagai *tunneling* dari jaringan internet. PiVPN juga membutuhkan IP *public* dari jaringan internet yang terkoneksi. IP *public* yang ada pada jaringan internet dari ISP selalu berubah dalam waktu tertentu, oleh karena itu dibutuhkan DDNS (*Dynamic Domain Name Server*) yang merekam perubahan IP *public* agar sistem tetap dapat diakses dari jaringan internet oleh pengguna menggunakan VPN. Pada sistem ini digunakan Duckdns sebagai DDNS dengan domain terdaftar hilmanm.duckdns.org.

Perangkat *access point* memiliki alamat IP *private* 192.168.100.1/24. Perangkat *access point* memiliki DHCP *server* yang memberikan alamat IP perangkat-perangkat yang terkoneksi pada *access point* secara acak dan berjangka waktu. Sebelumnya disebutkan perangkat Raspberry Pi diberikan alamat IP 192.168.100.70, agar perangkat Raspberry Pi tetap menggunakan alamat IP tersebut perlu dilakukan DHCP *reservation* pada alamat yang sudah disebutkan menggunakan MAC *address* Raspberry Pi. Perangkat *access point* juga perlu dikonfigurasi DNS yang digunakan adalah alamat IP Raspberry Pi agar setiap perangkat yang terkoneksi dapat menggunakan service DNS *ad-blocker*.

Setiap pengguna atau *client* yang terhubung memiliki alamat IP yang diberikan secara otomatis. Pengguna yang terkoneksi pada *access point* jaringan privat rumah akan diberikan alamat IP secara acak dengan *subnet* 192.168.100.x/24 dan yang terkonkesi dengan VPN *tunneling* akan mendapatkan IP dengan *subnet* 10.8.0.x/24.

3.2.3. Perancangan Perangkat Lunak

Perancangan perangkat lunak didasarkan pada rancangan diagram blok pada gambar 3.3. Pada tahap awal perancangan perangkat lunak yang perlu dilakukan adalah mendefinisikan *service* apa saja yang harus berjalan pada sistem, bagaimana tiap *service* berhubungan satu sama lain dalam sistem dan di mana *service* berjalan. Untuk penjabaran perancangan perangkat lunak pada sistem ini dapat dilihat pada Gambar 3.3



Gambar 3.3 Diagram blok perangkat lunak pada sistem

Gambar 3.3 adalah diagram blok dari perangkat lunak yang diimplementasikan pada sistem *cloud storage* dengan konfigurasi *disk* RAID 1 dan DNS *ad-blocker* untuk perlindungan privasi data pribadi. Berikut adalah penjelasan dari alur kerja perangkat lunak pada sistem:

Disk yang dikonfigurasi RAID 1 diimplementasikan menggunakan perangkat lunak MDADM. MDADM adalah perangkat lunak yang berfungsi untuk mengatur penyebaran blok data pada beberapa *disk* yang terpasang. Perangkat lunak ini membuat *Superblocks* pada tiap *disk* yang dikonfigurasi. *Superblocks* adalah bagian dari *disk* yang menyimpan *metadata* dari *disk* yang ada di konfigurasi RAID untuk memungkinkan dilakukannya perbaikan dari *array volume* RAID. *Disk* yang sudah dikonfigurasi RAID 1 menggunakan perangkat lunak MDADM lalu di-*mount* (dipasang) pada direktori /mnt untuk pengaksesan oleh sistem.

Pada sistem ini juga berjalan dua docker *container* yang masing-masing adalah *container* Nextcloud sebagai *cloud storage* dan *container* MariaDB sebagai *database* untuk Nextcloud yang menyimpan seluruh data dari Nextcloud *cloud storage server*. Kedua *container* yang berjalan ini seluruh volume atau datanya disimpan pada direktori lokal /mnt di mana direktori /mnt adalah *disk* yang sudah dikonfigurasi RAID 1. Kedua *container* ini juga di lakukan *expose port* pada masing-masing *container* agar dapat diakses melalui *port* yang digunakan. *Port* yang di-*expose* masing-masing adalah *port* 8080 untuk Nextcloud dan 3306 untuk MariaDB *database*.

Pi-hole sebagai perangkat lunak yang berfungsi untuk pemblokiran iklan di sini menggunakan dnscrypt-proxy sebagai DNS server di mana dnscrypt-proxy berjalan pada *port* 5053 di sistem. Pi-hole menggunakan beberapa *port* pada sistem, *port* 53 digunakan Pi-hole untuk merespons DNS *query* yang masuk dan *port* 80 digunakan untuk *web server* admin panel Pi-Hole. Untuk pemblokiran iklan Pi-hole menggunakan *ad list* atau daftar kumpulan *domain* yang mengandung iklan. Setiap DNS *query* yang terbaca melakukan akses ke *domain* yang mengandung iklan akan langsung diblokir oleh Pi-Hole.

Jika pengguna terhubung dengan koneksi internet yang sama dengan Raspberry Pi server pengguna dapat langsung mengakses *port* 80 untuk mengakses *cloud storage.* Namun, jika pengguna menggunakan jaringan internet dari luar pengguna tidak dapat langsung mengakses ke *port* 80. Dengan PiVPN server memungkinkan sistem dapat diakses dari jaringan internet. PiVPN berfungsi sebagai *point-to-point tunnel* yang menggunakan *port* 1194 dengan protokol UDP untuk pengaksesan. Setiap pengguna yang ingin mengakses sistem dari jaringan internet harus memiliki *credential* yang diperlukan untuk melakukan koneksi menggunakan VPN *client.* VPN *client* yang digunakan pada sistem ini adalah OpenVPN *client*.

BAB IV
IMPLEMENTASI DAN PEMBAHASAN

Implementasi dan perancangan akan membahas tahapan-tahapan perancangan sistem *cloud storage* dengan konfigurasi *disk* RAID 1 dan DNS *ad-blocker* untuk perlindungan privasi data pribadi. Pada tahap ini akan dibahas mengenai hasil dan analisis dari perancangan sistem yang dikembangkan.

4.1. Implementasi

Implementasi merupakan tahapan-tahapan yang dilakukan penulis dalam mengimplementasikan perancangan yang sebelumnya sudah dibuat. Pada tahap implementasi akan memperlihatkan hasil dari sistem *cloud storage* dengan konfigurasi *disk* RAID 1 dan DNS *ad-blocker* untuk perlindungan privasi data pribadi yang akan berjalan sesuai dengan fungsinya. Pada tahap implementasi penjabarannya akan dibagi dua, yang pertama adalah implementasi perangkat keras dan yang kedua adalah implementasi perangkat lunak.

4.1.1. Perangkat Keras yang Digunakan

Perangkat keras di sini adalah perangkat keras yang digunakan sebagai perangkat pendukung perancangan sistem *cloud storage* dengan konfigurasi *disk* RAID 1 dan DNS *ad-blocker* untuk perlindungan privasi data pribadi. Perangkat keras yang digunakan penulis dalam menunjang perancangan sistem adalah sebagai berikut:

a. Laptop Lenovo B490 Core i3-2348M 2.30GHz 8 GB RAM
b. Raspberry Pi 3 Model B CPU Quad Core 1.2GHz Broadcom BCM2837 64bit dan RAM 1GB
c. 2 *Flash Drive* Sandisk 16 GB
d. *Access Point Router* Huawei HG8245H5

4.1.2. Perangkat Lunak yang Digunakan

Dalam tahapan implementasi sistem *cloud storage* dengan konfigurasi *disk* RAID 1 dan DNS *ad-blocker* untuk perlindungan privasi data pribadi dibutuhkan beberapa perangkat lunak

yang digunakan untuk mendukung pengembangan sistem. Perangkat lunak yang digunakan sebagai pendukung pengembangan sistem adalah sebagai berikut:

a. Sistem Operasi

Sistem operasi merupakan perangkat lunak yang digunakan penulis untuk mengatur sumber daya dari perangkat keras dan perangkat lunak yang ada pada komputer. Dalam perancangan sistem *cloud storage* dan DNS *adblock* ini digunakan sistem operasi Windows 10.

b. Windows Terminal

Windows Terminal adalah perangkat lunak yang digunakan untuk menjalankan perintah-perintah dasar dalam komputer. Pada penelitian ini digunakan Windows Terminal untuk melakukan koneksi ke perangkat keras yang terhubung di internet melalui protokol SSH dan melakukan konfigurasi pada sistem.

c. OpenVPN *Client*

VPN *client* merupakan perangkat lunak yang digunakan untuk melakukan koneksi ke VPN server. Pada penelitian ini digunakan OpenVPN *client* yang dipasang pada komputer dan *smartphone* sebagai pendukung pengembangan sistem.

d. Nextcloud *Client*

Nextcloud *client* merupakan aplikasi OS *Mobile* (iOS dan Android) yang digunakan untuk pengaksesan data pada *cloud storage*. Semua data yang tersimpan pada *cloud storage* dapat ditampilkan pada Nextcloud *Client*.

4.1.3. Implementasi Perangkat Keras

Pada tahap implementasi perangkat keras diperlukan beberapa perangkat keras yang sudah ditentukan sebelumnya dalam analisis perangkat keras dalam bab3. Perangkat keras yang diperlukan adalah sebagai berikut:

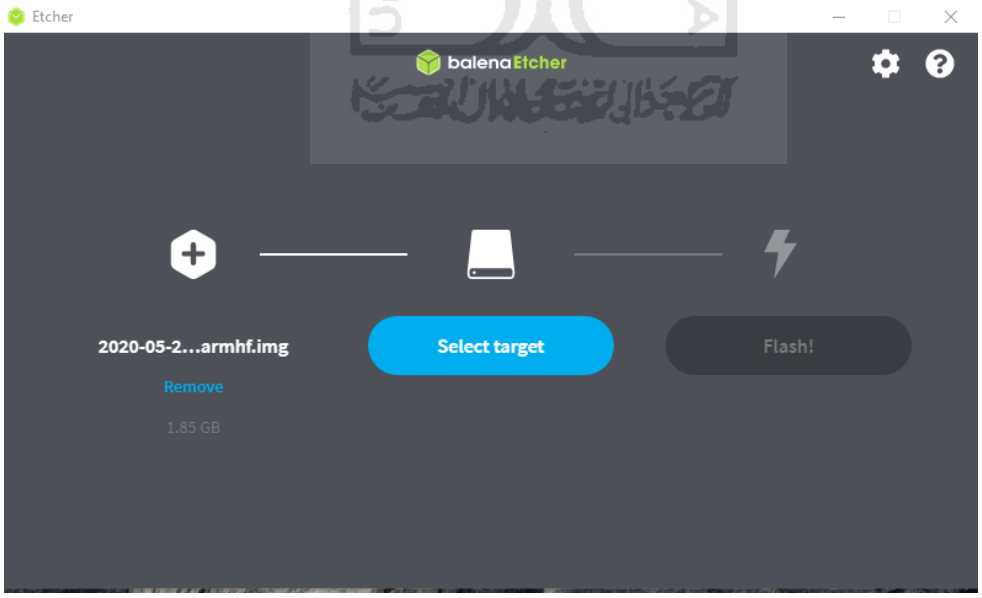
a. Raspberry Pi 3 Model B
b. MicroSD card 16 GB
c. 2 *Flash disk* 16GB
d. *Router Access Point*
e. Kabel USB
f. *Adaptor power* 5V

Setelah semua perangkat keras yang diperlukan telah tersedia, langkah berikutnya yang perlu dilakukan adalah melakukan konfigurasi pada tiap perangkat keras yang digunakan. Langkah-langkah konfigurasi perangkat keras yang dilakukan adalah sebagai berikut:

a. Instalasi dan konfigurasi Raspberry Pi

Langkah pertama yang dilakukan dalam instalasi dan konfigurasi Raspberry Pi adalah mengunduh sistem operasi Raspberry Pi OS yang akan digunakan sebagai sistem operasi yang berjalan. Untuk mengunduh sistem operasi Raspberry Pi OS dapat dilakukan pada halam resmi dari Raspberry Pi. Terdapat beberapa versi dalam sistem operasi Raspberry Pi OS, namun dalam penelitian ini digunakan Raspberry Pi OS (32-bit) Lite. Dalam versi Lite ini sistem operasi tidak dibekali dengan tampilan *desktop* melainkan hanya CLI (*Command Line Interface).*

Langkah selanjutnya adalah membuat *bootable disk* Raspberry Pi OS dengan sistem operasi yang sudah diunduh sebelumnya ke dalam MicroSD. Pada tahap ini digunakan perangkat lunak Balena Etcher sebagai media pembuatan *bootable disk.* Cara membuatnya cukup sederhana hanya dengan memilih *file* sistem operasi yang sudah diunduh dan media targetnya yaitu MicroSD seperti yang terlihat pada Gambar 4.1. Setelah itu sistem operasi Raspberry Pi OS akan melakukan instalasi secara otomatis saat dihidupkan dengan menghubungkan perangkat Raspberry Pi pada *power adaptor* seperti pada Gambar 4.2.

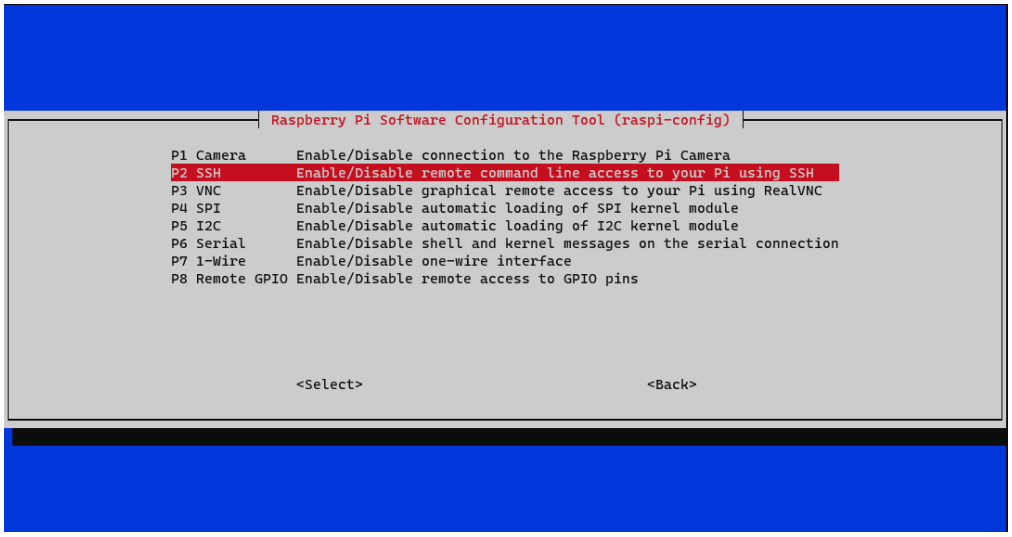


Gambar 4.1 Proses membuat *bootable disk*

Setelah perangkat dihubungkan dengan *power adaptor* menggunakan kabel USB, perangkat bisa dihubungkan dengan perangkat *keyboard* dan layar monitor untuk melihat progres instalasi sistem operasi dan melakukan konfigurasi awal.

Gambar 4.2 Pemasangan Raspberry Pi dengan *Power Adaptor*

Konfigurasi awal yang dilakukan adalah melakukan aktivasi *protocol* SSH pada perangkat. SSH berguna untuk pengaksesan perangkat secara *remote* melalui internet. Pengaktifan SSH dilakukan menggunakan perintah "raspi-config" pada terminal dan memilih "enable SSH" pada bagian *interface* seperti yang terlihat pada Gambar 4.3.



Gambar 4.3 Konfigurasi pengaktivan SSH Raspberry Pi

b. Pemasangan *Disk* pada perangkat Raspberry Pi.

Setelah melakukan konfigurasi pada perangkat Raspberry Pi, dua buah *disk* yang sudah disiapkan akan dihubungkan ke perangkat Raspberry Pi. Kedua *disk* dihubungkan ke *port* USB yang ada pada Raspberry Pi seperti pada Gambar 4.4

Gambar 4.4 Pemasangan *disk* pada *port* USB

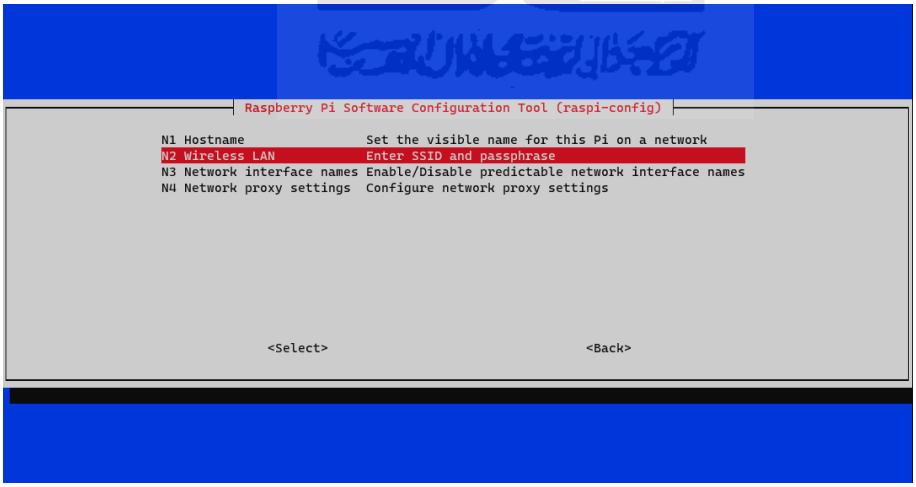
Untuk melakukan pengecekan apakah *disk* yang terpasang terbaca oleh sistem dapat dilakukan menggunakan terminal Raspberry Pi dengan perintah “fdisk -l”. Perintah tersebut akan memperlihatkan daftar perangkat *disk* apa saja yang terpasang pada Raspberry Pi. Seperti pada Gambar 4.5 ada dua buah *disk* berukuran 14.5 GB yang terpasang pada /dev/sda dan /dev/sdb.

```
Disk /dev/sda: 14.5 GiB, 15597568000 bytes, 30464000 sectors
Disk model: Cruzer Blade
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x006d7608

Device     Boot Start      End  Sectors  Size Id Type
/dev/sda1        2048 30463999 30461952 14.5G  7 HPFS/NTFS/exFAT


Disk /dev/sdb: 14.6 GiB, 15682240512 bytes, 30629376 sectors
Disk model: Cruzer Blade
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x80ec1767

Device     Boot Start      End  Sectors  Size Id Type
/dev/sdb1          32 30629375 30629344 14.6G  7 HPFS/NTFS/exFAT


Disk /dev/md0: 14.5 GiB, 15587082240 bytes, 30443520 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
pi@raspberrypi:~/duckdns $
```

Gambar 4.5 Daftar perangkat *disk* yang terbaca sistem

c. Konfigurasi WIFI dan *Access Point*

Konfigurasi WIFI dilakukan untuk menghubungkan perangkat ke jaringan internet. Untuk melakukan konfigurasi WIFI dapat dilakukan dengan perintah “raspi-config” pada terminal dan pilih bagian “*Network*” dan “*Wireless LAN”* lalu masukkan nama SSID WIFI yang ingin disambungkan beserta kata sandinya seperti pada Gambar 4.6



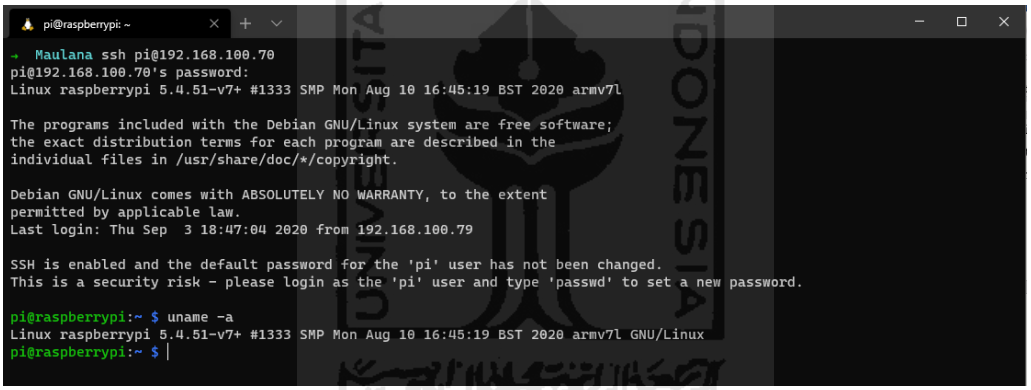
Gambar 4.6 Konfigurasi Wifi Raspberry Pi

Pada perangkat *access point* perlu dilakukan konfigurasi penetapan alamat IP permanen pada perangkat Raspberry Pi. Konfigurasi ini berguna untuk menghindari alamat IP dari perangkat Raspberry Pi berubah dan digunakan perangkat lain, karena *access point* secara

otomatis akan memberikan perangkat yang terhubung dengannya alamat IP yang berubah dalam jangka waktu tertentu.

4.1.4. Implementasi Perangkat Lunak

Implementasi perancangan perangkat lunak ini adalah penerapan dari perancangan beberapa perangkat lunak yang sudah disebutkan pada bab3. Pada implementasinya perancangan perangkat lunak dilakukan dengan menggunakan akses *remote* pada perangkat Raspberry Pi untuk pengembangan sistem. Akses *remote* dilakukan menggunakan protokol SSH dan dilakukan pada Windows Terminal seperti pada Gambar 4.7. Untuk melakukan akses *remote* diperlukan data *user* dan *password* dari Raspberry Pi. Data *user* yang otomatis diberikan oleh sistem saat instalasi adalah “pi” dengan *password* “raspberrry”.



Gambar 4.7 Koneksi *remote* menggunakan SSH

Pada tahap implementasi perancangan perangkat lunak, penulis membagi implementasi menjadi empat bagian yaitu implementasi konfigurasi RAID 1, *cloud storage*, DNS *ad-blocker* dan VPN *server*. Implementasi perancangan perangkat lunak pada sistem *cloud storage* dengan konfigurasi *disk* RAID 1 dan DNS *ad-blocker* untuk perlindungan privasi data pribadi adalah sebagai berikut:

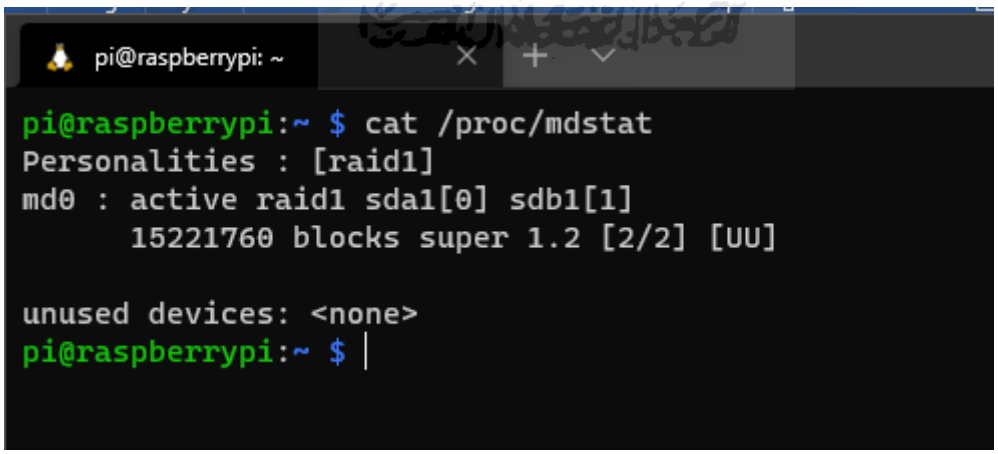
a. Implementasi Konfigurasi RAID 1

Implementasi konfigurasi RAID 1 pada *disk* pada sistem ini dilakukan dengan menggunakan perangkat lunak MDADM yang sudah dijelaskan pada kebutuhan perangkat lunak bab3. Adapun langkah-langkah konfigurasi RAID 1 pada *disk* menggunakan MDADM dapat dilihat pada kode program konfigurasi Tabel 4.1

Tabel 4.1 Kode Konfigurasi MDADM

```
$ sudo apt install mdadm

$ sudo mdadm -create -verbose /dev/md0 -level=mirror -raid-device=2 /dev/sda1
/dev/sdb1

$ cat /proc/mdstat

$ sudo -i

$ mdadm -detail -scan >> /etc/mdadm/mdadm.conf
```

Konfigurasi yang dilakukan pada Tabel 4.1 adalah melakukan instalasi perangkat lunak MDADM pada Raspberry Pi. Setelah dilakukan instalasi kemudian dilakukan pembuatan RAID *volume* pada dua buah *disk* yang sudah terdeteksi pada Raspberry Pi sebagai mana disebutkan pada implementasi perancangan perangkat keras. RAID *array* yang dibuat adalah RAID *level* 1 atau *mirror* yang dinamai /dev/md0. Setelah *volume* terbuat status prosesnya dapat terlihat seperti pada Gambar 4.8 dan selanjutnya konfigurasi *array* disimpan pada *file* konfigurasi mdadm.



Gambar 4.8 Status Proses Pembuatan RAID *array*

Setelah melakukan pembuatan RAID *volume* selanjutnya adalah melakukan *build file system* dari RAID *volume* yang sudah dibuat. RAID *volume* akan dilakukan *build* menjadi *file*

system ext4 dan akan di-*mount* (pasang) pada direktori /mnt. Konfigurasi *build file system* dan *mount* dapat dilihat pada Tabel 4.2.

Tabel 4.2 Kode Program Konfigurasi *Build* dan *Mount*

```
$ sudo mkfs.ext4 -v -m .1 -b 4096 -E stride=32,stripe=64 /dev/md0

$ sudo mount /dev/md0 /mnt

$ df -h
```

RAID *volume* yang sudah dilakukan *mount* ke direktori /mnt dapat terlihat pada daftar perangkat yang terpasang pada sistem seperti pada Gambar 4.9. Pada daftar perangkat yang sudah terpasang pada sistem juga diperlihatkan kapasitas penyimpanan yang sudah terpakai dan sisa yang masih bisa digunakan. Pada Gambar 4.9 juga terlihat bahwa kapasitas /dev/md0 adalah 14 GB.

```
pi@raspberrypi:~ $ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        15G  2.9G   12G  21% /
devtmpfs        431M     0  431M   0% /dev
tmpfs           463M  2.1M  461M   1% /dev/shm
tmpfs           463M   13M  451M   3% /run
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           463M     0  463M   0% /sys/fs/cgroup
/dev/mmcblk0p1  253M   54M  199M  22% /boot
/dev/md0         15G  567M   14G   4% /mnt
tmpfs            93M     0   93M   0% /run/user/999
tmpfs            93M     0   93M   0% /run/user/1000
```

Gambar 4.9 Daftar Perangkat *Disk* yang Terpasang Pada Sistem

Konfigurasi terakhir dilakukan pada RAID *disk* adalah membuat *disk* tetap terpasang pada direktori /mnt saat sistem melakukan *restart.* Untuk itu dilakukan konfigurasi pada file /etc/fstab dengan menambahkan baris /dev/md0 seperti pada Tabel 4.3 agar *disk* otomatis terpasang pada direktori /mnt setiap kali sistem melakukan *restart.*

Tabel 4.3 Kode Program Konfigurasi File /etc/fstab

```
proc                    /proc         proc    defaults           0       0
PARTUUID=eb98cd54-01    /boot         vfat    defaults           0       2
PARTUUID=eb98cd54-02    /             ext4    defaults,noatime   0       1
/dev/md0                /mnt          ext4    defaults           0       0
```

b. Implementasi *Cloud Storage*

Pada sistem *cloud storage* dengan konfigurasi *disk* RAID 1 dan DNS *ad-blocker* untuk perlindungan privasi data pribadi ini sistem memanfaatkan *platform* Docker untuk menjalankan dua buah *container*. Dua buah *container* yang berjalan adalah Nextcloud sebagai sistem manajemen *cloud storage* dan MariaDB sebagai *database* dari Nextcloud. Langkah pertama yang dilakukan dalam implementasi ini adalah melakukan instalasi Docker, instalasi Docker dapat dilakukan menggunakan kode program seperti pada Tabel 4.4. Setelah melakukan instalasi dilakukan penambahan *permission* pada user "pi" agar dapat menjalankan Docker tanpa perintah *superuser*.

Tabel 4.4 Kode Program Instalasi Docker

```
$ curl -sSL https://get.docker.com | sh

$ sudo usermod -aG docker pi

$ docker -v
```

Setelah melakukan instalasi docker selanjutnya adalah melakukan instalasi docker-compose. Docker-compose *tool* yang dapat mendefinisikan dan menjalankan beberapa *container* sekaligus. Pada penelitian ini *container* Nextcloud dan MariaDB akan didefinisikan dan dijalankan secara bersamaan dan bergantung satu sama lainnya. Untuk instalasi docker-compose pada Raspberry Pi dapat menggunakan kode program seperti pada Tabel 4.5.

Tabel 4.5 Kode Program Instalasi docker-compose

```
$ sudo curl -L --fail https://raw.githubusercontent.com/linuxserver/docker-docker-
compose/master/run.sh -o /usr/local/bin/docker-compose

$ sudo chmod +x /usr/local/bin/docker-compose
```

Untuk menjalankan *container* dengan docker-compose diperlukan sebuah file docker-compose.yml. File tersebut berguna untuk mendefinisikan *container* yang akan dijalankan, mulai dari nama, image yang digunakan, *port* yang diberikan dan konfigurasi lainnya. *File* docker-compose.yml lalu dijalankan dengan perintah "docker-compose" pada terminal. Isi konfigurasi docker-compose.yml pada penelitian ini dapat dilihat pada Tabel 4.6

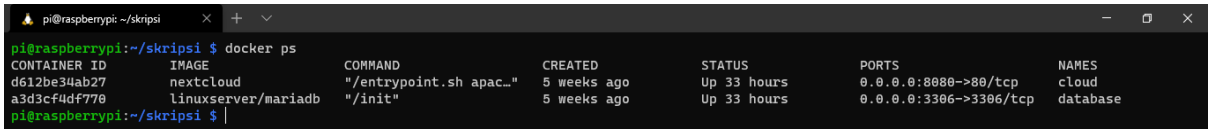
Tabel 4.6 Kode Program docker-compose.yml

```
version: "3"

services:
  db:
    container_name: database
    image: linuxserver/mariadb
    ports:
      - 3306:3306
    volumes:
      - /mnt/mysql/:/var/lib/mysql
      - /mnt/mysql/:/config
    environment:
      - MYSQL_ROOT_PASSWORD=root
      - MYSQL_PASSWORD=pass
      - MYSQL_DATABASE=nextcloud
      - MYSQL_USER=nextcloud
    restart: unless-stopped

  app:
    container_name: cloud
    image: nextcloud
    ports:
      - 8080:80
    links:
      - db
    volumes:
      - /mnt/nextcloud/:/var/www/html
    restart: always
```

Pada kode program di Tabel 4.6 yang berisi file docker-compose.yml terdapat definisi dari dua *container* yang akan dijalankan. *Container* yang pertama yaitu *database*, *container* ini menggunakan *image* linuxserver/mariadb yang akan berjalan pada *port* eksternal 3306. Isi *file* dari *container* ini dilakukan *binding* ke direktori /mnt/mysql yang merupakan direktori *disk* yang sudah dikonfigurasi RAID 1. Dalam *container* database juga ditambahkan konfigurasi *database* yang akan dibuat beserta *user* dan *password*-nya untuk keperluan *database* Nextcloud.

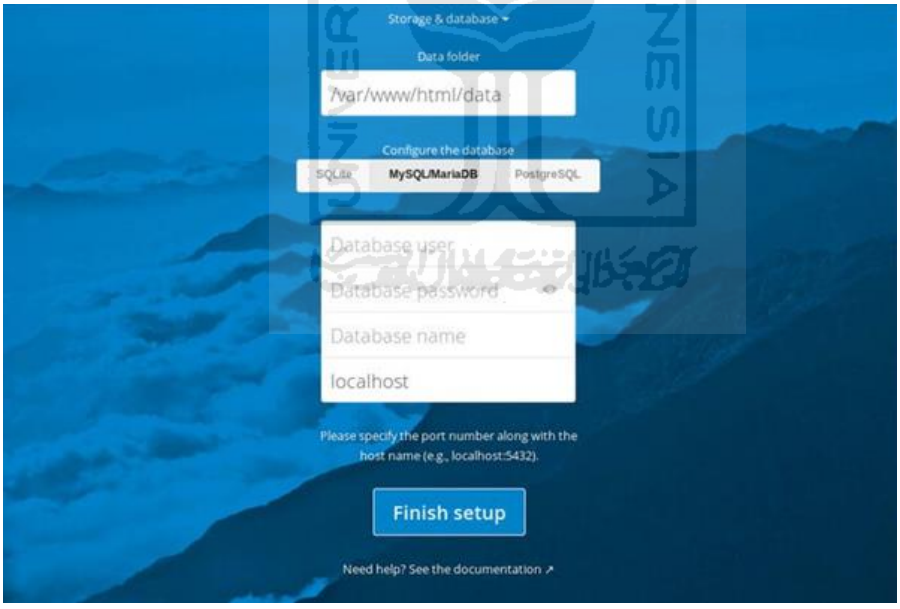
Container kedia yang didefinisikan adalah cloud. *Container* ini menggunakan *image* nextcloud dan berjalan pada *port* external 8080. *Container* cloud akan beketerkaitan dengan *container* database dan data dari *container* cloud dilakukan *binding* ke direktori /mnt/nextcloud yang juga merupakan direktori *disk* yang sudah dikonfigurasi RAID 1.

Setelah dilakukan pendefinisian *container* pada file docker-compose.yml, *container-container* yang sudah didefinisikan dapat dijalankan menggunakan perintah “docker-compose”. Untuk melihat apakah *container* sudah berjalan tanpa *error* dapat dilihat menggunakan perintah “docker ps” pada terminal seperti pada Gambar 4.10



Gambar 4.10 Container yang Berjalan Pada Sistem

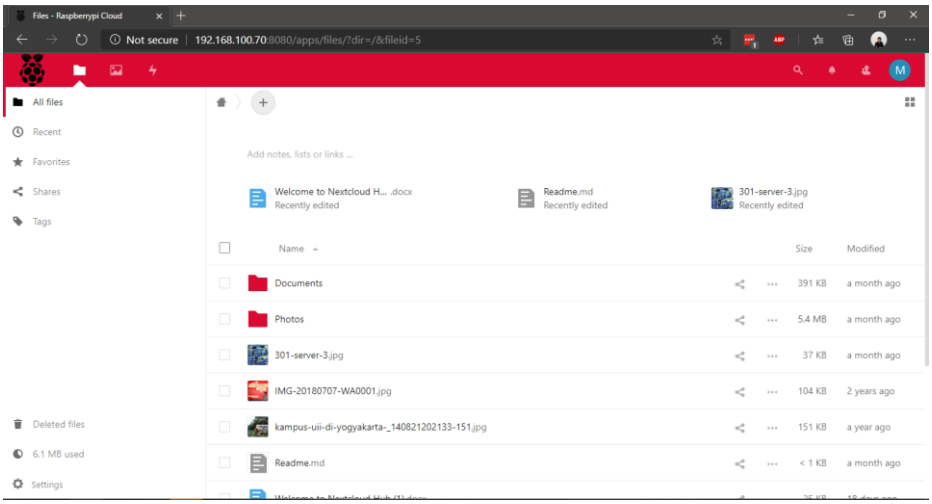
Container yang sudah berjalan dapat diakses pada *browser* menggunakan alamat IP dari Raspberry Pi diikuti dengan *port* di mana *container* berjalan. Untuk Nextcloud dapat diakses dengan alamat http://192.168.100.70:8080 pada *web browser*. Saat pertama kali akses Nextcloud setelah berjalan, Nextcloud akan meminta untuk melakukan konfigurasi awal yaitu pembuatan akun admin dan konfigurasi *database*. Pada tahap ini dilakukan pembuatan akun admin dengan nama "maulana" dan konfigurasi *database* dengan memilih jenis Mysql/MariaDB seperti pada Gambar 4.11 dan informasinya bisa diisikan sesuai yang sudah didefinisikan pada *file* docker-compose.yml



Gambar 4.11 Konfigurasi Database Nextcloud

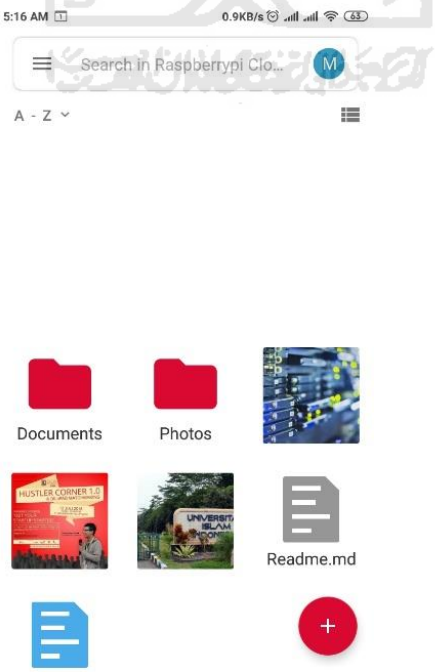
Setelah semua konfigurasi sudah dilakukan Nextcloud sebagai *cloud storage* sudah dapat digunakan untuk melakukan *upload, download* dan melihat *file* di dalam sistem. Dalam sistem Nextcloud ini juga dapat dilakukan manajemen pengguna jika ingin menambahkan atau

menghapus pengguna yang dapat masuk ke dalam sistem. Tampilan antarmuka Nextcloud sebagai *cloud storage* dapat dilihat pada Gambar 4.12



Gambar 4.12 Tampilan Antarmuka *Cloud Storage* Web

Nextcloud juga memiliki aplikasi *client* pada *platform mobile* android dimana pengguna dapat menggunakan layanan *cloud storage* ini pada *smartphone*. Untuk pengaksesan via aplikasi android dapat menggunakan alamat yang sama saat melakukan akses pada *web browser*. Tampilan antarmuka aplikasi Nextcloud *client* android dapat dilihat pada Gamber 4.13



Gambar 4.13 Tampilan Antarmuka Nextcloud Client Android

c. Implementasi DNS *ad-blocker*

Pada sistem ini DNS *ad-blocker* menggunakan Pi-Hole sebagai *ad-blocker* yang menggunakan dnscrypt-proxy sebagai DNS *resolver*. Langkah-langkah implementasi DNS *ad-blocker* pada Raspberry Pi adalah sebagai berikut:

1.1. Instalasi Pi-Hole

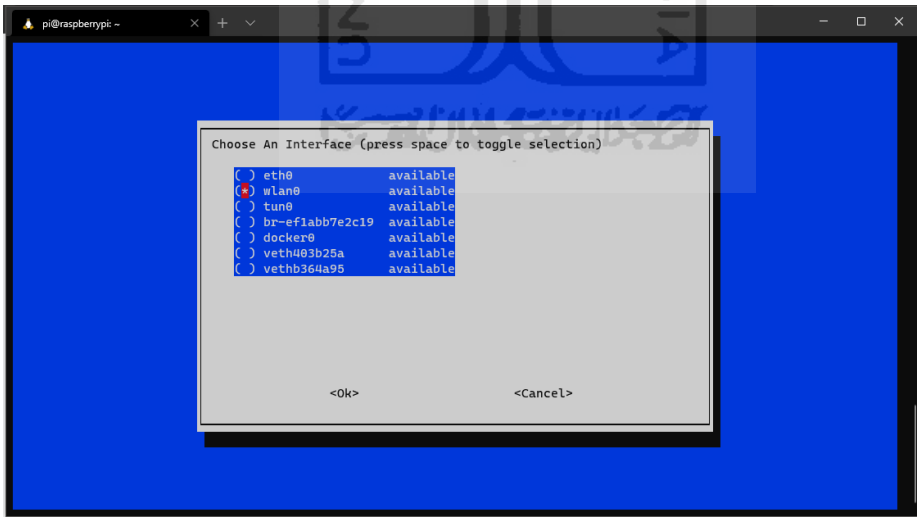
Instalasi Pi-Hole dilakukan menggunakan satu baris kode program yang dijalankan pada terminal. Baris kode instalasi Pi-Hole adalah seperti pada Tabel 4.7

Tabel 4.7 Kode Program Instalasi Pi-Hole

```
$ curl -sSL https://install.pi-hole.net | bash
```

1.2. Konfigurasi Pi-Hole

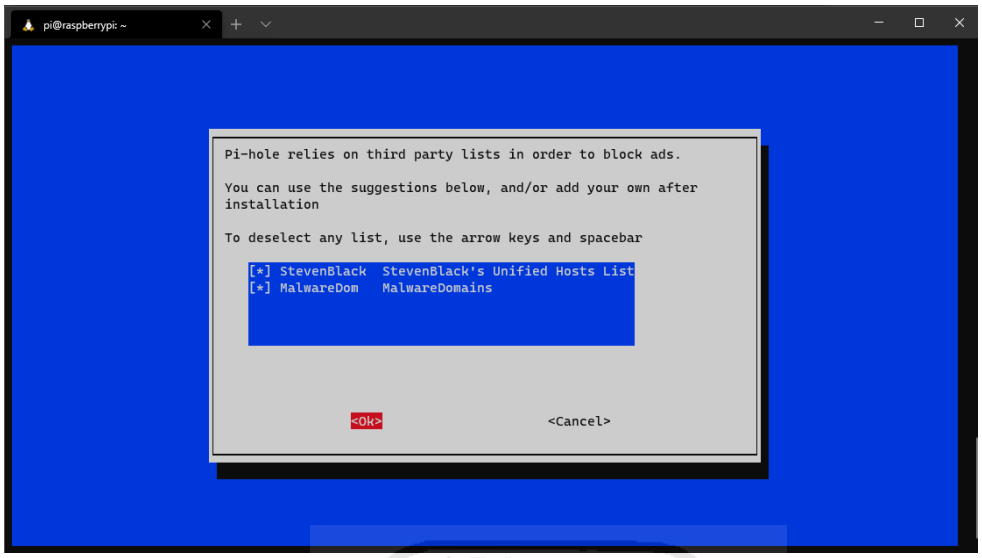
a) Saat proses instalasi berjalan ada beberapa hal yang perlu dikonfigurasi pada sistem Pi-Hole yang pertama adalah *Interface* yang digunakan untuk menangkap *traffic* pada jaringan. Karena Raspberry Pi terkoneksi pada jaringan WIFI maka *interface* yang dipilih adalah wlan0 seperti pada Gambar 4.14



Gambar 4.14 Konfigurasi *Interface* Pi-hole

b) Pi-Hole bergantung kepada daftar blokir iklan dari pihak ketiga. Daftar inilah yang digunakan Pi-Hole untuk menolak *query* yang masuk pada jaringan sesuai dengan daftar yang ada. Pada instalasi awal dapat digunakan dua pilihan bawaan dari Pi-Hole

dan setelahnya dapat ditambahkan sesuai kebutuhan. Dua buah daftar sumber daftar blokir adalah seperti pada Gambar 4.15

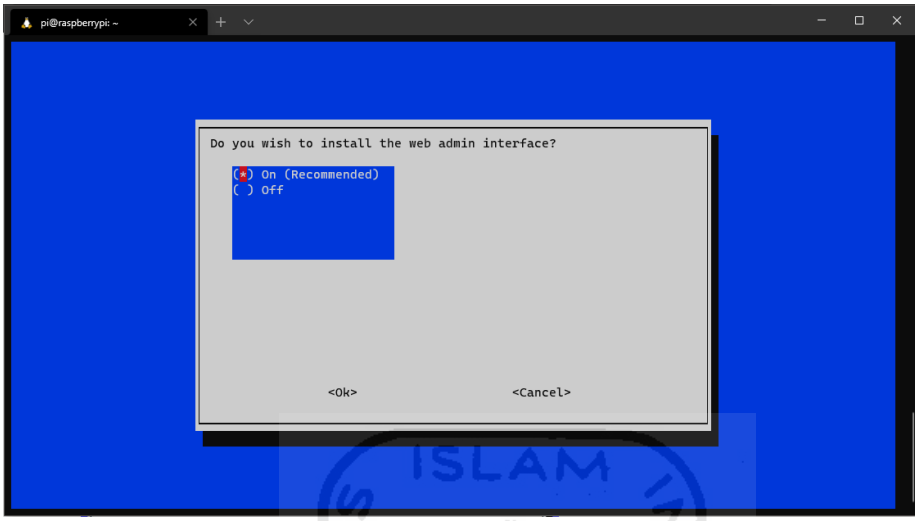


Gambar 4.15 Daftar Blokir Bawaan Pi-Hole

c) Pi-Hole akan membaca alamat IP dari perangkat Raspberry Pi dan meminta untuk mengkonfirmasi apakah alamat yang terbaca akan digunakan sebagai alamat statis dari Pi-Hole seperti pada Gambar 4.16

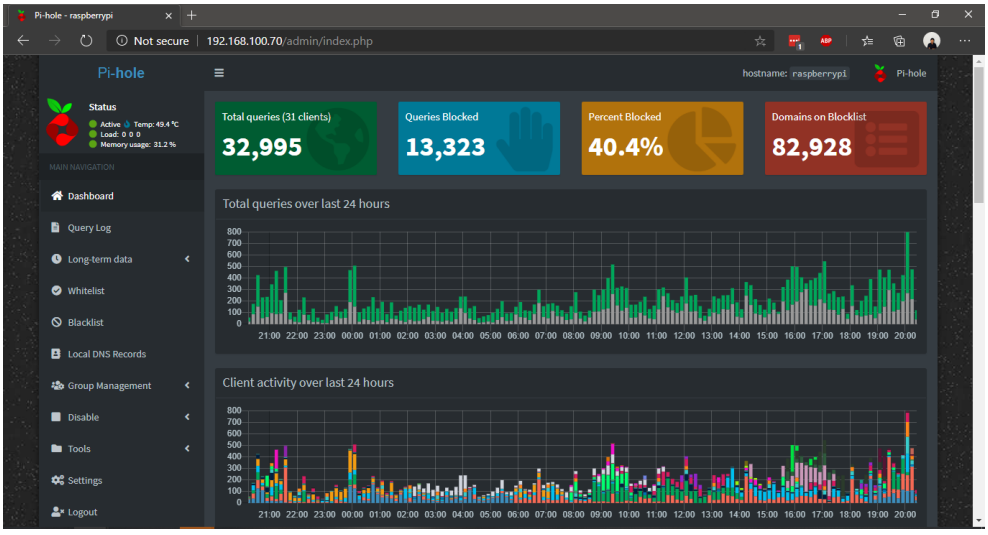
Gambar 4.16 Alamat Statis Pi-Hole

d) Konfigurasi terakhir adalah melakukan konfirmasi untuk melakukan instalasi *web admin interface* seperti pada Gambar 4.17. *Web admin* digunakan untuk melakukan pengaturan dengan mudah menggunakan antarmuka melalui *web browser*. Halaman *web admin* dapat diakses pada http://192.168.100.70/admin.



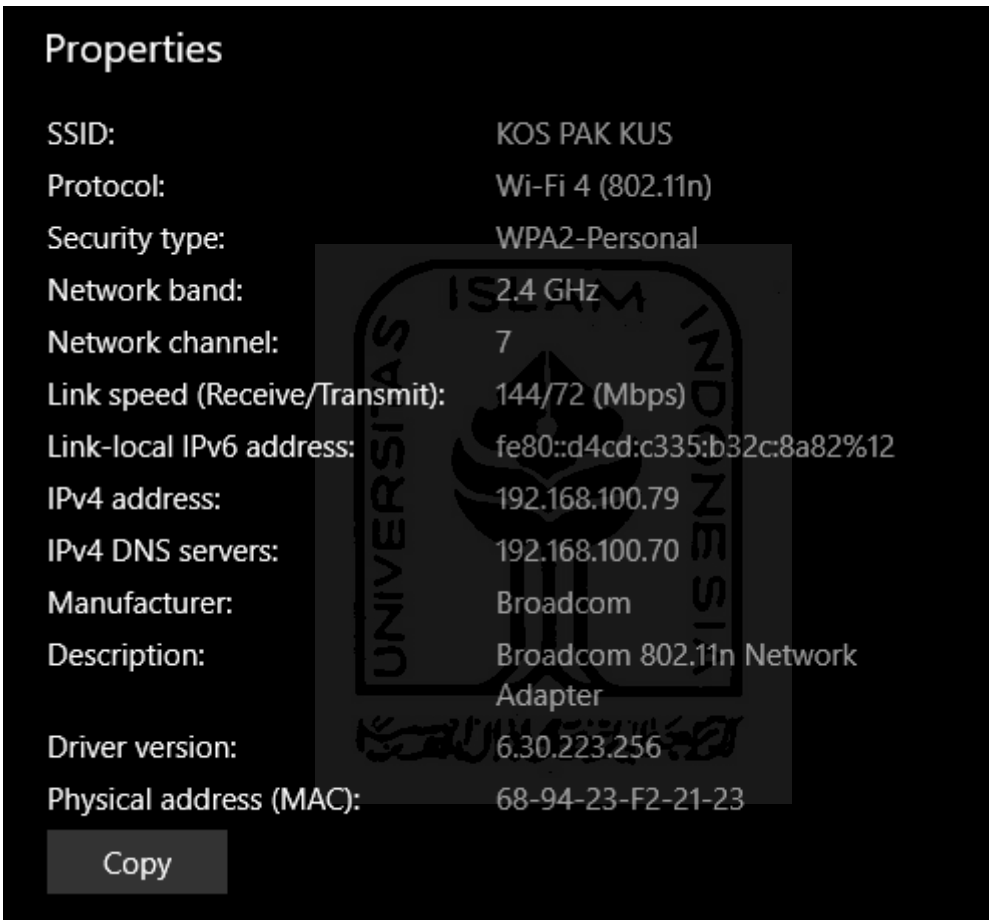
Gambar 4.17 Instalasi *Web Admin*

e) Setelah melakukan semua konfigurasi *web admin* dapat diakses pada alamat yang sudah ditentukan. Halaman *admin* akan memperlihatkan grafis dari data *query* yang berhasil diblokir dan diteruskan kepada *resolver* dan berapa banyak pengguna yang terkoneksi pada jaringan Pi-Hole seperti pada Gambar 4.18



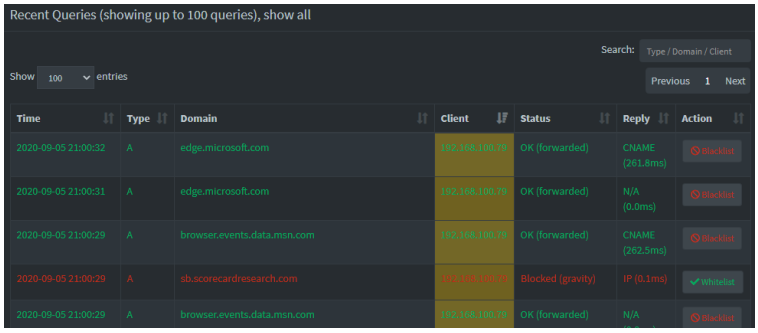
Gambar 4.18 Antarmuka *Web Admin* Pi-Hole

f) Agar perangkat dalam jaringan dapat menggunakan layanan Pi-Hole perlu dilakukan konfigurasi pada DNS agar menggunakan alamat IP dari Pi-Hole. Untuk konfigurasi ini dapat dilakukan pada DHCP *server access point* atau melakukan konfigurasi DNS manual pada tiap perangkat menggunakan alamat IP Pi-Hole. Seperti pada Gambar 4.19 komputer terkoneksi pada jaringan WIFI dengan alamat IP 192.168.100.79 dan menggunakan alamat IP Pi-Hole sebagai DNS.



Gambar 4.19 Konfigurasi DNS Komputer

g) Pi-Hole akan membaca seluruh *log query* dari perangkat yang terkoneksi pada jaringan dan menggunakan alamat IP Pi-Hole sebagai DNS. Seluruh *log query* akan ditampilkan pada halaman *web admin* Pi-Hole seperti pada Gambar 4.20



Gambar 4.20 *Log Query Web Admin* Pi-Hole

1.3. Konfigurasi dnscyprt-proxy Sebagai DNS *Resolver*

Pada penelitian ini digunakan dnscrypt-proxy dan DNS doh.tiar.app sebagai DNS *resolver* yang digunakan pada Pi-Hole. Dnscrypt-proxy adalah DNSCrypt *client* untuk DNS *privacy* dan menggunakan doh.tiar.app yaitu DNS-over-HTTPS *ad-blocker* agar Pi-Hole dapat melakukan pemblokiran iklan yang lebih luas. Untuk konfigurasinya pada sistem yang pertama adalah melakukan instalasi dnscrypt-proxy yang dapat dilihat pada baris kode di Tabel 4.8.

Tabel 4.8 Kode Program Instalasi dnscrypt-proxy

```
$wgethttps://github.com/DNSCrypt/dnscrypt-
proxy/releases/download/2.0.44/dnscrypt-proxy-linux_arm-2.0.44.tar.gz

# Extract
$ tar xzvf dnscrypt-proxy-linux_arm-2.0.44.tar.gz

# Rename
$ mv linux-arm dnscrypt-proxy

$ cp example-dnscrypt-proxy.toml dnscrypt-proxy.toml
```

Setelah dilakukan instalasi perlu dilakukan konfigurasi pada file dnscrypt-proxy.toml. Konfigurasi yang dilakukan adalah menentukan dnscrypt-provider dan dalam penelitian ini menggunakan doh.tiar.app. Dalam *file* juga ditentukan pada *port* berapa dnscrypt-proxy berjalan, secara *default* dnscrypt-proxy akan berjalan pada *port* 53. Namun karena *port* 53 sudah digunakan oleh Pi-Hole maka dalam penelitian ini digunakan *port* 5053. Berikut konfigurasi yang ditambahkan pada *file* dnscrypt-proxy.toml dapat dilihat pada Tabel 4.9

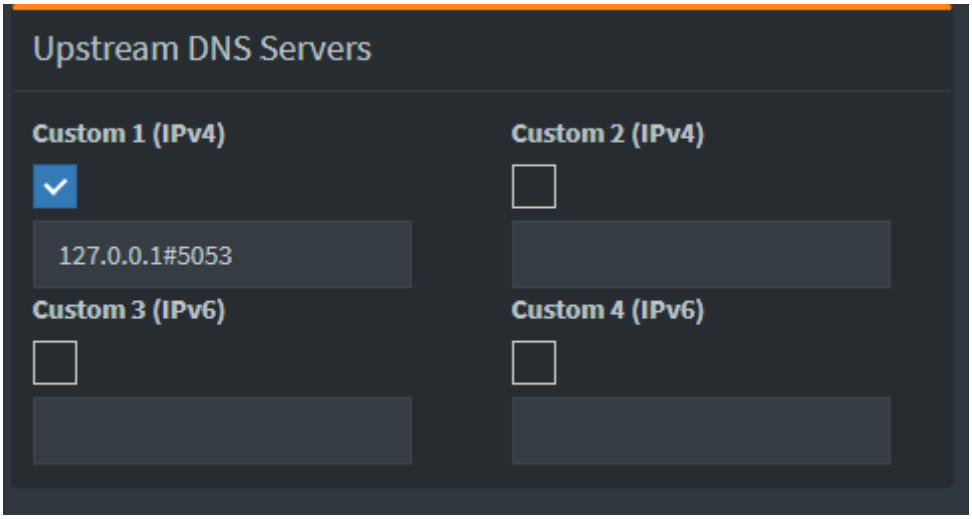
Tabel 4.9 Kode Program Konfigurasi File dnscrypt-proxy.toml

```
server_names = ['id-gmail', 'id-gmail-doh', 'id-gmail-ipv6', 'id-gmail-doh-ipv6']

listen_addresses = ['127.0.0.1:5053', '[::1]:5053']

[static.'doh-tiarapp-dnscrypt-v4']
   stamp        =        'sdns://AQMAAAAAAAAADjE3NC4xMzguMjEuMTI4IO-WgGbo2ZTwZdg-
3dMa7u31bYZXRj5KykfN1_6Xw9T2HDIuZG5zY3J5cHQtY2VydC5$


[static.'doh-tiarapp-dnscrypt-v6']
   stamp                                                                    =
'sdns://AQMAAAAAAAAAG1syNDAwOjYxODA6MDpkMDo6NWY2ZTo0MDAxXSDvloBm6NmU8GXYPt3TGu7t9
W2GV0Y-SspHzdf-l8PU9hwyLmR$


[static.'doh-tiarapp-org-v4']
   stamp    =    'sdns://AgMAAAAAAAAADDEwNC4yOC4yOC4zNCBPtWwTIp4-T40ZbjCdyCfeStS1-
WkKW8w_WWEQubJpyQ5kb2gudGlhcmFwLm9yZwovZG5$

[static.'doh-tiar-app-v4']
   stamp                                                                    =
'sdns://AgMAAAAAAAAADjE3NC4xMzguMjkuMTc1ID4aGg9sU_PpekktVwhLW5gHBZ7gV6sVBYdv2D_aP
bg4DGRvaC50aWFyLmFwcAovZG5$


[static.'doh-tiarapp-org-v6']
   stamp   =   'sdns://AgMAAAAAAAAAGVsyNjA2OjQ3MDA6MzA6OjY4MWM6MWQyMl0gT7VsEyKePk-
NGW4wncgn3krUtflpClvMP1lhELmyackOZG9oLnR$
```

Setelah dilakukan konfigurasi pada *file* dnscrypt-proxy.toml *service* dnscrypt-proxy dapat dijalankan pada Raspberry Pi dengan perintah kode program seperti pada Tabel 4.10 dan kemudian dilakukan konfigurasi pada *admin* Pi-Hole agar menggunakan *service* dnscrypt-proxy sebagai DNS seperti pada Gambar 4.21

Tabel 4.10 Kode Program Menjalankan *Service* dnscrypt-proxy

```
# Install
$ ./dnscrypt-proxy -service install

# Start and enable
$ ./dnscrypt-proxy -service start
```



Gambar 4.21 Konfigurasi DNS Pi-Hole menggunakan dnscrypt-proxy

d. Implementasi VPN server dengan PiVPN

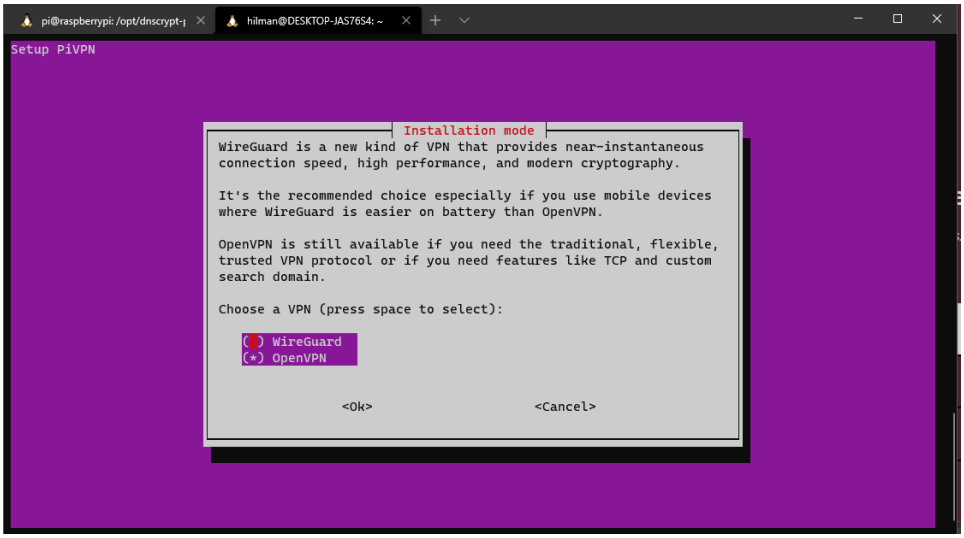
Untuk pengaksesan sistem melalui internet dapat dilakukan menggunakan VPN. Pada penelitian ini digunakan PiVPN sebagai VPN server dan OpenVPN *client*. Langkah-langkah implementasi VPN server pada Rasperry Pi pada sistem ini adalah yang pertama instalasi dan konfigurasi lalu pembuatan akun untuk akses.

Instalasi PiVPN dapat dilakukan menggunakan baris kode program seperti pada Tabel 4.10. Kode program dapat dijalankan pada terminal dan akan melakukan instalasi otomatis pada sistem.

Tabel 4.10 Kode Program Instalasi PiVPN Server

```
$ curl -L https://install.pivpn.io | bash
```

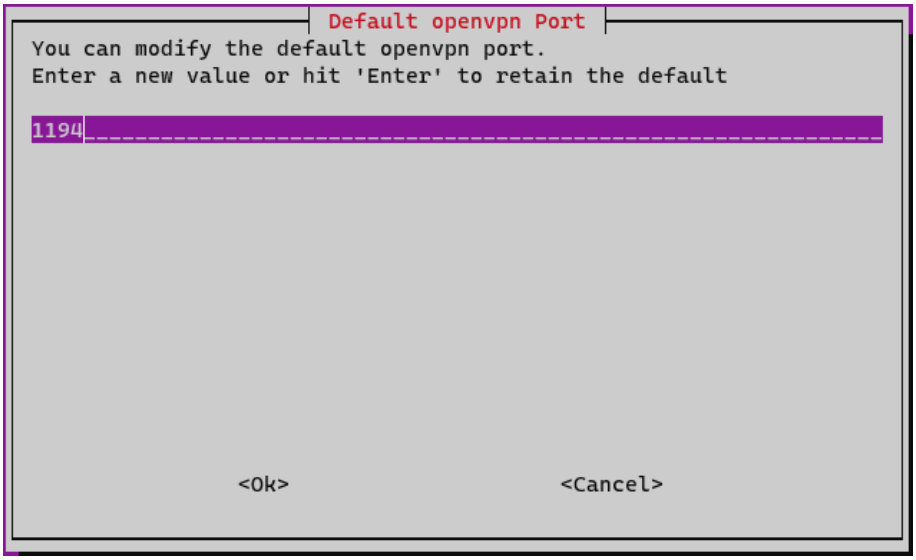
Setelah instalasi berjalan akan dilakukan konfigurasi pada VPN server, ada beberapa konfigurasi yang harus dilakukan. Hal pertama yang perlu dikonfigurasi adalah VPN *provider* yang akan digunakan. Pada penelitian ini digunakan OpenVPN sebagi VPN *provider* seperti pada Gambar 4.22



Gambar 4.22 Konfigurasi VPN *Provider*

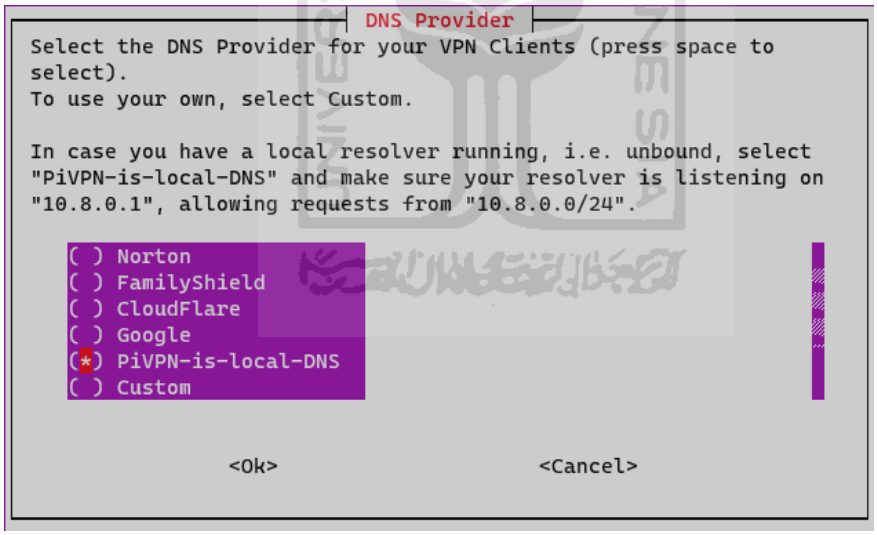
Selanjutnya melakukan pemilihan protokol yang digunakan dalam VPN *server*. Untuk VPN server yang digunakan pada penelitian ini adalah protokol UDP seperti pada Gambar 4.23. Protokol UDP dipilih karena sifatnya yang cepat dalam pengiriman paket dalam jaringan. Untuk *port* yang digunakan pada VPN ini adalah 1194 seperti pada Gambar 4.24.

Gambar 4.23 Konfigurasi Protokol VPN



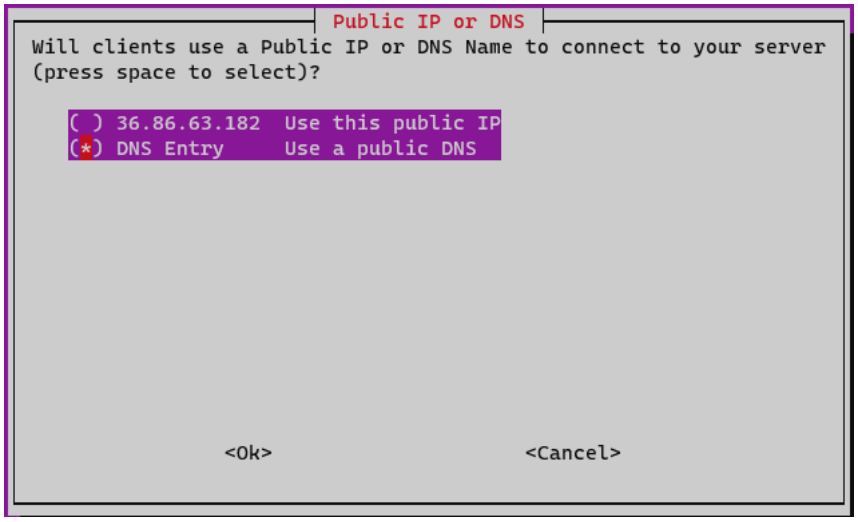
Gambar 4.24 Konfigurasi Port yang Digunakan VPN

VPN memerlukan DNS *provider* untuk koneksi VPN *client.* Untuk konfigurasi ini dipilih "PiVPN-is-local-DNS" agar VPN menggunakan Raspberry Pi sebagai DNS *resolver.* Konfigurasi DNS dapat dilihat pada Gambar 4.25



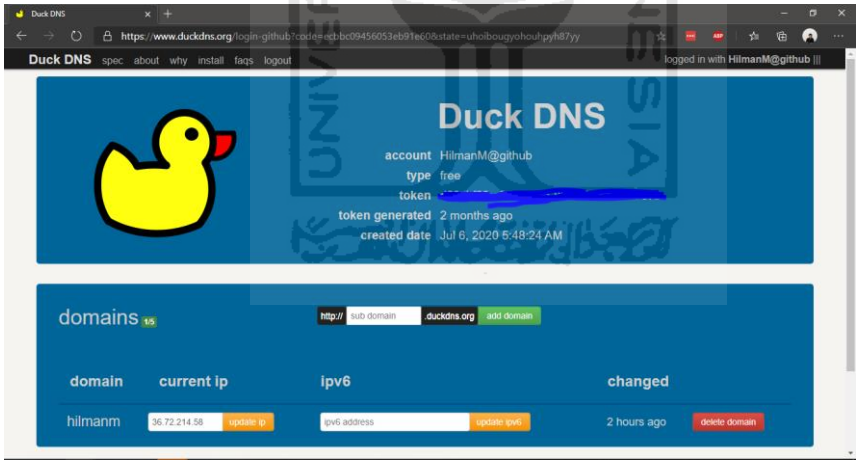
Gambar 4.25 Konfigurasi DNS Pada VPN *Server*

VPN *server* memerlukan IP *public* atau DNS *public* agar *client* dapat melakukan koneksi ke *server*. Jaringan internet dari ISP memiliki IP *public* yang berubah-ubah, oleh karena itu pada konfigurasi VPN *server* ini digunakan DNS *entry* seperti pada Gambar 4.26



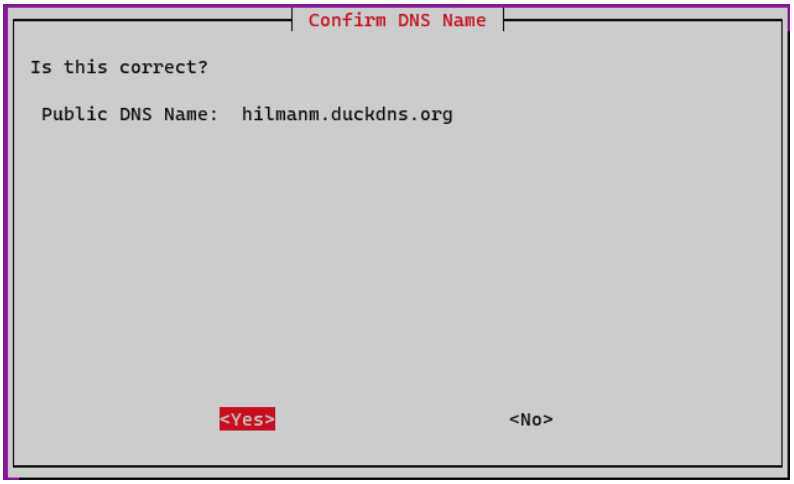
Gambar 4.27 Konfigurasi DNS *Entry* VPN

Pada penelitian ini diginakan DDNS (*Dynamic Domain Name Server).* DDNS yang digunakan adalah Duckdns. Pada halaman *website* duckdns.org didaftarkan *domain* yang ingin digunakan. Duckdns akan secara otomatis mendeteksi IP *public* dari ISP yang digunakan seperti pada Gambar 4.28.



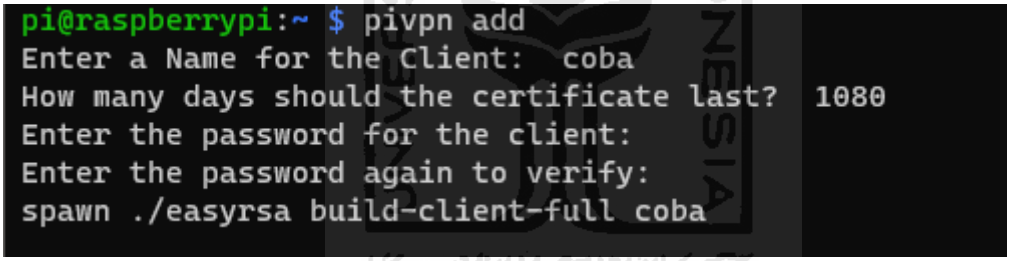
Gambar 4.29 Domain DDNS

Domain yang sudah didaftarkan pada Duckdns yang akan dimasukkan pada DNS *entry* pada VPN *server* seperti pada Gambar 4.30. Duckdns akan melakukan *update* setiap terjadi perubahan IP *public* dari ISP sehingga VPN *server* tetap bisa terkoneksi dari VPN *client* melalui DNS *entry domain* Duckdns.



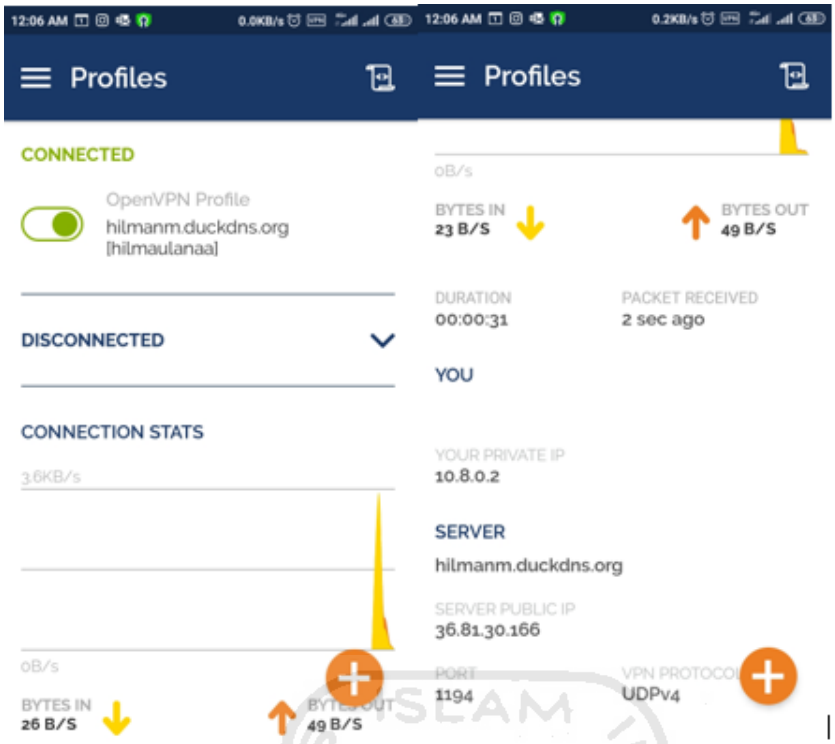
Gambar 4.30 Konfigurasi DNS Name VPN

Setelah semua konfigurasi VPN server sudah selesai dilakukan pembuatan *user client.* Pembuatan *user client* dapat dilakukan menggunakan perintah “pivpn add” pada terminal. PiVPN akan meminta memasukkan nama, *password* dan lama berlakunya seperti pada gambar 4.31.



Gambar 4.31 Pembuatan *User* VPN *Client*

User yang sudah dibuat langsung dapat dilakukan menggunakan aplikasi VPN *client* pada komputer maupun *smartphone.* Koneksi VPN dari OpenVPN *client* dari *platform* android dapat dilihat pada gambar 4.32. Perangkat *smartphone* berhasil terkoneksi ke VPN *server* menggunakan jaringan internet SIM *card.*



Gambar 4.32 Koneksi VPN *client* Dari Android

4.2. Pengujian Sistem

Pada tahap pengujian sistem dilakukan pengujian terhadap sistem yang telah dibuat. Pengujian dilakukan untuk mengetahui apakah sistem dapat berfungsi sesuai dengan perancangan yang telah dilakukan. Pada tahap pengujian diharapkan dapat mengetahui kekurangan dan kelebihan dari sistem *cloud storage* dengan konfigurasi *disk* RAID 1 dan DNS *ad-blocker* untuk perlindungan privasi data pribadi. Dilakukan beberapa tahapan pengujian dalam sistem ini, tahapan-tahapan pengujian fungsionalitas sistem berdasarkan subbab 3.2.1 adalah sebagai berikut:

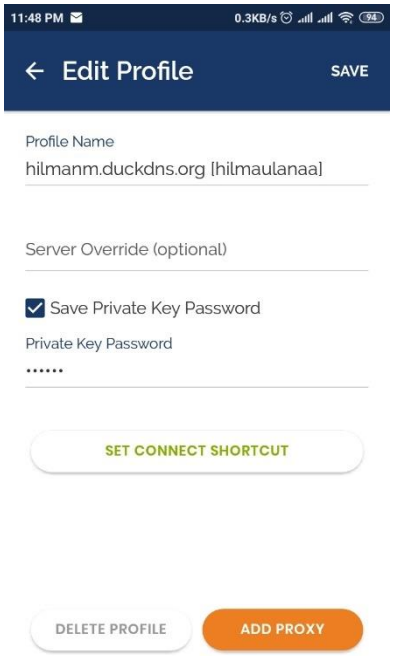
a. Melakukan uji fungsionalitas koneksi antara VPN *client* dan VPN *server* menggunakan aplikasi OpenVPN android.
b. Melakukan uji Fungsionalitas dan kecepatan *upload* serta *download* sistem *cloud server* dari internet.
c. Melakukan uji fungsionalitas *ad blocking.*

4.2.1. Uji Fungsionalitas dan Koneksi VPN Menggunakan OpenVPN

Aplikasi OpenVPN digunakan untuk melakukan koneksi dari perangkat komputer ataupun *smartphone* agar dapat terkoneksi ke sistem dari jaringan internet. Pada pengujian kali ini dilakukan menggunakan perangkat *smartphone* android dan jaringan internet SIM *card.* OpenVPN melakukan koneksi menggunakan file .ovpn yang dibuat pada *server* seperti pada Gambar 4.33.

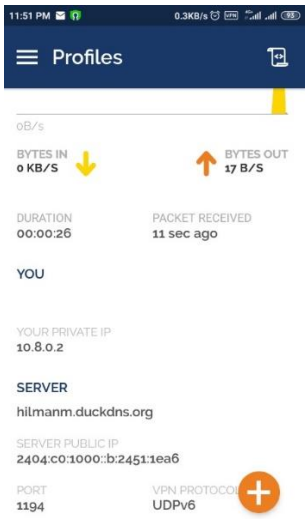
Gambar 4.33 Melakukan Koneksi Menggungakan File .ovpn

Setelah melakukan *import* file .ovpn pada OpenVPN *client, smartphone* dapat langsung melakukan koneksi ke VPN *server* dengan memasukkan *password* yang dibuat pada file .ovpn seperti pada gambar 4.34.



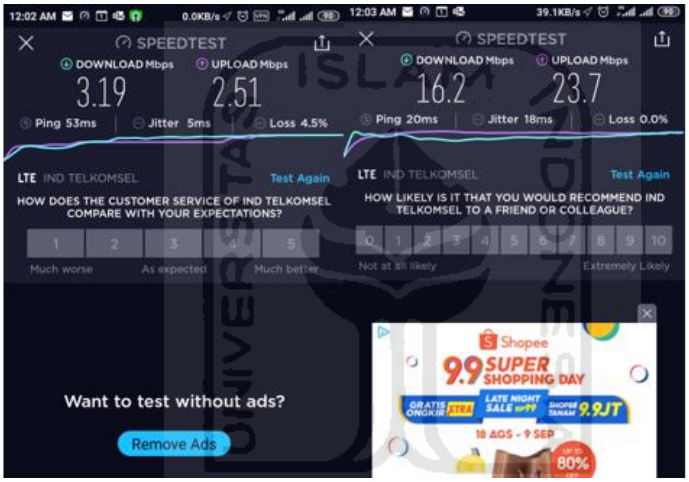
Gambar. 4.34 Memasukkan *Password* Untuk Koneksi VPN

Setelah melakukan *import file* .ovpn dan memasukkan *password* dari *user* yang digunakan *smartphone* dapat terkoneksi ke jaringan VPN. Aplikasi OpenVPN akan memberikan detail koneksi seperti berapa *bytes traffic* yang keluar dan masuk, berapa lama koneksi sudah dilakukan, alamat IP dari VPN *client*, *server* yang digunakan dan *port* serta protokol yang digunakan seperti pada Gambar 4.35.



Gambar 4.35 Informasi Koneksi OpenVPN Berhasil Terkoneksi

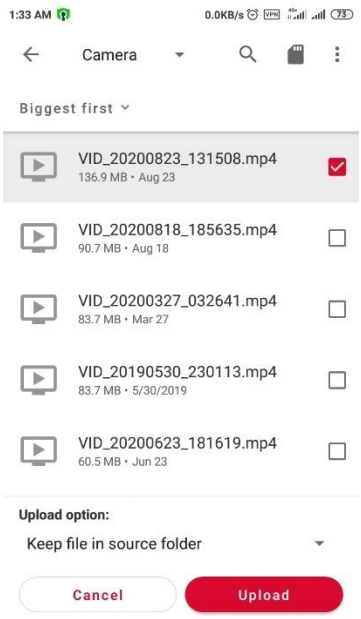
Dilakukan pengujian kecepatan internet menggunakan koneksi VPN dan tanpa koneksi VPN. Pengujian kecepatan dilakukan menggunakan aplikasi Speedtest pada *smartphone android*. Didapati terjadi penurunan kecepatan internet yang cukup signifikan saat menggunakan jaringan VPN. Penurunan kecepatan pada penggunaan VPN ini terjadi karena harus melewati dua jalur terlebih dahulu termasuk proses enkripsi yang menyebabkan *bandwith* yang ditawarkan menjadi lebih kecil (Raharjo, Fibriani, & Hadi, 2014). Sebelum menggunakan jaringan VPN kecepatan didapati kecepatan *download* 16.2 Mbps, *upload* 23.7 Mbps dengan ping 20ms. Setelah menggunakan jaringan VPN didapati kecepatan *download* 3.19 Mbps, *upload* 2.51 Mbps dengan ping 53ms. Hasil dari pengujian kecepatan internet dapat dilihat pada Gambar 4.36.



Gambar 4.36 Uji Kecepeatan Internet

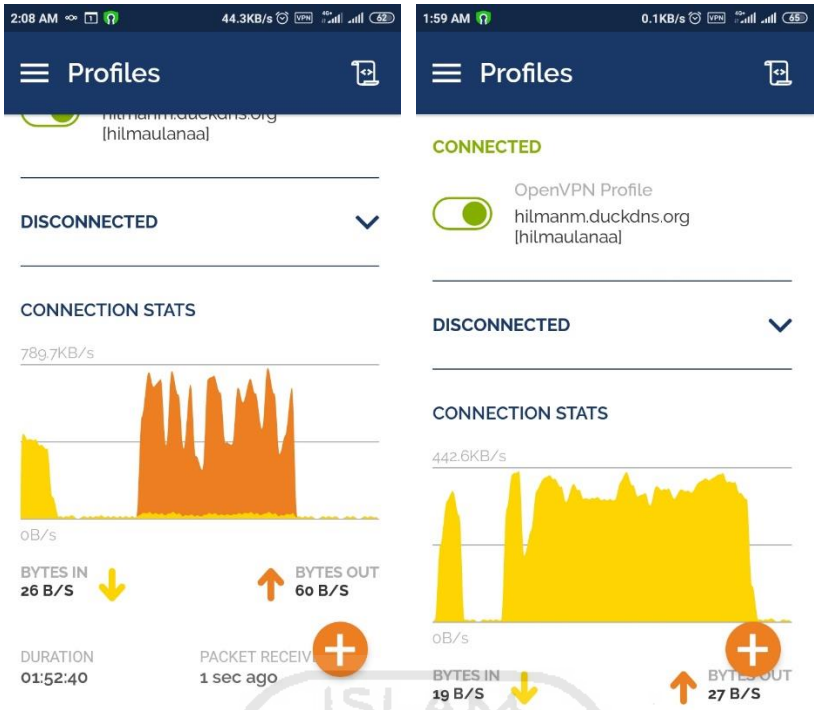
4.2.2. Uji Fungsionalitas dan Kecepatan *Upload* serta *Download Cloud Storage*

Dalam tahap pengujian ini dilakukan uji coba melakukan *download file* dan *upload file* pada sistem *cloud storage* Nextcloud. Pada tahap ini dilakukan proses *download* dan *upload* menggunakan aplikasi Nextcloud *client android.* Dalam pengujian ini dilakukan proses *upload file* berukuran 137 MB dan dilakukan perhitungan lama prosesnya. Untuk proses *download* juga dilakukan menggunakan *file* yang sama dengan proses yang sama juga seperti pada Gambar 4.37.



Gambar 4.37 Tampilan Upload File

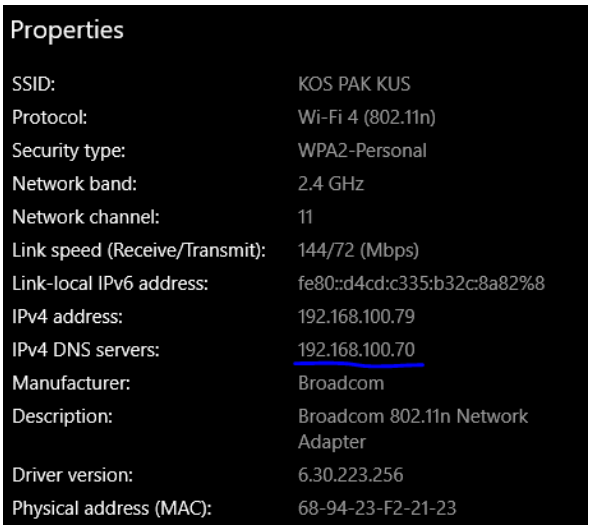
Proses *upload* yang dilakukan menggunakan *file* yang disebutkan di atas dengan kecepatan internet yang sudah diuji sebelumnya memakan waktu 6 menit 45 detik. Sedangkan untuk proses *upload* membutuhkan waktu 6 menit 51 detik. Dengan maksimal kecepatan *upload* adalah 798.7 KB/s dan *download* 442.6 KB/s seperti pada data di Gambar 4.38. Kecepatan dihasilkan dipengaruhi oleh koneksi yang menggunakan VPN. Kecepatan dan *bandwith* menurun saat melakukan proses *upload* atau *download* namun keamanan data yang dikirim menjadi lebih tinggi karena setiap prosesnya dilakukan enkripsi.



Gambar 4.38 Maksimal *Upload* dan *Download Cloud Storage*

4.2.3. Uji Fungsional *Ad Blocking*

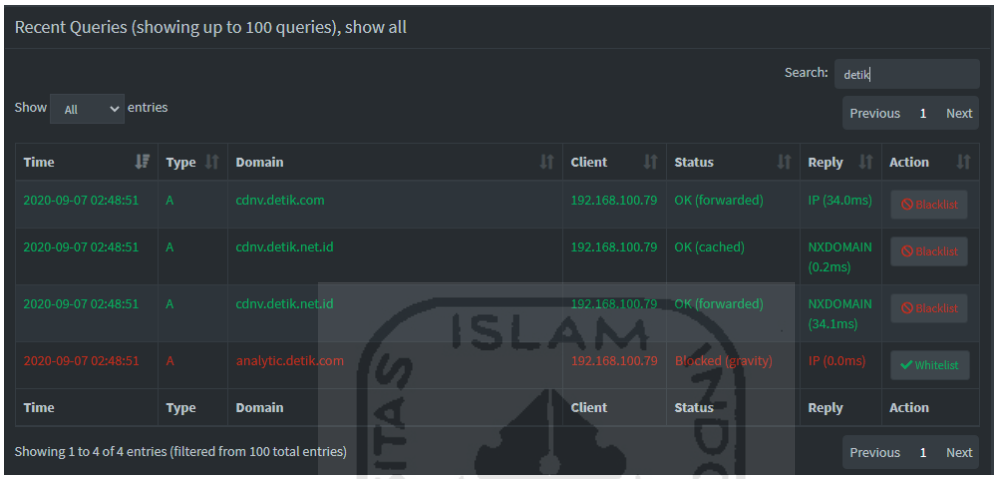
Tahap pengujian fungsional *ad blocking* dilakukan dengan mengunjungi situs menggunakan *web browser* yang memuat iklan dalam halamannya. Dalam pengujian ini dilakukan menggunakan tiga buah situs web yaitu detik.com, kompas.com dan youtube.com. Dalam pengujian ini digunakan perangkat *laptop* yang sudah terkoneksi internet dan menggunakan DNS Raspberry Pi seperti detail pada Gambar 4.39.



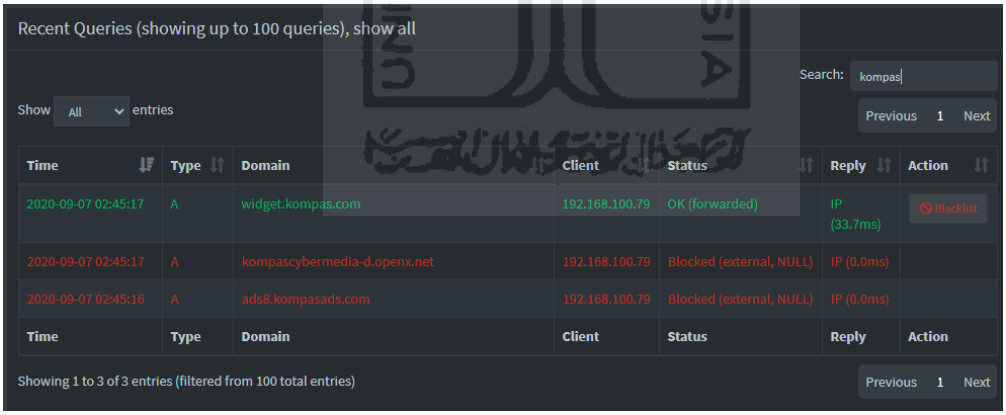
Gambar 4.39 Detail Koneksi Perangkat *Laptop*

Setelah perangkat *laptop* terkoneksi dilakukan pengaksesan ke situs web yang sudah disebutkan. Untuk mengetahui efektivitas pemblokiran akan dilihat dari *log query* DNS yang tercatat pada halaman admin Pi-Hole dan tangkapan layar dari halaman *website*.

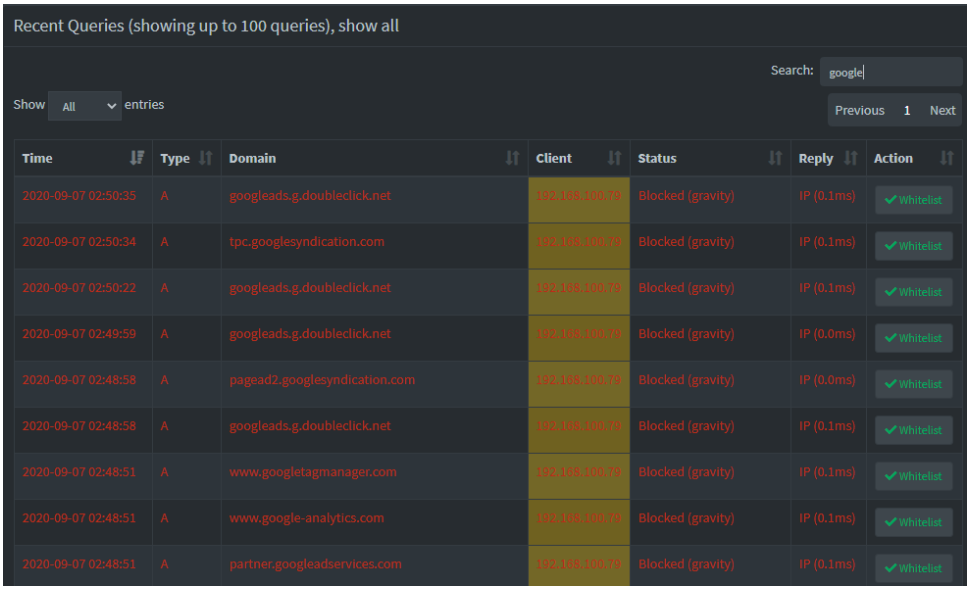
Untuk *log query* domain yang terblokir dapat terlihat pada Gambar 4.40 untuk situs detik.com, Gambar 4.41 untuk situs kompas.com dan Gambar 4.42 Untuk situs youtube.com yang tertangkap selama pengaksesan situs di halaman admin Pi-Hole.



Gambar 4.40 *Log Query* situs detik.com



Gambar 4.41 *Log Query* situs kompas.com



Gambar 4.42 *Log Query Situs youtube.com*

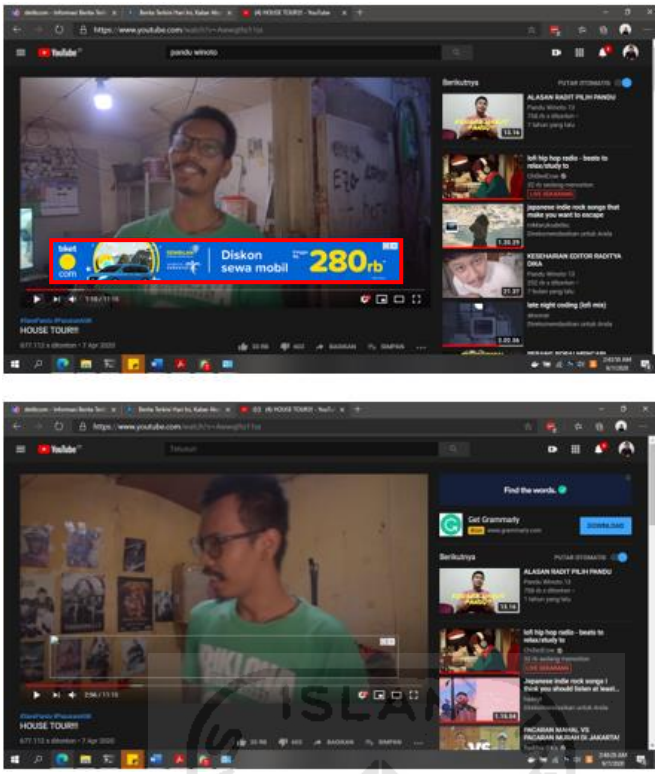
Berikut di bawah ini adalah hasil tangkapan layar dari situs *web* yang menggunakan dan tanpa menggunakan DNS Pi-Hole. Gambar 4.43 memperlihatkan perbedaan tampilan dari situs *web* detik.com di mana setelah menggunakan Pi-Hole iklan *banner* yang sebelumnya terlihat menonjol di halaman menghilang.

Gambar 4.43 Halaman Situs *Web* detik.com

Sama halnya dengan situs web detik.com, halaman situs kompas.com yang sebelumnya menampilkan iklan *banner* besar pada halaman situs *web* setelah menggunakan Pi-Hole iklan *banner* yang sebelumnya muncul jadi menghilang seperti pada Gambar 4.44.

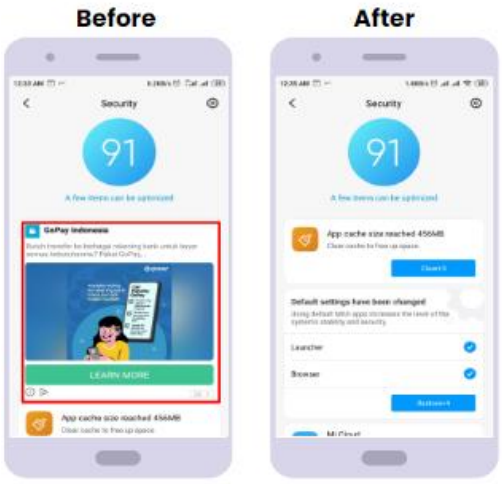
Gambar 4.44 Halaman situs *web* kompas.com

Pada halaman situs youtube.com sedikit berbeda dari dua situs yang sudah diuji sebelumnya. Di halaman youtube.com beberapa iklan masi terlewat dari blokir seperti iklan di awal video dan beberapa iklan sudah hilang namun masih tersisa *frame* dari iklan di halaman namun gambarnya tidak muncul seperti pada Gambar 4.45



Gambar 4.45 Halaman Situs *Web* youtube.com

Terakhir dilakukan pengujian pemblokiran iklan dalam aplikasi di perangkat *smartphone* android. Pada Gambar 4.46 dapat dilihat pada aplikasi *Security* Xiaomi yang sebelumnya pada tampilannya memuat iklan, namun setelah berpindah ke jaringan yang menggunakan *ad-blocker* iklan dapat dihilangkan.



Gambar 4.6 Iklan Dalam Aplikasi

4.3. Evaluasi Sistem

Pada tahap analisis penelitian dilakukan penilaian terhadap kelebihan dan kekurangan yang dimiliki oleh sistem yang sudah dibuat setelah dilakukan tahap pengujian. Kekurangan dan kelebihan dari penelitian ini akan menjadi pertimbangan untuk penulis maupun pembaca dalam menilai keberhasilan dari penelitian ini. Kekurangan dalam penelitian ini dapat menjadi acuan untuk pengembangan pada penelitian selanjutnya agar apa yang sudah dicapai dapat dikembangkan lebih jauh lagi. Berikut adalah kelebihan dan kekurangan dari hasil penelitian ini:

4.3.1. Kelebihan Sistem

Berdasarkan pembahasan dalam pengujian sistem oleh peneliti yang sudah dilakukan ditemukan beberapa kelebihan sistem sebagai berikut:

a. Sistem dapat terkoneksi kapan pun dan di mana pun melalui jaringan internet.
b. Sistem lebih aman karena terkoneksi dalam jaringan privat melalui VPN.
c. Kapasitas penyimpanan dapat di tambahkan sesuai kebutuhan.
d. Sistem dapat melakukan proses *upload* dan *download* serta melihat *file* di semua platform.
e. Antarmuka yang ditampilkan sistem sangat mudah digunakan.
f. Sistem dapat memblokir iklan tanpa menggunakan ekstensi tambahan pada perangkat pengguna, hanya perlu melakukan konfigurasi DNS otomatis pada DHCP *server access point.*

4.3.2. Kekurangan Sistem

Berdasarkan pembahasan dalam pengujian sistem yang sudah dilakukan ditemukan beberapa kekurangan sistem sebagai berikut:

a. Kecepatan internet mengalami penurunan saat menggunakan VPN.
b. Sistem masih menggunakan *flash disk* yang kecepatan *write* dan *write*-nya cukup kecil.
c. Beberapa situs tidak dapat terblokir iklannya karena teknologi yang digunakan sudah berkembang.

BAB V
KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil perancangan dan implementasi serta pengujian yang telah dilakukan pada penelitian ini, terdapat beberapa kesimpulan antara lain adalah sebagai berikut:

Sistem dapat berfungsi sebagai *cloud storage* melakukan proses *upload, download* dan melihat *file* melalui berbagai *platform* yang tersedia dengan media penyimpanan yang aman.

a. Sistem dapat melakukan pemblokiran terhadap iklan di situs *web* dengan aman tanpa aplikasi tambahan pada penggunanya.
b. Sistem dapat diakses dengan aman dari jaringan internet dengan menggunakan VPN yang dibuat.
c. Sistem dapat dikonfigurasi sesuai kebutuhan pengguna seperti kapasitas *storage* dan *domain* yang hendak diblokir.
d. Sistem ini bertujuan dapat menyediakan layanan *cloud storage* dan *ad-blocker* yang aman bagi penggunanya sesuai dengan tujuan yang telah diharapkan.
e. Sistem ini memiliki antarmuka yang mudah digunakan pengguna.
f. Sistem dapat melakukan proses *upload* dan *download* serta melihat *file* di semua platform

5.2. Saran

Setelah membahas tentang kelebihan dan kekurangan Sistem Mobile Private Network Storage Dan Dns Ad-Blocker Untuk Perlindungan Privasi Data Pribadi. yang sudah dibuat, maka diharapkan pada penelitian selanjutnya dapat melakukan pengembangan atas saran yang diberikan pada penelitian ini. Saran untuk pengembangan sistem ini pada penelitian selanjutnya adalah:

a. Menggunakan perangkat Raspberry Pi yang lebih baru agar dapat meningkatkan kecepatan akses ke sistem.
b. Membuat semua *service* yang berjalan pada Raspberry Pi berjalan sebagai docker *container* untuk kemudahan manajemen dan konfigurasi.
c. Menggunakan media penyimpanan dengan kapasitas yang lebih besar dan dengan kecepatan lebih tinggi seperti SSD.
d. Berkala melakukan *update* pada data blokir untuk situs yang sudah tidak terblokir.

DAFTAR PUSTAKA

Adiputra, F. (2015). *CONTAINER DAN DOCKER : TEKNIK VIRTUALISASI DALAM PENGELOLAAN BANYAK*. *4*(3).

Atmojo, Y. P. (2015). *Pemanfaatan Single-Board Computer pada Sistem Pengukur Suhu Ruangan : Studi Kasus Ruang Server STMIK STIKOM Bali*. 9–10.

Harsapranata, A. I. (2015). *Implementasi failover menggunakan jaringan vpn dan metronet pada astridogroup indonesia*.

Lanka, A. (2018). *Remotely Accessible , Low Power Network Attached Storage Device*. (Icicct), 1083–1088.

M. Nourman Hadi Satrio H, Nugroho Suharto, Martono Dwi Atmadja. (2019). *IMPLEMENTASI PROTOKOL JARINGAN OPENVPN DAN TEKNOLOGI REDUNDANT ARRAY OF INDEPENDENT DISK LEVEL 1 (RAID-1) PADA FILE*. *1*, 40–46.

Parra-arnau, J., Rodr, A., Parra-arnau, J., & Rodr, A. (2016). *Online Advertising : Analysis of Privacy Threats and AC PT*. https://doi.org/10.1016/j.comcom.2016.12.016

Raharjo, B. T., Fibriani, I., & Hadi, W. (2014). *PENGUKURAN QUALITY OF SERVICE (QoS) TERHADAP KUALITAS VIDEO CONFERENCE PADA VIRTUAL PRIVATE NETWORK (VPN) (MEASUREMENT OF QUALITY OF SERVICE (QoS) ON THE QUALITY OF VIDEO CONFERENCE ON VIRTUAL PRIVATE NETWORK (VPN))*.

Rolon, J., & Background, A. (2019). *SpartanShield : A Layered Defense Against Malvertising*. 185–190. https://doi.org/10.1109/CSCI49370.2019.00038

Santi, D., Rumani, R. M., & Purwanto, Y. (2013). *IMPLEMENTASI DAN ANALISIS PERFORMANSI RAID PADA DATA STORAGE INFRASTRUCTURE AS A SERVICE (IAAS) CLOUD COMPUTING*. *14*(2), 99–107.

Shrivastava, A. (2017). *Home Server and NAS using Raspberry Pi*. 2270–2275.

Suharta, K. (n.d.). *RANCANG BANGUN PERSONAL CLOUD STORAGE Karuniawan Suharta*.

Taib, A. M., Fikri, M., Ishak, H., Kamarudin, N. K., Darus, M. Y., Azira, N., & Radzi, M. (2020). *Securing Network Using Raspberry Pi by Implementing VPN , Pi-Hole , and IPS (VPiSec)*. *9*(1), 457–464.

Networks, O. (2015). How Much Does Cloud Storage Cost? Retrieved September 21, 2020, from https://resource.optimalnetworks.com/blog/2015/02/08/cost-cloud-storage

Abel, R. (2019, April 16). *SC Magazine Security News*. Retrieved from SC Magazine: https://www.scmagazine.com/home/security-news/vulnerabilities/independent-security-researcher-armin-sebastian-discovered-a-vulnerability-in-adblock-plus-which-can-allow-hackers-to-read-a-victims-gmail/

transiskom.com. (2016, March 30). *transiskom*. Retrieved from transiskom: https://www.transiskom.com/2016/03/pengertian-studi-

kepustakaan.html#:~:text=Studi%20kepustakaan%20adalah%20kegiatan%20untuk,masalah%20yang%20menjadi%20obyek%20penelitian.&text=Dengan%20melakukan%20studi%20kepustakaan%2C%20peneliti,pemikiran%20yang%20rele

Upton, E. (2016, September 8). *Raspberry Pi Blog*. Retrieved from Raspberry Pi Blog: https://www.raspberrypi.org/blog/ten-millionth-raspberry-pi-new-kit/

G. (2019). Global requests for user information. Retrieved October 30, 2020, from https://transparencyreport.google.com/user-data/overview?user_requests_report_period=series:requests,accounts;authority:ID;time:&lu=user_requests_report_period

LAMPIRAN