

Pengembangan Sistem Pemantauan Proses Backup Data Terdistribusi Menggunakan Owncloud Berbasis Web

Reshi Haryanzi^{#2}, Taufik Fuadi Abidin^{#2}, Ramzi Adriman^{#1,2}

^{#1}Jurusan Teknik Elektro dan Komputer, Fakultas Teknik, UniversitasSyiah Kuala

^{#2}Magister Teknik Elektro, Fakultas Teknik, UniversitasSyiah Kuala

Jl. Tgk. SyechAbdurrauf No. 7, Darussalam, Banda Aceh, Indonesia

reshi@unsyiah.ac.id

Abstrak— Universitas Syiah Kuala (Unsyiah) merupakan perguruan tinggi yang menerapkan dan memanfaatkan teknologi informasi dan komunikasi (TIK) sebagai salah satu kebutuhan pokok dalam menunjang aktifitas akademika. Teknologi informasi tersebut harus dijamin tetap aman, tersedia dan berjalan perlu dilakukan proses backup secara rutin. Proses backup dilakukan unsyiah menggunakan owncloud. Setiap aktifitas dari proses backup harus dipantau supaya berjalan sesuai rencana dan targetnya.

Pemantauan proses backup data dilakukan unsyiah saat ini secara manual dimana admin harus masuk satu persatu ke server melalui terminal. Semakin banyak sistem informasi dan database yang dibackup maka semakin besar waktu yang

I. PENDAHULUAN

Penerapan dan pemanfaatan teknologi informasi dan komunikasi (TIK) di Lingkungan perguruan tinggi baik negeri maupun swasta sudah menjadi salah satu kebutuhan pokok dalam aktifitas akademika. Untuk menjamin agar teknologi informasi yang ada tetap aman, tersedia dan berjalan perlu dilakukan proses backup secara rutin. setiap aktifitas dari proses *backup* harus dipantau supaya proses *backup* berjalan sesuai rencana dan targetnya. Pemantauan dapat memberikan informasi keberlangsungan proses untuk menetapkan langkah menuju ke arah perbaikan yang berkesinambungan.

Universitas Syiah Kuala (Unsyiah) sebagai salah satu perguruan tinggi negeri memiliki dan telah menjalankan proses *backup* secara rutin *serta disaster recovery plan*. Proses backup dilakukan dengan cara membuat salinan dari database dan aplikasi dari perangkat server tempat aplikasi itu berjalan ke perangkat khusus *backup*. Proses backup diinisialisasi oleh aplikasi crontab yang akan *men-trigger* proses penyalinan, sedangkan *tingger* dipicu setiap 24 jam. Dengan adanya *proses backup*, jika terjadi kegagalan hardware atau software maka kondisi terburuk adalah kehilangan data 24 jam terakhir. Mirror dari web utama ditempatkan pada perangkat yang secara fisik berada di luar Unsyiah. Web utama dimirror memiliki format, tampilan, isi dan data yang sama persis dengan yang ada pada data center UPT TIK[1]. Namun dalam proses pemantauan *backup* dilakukan setiap hari secara manual dimana admin harus

dibutuhkan untuk proses pemantauan. Supaya proses pemantauannya optimal dan efisien, maka diperlukan sebuah sistem pemantauan proses *backup*. Sistem pemantauan proses backup data merupakan sistem informasi dikembangkan menggunakan metode waterfall. Sistem informasi tersebut dapat memberikan informasi hasil dan besar file backup serta tersedia notifikasi berbasis email tentang status backup. Hasil pengujian sistem informasi tersebut diketahui fungsionalitas dapat berjalan dengan baik.

Kata Kunci : *Owncloud*, pemantauan, *backup*, sistem informasi, *waterfall*, notifikasi.

masuk satu persatu ke server melalui terminal untuk melihat jumlah dan ukuran file yang dikirim, selanjutnya masuk ke server *backup* utama melihat ukuran dan jumlah file apakah sesuai dan berhasil *dibackup*. Semakin banyak sistem informasi dan database yang dibackup semakin besar waktu yang dibutuhkan untuk proses pengecekan.

Untuk memudahkan proses pemantauan *backup* yang efektif, sederhana dan mudah dimengerti maka dibutuhkan sistem pemantauan *backup data*. Penelitian ini bertujuan untuk mengembangkan sistem pemantauan *backup data* terdistribusi menggunakan *owncloud* berbasis web sehingga memberikan informasi proses *backup* data secara detail. Seperti mengetahui apakah proses backup berhasil, mengetahui ukuran besar file yang *dibackup* serta memberitahukan informasi keberhasilan *backup* ke pada admin *backup* melalui email.

II. LANDASAN TEORI

2.1 Sistem Pemantauan

Sistem adalah suatu jaringan kerja dari berbagai prosedur yang saling berhubungan untuk mencapai tujuan sasaran tertentu bersama-sama kearah yang baik. Pemantauan didefinisikan sebagai proses pengumpulan data dari berbagai sumber daya. Pengumpulan data-data biasanya dilakukan secara real time[7][8].

Informasi yang diberikan oleh pemantauan merupakan informasi berkelanjutan kearah perbaikan dan berkesinambungan. Proses pemantauan pada pelaksanaannya

dilakukan secara realtime [9]. Beberapa macam alat bantu pada sistem pemantauan seperti aplikasi visual, dokumentasi dan observasi langsung [10].

Dua fungsi dasar Pemantauan yaitu *performance* pemantauan dan *compliance* Pemantauan [8]. *Performance* pemantauan berfungsi untuk mengetahui perkembangan organisasi dalam pencapaian target yang diharapkan sedangkan *Compliance* pemantauan berfungsi untuk memastikan proses sesuai dengan harapan/rencana. Umumnya, hasil pemantauan berupa *progress report* proses. *Output* tersebut diukur secara deskriptif maupun non-deskriptif. *Output* pemantauan bertujuan untuk mengetahui kesesuaian progress telah berjalan. *Output* pemantauan berguna pada perbaikan mekanisme proses/kegiatan dimana pemantauan dilakukan.

2.1.1 Efektifitas Sistem Pemantauan

Sistem pemantauan akan memberikan dampak yang baik bila dirancang dan dilakukan secara efektif. Berikut kriteria sistem pemantauan yang efektif [8]:

1. Sederhana dan mudah dimengerti (*user friendly*). pemantauan harus dirancang dengan sederhana namun tepat sasaran. Konsep yang digunakan adalah singkat, jelas, dan padat. Singkat berarti sederhana, jelas berarti mudah dimengerti, dan padat berarti bermakna (berbobot).
2. Fokus pada beberapa indikator utama. Indikator diartikan sebagai titik kritis dari suatu *scope* tertentu. Banyaknya indikator membuat pelaku dan obyek pemantauan tidak fokus. Hal ini berdampak pada pelaksanaan sistem tidak terarah. Maka itu, fokus diarahkan pada indikator utama yang benar-benar mewakili bagian yang dipantau.
3. Perencanaan matang terhadap aspek-aspek teknis. Tujuan perancangan sistem adalah aplikasi teknis yang terarah dan terstruktur. Maka itu, perencanaan aspek teknis terkait harus dipersiapkan secara matang. Aspek teknis dapat menggunakan pedoman 5W1H, meliputi apa, mengapa, siapa, kapan, di mana dan bagaimana pelaksanaan sistem pemantauan.
4. Prosedur pengumpulan dan penggalian data. Selain itu, data yang didapatkan dalam pelaksanaan pemantauan pada *on going process* harus memiliki prosedur tepat dan sesuai. Hal ini ditujukan untuk kemudahan pelaksanaan proses masuk dan keluarnya data. Prosedur yang tepat akan menghindari proses *input* dan *output* data yang salah (tidak akurat).

2.1.2 Tujuan Sistem Pemantauan

Terdapat beberapa tujuan sistem pemantauan. Tujuan sistem pemantauan dapat ditinjau dari beberapa segi, misalnya segi obyek dan subyek yang dipantau, serta hasil dari proses pemantauan itu sendiri. Adapun beberapa tujuan dari sistem pemantauan yaitu [11] yaitu:

1. Memastikan suatu proses dilakukan sesuai prosedur yang berlaku. Sehingga, proses berjalan sesuai jalur yang disediakan (*on the track*).
2. Menyediakan probabilitas tinggi akan keakuratan data bagi pelaku pemantauan
3. Mengidentifikasi hasil yang tidak diinginkan pada suatu proses dengan cepat (tanpa menunggu proses selesai).
4. Menumbuh kembangkan motivasi dan kebiasaan positif pekerja.

2.2 Backup Data

Untuk menjaga integritas sistem dan data pengguna yang aman dan baik perlu dilakukan proses backup dan recovery. *Hardware* atau *software failure* merupakan penyebab terjadinya kehilangan dan kerusakan dalam sistem, baik disengaja atau tidak disengaja misalnya bencana alam, listrik tidak stabil, hardisk rusak dan lain sebagainya. Proses backup dijalankan agar mendapatkan salinan data sehingga jika ada data yang rusak dan hilang dapat melakukan recovery ulang [12].

Dalam sistem TI modern, proses backup dan pemulihan data sangat penting untuk memberikan perlindungan terhadap kehilangan data baik dari kejadian alami maupun buatan manusia. Menurut ISO 27001, backup penting dilakukan guna untuk mempertahankan integritas dan ketersediaan informasi dan fasilitas pengolahan informasi. Mengurangi resiko data hilang, jika sudah dibackup, informasi mengenai data tersebut masih bisa diakses karena memiliki copy lain dari data tersebut.

Tujuan dari *backup* data adalah:

1. Tersedianya data dengan cepat setelah terjadi insiden keamanan seperti *hardware failure*, *application error*, dan sebagainya.
2. Memudahkan pemindahan data ke lokasi tujuan yang membutuhkan. Misalkan kantor pusat mengalami bencana alam maka dapat dipindahkan ke disaster recovery site.
3. Memenuhi regulasi yang mengharuskan data disimpan selama sepuluh tahun.

2.3 Cloud Computing

Cloud computing adalah paradigma komputasi, sistem yang di gunakan sebagian besar terhubung dalam jaringan publik dan privat, menyediakan serangkaian sumberdaya komputasi bersama yang dinamis dan terukur untuk penyimpanan file dan aplikasi data, melalui layanan jaringan internet dengan pola akses remote [14].

2.3.1 Model Cloud

Ada 3 model *cloud computing* berdasarkan infrastruktur [15]:

1. Public Cloud

Public cloud adalah sebuah layanan infrastruktur yang di sediakan oleh provider. Layanan ini berbayar sesuai dengan kebutuhan pengguna. Semua media penyimpanan dan aplikasi disediakan

oleh penyedia layanan. Contoh dari *public cloud* yang populer adalah *Microsoft's Azure Service Platform*, *Rackspace Cloud Suite*, dan *Amazon AWS* (EC2, S3 dll).

2. *Private Cloud*

Private cloud dibangun khusus untuk sebuah perusahaan yang bertujuan untuk mengatasi kekhawatiran pada keamanan data. *Private cloud* menawarkan tingkat kontrol dan keamanan yang tinggi lebih baik dari *public cloud*. Manfaat dari *private cloud* dapat mengurangi biaya dan keamanan data terjamin, hemat bandwidth.

3. *Hybrid Cloud*

Hybrid cloud adalah gabungan model *public cloud* dan *private cloud*.

2.4 *Owncloud*

Owncloud yang merupakan salah satu perangkat lunak berbagi berkas gratis dan bebas seperti [Dropbox](#), menyediakan pengamanan yang baik, memiliki tata cara yang baik bagi pengguna aplikasi untuk membagi dan mengakses data yang secara lancar terintegrasi dengan perangkat teknologi informasi yang tujuannya mengamankan, melacak, dan melaporkan penggunaan data. *Owncloud* menempatkan kontrol kepada pengguna teknologi informasi itu sendiri dan juga menawarkan penyedia layanan, pusat dan bagian transmisi yang berfungsi untuk menyediakan solusi sinkronisasi dan berbagi bagi pengguna. *Owncloud* memberikan akses terhadap berkas-berkas secara universal dengan menggunakan antarmuka jaringan [16].

Fungsi *owncloud storage* merupakan layanan berbasis internet yang bisa kita akses dari mana saja. Contoh layanan internet storage online saat ini *Dropbox*, *Google Drive*, dan lainnya. *Owncloud* sendiri sudah menggunakan berbagai *platform* contohnya *linux*, *windows* dan lain-lain. *Owncloud* termasuk dalam kategori *Infrastruktur as a Service* (IaaS), layanan awan. *Owncloud* dapat melakukan sinkronisasi file dengan menggunakan perangkat *mobile desktop* atau dengan menggunakan web. *Private Cloud storage* memungkinkan akses data oleh pengguna dari semua tempat, setiap waktu, sehingga pekerjaan dapat dilakukan dimana saja. *Owncloud* merupakan *open source* yang dirancang untuk layanan *cloud storage*, dengan *owncloud* pengguna dapat mengakses data melalui antar muka web atau aplikasi *client* [16].

Cloud Storage merupakan media penyimpanan yang dalam pengaksesannya memerlukan jaringan internet. Namun tentu saja filenya berada di komputer dimana harus membuat akun *cloud storage* terlebih dahulu. Contoh kecilnya jika kita seorang web designer maka kita memerlukan file-file yang berupa gambar, font, flash, file tutorial dll. Untuk memudahkan dalam pengerjaannya maka sebaiknya menggunakan media penyimpanan *cloud storage*. Karena *cloud storage* ini dapat mengefesienkan waktu dan

tempat dibandingkan menggunakan media penyimpanan seperti flashdisk maupun CD [17].

2.5 Uji pada Rekayasa Perangkat Lunak

Dalam pembuatan perangkat lunak tentunya sering menghadapi "*bug*" atau kesalahan "*error*" pada proses-proses tertentu. Untuk menghindari banyaknya *bug* maka diperlukan pengujian perangkat lunak, sebelum perangkat lunak yang telah dibuat diberikan ke pelanggan atau selama perangkat lunak masih terus dikembangkan. Pada dasarnya pentingnya pengujian perangkat lunak mengacu pada kualitas perangkat lunak tersebut [18].

Berikut tahapan pengujian perangkat lunak yaitu [18]:

1. Pengujian Unit / Unit Testing

Unit Testing adalah metode verifikasi perangkat lunak di mana programmer menguji suatu unit program layak untuk tidaknya dipakai. Unit testing ini fokusnya pada verifikasi pada unit yang terkecil pada desain perangkat lunak (komponen atau modul perangkat lunak). Karena dalam sebuah perangkat lunak banyak memiliki unit-unit kecil maka untuk mengujinya biasanya dibuat program kecil atau main program) untuk menguji unit-unit perangkat lunak. Unit-unit kecil ini dapat berupa prosedur atau fungsi, sekumpulan prosedur atau fungsi yang ada dalam satu file jika dalam pemrograman terstruktur, atau kelas, bisa juga kumpulan kelas dalam satu package dalam PBO. Pengujian unit biasanya dilakukan saat kode program dibuat.

2. Pengujian Integrasi

Pengujian integrasi lebih pada pengujian penggabungan dari dua atau lebih unit pada perangkat lunak. Pengujian integrasi sebaiknya dilakukan secara bertahap untuk menghindari kesulitan penelusuran jika terjadi kesalahan *error / bug*.

3. Pengujian Sistem

Unit-unit proses yang telah diintegrasikan diuji dengan antarmuka yang sudah dibuat sehingga pengujian ini dimaksud untuk menguji sistem perangkat lunak. Perlu diingat bahwa pengujian sistem harus dilakukan secara bertahap sejak awal pengembangan, jika pengujian hanya diakhir maka dapat dipastikan kualitas sistemnya kurang bagus.

4. Pengujian Penerimaan

Pengujian penerimaan perangkat lunak dilakukan oleh pengguna yang telah bekerja sama dengan pembuat program guna untuk mengetahui secara langsung bagaimana perangkat lunak yang telah dibuat dapat bekerja sebelum perangkat lunak yang dibuat disebar luaskan. Pengujian penerimaan ini bertujuan untuk mengetahui kepuasan pengguna atau *user*.

2.6 *White box* dan *Black box tes*

Ada beberapa jenis pengujian perangkat lunak, antara lain [19]:

1. Pengujian white box adalah pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara prosedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Secara sekilas dapat diambil kesimpulan white box testing merupakan petunjuk untuk mendapatkan program yang benar secara 100%.
2. *Black-Box Testing* merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program.

2.6.1 White Box Testing

White Box Testing adalah salah satu cara untuk menguji suatu aplikasi atau *software* dengan cara melihat modul untuk dapat meneliti dan menganalisa kode dari program yang di buat ada yang salah atau tidak. Kalau modul yang telah dan sudah di hasilkan berupa output yang tidak sesuai dengan yang di harapkan maka akan dikompilasi ulang dan di cek kembali kode-kode tersebut hingga sesuai dengan yang diharapkan [20].

Kasus yang sering menggunakan *white box testing* akan di uji dengan beberapa tahapan yaitu:

1. Pengujian seluruh keputusan yang menggunakan logikal.
2. Pengujian keseluruhan *loop* yang ada sesuai batasan-batasannya.
3. Pengujian pada struktur data yang sifatnya internal dan yang terjamin validitasnya.

Kelebihan White Box Testing antara lain [20] :

1. Kesalahan Logika
Menggunakan syntax 'if' dan syntax pengulangan. Langkah selanjutnya metode white box testing ini akan mencari dan mendeteksi segala kondisi yang di percaya tidak sesuai dan mencari kapan suatu proses perulangan di akhiri.
2. Ketidaksesuaian Asumsi
Menampilkan dan memonitor beberapa asumsi yang diyakini tidak sesuai dengan yang diharapkan atau yang akan diwujudkan, untuk selanjutnya akan dianalisa kembali dan kemudian diperbaiki.
3. Kesalahan Pengetikan
Mendeteksi dan mencari bahasa-bahasa pemrograman yang di anggap bersifat case sensitif. Kelemahan White Box Testing adalah pada perangkat lunak yang jenisnya besar, metode white box testing ini dianggap boros karena melibatkan banyak sumberdaya untuk melakukannya [20].

2.6.2 Black Box Testing

Box Testing berfokus pada spesifikasi fungsional dari perangkat lunak. Tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program.

Black Box Testing bukanlah solusi alternatif dari *White Box Testing* tapi lebih merupakan pelengkap untuk menguji hal-hal yang tidak dicakup oleh *White Box Testing*. *Black Box Testing* cenderung untuk menemukan hal-hal berikut [20]:

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*interface errors*).
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi (*performance errors*).
5. Kesalahan inisialisasi dan terminasi.

Pengujian didesain untuk menjawab pertanyaan berikut:

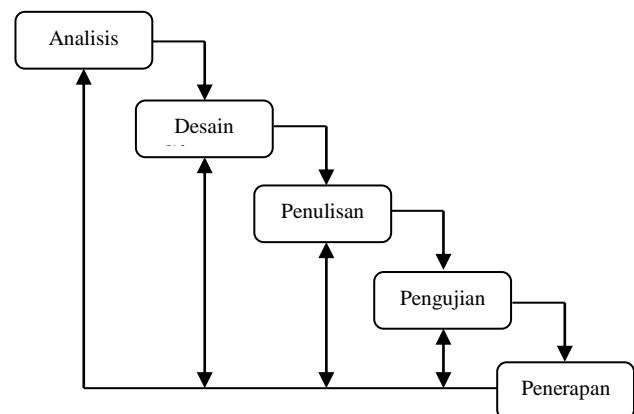
1. Bagaimana fungsi-fungsi diuji agar dapat dinyatakan valid?
2. Input seperti apa yang dapat menjadi bahan kasus uji yang baik?
3. Apakah sistem sensitif pada input-input tertentu?
4. Bagaimana sekumpulan data dapat diisolasi?
5. Berapa banyak rata-rata data dan jumlah data yang dapat ditangani sistem?

III. METODE PENELITIAN

3.1 Prosedur Penelitian

Metode Penelitian yang digunakan dalam penelitian ini adalah metode penelitian dan pengembangan. Metode perangkat lunak yang digunakan adalah metode *water fall*.

Ada lima tahapan dalam model *waterfall* seperti yang terlihat pada gambar 3.1.



Gambar 3.1 Model Waterfall.

3.2 Analisis Kebutuhan

Dalam tahap ini penulis akan menganalisa apa saja kebutuhan dari sistem, mulai dari kebutuhan fungsional sistem dan kebutuhan non fungsional dari sistem.

1. Kebutuhan fungsional (*Functional Requirement*)

Kebutuhan fungsional adalah kebutuhan yang memiliki keterkaitan langsung dengan website kebutuhan fungsional dari website ini meliputi:

- Sistem dijalankan pada *application* server
- Kebutuhan admin
 - ✓ Admin membutuhkan *username* dan *password*
 - ✓ Admin dapat menambahkan konfigurasi koneksi, untuk menambahkan konfigurasi koneksi membutuhkan nama koneksi, ip server, *port* server, *user* server dan *password* server.
 - ✓ Admin dapat mengelola menu database
 - ✓ Admin dapat mengelola menu aplikasi

2. Kebutuhan Tambahan (*Nonfunctional Requirement*)

Kebutuhan non-fungsional adalah kebutuhan yang tidak secara langsung terkait dengan fitur tertentu didalam website.

- Kebutuhan perangkat keras
Perangkat keras yang digunakan dalam membangun website ini adalah dengan menggunakan 1 buah server yang berfungsi sebagai webserver. Server tersebut berupa Virtual Machine (VM).
- Kebutuhan perangkat lunak
Perangkat lunak yang digunakan adalah perangkat yang mendukung penelitian ini meliputi:
 - a. Sistem operasi yang digunakan adalah Server Ubuntu 14.04.3 LTS untuk *webserver*.
 - b. Sistem operasi yang digunakan pada komputer *user* adalah window 10
 - c. *Webserver* yang digunakan adalah *Apache2*
 - d. *Database* yang digunakan adalah *MySQL*
 - e. *Owncloud* merupakan perangkat lunak untuk berbagi file berbasis internet yang bisa di akses dari mana saja.
 - f. *Ssh-key* berfungsi sebagai sebagai autologin server
 - g. *rsync* berfungsi sebagai untuk transfer dan sinkronisasi file
 - h. *crontab* sebagai penjadwalan *cronjob* yang berjalan pada waktu-waktu yang di tentukan.

3.3 Desain

Pada tahap ini akan dilakukan perancangan sistem pemantauan proses backup dengan mengalokasikan kebutuhan sistem baik perangkat keras dan lunak. Sistem perancangan meliputi perancangan sistem pemantauan keseluruhan dan perancangan antar muka.

3.4 Penulisan kode program

Pembuatan program harus sesuai dengan perancangan dan desain yang telah dibuat sebelumnya. Penulis menerapkan desain antar muka dan database kedalam bahasa pemrograman yang dipakai adalah menggunakan bahasa PHP dengan framework CodeIgniter 3 dan basis data MySQL.

3.5 Pengujian program

Pada tahap uji dilakukan dengan metode waterfall dengan menggunakan teknik pengujian blackbox terhadap semua fungsi dari aplikasi. Pengujian aplikasi atau perangkat lunak dengan teknik pengujian backbox berfokus pada persyaratan fungsional perangkat lunak. Karena itu uji coba blackbox memungkinkan pengembangan software untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program.

3.6 Penerapan Program

Pada tahapan ini sistem akan dipasang dan digunakan secara nyata. Perubahan bisa saja terjadi pada tahap ini karena adanya perubahan yang muncul dan tidak terdeteksi pada saat pengujian atau perangkat lunak.

IV. HASIL DAN PEMBAHASAN

4.1 Analisa Kebutuhan

1. Identifikasi Masalah

Permasalahan yang terjadi pada *backup* data yang ada di UPT.TIK diantaranya :

- Data *backup* yang didapat masih mengandalkan sistem *backup* yang dilakukan secara manual
- Data *backup* semalam ini belum bisa divisualisasikan secara baik.
- Pengecekan hasil *backup* Data masih dilakukan secara manual
- Tidak ada sistem notifikasi keberhasilan *backup* data

2. Analisa Sistem

- *Backup* data yang dilakukan dengan cara manual beresiko terhadap ke-*valid*-an data yang masih rendah, disamping itu juga efisiensi waktu dan tenaga juga tidak maksimal.
- Tidak ada sistem *notifikasi* keberhasilan *backup* data secara otomatis.

3. Kebutuhan Data

Data yang dibutuhkan dalam *desain* sistem *pemantauan backup* data diantaranya : nama sever, ip server, port server, user server, password server, besar file backup, status file backup.

4. Kebutuhan fungsional

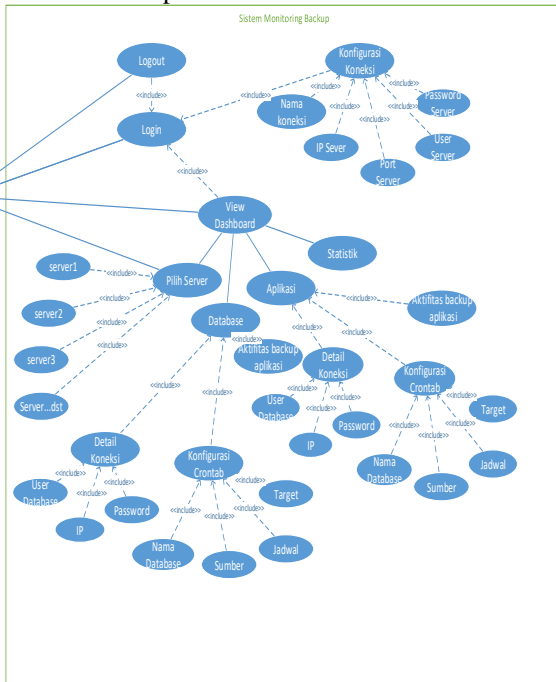
- Proses *Login* untuk admin
- Pengelolaan konfigurasi koneksi oleh admin meliputi: *input* nama server, *input*

- IP server, input port server, input user server, input password server.
- Pengelolaan menu *dashboard* oleh admin meliputi: menu statistic, menu database, menu aplikasi.
- Pengelolaan buat koneksi database dimenu database oleh admin meliputi: input User database, input password database, input IP.
- Pengelolaan buat konfigurasi crontab dimenu database oleh admin meliputi: input Sumber Server, input target server, input jawal backup.
- Pengelolaan buat koneksi database dimenu aplikasi oleh admin meliputi: input User database, input password database, input IP.
- Pengelolaan buat konfigurasi crontab dimenu aplikasi oleh admin meliputi: input Sumber Server, input target server, input jadwal backup.

4.2 Desain Sistem Pemantauan backup

1. Usecase Diagram

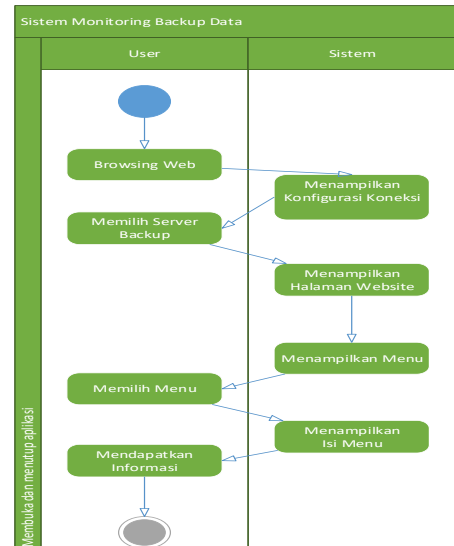
Usecase diagram merupakan fungsional yang di sediakan sistem sebagai unit-unit yang bertukar pesan antara unit dan aktor.



Gambar 4.1 Model use case diagram

2. Activity Diagram

Activity diagram adalah memodelkan alur kerja (workflow) sebuah proses bisnis dan urutan aktivitas dalam suatu proses.



Gambar 4.2 Activity diagram membuka dan menutup aplikasi

3. Desain sistem notifikasi email backup

Terdapat dua kondisi backup data:

1. Backup berhasil

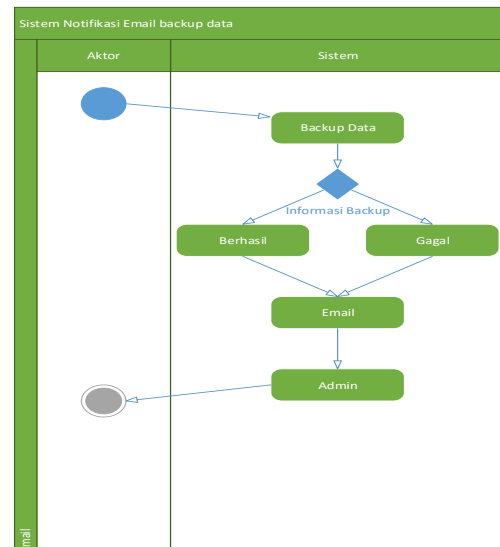
Backup berhasil artinya jika file dari server sumber dengan file yang dikirim keserver backup sama.

2. Backup gagal

Backup gagal artinya jika ukuran file dari sumber tidak sama dengan file yang telah dikirimkan ke server backup. Crontab gagal register atau tidak running di sistem linux.

Dari kedua kondisi tersebut sistem notifikasi email tetap mengirimkan email kepada admin backup.

4. Activity Diagram notifikasi email



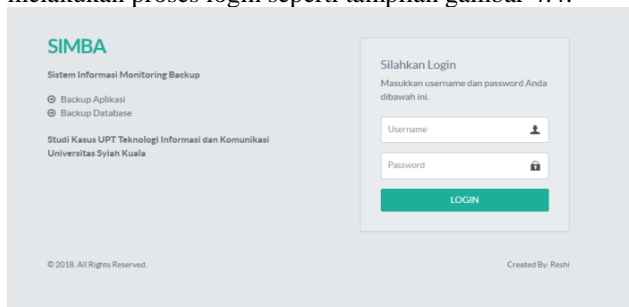
Gambar 4.3 Activity diagram notifikasi email backup

Dari *activity diagram* terdapat 5 tahap proses *notifikasi* email yaitu backup data, status backup berhasil, status backup gagal, sistem *notifikasi* email dan email admin.

Sistem *pemantauan backup* melakukan pengecekan hasil *backup* setiap satu menit. Pengecekan yang dilakukan sistem *pemantauan* yang pertama mengecek hasil backup dan ukuran file jika sama, berarti status backup berhasil dan terhubung ke *notifikasi* email. Jika ukuran file hasil backup tidak sama berarti gagal backup. Sistem akan memastikan proses kegagalan dengan cara cek jadwal *crontab*, cek *crontab* berhasil running dan teregister dan terakhir cek log yang error. Setelah proses ini di jalankan oleh sistem *pemantauan* langsung terhubung ke *notifikasi* email. *Notifikasi* email menginfokan status hasil backup tersebut kepada admin *backup*.

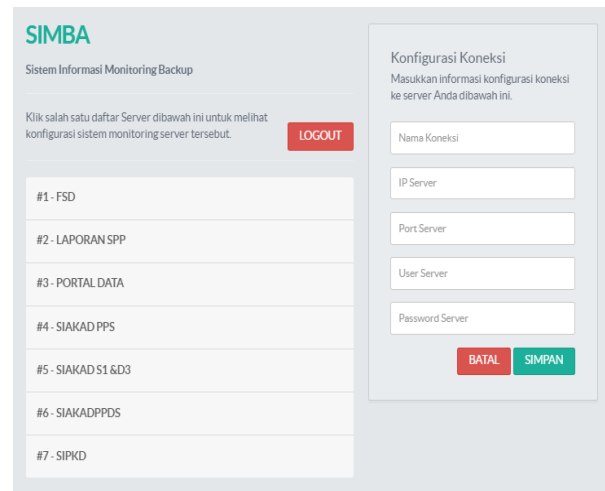
4.3 Implementasi Sistem Pemantauan

Sistem *pemantauan* merupakan aplikasi berbasis web sehingga untuk menjalankannya dibutuhkan *web browser*. Sistem ini terdiri dari 2 jenis proses *pemantauan backup* yaitu *backup* database dan *backup* aplikasi. Untuk akses ke sistem ini hanya satu jenis level akses (admin). Langkah awal untuk akses ke sistem ini admin harus melakukan proses login seperti tampilan gambar 4.4.



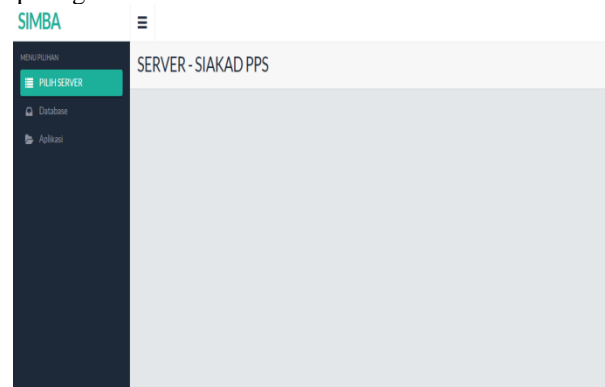
Gambar 4.4 Login sistem *pemantauan backup*

Setelah proses login berhasil maka akan tampil halaman konfigurasi koneksi seperti gambar 4.5. Konfigurasi koneksi merupakan fitur untuk menghubungkan server aplikasi dan database yang akan dipemantauan. Fitur ini disajikan *form input* data koneksi terdiri dari nama koneksi, ip server, port server, user server dan password server.



Gambar 4.5 Konfigurasi koneksi

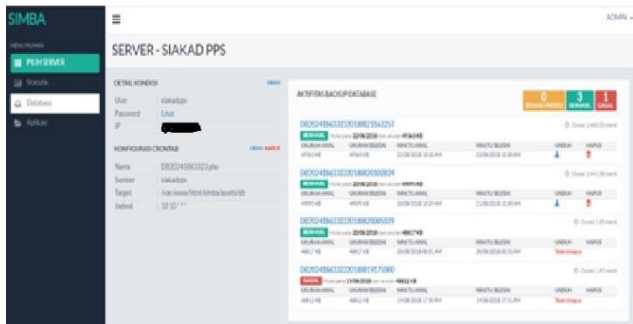
Langkah selanjutnya untuk melakukan *pemantauan*, admin harus memilih koneksi server untuk dipemantauan yang tersedia di fitur konfigurasi koneksi. Selanjutnya akan tampil halaman *dashboard* sistem *pemantauan* yang terdiri dari dua menu di samping kiri yaitu menu database dan aplikasi. Tampilan *dashboard* tersebut seperti gambar 4.6.



Gambar 4.6 Tampilan halaman *dashboard* sistem pemantauan

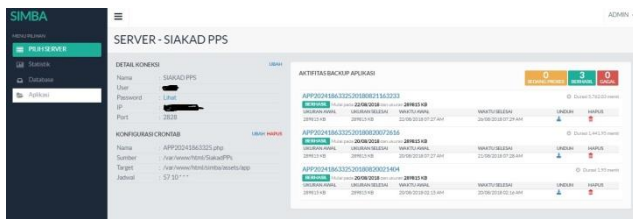
Untuk *mepemantauan* backup database maka admin memilih menu database dan akan tampil halaman *pemantauan* database. Halaman tersebut dapat dilihat di gambar 4.6. Informasi yang disajikan pada halaman *pemantauan* database terdiri dari detail koneksi, konfigurasi *crontab* dan sajian informasi.

1. Detail koneksi
Menyajikan informasi tentang user, password dan ip server
2. Konfigurasi *crontab*
Untuk menentukan nama, sumber, target dan jadwal backup
3. Sajian informasi
Fitur utama dalam proses pemantauan backup yang terdiri dari aktifitas backup database dan aplikasi.



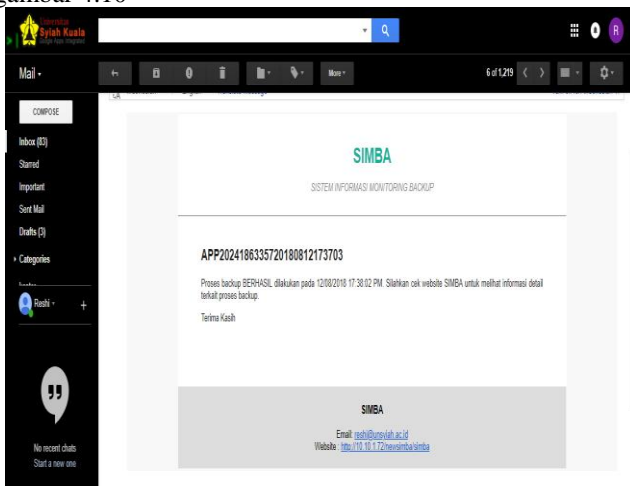
Gambar 4.7 Tampilan halaman pemantauan backup database

Jika admin ingin memantauan backup aplikasi maka memilih menu aplikasi. Tampilan pemantauan backup aplikasi sama dengan pemantauan backup database. Tampilan tersebut dapat dilihat pada gambar 4.9.



Gambar 4.8 Tampilan halaman pemantauan backup aplikasi

Setiap proses backup yang berhasil dilakukan sistem pemantauan akan mengirimkan informasi status backup melalui email ke admin backup seperti tampilan gambar 4.10



Gambar 4.9 Tampilan notifikasi email status sistem pemantauan

4.4 Uji Sistem

Uji sistem pemantauan dilakukan untuk memastikan sistem tersebut berjalan sesuai kebutuhan. Uji sistem ini dilakukan dengan melakukan proses backup ke salah satu server database dan aplikasi, studi kasus yang diuji yaitu server siacad PPs . Dari proses backup tersebut dilakukan pemantauan melalui sistem pemantauan

yang sudah dikembangkan. Proses uji sistem dengan cara menggunakan metode *black box testing*.

1. Pengujian *Black box testing* Pada menu Login

Berikut ini adalah table pengujian *black box* berdasarkan sistem pemantauan backup data untuk menu fungsi Login, yaitu sebagai berikut:

No	Skenario Pengujian	Hasil Yang Diharapkan	Kesimpulan
1	Mengosongkan <i>User name</i> dan <i>Password</i> Lalu mengklik tombol <i>Login</i>	Sistem akan Menolak akses login dan akan menampilkan pesan	Valid
	<p>Test case :</p>	<p>Hasil Pengujian:</p>	
2	Menghasilkan <i>Usemame</i> dan <i>Password</i> dengan benar	Sistem menerima tampilan login dan menghasilkan tampilan	Valid
	<p>Test case :</p>	<p>Hasil pengujian :</p>	

Dari hasil uji beberapa server yang terhubung dengan sistem pemantauan backup di peroleh hasil sebagai berikut:

- Setiap server yang telah konek dengan sistem informasi backup dapat melakukan proses backup sesuai dengan jadwal yang disetting di sistem pemantauan backup.
- Ukuran file yang dibackup sama dengan file hasil backup dari rata –rata server yang konek dengan sistem pemantauan backup.
- Dari semua server yang konek kesistem pemantauan backup jarang terdapat kegagalan backup.

4.5 Penerapan sistem pemantauan

Sistem pemantauan backup data akan dipasang dan digunakan secara nyata di UPT TIK Unsyiah. Perubahan bisa saja terjadi pada tahap ini karena adanya perubahan yang muncul dan tidak terdeteksi pada saat pengujian atau perangkat lunak. Sedangkan pemeliharaan sistem pemantauan backup dirancang dengan dilakukannya pemeriksaan periodik terhadap data pada aplikasi.

V. KESIMPULAN

Berdasarkan hasil penelitian dapat disusun kesimpulan sebagai berikut:

1. Adanya topologi sistem *pemantauan backup* data terdistribusi
2. Sistem *pemantauan backup* data dapat menyajikan informasi hasil dan besar file *backup*
3. Telah berhasil dibuat notifikasi berbasis email.
4. Setelah dilakukan pengujian diketahui fungsionalitas sistem pemantauan bisa berjalan dengan baik.

VI. DAFTAR PUSTAKA

- [1] UPT. TIK, "Dokumen Blueprint ICT Universitas Syiah Kuala (UNSYIAH)," *Banda Aceh.UPT TIK desember 2014*, 2014.
- [2] Aws.amazon.com, "cloud computing," <https://aws.amazon.com/what-is-cloud-computing/>, 2018.
- [3] J. Zhang and H. Li, "Research and implementation of a data backup and recovery system for important business areas," *Proc. - 9th Int. Conf. Intell. Human-Machine Syst. Cybern. IHMSC 2017*, vol. 2, pp. 432–437, 2017.
- [4] V. Javaraiah, "Backup for cloud and disaster recovery for consumers and SMBs," *Int. Symp. Adv. Networks Telecommun. Syst. ANTS*, 2011.
- [5] B. Martini and K. K. R. Choo, "Cloud storage forensics: OwnCloud as a case study," *Digit. Investig.*, vol. 10, no. 4, pp. 287–299, 2013.
- [6] M. Gregus and V. Karovic, "Practical Implementation of Private Cloud Based on Open Source ownCloud for Small Teams - Case Study," *2015 10th Int. Conf. P2P, Parallel, Grid, Cloud Internet Comput.*, pp. 183–187, 2015.
- [7] McLeod. Raymond. 1998. *Managemen Information Sitystems*. 7th Edition, New Jersey:Prentice Hall, Inc.
- [8] Mercy. 2005. *Design, Monitoring and Evaluation Guidebook*. Portland, USA
- [9] R.Wrihatnolo, *Monitoring, evaluasi, dan pengendalian : Konsep dan pembahasan*, 2008.
- [10] Chang, C. H., Hsu, C. N., dan Lui, S. C., 2003, Automatic Information Extraction from Semi Structured Web Pages by Pattern Discovery, *Journal Decision Support Systems – Web Retrieval and Mining*, 35, 1, 129 – 147.
- [11] Amsler, G. M., Findley, H. M., & Ingram, E., 2009, Performance monitoring: guidance for the modern workplace. *Supervision*, 70, 12-19
- [12] W.Stallings and L.Brown, "Operating System Security", In *Computer Security Principle and Practice*, New Jersey, Pearson Education, Inc., 2014.
- [13] O. W. Purbo, "Membuat Sendiri Cloud Computing Server Menggunakan Open Source.," *yogyakarta.andi*, 2012.
- [14] F. Doelitzscher, A. Sulistio, C. Reich, H. Kuijs, and D. Wolf, "Private cloud for collaboration and e-Learning services: From IaaS to SaaS," *Comput. (Vienna/New York)*, vol. 91, no. 1, pp. 23–42, 2011. McLeod, 1998
- [15] owncloud inc Community, "owncloud," <https://id.wikipedia.org/wiki/OwnCloud>, 2012. [Online]. Available: <https://id.wikipedia.org/wiki/OwnCloud>.
- [16] Cloudindonesia.com, "cloud storage," <http://cloudindonesia.com/apa-yang-di-maksud-cloud-storage/>, 2018.
- [17] Aunur R. Mulayanto, *Rekayasa Perangkat Lunak JILID 1*, Direktorat Pembinaan Sekolah Menengah Kejuruan, 2008
- [18] Khan, Mohd Ehmer, 2011, Different Approach to Blackbox Testing Technique for Finding Error, *International Journal of Software Engineering & Applications (IJSEA)*, Vol.2, No.4, October 2011
- [19] Nidhra, Srinivas, and Dondeti, Jagruthi, 2012, Blackbox and Whitebox Testing Techniques - A Literature Review, *International Journal of Embedded Systems and Applications (IJESA)* Vol.2, No.2, June 2012
- [20] Sugiyono, 2009, *Metode Penelitian Kuantitatif, Kualitatif dan R&D*, Bandung : Alfabeta.
- [21] Pressman, Roger S. (2002). *Rekayasa Perangkat Lunak Pendekatan Praktisi* (Buku. Satu). Yogyakarta: ANDI.