

Tujuan dari *performance testing* untuk memvalidasi kecepatan sebuah perangkat lunak terhadap kebutuhan sistem yang cepat. Secara umum harus mendefinisikan kombinasi waktu *respons* dan beban kerja

## **7. Metode Penelitian**

Bagian ini akan menjelaskan mengenai langkah – langkah yang akan dilakukan dalam merancang sistem manajemen layanan *web* berbasis *Platform as a Service (PaaS)* dengan *API openstack*.

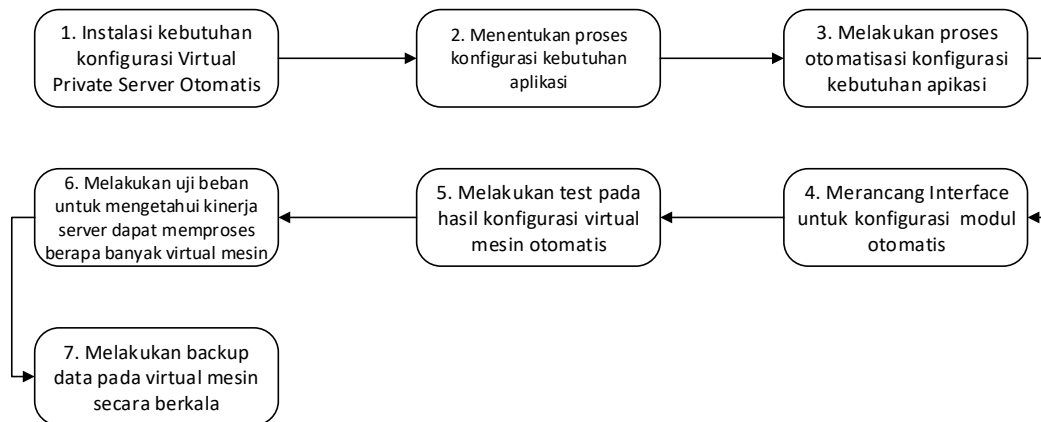
### **7.1. Analisis Kebutuhan**

Pada tahap ini dilakukan analisis kebutuhan sistem, analisis kebutuhan sistem meliputi data yang digunakan, pembelajaran dari referensi yang sudah ada dan perangkat yang digunakan baik perangkat lunak maupun perangkat keras:

1. Tahapan pertama yang dilakukan dalam penelitian ini adalah mengidentifikasi permasalahan. Tahap ini merupakan tahap yang paling penting dalam penelitian karena jalannya penelitian didasarkan atas permasalahan yang terjadi. Setelah menentukan masalah yang terjadi, tahapan yang diperlukan selanjutnya adalah menentukan rumusan masalah dan tujuan yang ingin dicapai dalam penelitian. Pada penelitian ini identifikasi permasalahan dilakukan dengan menggunakan teknik observasi, dari teknik ini maka akan dapat diketahui mengenai keluhan – keluhan yang ada di lapangan.
2. Tahap kedua yang dilakukan dalam metodologi penelitian ini adalah studi literatur. Studi literatur dilakukan dengan mengambil literatur – literatur pendukung dari jurnal – jurnal ilmiah, baik jurnal dalam negeri ataupun jurnal luar negeri dan dari beberapa buku. Dalam studi literatur ini, penulis mencari sumber terkait permasalahan – permasalahan yang perlu menjadi perbaikan dalam penelitian selanjutnya.

### **7.2. Kerangka Kerja Penelitian**

Bagian ini menjelaskan tentang bagaimana penelitian ini dilakukan. Berikut adalah beberapa proses penting yang dilakukan :



Gambar 6.10.2.1 Kerangka Kerja Sistem

#### 1. Instalasi kebutuhan konfigurasi VPS (*Virtual Private Server*)

Tahap pertama dalam penelitian ini dengan mempersiapkan sistem dan *software* yang dibutuhkan dalam mengkonfigurasi kebutuhan *web server* dalam menginstalasi modul-modul kebutuhan sistem seperti sistem operasi, *database*, *framework*, *web server*.

#### 2. Menentukan proses konfigurasi kebutuhan aplikasi

Pada tahap ini mempersiapkan kebutuhan konfigurasi untuk membangun sebuah *web server*, dari *web service* sampai *database* yang diperlukan untuk menampung setiap user.

#### 3. Melakukan proses otomatisasi konfigurasi modul aplikasi

Pada tahap ini melakukan otomatisasi dalam mengkonfigurasi sebuah *web server*. Dimana proses otomatisasi menggunakan *API* dari *Ansible*.

#### 4. Merancang *web front-end* untuk user

Pada tahap ini bila semua konfigurasi sudah berjalan dengan baik, dibuat *web* yang dapat digunakan oleh pengguna mengatur kebutuhan web yang dibuat serta dapat melihat informasi dari setiap *VPS* yang dimiliki.

#### 5. Melakukan *test* pada hasil konfigurasi

Pada tahap ini menguji hasil konfigurasi dari *web interface*. Untuk menguji konfigurasi berjalan dengan baik atau tidak

#### 6. Melakukan uji beban *server*

beban ini dilakukan untuk menguji tingkat kemampuan *server* dalam memberikan pelayanan pada pengembangan aplikasi. Bila *server* tidak sanggup untuk melayani pengembang baru akan menolak menyediakan layanan.

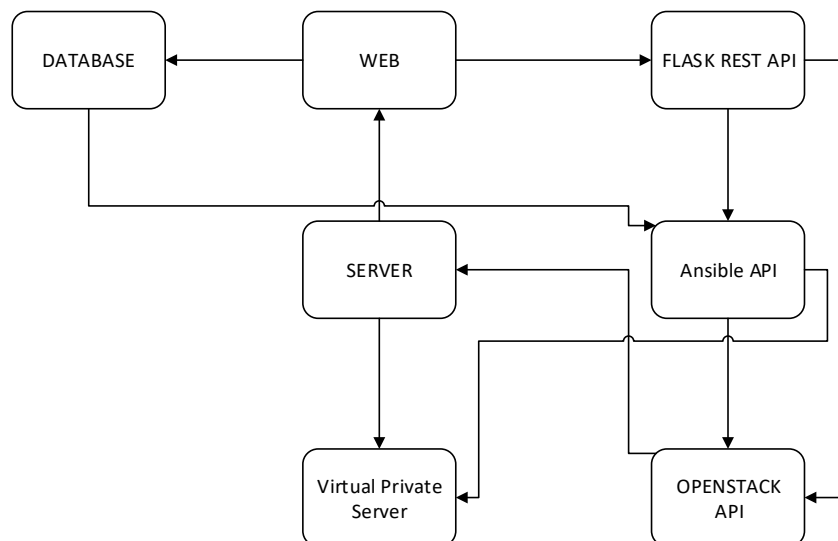
#### 7. Melakukan *Backup Server*

Ketika *server user* sudah berjalan, sistem akan melakukan *backup* data pengguna untuk menjaga ketersediaan *data* pengguna bila terjadi sesuatu yang dapat merugikan pengguna, seperti *file* data hilang atau rusak, dapat dikembalikan dengan *restore* pada sistem dengan menggunakan waktu *check point*.

### 7.3. Perancangan Sistem

Pada tahap Perancangan sistem, dilakukan berdasarkan hasil analisa kebutuhan sistem yang sudah dilakukan sebelumnya. Hal ini dilakukan agar, perancangan tidak keluar dari tujuan sistem yang dikembangkan.

#### 7.3.1. Desain Arsitektur Kerja Sistem



Gambar 7.3.1.1 Desain Kerja Sistem

Pada gambar 7.3.1.1 merupakan hubungan antara setiap perangkat lunak yang ada pada server utama. Berikut ini merupakan penjelasannya :

1. *Web dan database*

*Database* digunakan untuk menyimpan informasi pengguna serta informasi pada *VPS* yang dimiliki oleh setiap pengguna.

2. *Web dan framework flask REST API*

Flask REST API digunakan untuk menghubungkan web *front-end* dengan *back-end*. Dimana *back-end* dibuat dengan menggunakan *framework flask* dari *python*, selanjutnya dibuat *API* tersendiri agar web *front-end* dapat mengirim dan menerima informasi serta konfigurasi yang dilakukan yang nantinya akan diproses oleh *back-end server*.

3. *Database dan ansible API*

Hubungan *database* dengan *ansible API* akan mengambil informasi pengguna berupa *username*, *email* dan *password* yang nantinya akan dimasukkan pada konfigurasi dalam *virtual machine* yang selanjutnya sebagai *super admin* pada *VPS* yang dibuat.

4. *Framework flask REST API dan asible API*

Hubungan *framework flask REST API* dengan *asible API*. Ketika *user* melakukan konfigurasi pada *web* utama akan dikirim melalui *REST API flask* selanjutnya konfigurasi tersebut akan digunakan oleh *API ansible* untuk mengkonfigurasi *virtual machine* yang dibuat.

5. *Framework flask dan openstack API*

Hubungan *framework flask* dan *openstack API* adalah pada *flask* dikonfigurasi untuk dapat terhubung dengan *server devstack*, dimana *server openstack* yang mengelola dalam membuat sebuah *virtual machine*.

6. *Openstack API dan server*

*API openstack* digunakan untuk terhubung dengan server *back-end* yang dibuat dengan *framework flask*. Untuk dapat mengembangkan aplikasi *openstack* dari bahasa pemrograman yang berbeda.

#### 7. *Ansible API* dan *virtual server*

Hubungan *ansible API* dengan *virtual* mesin adalah *ansible* akan mengirimkan konfigurasi pada *VPS* melalui *SSH* yang yang dibuat. Selanjutnya pada *virtual* mesin akan melakukan konfigurasi yang diperlukan untuk membangun sebuah *web* aplikasi secara otomatis. Pengguna hanya perlu memilih keperluan yang ada *menu web front-end*.

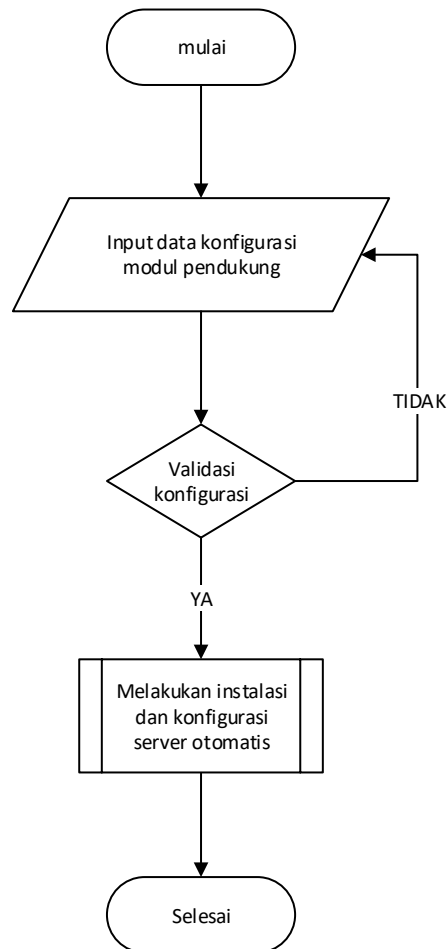
#### 8. *Server* dan *Virtual* Mesin

Hubungan antara *server* dan *virtual* mesin adalah *server* mengelola dan memberikan *resource* pada *virtual* mesin serta mengatur konektivitas setiap *virtual* mesin yang dimiliki.

### 7.4. **Flowchart Sistem**

Pada bagian ini akan menjelaskan proses yang dilakukan oleh perangkat lunak, bagaimana proses tersebut berjalan yang akan dijelaskan pada setiap *flowchart* berikut ini:

#### 7.4.1. Flowchart Konfigurasi Otomatis

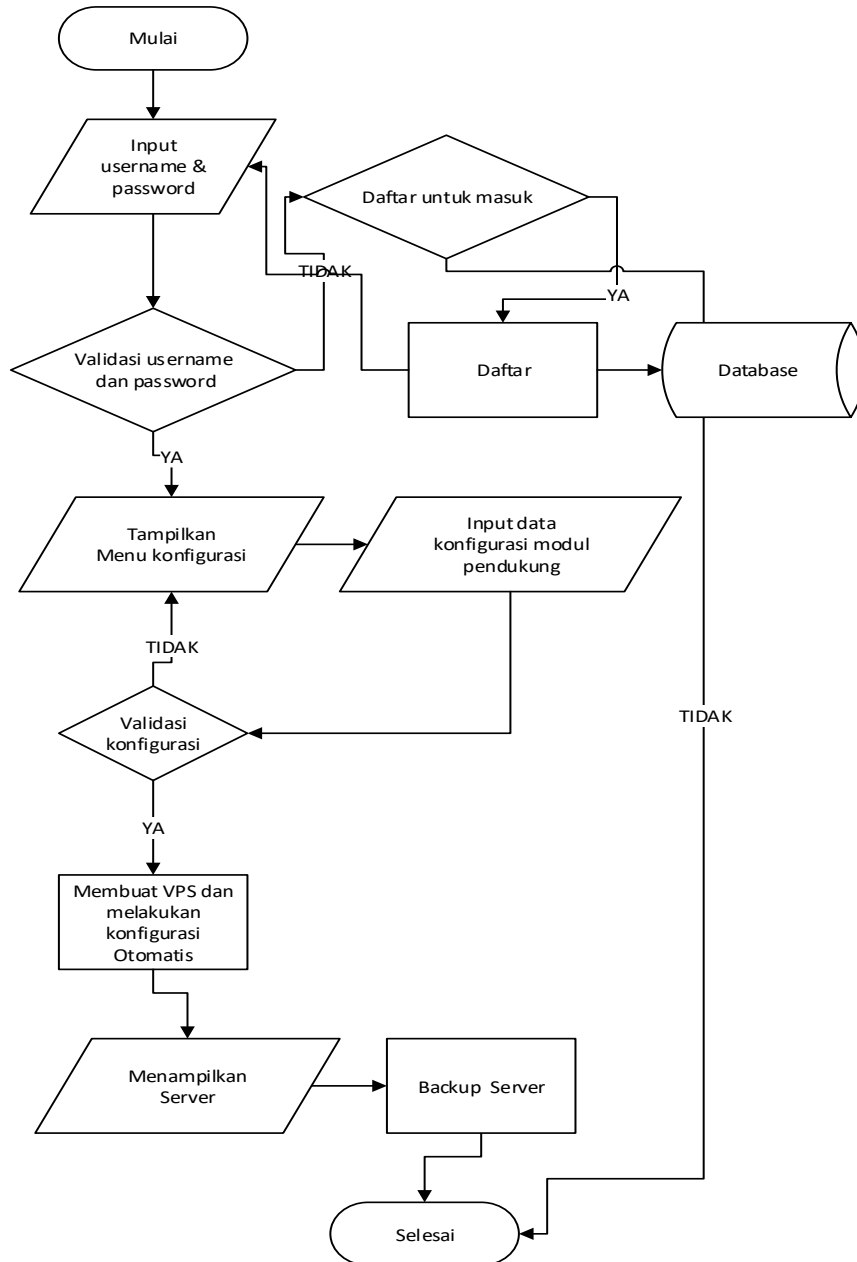


Gambar 7.4.1.1 Flowchart Konfigurasi Otomatis

Pada desain *flowchart* 7.4.1.1 merupakan garis besar gambaran *system* yang akan dikerjakan, bagaimana alur *platform* yang dibuat dapat mengotomatitasi dalam instalasi serta konfigurasi sebuah layanan pada *virtual machine*. Mulai dari memasukkan aplikasi pendukung yang ingin digunakan seperti *database*, *web service*, *username* dan *e-mail administrator* layanan aplikasi. Kemudian dari hasil *input* tersebut diolah oleh *flask API* yang dibuat untuk dimasukkan ke dalam konfigurasi yang terdapat ada *Ansible API*, selanjutnya dari *ansible API* dengan menggunakan *SSH* akan melakukan instalasi konfigurasi pada *virtual server* yang didapat setiap *user*. Bila

konfigurasi telah selesai *user* akan menerima sebuah *ip public* untuk dapat mengakses layanan aplikasi.

#### 7.4.2. Flowchart Front-end Web

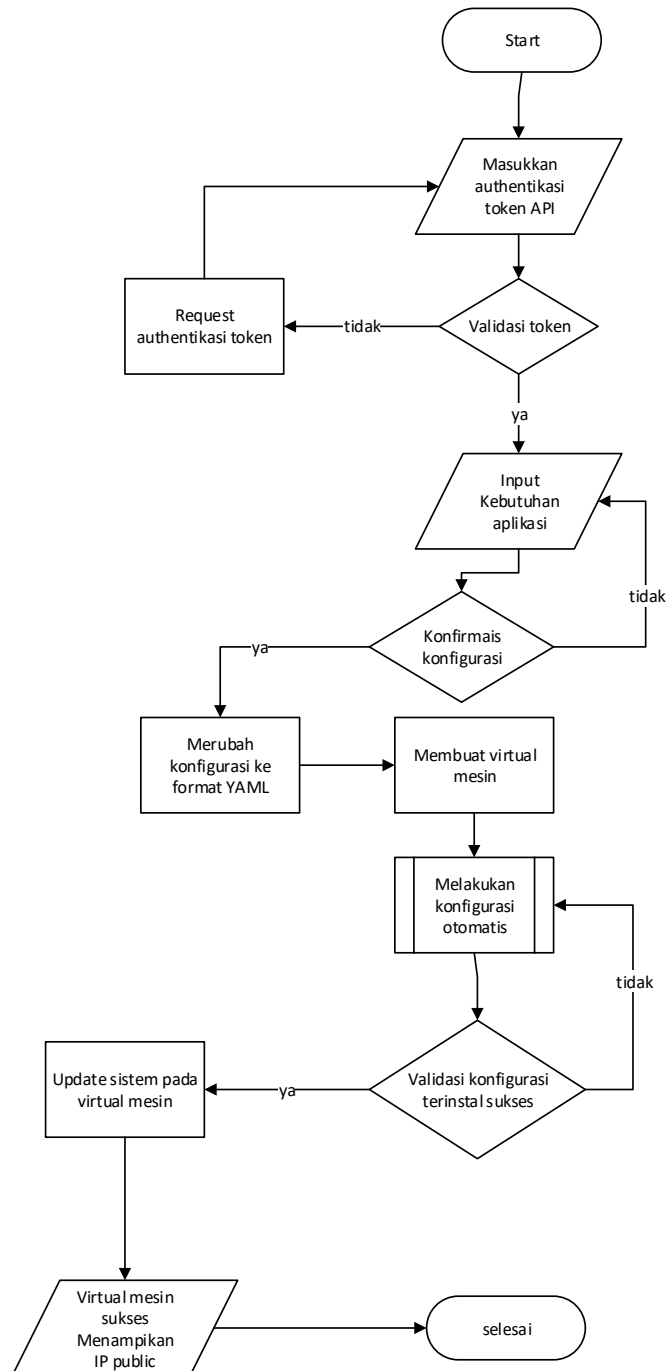


Gambar 7.4.2.1 Flowchart Front-end Web

Pada desain *flowchart* diatas merupakan alur kerja *web* dari sisi *front-side* atau pada sisi *user*. Setelah *user* melakukan registrasi pada *website*, selanjutnya *user* akan diarahkan ke *menu* instalasi dan konfigurasi dalam membangun layanan *virtual* mesin. Menu akan dibuat secara *user friendly* sehingga *user* hanya perlu memilih *menu* konfigurasi *system*. Ketika *user* sudah selesai memilih menu konfigurasi, *system* akan memvalidasi informasi yang dimasukkan oleh *user*, bila sudah benar *system* akan mengirimkan hasil instalasi dan konfigurasi dalam sebuah *data* berbentuk *JSON* ke dalam *system* yang nantinya akan dikirim ke *virtual sever* untuk melakukan konfigurasi dan instalasi modul kebutuhan aplikasi selanjutnya sistem akan melakukan *backup* berkala untuk menjaga keamanan data pengguna bila terjadi satu kesalahan dapat dilakukan *restore data*.



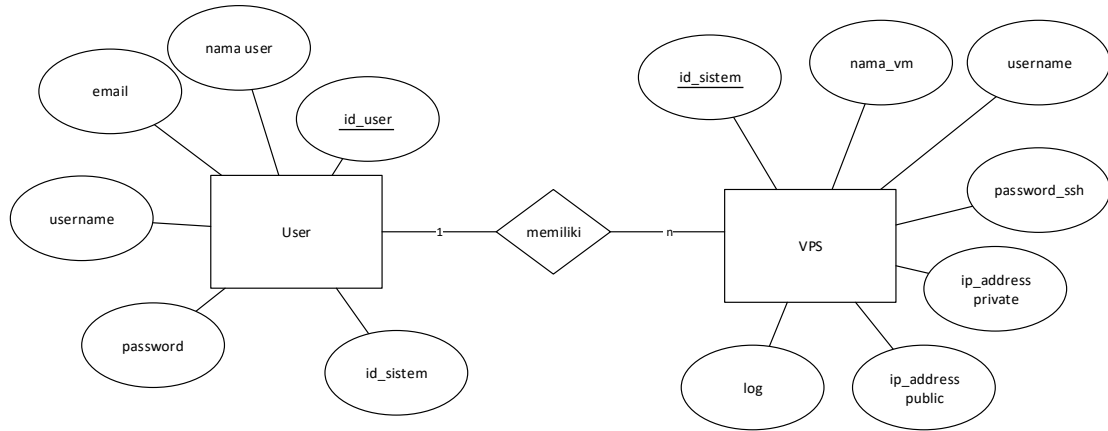
### 7.4.3. Flowchart Back-end



Gambar 7.4.3.1 Flowchart Back-End

Pada gambar flowchart 7.4..3.1 menjelaskan alur kerja sistem *back-end* sistem. Tahap pertama yang harus dilakukan adalah melakukan komunikasi dengan *openstack* agar dapat menggunakan *API* dari *openstack*. Melakukan komunikasi dengan melakukan *request token autentikasi*, hal ini dimaksudkan agar setiap komunikasi yang dibuat dapat melakukan *update data* pada *API* dan yang memiliki *token* dari *autentikasi* saja yang dapat berkomunikasi. Selanjutnya setelah melakukan *request*, sistem tetap akan melakukan validasi *token id* dari *openstack*, karena setiap melakukan *request token* memiliki batas waktu yang ditentukan. Setelah bisa melakukan komunikasi dengan *openstack* selanjutnya menerima masukan konfigurasi dari *front-end* dan sudah divalidasi oleh administrator, selanjutnya merubah setiap masukan konfigurasi dalam ke *YAML*, *YAML* merupakan format untuk memberi perintah pada *ansible* agar dapat melakukan instalasi maupun konfigurasi secara otomatis. Setelah *file* siap, sistem akan membuat *virtual* mesin dan mengirimkan konfigurasi tersebut ke *virtual* mesin dan langsung melakukan konfigurasi secara otomatis. Selanjutnya sistem akan kembali melakukan mengecek apakah sudah semua *terinstall* dan *terkonfigurasi* dengan baik, sehingga bila ada gagal dalam dilakukan *instalasi* kembali. Setelah semua dicek dan sukses maka selanjutnya sistem akan menampilkan *virtual* mesin siap dan sebuah *ip public* untuk diakses oleh *developer* dan *ssh key* bila *developer* memiliki kebutuhan aplikasi khusus yang tidak dimiliki oleh sistem.

## 7.5. Rancangan ERD



Gambar 7.4.3.1 *Entity Relationship Diagram* Sistem

Pada desain *ERD* diatas menjelaskan hubungan user dengan *VPS* yang dimiliki user memiliki sebuah vps dengan satu *ip public* untuk dapat mengakses *server*. Serta pada *vps* akan menyimpan *log* keadaan *vps* mulai dari kinerja dan *resource* yang dimiliki.

## 7.6. Tampilan Sistem

Pada tahap ini merupakan gambaran *menu* dan *tool – tool* dari sistem yang didapat oleh *user* dan administrator.

### 7.6.1. Menu User

Pada bagian ini menjelaskan *menu – menu* yang di dapat oleh pengguna pada sistem *PaaS* yang dibangun.

The screenshot shows a web form titled "IBRESWCloud" with the heading "Sign Up". It contains four input fields: "Nama Lengkap :", "Email :", "Username :", and "Password :". Below the fields are two buttons: "Sign Up" and "Back To Login".

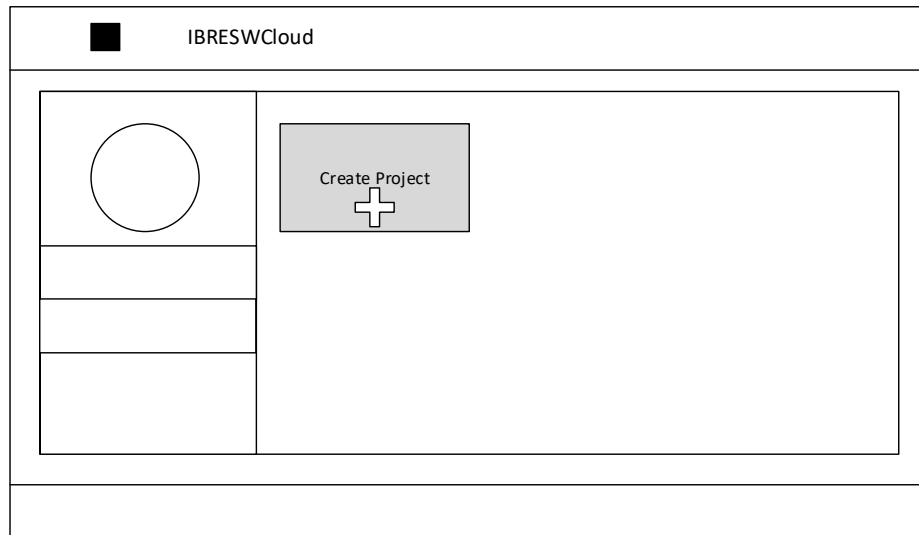
Gambar 7.6.1.1 Tampilan *form Sign up*

*Menu* pertama yaitu *form* untuk melakukan registrasi ke sistem, pada tahap ini pengguna melakukan registrasi dengan memasukkan *email*, *username*, dan *password*.

The screenshot shows a web form titled "IBRESWCloud" with the heading "Sign In". It contains two input fields: "Username :" and "Password :". Below the fields is a "Login" button. At the bottom, there are two links: "Register Now!" and "Forgot Password ?".

Gambar 7.6.1.2 Tampilan *form Sign in*

Bila pengguna sudah memiliki akun, selanjutnya akan melakukan *login* pada *menu sign in*.



Gambar 7.6.1.3 Tampilan *dashboard*

Pada gambar 7.6.1.3 menampilkan *dashboard user*, menu utama dari *dashboard* adalah membuat *project* baru, pada menu ini *user* dapat membuat serta melihat *project* yang sudah pernah dibuat oleh *user* tersebut.

Gambar 7.6.1.4 Tampilan *form Create Project*

Pada gambar 7.6.1.4 merupakan tampilan *form* untuk membuat *project* baru, setelah memilih *create project* selanjutnya akan tampil *form* seperti pada gambar

7.6.1.4. *User* selanjutnya *menginput* kebutuhan dari aplikasi *website* yang sedang dikerjakan, seperti sistem operasi, *database*, *web server* yang dibutuhkan dan lainnya.

The screenshot shows the IBRESWCloud web interface. At the top, there is a header bar with a black square icon and the text 'IBRESWCloud'. Below the header, the main content area is divided into two sections. On the left, there is a vertical sidebar containing a large circle icon at the top, followed by three empty rectangular boxes. On the right, there is a large rectangular area. Inside this area, there is a gray rectangular box labeled 'Project Name' and a button labeled 'Create Project' with a plus sign icon.

Gambar 7.6.1.5 Tampilan setelah membuat project

Pada gambar 7.6.1.5 menjelaskan ketika *user* telah berhasil membuat sebuah *project* akan tampil seperti gambar 7.6.1.5.

The screenshot shows the IBRESWCloud web interface. At the top, there is a header bar with a black square icon and the text 'IBRESWCloud'. Below the header, the main content area is divided into two sections. On the left, there is a vertical sidebar containing a large circle icon at the top, followed by three empty rectangular boxes. On the right, there is a large rectangular area. Inside this area, there is a gray rectangular box labeled 'Project Name' and a button labeled 'Detail' with a plus sign icon.

Gambar 7.6.1.6 Tampilan detail *project*

Pada gambar 7.6.1.6 menjelaskan ketika *user* memilih *project* yang telah dibuat, sistem akan menampilkan detail dari *project* tersebut, seperti status *virtual* mesin, *ip public* yang dapat di akses, *backup* terakhir dan lainnya.

The screenshot shows the IBRESWCloud web interface. At the top, there's a header with the logo and the text 'IBRESWCloud'. Below this, the main content area is titled 'BACKUP HISTORY'. To the right of the title is a search bar labeled 'Search :' with the text 'tgl' entered. Below the search bar is a table with the following columns: 'No', 'tanggal', 'Log', and 'Aksi'. The 'Log' column contains a modal dialog box titled 'Konfirmasi restore' with the text 'Pada tanggal : xx-xx-xxxx' and two buttons: 'Tidak' and 'Ya'. The 'Aksi' column contains two buttons: 'restore' and 'Delete'.

Gambar 7.6.1.7 Tampilan *Backup history*

Pada gambar 7.6.1.7 merupakan *form* yang menampilkan *backup history*. Sehingga bila terjadi kesalahan saat mengembangkan aplikasi, *user* dapat kembali *merestore* pada hasil *backup* sebelumnya.

## 7.6.2. Menu Sistem *administrator*

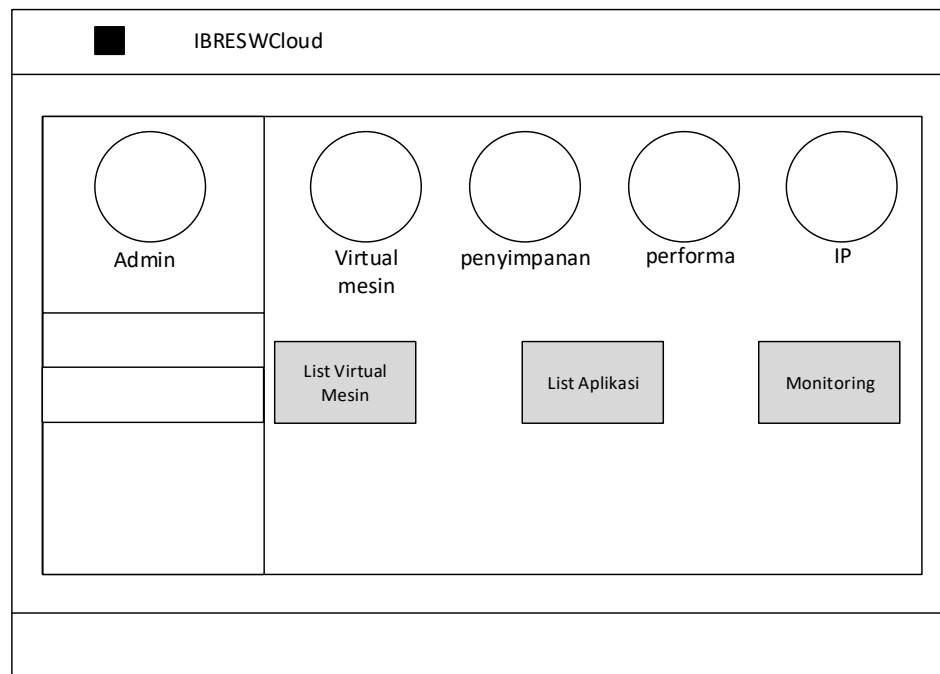
Pada tahap ini akan menjelaskan menu yang diperoleh *administrator* dalam sistem *PaaS*.



The image shows a login form titled "IBRESWCloud". Below the title is the text "Sign In". There are two input fields: "Username :" and "Password :". Below the password field is a "Login" button. At the bottom of the form, there are two links: "Register Now!" and "Forgot Password ?".

Gambar 7.6.2.1 Tampilan *form login administrator*

Untuk dapat masuk sistem administrator harus melakukan *login* pada *website*, *login administrator* sama seperti yang dilakukan oleh *user*, hanya saja hak akses yang dimiliki oleh *administrator* berbeda.

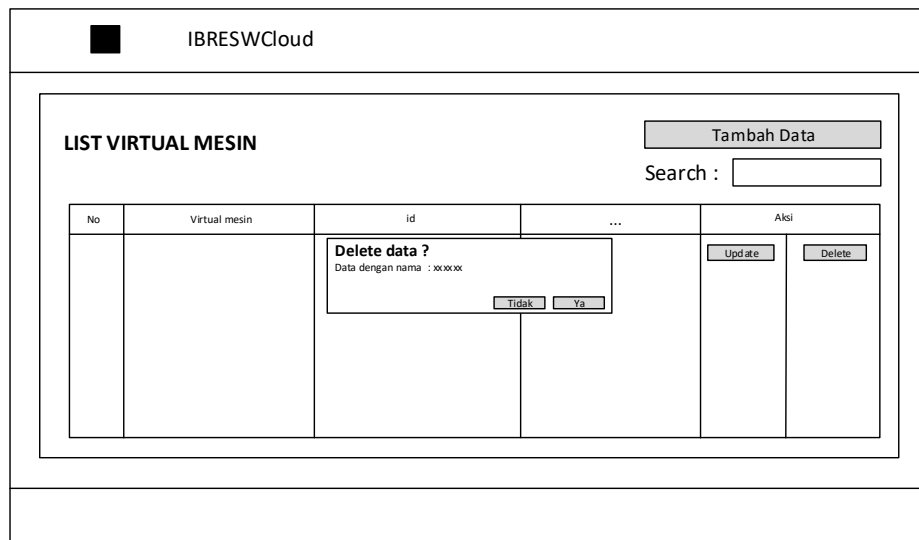


The image shows the administrator dashboard for "IBRESWCloud". The dashboard has a header with the logo and name "IBRESWCloud". The main content area is divided into two columns. The left column has a sidebar with a circular icon labeled "Admin" and three empty rectangular boxes below it. The right column has five circular icons labeled "Virtual mesin", "penyimpanan", "performa", and "IP". Below these icons are three rectangular buttons labeled "List Virtual Mesin", "List Aplikasi", and "Monitoring".

Gambar 7.6.2.2 Tampilan *dashboard administrator*




Setelah melakukan *login*, *administrator* akan dialihkan ke *dashboard* sistem seperti gambar 7.6.2.2. pada *dashboard* akan terlihat berapa *virtual* mesin yang ada, penyimpanan yang sudah digunakan, performa *CPU*, *IP public* yang sudah terpakai, serta *monitoring server*.



Gambar 7.6.2.3 Tampilan *list virtual* mesin *user*

Pada gambar 7.6.2.3. menampilkan *list virtual* mesin yang dibuat oleh *user*, pada bagian ini *administrator* dapat menghapus *virtual* mesin atau melakukan *update* bila *user* ingin mengganti sistem operasi.


IBRESWCloud

LIST APLIKASI

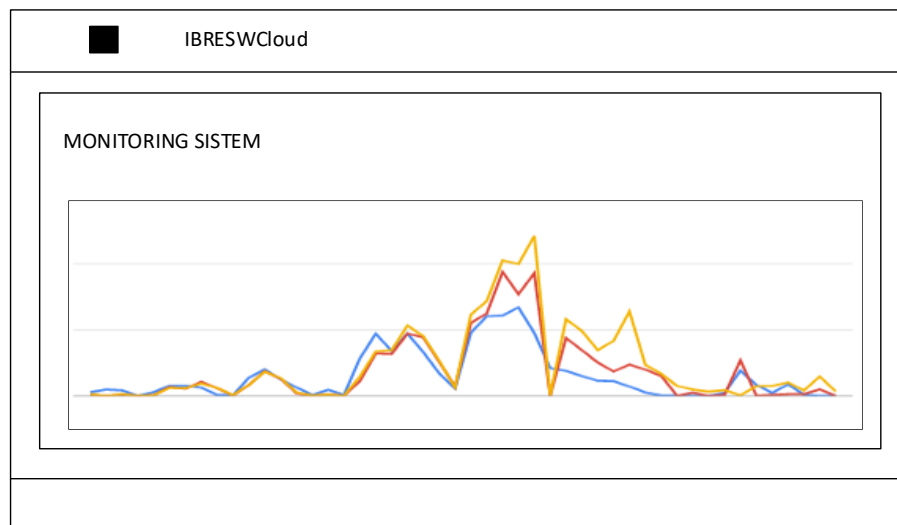
Tambah Data

Search :

| No | Virtual mesin | id   | ... | Aksi                                |
|----|---------------|--|-----|-------------------------------------|
|    |               | <div> Delete data ?<br/> Data dengan nama : xxxxxx<br/> <div>Tidak Ya</div> </div> |     | <div>Update</div> <div>Delete</div> |

Gambar 7.6.2.4 Tampilan *list* aplikasi tersedia

Pada gambar 7.6.2.4 merupakan *form* untuk menampilkan aplikasi yang dapat *diinstall* secara otomatis pada *virtual* mesin.



Gambar 7.6.2.5 Tampilan *menu monitoring*

Pada gambar 7.6.2.5 merupakan *menu* untuk melakukan *monitoring*. *Administrator* bertugas membuat *server* berjalan dengan baik serta menangani bila

terjadi sesuatu yang tidak diinginkan. Oleh karena itu penting dilakukan *monitoring* dari sisi *administrator*. Penerapan metode adaptif *threshold* untuk memberi ambang batas pada *load CPU* sehingga pada proses *monitoring* tersebut dapat mengetahui maksimal *CPU* bekerja pada *presentase* perhitungan yang dilakukan *threshold* adaptif.

### 7.7. Evaluasi Perancangan Sistem

Pada tahap ini dilakukan evaluasi terhadap desain perancangan sistem. Bila sistem telah sesuai dengan kebutuhan awal yang didefinisikan akan dilanjutkan ke tahap implementasi. Namun apabila desain sistem belum memenuhi kebutuhan awal yang didefinisikan, maka akan dilakukan perancangan ulang desain sistem.

## 8. Jadwal Pelaksanaan Penelitian

Jadwal pelaksanaan penelitian yang akan dilakukan dalam rangka menyelesaikan penelitian ini ditunjukkan pada Tabel 8.1.

Tabel 8.1 Rancangan Jadwal Pelaksanaan Penelitian

| No. | Kegiatan                     | Minggu ke- |   |   |   |   |   |   |   |   |    |    |    |
|-----|------------------------------|------------|---|---|---|---|---|---|---|---|----|----|----|
|     |                              | 1          | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 1   | Studi Literatur              |            |   |   |   |   |   |   |   |   |    |    |    |
| 2   | Pengumpulan Data             |            |   |   |   |   |   |   |   |   |    |    |    |
| 3   | Perancangan Sistem           |            |   |   |   |   |   |   |   |   |    |    |    |
| 4   | Pembuatan Sistem             |            |   |   |   |   |   |   |   |   |    |    |    |
| 5   | Pengujian Sistem             |            |   |   |   |   |   |   |   |   |    |    |    |
| 6   | Penulisan Laporan Penelitian |            |   |   |   |   |   |   |   |   |    |    |    |