

# **APLIKASI PENGELOLAAN VIRTUAL PRIVATE SERVER**

**JURNAL**



**Disusun Oleh :**

**Hamonangan Kinantan Prabu**

**123070222**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”  
YOGYAKARTA  
2014**

# APLIKASI PENGELOLAAN VIRTUAL PRIVATE SERVER

Disusun oleh Hamonangan Kinantan Prabu S.T.

Dosen Pembimbing Dessyanto Boedi P.,S.T.,M.T., Heru Cahya Rustamaji,S.Si.,M.T.

Jurusan Teknik Informatika Fakultas Teknologi Industri UPN “Veteran” Yogyakarta

---

## ABSTRAK

Berdasarkan perkembangan teknologi yang mengarah kepada bentuk digital, suatu individu atau lembaga baik itu profit maupun non-profit memerlukan sebuah media atau tempat penyimpanan data yang dapat diakses dengan mudah dan secara berkesinambungan, atau yang biasa dikenal dengan sebutan *server*. Dikarenakan masalah biaya dan mobilitas yang terbatas dikembangkanlah suatu cara untuk membangun suatu *server* yang “tidak nyata” yakni *virtual server*. Tanpa mengabaikan kekurangan-kekurangan *virtual server* diharapkan dengan teknik ini masyarakat dapat menjangkau tujuannya tanpa mempermasalahkan efisiensi biaya dan efektifitas pengelolaan dan pengolahan data yang dimiliki.

Adapun metodologi yang digunakan untuk membangun aplikasi ini adalah Metode *Waterfall*. Perangkat Lunak yang dihasilkan adalah aplikasi berbasis web yang berfungsi untuk mengelola *server-server virtual*. Bahasa pemrograman yang digunakan adalah PHP, HTML, Javascript, dan BASH, sedangkan perangkat lunak yang digunakan adalah Smultron, Cyberduck, Nginx, dan MySQL sebagai basis datanya.

Aplikasi ini dapat mempermudah *system administrator* dalam mengelola *server-server virtual* serta membantu pengguna untuk menggunakan dan menjalankan *virtual server* tanpa mengalami kesulitan yang berarti. Diharapkan dengan adanya aplikasi ini dapat menjadi motivasi baru untuk masyarakat agar siap dalam menghadapi era digital yang semakin dekat di depan mata.

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang Masalah**

Teknologi informasi khususnya *internet (interconnection-network)* mengalami kemajuan yang sangat pesat dari tahun ke tahun, hal ini mengakibatkan kebutuhan akan adanya komputer untuk menyediakan ruang atau tempat yang memiliki jaringan terhubung untuk menyimpan dan memproses data serta informasi baik itu bagi individu, organisasi, maupun perusahaan semakin meningkat. Komputer-komputer inilah yang dimaksud dengan *server computer* atau lebih dikenal dengan sebutan *server*. Seiring dengan meningkatnya kebutuhan *server*, kebutuhan akan perangkat keras untuk membangun sebuah *server* pun semakin meningkat, namun dikarenakan besarnya biaya untuk itu maka dibutuhkan alternatif lain yang tidak memerlukan biaya besar, yakni *Virtual Private Server*.

*Virtual Private Server (VPS)* merupakan salah satu bentuk implementasi dari *Virtual Computer* atau *Virtual Machine* yang terhubung dengan internet. Seiring dengan perkembangannya VPS menjadi salah satu pondasi kuat dalam dunia *hosting* dan/atau *web*, hal ini dikarenakan biaya yang jauh lebih murah dibandingkan dengan *server* fisik, keleluasaan yang luas hampir sama seperti menggunakan *server* fisik, serta kemudahan dalam menghadapi permasalahan pada sistem menjadi faktor utama pemicu perkembangannya seperti perpindahan atau migrasi data dari satu *server* ke *server* yang lain, instalasi ulang sistem operasi, proses *backup* yang lebih mudah dan cepat, dan masih banyak lagi keunggulan-keunggulan lain dari VPS. Meskipun demikian, masih sedikit sekali pengembang-pengembang perangkat lunak panel kontrol untuk mengelola VPS tersebut.

#### **1.2 Rumusan Masalah**

Berdasarkan latar belakang masalah tersebut, maka rumusan masalah yang dapat diambil adalah bagaimana membuat sebuah aplikasi panel kontrol yang berfungsi untuk mempermudah pengelolaan VPS agar lebih efisien bagi sistem administrator serta fitur-fitur yang dibutuhkan bagi pengguna sehingga dapat terjaga efektifitasnya.

#### **1.3 Batasan Masalah**

Penelitian ini menekankan pada pembuatan panel kontrol untuk mengelola VPS yang mencakup beberapa hal, antara lain :

1. Penelitian hanya dibatasi pada sisi *system administrator* dan *user* saja, yaitu

membuat aplikasi berbasis web untuk manajemen *VPS* dan jaringan dalam lingkup *server* fisik dan *VPS*.

2. Penelitian tidak membahas cara kerja API dari perangkat lunak LibVirt dan Virt-install yang merupakan dasar dari pembuatan aplikasi ini.

#### **1.4 Tujuan Penelitian**

Penelitian ini bertujuan untuk membuat aplikasi perangkat lunak yang berfungsi untuk pengelolaan *VPS* dan memantau kesehatan *server* dari sisi *system administrator*, serta antarmuka yang memudahkan pengguna dalam mengelola *VPS*.

#### **1.5 Manfaat Penelitian**

Manfaat dari penelitian ini adalah untuk memudahkan *system administrator* dalam mengelola sumber daya dan memantau kondisi *server*, serta mengurangi kompleksitas penggunaan bagi *user client*.

#### **1.6 Metodologi Penelitian**

Metodologi penelitian yang digunakan dalam pengembangan sistem adalah dengan metode *waterfall* (Pressman, 2010). Langkah-langkah metode *waterfall* adalah sebagai berikut :

- a) *Communication* (Komunikasi),
- b) *Planning* (Perencanaan),
- c) *Modelling* (Pemodelan),
- d) *Construction* (Pembangunan),
- e) *Deployment* (Penyebaran).

Tahapan pada penelitian ini hanya mencapai tahap *Construction*.

#### **1.7 Sistematika Penulisan**

Sistematika penulisan tugas akhir ini adalah sebagai berikut :

##### **BAB I : PENDAHULUAN**

Bab ini berisi tentang latar belakang kajian yang dilakukan, permasalahan yang terjadi, batasan yang dilakukan, tujuan dan manfaat penelitian, metodologi penelitian dan sistematika penulisan.

##### **BAB II : LANDASAN TEORI**

Bab ini berisi tentang aplikasi dan teori-teori yang mendukung penyelesaian masalah, yang meliputi teori tentang *VPS*, Virtualisasi Komputer/Server, Qemu, Linux-KVM, LibVirt, Virt-install, (Twitter) Bootstrap, adminer, Web Browser, Web Server, *VPS*, PHP, MySQL, Javascript, HTML, DFD, metode *waterfall*, basis data dan studi pustaka.

### BAB III : ANALISIS DAN PERANCANGAN

Bab ini diuraikan tentang rancangan implementasi sistem.

### BAB IV : IMPLEMENTASI SISTEM

Bab ini menjelaskan implementasi dari sistem yang dibangun.

### BAB V : PENUTUP

Bab ini berisi kesimpulan dan saran yang merupakan hasil dari pelaksanaan penelitian.

## **BAB II**

### **DASAR TEORI**

#### **2.1 Aplikasi**

Aplikasi adalah program yang dibuat oleh pemakai yang ditujukan untuk melakukan tugas khusus (Kadir, 2003).

#### **2.2 Jaringan Komputer**

Definisi komputer adalah suatu perangkat elektronik untuk menyimpan dan memproses data yang biasanya dalam bentuk satuan biner, berdasarkan dari instruksi-instruksi yang diberikan oleh program (Oxford, 2005).

Jaringan (Komputer) adalah suatu kelompok dari beberapa komputer, mesin, atau operasi yang saling terhubung (Oxford, 2005).

Jaringan komputer adalah sekumpulan peralatan atau komputer yang saling dihubungkan untuk berbagi sumber daya (Micro, 2012).

#### **2.3 Virtual Private Server**

Arti dari kata virtual dalam komputer dapat didefinisikan sebagai suatu bentuk yang secara fisik tidak ada tapi dibuat oleh suatu perangkat lunak untuk ada (Oxford, 2005).

Pada tahun 1960 awal virtualisasi komputer mulai diperkenalkan kepada khalayak umum dengan dipelopori oleh perusahaan-perusahaan besar yang bergerak di bidang komputer yaitu International Business Machines (IBM) dan General Electric (GE). Hal tersebut dipicu dengan adanya suatu proyek penelitian dari Massachusetts Institute of Technology (MIT), pada saat itu peneliti MIT menginginkan suatu sistem komputer yang dapat digunakan untuk melakukan pekerjaan yang banyak secara simultan, selain itu Bell Laboratories (Bell Labs) juga yang menginginkan perangkat keras dan lunak yang serupa (<http://www.cs.gmu.edu/>).

Server (Komputer) adalah suatu komputer atau program komputer terpusat yang mengelola sumber daya atau layanan di dalam satu jaringan (Oxford, 2005).

*Virtual Private Server* (VPS) merupakan suatu bentuk implementasi dari virtualisasi komputer dan merupakan yang paling populer, perbedaan paling mendasar dari VPS dengan virtual komputer terletak pada ada tidaknya koneksi *internet*.

Salah satu kegunaan ataupun keunggulan lain dari VPS adalah sebagai berikut :

- **Hosting** : Dapat menjalankan hosting yang multi fungsi (karena mampu

menjalankan banyak server dalam satu node)

- Eksperimen : Dapat melakukan penelitian ataupun pengerjaan layanan baru tanpa harus mengganggu server yang sudah dalam tahap produksi
- Edukasi : Setiap siswa/mahasiswa memiliki servernya sendiri berikut dengan password *root*-nya.
- Mengelola berbagai macam versi dari server dan proyek yang sama. (<http://www.solucorp.qc.ca/>).

Popularitas *Virtual Server* semakin meningkat selama satu dekade belakangan ini, terutama pada pihak pengembang dan perusahaan atau industri yang bergerak dibidang tersebut. Akan tetapi terdapat masalah pada pihak pengguna (*End User*) yang masih belum percaya dengan *Virtual Server* dikarenakan masih kurangnya transparansi dari perusahaan penyedia layanan tersebut dan juga aplikasi atau sistem panel kontrol yang digunakan dengan tidak adanya informasi atau detail penggunaan sumber daya suatu node server (Hernick, 2008).

## **2.4 Application Programming Interface (API)**

*Application Programming Interface* (API) adalah sekumpulan perintah, fungsi, dan protokol yang dapat digunakan oleh *programmer* saat membangun perangkat lunak untuk sistem operasi tertentu. API memungkinkan *programmer* untuk menggunakan fungsi standar untuk berinteraksi dengan sistem operasi. API dapat juga dihubungkan sebagai *framework*. *Framework* adalah sekumpulan perintah/fungsi dasar yang dapat membantu dalam menyelesaikan proses-proses yang lebih kompleks. (<http://wikipedia.org/>).

## **2.5 PHP Hypertext Preprocessor 5 (PHP 5)**

PHP merupakan bahasa bentuk *script* yang ditempatkan dalam *server* dan proses oleh *server*, kemudian hasilnya akan dikirim ke *client* menggunakan *browser* (Kadir, 2002). PHP diciptakan oleh Rasmus Lerdorf (Sutarman, 2002), untuk menghubungkan antara database dengan HTML dalam proses perhitungan.

## **2.6 MySQL**

*Structured Query Language* (SQL) merupakan sebuah bahasa relasional yang berisi pernyataan-pernyataan yang digunakan untuk memasukkan, mengubah, menghapus, memilih dan melindungi data (Prihatna, 2005). SQL bukan merupakan aplikasi basis data, namun lebih kepada suatu bahasa standar internasional yang digunakan untuk memanipulasi basis data (Oxford, 2005). Sistem basis data lain yang memiliki konsep sama dengan MySQL adalah PostgreSQL, SQLite, Oracle, MsSQL.

## 2.7 Qemu dan Linux-KVM

Qemu adalah emulator mesin yang bersifat generik dan *open source*. Yang dimaksud dengan emulator mesin adalah Qemu mampu menjalankan sistem operasi atau perangkat lunak yang hanya bisa dijalankan oleh satu perangkat lunak (ARM misalnya) di dalam mesin yang berbeda seperti PC dengan menggunakan penerjemah dinamik. (<http://qemu.org/>).

Linux-KVM atau lebih sering disingkat KVM adalah kepanjangan dari *Linux Kernel-based Virtual Machine*. KVM merupakan salah satu solusi virtualisasi penuh untuk Linux pada perangkat keras x86 yang memiliki ekstensi virtualisasi (Intel VT atau AMD-V). KVM juga memerlukan Qemu yang telah dimodifikasi untuk menjalankan virtualisasi tersebut (<http://www.linux-kvm.org>).

Keuntungan dan kelebihan dari virtualisasi dengan menggunakan KVM ini adalah dukungan penuh dari Linux (kernel) yang secara signifikan mengurangi *overhead* pada penggunaan *resource* server.

## 2.8 Libvirt dan Virtinst

Libvirt merupakan perangkat lunak untuk berinteraksi dengan kemampuan-kemampuan virtualisasi dari versi terbaru Linux (dan juga sistem operasi lainnya). Hingga saat penulisan skripsi ini Libvirt telah mendukung hampir seluruh jenis virtualisasi yang ada, seperti Qemu/KVM, OpenVZ, Xen, LXC, Virtualbox, VMWare, Microsoft Hyper-V, dan lain sebagainya (<http://libvirt.org/>).

Virt-install adalah bagian dari paket Virt-manager yang merupakan aplikasi antarmuka yang berfungsi untuk mengelola mesin virtual melalui Libvirt. Target utamanya adalah untuk mengelola mesin virtual KVM, akan tetapi bisa juga digunakan untuk mengelola virtualisasi Xen dan LXC. Sedangkan Virt-install adalah suatu alat di *command-line* yang digunakan untuk mempermudah proses pembagian dan pembuatan sistem operasi pada mesin virtual (<http://virt-manager.org/>).

## 2.9 Bootstrap

Bootstrap awalnya dibuat dan digunakan oleh desainer serta pengembang situs jejaring sosial Twitter, bootstrap menjadi *framework* yang paling populer dan proyek sumber terbuka di dunia (<http://getbootstrap.com/>). Bootstrap menggunakan JQuery, HTML 5, CSS 3, serta pustaka monokromatik ikon dan simbol Glyphicon sebagai landasan *framework*.



## 2.10 Impress.js

Impress.js merupakan alat untuk presentasi yang berbasis CSS3 yang mengutamakan pada bagian transisi dan perubahan bentuk sebagai pembentuk utama halaman antarmukanya (<http://bartaz.github.io/impress.js/>).

## 2.11 PHPSeclib

PHPSeclib adalah salah satu pustaka atau *library* yang berfungsi untuk menghubungkan PHP dengan SSH pada *server*, pustaka ini digunakan karena kemudahan dan kompatibilitas yang luas sehingga dapat digunakan dari PHP 4 hingga PHP 5 (<http://phpseclib.sourceforge.net/>).

## 2.12 Model Pengembangan Sistem Sekuensial Linier (Waterfall Model)

Model air terjun (*waterfall model*) atau disebut juga sebagai siklus hidup klasik merupakan sebuah model pendekatan yang sistematis berurutan (*sequential linear*) untuk pengembangan sebuah perangkat lunak atau sistem yang diawali dengan *Communication*, *Planning*, *Modelling*, *Construction* hingga *Deployment* secara berkelanjutan hingga perangkat lunak tersebut selesai dibuat (Pressman, 2010). Berikut urutannya :

1. *Communication* (komunikasi)
2. *Planning* (perencanaan)
3. *Modelling* (pemodelan)
4. *Construction* (pembangunan)
5. *Deployment* (penyebaran)

## 2.13 Basis Data

Basis data adalah penggunaan bersama dari data yang terhubung secara logis dan deskripsi dari data, yang dirancang untuk keperluan informasi (Connolly et al, 2002).

## 2.14 Entity Relationship Diagram

*Entity Relationship Diagram* (ERD) adalah model data untuk menggambarkan hubungan antara satu entitas dengan entitas lain yang mempunyai relasi (hubungan) dengan batasan-batasan.

Derajat hubungan menyatakan jumlah anggota entitas yang terlibat didalam ikatan yang terjadi dalam membentuk hubungan. Derajat hubungan pada *relationship tipe*, batasan *structural cardinality ratio* memiliki jenis (Waljiyanto, 2003) :

1. 1:1 (*One to one relationship*)  
Hubungan antara file pertama dengan file kedua adalah satu berbanding satu.
2. 1:M (*One to many relationship*)

Hubungan antar file pertama dengan file kedua adalah satu berbanding banyak atau dapat pula dibalik menjadi banyak berbanding satu.

3. M:M (*Many to many relationship*)

Hubungan antar satu atribut dengan atribut yang lain dalam satu file yang sama mempunyai hubungan banyak lawan banyak.

## **2.15 Data Flow Diagram**

*Data Flow Diagram* atau yang lebih sering disingkat DFD merupakan sebuah teknik grafis yang menggambarkan aliran informasi dan transformasi yang diaplikasikan pada saat data bergerak dari input menjadi output (Pressman, 2002).

## **2.16 Studi pustaka**

Pada penelitian yang dibuat oleh Ryuichi Ogawa, Yoshiharu Maeno, dan Masayuki Nakae dengan judul "*Authority Management Infrastructure for The Virtual Server Integrated Environment*" (Ogawa, 2010), yang secara garis besar menekankan pada tata cara pembagian kewenangan pengelolaan setiap pengguna sehingga hal-hal seperti akses tidak resmi (ilegal) dan kebocoran informasi dapat dihindari.

## BAB III

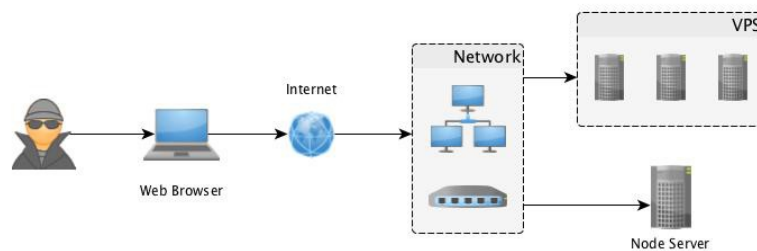
### ANALISIS DAN PERANCANGAN

#### 3.1 Analisis Sistem

Aplikasi Pengelolaan Virtual Private Server menggunakan bahasa pemrograman PHP ini merupakan suatu aplikasi yang berfungsi untuk menjaga efisiensi suatu *stakeholder* agar mendapatkan hasil maksimal dengan biaya yang minimal serta menjadi aplikasi yang efektif bagi *system administrator* maupun *user* dalam penggunaan dan pengelolaannya.

#### 3.2 Arsitektur Sistem

Arsitektur sistem ini dibuat untuk mengetahui bagaimana alur pengguna mendapatkan informasi yang dibutuhkan, proses pengolahan data dari admin, serta bagaimana bisa bersifat *user friendly*.



**Gambar 3.1** Arsitektur Sistem

#### 3.3 Rancangan Proses dengan Data Flow Diagram (DFD)

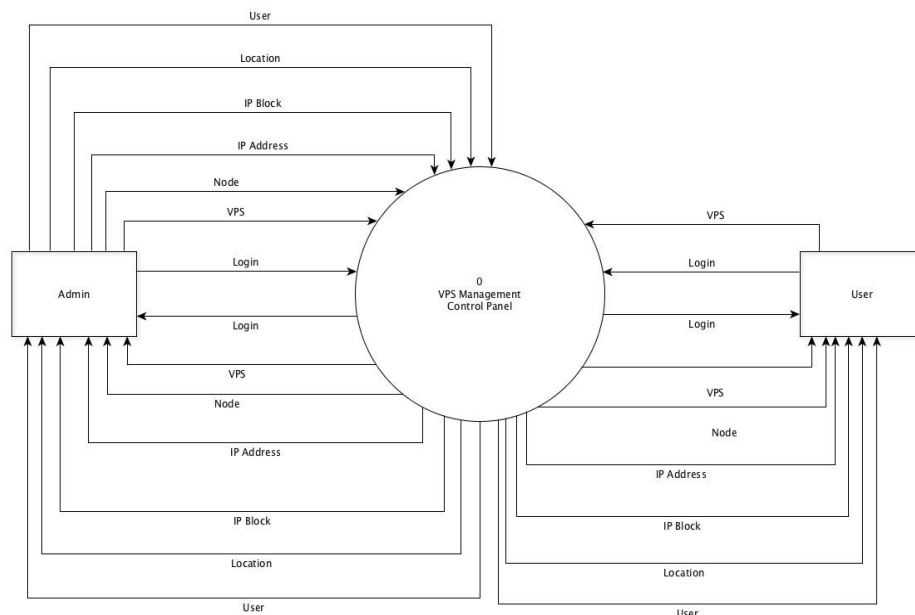
Hal pertama yang dilakukan dalam membuat DFD untuk merancang sebuah sistem adalah membuat diagram konteks yang menggambarkan sistem secara keseluruhan, kemudian dilanjutkan dengan level-level berikutnya.

##### 3.5.1 Rancangan DFD Level 0

Dalam aliran data ini terdapat dua entitas, yaitu 'Admin' dan 'User'. Aliran data Admin dari entitas ke sistem adalah proses *login*, *user*, *node server*, *VPS server*, lokasi *server*, IP Block, dan alamat IP. Untuk proses admin dari sistem ke entitas prosesnya sama yaitu proses *login*, *user*, *node server*, *VPS server*, lokasi *server*, IP Block, dan alamat IP.

Aliran data pada *user* dari entitas ke sistem hanya terdiri dari beberapa proses, sedangkan proses dari sistem ke entitas masih sama dengan aliran data admin, yaitu berupa proses *login*, *user*, *node server*, *VPS server*, lokasi *server*, IP Block, dan alamat IP. Sedangkan dari entitas ke sistem berupa proses login dan VPS. Untuk lebih jelasnya bisa

dilihat pada Gambar 3.2.

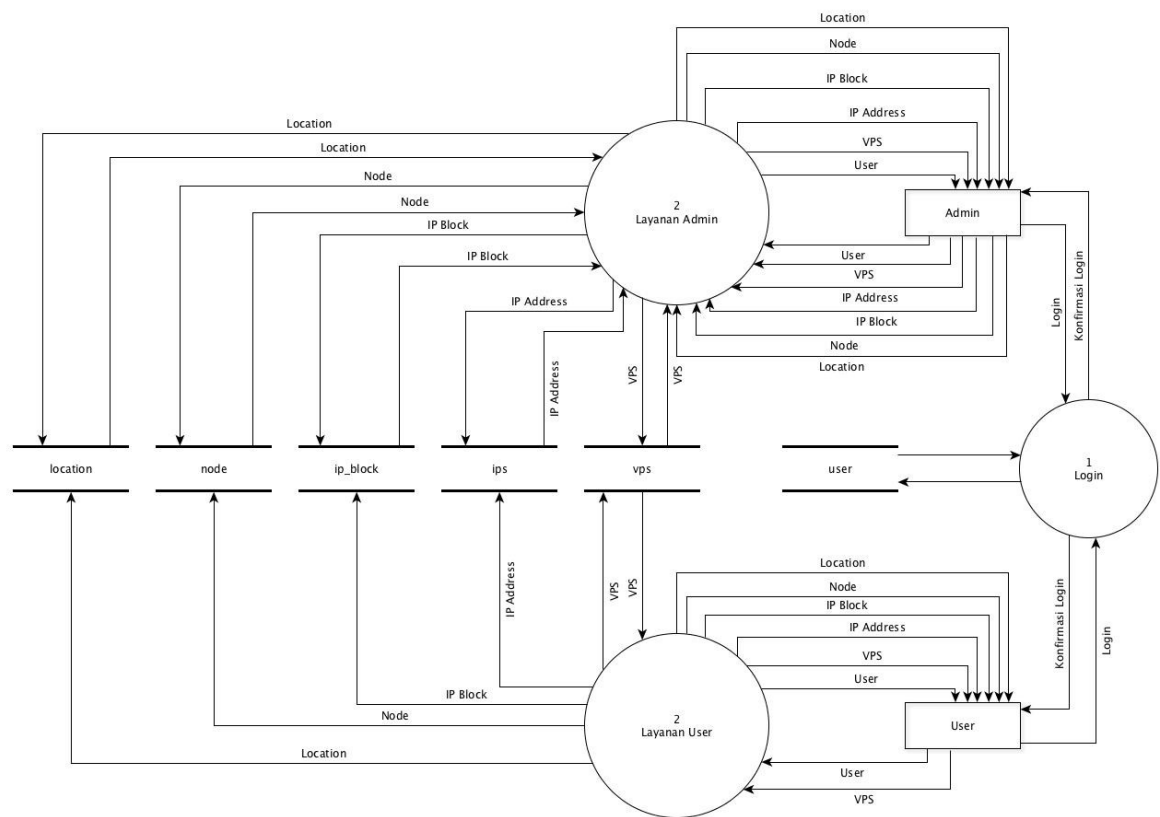


**Gambar 3.2** DFD Level 0

### 3.5.2 Rancangan DFD Level 1

Pada sistem DFD Level 1 ini dimana menggambarkan jalannya sistem yang semua prosesnya melibatkan enam (6) tabel *database*, yaitu *user*, *location*, *node*, *ip\_block*, *ips*, dan *vps*.

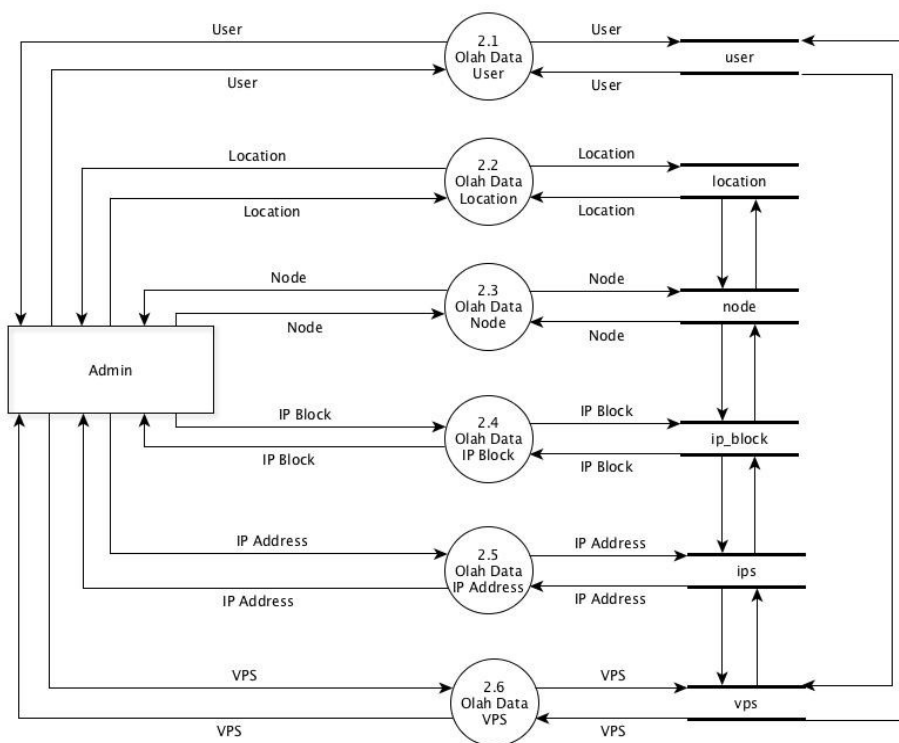
Dalam DFD Level 1 ini ada tiga proses yang dilakukan, yaitu proses login, layanan admin, dan layanan user. Pada proses pertama yaitu proses login disini admin dan user dapat mengakses halaman utama setelah melakukan login. Selanjutnya proses layanan admin disini admin terhubung dengan lima tabel lain yang ada dalam database. Admin dapat melakukan berbagai proses pengolahan data dalam halaman admin dan admin memiliki aliran data bolak-balik di setiap tabel sehingga dapat di pahami bahwa admin dapat melakukan penambahan, penghapusan, perubahan, dan melihat apa yang ada dalam sistem tersebut. Proses terakhir adalah proses layanan user. Dalam proses ini user hanya diberi kewenangan dalam sistem yaitu melihat dan menambah. Selain dapat melihat informasi yang ada, *user* hanya bisa mengubah konfigurasi VPS dan profil user tersebut tidak lebih dari itu. Hal ini dapat dilihat pada gambar 3.3.



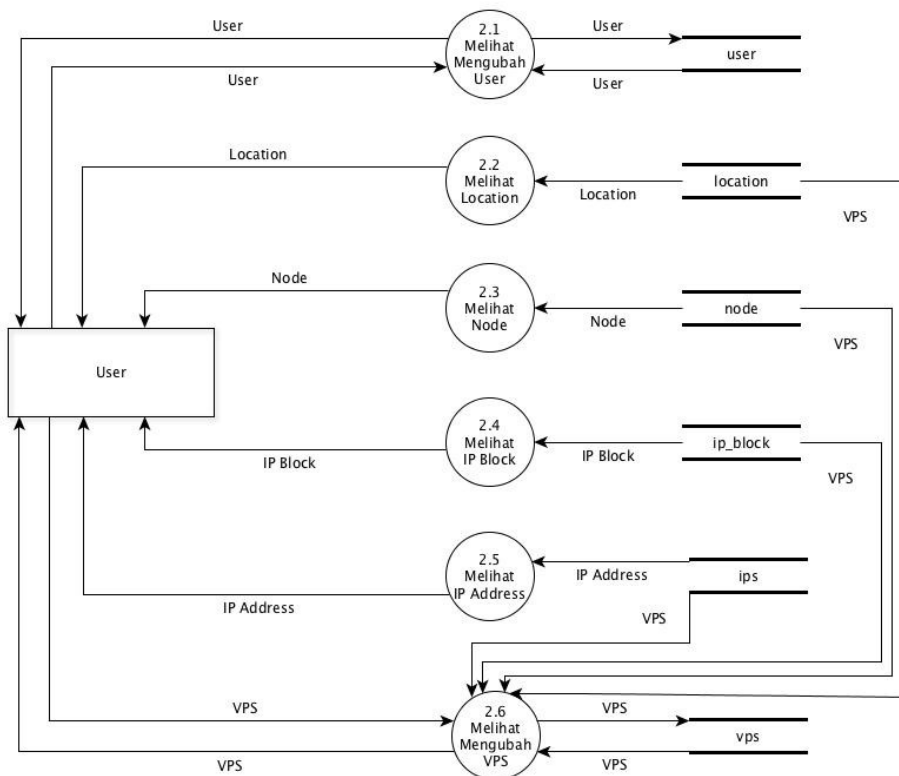
**Gambar 3.3 DFD Level 1**

### 3.5.3 Rancangan DFD Level 2

Pada Level ini terdapat dua Data Flow Diagram yaitu Admin dan User, yang masing-masing diturunkan dari entitas pada Level 1. Berikut Gambar 3.4 dan Gambar 3.5 dari kedua diagram tersebut :



**Gambar 3.4 DFD Level 2 Admin**



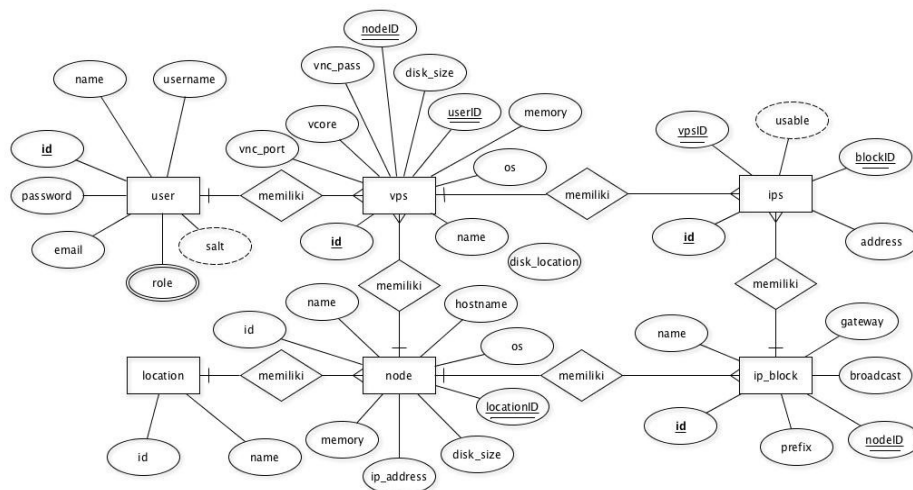
**Gambar 3.5 DFD Level 2 User**

### 3.4 Perancangan Basis Data

Pada sistem ini ada sebuah *database* dengan nama vmcp dimana *database* ini berfungsi sebagai penampung data yang di perlukan pada sistem ini. Dalam *database* tersebut ada enam tabel yang berupa *user* dimana tabel ini menampung data dari admin dan *user* untuk melakukan *login* pada sistem, *location* digunakan untuk menampung data dari lokasi yang akan digunakan untuk menandai lokasi *Node Server*. *Node* berisikan data-data dari *server* fisik atau *node*, *ip\_block* tabel ini diisi oleh data kumpulan alamat IP yang telah dikumpulkan per-block, *ips* berisikan alamat-alamat IP, *vps* adalah tabel yang diisi dengan data-data VPS yang telah dibuat.

#### 3.6.1 Perancangan ERD

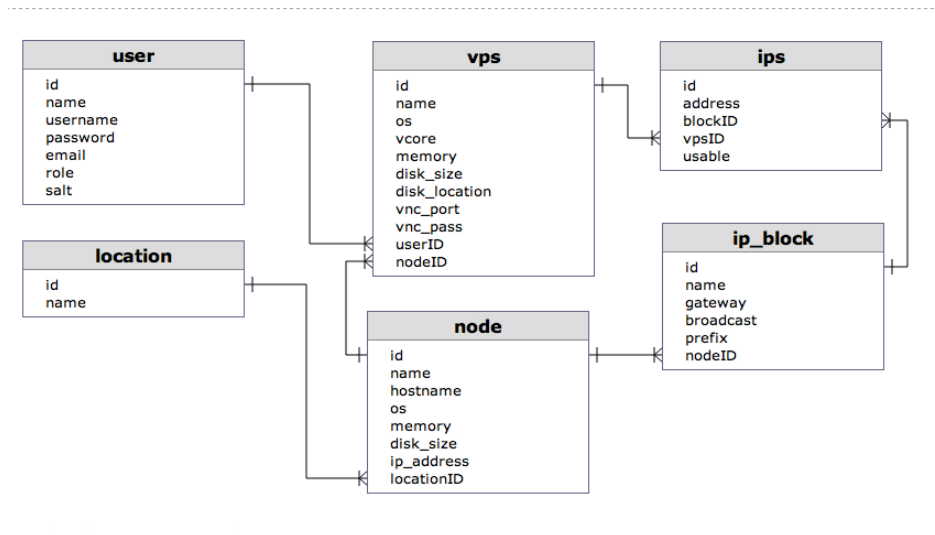
Secara keseluruhan, sistem ini dapat digambarkan menggunakan *Entity Relationship Diagram* (ERD) seperti pada gambar 3.6



**Gambar 3.6** Entity Relationship Diagram

#### 3.6.2 Relasi Antar Tabel (RAT)

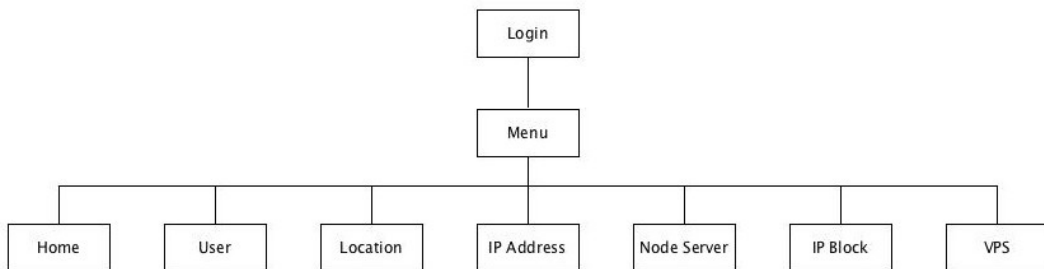
Dalam *database* ini tabel saling berhubungan, dan terdapat tiga tabel yang saling terkait dengan tabel *vps* yaitu tabel *user*, *node*, dan *ips*. Untuk lebih jelasnya dapat dilihat pada gambar 3.7



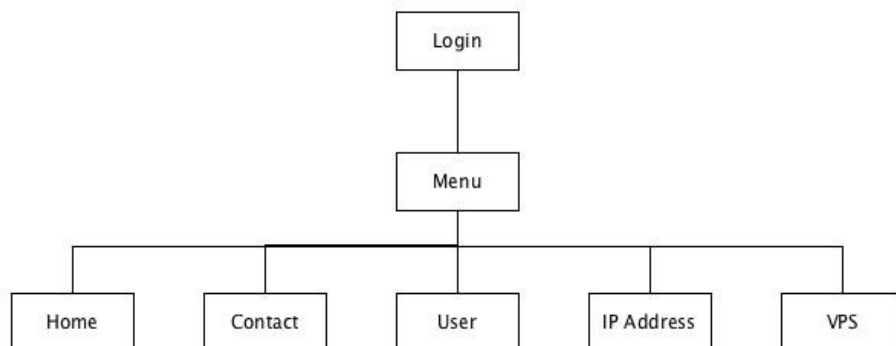
**Gambar 3.7** Relasi Antar Tabel (RAT)

### 3.7 Skema Antar Muka

Skema struktur menu pada sistem ini dapat dilihat pada Gambar 3.8 dan Gambar 3.9.



**Gambar 3.8** Skema Antar Muka Admin



**Gambar 3.9** Skema Antar Muka User



## BAB IV

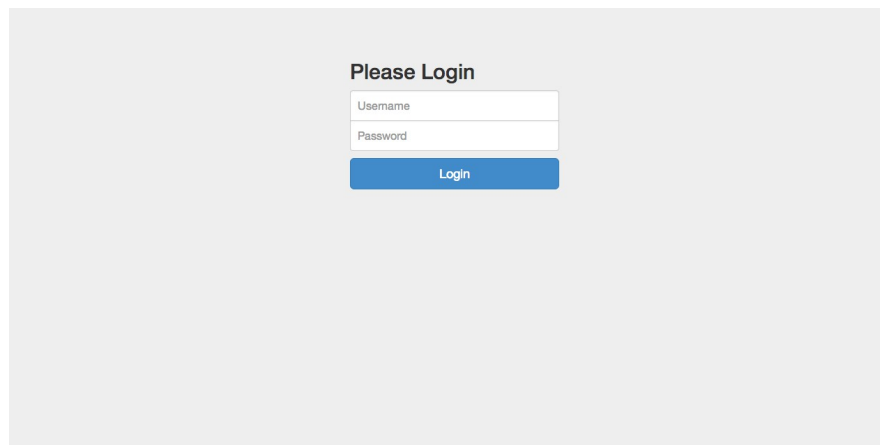
### IMPLEMENTASI

#### 4.1 Implementasi

Implementasi bertujuan untuk membuat versi operasional pertama dari sistem berdasarkan arsitektur desain yang telah dirancang dan dibangun berdasarkan metode penelitian yang digunakan.

##### 4.1.1 Halaman *Login*

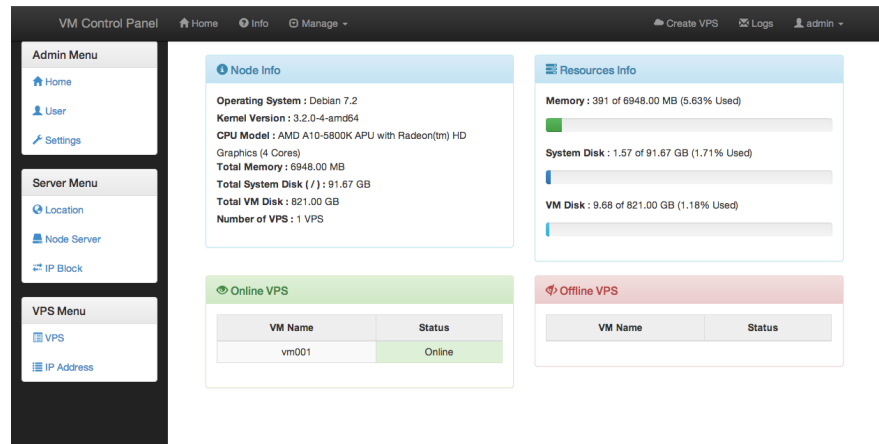
Pada halaman *login* terdapat 2 *form* masukan berupa nama user (*username*) dan password. Saat tombol login ditekan maka terjadi proses untuk memeriksa apakah masukan tersebut salah atau tidak, jika benar maka akan dilakukan proses untuk memeriksa wewenang pengguna (*role*), jika *valid* maka pengguna tersebut diijinkan untuk masuk ke dalam halaman *Home* berdasarkan *role* yang dimiliki.

The image shows a login form on a light gray background. At the top, the text "Please Login" is displayed. Below it are two white input fields with gray borders. The first field is labeled "Username" and the second is labeled "Password". Below these fields is a blue button with the word "Login" in white text.

**Gambar 4.1** *Login Form*

##### 4.1.2 Halaman *Home Admin*

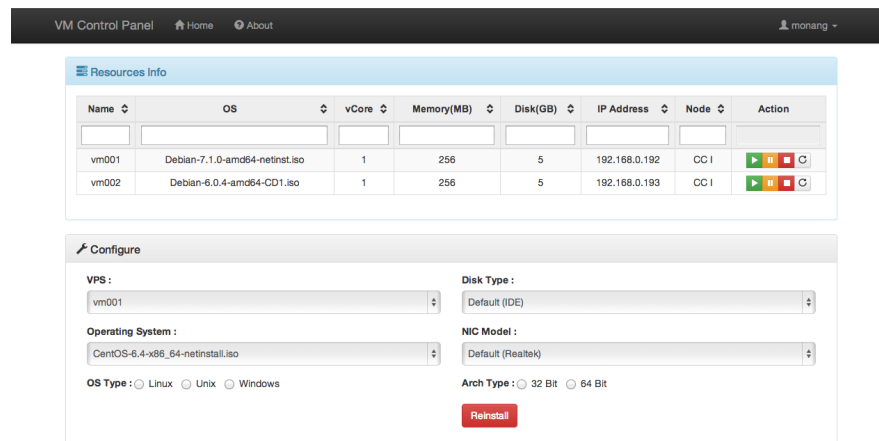
Pada halaman *Home Admin* terdapat beberapa tampilan informasi-informasi umum tentang spesifikasi server, daftar VPS yang sedang berjalan dan/atau mati, total, dan sumber daya (*resources*) yang telah digunakan.



**Gambar 4.2** Halaman *Home Admin*

#### 4.1.3 Halaman *Home User*

Seperti halnya halaman *Home admin*, pada halaman *user* ini terdapat informasi-informasi yang diperlukan oleh *user*, seperti daftar VPS yang dikelola oleh pengguna tersebut, aksi-aksi seperti mematikan dan menghidupkan VPS, serta menu untuk melakukan instalasi ulang Sistem Operasi VPS tertentu.



**Gambar 4.3** Halaman *Home User*

#### 4.1.4 Halaman *Input dan Output Lokasi (Admin)*

Pada halaman *Input* terdapat *form* bagi *admin* untuk memasukan lokasi-lokasi pusat data (*datacenter*) yang akan digunakan untuk menyimpan *node server* atau server utama yang akan digunakan sebagai wadah penyimpanan VPS-VPS. Sedangkan didalam halaman *Output* lokasi terdapat daftar lokasi-lokasi yang telah dimasukkan sebelumnya.

VM Control Panel Home Info Manage Create VPS Logs admin

**Admin Menu**

- Home
- User
- Settings

**Server Menu**

- Location
- Node Server
- IP Block

**VPS Menu**

- VPS
- IP Address

### Create new Location

Name :

Location name (ex. Datacenter I, etc)

Save List Location

**Gambar 4.4** Halaman *Input* Lokasi

VM Control Panel Home Info Manage Create VPS Logs admin

### List of Location

Name	Action
Babarsari 1	Hapus

+ Add Location

**Gambar 4.5** Halaman *Output* Lokasi

#### 4.1.5 Halaman *Input* dan *Output Node Server (Admin)*

Pada halaman *Input Node Server* terdapat masukan-masukan tentang spesifikasi server utama yang diperlukan, seperti jumlah memori utama, cakram keras, sistem operasi, dan lain sebagainya, sedangkan pada halaman *output* terdapat hasil dari masukan yang telah dilakukan sebelumnya.

VM Control Panel

Admin Menu

- Home
- User
- Settings

Server Menu

- Location
- Node Server
- IP Block

VPS Menu

- VPS
- IP Address

Create new Node Server

Name :

Hostname :

Operating System :

Memory Size :

Disk Size :

IP Address :

Location :

**Gambar 4.6** Halaman *Input Node Server*

VM Control Panel

Admin Menu

- Home
- User
- Settings

Server Menu

- Location
- Node Server
- IP Block

VPS Menu

- VPS
- IP Address

List of Node Server

Node Name	Hostname	Operating System	Memory Size	Disk Size	IP Address	Location	Action
BS 1	skripsi	Debian 7 64 Bit	4096	200	192.168.0.124	Babarsari 1	<input type="button" value="Update"/> <input type="button" value="Hapus"/>

**Gambar 4.7** Halaman *Input Node Server*

#### 4.1.6 Halaman *Input dan Output Block IP Address (Admin)*

Pada halaman *Input dan Output Block IP Address* terdapat masukan dan keluaran untuk data suatu sekumpulan alamat IP yang telah dikelompokkan berdasarkan *prefix*, *gateway*, dan *broadcast* yang sama.

VM Control Panel

Admin Menu

- Home
- User
- Settings

Server Menu

- Location
- Node Server
- IP Block

VPS Menu

- VPS
- IP Address

Create new IP Block

Name :

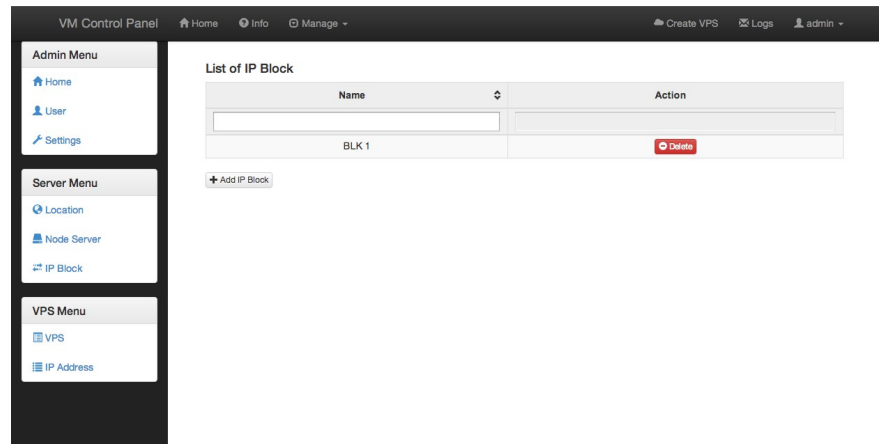
Gateway :

Broadcast :

Prefix :

Node Server :

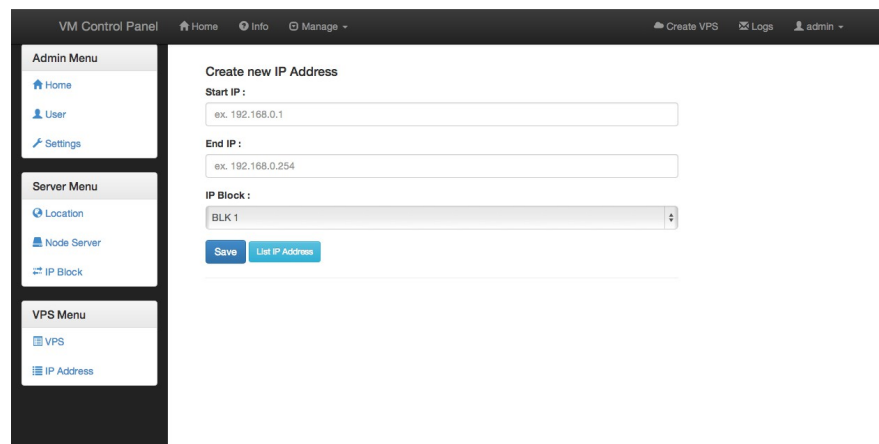
**Gambar 4.8** Halaman *Input Block IP Address*



**Gambar 4.9** Halaman *Output Block IP Address*

#### 4.1.7 Halaman *Input dan Output* Alamat IP (*Admin*)

Dalam halaman *Input* alamat IP terdapat *form* masukan untuk rentang alamat IP yang akan dimasukkan, setelah tombol “Save” ditekan maka akan terjadi proses perulangan yang menambahkan alamat-alamat IP tersebut ke dalam *database*. Sedangkan pada halaman *output* akan ditampilkan daftar alamat IP beserta ketersediaannya.



**Gambar 4.10** Halaman *Input* Alamat IP

IP Address	Assigned to VPS	Block	Action
192.168.0.124		BLK 1	Hapus
192.168.0.125		BLK 1	Hapus
192.168.0.126		BLK 1	Hapus
192.168.0.127		BLK 1	Hapus
192.168.0.128		BLK 1	Hapus
192.168.0.129		BLK 1	Hapus
192.168.0.130		BLK 1	Hapus
192.168.0.131		BLK 1	Hapus
192.168.0.132		BLK 1	Hapus
192.168.0.133		BLK 1	Hapus

**Gambar 4.11** Halaman *Output* Alamat IP

#### 4.1.8 Halaman *Input* dan *Output User (Admin)*

Di dalam halaman *Input* dan *Output User* terdapat masukan dan keluaran untuk data-data pengguna yang diperlukan, seperti nama, *username*, *password*, alamat *email*, dan lain sebagainya.

**Create new User**

Name :  
Your full name (ex. John Smith)

Email :  
Your email address (ex. john@example.com)

Username :  
Your username (ex. johnsmith)

Password :  
Your password

Salt :  
\$1\$cAKw6EgT\$

Role :  
Normal User

[Save](#) [List User](#)

**Gambar 4.12** Halaman *Input User*

Name	Role	Action
Administrator	Administrator	Ubah Hapus

[Add User](#)

**Gambar 4.13** Halaman *Output User*

#### 4.1.9 Halaman *Input* dan *Output* VPS (Admin)

Pada halaman *Input* VPS terdapat form masukan yang diperlukan untuk membangun suatu VPS beserta data-data yang diperlukan oleh pengguna baik untuk akses VNC maupun data untuk melakukan instalasi ulang sistem operasi. Data tersebut sebagian disimpan di dalam *database*, sebagian di dalam file berupa XML yang digunakan oleh Libvirt sebagai *database* perangkat lunak tersebut.

The screenshot shows the 'Create new VPS' form in the VM Control Panel. The form includes fields for Name, Operating System, OS Type, Arch Type, Number of Core(s), Memory, HDD, Disk Type, IP Address, NIC Model, User, and Node. The 'Create VPS' and 'List VPS' buttons are at the bottom.

VM Control Panel | Home | Info | Manage | Create VPS | Logs | admin

**Admin Menu**

- Home
- User
- Settings

**Server Menu**

- Location
- Node Server
- IP Block

**VPS Menu**

- VPS
- IP Address

**Create new VPS**

Name :

Operating System :

OS Type : ☐ Linux ☐ Unix ☐ Windows

Arch Type : ☐ 32 Bit ☐ 64 Bit

Number of Core(s) :

Memory :

HDD :

Disk Type :

IP Address :

NIC Model :

User :

Node :

Gambar 4.14 Halaman *Input* VPS

The screenshot shows the 'List VPS' table in the VM Control Panel. The table has columns for Name, OS, vCore, Memory(MB), Disk(GB), Owner, Node, and Action. The table contains one row for 'vm001' with details: Debian-7.1.0-amd64-netinst.iso, 4 vCore, 2048 Memory(MB), 200 Disk(GB), admin Owner, BS I Node, and an Action button.

VM Control Panel | Home | Info | Manage | Create VPS | Logs | admin

**Admin Menu**

- Home
- User
- Settings

**Server Menu**

- Location
- Node Server
- IP Block

**VPS Menu**

- VPS
- IP Address

Name	OS	vCore	Memory(MB)	Disk(GB)	Owner	Node	Action
vm001	Debian-7.1.0-amd64-netinst.iso	4	2048	200	admin	BS I	<input type="button" value="VNC"/>

Gambar 4.15 Halaman *Output* VPS

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan hasil dari penelitian ini, maka dapat diambil kesimpulan bahwa telah berhasil dibangun suatu sistem pengelolaan *Virtual Private Server* (VPS) yang dapat membantu *System Administrator* (*SysAdmin*) mengelola sumber daya suatu server yang lebih efisien dan efektif serta menghasilkan Panel Kontrol yang mudah untuk dipahami dan digunakan. Diharapkan penelitian ini dapat dilanjutkan hingga mencapai tahap penyebaran (*Deployment*) sehingga dapat berguna dan bermanfaat bagi masyarakat banyak.

#### **5.2 Saran**

Saran yang dapat diberikan untuk mengembangkan sistem ini selanjutnya adalah sebagai berikut :

1. Menambahkan dukungan terhadap jenis virtualisasi yang lain seperti Xen, OpenVZ, LXC, dan sebagainya sesuai dengan dukungan yang telah disediakan oleh LibVirt
2. Melakukan instalasi sistem operasi tidak hanya menggunakan file ISO tetapi juga bisa melalui *templates* atau jaringan menggunakan PXE (*Preboot eXecution Environment* yang juga dikenal dengan *Pre-Execution Environment*)
3. Menambahkan migrasi VPS antar Node Server atau mengubah aplikasi ini menjadi sistem *cloud server*