

## Desain Skalabel Website Menggunakan *Elastic Load Balancing* pada Amazon *Virtual Private Cloud* (VPC)

Shinta Fesri Wandira<sup>1\*)</sup>, T. Yudi Hadiwandra<sup>2)</sup>

<sup>1)2)</sup> Program Studi Teknik Informatika, Fakultas Teknik, Universitas Riau

<sup>\*)</sup>Correspondence Author: [shinta.fesri1419@student.unri.ac.id](mailto:shinta.fesri1419@student.unri.ac.id), Pekanbaru, Indonesia

DOI: <https://doi.org/10.37012/jtik.v9i2.1810>

### Abstrak

Website sekolah adalah media informasi untuk mengetahui profil dan kegiatan sekolah. Beberapa standar yang perlu diketahui untuk suatu website yang memiliki performa tinggi seperti standar web server yang berkinerja tinggi, tingkat reliabilitas dan *high availability*. SMA Cendana Pekanbaru merupakan salah satu sekolah swasta yang berada di Pekanbaru, memiliki website yang bergabung dengan website sekolah lain seperti website sekolah SMA Cendana Mandau (Duri), SMP Cendana Pekanbaru, SD Cendana Pekanbaru dan lainnya. Website tersebut bergabung menjadi satu pada website Yayasan Pendidikan Cendana Rumbai atau YPCR. Oleh karena itu, penelitian ini dilakukan untuk membuat suatu website sekolah yang berfokus pada SMA Cendana Pekanbaru, seperti memperlihatkan keunggulan dari SMA Cendana Pekanbaru dan hal yang menjadi ciri khas atau iconic SMA Cendana Pekanbaru dengan menggunakan layanan-layanan *cloud computing* untuk menjadikan website sekolah yang *high availability* serta memiliki tingkat reliabilitas tinggi. Layanan *cloud computing* pada Amazon Web Service (AWS) seperti *Elastic Load Balancing*, *Amazon Elastic Compute Cloud* (EC2) dan *Auto Scaling Group* (ASG) serta *Virtual Private Cloud* (VPC) berhasil diimplementasikan. Hasil dari pengujian yang telah dilakukan menggunakan *Apache Jmeter*, pengujian *loadtesting* dengan 1000 traffic dalam waktu 120 detik (2 menit) menghasilkan status "*successful*" yang menunjukkan bahwa website tersebut telah *high availability* dan memiliki tingkat reliabilitas tinggi.

**Kata Kunci:** Website sekolah, *Cloud Computing*, *Elastic Load Balancing*, *Auto Scaling Group*, *Amazon Web Service*

### Abstract

The school website is an information medium for knowing the school's profile and activities. There are several standards you need to know for a website that has high performance, such as high performance web server standards, level of reliability and high availability. SMA Cendana Pekanbaru is one of the private schools in Pekanbaru, has a website that joins other school websites such as the school websites of SMA Cendana Mandau (Duri), SMP Cendana Pekanbaru, SD Cendana Pekanbaru and others. These websites merge into one website of the Cendana Rumbai Education Foundation or YPCR. Therefore, this research was carried out to create a school website that focuses on SMA Cendana Pekanbaru, such as showing the advantages of SMA Cendana Pekanbaru and the things that are characteristic or iconic of SMA Cendana Pekanbaru by using cloud computing services to create a high school website. availability and has a high level of reliability. Cloud computing services on Amazon Web Service (AWS) such as *Elastic Load Balancing*, *Amazon Elastic Compute Cloud* (EC2) and *Auto Scaling Group* (ASG) and *Virtual Private Cloud* (VPC) were successfully implemented. The results of testing carried out using *Apache Jmeter*, load testing with 1000 traffic in 120 seconds (2 minutes) resulted in a "successful" status which shows that the website has high availability and has a high level of reliability.

**Keywords:** school website, *Cloud Computing*, *Elastic Load Balancing*, *Auto Scaling Group*, *Amazon Web Service*

---

## PENDAHULUAN

Peningkatan layanan dan aplikasi web yang semakin meningkat diberbagai bidang, membuat permintaan layanan dari pengguna ikut meningkat sesuai dengan kebutuhan yang dibutuhkan. Contoh layanan dan aplikasi web yang sudah ditingkatkan dan sering digunakan yaitu aplikasi web pendidikan (*e-learning*), aplikasi web berita (*-news*), aplikasi web bisnis atau penjualan (*e-commerce*) dan sebagainya (Riskiono & Pasha, 2020). Terdapat beberapa standar yang perlu diketahui untuk suatu website yang memiliki performa tinggi seperti standar web server yang berkinerja tinggi, standar untuk tingkat reliabilitas dan standar untuk *high availability*.

SMA Cendana merupakan salah satu Yayasan Pendidikan Cendana Rumbai (YPCR) yang terletak di wilayah PT. *Chevron Pacific* Indonesia, Rumbai pesisir, Pekanbaru. SMA Cendana Pekanbaru salah satu sekolah swasta yang telah terakreditasi A. Dari hasil observasi penulis ke SMA Cendana Pekanbaru yang telah dilakukan, penulis menyimpulkan bahwa SMA Cendana Pekanbaru memiliki website sekolah yang dikelola oleh Yayasan Pendidikan Cendana Riau atau bisa disingkat dengan YPCR. Website Yayasan Pendidikan Cendana Riau menaungi beberapa website sekolah lainnya seperti SMP Cendana Pekanbaru, SD Cendana Pekanbaru dan termasuk SMA Cendana Pekanbaru. Kelebihan dari website yayasan ini adalah fitur atau konten yang terdapat dalam website tersebut sangat lengkap dan memberikan informasi sesuai dengan profil di setiap masing masing sekolah. Agar tetap dapat mengakses website sekolah, pihak yayasan memberikan perwakilan admin atau user dari masing-masing sekolah untuk dapat mengakses website tersebut. Namun akses yang diberikan tidak dapat dilakukan secara bebas atau keterbatasan dalam mengakses website sekolah. Hal tersebut tentunya membuat pihak sekolah tidak dapat mengeksplor secara luas kegiatan yang terjadi disetiap SMA, salah satunya yaitu SMA Cendana Pekanbaru. SMA Cendana Pekanbaru ingin memiliki website tersendiri yang tujuannya untuk dapat mengeksplor kegiatan sekolah diberbagai bidang baik akademik maupun non akademik tanpa memiliki keterbatasan dalam mengakses website tersebut. Pihak yang dapat mengakses website tersebut nantinya adalah Staff Admin Kesiswaan dan Humas yaitu bapak M. A Wahyudi H, S.Kom., M.Pd, wakil Kepala Sekolah bagian Kesiswaan yaitu bapak Riko Kurniawan Putra, S.Pd., M.Pd dan Pengurus OSIS bagian dokumtasi.

Untuk meningkatkan sistem layanan website sekolah, dibutuhkan desain arsitektur yang dapat mengatasi tingginya akses diwebsite tersebut agar memiliki penskalaan yang otomatis. Sehingga ketika website sekolah menjadi media promosi kepada siswa baru dan siswa baru mengakses, performa pada website sekolah tersebut tidak menjadi *down*. *Cloud computing* adalah model yang memungkinkan untuk dapat mengakses jaringan dari manapun secara cepat, aman serta sesuai dengan permintaan atau kebutuhan terhadap suatu kumpulan sumber daya komputasi yang dirilis dengan upaya manajemen interaksi penyedia layanan (Ramsari & Ginanjar, 2022).

Metode yang sering diterapkan oleh peneliti terdahulu untuk mengatasi permasalahan tersebut adalah *load balancing* dan *auto scaling*. *Load balancing* dan *auto scaling* merupakan layanan *cloud computing* yang dapat membantu permasalahan mengenai tingginya beban traffic pada suatu website yang disebabkan oleh tingginya akses pada website. *Load balancing* berperan untuk secara otomatis mendistribusikan lalu lintas aplikasi atau website masuk di beberapa target. Sedangkan *auto scaling* berperan untuk menjaga ketersediaan suatu aplikasi dan memungkinkan secara otomatis menambah atau menghapus *instance* sesuai kebutuhan yang ditetapkan. Peneliti terdahulu menggunakan layanan *cloud computing* disalah satu provider yaitu *Google Cloud Platform* (GCP). Peneliti terdahulu menjelaskan hasil dari penerapan penggunaan metode *load balancing* dan *auto scaling* menggunakan teknologi *Google Cloud Platform* (GCP) berhasil diimplementasikan dan metode *high availability* yang diterapkan berupa *autohealing*, *multiple zones*, *load balancing*, *automatic updating* serta *failover* telah berhasil diimplementasikan sehingga dapat menjaga realibilitas dan *availability* infrastruktur server yang telah dibangun (Ramsari & Ginanjar, 2022).

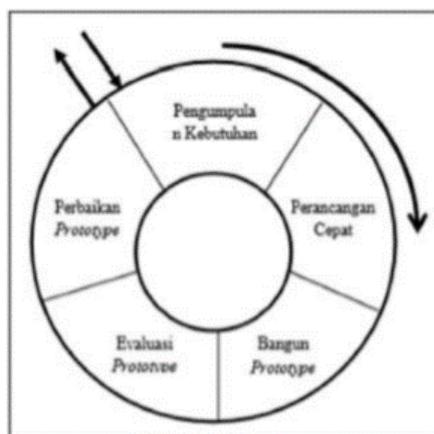
Untuk mengetahui apakah suatu website tersebut telah menerapkan *high availability zone*, memiliki *scalable* yang otomatis serta tingkat reabilitas tinggi penulis akan melakukan pengujian dengan menggunakan *software* tambahan yaitu *Apache Jmeter*. Dari hasil pengujian menggunakan *apache jmeter* terhadap website yayasan SMA Cendana Pekanbaru dengan mengirimkan 500 user dalam waktu 120 detik, website yayasan SMA Cendana Pekanbaru mampu menangani dengan *error rate* 0% dengan *throughput* 4,2 kb/s. Namun ketika user ditambah menjadi 1000 dalam jumlah waktu yang sama yaitu 120 detik, website yayasan SMA Cendana Pekanbaru mengalami *error rate* sebesar 1.90% dengan

*throughput* 8,3 kb/s. Hal tersebut menjelaskan bahwa website yayasan SMA Cendana Pekanbaru tidak mampu menangani penambahan jumlah user yang berlebih dalam jumlah waktu yang sama.

Berdasarkan uraian yang telah dijelaskan penulis membuat penelitian mengenai *Desain Skalabel Website Menggunakan Elastic Load Balancing Pada Amazon Virtual Private Cloud (VPC)*. Penulis akan menggunakan beberapa metode dan layanan yang telah dilakukan oleh peneliti sebelumnya, namun penulis menggunakan teknologi *cloud computing* yang berbeda, yaitu provider *cloud* yang juga mempunyai layanan yang dapat diterapkan pada studi kasus penulis adalah *Amazon Web Service (AWS)*.

## METODE

Metode yang digunakan pada penelitian ini yaitu metode *prototyping* (Ramsari & Ginanjar, 2022). Berikut ini adalah tahapan-tahapan dalam metode *prototyping*:



**Gambar 1.** Metode Prototyping (Ramsari, N., & Ginanjar, A. 2022)

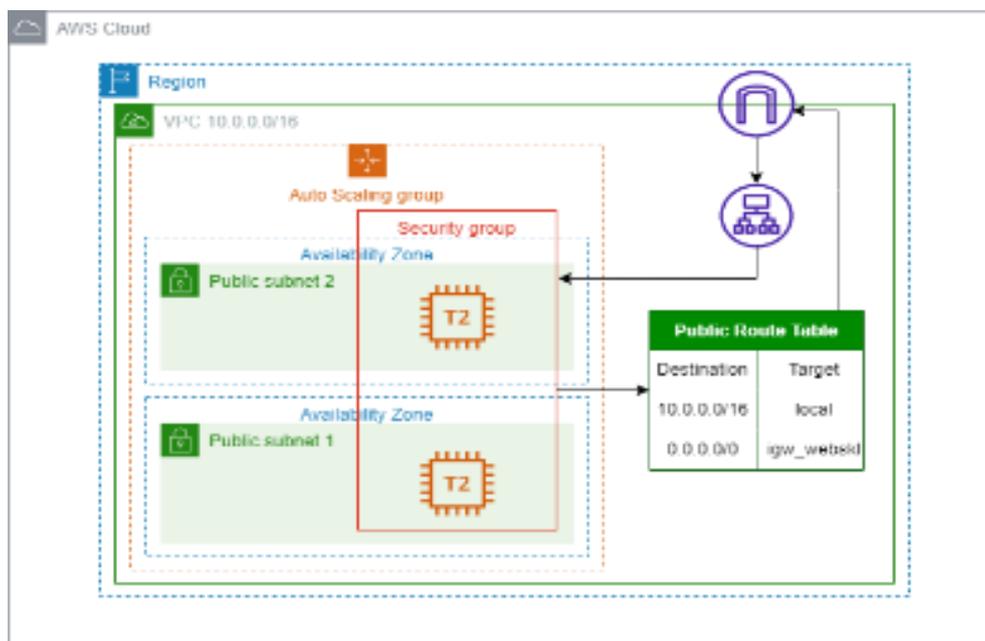
a. Pengumpulan Kebutuhan

Pengumpulan kebutuhan ini dilakukan untuk proses pengumpulan data mengenai layanan-layanan *cloud computing* dari *Amazon Web Service (AWS)* yang akan digunakan dalam perancangan skalabel website

b. Perancangan Cepat

Setelah mendapatkan layanan-layanan *cloud computing* yang digunakan, tahap selanjutnya adalah membangun desain arsitektur skalabel website. *Prototype* yang

dibuat berupa desain arsitektur skalabel pada website. Berikut adalah desain arsitektur skalabel yang telah dirancang:



**Gambar 2.** Desain Arsitektur Skalabel Website

c. Bangun *Prototype*

*Prototype* yang telah dirancang pada tahap sebelumnya menjadi acuan untuk membangun *prototype* pada web server agar menjadi website yang *high availability*

d. Evaluasi *Prototype*

Tahap evaluasi dilakukan dengan cara pengujian, pengujian dilakukan untuk mengetahui apakah *prototype* yang telah dibuat sudah sesuai dengan yang diharapkan. Dalam proses bangun *prototype* terdapat beberapa keterbatasan dalam layanan *cloud computing* yang digunakan, maka dari itu penulis membuat perbaikan pada perancangan cepat dengan memperbaiki kembali desain arsitektur yang telah dibuat.

e. Perbaiki *Prototype*

Pada tahap ini merupakan tahap perbaikan terhadap *prototype* dari hasil evaluasi, jika terdapat evaluasi.



1. User, sebagai individu yang mengakses atau pengunjung yang mengakses dan berinteraksi dengan situs website
2. Internet *gateway*, merupakan komponen dari *Virtual Private Cloud* (VPC) yang menghubungkan VPC dengan internet public
3. Route table, merupakan sebuah route dalam jaringan komputer untuk menentukan rute terbaik yang harus diambil oleh paket data dalam perjalanan dari satu jaringan ke jaringan lainnya
4. *Application load balancer*, yang bertujuan untuk mendistribusikan beban kerja (*workload*) secara merata diantara beberapa sumber daya komputasi server jaringan serta sumber daya *cloud* yang berguna untuk meningkatkan kinerja serta kehandalan sistem
5. Terdapat 3 *instance* dalam dua *availability zone* yang berbeda., sehingga total *instance* secara keseluruhan menjadi 6 *instance*
6. Amazon *Relational Database Service* (RDS), sebagai database online

### **Bangun *Prototype***

Dalam penelitian ini, penulis menerapkan desain arsitektur yang telah dipersiapkan dengan menggunakan layanan-layanan *cloud computing* di Amazon Web Services (AWS). Terdapat dua bagian dalam membangun *prototype*, yaitu *cloud service* dan *deployment*.

#### **1. *Cloud Service***

*Cloud service* adalah berbagai jenis layanan yang disediakan melalui infrastruktur *cloud computing* yang memungkinkan suatu organisasi atau individu untuk menyewa sumber daya komputasi pada suatu platform *cloud computing* seperti penyimpanan data, server, jaringan dan lain sebagainya. Dalam penelitian ini layanan *cloud service* yang digunakan yaitu *Infrastructure As a Service* (IaaS).

##### **a. *Virtual Private Cloud* (VPC)**

Tahap VPC merupakan langkah dalam mengatur lingkungan jaringan untuk membangun infrastruktur yang telah direncanakan. Terdapat beberapa poin penting yang perlu diperhatikan dalam pembangunan VPC seperti pembuatan subnet, pengaturan internet *gateway*, *security group* serta pembuatan *route tables*.

b. *Amazon Relational Database Service (RDS)*

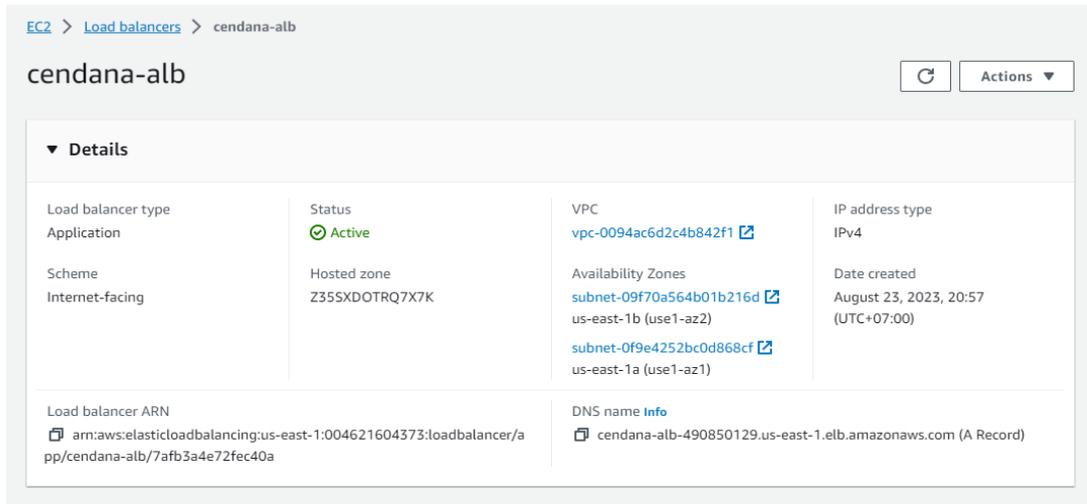
*Amazon Relational Database Service (RDS)* merupakan salah satu layanan untuk dengan mudah membuat, mengelola serta mengoperasikan database *relational* dalam lingkungan *cloud*. Tujuan dalam menerapkan layanan tersebut adalah untuk mensinkronisasikan web server dengan semua *instance* yang dihasilkan oleh *Auto Scaling Group (ASG)*.

c. *Target Groups*

Tujuan membuat target *groups* dengan nama “cendana-tg” yaitu untuk mengelola bagaimana lalu lintas jaringan didistribusikan ke *instance-instance* yang berjalan dibelakang *load balancer*. Dalam konteks ELB, terget group berfungsi sebagai sasaran untuk rute lalu lintas yang diterima oleh *load balancer*.

d. *Application Load Balancing (ALB)*

Tujuan dari pembuatan ALB untuk membantu meningkatkan ketersediaan dan kinerja aplikasi dengan mendistribusikan lalu lintas ke *instance* yang berjalan dibelakangnya. Dengan mendistribusikan lalu lintas secara merata, ALB dapat menghindari *overloading* pada satu *instance* dan membantu menjaga kinerja yang baik. Berikut tampilan dari ALB:



**Gambar 4.** Tampilan Application Load Balancing (ALB)

e. *Amazon Elastic Compute Cloud (EC2)*

*Elastic Compute Cloud (EC2)* adalah layanan *cloud computing* yang memungkinkan untuk menyewa mesin virtual yang disebut "*instance*" dalam

lingkungan *cloud* AWS. *Instance* tersebut dapat diatur sesuai dengan kebutuhan yang dibutuhkan, termasuk jenis *instance*, sistem operasi, kapasitas CPU, RAM serta penyimpanan. Tujuan pembuatan *instance* EC2 adalah untuk menginstal dependensi yang diperlukan dalam menjalankan aplikasi, seperti MYSQL, PHP, dan *apache*. Kemudian *instance* akan dijadikan *image* melalui *snapshot*.

f. *Snapshots*

*Snapshot* adalah salah satu layanan *cloud* pada AWS merupakan salinan berkelanjutan dari suatu *volume* atau *disk* pada *instance* virtual (seperti EC2). *Snapshot* dapat merekam data pada suatu *volume* disuatu titik tertentu. Namun *snapshot* bukanlah salinan langsung dari file individu, melainkan gambar yang mempresentasikan keseluruhan *volume* atau *disk* pada saat tertentu.

g. *Amazon Machine Images (AMIs)*

*Amazon Machine Images (AMIs)* adalah gambar virtual lengkap yang berisikan informasi mengenai sistem operasi, aplikasi, konfigurasi data serta data yang digunakan untuk meluncurkan *instance* virtual dilayanan *cloud* AWS, seperti *Amazon EC2 (Elastic Compute Cloud)*. AMIs berfungsi sebagai *template* atau cetakan dari suatu *instance*. Dalam pembuatannya, AMIs diciptakan representasi yang dapat digunakan untuk membuat *instance* baru dengan konfigurasi yang telah ditentukan.

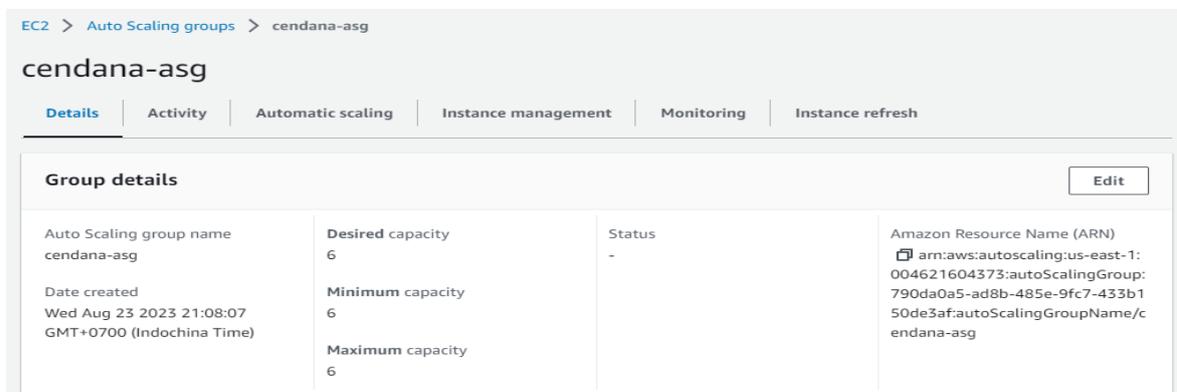
h. *Launch Template*

*Launch template* menyediakan cara untuk dapat mendefinisikan detail seperti jenis *instance*, *volume* yang terpasang, konfigurasi jaringan, pengaturan keamanan dan yang lainnya. *Launch template* mampu memastikan bahwa setiap *instance* diluncurkan, konfigurasi yang ditentukan dalam *template* akan diterapkan, sehingga menciptakan konsistensi dalam lingkungan pengembangan. Selain itu, dengan menggabungkan *launch template* dengan *Amazon Machine Images (AMIs)* dapat menghindari konfigurasi manual berulang saat meluncurkan *instance* baru secara *otomasi*.

i. *Auto Scaling Group (ASG)*

*Auto Scaling Group (ASG)* memungkinkan untuk secara dinamis menyesuaikan jumlah *instance* yang berjalan sesuai dengan permintaan lalu lintas atau

kebutuhan komputasi. ASG yang dibuat akan diintegrasikan dengan layanan ALB yang telah dibuat sebelumnya, yang mampu mendistribusikan lalu lintas dengan merata diantara *instance* yang berjalan. Berikut tampilan ASG yang telah dibuat:



**Gambar 5.** Tampilan Auto Scaling Group (ASG)

## 2. Deployment

*Deployment* dalam konteks *cloud computing* adalah proses penerapan dan pengaturan aplikasi, layanan atau solusi perangkat lunak yang melibatkan pengunggahan dan konfigurasi komponen perangkat lunak, sumber daya serta layanan yang diperlukan untuk menjalankan aplikasi atau layanan yang diinginkan diplatform *cloud*. Tujuan dari *deployment* dalam konteks *cloud computing* untuk berhasil dan efisien mengimplementasikan aplikasi, layanan, atau solusi perangkat lunak lainnya dilingkungan *cloud*. Dalam penelitian ini *deploy* dilakukan pada sistem yang menjalankan distribusi *Linux* yang menggunakan manajer paket APT (*Advanced Package Tool*), seperti Ubuntu atau Debian.

### a. Install Package

*Install package* merupakan tahap yang merujuk pada proses instalasi perangkat lunak atau komponen tambahan tertentu kedalam sistem komputer. Berikut adalah tahap yang perlu dilakukan untuk proses *package*:

1. Melakukan update pada sistem operasi berbasis debian
2. Melakukan *install* perangkat lunak server web
3. Melakukan *install* mysql-client-core-8.0
4. Melakukan *install* paket “mysql-server”

5. *Install* pengembangan web berbasis PHP pada sistem

Menggunakan perintah “`sudo apt install php libapache2-mod-php php-mysql-y`” untuk menginstall beberapa komponen yang terkait dengan pengembangan web berbasis PHP pada sistem. **Instalasi PHP**, dengan menginstall paket PHP, agar server mampu menjalankan kode PHP, bahasa pemrograman yang umum digunakan dalam pengembangan web dan bagian integral dari banyak situs web dinamis. **Instalasi libapache2-mod-php**, adalah modul untuk server web Apache yang memungkinkan Apache untuk menjalankan kode PHP. Dengan menginstallnya, dapat dipastikan bahwa server Apache dapat memproses dan menghasilkan halaman web yang mengandung skrip PHP. **Instalasi php-mysql**, menyediakan dukungan untuk berinteraksi dengan basis data MySQL dari PHP. Penting dilakukan untuk mengembangkan aplikasi web yang menggunakan PHP dan berhubungan dengan server MySQL.

6. Melakukan *restart* layanan web server *apache*

b. *Project File Configuration*

Untuk melakukan konfigurasi file, dilakukan dengan cara sebagai berikut:

1. Melakukan perintah "`sudo nano /etc/apache2/mods-enabled/dir.conf`"
2. Mengunduh repositori Git pada GitHub
3. Menyalin repositori dari GitHub ke direktori web root pada server
4. Melakukan pemindahan ke direktori “`/var/www/html`”

c. *Checking Word On File Directory*

Tahap yang dilakukan untuk memeriksa atau mencari kata dalam file direktori.

Berikut langkah-langkah yang dilakukan:

1. Menggunakan perintah “`grep -Ril "localhost" *`”
2. Mencari dan menganalisis endpoint RDS pada lingkungan *development*
3. `grep -Ril "cendana-alb-1657713023.us-west-2.elb.amazonaws.com" *`

d. *File Replacement With Appropriate Endpoint*

Merupakan proses mengganti file dengan endpoint yang sesuai atau yang dibutuhkan. Berikut langkah-langkahnya:

1. Mengganti "`localhost`" dengan "`endpoint-rds-production`"

---

2. `sudo find * -type f -exec sed -i 's/cendana-db.cesbflvkxtyc.us-west-2.rds.amazonaws.com/(endpoint-rds-production)/g' {} +`

3. Mengganti teks "`cendana-alb-1657713023.us-west-2.elb.amazonaws.com`" dengan "`dns-alb-production`"

e. *Database Configuratin On Local*

Tahap selanjutnya yaitu melakukan konfigurasi database lokal, dengan cara sebagai berikut:

1. `mysql -h (endpoint-rds-production) -P 3306 -u admin -p --password=admin123`

2. Membuat database

3. `use wp;`

4. `exit;`

5. `cd database` Perintah "`cd database`" digunakan untuk berpindah dari direktori kerja saat ini ke direktori bernama "`database`".

6. `sudo sed -i 's/localhost/(dns-alb-production)/g' wp.sql`. Perintah ini berguna untuk mengubah nilai atau konfigurasi tertentu dalam file teks, seperti skrip SQL, dengan menggantikan string tertentu dengan yang lain. Dalam kasus ini, digunakan untuk mengganti semua kemunculan teks "`localhost`" dengan "`dns-alb-production`" dalam file "`wp.sql`" menggunakan utilitas "`sed`" dengan akses superuser.

7. `sudo sed -i 's/cendana-alb-1657713023.us-west-2.elb.amazonaws.com/(dns-alb-production)/g' wp.sql`. Perintah ini akan mengubah "`dns-alb-development`" dalam file "`wp.sql`" dengan "`dns-alb-production`", sehingga mengganti informasi khusus dalam file tersebut. Ini sering digunakan untuk mengganti nilai yang sudah ada dalam file dengan nilai yang lebih sesuai atau aktual, seperti mengganti DNS atau alamat yang sudah usang dengan alamat yang baru.

f. *Database Configuration With RDS*

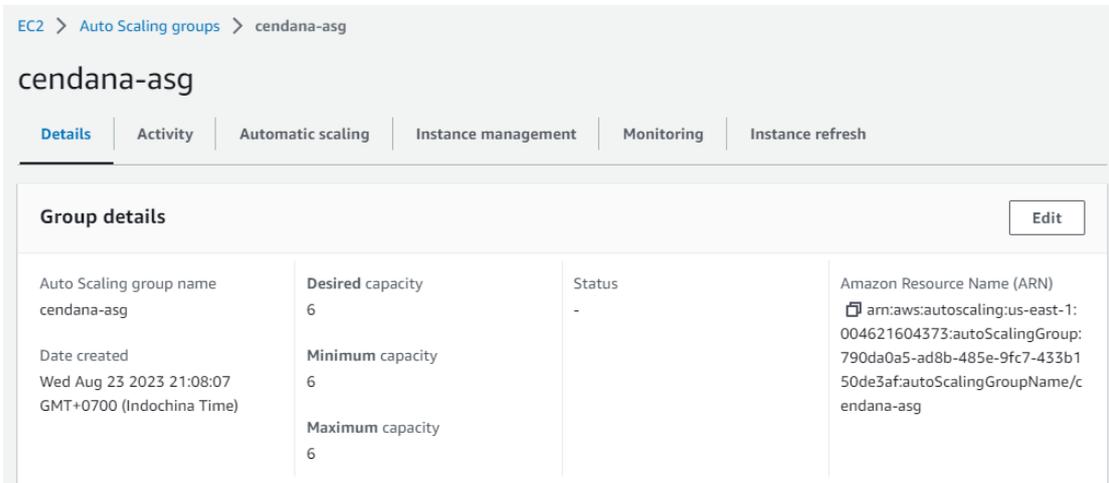
Melakukan konfigurasi database dengan *Relational Database Service* (RDS) agar database dapat terhubung ke internet. Berikut langkah-langkah konfigurasi database:

1. `mysql -h (endpoint-rds) -P 3306 -u admin -p --password=admin123 wp < wp.sql`. Perintah ini akan menghubungkan ke server database MySQL di *(endpoint-rds)* dengan informasi kredensial yang diberikan, dan mengimpor data dari file SQL "*wp.sql*" ke basis data "*wp*". Perintah ini berguna ketika ingin mengambil data dari file SQL dan mengintegrasikannya ke dalam basis data yang ada di server MySQL.
2. `exit`;

## Pengujian

Dalam penelitian ini pengujian kinerja suatu sistem web sever berdasarkan parameter metrik pengujian *throughput* dan *error rate*. Pengukuran atau pengujian dapat dilakukan dengan pemuatan permintaan yang disimulasikan menggunakan aplikasi *Apache Jmeter* (Hadiwandura & Candra, 2021). Berikut skenario yang dilakukan:

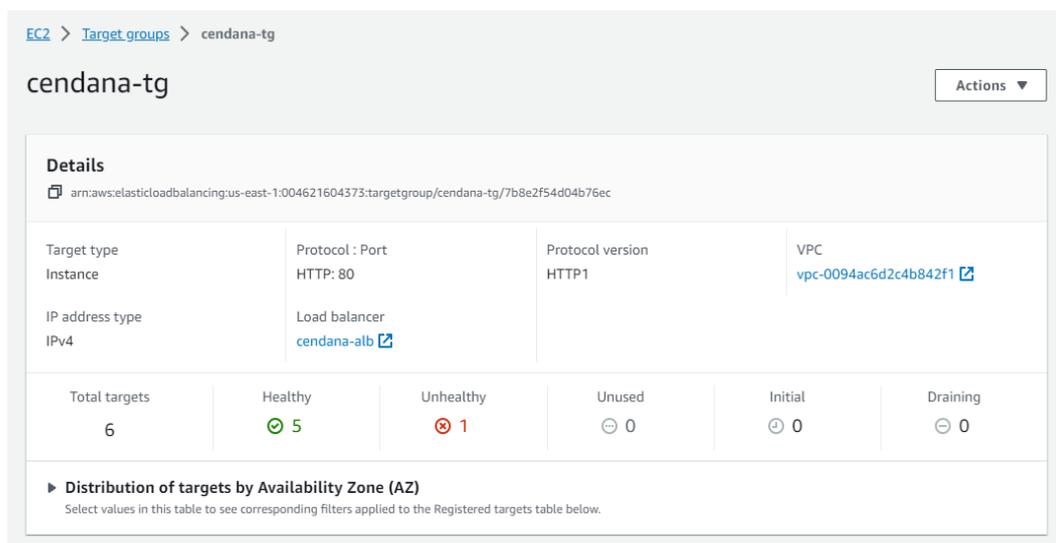
- a. Untuk skenario pertama Penulis menyiapkan 6 *instance* untuk menjaga kestabilan server pada saat dijalankan. Pada kasus ini penulis menetapkan *desired capacity*, *minimum capacity* dan *maximum capacity* dengan jumlah yang sama yaitu 6 *instance* untuk dapat mempermudah pengelolaan dan pemeliharaan sistem. Konsistensi ini memudahkan dalam pemantauan dan manajemen *instance* di lingkungan AWS.



**Gambar 6.** Skalabel pada *Auto Scaling Group*

- b. Pada skenario kedua, pemantauan dari *console* AWS pada *target groups* menghasilkan bahwa 6 server yang telah dibuat masih terjaga ketersediaannya saat dilakukan

pengujian meskipun ada satu server yang mengalami *downtime*. Namun 5 server yang lainnya masih dalam keadaan *uptime*.



**Gambar 7.** Tampilan *Console Target Groups*

c. Skenario pengujian terakhir dilakukan dengan memberikan traffic dalam jumlah tertentu dalam waktu yang telah ditentukan kepada website yang telah dikonfigurasi dengan *Application Load Balancer (ALB)* dan *Auto Scaling Group (ASG)*. Berikut adalah proses pengujian yang akan dilakukan dalam penelitian ini:

1. *Install Java*
2. *Install Apache Jmeter*

Untuk melakukan *install apache jmeter*, unduh *apache jmeter* pada situs website [https://jmeter.apache.org/download\\_jmeter.cgi](https://jmeter.apache.org/download_jmeter.cgi). Apabila telah mengunduh file *apache jmeter* maka selanjutnya ekstrak file dan cari file *ApacheJmeter.jar* kemudian klik. Maka akan muncul tampilan awal pada *apache jmeter*

3. Melakukan Persiapan Pengujian

Pengujian kali ini menggunakan pengujian *loadtesting*. Berikut tahapan persiapan pengujian:

- a. Membuat "*Test Plan*" baru, kemudian dalam test plan tersebut tambahkan "*Thread Group*". *Thread group* mengatur berapa jumlah user yang akan mengakses, dalam waktu yang ditentukan. Pada pengujian kali ini, penulis membuat 1000 user dalam waktu 120 detik (2 menit).

- b. Selanjutnya dalam “*Thread Group*” tambahkan “*Sampler*” kemudian pilih “*HTTP Request*”. Pada *HTTP Request* masukan link website yang akan diuji, penulis memasukan link <http://cendana-alb-490850129.us-east-1.elb.amazonaws.com/>
  - c. Tambahkan “*View Results In Table*”, “*Graph Result in Table*” dan “*Summary Report*” untuk melihat hasil yang diperoleh ketika melakukan proses pengujian
4. Selanjutnya klik *start* untuk memulai pengujian, berikut adalah hasil pengujian yang dilakukan :

**Tabel 1.** Hasil Pengujian HTTP Request 1000 User

1000 User (Label)	Error %	Throughput	Sent KB/sec
<i>HTTP Request</i>	0%	8.3	1.22

Dalam Percobaan 1000 user dalam waktu 2 menit menunjukkan status “*successfull*” yang berarti server mampu menangani 1000 user dalam waktu 2 menit.

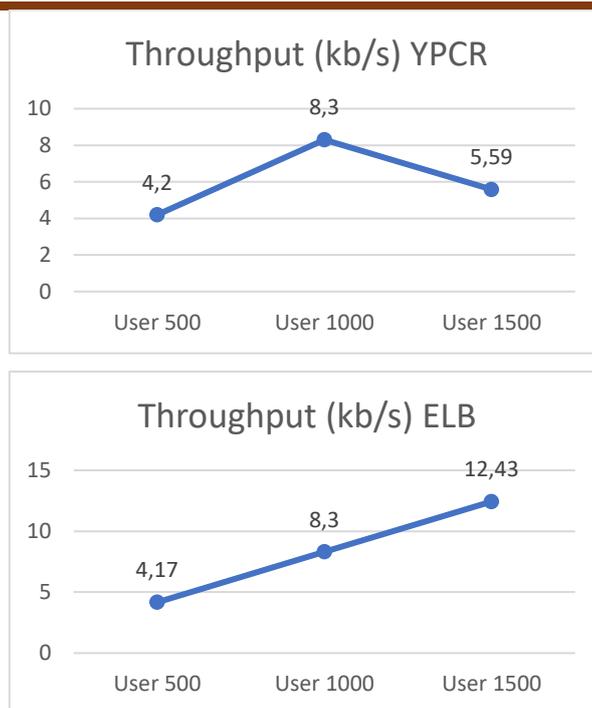
5. Selanjutnya yaitu perbandingan uji coba <https://smacendana-pekanbaru.ypcrtau.or.id/> (SMA Cendana YPCR) dengan <http://cendana-alb-490850129.us-east-1.elb.amazonaws.com/> (Cendana ALB) dalam waktu 120 detik (2 menit)

**Tabel 2.** Hasil pengujian SMA Cendana YPCR

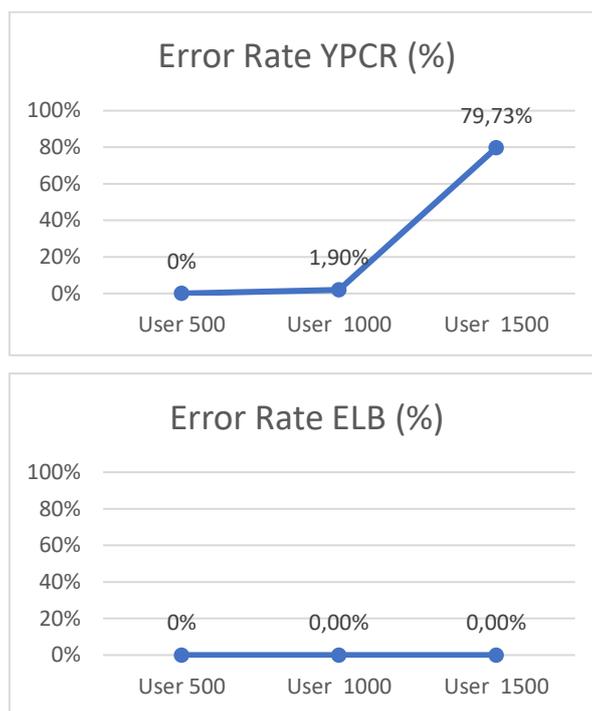
<i>HTTP Request</i>	Error %	Throughput	Sent KB/sec
500 user	0%	4,2	1,1 KB/sec
1000 user	1,90%	8,3	2,19 KB/sec
1500 user	79,73%	5,6	0,37 KB/sec

**Tabel 3.** Hasil pengujian Cendana ALB

<i>HTTP Request</i>	Error %	Throughput	Sent KB/sec
500 user	0%	4,1	1,1 KB/sec
1000 user	0%	8,3	1,22 KB/sec
1500 user	0%	12,4	1,83 KB/sec



**Gambar 8.** Grafik Perbandingan Throughput YPCR dengan ELB



**Gambar 9.** Grafik Perbandingan Error Rate YPCR dan ELB

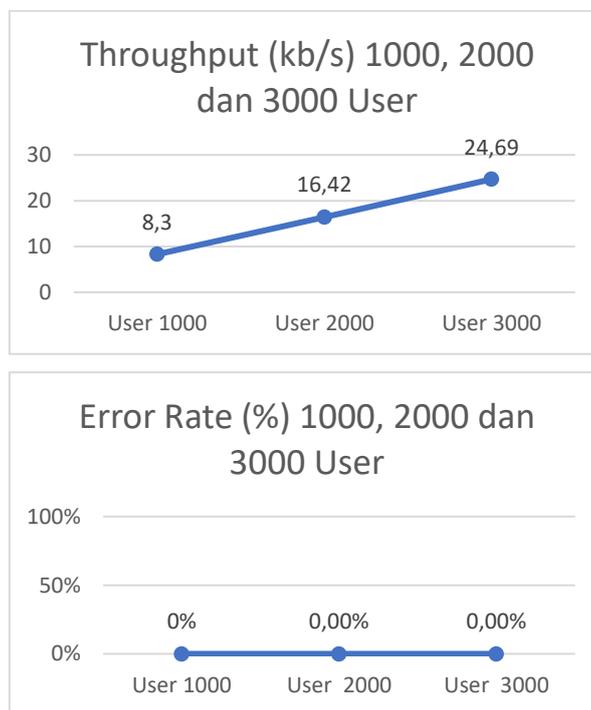
Hasil dari perbandingan pengujian pada kedua website menghasilkan bahwa website <http://cendana-alb-490850129.us-east-1.elb.amazonaws.com/> yang telah dibuat

mampu menembus 1500 user dalam waktu 2 menit dengan *error rate* 0%. Sedangkan untuk website <https://smacendana-pekanbaru.ypcriau.or.id/> hanya mampu 500 user dalam 2 menit dengan *error rate* 0%, dalam pengujian 1000 user website tersebut telah mengalami *error rate* sebesar 1,90%. Hal tersebut menandakan bahwa website sekolah yang telah dibuat oleh penulis mampu mengatasi 1500 user dalam waktu 2 menit sedangkan website yayaan tidak.

- Selanjutnya pengujian untuk menguji server dengan user dalam jumlah 1000, 2000 dan 3000 dalam waktu 2 menit. Apakah server mampu menangani atau tidak. Berikut ini adalah hasil uji coba *loadtesting* pada 1000 user, 2000 user dan 3000 user dalam waktu yang sama yaitu selama 2 menit:

**Tabel 4.** Penjelasan Hasil Pengujian 1000 user, 2000 user dan 3000 user

User	Error %	Throughput	Sent KB/sec	waktu
1000 user	0%	8,3	1,22 KB/sec	2 menit
2000 user	0%	16,4	2,42 KB/sec	2 menit 2 detik
3000 user	0%	24,7	3,64 KB/sec	2 menit 1 detik



**Gambar 10.** Grafik Perbandingan Throughput dan Error Rate 1000, 2000 dan 3000 User

Dalam percobaan antara 1000 user, 2000 user dan 3000 user mampu menghasilkan *error rate* 0% dengan waktu yang berbeda. Dalam waktu 120 detik, hanya 1000 user yang dapat selesai tepat waktu dengan *error rate* 0%. Namun untuk 2000 dan 3000 user dapat menyelesaikan dalam waktu yang berbeda dengan *error rate* 0%. Hal tersebut membuktikan bahwa server mampu menangani beban user yang ditambah, meskipun dalam jumlah waktu yang berbeda.

## KESIMPULAN DAN REKOMENDASI

Setelah melakukan pengujian pada penelitian yang berjudul *Desain Skalabel Website Menggunakan Elastic Load Balancing Pada Amazon Virtual Private Cloud (VPC)*, Maka dapat ditarik kesimpulan yaitu perancangan desain arsitektur skalabel yang telah dirancang, dapat diimplementasikan dengan baik pada layanan *cloud computing* amazon web service. Meskipun terbatas atau *free tier*, layanan *cloud computing* yang digunakan masih dapat terpasang dan berjalan sesuai yang diharapkan. Implementasi yang dihasilkan seperti *instance* yang digunakan berjumlah 3 dalam dua *availability zone* yang berbeda, Jadi secara keseluruhan berjumlah 6 *instance* yang dapat berjalan dengan baik.

## REFERENSI

- Amazon EC2 Auto Scaling. aws.amazon.com.  
<https://aws.amazon.com/id/ec2/autoscaling/#:~:text=Amazon%20EC2%20Auto%20Scaling%20membantu,status%20dan%20ketersediaan%20armada%20Anda>
- Amazon Virtual Private Cloud (Amazon VPC). aws.amazon.com.  
<https://aws.amazon.com/id/vpc/>
- Duo, Z, & Dhablania, A (2023). Elastic security services load balancing in a wireless mesh network. US Patent App. 18/234,152, Google Patents,  
<https://patents.google.com/patent/US20230403558A1/en>
- Elastic Load Balancing. aws.amazon.com.  
[https://docs.aws.amazon.com/id\\_id/elasticloadbalancing/latest/userguide/what-is-load-balancing.html](https://docs.aws.amazon.com/id_id/elasticloadbalancing/latest/userguide/what-is-load-balancing.html)
- Hadiwandura, T. Y., & Candra, F. (2021). High Availability Server Using Raspberry Pi 4

- Cluster and Docker Swarm. *IT Journal Research and Development*, 6(1), 43–51.  
[https://doi.org/10.25299/itjrd.2021.vol6\(1\).5806](https://doi.org/10.25299/itjrd.2021.vol6(1).5806)
- Key Server Performance Metrics For Actionable Monitoring. *appdynamics.com*.  
<https://www.appdynamics.com/product/server-visibility/server-performance-metrics>
- Pratama, R., Lubis, A., & Wahyuni, S. (2022). Rancang Bangun Sistem Load Balancer Dengan Layanan Cloud Amazone Web Services. *INTECOMS: Journal of Information Technology and Computer Science*, 5(2), 156-160
- Putra, G. M. (2022). Pengujian Kinerja Web Server Atas Penyedia Layanan Elastic Cloud Compute (EC2) Pada Amazon Web Services (AWS). *Adopsi Teknologi dan Sistem Informasi (ATASI)*, 1(1), 21-35
- Rahmatulloh, A., & MSN, F. (2017). Implementasi Load Balancing Web Server menggunakan Haproxy dan Sinkronisasi File pada Sistem Informasi Akademik Universitas Siliwangi. *Jurnal Nasional Teknologi Dan Sistem Informasi*, 3(2), 241–248. <https://doi.org/10.25077/teknosi.v3i2.2017.241-248>
- Ramsari, N., & Ginanjar, A. (2022). Implementasi Infrastruktur Server Berbasis Cloud Computing Untuk Web Service Berbasis Teknologi Google Cloud Platform. <https://doi.org/10.28989/senatik.v7i1.472>
- Riskiono, S. D., & Pasha, D. (2020). Analisis Metode Load Balancing Dalam Meningkatkan Kinerja Website E-Learning. *Jurnal Teknoinfo*, 14(1), 22. <https://doi.org/10.33365/jti.v14i1.466>
- Somasundaram, A (2021). DDOS Mitigation In Cloud Computing Environment By Dynamic Resource Scaling With Elastic Load Balancing. *Turkish Journal of Computer and Mathematics ...*, *turcomat.org*, <https://www.turcomat.org/index.php/turkbilmal/article/view/6380>
- Subhi, R., Ruslianto, I., & Ristian, U. Implementasi Teknik Scaling Pada Sistem Manajemen Balancing Server Berbasis Website. *Coding Jurnal Komputer dan Aplikasi*, 9(02), 316-326
- Sugiyanta, L., & Nurahma, W. A. (2017). Analisis Perbandingan Antara Colocation Server Dengan Amazon Web Services (Cloud) Untuk Usabilitas Portal Swa.co.id Di PT. Swa Media Bisnis). *PINTER : Jurnal Pendidikan Teknik Informatika Dan Komputer*, <http://journal.thamrin.ac.id/index.php/jtik/article/view/1810/pdf>

1(1), 58–63. <https://doi.org/10.21009/pinter.1.1.8>

Ting-ting, WEN, & Hong-zhe, LI (2019). Elastic Load Balancing Algorithm for Cloud Computing Service. Computer and Modernization, cam.org.cn, <https://doi.org/10.3969/j.issn.1006-2475.2019.10.006>