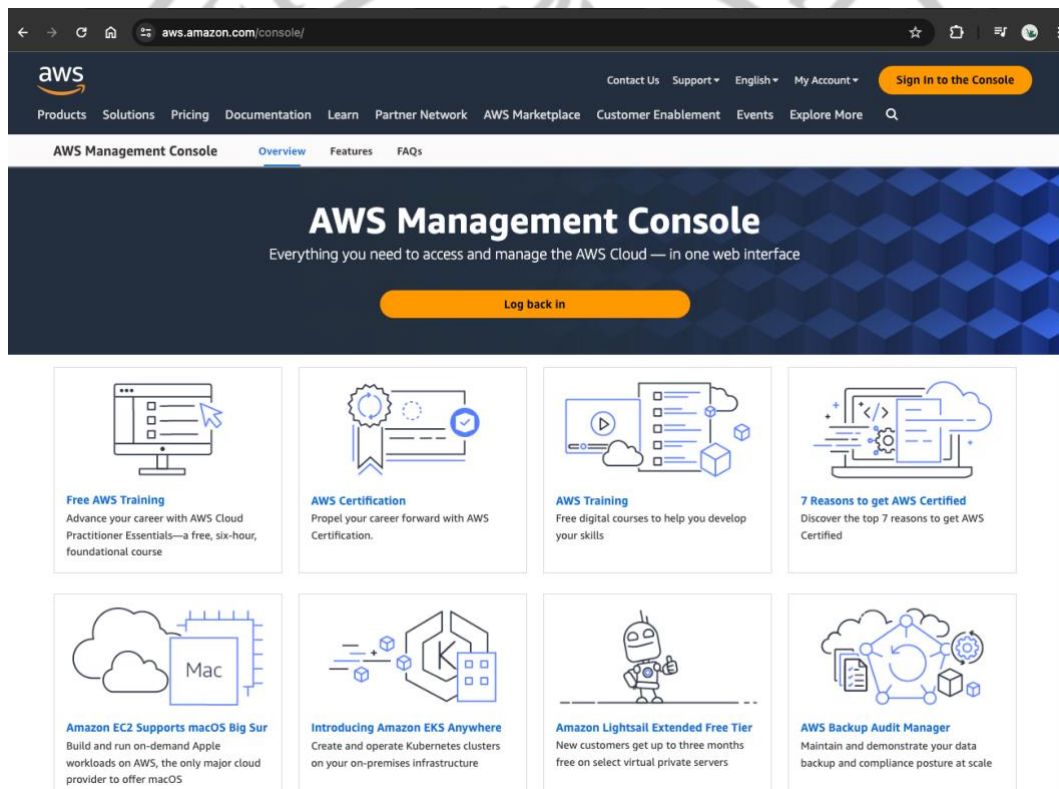


## BAB IV

### IMPLEMENTASI DAN PEMBAHASAN

#### 4.1. Instalasi Dan Konfigurasi AWS

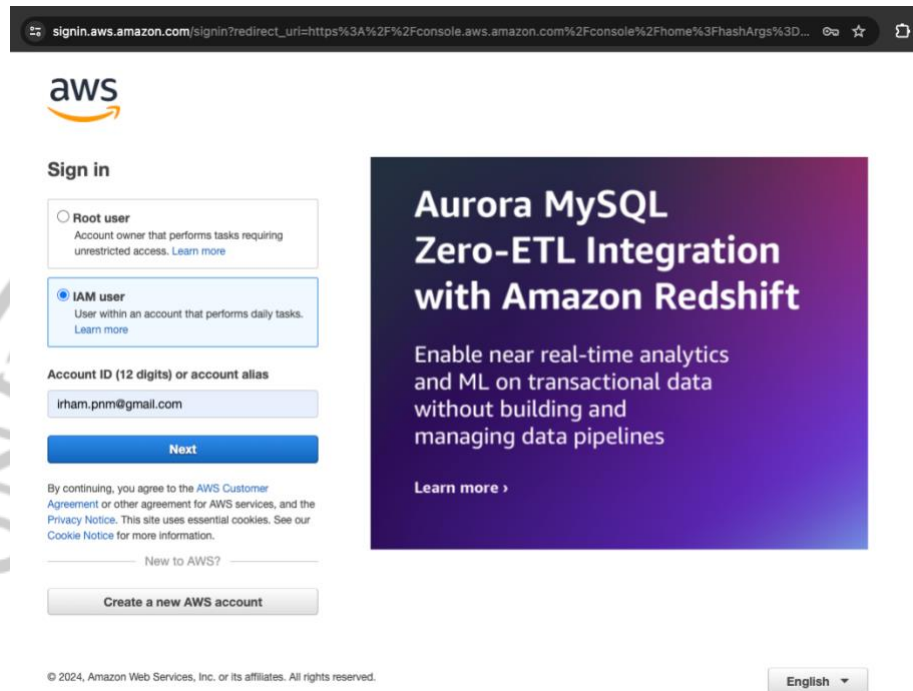
Amazon Web Services (AWS) adalah cloud terlengkap dan paling banyak digunakan di dunia, menawarkan lebih dari 200 layanan berfitur lengkap dari pusat data secara global. Jutaan pelanggan termasuk perusahaan rintisan dengan pertumbuhan tercepat, perusahaan terbesar, dan lembaga pemerintah terkemuka menggunakan AWS untuk menurunkan biaya, menjadi lebih gesit, dan berinovasi lebih cepat ( <https://aws.amazon.com> ).



Gambar 4.1. Menu Utama AWS

#### 4.1.1. Login Ke Konsol AWS

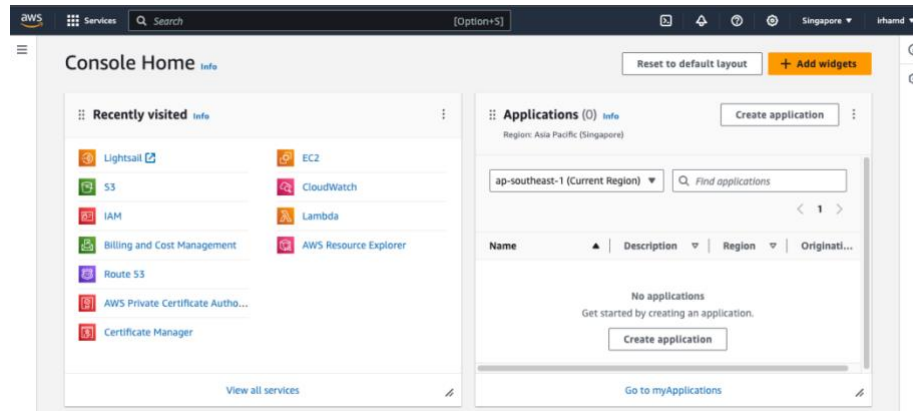
Langkah pertama untuk melakukan instalasi pada platform AWS adalah melakukan login ke konsol halaman AWS dengan menunjung alamat <https://signin.aws.amazon.com/signin> dan memasukkan *email* dan *password* tampak seperti gambar dibawah ini.



Gambar 4.2. Login Ke Dashboard AWS

#### 4.1.2. Tampilan Dashboard Utama AWS

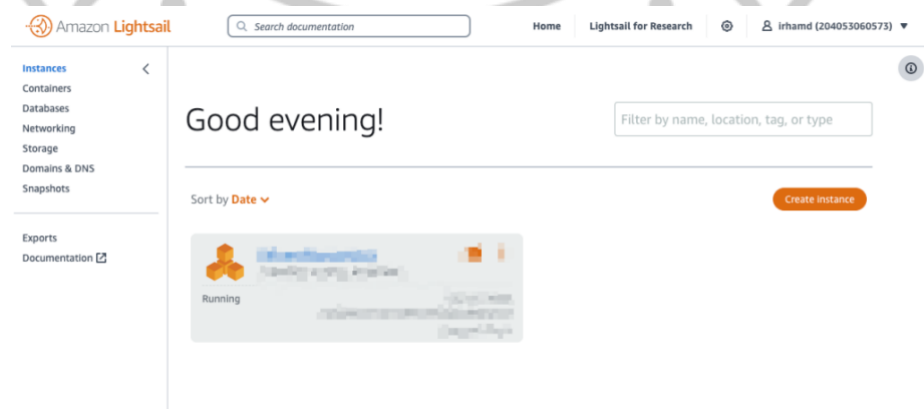
Setelah melakukan login, AWS akan mengarahkan ke halaman Console Home yang menjadi awal untuk mengakses semua layanan dan sumber daya yang tersedia di lingkungan AWS. Di sini, dapat melihat informasi penting tentang lingkungan AWS Anda, mengakses layanan yang digunakan secara reguler, dan menjelajahi berbagai opsi yang tersedia untuk mengelola sumber daya.



Gambar 4.3. Console Home AWS

#### 4.1.3. Instalasi *Lightsail* AWS

Langkah ini adalah tahapan dimana user akan melakukan instalasi lightsail AWS dengan memanfaatkan fitur-fitur yang telah disediakan. Lightsail AWS adalah layanan cloud yang disediakan oleh Amazon Web Services (AWS) yang dirancang untuk menyederhanakan proses pembuatan, manajemen, dan skalabilitas infrastruktur. Lightsail menyediakan antarmuka yang sederhana yang memungkinkan pengguna untuk dengan mudah membuat, mengelola, dan mengonfigurasi instance virtual, database, dan penyimpanan. Pada gambar dibawah ini dapat dimulai dengan memilih tombol Create Instance.



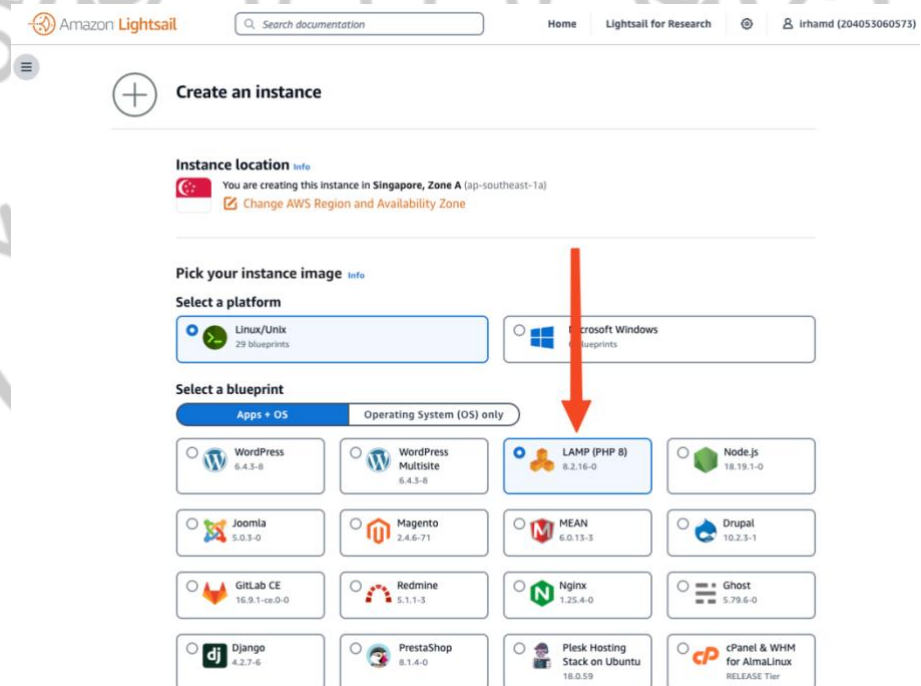
Gambar 4.4. Menu AWS Lightsail

#### 4.1.4. Create Instance Lightsail

Tahap ini adalah proses untuk menentukan platform yang akan digunakan oleh user dan kebutuhan-kebutuhan sistem operasi yang akan dibutuhkan serta kebutuhan server, dengan rincian sebagai berikut:

- Platform / Operation System	:	Linux
- Plugin	:	LAMP (PHP 8)
- Region	:	Singapore / ap-southeast-1
- Storage	:	40 GB
- Bandwidth	:	2 TB

Lebih jelasnya terlihat seperti gambar dibawah ini.



Gambar 4.5. Menentukan *Platform*, Sistem Operasi dan *Plugin*

**Select a size**

Sort by Price per month ▾

<input checked="" type="radio"/> <b>\$5</b> USD per month 512 MB Memory 2 vCPUs Processing 20 GB SSD Storage 1 TB Transfer First 3 months free	<input type="radio"/> <b>\$7</b> USD per month 1 GB Memory 2 vCPUs Processing 40 GB SSD Storage 2 TB Transfer First 3 months free	<input type="radio"/> <b>\$12</b> USD per month 2 GB Memory 2 vCPUs Processing 60 GB SSD Storage 3 TB Transfer First 3 months free	<input type="radio"/> <b>\$24</b> USD per month 4 GB Memory 2 vCPUs Processing 80 GB SSD Storage 4 TB Transfer
<input type="radio"/> <b>\$44</b> USD per month 8 GB Memory 2 vCPUs Processing 160 GB SSD Storage 5 TB Transfer	<input type="radio"/> <b>\$84</b> USD per month 16 GB Memory 4 vCPUs Processing 320 GB SSD Storage 6 TB Transfer	<input type="radio"/> <b>\$164</b> USD per month 32 GB Memory 8 vCPUs Processing 640 GB SSD Storage 7 TB Transfer	

For a limited time, new Lightsail customers can try the selected plan for free for three months. [Learn more about the free trial in Lightsail.](#)

---

**Identify your instance**

Your Lightsail resources must have unique names.

WismaNusantara × 1

**Tagging options**

Use tags to filter and organize your resources in the Lightsail console. Key-value tags can also be used to organize your billing, and to control access to your resources. [Learn more about tagging.](#)

**Key-only tags** [Info](#)

+ Add key-only tags

**Key-value tags** [Info](#)

+ Add key-value tag

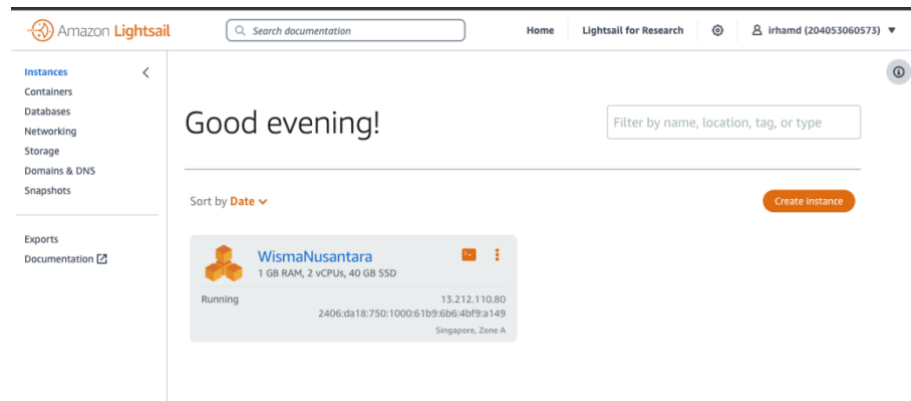
**Create instance**

Your use of AWS services is subject to the [AWS Customer Agreement](#).

Gambar 4.6. Menentukan Kebutuhan *Server*

#### 4.1.5. Hasil Instalasi *Lightsail* AWS

Setelah berhasil membuat instance *Lightsail*, maka akan diberikan informasi rinci tentang instance tersebut, termasuk nama *instance*, *IP* publik, *IP* privat, lokasi dan kebutuhan *server* lainnya, maka hasil instalasi *Lightsail* tersebut siap untuk digunakan sesuai kebutuhan, mulai dari *hosting* situs *web* hingga menjalankan aplikasi atau pengembangan *project* yang akan dibangun. Untuk hasil instalasi tersebut dapat dilihat dari gambar dibawah ini.



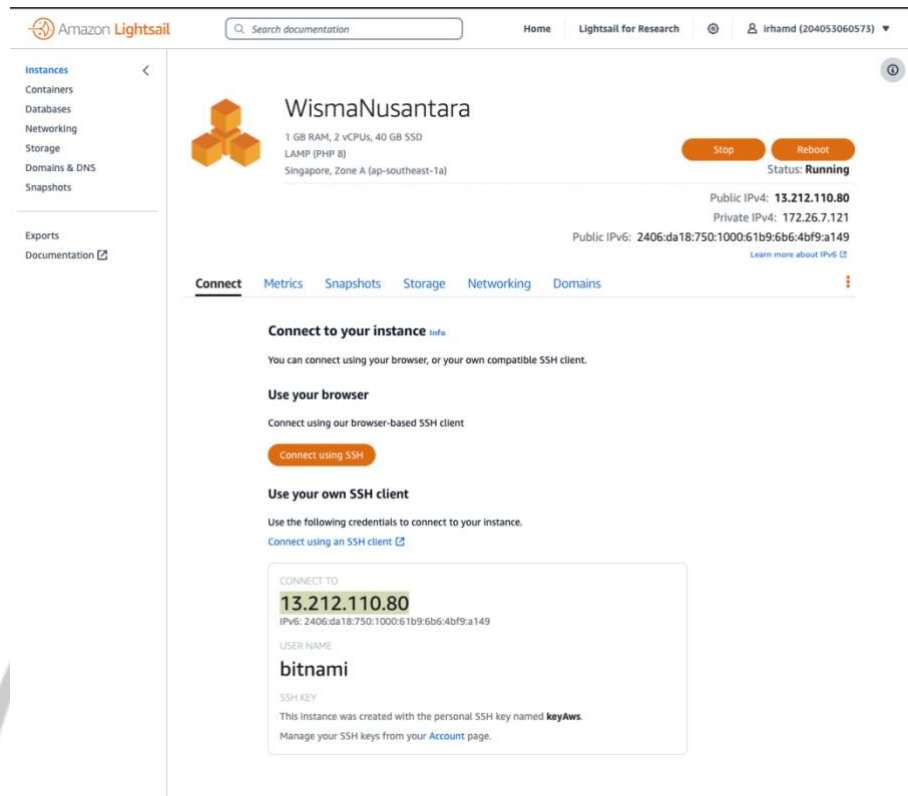
Gambar 4.7. Hasil Instalasi Lightsail AWS

#### 4.1.6. *Dashboard Lightsail Wisma Nusantara*

Berikut ini merupakan uraian tentang klasifikasi pada AWS lightsail yang sudah siap digunakan. Pada halaman ini user diberikan hak akses penuh untuk melakukan aksi seperti mematikan me-restart server atau bahkan me-nonaktifkan server lightsail yang sudah dilakukan instalasi. Dan berikut ini urian singkat terkait dengan server AWS lightsail yang sudah diselesaikan.

Deskripsi	:	WismaNusantara
User	:	bitnami
RAM	:	1 GB
Processor	:	2 CPU
Storage / Penyimpanan	:	40 GB SSD
Sistem Operasi	:	Linux
IP Publik	:	13.212.110.80
IP Lokal	:	172.26.7.121

Table 4.1. Urian Penggunaan *Lightsail*



Gambar 4.8. Uraian *Server Lightsail AWS*

#### 4.1.7. Instalasi *Lightsail AWS* Dengan Domain

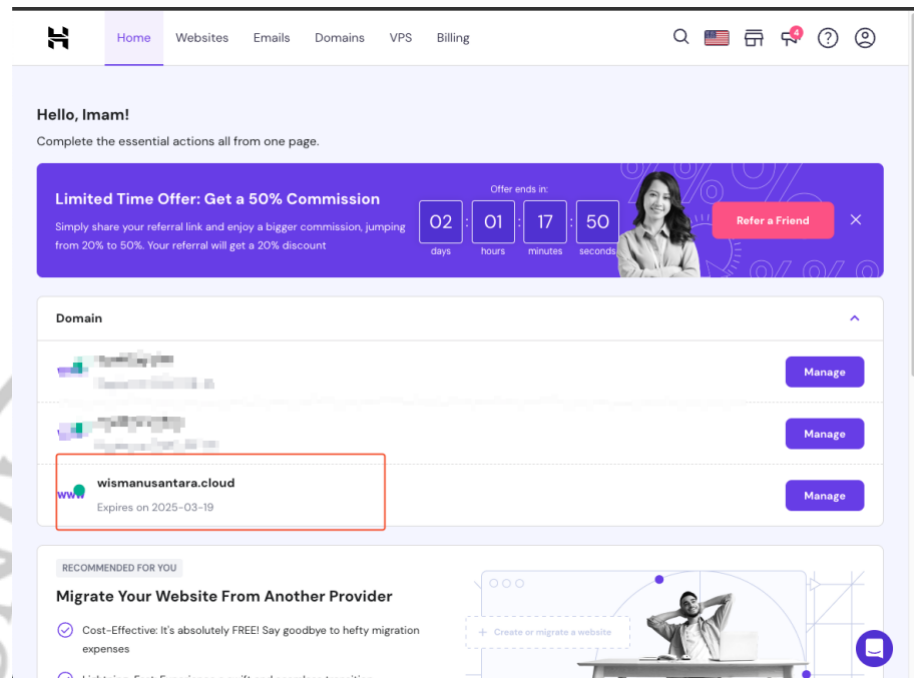
Untuk menghubungkan *Lightsail* di *AWS* dengan domain yang di hosting di *Hostinger*, perlu melakukan beberapa langkah berikut .

##### 4.1.7.1. *Hostinger*

*Hostinger* adalah penyedia layanan web hosting maupun domain serta menawarkan berbagai layanan hosting web, termasuk hosting bersama (*shared hosting*), *VPS (Virtual Private Server)*, *cloud hosting*, dan layanan lainnya.



Langkah pertama adalah dengan mengunjungi dan *Login* ke halaman <https://hpanel.hostinger.com/>, dan pilih domain *wismanusantara.cloud* yang nampak seperti gambar berikut ini.

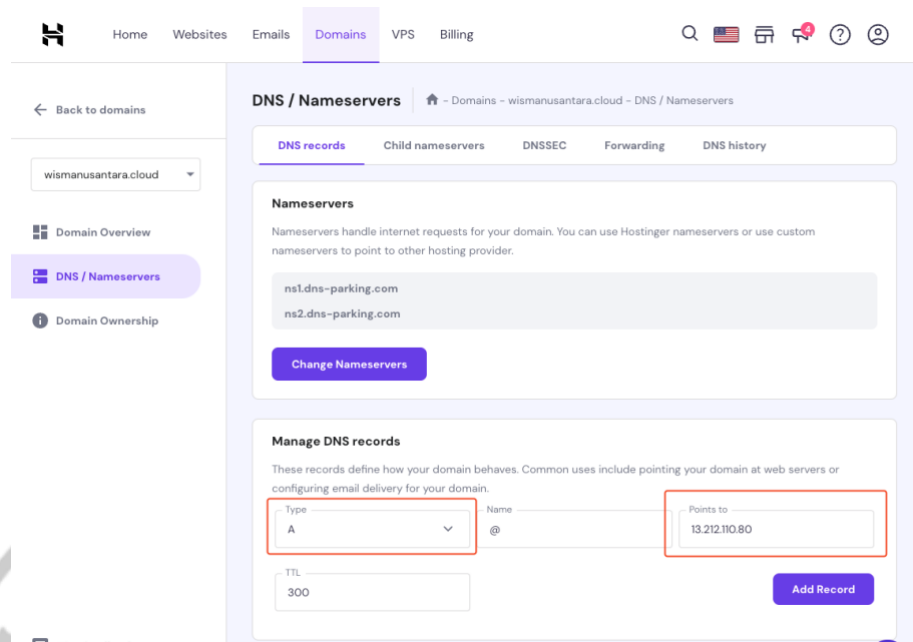


Gambar 4.9. Domain wismanusantara.co.id

#### 4.1.7.2. Setting DNS Hostinger

Langkah berikutnya adalah dengan melakukan pengaturan pada menu *DNS* pada *hostinger* yang bertujuan untuk menghubungkan antara *IP* Publik yang diterbitkan oleh *lightsail* AWS dengan domain yang telah disediakan oleh pihak *hostinger.com*.

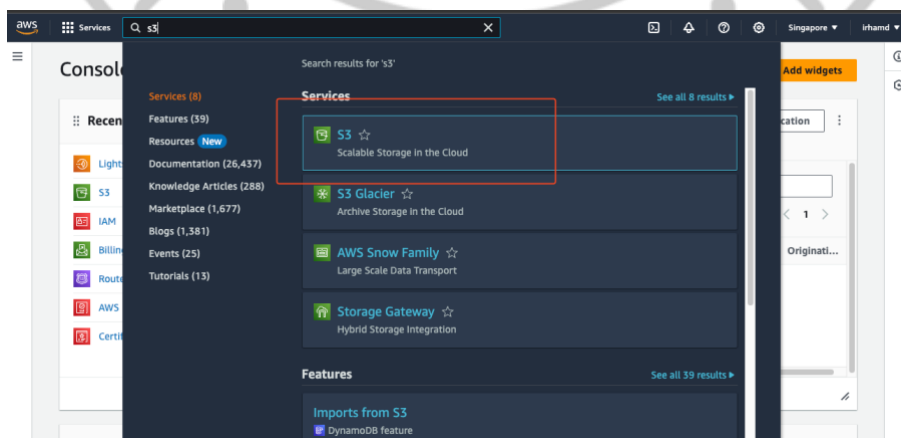




Gambar 4.10. *Setting A Record Ip Public AWS Lightsail*

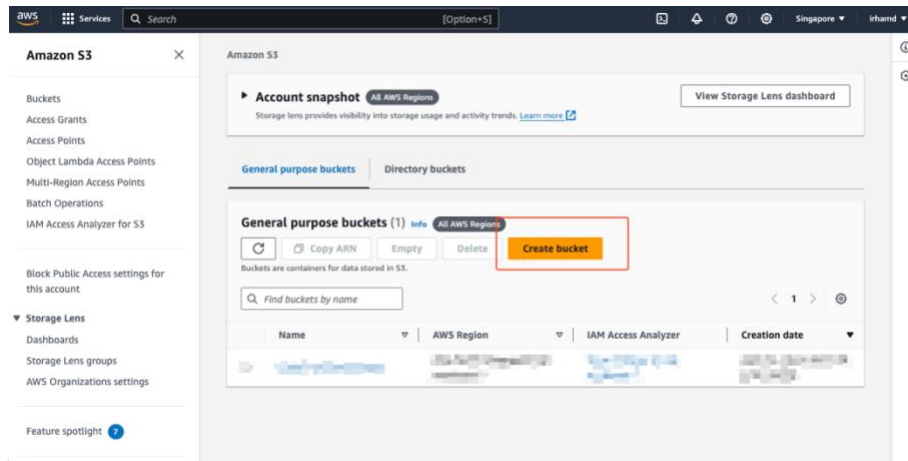
#### 4.2. Instalasi *S3 Bucket AWS*

*S3 (Simple Storage Service)* adalah layanan penyimpanan objek yang disediakan oleh *Amazon Web Services (AWS)*. Ini adalah salah satu layanan paling populer yang digunakan untuk menyimpan dan mengelola data secara aman di *cloud*. Dengan *S3*, Kita dapat menyimpan berbagai jenis data, termasuk file gambar foto kamar, video, dokumen teks, file dan banyak lagi dan memungkinkan akses data yang cepat dan andal dari mana saja di internet.

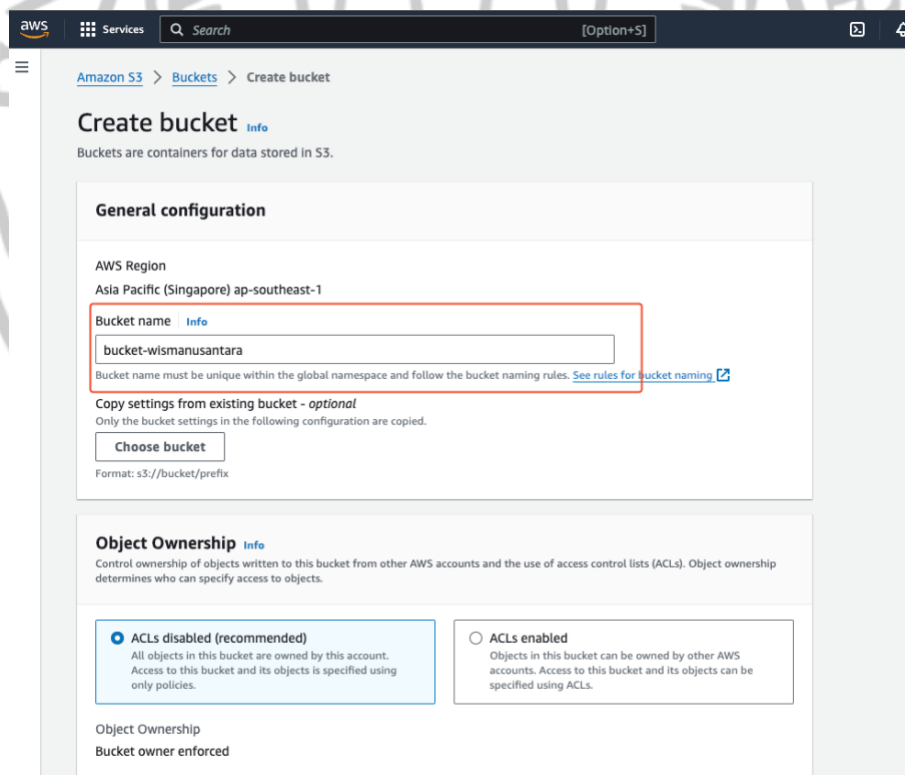


Gambar 4.11. *Instalasi S3 Bucket*

Setelah itu selanjutnya membuat *bucket* baru dengan memilih tombol “*Create Bucket*” tampak pada gambar berikut ini.



Gambar 4.12. Buat Bucket Baru



Gambar 4.13. Menentukan nama bucket

**Tags - optional (0)**  
 You can use bucket tags to track storage costs and organize buckets. [Learn more](#)

No tags associated with this bucket.  

Add tag

**Default encryption** [Info](#)  
 Server-side encryption is automatically applied to new objects stored in this bucket.

**Encryption type** [Info](#)

- ☒ Server-side encryption with Amazon S3 managed keys (SSE-S3)
- ☐ Server-side encryption with AWS Key Management Service keys (SSE-KMS)
- ☐ Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)  
 Secure your objects with two separate layers of encryption. For details on pricing, see [DSSE-KMS pricing](#) on the [Storage](#) tab of the [Amazon S3 pricing page](#).

**Bucket Key**  
 Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. [Learn more](#)

- ☐ Disable
- ☒ Enable

**Advanced settings**

ⓘ After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

Cancel

Create bucket

Gambar 4.14. *Create bucket baru*

Setelah Anda membuat S3 bucket di AWS, langkah-langkah berikutnya bergantung pada apa yang ingin Anda lakukan dengan bucket tersebut, seperti mengunggah file ke bucket S3 menggunakan konsol AWS, atau mengatur izin akses bucket S3 agar sesuai dengan kebutuhan.

AWS
 Services
 Search
 [Option+S]
 Singapore
 iham

Amazon S3 > Buckets

**Account snapshot** [All AWS Regions](#)  
 Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

View Storage Lens dashboard

General purpose buckets | Directory buckets

**General purpose buckets (1)** [Info](#) [All AWS Regions](#)

Copy ARN

Empty

Delete

Create bucket

Buckets are containers for data stored in S3.

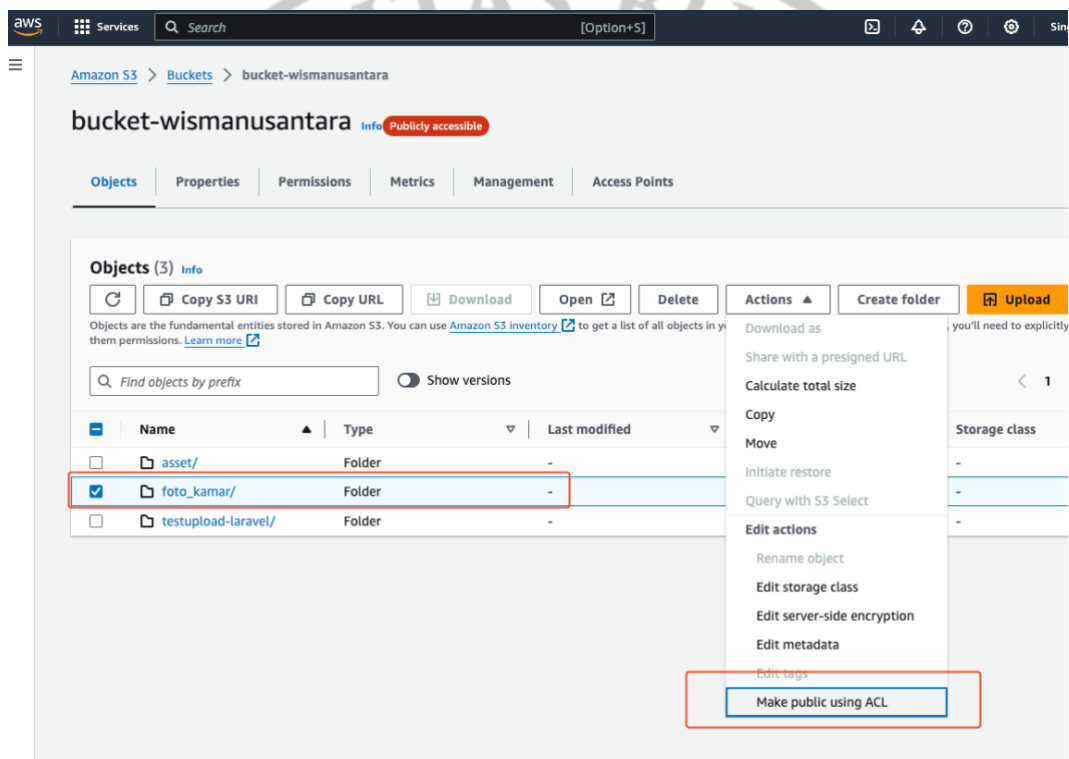
Find buckets by name

Name	AWS Region	IAM Access Analyzer	Creation date
<input type="radio"/> bucket-wismanusantara	Asia Pacific (Singapore) ap-southeast-1	<a href="#">View analyzer for ap-southeast-1</a>	April 24, 2024, 08:27:39 (UTC+08:00)

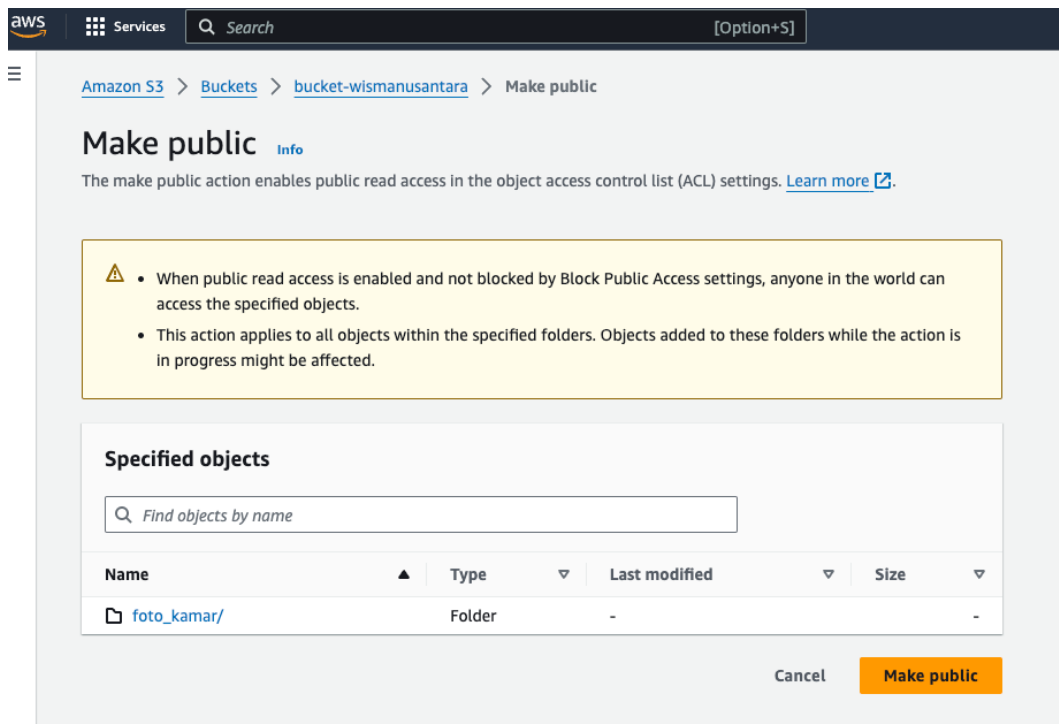
Gambar 4.15. Hasil pembuatan bucket

### 4.3. Akses Public Directory

Memberikan akses publik pada directory atau objek di bucket S3 memiliki beberapa tujuan. Salah satu tujuan umum memberikan akses publik pada directory di bucket S3 adalah untuk hosting situs web statis. Dengan membuat objek misalnya gambar kamar di *bucket S3* agar dapat diakses secara publik, Anda dapat menyajikan situs web statis dengan menggunakan *URL* objek tersebut. Sebagai contoh pada folder foto\_kamar akan diberikan akses agar dapat diakses dari internet.



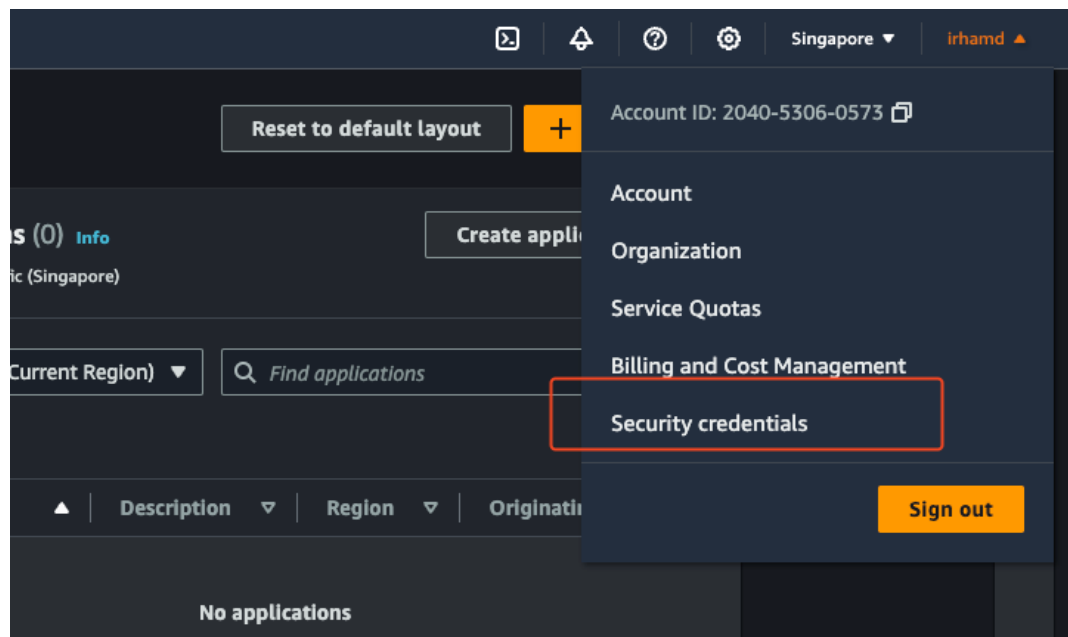
Gambar 4.16. Memberikan izin akses folder



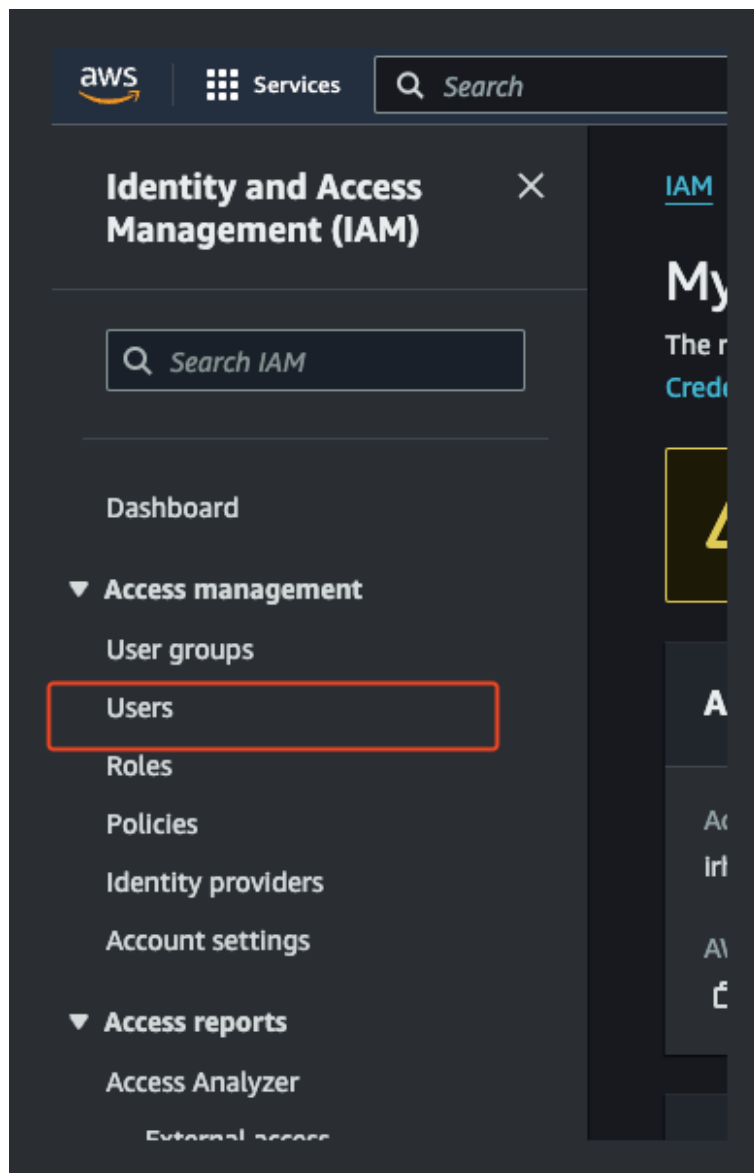
Gambar 4.17. Hasil akses *public* folder

#### 4.4. Membuat User Kridensial Pada AWS

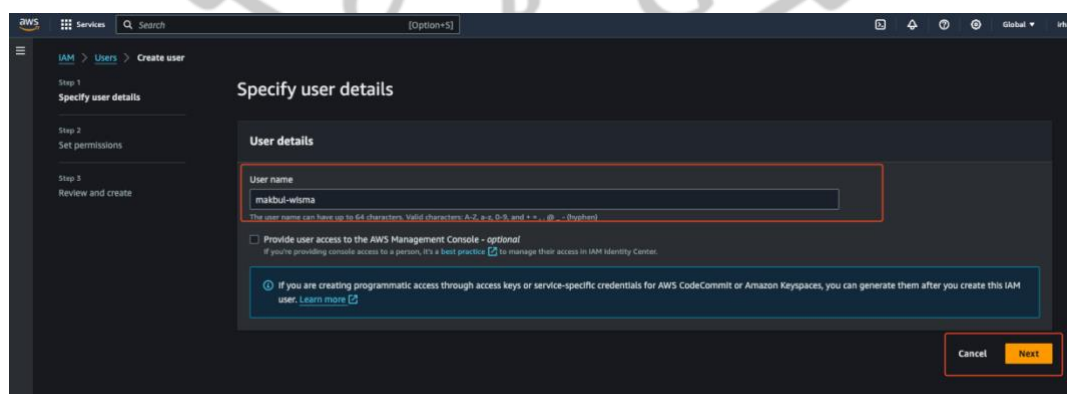
Tujuan membuat user kredensial pada AWS menggunakan layanan IAM adalah untuk mengelola akses ke layanan dan sumber daya AWS secara aman. Dengan membuat user kredensial, Anda dapat mengontrol akses ke layanan dan sumber daya AWS dengan lebih detail. Anda dapat menentukan izin yang spesifik untuk setiap user, yang memungkinkan Anda untuk memberikan akses hanya kepada layanan dan sumber daya yang diperlukan untuk tugas tertentu. Dan berikut ini adalah langkah-langkah untuk membuat *User security credentials*.



Gambar 4.18. Langkah pertama *Install User security credentials*

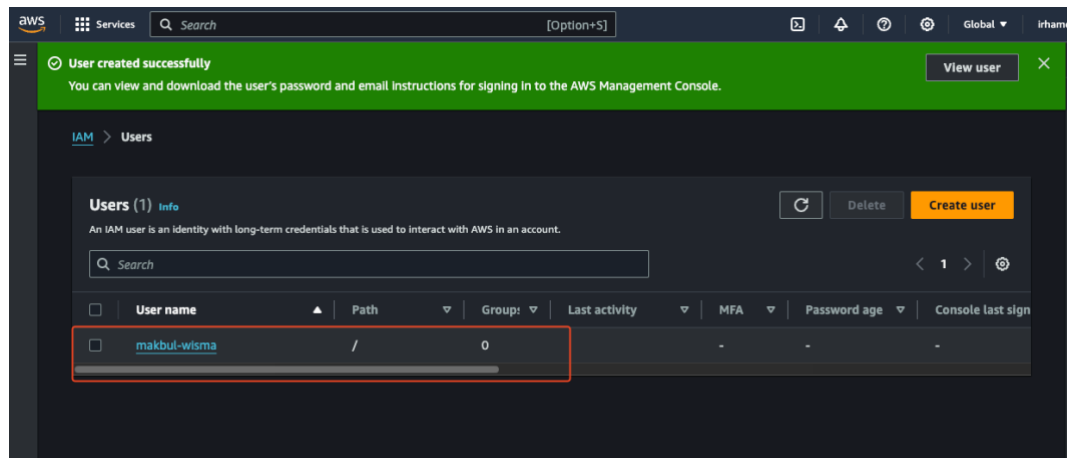


Gambar 4.19. Langkah kedua Install *User security credentials*



Gambar 4.20. Menentukan nama user *User security credentials*

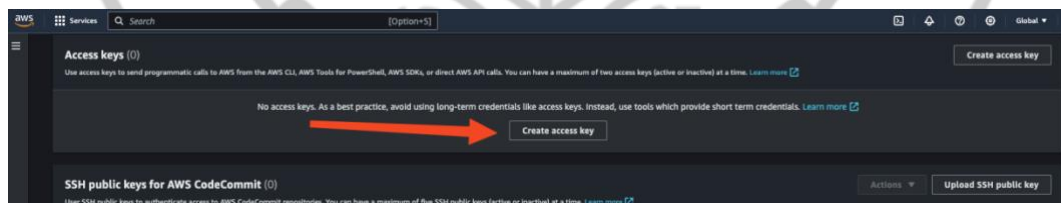




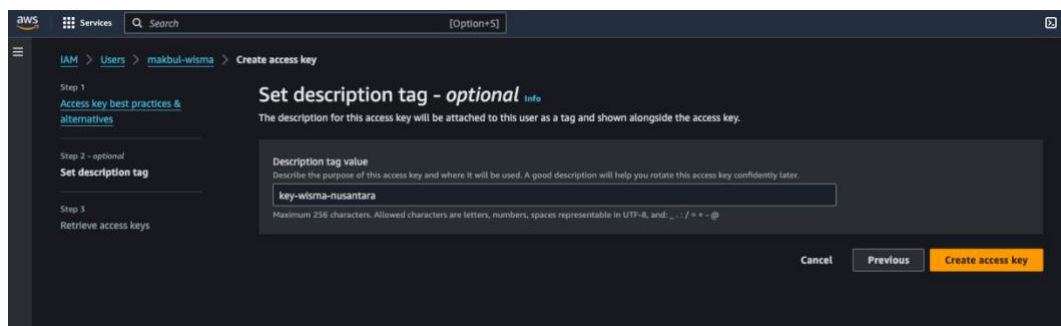
Gambar 4.21. Hasil pembuatan *User security credentials*

#### 4.5. Membuat Access Key User

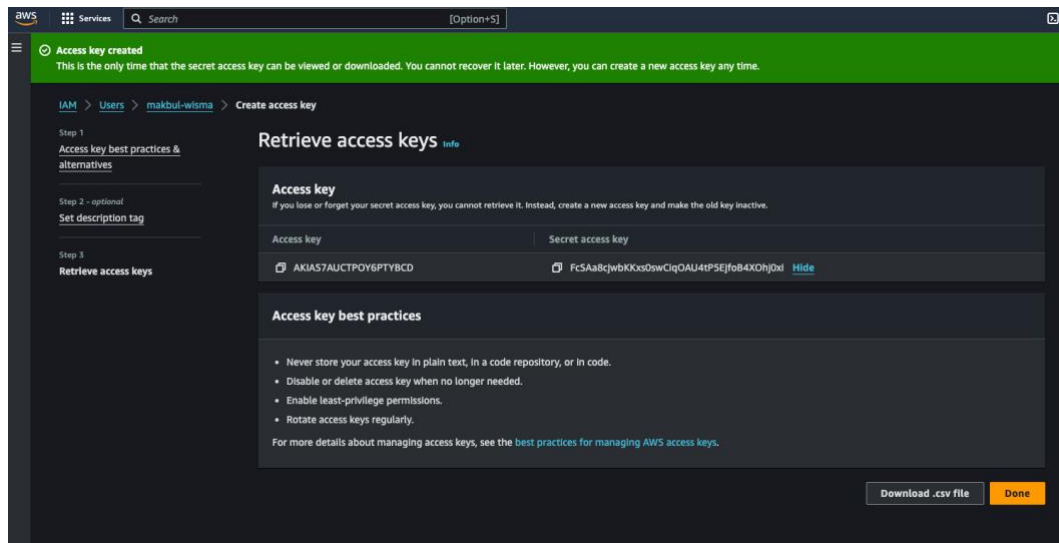
Tujuan utama membuat Access Key untuk user pada AWS adalah untuk memberikan cara aman bagi user tersebut untuk mengakses dan mengelola layanan AWS melalui *API* Laravel atau aplikasi lainnya. *Qaw2q* digunakan untuk otentikasi user ketika mengakses layanan AWS. Dengan menggunakan Access Key, user dapat melakukan operasi seperti mengelola bucket *S3*, dan mengonfigurasi layanan lainnya melalui *API* laravel. Dengan menggunakan *Access Key*, user dapat mengakses layanan AWS secara aman dan efisien, sambil memberikan kontrol yang tepat kepada administrator untuk mengelola akses dan melacak aktivitas pengguna. Berikut ini adalah langkah-langkah untuk membuat *user access key*.



Gambar 4.22. *Create access key*



Gambar 4.23. Deskripsi *access key*



Gambar 4.24. Hasil *create access key*

## 4.6. Konfigurasi Dan Instalasi Package AWS S3 Bucket Pada Laravel

### 4.6.1. Instalasi *Package AWS S3 Bucket*

Untuk menghubungkan *AWS S3 Bucket* dengan *Laravel*, diperlukan library resmi *AWS SDK for PHP* atau menggunakan library pihak ketiga seperti "*league/flysystem-aws-s3-v3*". Tahap ini adalah untuk melakukan instalasi *package* AWS pada *Laravel* melalui terminal dengan perintah

**"composer require --with-all-dependencies league/flysystem-aws-s3-v3 "^1.0"**

#### # Driver Prerequisites

##### # Composer Packages

Before using the S3 or SFTP drivers, you will need to install the appropriate package via the Composer package manager:

○ Amazon S3:

```
composer require --with-all-dependencies league/flysystem-aws-s3-v3 "^1.0"
```

○ SFTP: `composer require league/flysystem-sftp "~1.0"`

In addition, you may choose to install a cached adapter for increased performance:

○ CachedAdapter: `composer require league/flysystem-cached-adapter "~1.0"`

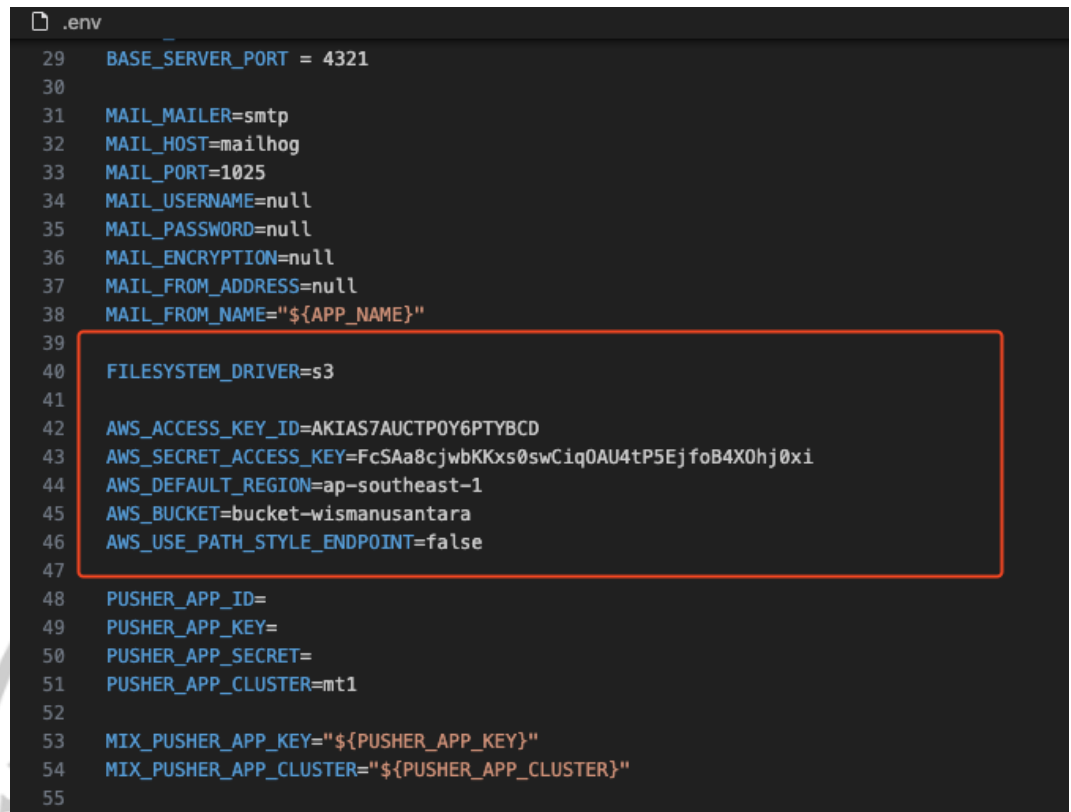
Gambar 4.25. Instalasi Package AWS S3 Bucket Laravel

```
DEBUG CONSOLE  TERMINAL  PORTS
● irhamdifaizard@IrhamdMac skripsi-makbul % composer require --with-all-dependencies league/flysystem-aws-s3-v3 "^1.0"
./composer.json has been updated
Running composer update league/flysystem-aws-s3-v3 --with-all-dependencies
Loading composer repositories with package information
Updating dependencies
Lock file operations: 4 installs, 9 updates, 0 removals
- Locking aws/aws-crt-php (v1.2.5)
- Locking aws/aws-sdk-php (3.305.2)
- Upgrading guzzlehttp/guzzle (7.5.0 => 7.8.1)
- Upgrading guzzlehttp/promises (1.5.2 => 2.0.2)
- Upgrading guzzlehttp/psr7 (2.4.3 => 2.6.2)
- Locking league/flysystem-aws-s3-v3 (1.0.30)
- Upgrading league/mime-type-detection (1.11.0 => 1.15.0)
- Locking mtdowling/jmespath.php (2.7.0)
- Upgrading psr/http-client (1.0.1 => 1.0.3)
- Upgrading psr/http-factory (1.0.1 => 1.0.2)
- Upgrading psr/http-message (1.0.1 => 2.0)
- Upgrading symfony/deprecation-contracts (v2.5.2 => v2.5.3)
- Upgrading symfony/polyfill-mbstring (v1.27.0 => v1.29.0)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 4 installs, 9 updates, 0 removals
- Downloading aws/aws-crt-php (v1.2.5)
- Downloading mtdowling/jmespath.php (2.7.0)
- Downloading aws/aws-sdk-php (3.305.2)
- Downloading league/flysystem-aws-s3-v3 (1.0.30)
- Installing aws/aws-crt-php (v1.2.5): Extracting archive
- Upgrading symfony/polyfill-mbstring (v1.27.0 => v1.29.0): Extracting archive
- Upgrading symfony/deprecation-contracts (v2.5.2 => v2.5.3): Extracting archive
- Upgrading league/mime-type-detection (1.11.0 => 1.15.0): Extracting archive
- Upgrading guzzlehttp/promises (1.5.2 => 2.0.2): Extracting archive
- Upgrading psr/http-message (1.0.1 => 2.0): Extracting archive
- Upgrading psr/http-factory (1.0.1 => 1.0.2): Extracting archive
- Upgrading guzzlehttp/psr7 (2.4.3 => 2.6.2): Extracting archive
- Upgrading psr/http-client (1.0.1 => 1.0.3): Extracting archive
- Upgrading guzzlehttp/guzzle (7.5.0 => 7.8.1): Extracting archive
- Installing mtdowling/jmespath.php (2.7.0): Extracting archive
- Installing aws/aws-sdk-php (3.305.2): Extracting archive
- Installing league/flysystem-aws-s3-v3 (1.0.30): Extracting archive
3 package suggestions were added by new dependencies, use `composer suggest` to see details.
Package swiftmailer/swiftmailer is abandoned, you should avoid using it. Use symfony/mailer instead.
Generating optimized autoload files
> illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
Discovered Package: facade/ignition
Discovered Package: fruitcake/laravel-cors
Discovered Package: irazasyed/telegram-bot-sdk
Discovered Package: laravel/sail
Discovered Package: laravel/sanctum
Discovered Package: laravel/tinker
Discovered Package: nesbot/carbon
Discovered Package: nunomaduro/collision
Discovered Package: php-flasher/flasher-laravel
```

Gambar 4.26. Instalasi *Package* S3 Bucket

#### 4.6.2. Konfigurasi *environment* pada Laravel

Setelah package terinstal, selanjutnya buka file `.env` di direktori proyek Laravel dan Tambahkan kredensial AWS Anda sebagai variabel seperti `AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY`, `AWS_DEFAULT_REGION`, `AWS_BUCKET` dengan nilai-nilai yang sesuai dari akun AWS yang sudah didapat.



```

29  BASE_SERVER_PORT = 4321
30
31  MAIL_MAILER=smtp
32  MAIL_HOST=mailhog
33  MAIL_PORT=1025
34  MAIL_USERNAME=null
35  MAIL_PASSWORD=null
36  MAIL_ENCRYPTION=null
37  MAIL_FROM_ADDRESS=null
38  MAIL_FROM_NAME="${APP_NAME}"
39
40  FILESYSTEM_DRIVER=s3
41
42  AWS_ACCESS_KEY_ID=AKIAS7AUCTP0Y6PTYBCD
43  AWS_SECRET_ACCESS_KEY=FcSAa8cjwbKKxs0swCiq0AU4tP5EjfoB4X0hj0xi
44  AWS_DEFAULT_REGION=ap-southeast-1
45  AWS_BUCKET=bucket-wismanusantara
46  AWS_USE_PATH_STYLE_ENDPOINT=false
47
48  PUSHER_APP_ID=
49  PUSHER_APP_KEY=
50  PUSHER_APP_SECRET=
51  PUSHER_APP_CLUSTER=mt1
52
53  MIX_PUSHER_APP_KEY="${PUSHER_APP_KEY}"
54  MIX_PUSHER_APP_CLUSTER="${PUSHER_APP_CLUSTER}"
55

```

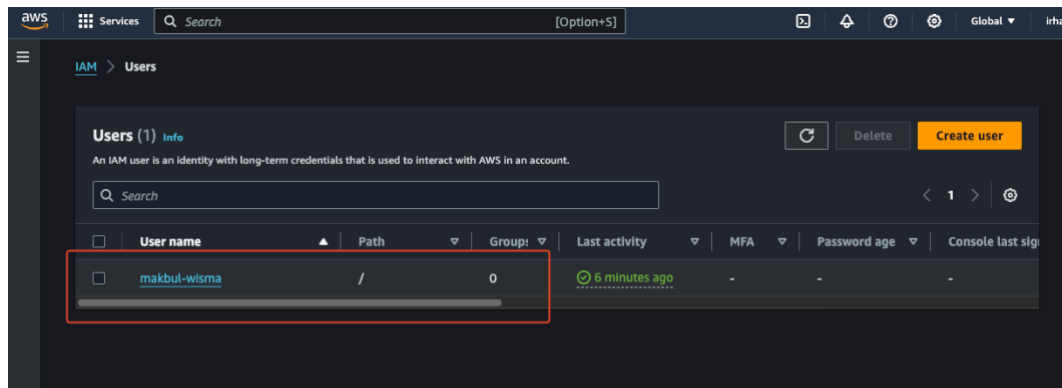
Gambar 4.27. Hasil perubahan nilai *.env file* pada Laravel

Dengan langkah-langkah di atas, sekarang telah terhubung AWS S3 Bucket dengan aplikasi Laravel Anda dan dapat mulai mengelola file di bucket tersebut melalui Laravel menggunakan Storage Facade dan pastikan untuk mengganti nilai kredensial dan konfigurasi dengan nilai-nilai yang sesuai dari akun AWS S3 Bucket yang telah dibuat.

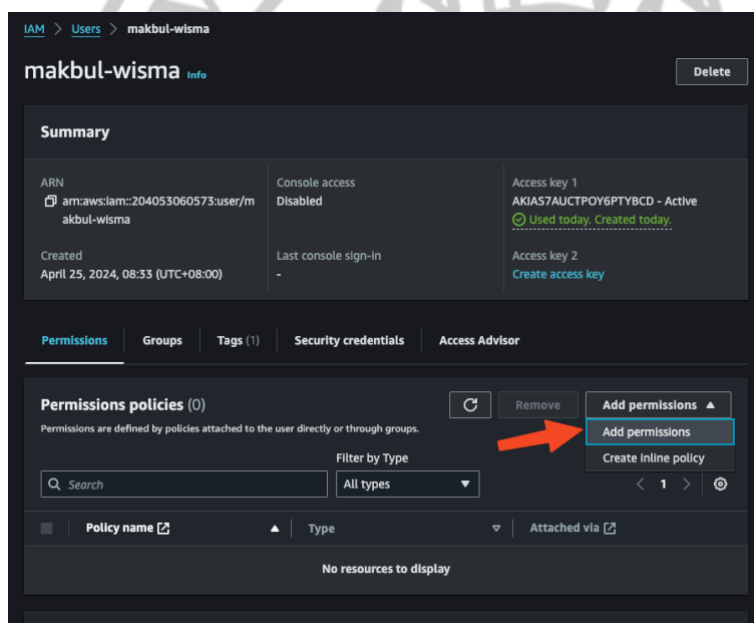
#### 4.7. Setting *Permissions policies* AWS S3 Bucket

Dalam mengatur kebijakan izin untuk bucket AWS S3 sangat penting karena beberapa alasan seperti keamanan data, mencegah pelanggaran data, mencegah kehilangan data, atau optimisasi sumber daya. Dengan mengatur izin yang sesuai, dengan memastikan bahwa hanya pengguna atau layanan yang diotorisasi yang dapat membaca, menulis, atau menghapus objek dalam bucket. Selain melindungi data dari akses yang tidak diizinkan, kebijakan izin juga membantu mencegah penghapusan atau modifikasi tidak sengaja dari objek dalam bucket S3. Dengan

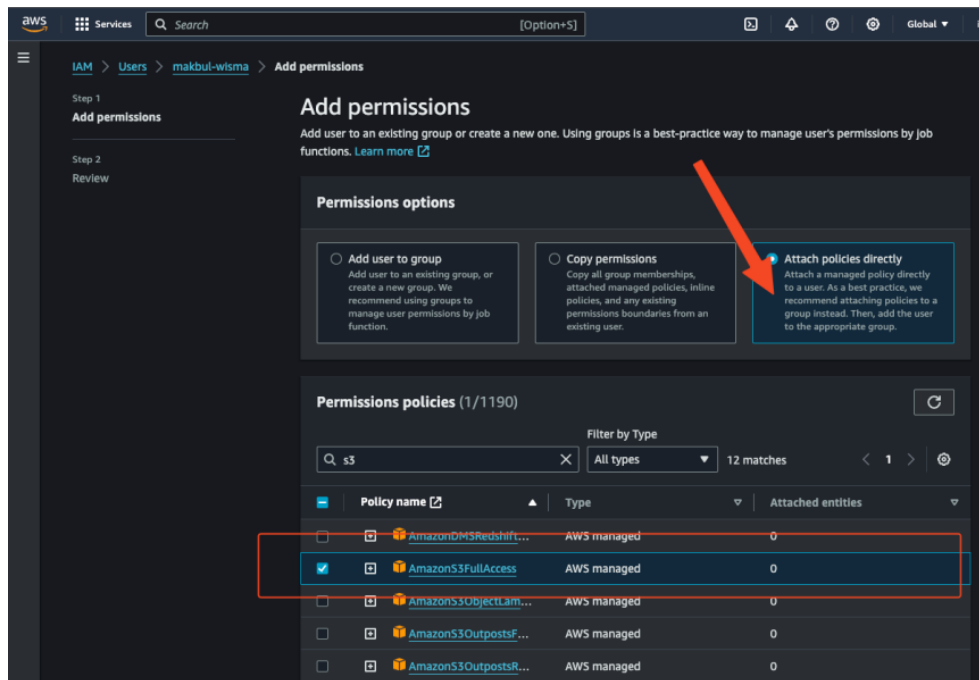
membatasi izin menulis dan menghapus hanya kepada pengguna yang diotorisasi, Anda dapat mengurangi risiko kehilangan data akibat kesalahan manusia. Berikut ini adalah tata cara pengelolaan permissions policies.



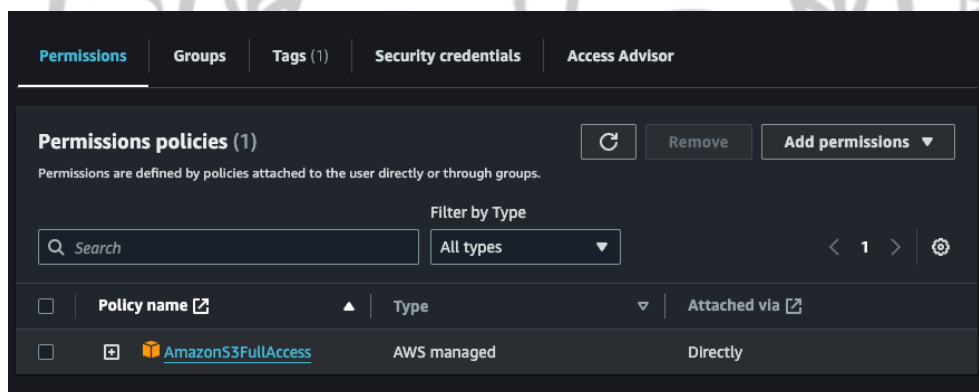
Gambar 4.28. Pilih *username*



Gambar 4.29. Add *permission policies*



Gambar 4.30. Pilih *policy name AmazonS3FullAccess*

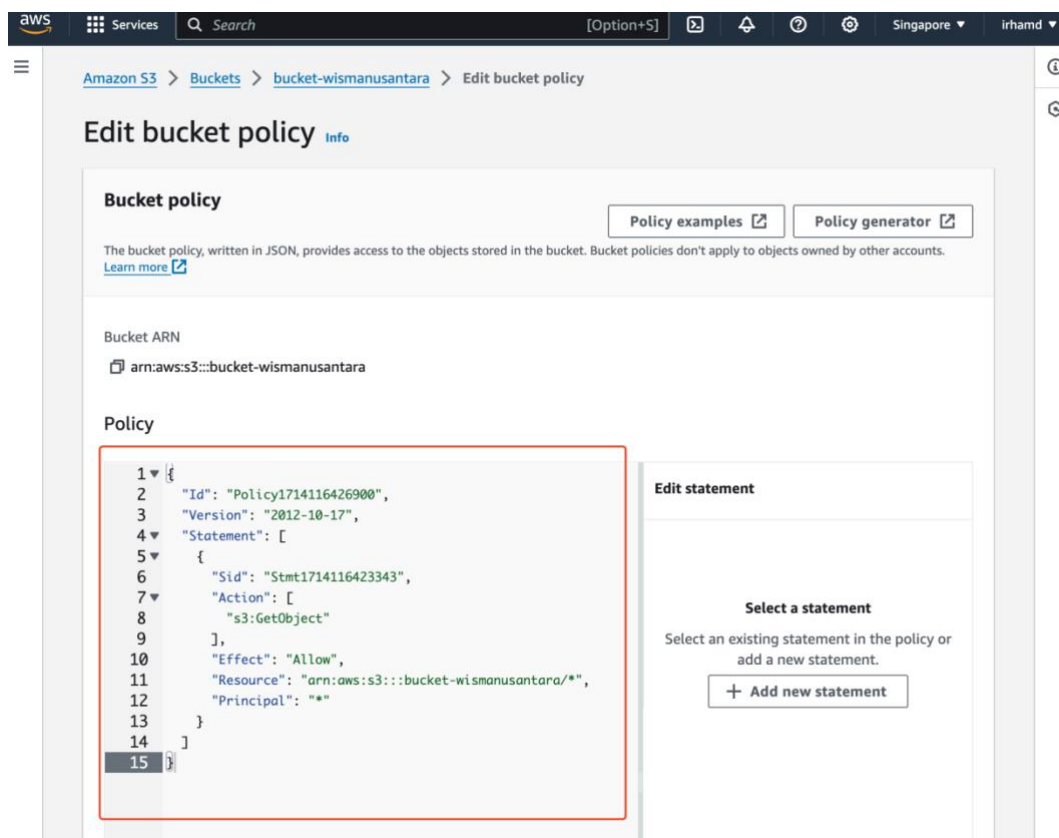


Gambar 4.31. Hasil pengaturan *policy permission*

#### 4.8. Berikan Akses *Bucket Policy S3 Bucket*

Kebijakan bucket S3 untuk membuat bucket dapat diakses oleh publik diperlukan terutama untuk situasi di mana hal ini diperlukan untuk membuat konten atau data yang disimpan dalam bucket dapat diakses dari proyek Laravel secara publik dari proyek laravel seperti gambar, atau file yang ingin didistribusikan secara luas, membuat bucket S3 publik memudahkan distribusi konten tersebut

tanpa perlu membatasi akses serta kemudahan akses dalam membuat bucket publik juga bisa bermanfaat jika ingin mengakses data tersebut dari proyek aplikasi yang lain tanpa harus menggunakan autentikasi khusus. Berikut ini perintah json yang diperlukan seperti *Effect, Principal, Action, Resource*.



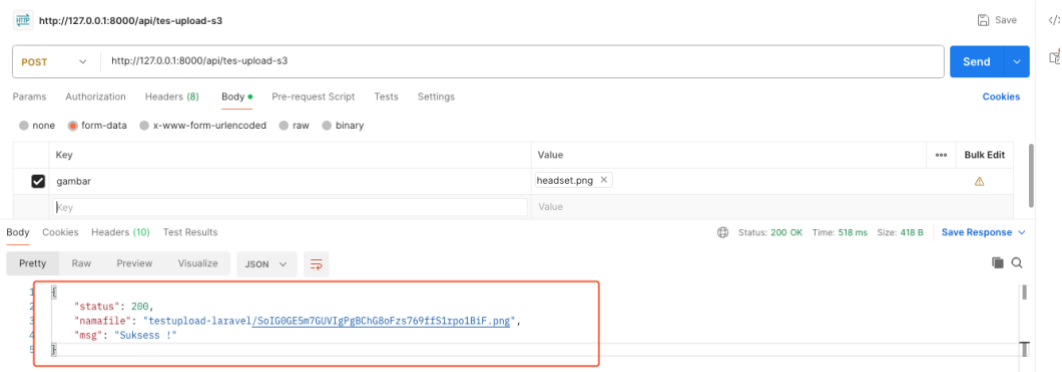
Gambar 4.32. Allow access policy S3 bucket

#### 4.9. Pengujian Upload File dari Laravel ke S3 Bucket AWS

Untuk menguji proses upload file dari aplikasi Laravel ke bucket AWS S3 menggunakan Postman. Dengan memastikan bahwa kita telah melakukan konfigurasi pada aplikasi Laravel dan mengonfigurasi pengaturan untuk koneksi ke AWS S3 dan endpoint ini memiliki metode Post dan menggunakan header Content-Type untuk menentukan tipe konten yang dikirim dalam proyek laravel menggunakan *form-data* di *Postman* dan juga memastikan bahwa kredensial AWS yang digunakan oleh aplikasi Laravel telah memiliki izin yang cukup untuk mengunggah file ke bucket S3. Sekarang dari proyek laravel dapat mengirimkan

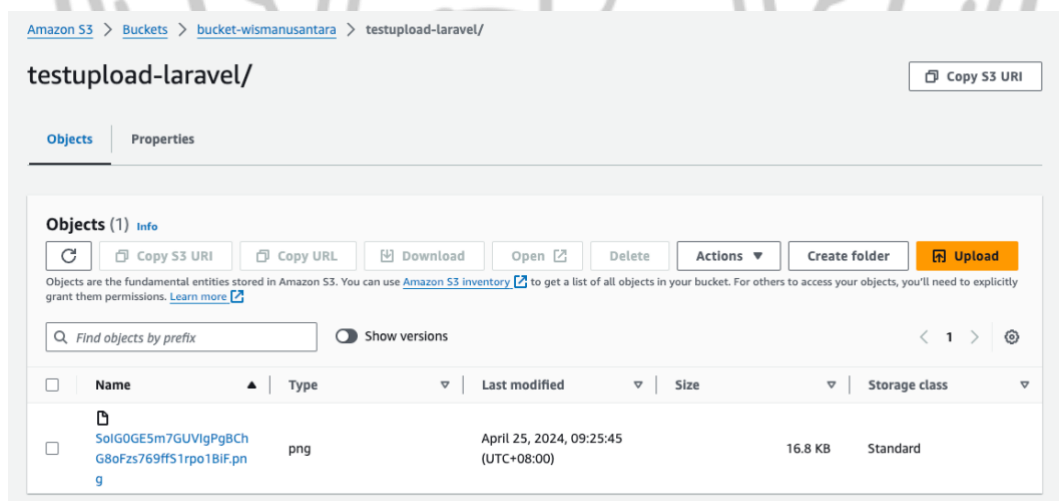


permintaan dari Postman dan memeriksa responsnya. Berikut ini adalah bukti pengujian upload file dari postman.



Gambar 4.33. *Testing upload file dari Postman*

Uji coba *upload file* juga merupakan pengujian integrasi antara aplikasi Laravel dan layanan eksternal AWS S3 Bucket memastikan bahwa integrasi antara kedua platform berjalan dengan baik dan tidak ada masalah komunikasi atau keamanan yang muncul. Dengan melakukan uji coba upload file ke bucket AWS S3 dari aplikasi Laravel, akan dapat memastikan bahwa sistem beroperasi dengan baik dan siap untuk digunakan dalam lingkungan produksi yang dapat membantu mencegah masalah dan kegagalan yang mungkin terjadi ketika aplikasi.



Gambar 4.34. Hasil upload file dari postman ke S3 Bucket

#### 4.10. Tampilan Antar Muka Aplikasi

Berikut adalah deskripsi antarmuka pencarian kamar di aplikasi Pemesanan Kamar pada Wisma Nusantara. Terdapat form pencarian yang mencakup input untuk tanggal check-in serta jumlah yang diinginkan dan pengguna dapat memilih tanggal menggunakan kalender atau memasukkan manual. Klik tombol "Cari Kamar" untuk memulai pencarian berdasarkan kriteria yang dimasukkan. Filter Opsi Pencarian adalah opsi untuk memfilter hasil pencarian berdasarkan kriteria tertentu seperti jumlah bed, fasilitas, dan tanggal penginapan yang diinginkan. Untuk lebih jelasnya dapat dilihat pada lampiran gambar berikut ini.

**WISMA NUSANTARA**

» HOME » DAFTAR KAMAR » TENTANG KAMI [Login](#)

**HOTEL**

**WISMA NUSANTARA**

**IMPLEMENTASI PEMESAN RUANGAN WISMA NUSANTARA**

**Boking Sekarang**

Kamar Biasa Kamar Kelompok Kamar Vip WhatsApp Telpon Email

**CARI KAMAR DISINI**

Tanggal (DD/MM/YYYY) :: 25/04/2024 << Sampai >> 26/04/2024

Urutkan Berdasarkan :: Semua Kriteria Ru

Kategori Produk :: Semua Kategori

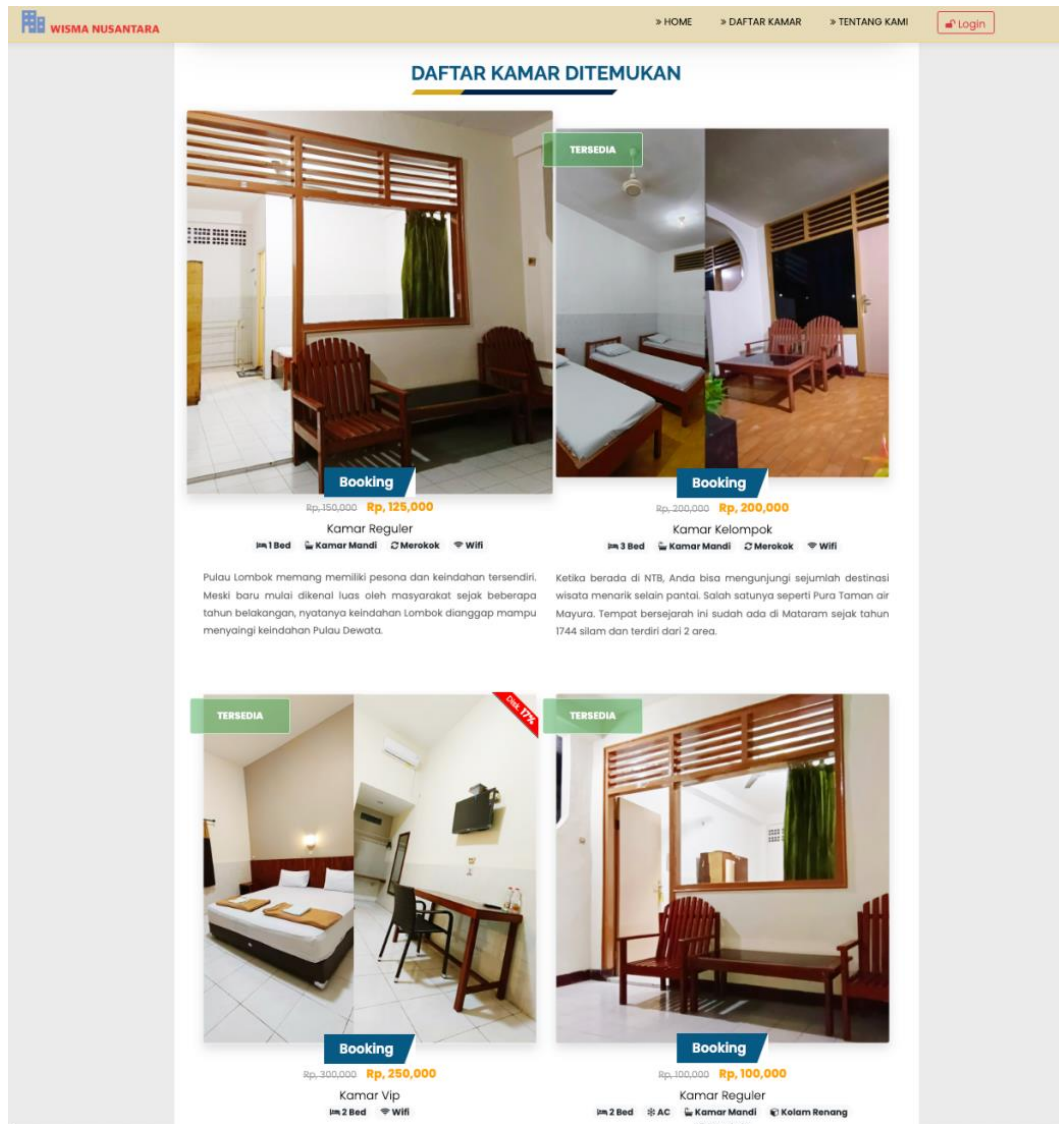
Jumlah Bed :: 2 Bed

Fasilitas :: ☐ AC ☒ Kamar Mandi ☐ Kolam Renang ☒ Merokok ☐ Wifi

**Cari Kamar**

Gambar 4.35. Form pencarian kamar

Setelah pengunjung melakukan pencarian kamar berdasarkan pada gambar diatas, maka aplikasi akan mengarahkan pengguna untuk melihat daftar kamar-kamar yang masih tersedia atau belum diboeking oleh pengunjung yang lain berdasarkan kriteria-kritea yang sudah ditentukan pada gambar 4.35 diatas.



Gambar 4.36. Hasil pencarian kamar

Saat pengguna mengklik salah satu kamar dari daftar hasil pencarian, mereka akan dibawa ke halaman detail kamar dan halaman detail kamar menampilkan informasi lebih lengkap tentang kamar tersebut, termasuk deskripsi, fasilitas, harga per malam,

gambar-gambar kamar, dan tombol untuk melakukan pemesanan. Berikut adalah deskripsi interface hasil pencarian kamar detail pada aplikasi seperti informasi kamar, foto kamar, fasilitas, tipe , harga permalam, dan memberikan daftar kamar sejenis yang dapat dipilih jumlahnya dan melakukan konfirmasi terhadap tanggal dan harga akumulasi pada tanggal yang akan di booking, setelah itu klik tombol “Booking Sekarang”

WISMA NUSANTARA

» HOME » DAFTAR KAMAR » TENTANG KAMI Login

Home / Kamar / Kamar Vip

### Kamar Vip

Diskon: 17 %

**Rp, 300,000** Rp: Rp, 0

**Deskripsi:**  
Wisma Nusantara adalah sebuah penginapan kelas melati yang letaknya ada di pusat Kota Mataram, tepatnya di Jl. Beo No. 10-12, Cakranegara Barat, Mataram, Nusa Tenggara Barat, 83239, Indonesia

Jumlah: 1

Share: [WhatsApp](#) [Twitter](#) [Facebook](#)

**Daftar Kamar Sejenis :**

Kamar Vip-KMR88493

Booking Kamar No.KMR88493

Tanggal (DD/MM/YYYY) : 25/04/2024 Sampai : 26/04/2024 Lama : 1 malam Jumlah Kamar Dibooking : 1 kamar

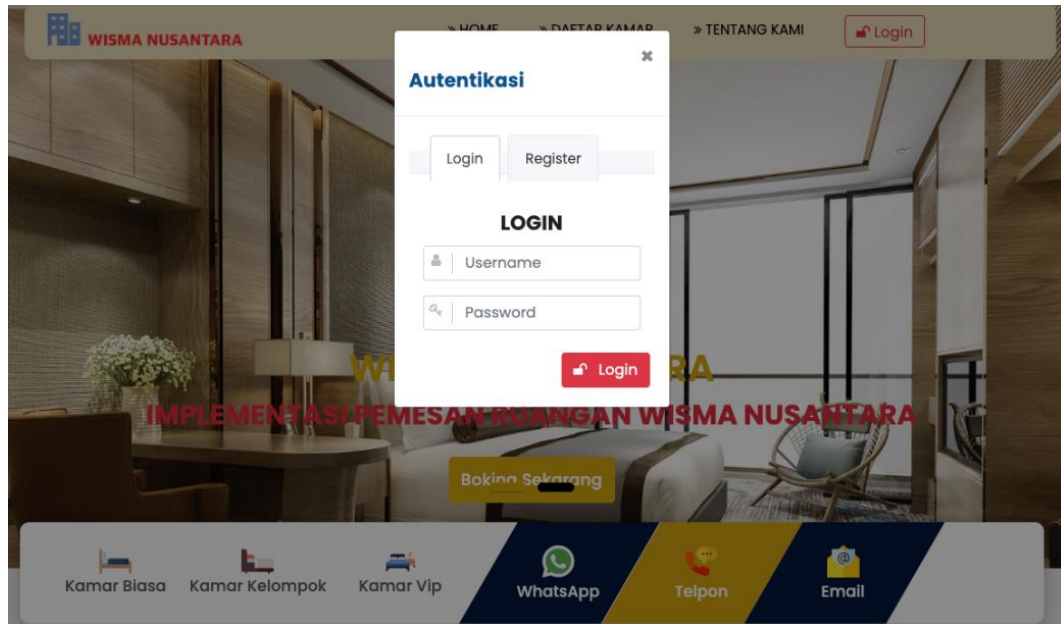
**Rp, 300,000**  
Tiga ratus ribu rupiah

**Booking Sekarang**

Gambar 4.37. Konfirmasi form booking kamar


Dalam kasus di mana pengguna perlu login terlebih dahulu sebelum melakukan pemesanan kamar pada aplikasi Wisma Nusantara dan antarmuka pengguna (UI) dapat disusun dengan masuk ke popup login yang mencakup username dan kata

sandi pengguna jika pengguna atau pengunjung sudah meilikinya. Berikut ini adalah UI untuk popup screen login.



*Gambar 4.38. UI popup screen untuk login*

Jika pengguna harus mendaftar terlebih dahulu sebelum login ke aplikasi, berikut adalah susunan antarmuka pengguna (UI) yang akan dilalui oleh pengunjung yakni dengan melakukan registrasi pada halaman Pendaftaran (Registrasi). Pada halaman ini form mencakup nama lengkap pengunjung, no whatsapp yang dapat digunakan oleh admin wisma Wisma Nusantara untuk melakukan follow up jika diperlukan, alamat lngkap, username, dan kata sandi yang digunakan untuk autentikasi pada aplikasi. Setelah itu klik tombol “Simpan”.

 **WISMA NUSANTARA**

» HOME » DAFTAR KAMAR » TENTANG KAMI

Login

---

**REGISTER**

Nama Lengkap \*

Imam Makbul

No. WA \*

0853336954

Alamat Lengkap \*

Jl. Bungarno Mo. 34 Mataram

Pastikan alamat anda diisi dengan benar, sebagai alamat pengiriman diet

---

Username \*

makbul

Password \*

.....

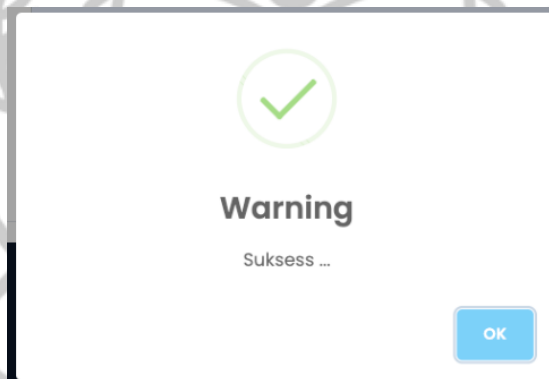
Password Lagi \*

.....

Reset

Simpan

Gambar 4.39. Form registrasi pengunjung

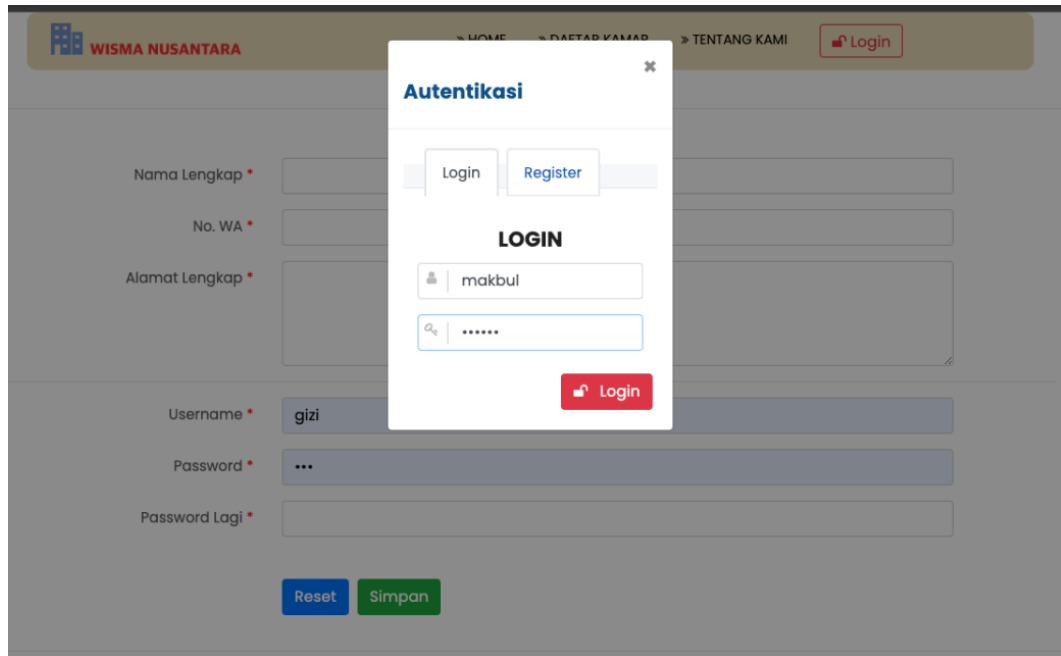


Gambar 4.40. Informasi registrasi berhasil

Untuk login ulang setelah berhasil registrasi pada aplikasi langkah selanjutnya adalah melakukan login ulang dengan mencari opsi "Login" atau "Masuk" di layar utama atau menu navigasi aplikasi klik atau klik opsi tersebut untuk membuka halaman login. Pada halaman login, masukkan kredensial yang Anda gunakan saat mendaftar, seperti nama pengguna (username) dan kata sandi (password).



Setelah memasukkan informasi login, klik atau ketuk tombol "Login" untuk mengakses akun yang tadi di-*register*.




The screenshot displays the login page of 'WISMA NUSANTARA'. A modal dialog titled 'Autentikasi' is centered on the screen. The dialog has two tabs: 'Login' (selected) and 'Register'. Under the 'LOGIN' tab, there is a form with the following fields: a username field containing 'makbul', a password field with masked characters '\*\*\*\*\*', and a red 'Login' button. The background page is dimmed and shows registration fields: 'Nama Lengkap \*', 'No. WA \*', 'Alamat Lengkap \*', 'Username \*' (with value 'gizi'), 'Password \*' (with masked characters '...'), and 'Password Lagi \*'. At the bottom of the background page are 'Reset' and 'Simpan' buttons. The top navigation bar includes links for 'HOME', 'DAFTAR KAMAR', 'TENTANG KAMI', and a 'Login' button.


Gambar 4.41. Menu login ke Aplikasi

Setelah berhasil login, makan akan diarahkan kembali pada menu booking yang seperti pada gambar 4.37




[» HOME](#)
[» INVOICE](#)
[» DAFTAR KAMAR](#)
[» TENTANG KAMI](#)
Imam Makbul

[Home](#) / [Kamar](#) / [Kamar Vip](#)



### Kamar Vip




Diskon: 17 %

**Rp, 300,000** Rp:-Rp; 0


**Deskripsi:**  
Wisma Nusantara adalah sebuah penginapan kelas melati yang letaknya ada di pusat Kota Mataram, tepatnya di Jl. Beo No. 10-12, Cakranegara Barat, Mataram, Nusa Tenggara Barat, 83239, Indonesia

Jumlah

Share :

**Daftar Kamar Sejenis :**



Kamar Vip-KMR88493

Booking Kamar No.KMR88493

Tanggal (DD/MM/YYYY) :

Sampai :

Lama :

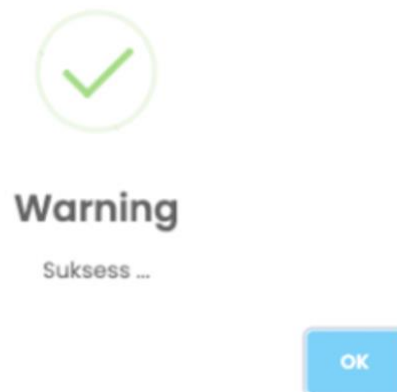
Jumlah Kamar Dibooking :

**Rp, 300,000**  
Tiga ratus ribu rupiah

[Booking Sekarang](#)


Gambar 4.42. Menu Boking

Notifikasi berhasil booking pada aplikasi akan muncul setelah pengunjung menyelesaikan proses pemesanan dan pengiriman data yang diperlukan. Berikut adalah bentuk notifikasi yang diterima.



Gambar 4.43. Informasi Pemesanan berhasil

Setelah melakukan pemesanan pada aplikasi pengunjung akan menerima tagihan atau invoice dalam aplikasi invoice tersebut akan berisi rincian biaya yang harus dibayar untuk pemesanan kamar tersebut. Tujuan dari invoice tersebut adalah untuk memberikan rincian yang jelas dan terperinci tentang transaksi tagihan yang akan dilunasi pengunjung sebelum status booking didapat. Batas pembayaran dalam invoice ini akan diberikan jangka waktu 12 jam sebagai masa tenggang pembayaran sebelum tagihan tersebut kedaluarsa.

 WISMA NUSANTARA

[» HOME](#) [» INVOICE](#) [» DAFTAR KAMAR](#) [» TENTANG KAMI](#) [Imam Makbul](#)

DAFTAR INVOICE

Status :  Tanggal Invoice :  Nama Pengunjung :  [Filter](#)

Show  entries

No	Kode Booking	Nama	Tanggal Booking	Total Tagihan	Total Bayar	Batas Pembayaran	Status	Action
1	WSN-2545558	Imam Makbul	2024-04-25 21:14:17 Sampai 2024-04-26 12:00:00	Rp. 300,000	Rp. 0	2024-04-26 09:14:17 11 jam dari sekarang	Menunggu Pembayaran	<a href="#">Konfirmasi Pembayaran</a> <a href="#">Batal Booking</a> <a href="#">Detail</a>

Gambar 4.44. Daftar Invoice

**WISMA NUSANTARA**

Imam Makbul

No. WA (Wajib)

0877898888

Tanggal Invoice :

2024-04-25

10 entries

Kode Booking	Nama	Tanggal Booking
WSN-2545558	Imam Makbul	2024-04-25 21:00 Sampai 2024-04-26 12:00

Jumlah Tagihan :

**Rp. 300,000**

Tiga ratus ribu rupiah

Harap lakukan pembayaran ke :

Sebelum

**11 : 50 : 47 detik**

SUDAH DITRANSFER KE

☐ **Bank BRI**  
Atas Nama : **ATAS NAMA AJA**  
Nomor Rekening : **0066-6745554-35443**

☐ **Bank NTB**  
Atas Nama : **ATAS NAMA AJA**  
Nomor Rekening : **10011-2998-333**

☐ **BSI**  
Atas Nama : **ATAS NAMA AJA**  
Nomor Rekening : **453-353-5444**

☐ **BCA**  
Atas Nama : **ATAS NAMA AJA**  
Nomor Rekening : **333-555-33334**

☐ **mandiri**  
Atas Nama : **ATAS NAMA AJA**  
Nomor Rekening : **3345-6434-334**

Upload Bukti Pembayaran (Transfer) disini :

Pilih Bukti Transfer

Jangan Lupa Klik Konfirmasi

**KONFIRMASI**

Carl Sesuatu Disini :

Status	Action
Konfirmasi Pembayaran	<a href="#">Konfirmasi Pembayaran</a>
Menunggu Pembayaran	<a href="#">Batal Booking</a>
	<a href="#">Detail</a>

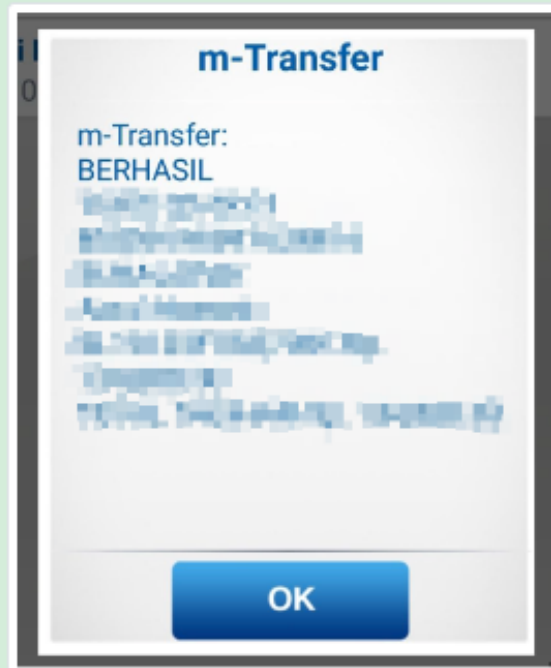
WISMA NUSANTARA

Jl. Bung Karno No.3 Kota Mataram

Gambar 4.1.

**Upload Bukti Pembayaran (Transfer) disini :**

  
Pilih Bukti Transfer



Jangan Lupa Klik Konfirmasi

 **KONFIRMASI**

Gambar 4.2.

#### 4.11. Tabel Hasil Pengujian Black-box

Table pengujian Black-box adalah tabel hasil pengujian dari setiap antarmuka.

Table 4.2. Pengujian *Black-Box*

Pengujian	Jenis Antarmuka	Keterangan
Login dan Register	<ul style="list-style-type: none"><li>• Tombol <i>Login</i></li><li>• Tombol Logout</li></ul>	Berfungsi
Akses Home	<ul style="list-style-type: none"><li>• Tanggal</li><li>• Jumlah Bed</li><li>• Fasilitas</li></ul>	Berfungsi
Cari Kamar	<ul style="list-style-type: none"><li>• Daftar Kamar</li></ul>	Berfungsi
Detail Kamar	<ul style="list-style-type: none"><li>• Data Kamar</li><li>• Pilih Data Kamar</li></ul>	Berfungsi
Booking	<ul style="list-style-type: none"><li>• Booking Kamar</li></ul>	Berfungsi
Upload Bukti Transfer	<ul style="list-style-type: none"><li>• Upload Bukti Transfer Oleh Pelanggan</li></ul>	Berfungsi
Verifikasi Bukti Transfer	<ul style="list-style-type: none"><li>• Oleh Petugas Wisma Nusantara</li></ul>	Berfungsi

#### 4.12. White-box Testing

Pengujian ini dilakukan bertujuan untuk mengetahui hasil penelitian dari struktur *internal* berdasarkan keputusan yang telah ditetapkan.

#### 4.13. Tabel Hasil Pengujian White-box

Tabel hasil pengujian white-box adalah hasil pengujian yang dilakukan oleh beberapa user terkait dengan fungsi dan juga *user experience* dalam menggunakan layanan *Website SMPN 03 PEKAT*.

Table 4.3.      Tabel 4. 1 Pengujian *White-box*



#### 4.14. Pembahasan

Solusi dan implementasi *Cloud Computing* dengan layanan AWS *Lightsail* dan *S3 bucket* di Wisma Nusantara dapat memberikan berbagai keuntungan, termasuk skalabilitas, keamanan, ketersediaan data, dan manajemen biaya yang lebih efisien. Dengan diterapkannya AWS pada aplikasi Pemesanan Kamar Wisma Nusantara ditemukan berapa solusi yang ditemukan, antara lain sebagai berikut :

1. Memudahkan Penunjung Dan Admin Wisma Nusantara  
Dengan adanya system ini memudahkan pengunjung dalam melakukan pemesanan kamar pada Wisma Nusantara dan memudahkan staf atau pengelola dalam melihat dan mengelola semua pemesanan kamar melalui aplikasi. Mereka dapat melihat ketersediaan kamar, menambah atau mengurangi ketersediaan, dan mengelola detail pemesanan dengan mudah.
2. Penggunaan AWS Lightsail untuk Hosting Aplikasi  
Anda dapat menggunakan AWS Lightsail untuk menampung aplikasi web Wisma Nusantara dan menyediakan instance virtual yang mudah dikonfigurasi dan dikelola dengan harga tetap yang terjangkau.
3. Pemanfaatan S3 Bucket untuk Penyimpanan File  
S3 bucket dapat digunakan untuk menyimpan berbagai jenis file, seperti gambar kamar, dokumen Wisma Nusantara, dan data lainnya yang memungkinkan akses cepat dan aman ke file-file tersebut dari aplikasi.
4. Backup dan Penyimpanan Data  
Dapat mengatur rutinitas backup data dari aplikasi dan menyimpannya di S3 bucket. Hal ini memastikan keamanan dan ketersediaan data Anda jika terjadi kegagalan sistem atau bencana.
5. Penggunaan CDN (Content Delivery Network)  
Gunakan CloudFront, layanan CDN dari AWS, untuk mengirimkan konten statis (seperti gambar Kamar, CSS, dan JavaScript) dari S3 bucket Anda ke pengguna dengan cepat dan efisien di seluruh dunia secara *online*.
6. Keamanan dan Manajemen Akses



Terapkan kebijakan keamanan yang ketat menggunakan layanan AWS Identity and Access Management (IAM) untuk mengendalikan akses ke sumber daya AWS. Dan AWS memastikan hanya pengguna yang sah yang memiliki akses yang diperlukan.

