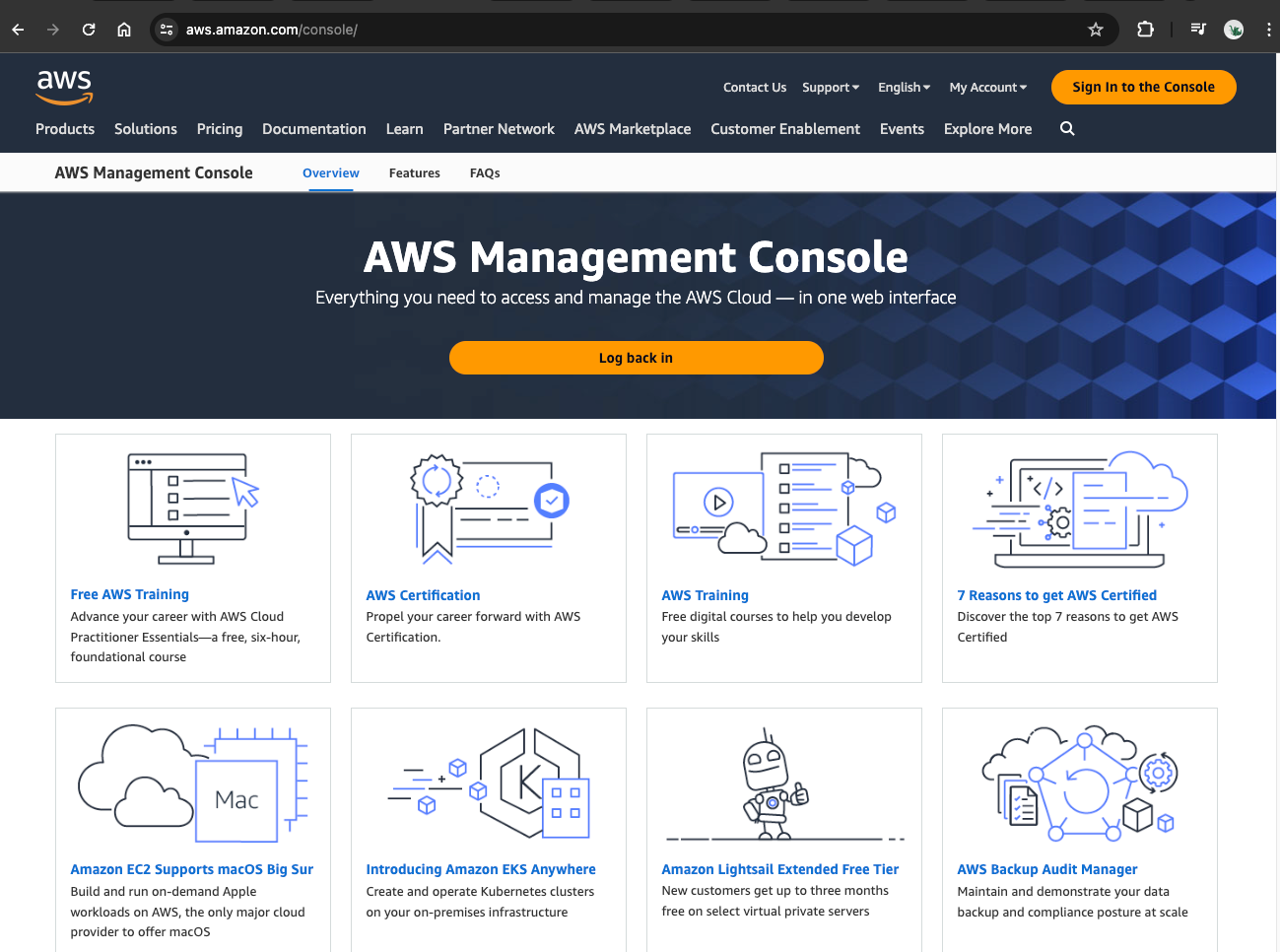
# BAB IV IMPLEMENTASI DAN PEMBAHASAN

## Instalasi Dan Konfigurasi *AWS*

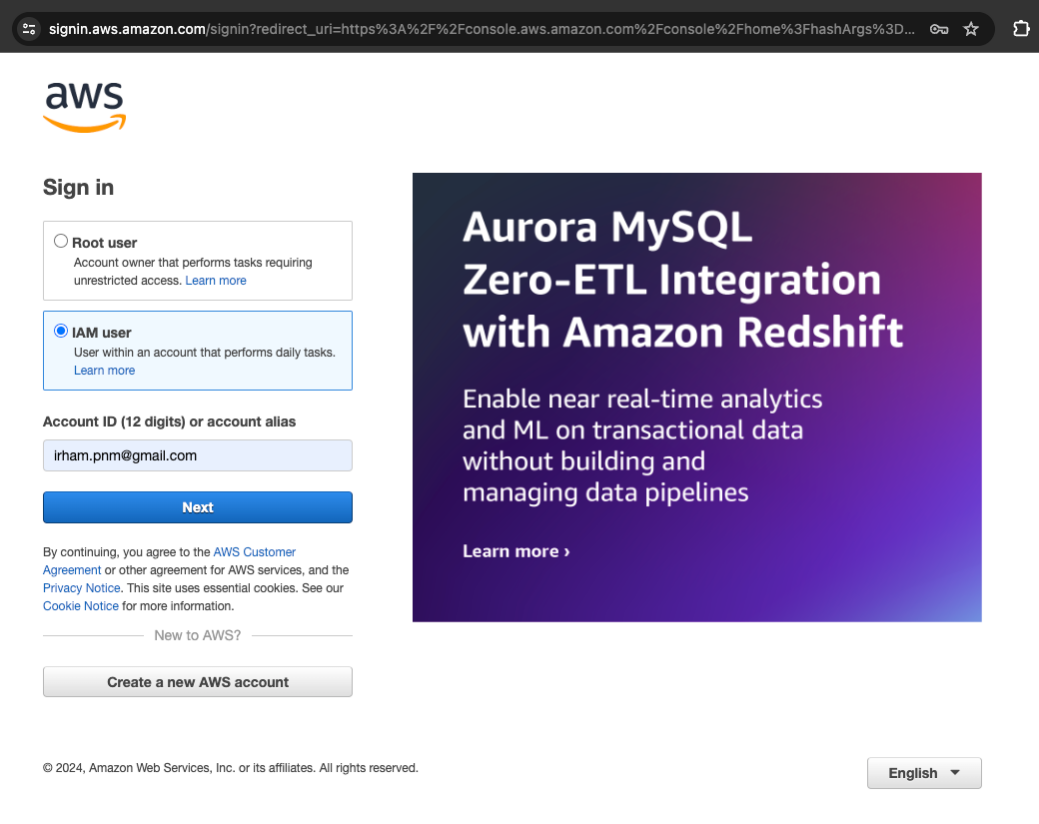
## Amazon Web Services (AWS) adalah cloud terlengkap dan paling banyak digunakan di dunia, menawarkan lebih dari 200 layanan berfitur lengkap dari pusat data secara global. Jutaan pelanggan termasuk perusahaan rintisan dengan pertumbuhan tercepat, perusahaan terbesar, dan lembaga pemerintah terkemuka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



1. Menu Utama AWS

## Login Ke Konsol *AWS*

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



1. *Login* Ke *Dashboard AWS*

## Tampilan *Dashboard* Utama AWS

Setelah melakukan login, AWS akan mengarahkan ke hlaman 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.

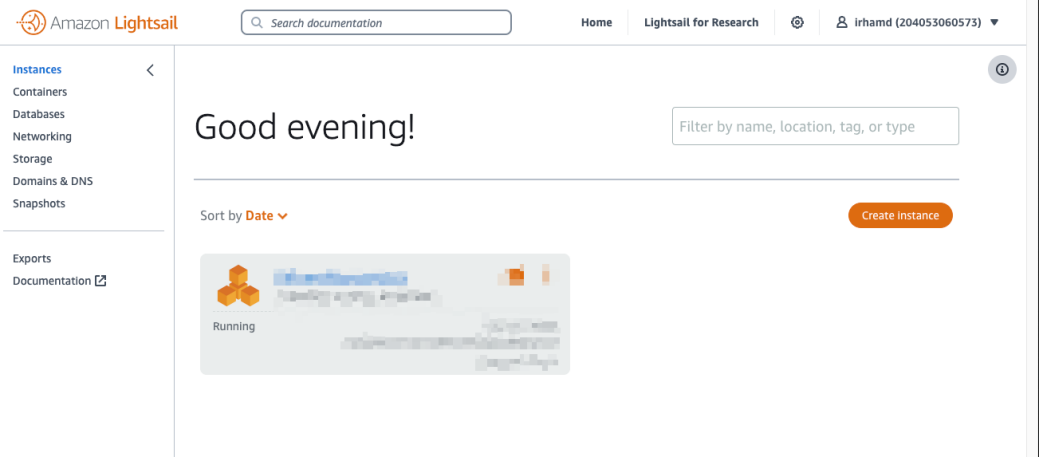
## 

### 

1. Console Home AWS

## Instalasi *Lightsail AWS*

## Langkah ini adalah tahapan dimana user akan melakukan instalasi lightsail AWS dengan meamanfaatkan 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.



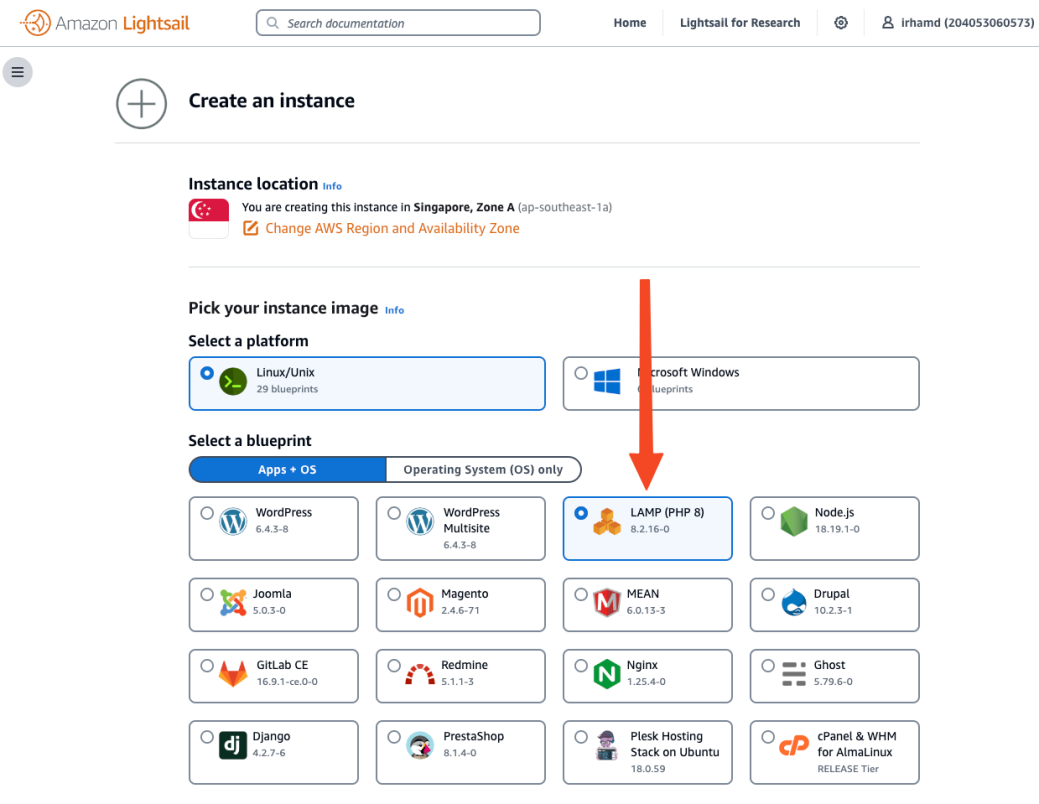
1. Menu AWS Lightsail

## Create Instanse Lightsail

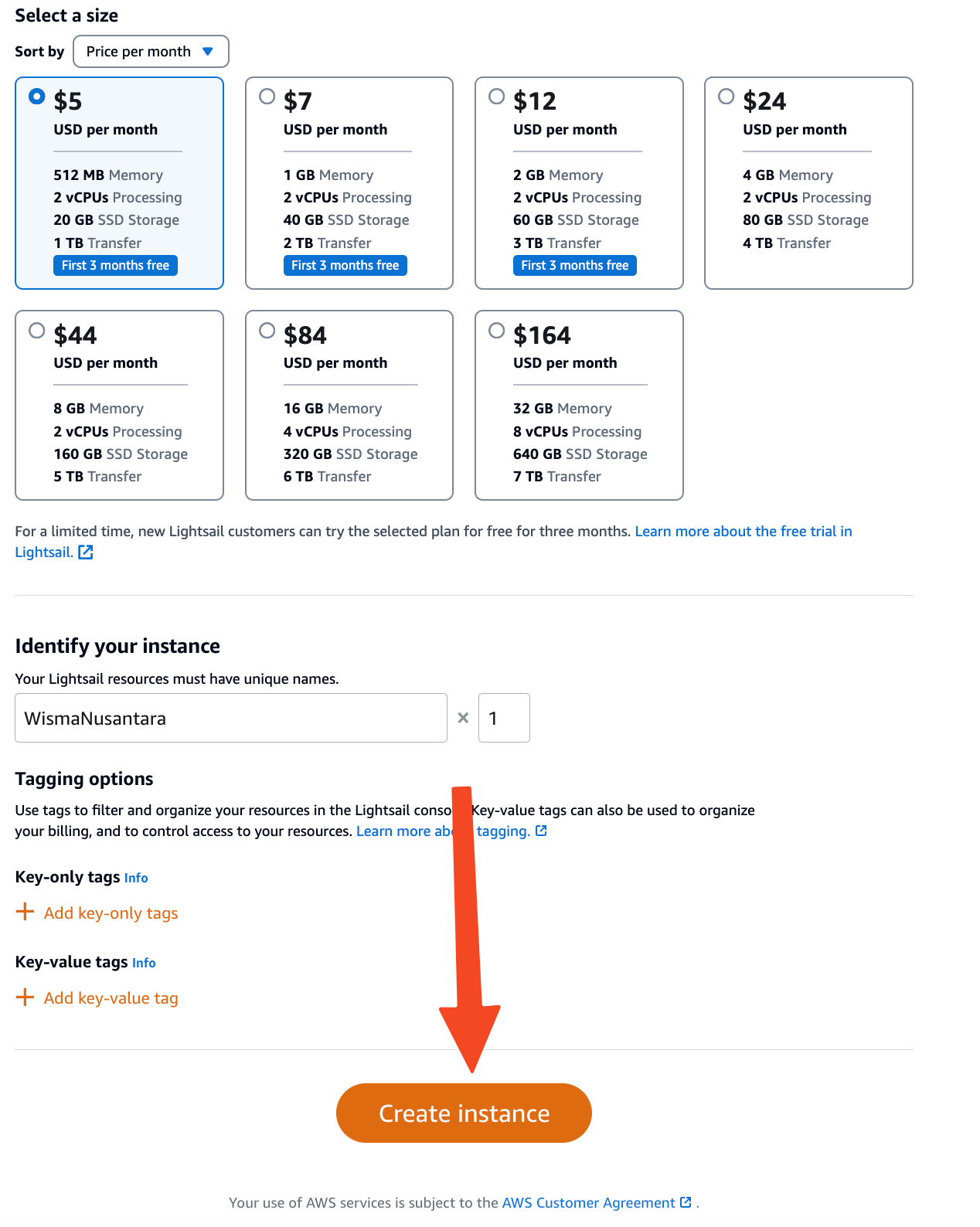
## Tahap ini adalah proses untuk menuntukan platform yang akan digunakan oleh user dan kebutuhan-kebutuhan sistem operasi yang akan dibutuhkan serta kebutuhan server, dengan rincian sebagai berikut:

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

## Lebih jelasnya terlihat seperti gambar dibawah ini.



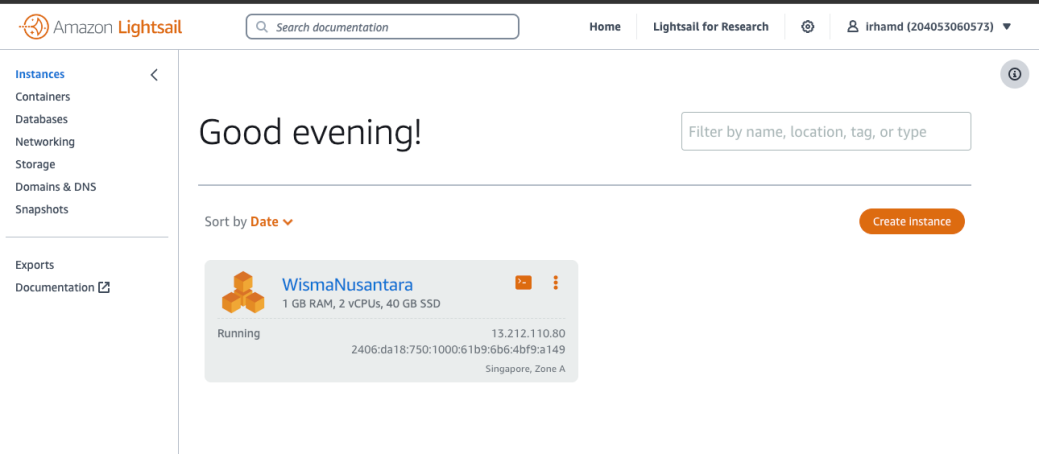
1. Menentukan *Platform*, Sistem Operasi dan *Plugin*



1. Menentukan Kebutuhan *Server*

## 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.



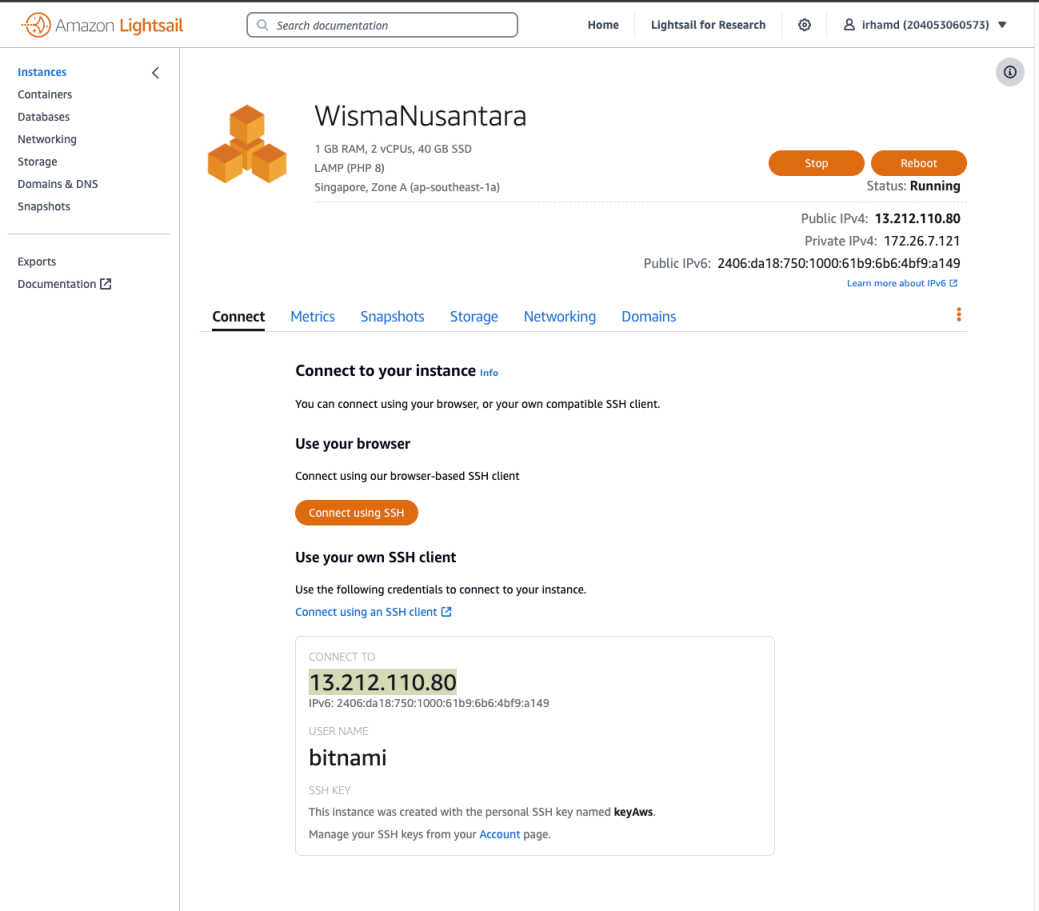
1. Hasil Instalasi Ligthsail AWS

## *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 |

1. Urian Penggunaan *Lightsail*



1. Uraian *Server Lightsail AWS*

## Instalasi *Lightsail AWS* Dengan Domain

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

## *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 menunjungi dan *Login* ke halaman <https://hpanel.hostinger.com/>, dan pilih domain wismanusantara.cloud yang nampak seperti gambar berikut ini.

## 

1. Domain wismanusantara.co.id

## Setting DNS Hostinger

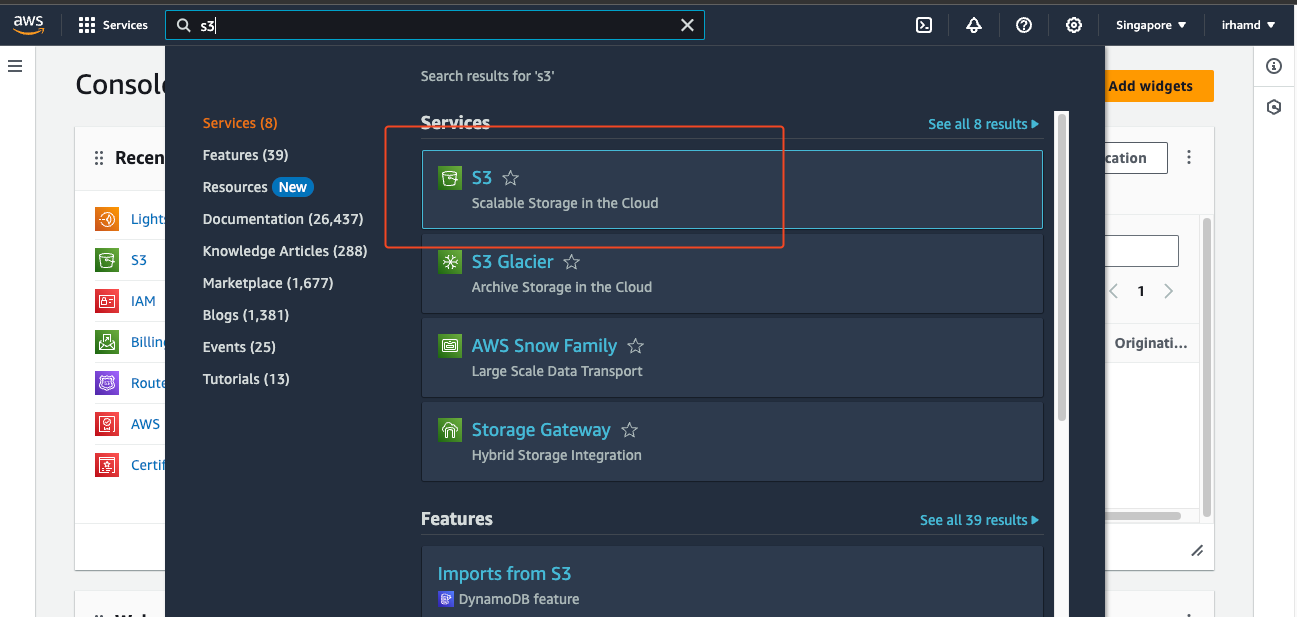
Langkah berikutnya adalah dengan melakukan pengaturan pada menu *DNS* pada *hostinger* yang bertujuan untuk menghubungkan antara *IP* Publik yang diterbitkan oleh *lightsail AWS* dengan domain yang telah disediakan oleh pihak *hostinger.com.*

## 

1. *Setting* *A Record Ip Public AWS Lightsail*

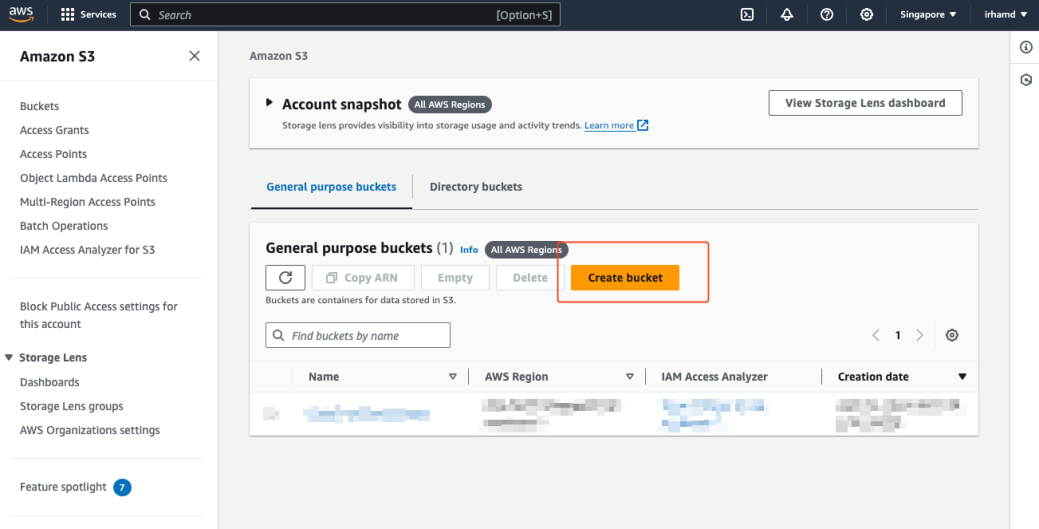
## Instalasi *S3 Bucket AWS*

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.

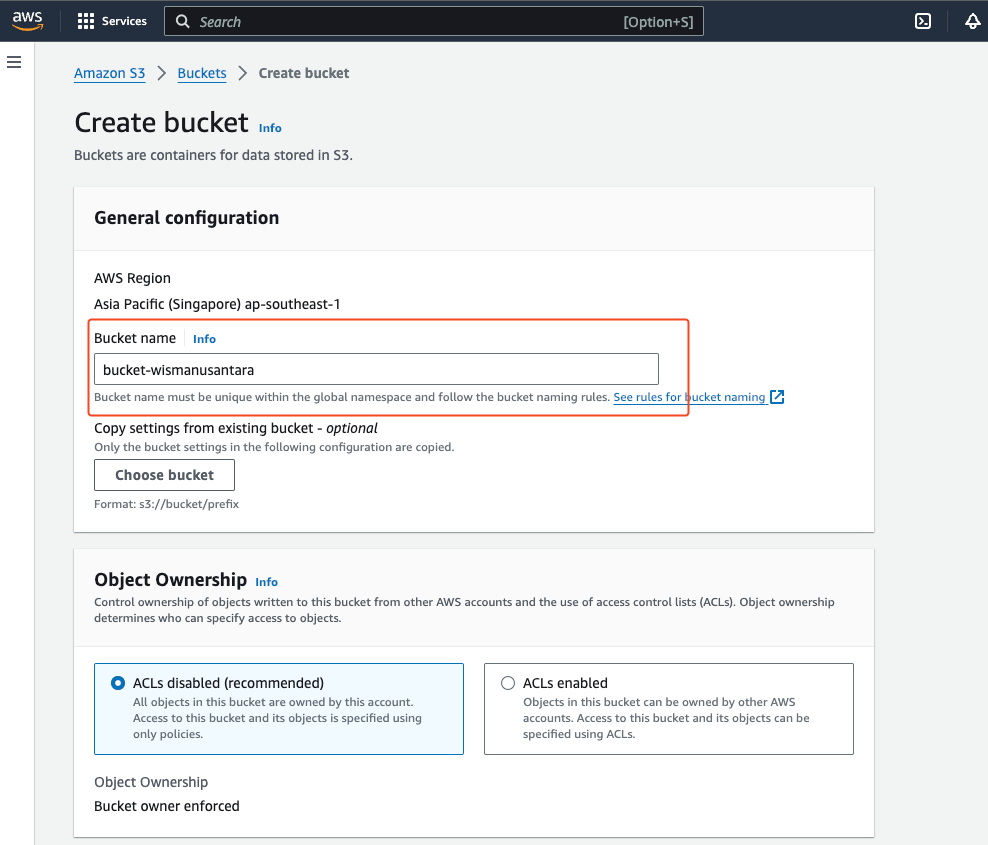


1. Instalasi *S3* *Bucket*

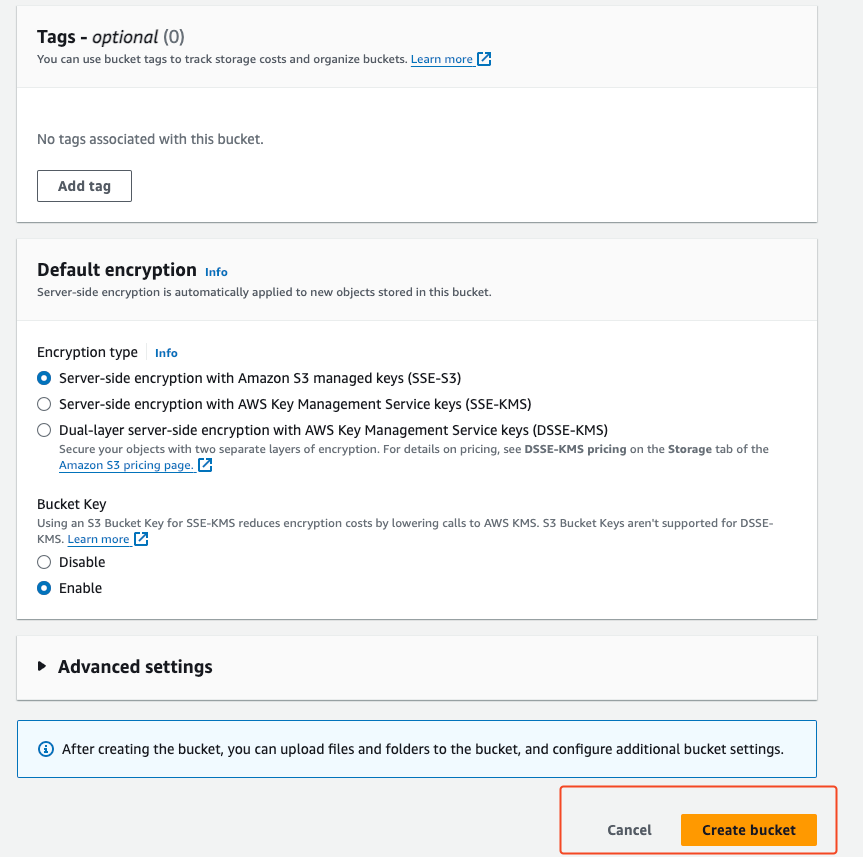
Setelah itu selanjutnya membuat *bucket* baru dengan memilih tombol “*Create Bucket*” tampak pada gambar berikut ini.



1. Buat Bucket Baru

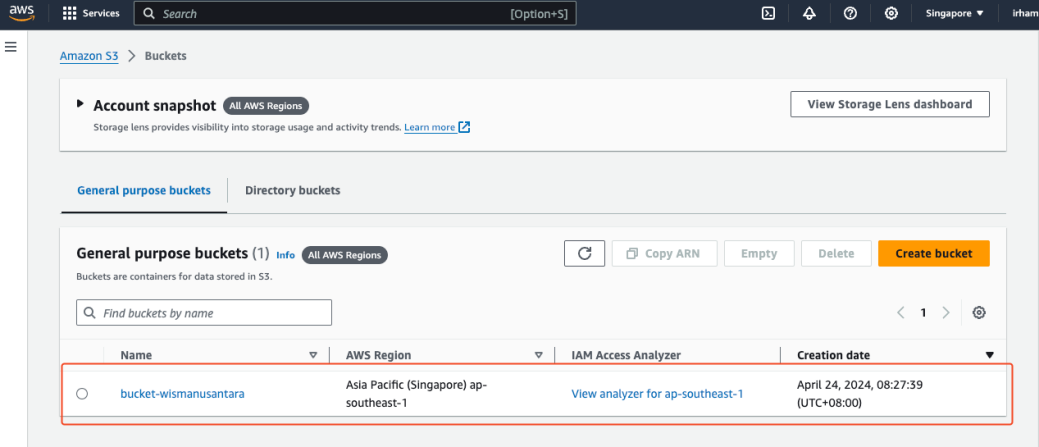


1. Menentukan nama bucket



1. *Create* bucket baru

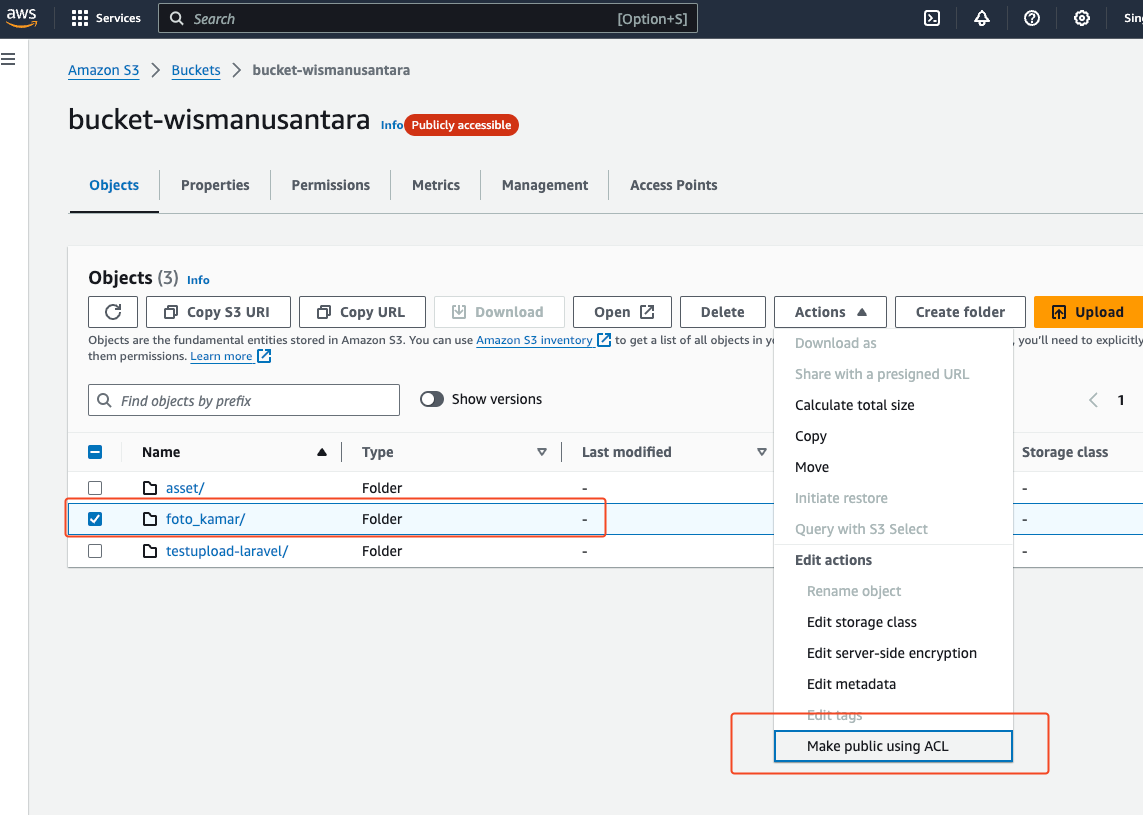
Setelah Anda membuat S3 bucket di *AWS*, langkah-langkah berikutnya bergantung pada apa yang ingin Anda lakukan dengan bucket tersebut, seperti mengunggah file ke bucket S3 menggunakan konsol AWS,atau mengatur izin akses bucket S3 agar sesuai dengan kebutuhan.



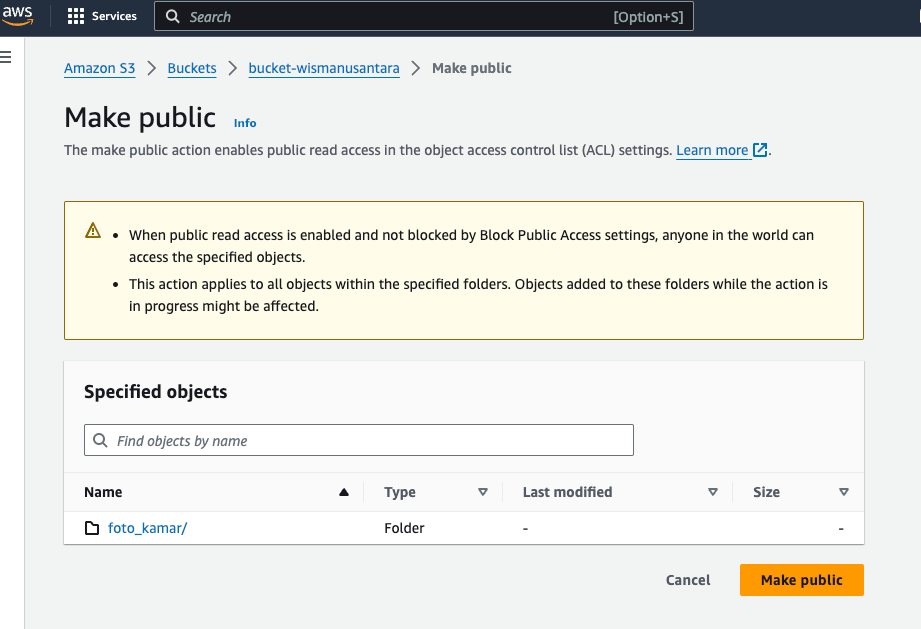
1. Hasil pembuatan bucket

## 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.



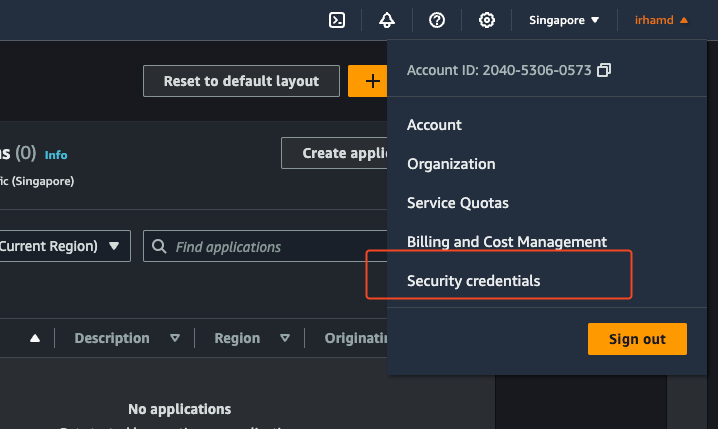
1. Memberikan izin akses folder



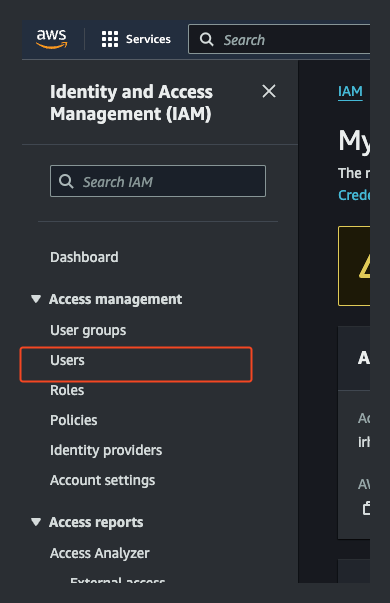
1. Hasil akses *public* folder

## Membuat User Kridensial Pada *AWS*

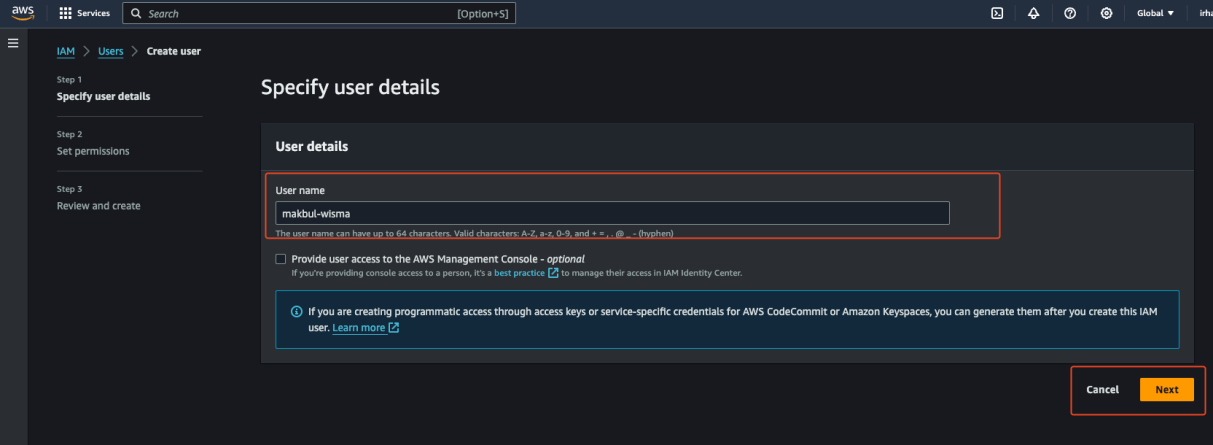
Tujuan membuat user kredensial pada AWS menggunakan layanan IAM adalah untuk mengelola akses ke layanan dan sumber daya AWS secara aman. Dengan membuat user kredensial, Anda dapat mengontrol akses ke layanan dan sumber daya AWS dengan lebih detail. Anda dapat menentukan izin yang spesifik untuk setiap 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*.



1. Langkah pertama *Install* *User* *security* *credentials*



1. Langkah kedua Install *User security credentials*



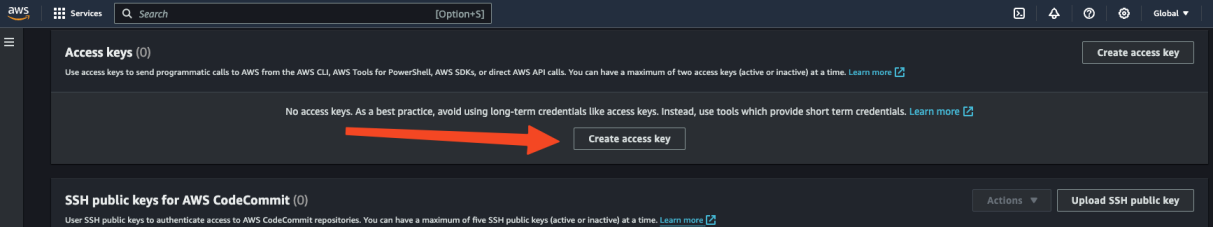
1. Menentukan nama user *User security credentials*



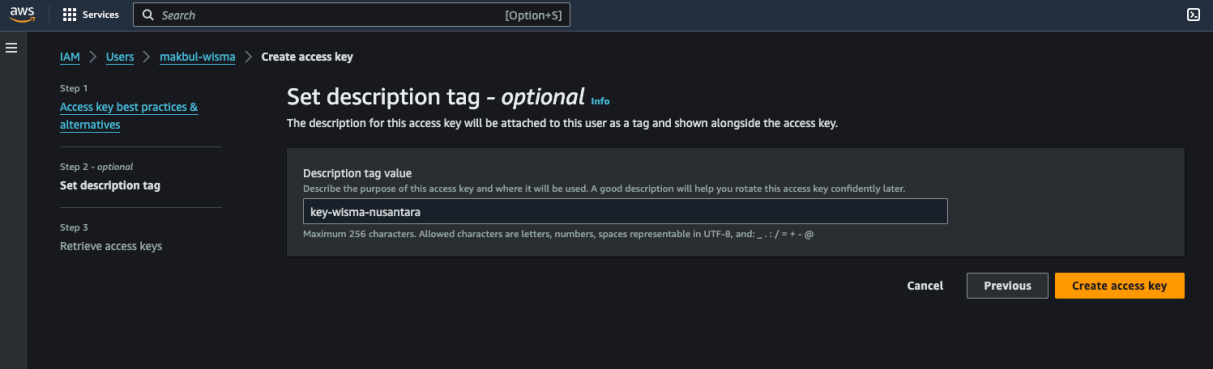
1. Hasil pembuatan *User security credentials*

## Membuat Access Key User

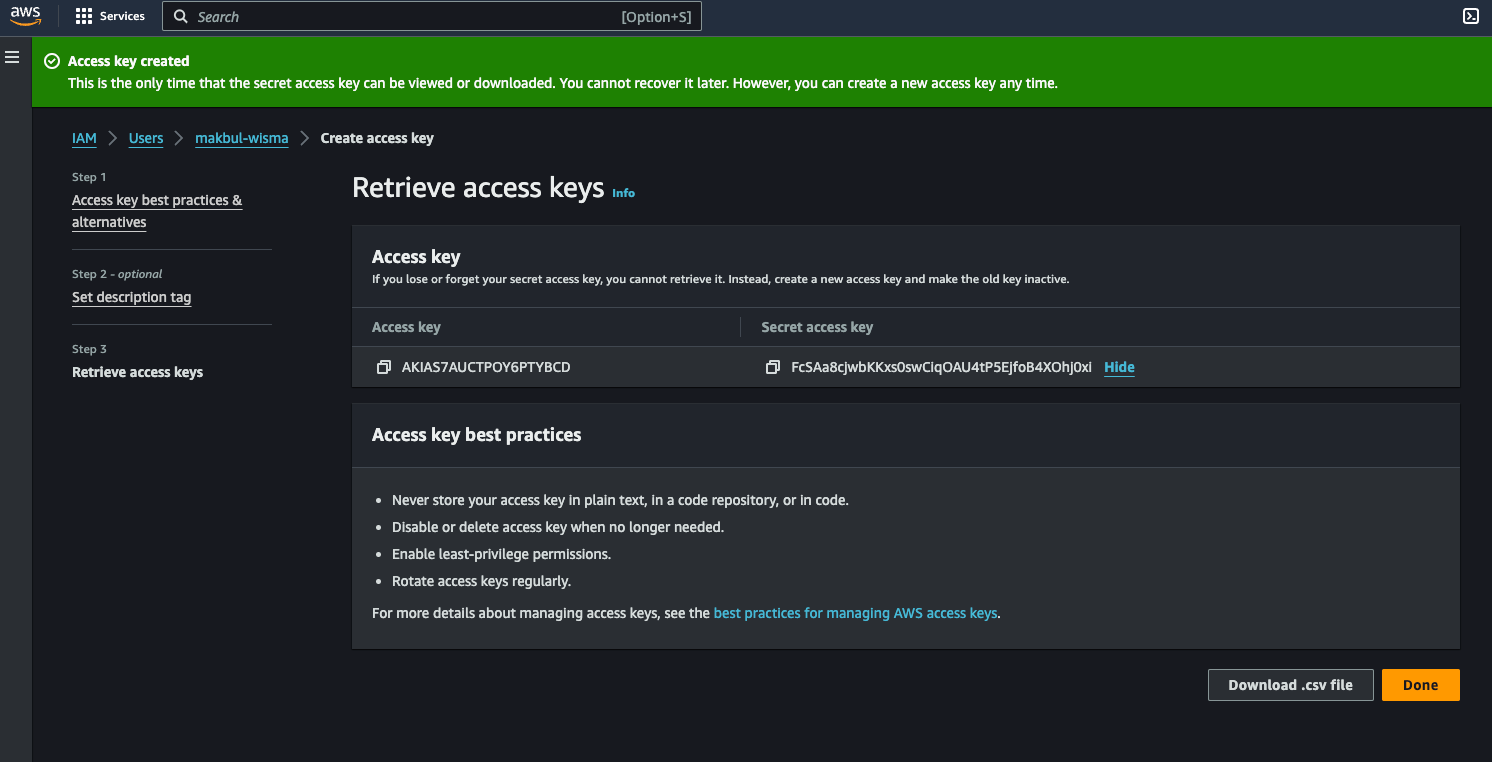
Tujuan utama membuat Access Key untuk user pada AWS adalah untuk memberikan cara aman bagi user tersebut untuk mengakses dan mengelola layanan AWS melalui *API* Laravel atau aplikasi lainnya.  *Qaw2q* digunakan untuk otentikasi user ketika mengakses layanan *AWS*. Dengan menggunakan Access Key, user dapat melakukan operasi sperti 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*.



1. *Create access key*



1. Deskripsi *access key*



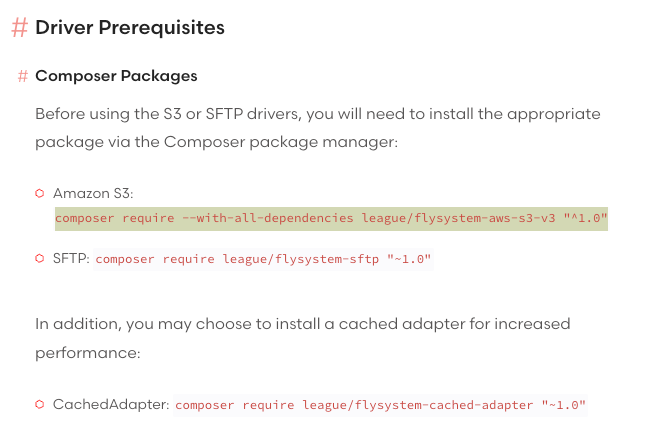
1. Hasil *create access key*

## Konfigurasi Dan Instalasi Package AWS S3 Bucket Pada Laravel

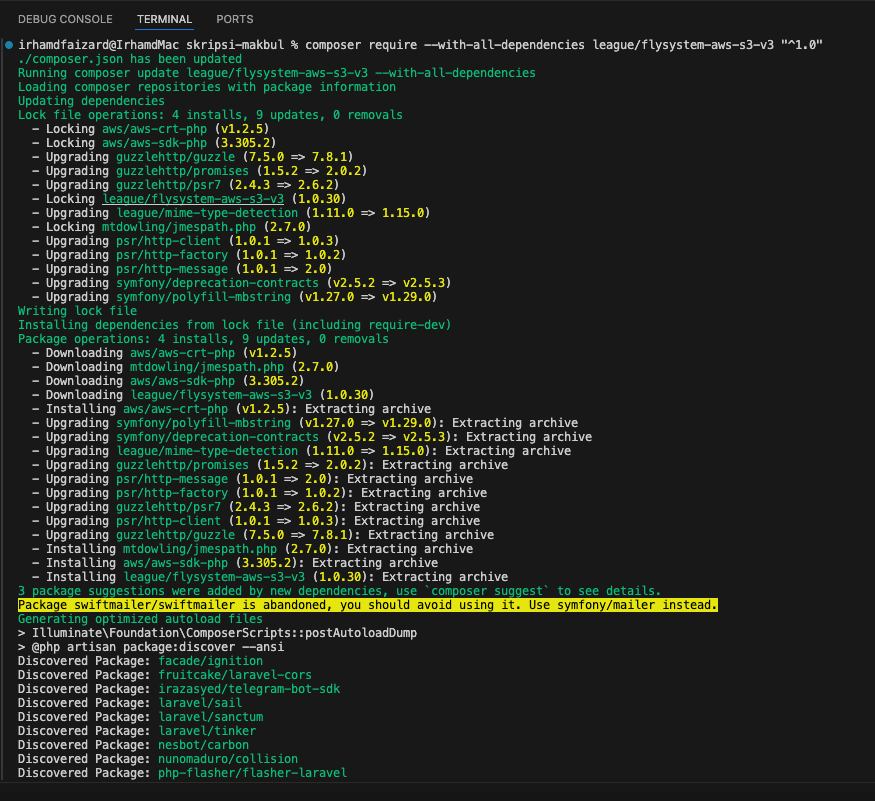
## Instalasi *Package AWS S3 Bucket*

## 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**"”



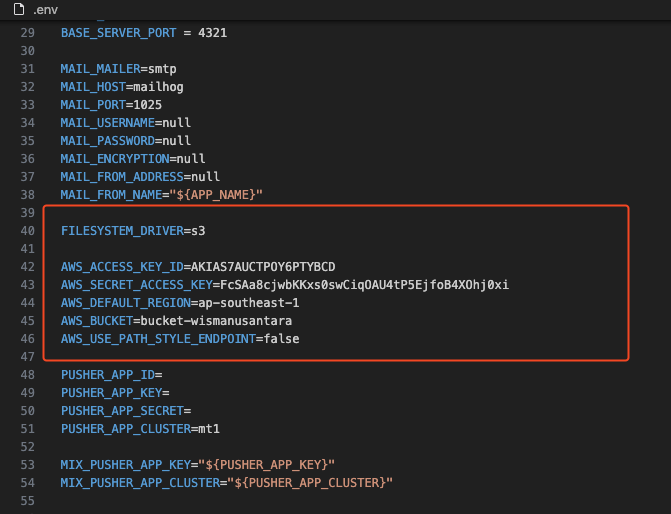
1. Instalasi Package AWS S3 Bucket Laravel



1. Instalasi *Package* S3 Bucket

## 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.

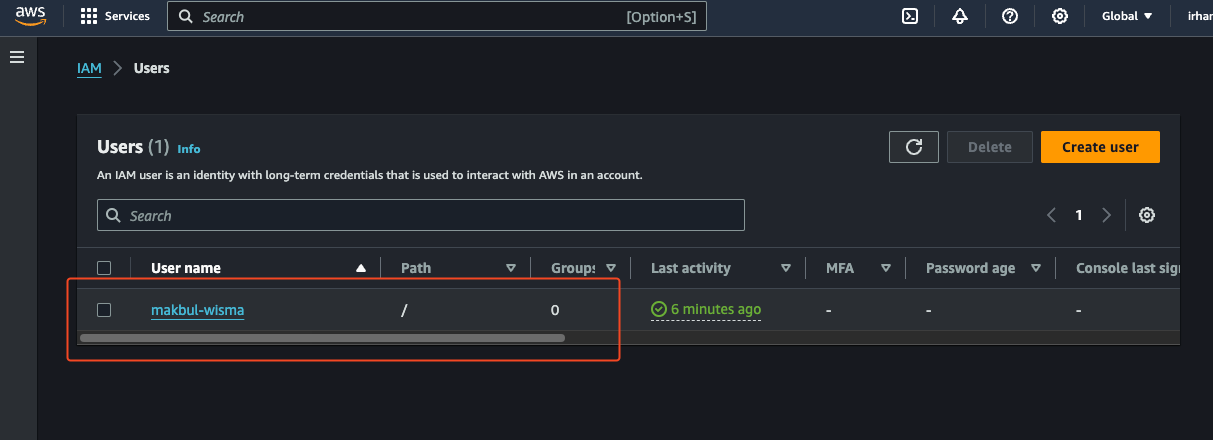


1. Hasil perubahan nilai *.env file* pada Laravel

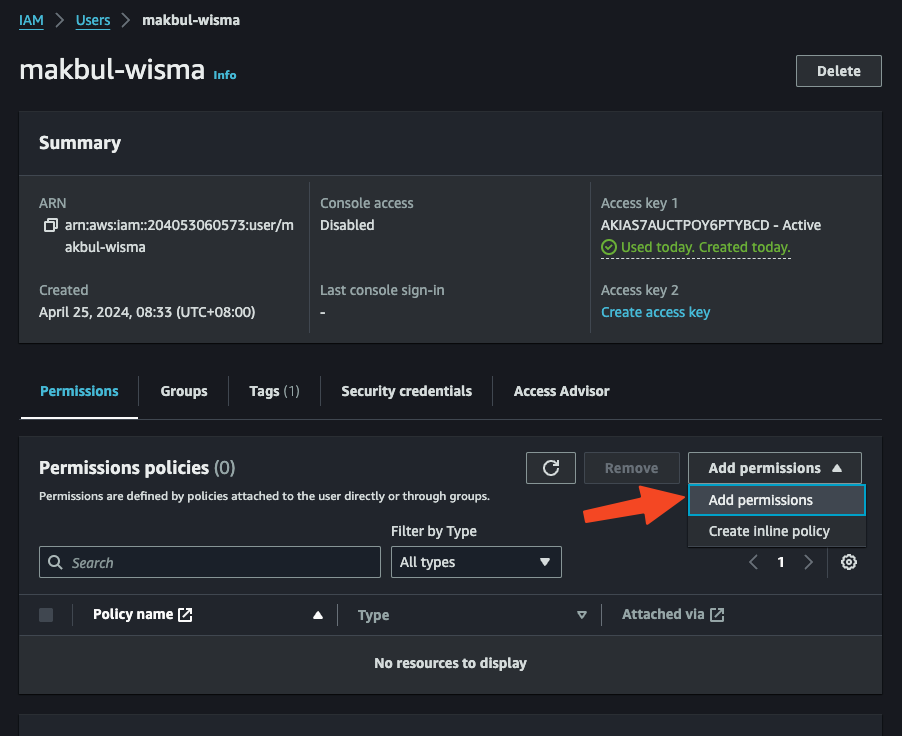
Dengan langkah-langkah di atas, sekarang telah terhubung AWS S3 Bucket dengan aplikasi Laravel Anda dan dapat mulai mengelola file di bucket tersebut melalui Laravel menggunakan Storage Facade dan pastikan untuk mengganti nilai kredensial dan konfigurasi dengan nilai-nilai yang sesuai dari akun *AWS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

## Setting *Permissions* policies AWS S3 Bucket

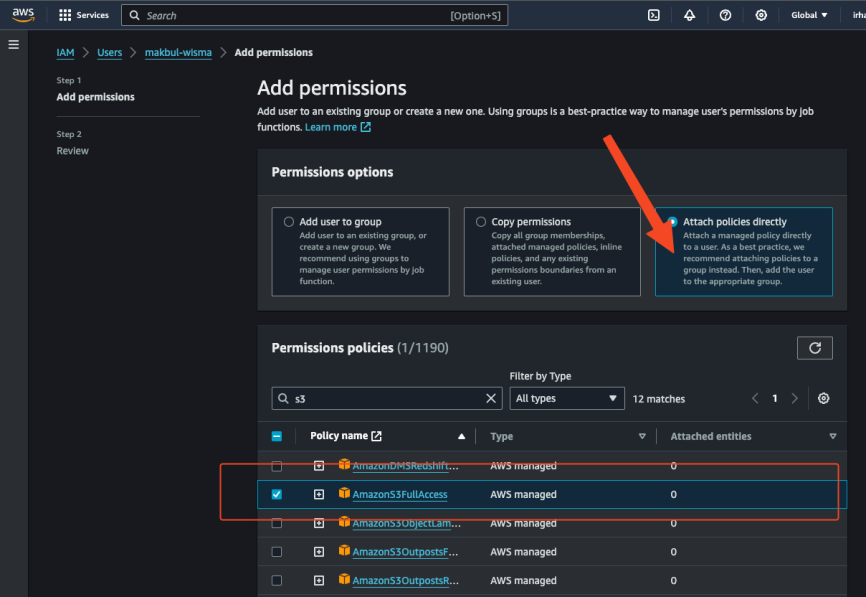
Dalam mengatur kebijakan izin untuk bucket AWS S3 sangat penting karena beberapa alasan seperti keamanan data, mencegah pelanggaran data, mencegah kehilangan data, atau optimisasi dumber daya. Dengan mengatur izin yang sesuai, dengan memastikan bahwa hanya pengguna atau layanan yang diotorisasi yang 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.



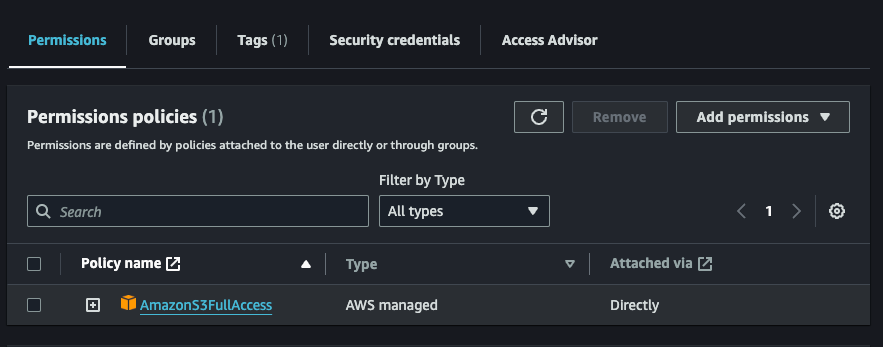
1. Pilih *username*



1. Add *permission policies*



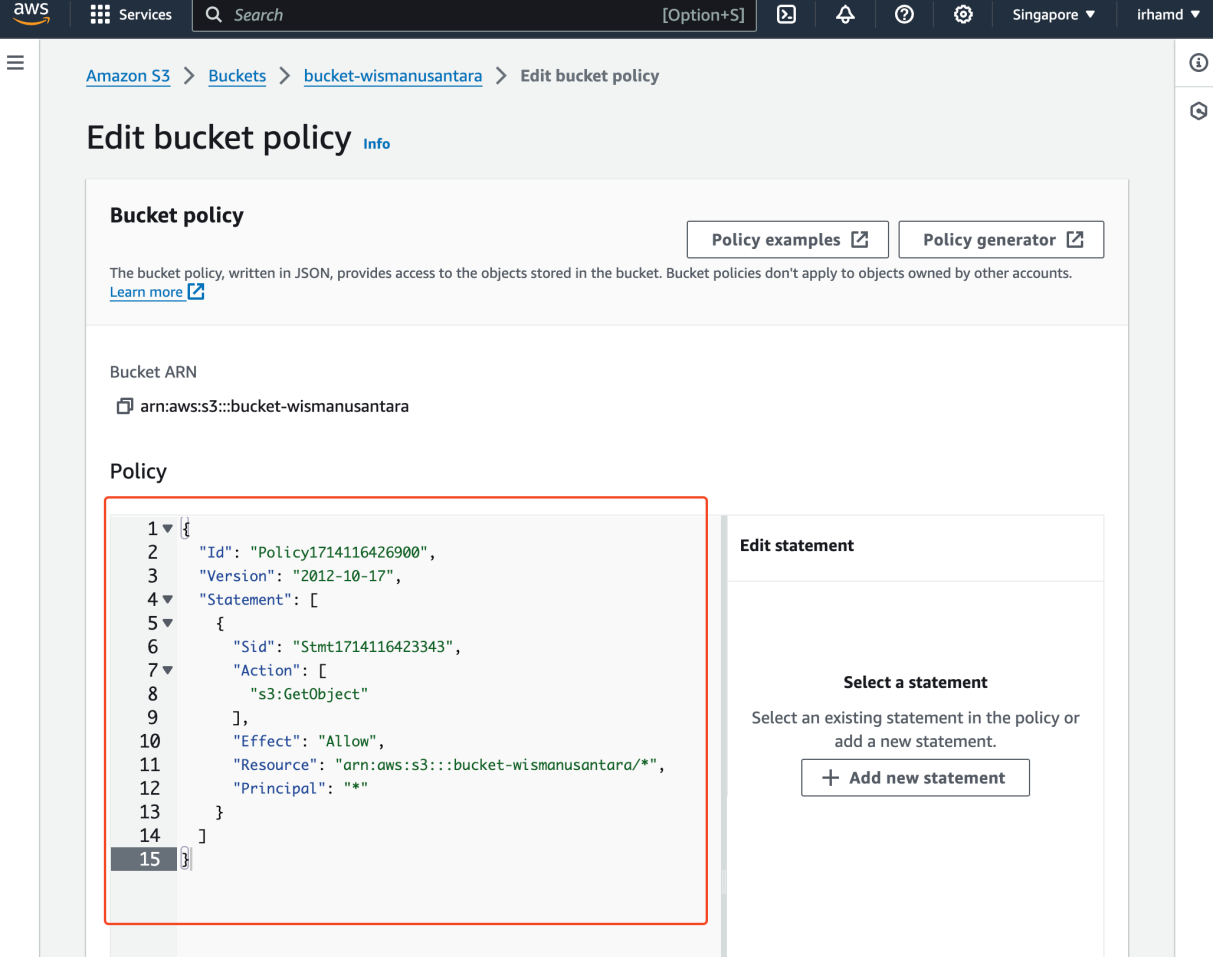
1. Pilih *policy name AmazonS3FullAccess*



1. Hasil pengaturan *policy permission*

## Berikan Akses *Bucket Policy S3 Bucket*

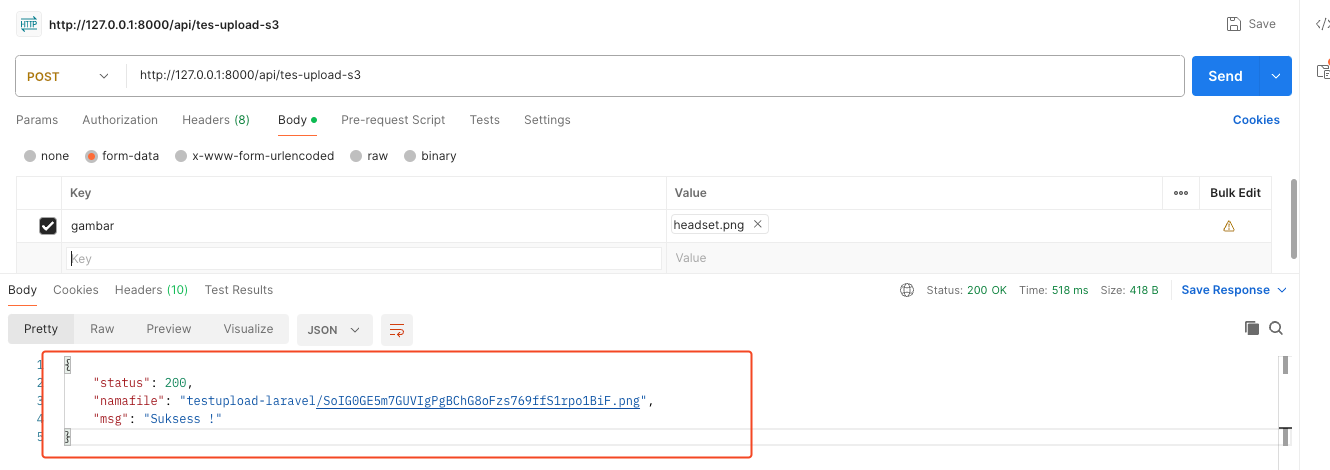
Kebijakan bucket S3 untuk membuat bucket dapat diakses oleh publik diperlukan terutama untuk situasi di mana hal ini diperlukan untuk membuat konten atau data yang disimpan dalam bucket dapat diakses dari proyek Laravel secara publik dari proyek laravel seperti gambar, atau file yang ingin didistribusikan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



1. Allow access *policy S3 bucket*

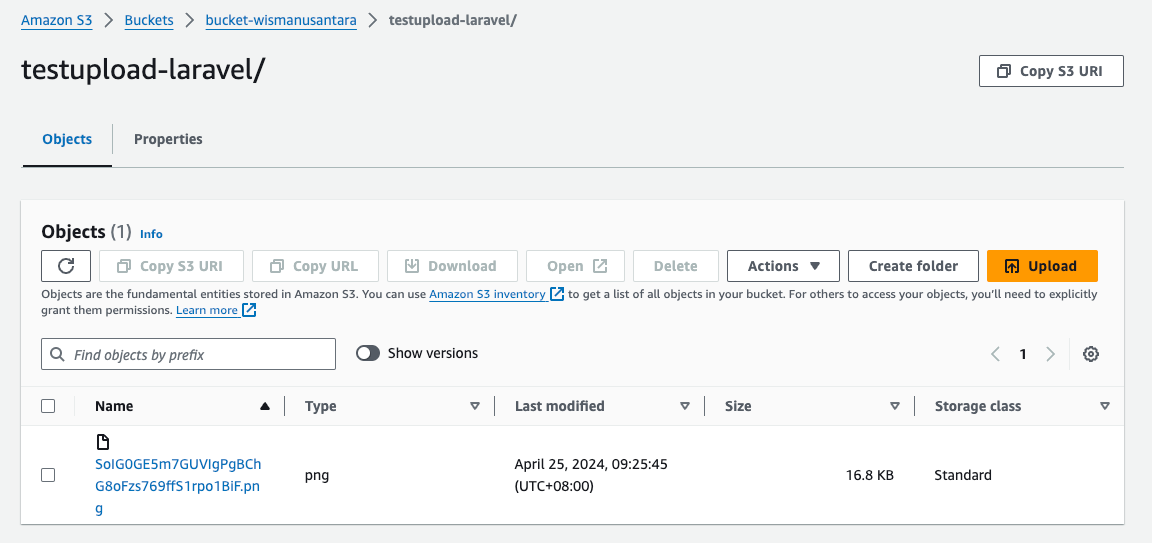
## Pengujian Upload File dari Laravel ke S3 Bucket AWS

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.



1. *Testing* upload *file* dari *Postmant*

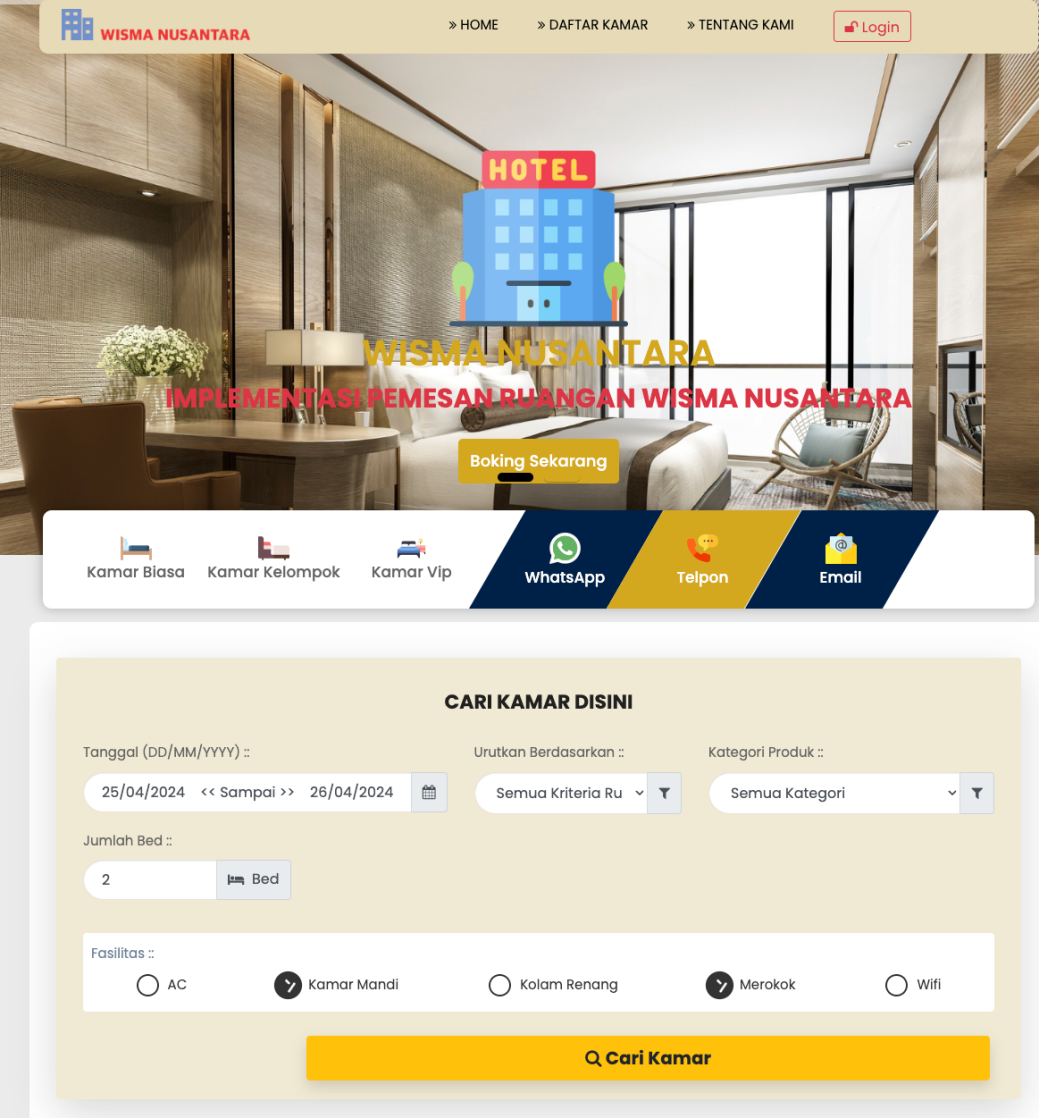
Uji coba *upload* *file* juga merupakan pengujian integrasi antara aplikasi Laravel dan layanan eksternal AWS S3 Bucket memastikan bahwa integrasi antara kedua platform berjalan dengan baik dan tidak ada masalah komunikasi atau keamanan yang muncul. Dengan melakukan uji coba upload file ke bucket AWS 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.



1. Hasil upload file dari postman ke S3 Bucket

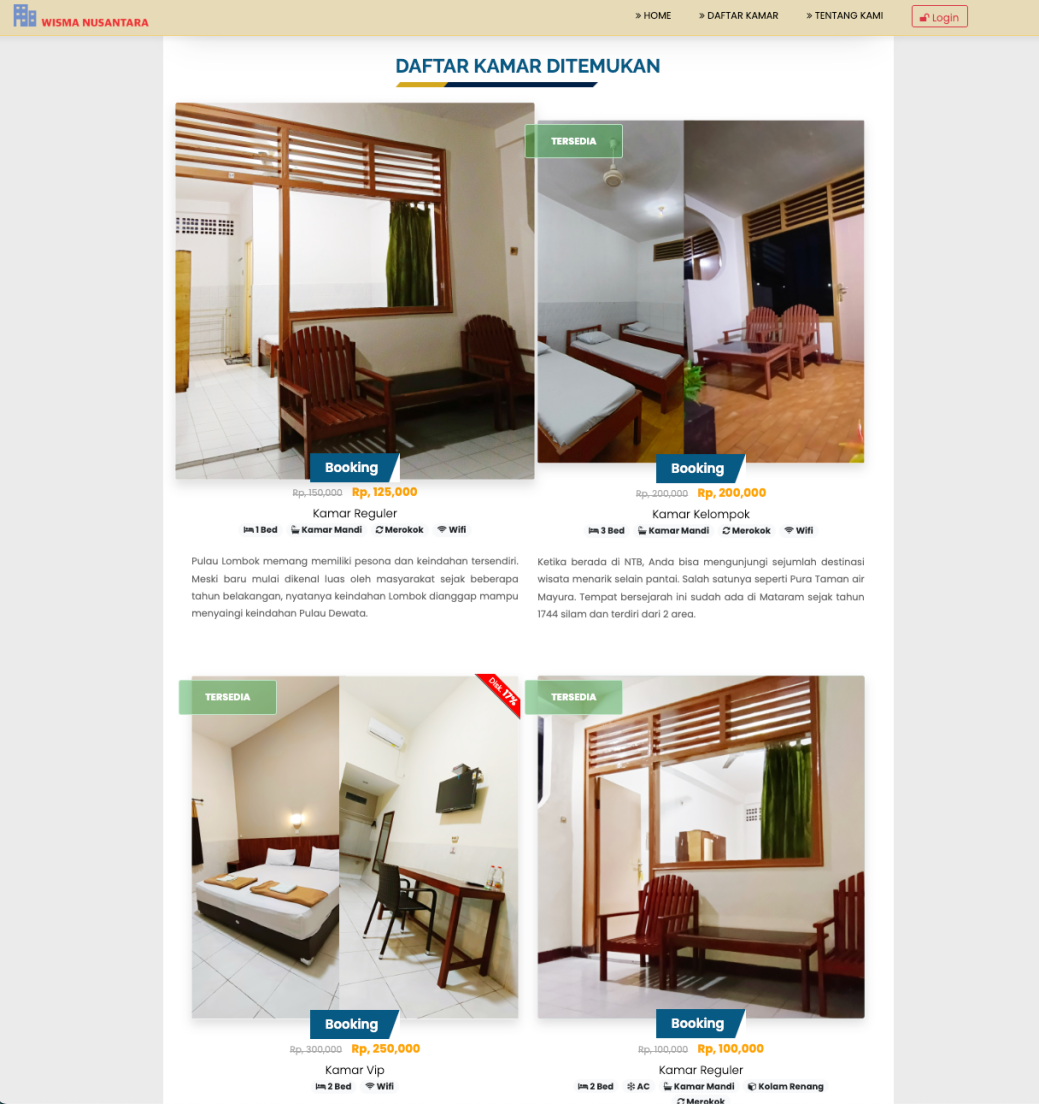
## Tampilan Antar Muka Aplikasi

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.



1. Form pencarian kamar

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



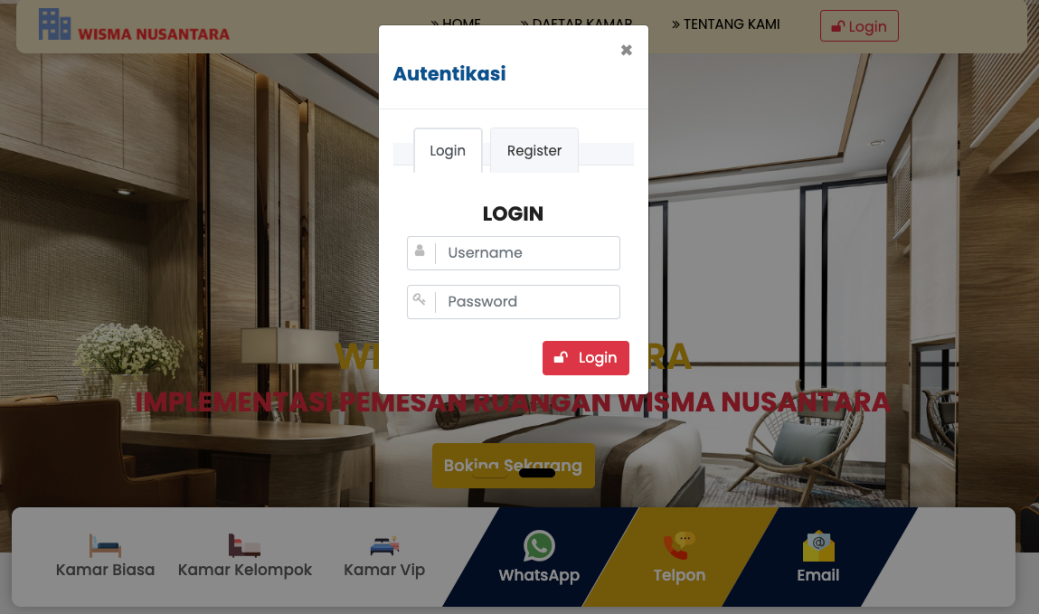
1. Hasil pencarian kamar

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”



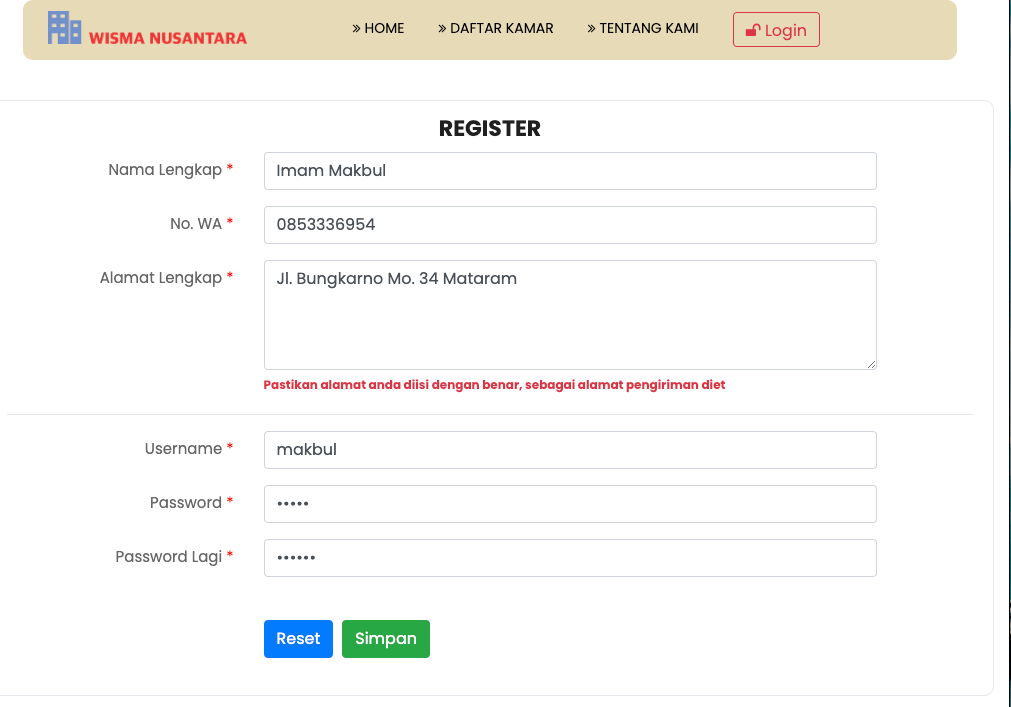
1. Konfirmasi form booking kamar

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.

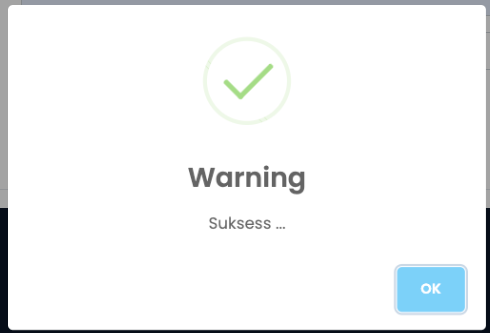


1. *UI popup screen* untuk *login*

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 authentikasi pada aplikasi. Setelah itu klik tombol “Simpan”.



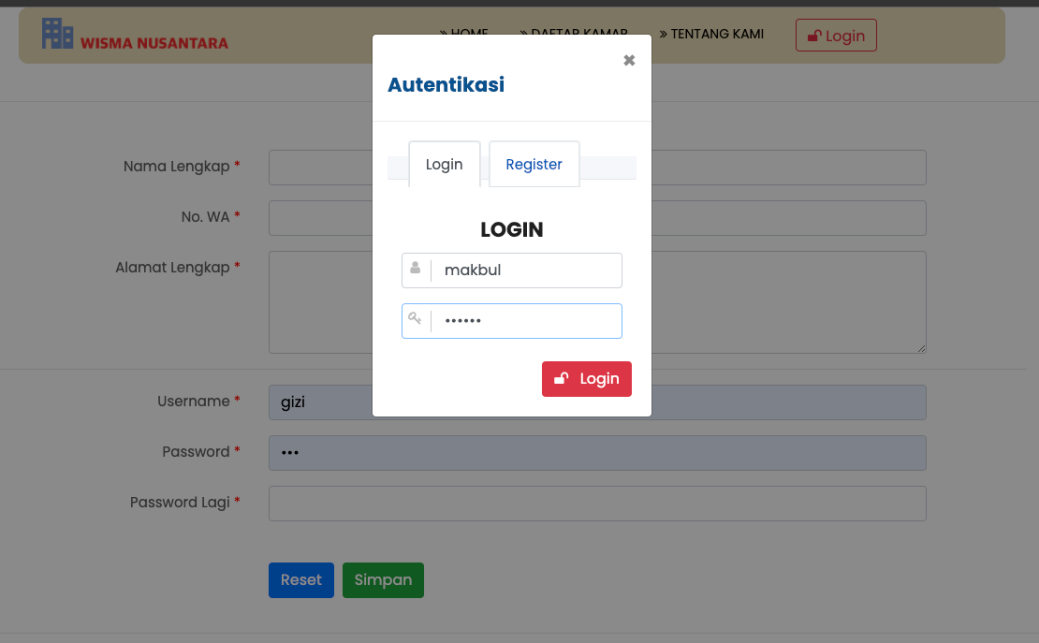
1. Form registrasi pengunjung



1. Informasi registrasi berhasil

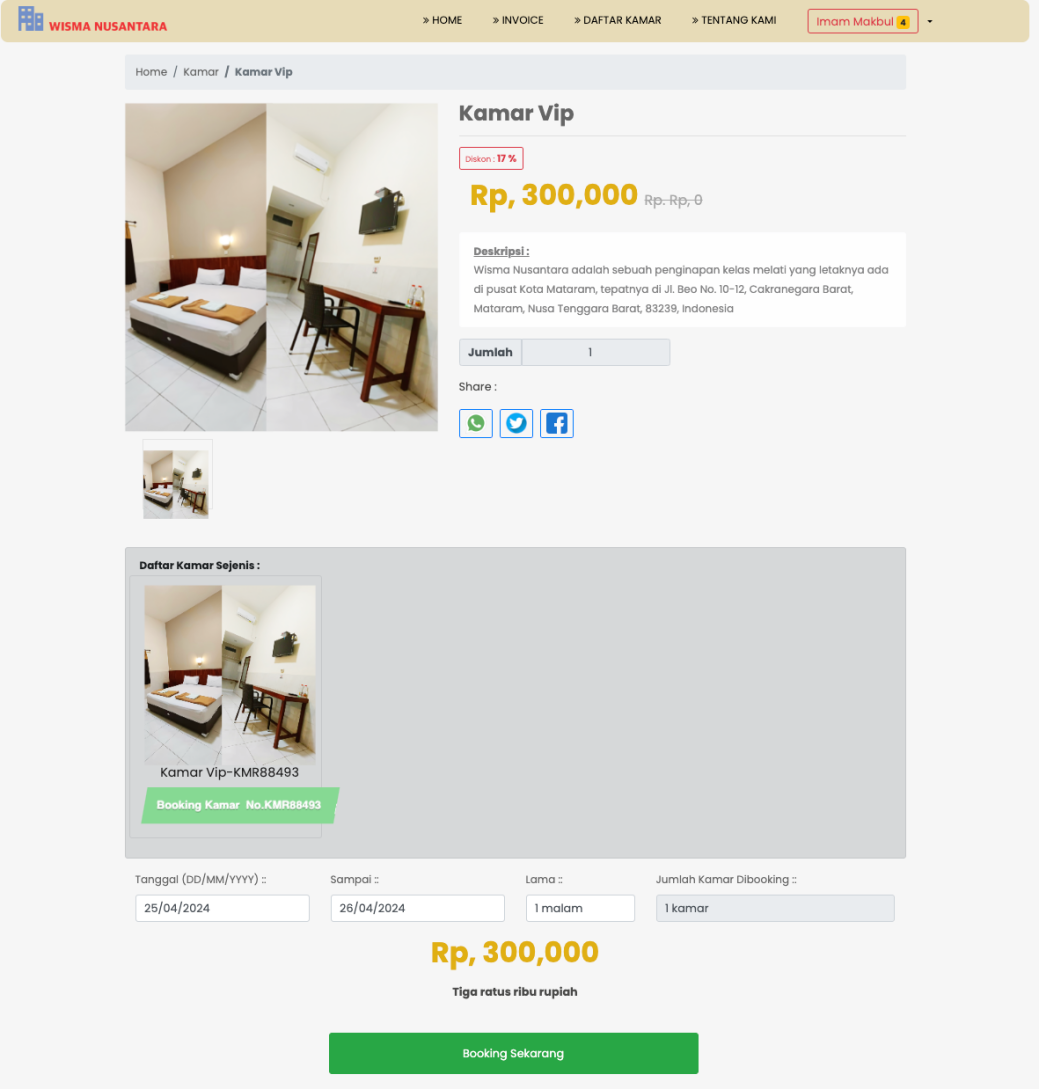
Untuk login ulang setelah berhasil registrasi pada aplikasi langkah selanjutnya adalah melakukan login ulang dengan mencari opsi "Login" atau "Masuk" di layar utama atau menu navigasi aplikasi klik atau klik opsi tersebut untuk membuka halaman login. Pada halaman login, masukkan kredensial yang Anda 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*.



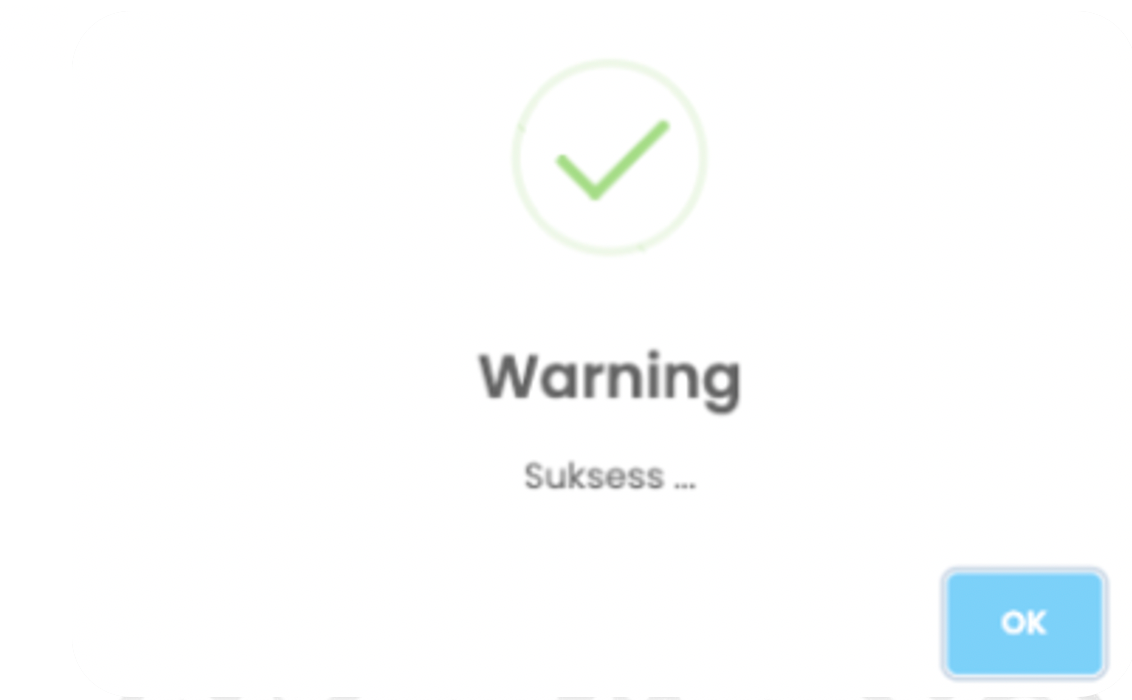
1. Menu login ke Aplikasi

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



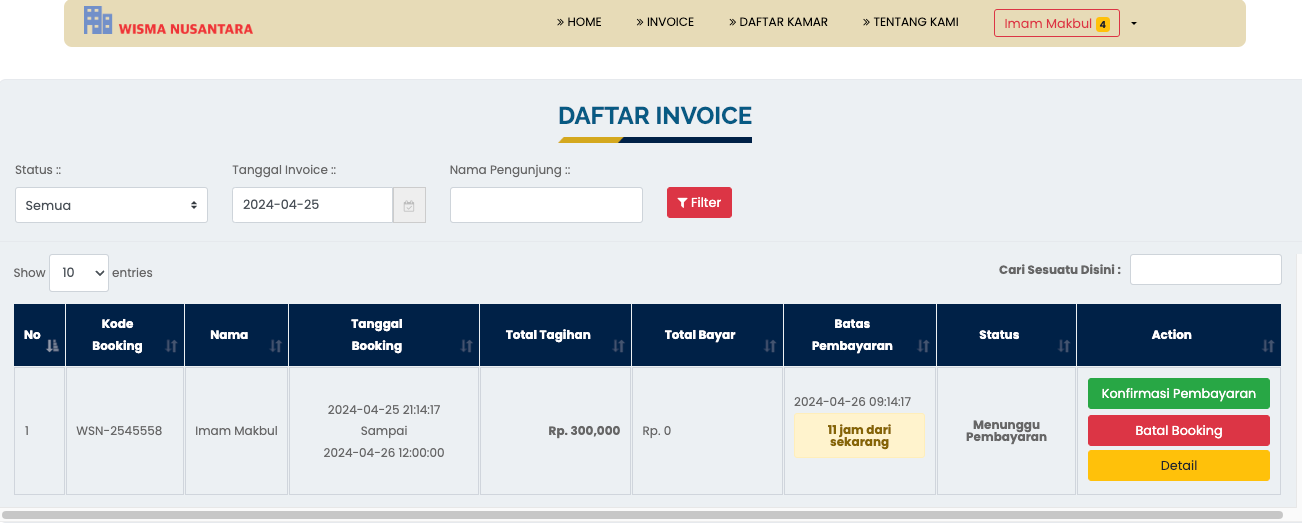
1. Menu Boking

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



1. 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 pengnung sebelum status boking didapat. Batas pembayaran dalam invoice ini akan diberikan jangka waktu 12 jam sebagai masa tenggang pembayaran sebelum tagihan tersebut kedaluarsa.



1. Daftar Invoice





## Tabel Hasil Pengujian Black-box

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

1. Pengujian *Black-Box*

|  |  |  |
| --- | --- | --- |
| **Pengujian** | **Jenis Antarmuka** | **Keterangan** |
| *Login* dan *Register* | * Tombol *Login* * Tombol Logout | Berfungsi |
| Akses Home | * Tanggal * Jumlah Bed * Fasilitas | Berfungsi |
| Cari Kamar | * Daftar Kamar | Berfungsi |
| Detail Kamar | * Data Kamar * Pilih Data Kamar | Berfungsi |
| Booking | * Booking Kamar | Berfungsi |
| Upload Bukti Transfer | * Upload Bukti Transfer Oleh Pelanggan | Berfungsi |
| Verifikasi Bukti Transfer | * Oleh Petugas Wisma Nusantara | Berfungsi |

## White-box Testing

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

## Tabel Hasil Pengujian White-box

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

1. Tabel 4. 2 Pengujian *White-box*

## 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 pemesan 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.

* + - 1. 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.

* + - 1. 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.

* + - 1. 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.

* + - 1. 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*.

* + - 1. 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.