

**LAPORAN PROYEK AKHIR  
PERANCANGAN DAN IMPLEMENTASI APLIKASI WEB  
SKINCARE “SAVIENNA”  
BERBASIS WEBSITE DAN REST API  
PADA MATA KULIAH PEMROGRAMAN WEB**

Disusun untuk memenuhi tugas Mata Kuliah Pemrograman Web

Dosen Pengampu : Arik Sofan Tohir, M.Kom



**Disusun Oleh :**

- |                                 |              |
|---------------------------------|--------------|
| 1. Davina Sava Anthea           | (2413030012) |
| 2. Nabila Fadilatul Puspita Ayu | (2413030015) |
| 3. Enjelika                     | (2413030101) |
| 4. Agvita Indah Anggraini       | (2413030059) |

**PROGRAM STUDI SISTEM INFORMASI  
FAKULTAS TEKNIK DAN ILMU KOMPUTER  
UNIVERSITAS NUSANTARA PGRI KEDIRI  
TAHUN 2025**

## KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan Laporan Proyek Akhir pada mata kuliah Pemrograman Web yang berjudul “Perancangan dan Implementasi Aplikasi Web Skincare ‘Savienna’ Berbasis Website dan REST API” dengan baik dan tepat waktu.

Laporan proyek akhir ini disusun sebagai bentuk dokumentasi dan pertanggungjawaban atas pelaksanaan tugas akhir semester pada mata kuliah Pemrograman Web. Proyek ini bertujuan untuk mengimplementasikan materi dan konsep pemrograman web yang telah dipelajari, meliputi perancangan tampilan website, pengelolaan basis data, serta pembuatan dan pengujian REST API sebagai penyedia data dalam format JSON.

Pada proyek ini, penulis membangun sebuah aplikasi web bertema skincare dengan nama Savienna, yang menampilkan data produk skincare melalui integrasi antara website, database MySQL, dan REST API. Seluruh proses perancangan, implementasi, serta hasil akhir sistem dijelaskan secara runtut dan sistematis dalam laporan ini.

Penulis menyadari bahwa laporan ini masih memiliki keterbatasan dan kekurangan. Oleh karena itu, penulis mengharapkan kritik dan saran yang bersifat membangun demi penyempurnaan laporan di masa mendatang. Akhir kata, penulis berharap laporan proyek akhir ini dapat memberikan manfaat dan menambah wawasan, khususnya dalam bidang pemrograman web berbasis website dan REST API.

Demikian laporan ini disusun. Atas perhatian dan bimbingan dari berbagai pihak, penulis mengucapkan terima kasih.

Kediri, 15 Desember 2025

Penulis

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Perkembangan teknologi informasi, khususnya di bidang pemrograman web, telah memberikan kemudahan dalam penyediaan dan pengelolaan informasi secara digital. Website tidak hanya berfungsi sebagai media informasi, tetapi juga sebagai sarana pengolahan data yang terintegrasi dengan database dan layanan Application Programming Interface (API).

Dalam industri skincare, penyajian informasi produk yang akurat dan terstruktur sangat penting untuk membantu pengguna memahami detail produk yang tersedia. Oleh karena itu, dibutuhkan sebuah aplikasi web yang mampu menampilkan data produk skincare secara dinamis, mudah diakses, serta terintegrasi dengan sistem penyimpanan data.

Melalui mata kuliah Pemrograman Web, mahasiswa dituntut untuk mampu menerapkan konsep pemrograman web ke dalam sebuah proyek nyata. Berdasarkan hal tersebut, penulis mengembangkan sebuah aplikasi web skincare bernama Savienna, yang memanfaatkan teknologi website, database MySQL, dan REST API sebagai penyedia data produk dalam format JSON.

### **1.2 Rumusan Masalah**

Rumusan masalah dalam proyek akhir ini adalah:

1. Bagaimana merancang dan membangun aplikasi web skincare Savienna berbasis website?
2. Bagaimana mengelola data produk skincare menggunakan database MySQL?
3. Bagaimana mengimplementasikan REST API untuk menampilkan data produk dalam format JSON?

### **1.3 Tujuan Proyek**

Tujuan dari proyek akhir ini adalah:

1. Membangun aplikasi web skincare berbasis website.
2. Mengimplementasikan database sebagai penyimpanan data produk.
3. Menerapkan REST API untuk penyediaan data produk skincare.

### **1.4 Manfaat Proyek**

Manfaat dari proyek ini antara lain:

1. Sebagai sarana penerapan ilmu pemrograman web.
2. Menambah pemahaman tentang integrasi website, database, dan REST API.
3. Menjadi referensi pengembangan aplikasi web sederhana berbasis API.

## **BAB II**

### **LANDASAN TEORI**

#### **A. Website**

Website adalah aplikasi berbasis web yang dapat diakses melalui browser dan digunakan sebagai media penyampaian informasi. Pada proyek ini, website digunakan sebagai antarmuka utama aplikasi skincare Savienna untuk menampilkan data produk skincare kepada pengguna. Website menampilkan informasi produk yang diambil secara dinamis dari database melalui REST API, sehingga data yang ditampilkan selalu sesuai dengan data yang tersimpan pada sistem.

Website Savienna dirancang menggunakan bahasa pemrograman PHP dan HTML, serta diakses melalui browser dengan bantuan web server.

#### **B. Pemrograman Web**

Pemrograman web merupakan proses pengembangan aplikasi berbasis web yang melibatkan pemrograman sisi server dan sisi klien. Pada proyek ini, pemrograman web diterapkan untuk membangun logika aplikasi, mengelola permintaan data, serta menampilkan informasi produk skincare ke halaman website.

Pemrograman web memungkinkan integrasi antara website dengan database dan REST API, sehingga aplikasi dapat berjalan secara dinamis dan interaktif.

#### **C. PHP**

PHP merupakan bahasa pemrograman server-side yang digunakan untuk mengelola proses pengambilan dan pengolahan data dari database. Dalam aplikasi Savienna, PHP digunakan untuk :

- Menghubungkan website dengan database MySQL
- Mengambil data produk skincare
- Membuat REST API yang menghasilkan data dalam format JSON

PHP dijalankan di sisi server sehingga pengguna hanya menerima hasil pemrosesan berupa tampilan website.

Database MySQL digunakan sebagai media penyimpanan data produk skincare. Data yang disimpan meliputi nama produk, harga, deskripsi, dan informasi lainnya. Database dirancang dalam bentuk tabel yang saling terstruktur untuk memudahkan proses penyimpanan dan pengambilan data.

Pada sistem Savienna, database MySQL diakses oleh PHP baik untuk keperluan pengelolaan data maupun penyediaan data melalui REST API.

#### **D. Application Programming Interface (API)**

Application Programming Interface (API) adalah antarmuka yang memungkinkan suatu aplikasi berkomunikasi dengan aplikasi lain. Pada proyek ini, API digunakan sebagai penghubung antara database dan website.

API Savienna bertugas untuk mengambil data produk dari database MySQL dan menyediakannya dalam format JSON, sehingga data dapat diakses dengan mudah melalui endpoint tertentu.

#### **E. REST API dan JSON**

REST API merupakan arsitektur API yang menggunakan protokol HTTP dalam pertukaran data. API pada aplikasi Savienna menerapkan konsep REST dengan metode HTTP, seperti GET untuk mengambil data produk.

Data yang dikirimkan oleh API disajikan dalam format JSON (JavaScript Object Notation), yang bersifat ringan dan mudah dibaca oleh sistem maupun manusia. Format JSON memungkinkan website Savienna menampilkan data produk secara dinamis berdasarkan data yang diperoleh dari API.

## **BAB III**

### **ANALISIS DAN PERANCANGAN SISTEM**

#### **A. Analisis Sistem Berjalan**

Pada tahap awal, dilakukan analisis terhadap sistem yang akan dibangun. Sistem yang dikembangkan merupakan aplikasi web skincare Savienna yang berfungsi untuk menampilkan data produk skincare kepada pengguna secara online.

Sistem bekerja dengan konsep client-server, di mana:

- **Client** berupa web browser yang digunakan pengguna
- **Server** menjalankan aplikasi web, database, dan REST API

Website tidak menyimpan data secara langsung, melainkan mengambil data produk dari database melalui REST API yang disediakan oleh server. Dengan demikian, data dapat dikelola secara terpusat dan ditampilkan secara dinamis.

#### **B. Analisis Kebutuhan Sistem**

- ❖ Kebutuhan Perangkat Keras
  - Komputer atau laptop sebagai server dan client
  - Perangkat input (keyboard, mouse)
  - Koneksi jaringan lokal atau internet
- ❖ Kebutuhan Perangkat Lunak
  - Sistem Operasi (Windows/Linux)
  - Web Server (Apache)
  - PHP sebagai bahasa pemrograman server-side
  - Database MySQL
  - Browser (Google Chrome / Mozilla Firefox)

#### **C. Analisis Kebutuhan Fungsional**

Sistem yang dikembangkan harus mampu:

1. Menampilkan daftar produk skincare Savienna.
2. Mengambil data produk dari database.
3. Menyediakan data produk dalam format JSON melalui REST API.
4. Menampilkan data API ke halaman website.
5. Menyajikan informasi produk secara dinamis.

#### **D. Analisis Kebutuhan Non-Fungsional**

Kebutuhan non-fungsional sistem meliputi:

- **Kemudahan penggunaan:** tampilan sederhana dan mudah dipahami
- **Kecepatan akses:** data ditampilkan secara real-time dari database
- **Keandalan sistem:** website dan API dapat diakses tanpa error

#### **E. Perancangan Arsitektur Sistem**

Arsitektur sistem Savienna terdiri dari tiga komponen utama:

1. Website Frontend Berfungsi sebagai antarmuka pengguna untuk menampilkan produk skincare.

2. Rest API Backend Bertugas mengambil data dari database dan mengirimkannya dalam format JSON.
3. Database mySQL Menyimpan seluruh data produk skincare.

Alur kerja sistem:

1. Pengguna membuka website Savienna melalui browser.
2. Website mengirim permintaan ke REST API.
3. REST API mengambil data dari database.
4. Data dikirim kembali ke website dalam format JSON.
5. Website menampilkan data produk kepada pengguna.

#### F. Perancangan Flowchart Sistem

Flowchart sistem secara umum adalah sebagai berikut:

1. Mulai
2. Website meminta data ke REST API
3. REST API memproses permintaan
4. Database mengirimkan data produk
5. REST API mengembalikan data dalam format JSON
6. Website menampilkan data produk
7. Selesai

#### G. Perancangan Struktur Database

Database dirancang untuk menyimpan data produk skincare Savienna secara terstruktur.

##### 3.1 Tabel Produk

Nama Field	Tipe Data	Keterangan
id_produk	Int	Primary Key
nama_produk	Varchar	Nama produk skincare
harga	Int	Harga produk
deskripsi	Text	Deskripsi produk
gambar	Varchar	Nama file gambar

#### H. Perancangan REST API

REST API dibuat menggunakan PHP dengan metode **GET**.

**Endpoint API:**

api\_produk.php

**Fungsi API:**

- Mengambil data produk dari database
- Mengubah data menjadi format JSON
- Mengirimkan data ke client (website)

#### I. Perancangan Antarmuka Website

Antarmuka website dirancang dengan memperhatikan:

- Tampilan produk yang rapi
- Informasi produk mudah dibaca
- Gambar produk ditampilkan dengan jelas

Website menampilkan data yang berasal langsung dari REST API sehingga perubahan data di database akan langsung terlihat di website.

#### **J. Diagram Alur Kerja Sistem**

Secara keseluruhan, sistem bekerja dengan alur:

- User → Website → REST API → Database → REST API → Website → User
- Diagram ini menunjukkan hubungan langsung antara website, API, dan database.



## **BAB IV**

### **IMPLEMENTASI DAN HASIL**

#### **4.1 Implementasi Sistem**

Implementasi sistem merupakan tahap penerapan hasil perancangan yang telah dijelaskan pada Bab III. Pada tahap ini, seluruh komponen sistem aplikasi web skincare Savienna diintegrasikan sehingga dapat berjalan sesuai dengan tujuan yang telah ditetapkan.

Sistem dibangun menggunakan bahasa pemrograman PHP, database MySQL, serta konsep REST API untuk pertukaran data antara website dan database. Proses implementasi dilakukan secara bertahap, mulai dari pembuatan database, pengembangan REST API, hingga pembuatan tampilan website.

#### **4.2 Implementasi Database**

Database MySQL digunakan sebagai media penyimpanan data produk skincare Savienna. Database dirancang sesuai dengan kebutuhan sistem, khususnya untuk menyimpan informasi produk secara terstruktur.

Tabel produk berisi data seperti:

- ID produk
- Nama produk
- Harga produk
- Deskripsi produk
- Gambar produk

Dengan adanya database ini, seluruh data produk dapat dikelola secara terpusat dan dapat diakses oleh REST API untuk ditampilkan pada website.

**Hasil implementasi database** menunjukkan bahwa data produk berhasil disimpan dan ditampilkan dengan baik melalui sistem manajemen database.

#### **4.3 Implementasi REST API**

REST API merupakan komponen utama yang menghubungkan database dengan website. REST API dibuat menggunakan PHP dan berfungsi untuk mengambil data produk dari database dan mengembalikannya dalam format JSON.

Endpoint API yang digunakan adalah:

api\_produk.php

REST API menggunakan metode GET untuk mengambil seluruh data produk skincare. Saat endpoint diakses melalui browser, sistem akan menampilkan data dalam format JSON yang berisi seluruh informasi produk.

**Hasil implementasi API** menunjukkan bahwa data berhasil diambil dari database dan ditampilkan dalam format JSON tanpa error.

#### **4.4 Implementasi Halaman Website**

Website Savienna dibuat sebagai antarmuka pengguna yang menampilkan data produk skincare kepada pengguna. Website mengambil data dari REST API

secara dinamis, sehingga data yang ditampilkan selalu sesuai dengan data di database.

Pada halaman website:

- Produk ditampilkan dalam bentuk daftar
- Setiap produk menampilkan nama, harga, deskripsi, dan gambar
- Tampilan dibuat sederhana dan mudah dipahami

Implementasi ini memastikan bahwa website tidak bergantung langsung pada database, melainkan menggunakan REST API sebagai penghubung.

#### **4.5 Integrasi Website dengan REST API**

Integrasi dilakukan dengan cara memanggil endpoint REST API dari halaman website menggunakan PHP. Data JSON yang diterima kemudian diolah dan ditampilkan dalam bentuk elemen HTML.

Proses integrasi berjalan sebagai berikut:

1. Website mengirim permintaan ke REST API
2. REST API mengambil data dari database
3. Data dikirim kembali dalam format JSON
4. Website memproses data JSON
5. Data ditampilkan ke halaman website

**Hasil integrasi** menunjukkan bahwa data produk dapat ditampilkan secara otomatis tanpa perlu input manual pada halaman website.

#### **4.6 Pengujian Sistem**

Pengujian sistem dilakukan untuk memastikan bahwa seluruh fungsi berjalan sesuai dengan yang diharapkan. Pengujian meliputi:

##### **4.6.1 Pengujian REST API**

- API diakses melalui browser
- Data ditampilkan dalam format JSON
- Tidak terjadi error saat pengambilan data

##### **4.6.2 Pengujian Website**

- Website berhasil memuat data produk
- Tampilan produk sesuai dengan data pada database
- Website dapat diakses dengan baik melalui browser

Hasil pengujian menunjukkan bahwa sistem berjalan dengan baik dan sesuai dengan tujuan pengembangan.

#### **4.7 Hasil Implementasi Sistem**

Berdasarkan hasil implementasi dan pengujian, dapat disimpulkan bahwa:

1. Sistem website skincare Savienna berhasil dibangun.
2. REST API berfungsi dengan baik sebagai penghubung antara database dan website.
3. Data produk ditampilkan secara dinamis dan real-time.
4. Sistem dapat digunakan sebagai media informasi produk skincare berbasis web.

#### **4.8 Dokumentasi Hasil Implementasi (Screenshot)**

Dokumentasi hasil implementasi ditampilkan dalam bentuk tangkapan layar (screenshot) yang menunjukkan:

- Struktur database dan tabel produk
- Tampilan hasil REST API dalam format JSON
- Tampilan halaman website Savienna

Dokumentasi ini menjadi bukti bahwa sistem telah berhasil diimplementasikan sesuai dengan perancangan yang telah dibuat.

## **BAB V**

### **PENUTUP**

#### **Kesimpulan**

Berdasarkan hasil perancangan, implementasi, dan pengujian sistem yang telah dibahas pada bab-bab sebelumnya, dapat disimpulkan bahwa aplikasi web skincare Savienna berbasis website dan REST API telah berhasil dibangun dan diimplementasikan sesuai dengan tujuan yang telah ditetapkan pada awal perancangan.

Aplikasi ini mampu menampilkan data produk skincare secara dinamis dengan memanfaatkan REST API sebagai penghubung antara database dan tampilan website. Seluruh data produk disimpan dalam database MySQL dan diakses melalui endpoint API yang menghasilkan output dalam format JSON. Dengan demikian, proses pengelolaan dan penyajian data menjadi lebih terstruktur, fleksibel, dan terpisah antara sisi backend dan frontend.

Implementasi REST API pada sistem ini membuktikan bahwa konsep layanan web dapat diterapkan secara nyata dalam pengembangan aplikasi web, sehingga website tidak perlu terhubung langsung ke database. Hal ini memberikan keuntungan dari segi keamanan, efisiensi, dan kemudahan pengembangan di masa mendatang.

Berdasarkan hasil pengujian yang telah dilakukan, seluruh fungsi sistem, mulai dari pengambilan data melalui API hingga penampilan data pada halaman website, berjalan dengan baik dan sesuai dengan yang diharapkan. Dengan demikian, sistem aplikasi web skincare Savienna dapat digunakan sebagai media informasi produk yang efektif dan dapat dikembangkan lebih lanjut.

#### **SARAN**

Sistem aplikasi web skincare Savienna yang telah dibangun masih dapat dikembangkan lebih lanjut, khususnya dari sisi keamanan dengan menambahkan mekanisme autentikasi pada REST API agar akses data lebih terkontrol. Selain itu, fitur API dapat diperluas untuk mendukung pengelolaan data secara lebih lengkap, serta tampilan website dapat ditingkatkan agar lebih menarik dan responsif. Ke depannya, sistem ini juga dapat dikembangkan ke platform lain seperti aplikasi mobile sehingga jangkauan pengguna menjadi lebih luas.

# DOKUMENTASI FOTO

## ➤ Tampilan Website

The image displays three screenshots of a web application for Savienna Skincare, showing the user interface for booking treatments.

**Top Screenshot: Main Menu**

The browser address bar shows `localhost/savienna/adminlte/`. The website header is dark red with the text "Savienna Skincare" and "Luxury Skin Care Experience". Below the header, a light pink banner contains the text: "Dapatkan kemudahan booking treatment, melihat pricelist treatment, menyampaikan keluhan & saran, serta pembelian produk Savienna Skincare." The main content area features four circular icons: "Booking Treatment" (calendar icon), "Pricelist Treatment" (diamond icon), "Keluhan & Saran" (document icon), and "Pembelian Produk" (shopping cart icon). The footer is dark red with the text "© 2023 Savienna Skincare. All Rights Reserved."

**Middle Screenshot: Booking Treatment Form**

The browser address bar shows `localhost/savienna/adminlte/booking.php`. The header is dark red with the text "Booking Treatment" and "Savienna Skincare". The main content area is titled "Form Booking Treatment" and contains the following fields:

- Nama Lengkap:
- No. WhatsApp:
- Pilih Treatment:
- Tanggal Booking:
- Jam Booking:

At the bottom of the form are two buttons: "Booking Sekarang" (dark red) and "Kembali ke Menu Utama" (light gray).

**Bottom Screenshot: Pricelist Treatment**

The browser address bar shows `localhost/savienna/adminlte/pricelist.php`. The header is dark red with the text "Pricelist Treatment" and "Savienna Skincare". The main content area lists four treatment options, each with a description, price, and a "Booking Sekarang" button:

- Facial Basic**  
Perawatan dasar untuk membersihkan dan menjaga kelembapan kulit wajah.  
Rp 180.000  
[Booking Sekarang](#)
- Facial Acne**  
Perawatan intensif untuk kulit berjerawat dan berminyak.  
Rp 230.000  
[Booking Sekarang](#)
- Facial Brightening**  
Membantu mencerahkan dan meratakan warna kulit wajah.  
Rp 270.000  
[Booking Sekarang](#)
- Laser Treatment**  
Perawatan teknologi laser untuk flek, pori, dan regenerasi kulit.  
Rp 520.000  
[Booking Sekarang](#)

The image displays two screenshots of a web application interface for 'Savienna Skincare'.

**Top Screenshot: Keluhan & Saran (Complaints & Suggestions)**

- Header:** Keluhan & Saran, Savienna Skincare
- Form Title:** Form Keluhan & Saran
- Fields:**
  - Nama (Name): Nama Anda
  - No. WhatsApp: 080000000000
  - Pesan / Keluhan / Saran (Message / Complaint / Suggestion): Tulisakan keluhan atau saran Anda
- Buttons:** Kirim Pesan (Send Message), — Kembali ke Menu Utama (Back to Main Menu)
- Footer:** © 2025 Savienna Skincare. All Rights Reserved.

**Bottom Screenshot: Pilih Produk (Choose Product)**

- Header:** Pilih Produk
- Products:**

Product Name	Price	Action
Facial Wash	Rp 85.000	Pilih
Toner Brightening	Rp 95.000	Pilih
Serum Acne	Rp 120.000	Pilih
Serum Brightening	Rp 135.000	Pilih
Day Cream	Rp 110.000	Pilih
Night Cream	Rp 115.000	Pilih
- Buttons:** Tambah ke Keranjang (Add to Cart)
- Footer:** Keranjang

The screenshot shows the phpMyAdmin interface. On the left, the database structure is visible, including 'db\_savienna'. The main panel shows the 'Structure' tab for 'db\_savienna'. A table list is displayed with columns: Table, Action, Rows, Type, Collation, Size, and Overhead. The table 'treatment' is highlighted. Below the table list, there is a 'Check all' checkbox and a 'With selected:' dropdown menu. At the bottom, there is a 'Create new table' button and a 'Console' tab.

localhost/phpmyadmin/index.php?route=/sql&db=db\_savienna&table=admin&pos=0

Showing rows 0 - 0 (1 total. Query took 0.0005 seconds.)

```
SELECT * FROM `admin`
```

Extra options

	id	username	password
<input type="checkbox"/>	1	admin	0192023a7bbd73250516069d18b500

Query results operations

Print Copy to clipboard Export Display chart Create view

Bookmark this SQL query

localhost/phpmyadmin/index.php?route=/table/structure&db=db\_savienna&table=admin

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	username	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
3	password	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	id	1	A	No	

Create an index on 1 column(s) after password Go

localhost/phpmyadmin/index.php?route=/sql&db=db\_savienna&table=booking&pos=0

Showing rows 0 - 0 (1 total. Query took 0.0004 seconds.)

```
SELECT * FROM `booking`
```

Extra options

	id	nama	no_wa	treatment	tanggal	jam	status	created_at
<input type="checkbox"/>	2	enjel	0897654321	Facial Basic	2025-12-30	16:37:00		2025-12-28 16:37:08

Query results operations

Print Copy to clipboard Export Display chart Create view

Bookmark this SQL query

localhost/phpmyadmin/index.php?route=/table/structure&db=db\_savienna&table=booking

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	nama	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
3	no_wa	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More
4	treatment	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
5	tanggal	date			No	None			Change Drop More
6	jam	time			No	None			Change Drop More
7	status	enum('Menunggu', 'Dikonfirmasi', 'Selesai')	utf8mb4_general_ci		Yes	Menunggu			Change Drop More
8	created_at	timestamp			No	current_timestamp()			Change Drop More

Indexes

Create an index on 1 column(s) after created\_at Go

localhost/phpmyadmin/index.php?route=/sql&pos=0&db=db\_savienna&table=produk

Showing rows 0 - 5 (6 total, Query took 0.0004 seconds.)

```
SELECT * FROM `produk`
```

Table: produk

	id	nama_produk	deskripsi	harga	stok	created_at
<input type="checkbox"/>	4	Savienna Brightening Toner	Toner pencerah wajah	95000	50	2025-12-28 19:20:17
<input type="checkbox"/>	5	Savienna Acne Serum	Serum wajah untuk jerawat	120000	40	2025-12-28 19:20:17
<input type="checkbox"/>	6	Savienna Brightening Serum	Serum pencerah wajah	135000	35	2025-12-28 19:20:17
<input type="checkbox"/>	7	Savienna Day Cream	Krim siang wajah	110000	45	2025-12-28 19:20:17
<input type="checkbox"/>	8	Savienna Night Cream	Krim malam wajah	115000	30	2025-12-28 19:20:17
<input type="checkbox"/>	9	Savienna Facial Wash	Sabun pembersih wajah	85000	60	2025-12-28 19:21:43

localhost/phpmyadmin/index.php?route=/table/structure&db=db\_savienna&table=produk

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	nama_produk	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
3	deskripsi	text	utf8mb4_general_ci		Yes	NULL			Change Drop More
4	harga	int(11)			No	None			Change Drop More
5	stok	int(11)			No	None			Change Drop More
6	created_at	timestamp			No	current_timestamp()			Change Drop More

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No		id	6	A	No	

localhost/phpmyadmin/index.php?route=/sql&pos=0&db=db\_savienna&table=transaksi

Showing rows 0 - 0 (1 total, Query took 0.0004 seconds.)

```
SELECT * FROM `transaksi`
```

Table: transaksi

	id	nama_pembeli	no_wa	id_produk	jumlah	total	tanggal
<input type="checkbox"/>	1	Vita	081234567890	4	1	95000	2025-12-28 19:49:52

localhost/phpmyadmin/index.php?route=/table/structure&db=db\_savienna&table=transaksi

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	nama_pembeli	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
3	no_wa	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More
4	id_produk	int(11)			No	None			Change Drop More
5	jumlah	int(11)			No	None			Change Drop More
6	total	int(11)			No	None			Change Drop More
7	tanggal	timestamp			No	current_timestamp()			Change Drop More

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No		id	1	A	No	



Showing rows 0 - 4 (5 total, Query took 0.0006 seconds)

```
SELECT * FROM `treatment`
```

Number of rows: 25 | Filter rows: Search this table | Sort by key: None

	id	nama_treatment	deskripsi	harga	created_at
<input type="checkbox"/>	6	Facial Basic	Perawatan dasar untuk membersihkan dan menjaga kel...	180000	2025-12-28 20:57:21
<input type="checkbox"/>	7	Facial Acne	Perawatan intensif untuk kulit berjerawat dan berm...	230000	2025-12-28 20:57:21
<input type="checkbox"/>	8	Facial Brightening	Membantu mencerahkan dan meratakan warna kulit waj...	270000	2025-12-28 20:57:21
<input type="checkbox"/>	9	Laser Treatment	Perawatan teknologi laser untuk flek, pori, dan re...	520000	2025-12-28 20:57:21
<input type="checkbox"/>	10	Infusion Whitening	Perawatan infus untuk membantu mencerahkan kulit d...	780000	2025-12-28 20:57:21

## ➤ Tampilan API

**DELETE Request:**

- Method: DELETE
- URL: http://localhost/savenna/admin/api/api\_admin.php?id=3
- Query Parameters: id = 3
- Response: 200 OK. Status: true, message: "Admin berhasil dihapus"

**GET Request:**

- Method: GET
- URL: http://localhost/savenna/admin/api/api\_admin.php
- Response: 200 OK. Result: [{"id": "1", "username": "admin", "password": "89192823e7bbd73298516f8e3df18b5ee"}]

API Tester interface showing a POST request to `http://localhost:8080/admin/api/admin.php`. The request body is a JSON object: `{ "username": "admin", "password": "123456" }`. The response is a 200 OK status with a JSON body: `{ "status": true, "message": "Admin berhasil di login" }`.

API Tester interface showing a PUT request to `http://localhost:8080/admin/api/admin.php`. The request body is a JSON object: `{ "id": "1", "username": "admin", "password": "98765" }`. The response is a 200 OK status with a JSON body: `{ "status": true, "message": "Admin berhasil di update" }`.

API Tester interface showing a DELETE request to `http://localhost:8080/admin/api/admin.php?id=1`. The response is a 200 OK status with a JSON body: `{ "status": true, "message": "Admin berhasil dihapus" }`.

API Tester interface showing a GET request to `http://localhost:8080/admin/api/admin.php`. The response is a 200 OK status with a JSON body: `{ "result": "no data found" }`.

API Tester interface showing a POST request to `http://localhost:8080/admin/api/api_booking.php`. The request body is a JSON object:

```
{
  "name": "Nella",
  "no_hp": "0812545789",
  "email": "facial.acne",
  "tanggal": "2023-12-18",
  "jam": "14:30",
  "status": "tersempai"
}
```

The response is a 200 OK status with a JSON body:

```
{
  "status": true,
  "message": "Booking berhasil diambatkan"
}
```

API Tester interface showing a PUT request to `http://localhost:8080/admin/api/api_booking.php`. The request body is a JSON object:

```
{
  "id": "1",
  "name": "Nella",
  "no_hp": "0812545789",
  "email": "facial.acne",
  "tanggal": "2023-12-18",
  "jam": "14:30",
  "status": "tersempai"
}
```

The response is a 200 OK status with a JSON body:

```
{
  "status": true,
  "message": "Booking berhasil diupdate"
}
```

API Tester interface showing a DELETE request to `http://localhost:8080/admin/api/api_produk.php?id=12`. The request body is empty. The response is a 200 OK status with a JSON body:

```
{
  "status": true,
  "message": "Produk berhasil dihapus"
}
```

API Tester interface showing a GET request to `http://localhost:8080/admin/api/api_produk.php`. The response is a 200 OK status with a JSON body:

```
{
  "result": [
    {
      "id": "1",
      "nama_produk": "Savienne Brightening Toner",
      "deskripsi": "Toner pencerah w",
      "harga": "15000"
    },
    {
      "id": "2",
      "nama_produk": "Savienne Acne Serum",
      "deskripsi": "Serum wajah untuk jerawat",
      "harga": "20000"
    },
    {
      "id": "3",
      "nama_produk": "Savienne Brightening Serum",
      "deskripsi": "Serum pencerah w",
      "harga": "25000"
    },
    {
      "id": "4",
      "nama_produk": "Savienne Day Cream",
      "deskripsi": "Krim siang wajah",
      "harga": "30000"
    },
    {
      "id": "5",
      "nama_produk": "Savienne Night Cream",
      "deskripsi": "Krim malam wajah",
      "harga": "35000"
    },
    {
      "id": "6",
      "nama_produk": "Savienne Facial Wash",
      "deskripsi": "Sabun pembersih wajah",
      "harga": "10000"
    }
  ]
}
```

API Tester - Response

Scenario

METHOD: POST URL: http://localhost:8080/admin/api/api\_produk.php

HEADERS: Content-Type: application/json

BODY: { "nama\_produk": "Selenium Risetlar Muter", "kategori": "Penerbitan keji Berkas 100", "harga": 50000, "stock": 20 }

Response: 200 OK

HEADERS: Date: Sun, 28 Dec 2025 12:07:09 GMT, Server: Apache/2.4.18 (Ubuntu), X-Powered-By: PHP/7.2.12, Content-Length: 54 bytes, Keep-Alive: timeout=5, max=100, Connection: Keep-Alive

BODY: { "status": true, "message": "Produk berhasil ditambahkan" }

API Tester - Response

Scenario

METHOD: PUT URL: http://localhost:8080/admin/api/api\_produk.php

HEADERS: Content-Type: application/json

BODY: { "id": "1", "nama\_produk": "Selenium Risetlar Muter", "kategori": "Penerbitan keji Berkas 100", "harga": 50000, "stock": 20 }

Response: 200 OK

HEADERS: Date: Sun, 28 Dec 2025 12:07:10 GMT, Server: Apache/2.4.18 (Ubuntu), X-Powered-By: PHP/7.2.12, Content-Length: 54 bytes, Keep-Alive: timeout=5, max=100, Connection: Keep-Alive

BODY: { "status": true, "message": "Produk berhasil diupdate" }

API Tester - Response

Scenario

METHOD: DELETE URL: http://localhost:8080/admin/api/api\_transaksi.php?id=2

HEADERS: Content-Type: application/json

BODY: { "id": 2 }

Response: 200 OK

HEADERS: Date: Sun, 28 Dec 2025 12:07:11 GMT, Server: Apache/2.4.18 (Ubuntu), X-Powered-By: PHP/7.2.12, Content-Length: 54 bytes, Keep-Alive: timeout=5, max=100, Connection: Keep-Alive

BODY: { "status": true, "message": "Transaksi berhasil dihapus" }

API Tester - Response

Scenario

METHOD: GET URL: http://localhost:8080/admin/api/api\_transaksi.php

HEADERS: Content-Type: application/json

BODY: { "result": [ { "id": "1", "nama\_produk": "Vista", "no\_ho": "001234567890", "id\_produk": "A", "stock": 20 } ] }

Response: 200 OK

HEADERS: Date: Sun, 28 Dec 2025 12:07:12 GMT, Server: Apache/2.4.18 (Ubuntu), X-Powered-By: PHP/7.2.12, Content-Length: 345 bytes, Keep-Alive: timeout=5, max=100, Connection: Keep-Alive, Content-Type: application/json

BODY: { "result": [ { "id": "1", "nama\_produk": "Vista", "no\_ho": "001234567890", "id\_produk": "A", "stock": 20 } ] }

API Tester interface showing a POST request to `http://localhost:8080/admin/api/apu_treatment.php`. The request body is a JSON object:

```
{  "name_treatment": "Gigitan",  "name_animal": "081215151515",  "id_animal": 4,  "petugas": 1,  "tanggal": "2020-12-28"}
```

The response is a 200 OK status with a JSON body:

```
{  "status": true,  "message": "Transaksi berhasil dilakukan"}
```

API Tester interface showing a PUT request to `http://localhost:8080/admin/api/apu_treatment.php`. The request body is a JSON object:

```
{  "id": 2,  "name_treatment": "Gigitan",  "name_animal": "081215151515",  "id_animal": 4,  "petugas": 1,  "tanggal": "2020-12-28"}
```

The response is a 200 OK status with a JSON body:

```
{  "status": true,  "message": "Transaksi berhasil diupdate"}
```

API Tester interface showing a POST request to `http://localhost:8080/admin/api/apu_treatment.php`. The request body is a JSON object:

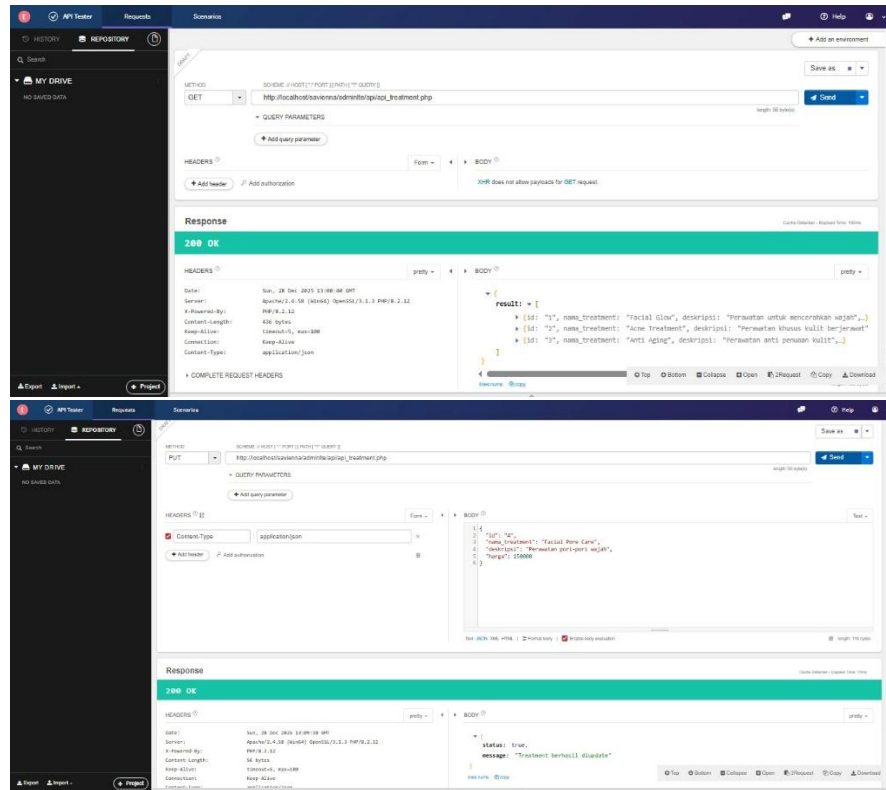
```
{  "name_treatment": "Tetapi Sisi Kontrol",  "name_animal": "Transaksi gagal terkirim",  "tanggal": "2020-12-28"}
```

The response is a 200 OK status with a JSON body:

```
{  "status": true,  "message": "Transaksi berhasil dilakukan"}
```

API Tester interface showing a DELETE request to `http://localhost:8080/admin/api/apu_treatment.php?id=4`. The request body is empty. The response is a 200 OK status with a JSON body:

```
{  "status": true,  "message": "Treatment berhasil dihapus"}
```



## ➤ Tampilan hasil response REST API produk Savienna



