

Perancangan *Server-Side* Website *E-Commerce* Menggunakan Metode Waterfall pada PT Bizhub Digital Indonesia

Muhammad Hilmy Aziz^{1*}, Bagus Satrio Wicaksono², Muhammad Akbar Nurdianto³,
Ardhon Rakhmadi⁴

Fakultas Ilmu Komputer, Universitas Pembangunan Nasional “Veteran” Jawa Timur, Surabaya,
Indonesia^{1,2,3,4}

*Email Korespondensi: mhaziz03@gmail.com

Sejarah Artikel:

Diterima 14-06-2026
Disetujui 18-06-2026
Diterbitkan 20-06-2026

ABSTRACT

PT Bizhub Digital Indonesia previously relied on third-party affiliate platforms for its promotional activities. This reliance incurred additional operational costs and limited the company's control over centralized partner management. This study aims to design and implement a server-side infrastructure (REST API) for an e-commerce platform with an independent affiliate feature to resolve these operational issues. The system development adopted the Waterfall method, which includes requirements analysis, system design, implementation, and testing. The backend infrastructure was built using the Hono.js framework, TypeScript language, and Bun runtime environment. The design resulted in 16 relational database tables and 12 API module groups documented via Swagger UI. Core functionalities include JSON Web Token (JWT) authentication, order management, and payment gateway integration via webhook to automate payment status conversion and affiliate commission distribution in real-time. Functional evaluation using Black-Box Testing on 54 API endpoints demonstrated a 100% operational success rate. The resulting system is proven reliable and capable of reducing third-party dependency, providing operational control efficiency, and enhancing the transparency of the company's financial reporting.

Keywords: *Affiliate System, Server-Side, Backend, Black-Box Testing, E-Commerce, Waterfall*

ABSTRAK

PT Bizhub Digital Indonesia sebelumnya bergantung pada platform afiliasi pihak ketiga untuk aktivitas promosinya. Hal ini menimbulkan beban biaya operasional tambahan serta membatasi kontrol perusahaan terhadap pengelolaan mitra secara terpusat. Penelitian ini bertujuan untuk merancang dan mengimplementasikan infrastruktur *server-side* (REST API) untuk platform *e-commerce* dengan fitur afiliasi mandiri sebagai solusi atas permasalahan tersebut. Pengembangan sistem dilakukan menggunakan metode *Waterfall* yang mencakup tahapan analisis kebutuhan, desain sistem, implementasi, dan pengujian. Infrastruktur *backend* dibangun menggunakan *framework* Hono.js, bahasa TypeScript, dan *runtime* Bun. Hasil perancangan menghasilkan 16 tabel basis data relasional dan 12 kelompok modul API yang terdokumentasi melalui Swagger UI. Fungsionalitas utama mencakup autentikasi berbasis *JSON Web Token* (JWT), manajemen pesanan, dan integrasi *payment gateway* melalui *webhook* untuk otomatisasi konversi status pembayaran serta distribusi komisi afiliasi secara *real-time*. Evaluasi fungsional menggunakan metode *Black-Box Testing* terhadap 54 *endpoint* API menunjukkan tingkat keberhasilan operasional mencapai 100%. Sistem yang dihasilkan terbukti andal, dan mampu mengurangi ketergantungan pada pihak ketiga, memberikan efisiensi kontrol operasional, serta meningkatkan transparansi pelaporan finansial perusahaan.

Kata kunci: *Sistem Afiliasi, Server-Side, Backend, Black-Box Testing, E-Commerce, Waterfall*

Bagaimana Cara Sitasi Artikel ini:

Aziz, M. H., Wicaksono, B. S. ., Nurdianto, M. A. ., & Rakhmadi, A. . (2026). Perancangan Server-Side Website E-Commerce Menggunakan Metode Waterfall pada PT Bizhub Digital Indonesia. Jejak Digital: Jurnal Ilmiah Multidisiplin, 2(4), 6223-6238. <https://doi.org/10.63822/qyp25h29>

PENDAHULUAN

Perkembangan teknologi informasi mendorong perusahaan untuk mulai memanfaatkan internet sebagai media penjualan melalui platform *e-commerce* (Hibatullah et al., 2025). Hadirnya *e-commerce* memfasilitasi transaksi secara elektronik dan memberikan kemudahan bagi konsumen untuk berbelanja tanpa keharusan mengunjungi lokasi bisnis secara fisik (Ardiansyah, 2023; Manurung & Heryana, 2023). Selain itu, situs *e-commerce* juga menjadi solusi alternatif untuk mengatasi keterbatasan area jangkauan bisnis dan mampu meningkatkan kecepatan penawaran produk (Revina Pravita Sari et al., 2025). Sejalan dengan digitalisasi perdagangan, model pemasaran afiliasi (*affiliate marketing*) juga mulai diterapkan sebagai sarana promosi melalui penyebaran tautan oleh pihak ketiga di ekosistem digital (Andrianata et al., 2025).

PT Bizhub Digital Indonesia merupakan perusahaan pemasaran digital yang berfokus pada pemberdayaan wirausahawan melalui strategi *digital marketing* dan pemasaran berbasis komunitas. Selama ini, perusahaan memanfaatkan platform afiliasi pihak ketiga untuk mendukung aktivitas promosi. Namun, penggunaan platform eksternal tersebut menimbulkan biaya operasional tambahan dan membatasi fleksibilitas perusahaan dalam mengatur strategi pemasaran. Ketiadaan sistem internal menyebabkan perusahaan belum memiliki pondasi untuk mengimplementasikan mekanisme pelacakan kontribusi mitra dan pengelolaan komisi yang terintegrasi (Haekal Aqila et al., 2025).

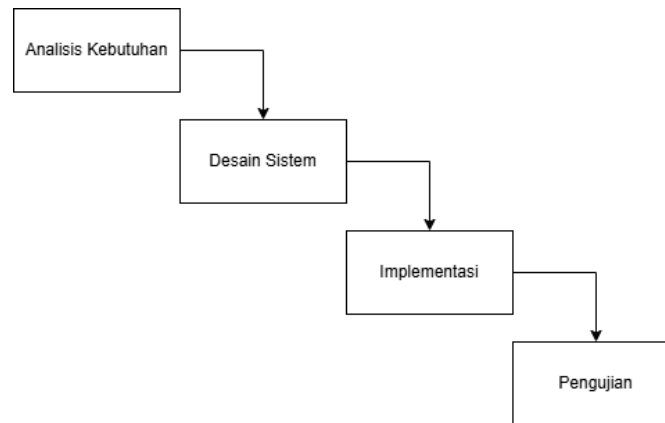
Penelitian sebelumnya oleh Andrianata dkk (2025) membuktikan bahwa integrasi model afiliasi yang dikelola dengan baik dapat meningkatkan jangkauan pasar secara signifikan. Untuk mengimplementasikan sistem yang solid, diperlukan metodologi perancangan perangkat lunak yang sistematis. Perancangan tersebut dapat direalisasikan melalui metode *Waterfall* (Diatullah Guna Darma et al., 2024; Jannah et al., 2023). Metode ini mengharuskan pengembang untuk menyelesaikan tahapan sistematis yang mencakup proses perencanaan, analisis, desain, implementasi, serta pengujian (Angel et al., 2025). Pendekatan langkah demi langkah pada metode *Waterfall* memberikan tingkat kontrol yang lebih besar, sebab satu fase harus diselesaikan secara penuh sebelum beralih ke fase selanjutnya (Fernando, 2023). Penerapan metode ini dinilai relevan untuk perancangan sistem *e-commerce* karena fitur-fitur fundamentalnya memiliki karakteristik stabilitas kebutuhan fungsional yang sudah dapat diidentifikasi secara jelas sejak tahap awal (Gilang & Marlindawati, 2022). Pada tahap akhir, sistem juga dievaluasi menggunakan teknik *Black-Box Testing* yang menguji setiap fungsi (Diyah Saputri et al., 2023).

Penelitian ini berfokus pada pembangunan arsitektur *server-side* (REST API) untuk mengotomatisasi seluruh logika perhitungan afiliasi dan transaksi *e-commerce* di dalam satu platform yang terpusat. Oleh karena itu, penelitian ini bertujuan untuk merancang dan membangun infrastruktur *backend* sistem *e-commerce* yang terstruktur menggunakan metode *Waterfall*, mengimplementasikan autentikasi dan otorisasi pengguna yang aman, serta mengintegrasikan *payment gateway* beserta pelacakan komisi agar seluruh operasional berjalan efisien, transparan, dan terkontrol.

METODE PENELITIAN

Pengembangan infrastruktur *server-side* pada penelitian ini dilakukan menggunakan metode *Waterfall*. Metode *Waterfall* merupakan salah satu model *System Development Life Cycle* / SDLC yang bersifat sekuensial linier, di mana setiap tahapan pengembangan harus diselesaikan dan didokumentasikan secara tuntas sebelum melangkah ke tahapan berikutnya (Akbar Wibowo & Indrayana, 2025; Ratna Sari & Abdul Aziz, 2021). Pemilihan metode ini didasarkan pada karakteristik platform *e-commerce* yang

menuntut standar keamanan tinggi dan memiliki spesifikasi kebutuhan fungsional yang sudah stabil serta dapat diidentifikasi secara jelas sejak awal proyek (Madani, 2023). Alur kerja dari metode *Waterfall* yang diterapkan dalam perancangan arsitektur *backend* ini diilustrasikan pada Gambar 1.



Gambar 1. Tahapan Penelitian

Berdasarkan Gambar 1, masing-masing tahapan pengembangan yang diaplikasikan pada perancangan sistem ini yaitu:

1) Analisis Kebutuhan

Tahap ini difokuskan pada proses pengumpulan spesifikasi kebutuhan perangkat lunak secara mendalam untuk mengatasi permasalahan operasional entitas bisnis. Analisis dilakukan untuk memetakan alur bisnis terpusat serta mengidentifikasi hak akses tiga aktor utama sistem, yaitu Admin, *Customer*, dan *Affiliator*. Kebutuhan fungsional sistem dirumuskan secara mendetail untuk merancang modul-modul *server-side* yang dibutuhkan.

2) Desain Sistem

Pada tahapan desain, spesifikasi kebutuhan yang telah didefinisikan ditransformasikan menjadi *blueprint* arsitektur teknis. Fokus perancangan di sisi *server-side* adalah memodelkan struktur basis data relasional melalui diagram *Physical Data Model* (PDM). Proses ini menghasilkan tabel entitas yang saling terhubung melalui *foreign key*. Pemodelan database yang detail dirancang guna memastikan aliran data transaksi, mulai dari pembentukan pesanan hingga alokasi komisi mitra, tersimpan secara akurat tanpa adanya redundansi data.

3) Implementasi

Tahap implementasi merupakan proses penerjemahan rancangan *blueprint* ke dalam baris-baris kode pemrograman. Infrastruktur *backend* pada penelitian ini dibangun dengan arsitektur *Representational State Transfer Application Programming Interface* (REST API) menggunakan framework Hono.js, dengan TypeScript, dan lingkungan *runtime* Bun. Implementasi ini mencakup penyusunan modul fungsional API utama yang didokumentasikan melalui Swagger UI. Selain itu, sistem diintegrasikan secara langsung dengan *payment gateway* guna mengotomatisasi konversi status pembayaran melalui *webhook* secara *real-time*.

4) Pengujian

Fase pengujian sebagai penjaminan mutu guna memverifikasi bahwa perangkat lunak

beroperasi sesuai standar fungsionalitas yang diharapkan. Proses validasi menggunakan metode *Black-Box Testing*. Pengujian ini mengadopsi teknik *equivalence partitioning*, yaitu pengujian fungsi dengan memberikan masukan (*input*) parameter berdasarkan rentang nilai valid dan tidak valid. Evaluasi dilakukan berbantuan perangkat lunak Postman terhadap *endpoint* API untuk memastikan ketepatan struktur respons keluaran berformat JSON dan validasi penanganan galat (*error handling*).

HASIL DAN PEMBAHASAN

A. Analisis Kebutuhan

Berdasarkan analisis terhadap alur bisnis yang berjalan, ditemukan bahwa ketergantungan perusahaan pada platform afiliasi pihak ketiga telah menambah beban biaya operasional dan menyulitkan proses *tracking* kontribusi mitra serta kalkulasi komisi. Ketiadaan sistem internal juga menghambat pengelolaan laporan kinerja pemasaran. Oleh karena itu, kebutuhan fungsional *backend* dirancang secara detail untuk memfasilitasi tiga aktor utama sistem, yaitu Admin, Customer, dan Affiliator. Berdasarkan pemodelan Use Case sistem, hak akses dan kebutuhan dari masing-masing aktor diidentifikasi sebagai berikut:

1) Admin

Bertindak sebagai pengelola utama sistem. Kebutuhan *endpoint* untuk Admin mencakup fungsi login, akses Dashboard Admin (untuk memantau performa penjualan dan afiliasi), melakukan operasi CRUD (Create, Read, Update, Delete) pada entitas produk, memantau seluruh daftar order pelanggan secara real-time, mengelola daftar Affiliator, serta melakukan validasi transaksi dan persetujuan pengajuan *withdrawal* komisi, Use Case Diagram Admin dapat dilihat pada Gambar 2.



Gambar 2. Use Case Diagram Admin

2) Customer

Bertindak sebagai pembeli produk. Kebutuhan *endpoint* mencakup fitur *register* dan *login* dengan verifikasi OTP, melihat katalog dan detail produk, menambahkan item ke dalam *cart* (keranjang belanja), melihat *Dashboard Akun* (riwayat pesanan), serta melakukan *checkout* pesanan, Use Case Diagram Customer dapat dilihat pada Gambar 3.



Gambar 3. Use Case Diagram Customer

3) Affiliator

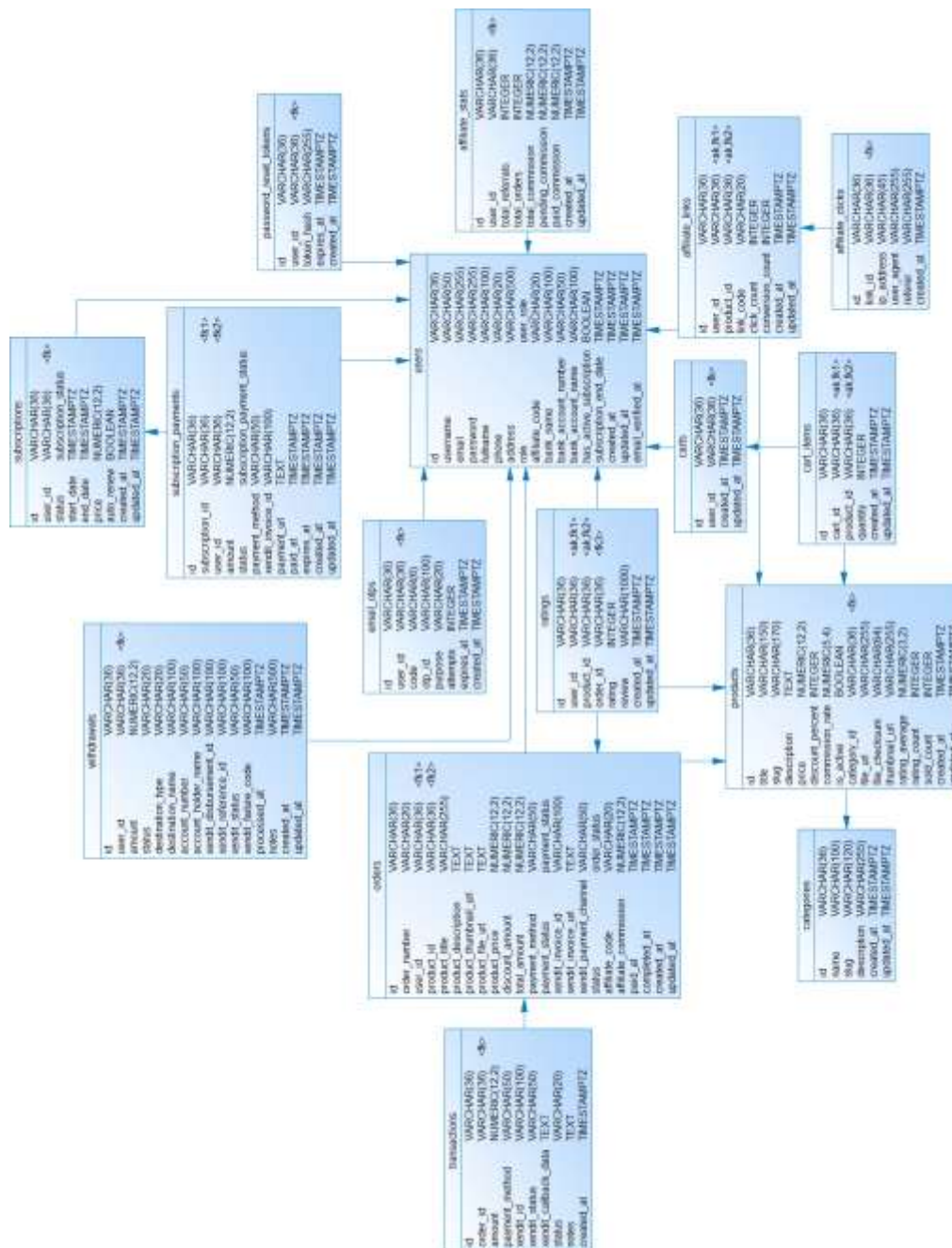
Bertindak sebagai mitra pemasaran. Kebutuhan *endpoint* mencakup fitur *generate link* afiliasi yang unik, memonitor statistik konversi dan klik (*tracking*), mengelola profil, serta melakukan pengajuan penarikan dana (*withdrawal*) saldo komisi berdasarkan *commission rate* yang telah ditetapkan, Use Case Diagram Affiliator dapat dilihat pada Gambar 4.



Gambar 4. Use Case Diagram Affiliator

B. Desain Sistem

Tahap desain sistem bertujuan untuk mentransformasikan hasil spesifikasi kebutuhan fungsional menjadi *blueprint* yang terstruktur sebelum memasuki fase penulisan kode program. Mengingat penelitian ini berfokus pada infrastruktur *server-side*, maka inti dari perancangan arsitektur terletak pada pemodelan struktur database.



Gambar 5. Physical Data Model (PDM)

Berdasarkan gambar 5, perancangan database direpresentasikan melalui *Physical Data Model* (PDM). Proses pemodelan ini menghasilkan 16 tabel entitas yang saling berelasi. Entitas fungsional utama mencakup tabel *users* untuk menyimpan identitas serta peran pengguna (*role*), tabel *categories* dan *products* untuk manajemen katalog digital, tabel *carts* dan *cart_items* untuk alur keranjang belanja, serta tabel *orders* dan *transactions* sebagai pusat pencatatan pesanan *checkout*. Untuk mengakomodasi fitur

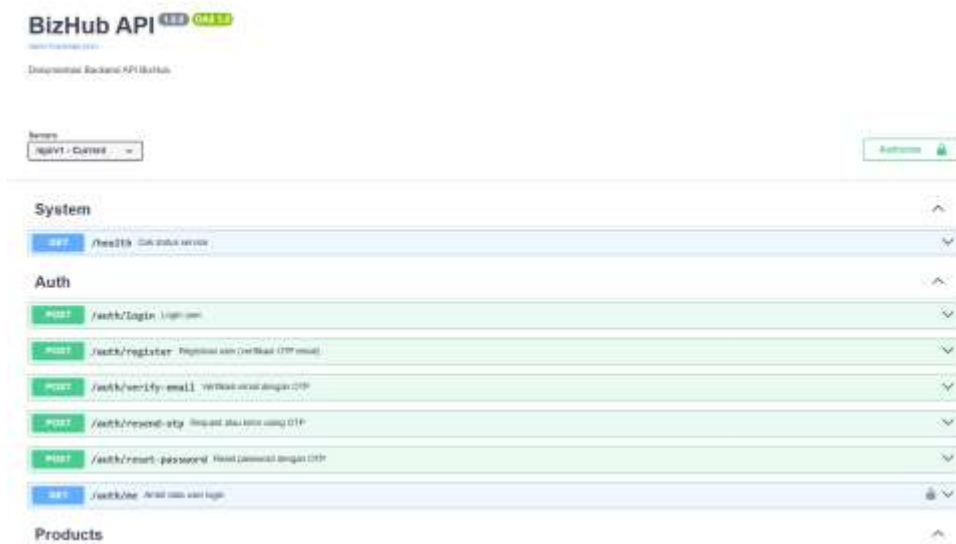
afiliasi, perancangan diperluas dengan menambahkan entitas spesifik seperti *affiliate_links* untuk pembuatan tautan referal, *affiliate_clicks* untuk pelacakan kunjungan ke website dari link afiliator, *affiliate_stats* untuk rekapitulasi data analitik, serta *subscriptions* dan *withdrawals* guna mengelola status langganan dan pencairan dana komisi mitra afiliasi.

Seluruh hubungan antar-entitas (*relationship*) di dalam database dikonfigurasi menggunakan *foreign key*, untuk mendistribusikan proses integrasi data di kedua belah pihak, yakni *Customer* dan *Affiliator*. Sebagai contoh konkret, pada tabel *orders*, data pesanan pembeli dihubungkan secara langsung dengan identitas mitra afiliasi melalui *foreign key*. Mekanisme relasional ini menjamin bahwa setiap kali transaksi dinyatakan berhasil, persentase nilai komisi dapat langsung dikalkulasi dan dialokasikan ke saldo *affiliator* yang bersangkutan. Desain ini meminimalkan risiko kesalahan perhitungan serta memastikan pelaporan keuangan perusahaan berjalan transparan dan efisien.

C. Implementasi

Tahap implementasi merupakan fase penerjemahan *blueprint* arsitektur sistem dan desain database ke dalam bahasa pemrograman yang dapat dieksekusi secara fungsional. Pada tahapan ini, infrastruktur *server-side* platform *e-commerce* dibangun dengan menerapkan arsitektur *Representational State Transfer Application Programming Interface* (REST API). *Tech stack* yang digunakan meliputi framework Hono.js, bahasa pemrograman TypeScript, dan *runtime* dengan Bun.

Hasilnya adalah 12 modul API yang seluruh *endpoint*-nya didokumentasikan menggunakan antarmuka Swagger UI. Penggunaan dokumentasi ini bertujuan untuk menstandarisasi visualisasi spesifikasi API, menguji parameter secara langsung, serta memfasilitasi proses integrasi data yang efisien bagi tim pengembang antarmuka pengguna.



Gambar 6. Tampilan Swagger UI

Pengembangan 12 kelompok modul REST API tersebut difokuskan pada pemenuhan spesifikasi kebutuhan operasional Admin, *Customer*, dan *Affiliator*. Berikut adalah ke-12 modul API yang berhasil

diimplementasikan.

1) Modul System

Menyediakan *endpoint* /health sebagai sarana *health check* untuk memverifikasi bahwa API dan koneksi database berada dalam kondisi aktif dan stabil.

2) Modul Auth

Menangani proses pengelolaan identitas dan keamanan akses. Fungsionalitasnya mencakup pendaftaran akun (*register*), *login*, verifikasi *email* menggunakan *One Time Password* (OTP), serta pembaruan kata sandi (*reset password*). Keamanan dilindungi menggunakan *JSON Web Token* (JWT) dan algoritma *hashing* *bcryptjs*.

3) Modul Products

Menyediakan *endpoint* operasional *CRUD* (*Create, Read, Update, Delete*) untuk entitas produk. Implementasi ini mencakup fungsionalitas *upload* berkas produk dan *thumbnail*, pengaturan harga, persentase diskon, serta penentuan *commission rate* untuk pembagian hasil afiliasi.

4) Modul Categories

Mengelola klasifikasi atau pengelompokan produk, misalnya "E-book", "Kelas", atau "Template". Modul ini berfungsi untuk memudahkan proses *filtering* dan navigasi pencarian pada katalog produk di sisi *frontend*.

5) Modul Cart

Mengelola penyimpanan sementara item produk yang dipilih oleh pelanggan. Logika *backend* pada modul ini menangani penambahan produk, penyesuaian kuantitas, penghapusan item, hingga kalkulasi total harga sebelum masuk ke tahap pembayaran.

6) Modul Orders

Modul ini bertugas memproses konversi dari keranjang belanja menuju pesanan (*checkout*), serta pembuatan nomor pesanan secara otomatis. Pembuatan pesanan diintegrasikan langsung dengan *payment gateway* untuk memproduksi tautan *invoice*. Modul ini dilengkapi dengan *Webhook Handler* yang bertugas menyinkronkan pembaruan status pembayaran secara *real-time*.

7) Modul Affiliate

Merupakan modul untuk menjalankan program pemasaran afiliasi. Fungsionalitasnya mencakup pembuatan tautan *referral* per mitra, pelacakan trafik kunjungan, serta perekaman konversi penjualan. Ketika *webhook* menyatakan transaksi berhasil, sistem akan otomatis mengkalkulasi persentase komisi dan mendistribusikannya ke dalam ringkasan saldo *Affiliator*.

8) Modul Profile

Menyediakan *endpoint* bagi pengguna untuk memperbarui data identitas personal, mengelola kredensial kata sandi, serta melihat dasbor riwayat pesanan bagi *Customer*.

9) Modul Subscription

Menangani paket keanggotaan pengguna. Modul ini melakukan pengecekan masa aktif status langganan yang menjadi prasyarat bagi seorang pengguna untuk dapat mengakses fitur kemitraan afiliasi.

10) Modul Ratings

Memfasilitasi interaksi pengguna untuk memberikan umpan balik berupa skor bintang 1-5 dan ulasan teks terhadap produk yang telah dibeli. Sistem dilengkapi dengan *eligibility check* untuk memastikan bahwa hanya pembeli yang dapat memberikan ulasan, sekaligus mengkalkulasi rata-rata penilaian produk.

11) Modul Withdrawal

Mengelola alur pengajuan penarikan dana saldo komisi oleh mitra afiliasi ke rekening bank atau dompet digital. Di sisi administrator, modul ini menyediakan fungsi *approval* atau penolakan pengajuan, yang terhubung dengan layanan *disbursement* payment gateway untuk mengeksekusi transfer dana secara otomatis.

12) Modul Dashboard Admin

Menyediakan *endpoint* untuk administrator. Modul ini mengumpulkan dan menghitung Key *Performance Indicators*, seperti ringkasan arus kas, total pendapatan, jumlah pertumbuhan afiliator, serta riwayat pesanan terbaru.

D. Pengujian

Pengujian pada penelitian ini dilakukan menggunakan metode *Black-Box Testing* berbantuan perangkat lunak Postman. Berfokus pada evaluasi *input* seperti pengiriman parameter dan *request body* serta *output* berupa respons format JSON dan kode status HTTP, tanpa melihat struktur kode internal dari sistem. Pengujian disimulasikan secara langsung terhadap 54 *endpoint* API yang mencakup keseluruhan modul sistem. Skenario pengujian mengevaluasi respon sistem terhadap *request* yang valid maupun perlindungan terhadap *request* yang tidak valid seperti pengujian *middleware* otorisasi JWT yang menolak akses ilegal dengan kode 401 *Unauthorized*. Hasil pengujian terhadap 54 *endpoint* tersebut disajikan pada Tabel 1.

Tabel 1. *Black-Box Testing*

No	Modul	Endpoint	Skenario Pengujian	Hasil Yang Diharapkan	Hasil
1	System	GET /health	Cek status server.	Status 200, status "ok".	Berhasil
2	Auth	POST /auth/register	Registrasi user baru.	Status 201, memicu OTP.	Berhasil
3		POST /auth/verify-email	Verifikasi OTP email.	Status 200, akun aktif.	Berhasil
4		POST /auth/login	Login kredensial valid.	Status 200, kirim JWT.	Berhasil
5		POST /auth/resend-otp	Minta ulang OTP.	Status 200, OTP dikirim.	Berhasil
6		POST /auth/reset-password	Ganti password via OTP.	Status 200, update sukses.	Berhasil
7		GET /auth/me	Ambil sesi user aktif.	Status 200, detail profil.	Berhasil

8	Products	GET /products	List produk + filter.	Status 200, array produk.	Berhasil
9		POST /products	Tambah produk + upload.	Status 201, aset tersimpan.	Berhasil
10		GET /products/id	Ambil produk via ID.	Status 200, data detail.	Berhasil
11		PUT /products/id	Update data produk.	Status 200, data berubah.	Berhasil
12		GET /products/slug/{slug}	Cari produk via slug.	Status 200, SEO friendly.	Berhasil
13		GET /products/id/related	Rekomendasi produk.	Status 200, produk sejenis.	Berhasil
14		DELETE /products/id	Hapus data produk.	Status 200, data terhapus.	Berhasil
15	Categories	GET /categories	Ambil semua kategori.	Status 200, list kategori.	Berhasil
16		POST /categories	Buat kategori baru.	Status 201, kategori aktif.	Berhasil
17		GET /categories/id	Detail satu kategori.	Status 200, info kategori.	Berhasil
18		PUT /categories/id	Edit nama/slug kategori.	Status 200, update tersimpan.	Berhasil
19		DELETE /categories/id	Hapus kategori.	Status 200, kategori hilang.	Berhasil
20	Cart	GET /cart	Lihat isi keranjang.	Status 200, list item belanja.	Berhasil
21		POST /cart	Tambah item ke cart.	Status 201, item bertambah.	Berhasil
22		DELETE /cart	Kosongkan keranjang.	Status 200, cart kosong.	Berhasil
23		DELETE /cart/{itemId}	Hapus satu item cart.	Status 200, item terhapus.	Berhasil

24	Orders	POST /orders	Checkout dari cart.	Status 201, buat invoice.	Berhasil
25		GET /orders	Riwayat pesanan user.	Status 200, list transaksi.	Berhasil
26		POST /orders/direct	Beli langsung (Buy Now).	Status 201, bypass cart.	Berhasil
27		GET /orders/{orderId}	Detail satu pesanan.	Status 200, rincian bayar.	Berhasil
28	Affiliate	GET /affiliate/dash	Statistik dashboard.	Status 200, total klik/komisi.	Berhasil
29		GET /affiliate/comms	Riwayat komisi masuk.	Status 200, detail per order.	Berhasil
30		GET /affiliate/balance	Saldo komisi saat ini.	Status 200, info nominal.	Berhasil
31		GET /affiliate/withdraws	List riwayat penarikan.	Status 200, status dana.	Berhasil
32		POST /affiliate/withdraw	Ajukan tarik dana.	Status 201, request diproses.	Berhasil
33		GET /affiliate/links	Daftar link promosi.	Status 200, list link unik.	Berhasil
34		POST /affiliate/links	Buat link referral baru.	Status 201, link unik dibuat.	Berhasil
35		GET /affiliate/track/{c}	Pelacakan klik link.	Status 302, redirect produk.	Berhasil
36		GET /affiliate/profile/{i}	Profil publik affiliator.	Status 200, informasi affiliator.	Berhasil
37		GET /affiliate/list	Admin: List affiliator.	Status 200, semua mitra.	Berhasil
38	Profile	GET /profile/dash	Dashboard customer.	Status 200, stats belanja.	Berhasil
39		PUT /profile	Update data diri user.	Status 200, data terbaru.	Berhasil

40		PUT /profile/pass	Ubah password user.	Status 200, keamanan terupdate.	Berhasil
41	Subscription	GET /sub/status	Cek masa aktif membership.	Status 200, info durasi.	Berhasil
42		POST /sub/subscribe	Daftar langganan baru.	Status 201, buat tagihan sub.	Berhasil
43		GET /sub/payments	Riwayat bayar sub.	Status 200, histori invoice.	Berhasil
44	Ratings	POST /ratings	Beri ulasan produk.	Status 201, rating disimpan.	Berhasil
45		PUT /ratings/id	Edit ulasan lama.	Status 200, ulasan diubah.	Berhasil
46		GET /ratings/my	Lihat ulasan pribadi.	Status 200, list ulasan.	Berhasil
47		GET /ratings/prod/id	Ulasan per produk.	Status 200.	Berhasil
48		GET /ratings/check/id	Cek hak memberi ulasan.	Status 200, true/false.	Berhasil
49		GET /ratings/id	Detail ulasan tunggal.	Status 200, info rating.	Berhasil
50	Withdrawal	GET /withdrawals	Daftar antrean tarik dana.	Status 200, list pending.	Berhasil
51		POST /withdraws/id/prc	Admin: Setujui penarikan.	Status 200.	Berhasil
52		POST /withdraws/id/rej	Admin: Tolak penarikan.	Status 200, penarikan dibatalkan.	Berhasil
53	Admin Dashboard	GET /dashboard/stats	Analitik performa bisnis.	Status 200, statistik revenue.	Berhasil
54		GET /dashboard/saldo	Audit rincian saldo.	Status 200, rincian saldo keseluruhan.	Berhasil

Berdasarkan pengujian *Black-Box Testing* terhadap 54 *endpoint* REST API, hasil menunjukkan bahwa arsitektur *backend* sistem e-commerce PT Bizhub Digital Indonesia telah mencapai tingkat kesuksesan sebesar 100%. Tidak ditemukan adanya *error* pada *ouput* maupun anomali pada struktur database. Proses sinkronisasi, seperti konfirmasi status *payment gateway* melalui *Webhook Handler* dan kalkulasi *commission rate* saat transaksi sukses berjalan sesuai yang direncanakan.

KESIMPULAN

Berdasarkan hasil perancangan dan implementasi yang telah dilakukan, dapat disimpulkan bahwa pembangunan infrastruktur *server-side* berbasis REST API untuk platform *e-commerce* dan afiliasi pada PT Bizhub Digital Indonesia telah berhasil direalisasikan sesuai dengan kebutuhan bisnis menggunakan metode *Waterfall*. Penggunaan *framework* Hono.js, bahasa TypeScript, dan *runtime* Bun efektif dalam mengonstruksi 12 modul API untuk menjalankan seluruh alur sistem.

Hadirnya sistem *server-side* ini memberikan solusi dalam ketergantungan perusahaan terhadap *platform* pihak ketiga, sehingga kontrol operasional menjadi jauh lebih fleksibel dan beban biaya dapat ditekan secara signifikan. Ditambah dengan adanya integrasi *payment gateway*, proses verifikasi pesanan dan pencairan dana komisi mitra kini dapat terekam dan berjalan secara otomatis.

Hasil pengujian fungsionalitas melalui metode *Black-Box Testing* terhadap 54 *endpoint* API menunjukkan tingkat keberhasilan sebesar 100% tanpa ditemukan adanya *error* pada respons *output* maupun pada *database*. Secara keseluruhan, platform *server-side* yang dirancang sudah cukup aman, dan siap untuk diimplementasikan guna menunjang strategi pemasaran digital PT Bizhub Digital Indonesia.

DAFTAR PUSTAKA

- Akbar Wibowo, R., & indrayana, D. (2025). Implementasi Metode Waterfall dalam Pengembangan Aplikasi Toko Online Bebas Web di Cv Aishastore.Id. In *Syntax Admiration* (Vol. 6, Number 1).
- Andrianata, M., Suharsono, J., Nafis, R. W., & Fithrianto, N. (2025). The Effectiveness of the Affiliate Marketing Model in Enhancing the Competitiveness of MSMEs in the Digital Economy Era. *JUMANSI : Jurnal Ilmiah Manajemen Dan Akuntansi Medan*.
- Angel, R., Agustina, W., Nurhasanah, N., Mauluddin, A. C., & Handayani, R. N. (2025). Pengembangan Platform E-Commerce UMKM Berbasis Laravel dengan Blackbox Testing dan Metode Waterfall. *Jurnal Pendidikan Dan Teknologi Indonesia*, 5(2). <https://doi.org/10.52436/1.jpti.684>
- Ardiansyah, H. (2023). Implementasi Sistem Informasi Enterprise Resource Planning (ERP) Purchase And Point Of Sale Pada Apotek Puri Sehat Berbasis Web. *OKTAL : Jurnal Ilmu Komputer Dan Science*, 2(12).
- Diatullah Guna Darma, F., Rofni Wulandari, I., Kusumaningtyas, K., & Tri Hartanti, N. (2024). Implementasi Metode Waterfall Pada Perancangan Website Seminar Nasional KOMA. *Information Technology Journal*, 6(1).
- Diyah Saputri, R., Setyadi, R., Informasi, S., Purwokerto, I., Panjaitan No, J. DI, Purwokerto Selatan, K., & Tengah, J. (2023). Perancangan Website E-Commerce IFSTORE Dengan Metode Waterfall. *Jurnal Ilmu-Ilmu Informatika Dan Manajemen*, 17(2).

- Fernando, E. (2023). Designing E-Commerce Sales Of Footwear Using The Waterfall Method. In *International Journal of Science*. <http://ijstm.inarah.co.id>
- Gilang, F., & Marlindawati, P. R. (2022). Perancangan Website Pada Kantor Camat Mulak Ulu Kabupaten Lahat Dengan Metode Waterfall Menggunakan Framework Laravel. In *Jurnal Nasional Ilmu Komputer* (Vol. 3, Number 3).
- Haekal Aqila, H. A., Firda Salsabila Putri, & Wasis Haryono. (2025). Sistem Manajemen Bengkel Berbasis Web Menggunakan Metode Waterfall (Studi Kasus: Fam Auto). *Neptunus: Jurnal Ilmu Komputer Dan Teknologi Informasi*, 3(3), 86–102. <https://doi.org/10.61132/neptunus.v3i3.937>
- Hibatullah, M. H., Tukino, & Hananto, A. L. (2025). PERANCANGAN WEBSITE E-COMMERCE MENGGUNAKAN METODE WATERFALL PADA PENJUALAN ALAT KOMPUTER. *Jurnal SINTA: Sistem Informasi Dan Teknologi Komputasi*, 2(3). <https://doi.org/10.61124/sinta.v2i3.61>
- Jannah, A., Meuraxa, A. M., & Azzahrah, A. (2023). Web Based E-Commerce System Design at EXO Shop Using The Waterfall Method. *Hanif Journal of Information Systems*, 1(1), 1–5. <https://doi.org/10.56211/hanif.v1i1.3>
- Madani, M. (2023). Perancangan Website Media Promosi Produk Gerabah Menggunakan Metode Waterfall. *Jurnal Bumigora Information Technology (BITE)*, 5(2). <https://doi.org/10.30812/bite/v5i2.3370>
- Manurung, H. R., & Heryana, N. (2023). PERANCANGAN WEBSITE E-COMMERCE MENGGUNAKAN METODE WATERFALL PADA PENJUALAN PAKAIAN IMPOR. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7 No.2.
- Ratna Sari, E., & Abdul Aziz, M. (2021). Rancang Bangun Sistem Informasi BUMDes Giri Bangun Berbasis Website Menggunakan Metode Waterfall. *Jurnal Riset Teknologi Informasi Dan Komputer (JURISTIK)*.
- Revina Pravita Sari, Karnadi, & Jimmie. (2025). Applying the Waterfall method to build applications E-Commerce at Palembang City Computer Embroidery Partners. *JURNAL TEKNOLOGI DAN OPEN SOURCE*, 8(2), 1243–1256. <https://doi.org/10.36378/jtos.v8i2.5199>