

Penerapan Metode *Scrum* dalam Perancangan *REST API Website E-Commerce* Toko Oleh-Oleh

Farhan Wundari
S1 Rekayasa Perangkat Lunak
Telkom University
Purwokerto

farhanwunda@student.telkomuniversity.ac.id

Abednego Dwi Septiadi, S.Kom., M.Kom
S1 Rekayasa Perangkat Lunak
Telkom University
Purwokerto

abednego@telkomuniversity.ac.id

Maryona Septiara, S.Pd., M.Kom
S1 Rekayasa Perangkat Lunak
Telkom University
Purwokerto

septiara@telkomuniversity.ac.id

Abstrak — Meningkatnya penggunaan internet memberikan peluang besar bagi para pelaku usaha untuk memperluas jangkauan pasar mereka melalui media digital, termasuk Pusat Oleh-oleh Sawangan No.1 Purwokerto. Namun, toko ini belum memiliki sistem penjualan berbasis *online*, sehingga menyulitkan pelanggan dari luar daerah untuk melihat dan membeli produk. Penelitian ini bertujuan untuk merancang dan mengembangkan *backend* sistem *REST API* menggunakan metode *Scrum* guna mendukung *platform e-commerce* toko Sawangan. Pengembangan *REST API* dilakukan menggunakan *Express.js* dan *PostgreSQL*, serta diuji menggunakan metode *whitebox testing* dengan bantuan *framework Jest*. Sebanyak 189 skenario pengujian dijalankan dengan hasil seluruh skenario berhasil dalam waktu 25.827 detik. Hasil pengujian ini menunjukkan bahwa fungsi sistem seperti *autentikasi*, pengelolaan data produk, dan transaksi telah berjalan sesuai dengan kebutuhan. Sistem ini diharapkan dapat memudahkan pelanggan dalam melakukan pembelian secara *online* serta meningkatkan kualitas layanan dan daya saing toko di tengah transformasi digital.

Kata kunci— *REST API, Scrum, E-commerce, Whitebox*

I. PENDAHULUAN

Pesatnya perkembangan internet dan teknologi telah mendorong pertumbuhan *e-commerce*, sehingga membuka peluang sekaligus tantangan baru bagi pelaku usaha [1]. *Website e-commerce* memungkinkan konsumen mengakses produk tanpa harus ke toko fisik [2], serta membantu pelaku UMKM menjangkau pasar lebih luas dan meningkatkan visibilitas [3].

Pusat Oleh-oleh Sawangan No.1 Purwokerto sebagai salah satu pelaku UMKM belum memanfaatkan sistem digital, sehingga konsumen luar daerah kesulitan dalam melihat dan membeli produk secara langsung. Hal tersebut didukung oleh hasil survei terhadap 56 responden, yang mana sebanyak 73,2% mengaku kesulitan mengetahui detail, harga, dan ketersediaan produk, serta 67,8% merasa sulit melakukan pembelian dari luar kota.

Tujuan utama penelitian untuk merancang dan membangun sistem *backend* berbasis *REST API* untuk mendukung *website e-commerce* Pusat Oleh-oleh Sawangan No.1 Purwokerto. Sistem ini dirancang untuk mengatur, menyimpan, dan menyajikan data produk, transaksi, dan

layanan lain secara terstruktur dan terintegrasi karena pengembangan *backend* menggunakan *REST API* yang memungkinkan pertukaran data antara klien dan server secara efisien [4].

Selain itu, pengembangan sistem memanfaatkan metode *scrum* karena mampu mengakomodasi perubahan kebutuhan secara *iterative*, menjadikan pengembangan lebih cepat, dan *responsive* [5]. Pengujian dilakukan dengan metode *whitebox* menggunakan *Jest* guna memastikan bahwa logika sistem, termasuk autentikasi, pengolahan produk, dan transaksi, berjalan sesuai fungsinya [6].

II. KAJIAN TEORI

Beberapa kajian teori yang digunakan sebagai referensi adalah:

A. *REST (Representational State Transfer)*

REST merupakan arsitektur komunikasi yang banyak digunakan dalam pengembangan aplikasi dan layanan *website* karena sifatnya yang sederhana dan efisien dalam mentransfer data melalui protokol HTTP [7].

B. *API (Application Programming Interface)*

API adalah interface yang dirancang oleh *system developers* untuk memungkinkan sebagian atau seluruh fungsi sistem diakses secara terprogram [7].

C. *Website*

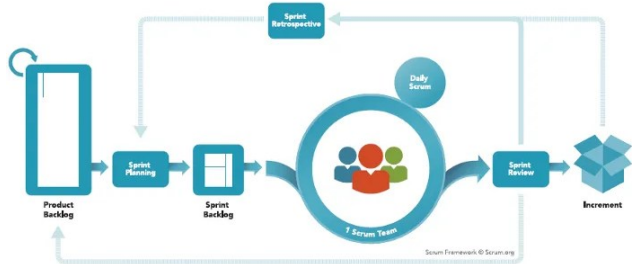
Website adalah rumah digital yang berisi berbagai halaman terhubung dan bisa diakses kapan saja melalui internet yang di dalamnya terdapat teks, gambar, video, dan fitur interaktif untuk menyampaikan informasi atau layanan [8]. *Website* digunakan untuk berbagai keperluan seperti belanja *online*, hiburan, komunikasi, dan diakses melalui alamat unik yang disebut URL [9].

D. *E-Commerce*

E-commerce adalah aktivitas jual beli secara *online* yang melibatkan transfer data dan pembayaran elektronik antara pembeli dan penjual [10]. Transaksinya mencakup produk fisik, layanan, dan menjadi solusi praktis dalam perdagangan *modern* karena cepat, mudah, dan lintas batas [11].

E. Scrum

Scrum adalah metode pengembangan *software* berbasis *agile* yang berfokus pada kolaborasi tim, pengembangan bertahap, dan iteratif untuk mencapai hasil optimal [12]. Metode ini fleksibel, mudah dikendalikan, membantu mengelola resiko, dan menyederhanakan pengembangan produk kompleks [13].



GAMBAR 1
(SIKLUS METODE SCRUM)

F. Whitebox Testing

Whitebox testing adalah teknik pengujian yang fokus pada pemeriksaan langsung alur logika dan struktur kode untuk memastikan fungsinya berjalan benar [14]. Pengujian ini membantu menemukan *bug* tersembunyi yang sering luput dalam pengujian luar, sehingga meningkatkan performa dan keandalan sistem [15].

Metode penelitian dimulai dari tahap pengumpulan data yang dilakukan dengan wawancara langsung kepada pemilik toko Pusat Oleh-oleh Sawangan No.1 Purwokerto dan kuesioner *online* kepada calon pelanggan; identifikasi masalah dilakukan dengan analisis dan pengamatan menyeluruh untuk mengetahui kebutuhan dan permasalahan; studi literatur dari buku dan penelitian terdahulu; *product backlog* menentukan fitur yang harus dimiliki sistem; *sprint planning* merupakan pertemuan awal pada tiap *sprint* untuk tim memilih *product backlog*, menentukan *sprint goal*, dan menyusun rencana kerja; *sprint* dilakukan dengan mengerjakan tugas yang sudah ditentukan; *daily scrum* adalah rapat harian selama 10-15 menit untuk menyampaikan *progress*; *sprint review* adalah pertemuan tiap akhir *sprint* untuk menyampaikan hasil kerja; *sprint retrospective* dilakukan evaluasi proses kerja tim selama *sprint*; *deploy* adalah proses implementasi REST API.

IV. HASIL DAN PEMBAHASAN

A. Product Backlog

Hasil yang diperoleh pada tahap ini berupa *backlog* atau daftar kebutuhan, fitur, atau tujuan yang diharapkan pengguna. *Product backlog* yang diperoleh dipaparkan pada Tabel 1.

TABEL 1
(HASIL *PRODUCT BACKLOG*)

No	<i>Product Backlog</i>	Prioritas
1	Pembuatan UML	4
2	Endpoint Auth	9
3	Endpoint Profile	8
4	Endpoint Produk	9
5	Endpoint Keranjang	10
6	Endpoint Checkout	8
7	Endpoint Detail Product	7
8	Endpoint Review Product	7
9	Endpoint Wishlist	5
10	Endpoint Manajemen Product	10
11	Endpoint Manajemen User	7
12	Endpoint Manajemen Order	10
13	Unit Pengujian	7
14	Deploy	8

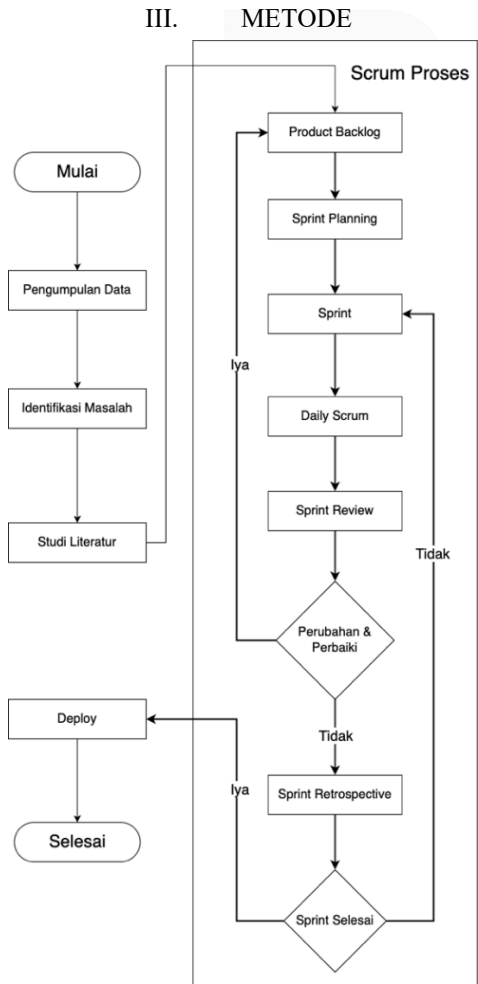
B. Sprint Planning

Tahap yang dilakukan pada setiap awal *sprint* untuk memutuskan *backlog* yang akan dikerjakan dan memecahnya menjadi tugas yang lebih kecil. Pada penelitian ini terbagi menjadi tiga *sprint* yang masing-masing *sprint* berdurasi 2 pekan.

GAMBAR 3
(*SPRINT 1*)

Setup Project	Membuat UML	Endpoint Authentication	Endpoint Profile	Endpoint Profile	Endpoint Produk
Setup project dari library api yang dibutuhkan dan buat repos di github	Membuat desain UML untuk memvisualisasikan sistem dan revisi prior komponenn	Mulai membangun API untuk login, register, dan autentikasi pengguna	Mengembangkan API untuk pengelolaan data profil pengguna, termasuk update informasi pribadi.	Mengembangkan API untuk pengelolaan data profil pengguna, termasuk update informasi pribadi.	Mengembangkan API untuk menampilkan daftar produk dari database

Sprint pertama dimulai dari pembuatan desain sistem dari *requirements* yang disepakati dan membangun infrastruktur awal sistem dengan teknologi yang telah ditentukan. Fokus utama dalam *sprint* 1 adalah membangun *endpoint auth* dan *profile user*.



GAMBAR 2
(ALUR PENELITIAN)

GAMBAR 4
(SPRINT 2)

Endpoint Detail Produk	Endpoint Keranjang	Endpoint Checkout	Endpoint Order Detail & Tracking	Endpoint Review Produk	Endpoint Wishlist
Membangun API untuk menampilkan detail informasi produk, termasuk deskripsi, harga, dan gambar.	Membuat API untuk mengelola keranjang belanja, seperti menambah, menghapus, atau mengupdate item.	Membangun API untuk memproses transaksi checkout dan pembayaran.	Membuat API untuk melihat detail order & tracking pesanan.	Mengembangkan API untuk memungkinkan pengguna menulis dan melihat review produk.	Membuat API untuk fitur wishlist, memungkinkan pengguna menyimpan produk favorit.

Sprint kedua dilakukan pengembangan fitur utama yang digunakan pengguna meliputi daftar produk, detail produk, pengelolaan keranjang belanja, *checkout*, *review*, dan *wishlist*. Seluruh *endpoint* direlasikan dengan *database* guna memastikan alur transaksi *user* berjalan.

GAMBAR 5
(SPRINT 3)

Endpoint Manajemen Produk	Endpoint Manajemen User	Endpoint Manajemen Order	Unit Testing	Documentasi & Deploy
Membangun API untuk admin mengelola data produk, termasuk tambah, edit, atau hapus produk.	Mengembangkan API untuk admin mengelola data pengguna.	Membangun API untuk admin memantau dan memproses pesanan pengguna.	Membuat dan menjalankan unit test untuk memastikan setiap modul berjalan sesuai harapan.	Membuat dokumentasi API dan melakukan deployment agar aplikasi siap digunakan.

Sprint ketiga memfokuskan pada pengembangan fitur pada sisi admin yaitu manajemen data produk, pengguna, dan pesanan. Apabila seluruh proses pengembangan selesai, maka dilakukan *unit testing* guna memastikan kualitas sistem yang kemudian dilanjutkan pada proses *deployment* supaya sistem dapat digunakan di lingkungan produksi.

C. *Sprint*

Sprint memuat serangkaian aktivitas yang berkaitan satu sama lain, mulai dari proses implementasi sampai evaluasi. Berikut merupakan hasil pengerjaan *sprint* pertama hingga ketiga.

TABEL 2
(HASIL Pengerjaan *SPRINT 1*)

Task	Start	Day	Finish	Duration
	3/3/2025	14	16/3/2025	
Usecase diagram				6
Activity diagram				12
Sequence diagram				8
ERD Diagram				4
Endpoint <i>auth</i> untuk <i>login</i> , <i>register</i> , <i>logout</i> , <i>forgot password</i> , dan <i>Google OAuth</i>				24
Endpoint <i>profile</i> dari <i>upload</i> foto, <i>update</i> akun, dan <i>reset password</i> .				14
Endpoint daftar produk				6
Total				74

Selanjutnya, *sprint 2* fokus pada pengembangan fitur-fitur utama yang diaskes oleh pengguna.

TABEL 3
(HASIL Pengerjaan *SPRINT 2*)

Task	Start	Day	Finish	Duration
	17/3/2025	14	30/3/2025	
API untuk detail informasi produk, meliputi deskripsi, harga, dan gambar				8
API pada mengelola keranjang belanja, seperti menambah, menghapus, atau mengupdate item				14
API untuk memproses transaksi <i>checkout</i> dan <i>payment</i>				48
API untuk melihat detail <i>order</i> dan <i>tracking</i> pesanan				18
API yang memungkinkan pengguna menulis dan melihat <i>review</i> produk				24

API untuk fitur <i>wishlist</i> , memungkinkan pengguna menyimpan produk <i>favorit</i>	14
Total	126

Sprint ketiga menjadi tahap penyempurnaan dan persiapan *launching*. Fokus *sprint* terakhir ini pada pengembangan fitur-fitur di sisi admin.

TABEL 3
(HASIL Pengerjaan *SPRINT 3*)

Task	Start	Day	Finish	Duration
	31/3/2025	14	13/4/2025	
API untuk admin mengelola data produk, termasuk tambah, edit, atau hapus produk.				19
AP untuk admin mengelola data <i>user</i>				13
API untuk admin memantau dan memproses pesanan (<i>order</i>) pengguna.				24
Membuat dan menjalankan unit <i>test</i> untuk memastikan setiap modul berjalan sesuai harapan.				72
Membuat dokumentasi API				5
Deployment agar aplikasi siap digunakan				12
Total				145

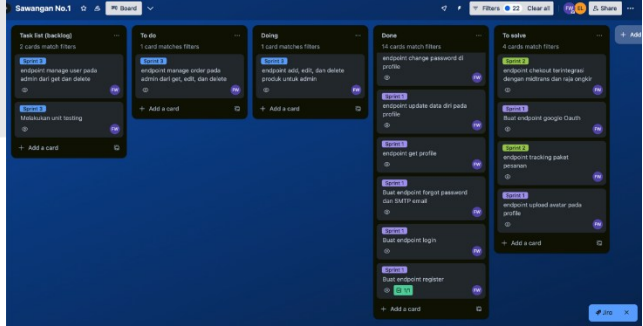
D. *Sprint Review*

Sprint review menjadi *final state* untuk setiap *sprint* yang digunakan untuk meninjau dan memverifikasi hasil pekerjaan tim. Tujuannya tidak hanya memastikan fitur selesai, tetapi juga menilai kualitas dan kesiapan sistem dalam berbagai situasi. *Sprint review* ini difokuskan pada pengujian fungsi sistem secara menyeluruh dan penyusunan dokumentasi API untuk mendukung pengembangan berkelanjutan. Tiga aspek penting yang dibahas dalam sesi ini meliputi pengujian unit (*unit testing*), *blackbox testing*, dan dokumentasi API.

E. Trello

Trello digunakan sebagai dokumentasi visual selama *sprint review* untuk memantau perkembangan tugas secara sistematis. Setiap tugas dikategorikan berdasarkan statusnya seperti *blacklog* sedang dikerjakan hingga selesai, sehingga memudahkan tim melihat progres dan memastikan seluruh fitur dikerjakan dengan rapi dan terstruktur.

GAMBAR 6
(TRELLO)

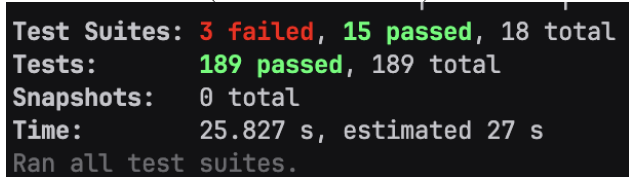


Semua *task* yang tercatat di *Trello* telah diselesaikan dalam periode *sprint* mencakup fitur penting seperti CRUD produk, pemesanan, manajemen pengguna, *wishlist*, proses *checkout*, hingga integrasi dengan layanan ongkir dan *payment gateway*. Beberapa tugas sempat masuk ke kolom *"To solve"* karena memerlukan penanganan khusus, seperti integrasi pengiriman, namun kini telah berhasil diselesaikan.

F. Unit Testing

Proses *unit testing* dilakukan untuk memastikan setiap fitur yang dikembangkan berfungsi sebagaimana mestinya dan pada penelitian ini pengujian dilakukan menggunakan *framework Jest*, sementara *Supertest* digunakan untuk menguji respons HTTP secara langsung pada *endpoint backend* yang dibangun dengan *Express.js*.

GAMBAR 7
(HASIL UNIT TESTING)



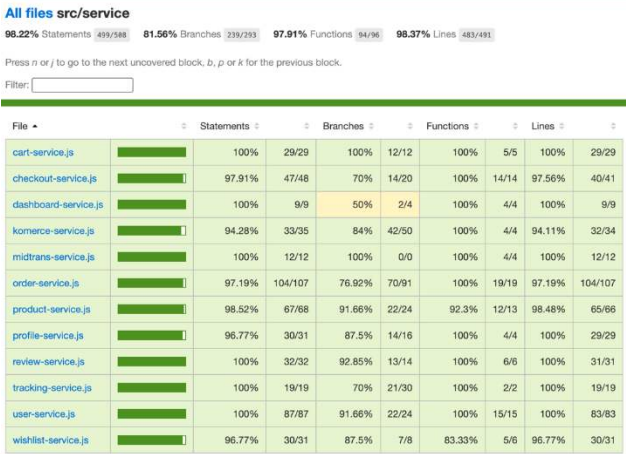
Gambar di atas menunjukkan hasil pengujian *unit test* pada sistem *backend*, di mana terdapat 18 *test suite* dengan 15 di antaranya berhasil dan 3 gagal. Namun, dari total 189 skenario pengujian, seluruhnya berhasil dijalankan (189 *passed* dari 189 total) yang menunjukkan bahwa fitur utama sistem berjalan sesuai harapan. Tiga *test suite* yang gagal tidak disebabkan oleh kesalahan logika sistem, melainkan merupakan *mock test* yang digunakan untuk simulasi pengujian dan tidak terhubung langsung dengan fungsi utama. Seluruh pengujian selesai dijalankan dalam waktu 25,827 detik.

GAMBAR 8
(HASIL UNIT TEST CONTROLLER)



Gambar tersebut menampilkan hasil *unit test* pada folder *controller* yang menunjukkan bahwa pengujian dilakukan secara menyeluruh dan berhasil dengan baik. Berdasarkan 12 file *controller* yang diuji, sebagian besar memiliki tingkat keberhasilan sangat tinggi dengan rata-rata 96.51% untuk *statements*, 89.28% untuk *branches*, 98% untuk fungsi, dan 96.51% untuk baris kode. Beberapa file seperti *checkout-controller.js*, *tracking-controller.js*, dan *review-controller.js* bahkan mencapai 100% di semua kategori. Meskipun ada beberapa file seperti *dashboard-controller.js* dan *kommerce-controller.js* yang memiliki cakupan *branch* sedikit lebih rendah, hal tersebut tidak memengaruhi stabilitas sistem. Secara keseluruhan, hasil ini menunjukkan bahwa fungsi-fungsi pada *controller* telah diuji dengan efektif dan siap digunakan.

GAMBAR 9
(HASIL UNIT TEST SERVICE)



Gambar di atas menunjukkan hasil *unit testing* pada folder *service*, dengan hasil keseluruhan yang sangat baik yaitu 98.22% *statements*, 81.56% *branches*, 97.91% *functions*, dan 98.37% *lines*. Beberapa file seperti *cart-service.js* dan *midtrans-service.js* berhasil diuji secara penuh di semua aspek, menandakan bahwa logika di dalamnya telah tervalidasi dengan baik. Namun, beberapa file seperti *dashboard-service.js* dan *checkout-service.js* masih memiliki kekurangan pada pengujian cabang logika dan fungsi, sehingga belum sepenuhnya teruji. Meski begitu, hasil ini menunjukkan bahwa sebagian besar fungsi layanan sudah diuji dengan baik dan sesuai harapan.

G. Sprint Retrospective

Sprint retrospective merupakan momen untuk merefleksikan dan mengevaluasi jalannya proses setelah satu *sprint* berakhir. Fokusnya bukan hanya pada jumlah tugas yang diselesaikan, tetapi juga pada kualitas proses dan pelajaran yang diperoleh. Setiap kendala yang dihadapi dijadikan bahan introspeksi guna menemukan cara kerja yang lebih efektif dan responsif di *sprint* berikutnya.

H. Dokumentasi API

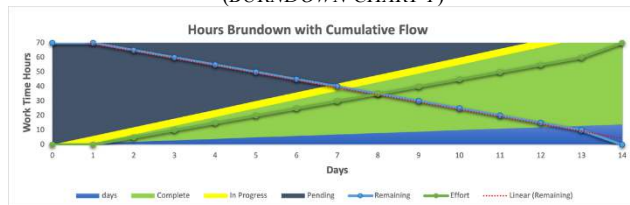
Pembuatan dokumentasi API dilakukan menggunakan *postman* untuk memastikan setiap *endpoint* yang dikembangkan selama *sprint* dapat dipahami dengan mudah oleh semua pihak, khususnya tim pengembang. Dokumentasi ini mencakup daftar lengkap *endpoint* beserta prosedur kerja masing-masing, termasuk detail *request*, *response*, *headers* yang dibutuhkan, serta contoh *payload* dalam format JSON. Selain menjelaskan fungsi dari setiap API, dokumentasi ini juga mendukung proses pengujian agar sesuai dengan spesifikasi yang diharapkan. Untuk memudahkan akses dan penggunaan, dokumentasi telah dibagikan melalui tautan (<https://bit.ly/4kNxVsG>) yang bisa diakses oleh siapa saja yang terlibat dalam pengembangan sistem.

I. Burndown Chart

Pada bagian ini, pembuatan *burndown chart* berperan dalam memantau perkembangan tugas selama *sprint* secara sistematis. Trello dimanfaatkan untuk mengelola pembagian tugas ke dalam tahap "To Do", "In Progress", dan "Done", dengan setiap perubahan status dicatat secara berkala. Data tersebut kemudian digunakan untuk memperbarui *burndown chart* di Excel, yang menyajikan grafik visual guna melihat

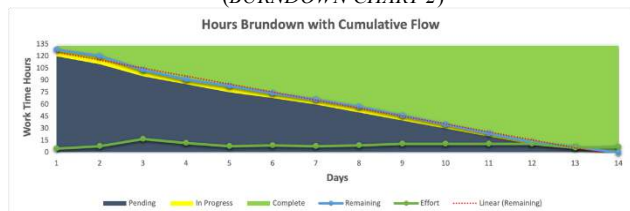
sejauh mana progres berjalan sesuai dengan rencana waktu, serta menjadi acuan dalam melakukan evaluasi dan penyesuaian jika diperlukan.

GAMBAR 10
(BURNDOWN CHART 1)



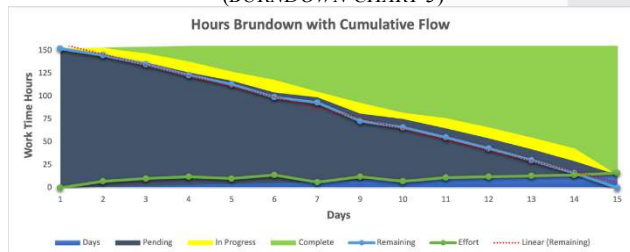
Analisis pada *sprint* pertama ditampilkan dalam bentuk *burndown chart* yang dikombinasikan dengan *Cumulative Flow Diagram* (CFD), guna menggambarkan progres awal pengembangan sistem *e-commerce*. Total waktu pengerjaan yang direncanakan adalah 74 jam selama 14 hari kerja. Grafik menunjukkan penurunan beban kerja (*remaining*) secara bertahap hingga tuntas di hari terakhir, yang menandakan seluruh tugas telah diselesaikan tepat waktu. Warna pada diagram mewakili status tugas yaitu hijau untuk yang telah selesai, kuning untuk yang sedang dikerjakan, dan abu-abu untuk yang masih menunggu.

GAMBAR 11
(BURNDOWN CHART 2)



Grafik tersebut menunjukkan progres *sprint* kedua yang berlangsung selama 14 hari dengan estimasi waktu kerja sebanyak 126 jam. Garis biru menggambarkan waktu kerja yang tersisa, sedangkan garis merah putus-putus menunjukkan jalur ideal penyelesaian tugas secara linier. Dalam pelaksanaannya, terdapat kendala teknis yang cukup menghambat, terutama pada proses integrasi sistem *backend* dengan layanan pihak ketiga seperti *RajaOngkir* dan *Midtrans* yang membutuhkan banyak percobaan, sehingga beberapa tugas memerlukan waktu lebih lama untuk diselesaikan.

GAMBAR 12
(BURNDOWN CHART 3)



Gambar 12 menunjukkan *progress sprint* ketiga yang berlangsung selama 14 hari kerja dengan estimasi total waktu 145 jam. Pada *sprint* ini, intensitas kerja cukup tinggi karena banyak waktu difokuskan pada proses *unit testing*. Setiap pengujian unit biasanya melibatkan 3 hingga 4 skenario, sehingga memerlukan durasi pengerjaan yang lebih panjang dibandingkan *sprint* sebelumnya.

Berdasarkan analisis dari ketiga grafik *burndown* yang telah dibuat, dapat disimpulkan bahwa metode ini efektif untuk memantau dan mengendalikan progres *sprint* selama pengembangan sistem *e-commerce*. Pada *sprint* pertama, penurunan waktu kerja berjalan konsisten hingga akhir *sprint*, menandakan tugas terselesaikan sesuai target. Sementara itu, *sprint* kedua dan ketiga, meskipun memiliki estimasi waktu lebih besar yaitu 126 dan 145 jam, tetap menunjukkan penyelesaian tugas yang stabil dan mendekati garis estimasi ideal. Pembagian tugas melalui *Trello* juga terbukti membantu dalam menjaga keteraturan proses kerja. Kombinasi *burndown* dan CFD memberikan gambaran visual yang jelas terhadap status pengerjaan setiap hari, sehingga tim dapat segera merespons jika terjadi hambatan, dan memastikan *sprint* tetap efisien serta terukur.

V. KESIMPULAN

Berdasarkan hasil penelitian mengenai perancangan *backend* REST API untuk *website e-commerce* Toko Sawangan No.1 Purwokerto, dapat disimpulkan bahwa sistem *backend* berhasil dikembangkan untuk menjawab permasalahan utama toko, seperti penyajian informasi produk secara daring, kemudahan transaksi *online*, dan perluasan jangkauan layanan ke luar daerah. Pengembangan dilakukan secara terstruktur mulai dari perencanaan kebutuhan, penyusunan dokumen teknis seperti *product backlog*, *use case diagram*, *sequence diagram*, hingga *erd*, dan diimplementasikan melalui pembangunan REST API yang langsung terhubung ke basis data. Metode *scrum* diterapkan dalam seluruh tahapan pengembangan, mulai dari perencanaan hingga evaluasi melalui *sprint review* dan *retrospective*, sehingga memungkinkan proses berjalan secara kolaboratif, bertahap, dan adaptif terhadap perubahan kebutuhan. Pengujian sistem dilakukan menggunakan *white box testing* dengan pendekatan *unit testing*, di mana *jest* digunakan untuk menguji fungsi internal dan *supertest* untuk pengujian respons *endpoint*. Hasilnya, sebanyak 189 skenario pengujian berhasil dijalankan (100% *passed*) dalam waktu 25,827 detik, menunjukkan bahwa seluruh logika sistem, termasuk autentikasi, pengelolaan data produk, dan proses transaksi, telah berjalan sesuai spesifikasi dan bebas dari kesalahan.

REFERENSI

- [1] R. Raharti, M. Irwan, and P. Nasution, "Analisis Dampak E-Commerce Terhadap Pengaruh Penjualan UMKM," *Jurnal Riset Manajemen*, vol. 2, no. 1, pp. 171–178, 2024, doi: 10.54066/jurma.v2i1.1336.
- [2] N. Dimas, I. Dewi, Z. Latiefa, A. Rosanto, and S. Amin, "Perancangan Website sebagai Media Digital Marketing untuk Mendorong Pemasaran UMKM," *Jurnal Bina Desa*, vol. 3, no. 3, pp. 176–180, 2021, [Online]. Available: <https://journal.unnes.ac.id/nju/index.php/jurnalbinadesa>
- [3] S. Kosasi and S. Pontianak, "Perancangan Sistem E-Commerce Untuk Memperluas Pasar Produk Oleh-Oleh Khas Pontianak".
- [4] Rangga Gelar Guntara and V. Azkarin, "Implementasi dan Pengujian REST API Sistem Reservasi Ruang Rapat dengan Metode Black Box

- Testing,” *Jurnal Minfo Polgan*, vol. 12, no. 1, pp. 1229–1238, Jul. 2023, doi: 10.33395/jmp.v12i1.12691.
- [5] I. Bibik, *How to Kill the Scrum Monster: Quick Start to Agile Scrum Methodology and the Scrum Master Role*. Apress Media LLC, 2018. doi: 10.1007/978-1-4842-3691-8.
- [6] S. Khadafi, A. Salim, R. Prabowo, dan Choirul Anam, and I. Teknologi Adhi Tama Surabaya, “Rancang Bangun Website UKM Reviora Tanggulangin Sidoarjo Menggunakan Metode Waterfall Sebagai Media Pemasaran Online.” [Online]. Available: <http://diskopukm.jatimprov.go.id>,
- [7] I. Kurniawan and F. Rozi, “REST API Menggunakan NodeJS pada Aplikasi Transaksi Jasa Elektronik Berbasis Android,” 2020. [Online]. Available: <http://jurnal-itsi.org>
- [8] M. F. Rizaldi *et al.*, “Pengenalan Website sebagai Pengembangan Profil Pondok Pesantren Maqis Al-Hamidy 4,” 2023. [Online]. Available: https://jurnalkip.samawa-university.ac.id/karya_jpm/index
- [9] F. R. Khamdani, I. Setiawan, and B. D. Putranto, “Analisis Kualitas Layanan Website Program Studi Sistem Informasi Menggunakan Metode Webqual 4.0 dan Importance Performance Analysis (IPA).” [Online]. Available: <https://si.fik.amikompurwokerto.ac.id/>.
- [10] M. Iqbal Maliki, “Rancang Bangun Aplikasi Penjualan Grosir Sembako pada Toko La-Ris,” *Jurnal Informatika dan Rekayasa Perangkat Lunak (JATIKA)*, vol. 2, no. 3, pp. 304–311, 2021, [Online]. Available: <http://jim.teknokrat.ac.id/index.php/informatika>
- [11] R. B. Prasetyo, “Pengaruh E-Commerce dalam Dunia Bisnis,” 2023. [Online]. Available: <https://journal.sabajayapublisher.com/index.php/jme>
- [12] T. Setya Nugraha and R. Hardian, “Rancang Bangun Sistem Informasi Company Profile dengan Menggunakan Metode Scrum pada PT. Hasna Satya Negara Berbasis Web Design and Build Company Profile Information System Using Scrum Method At PT. Hasna Satya Negara WEB-Based,” *Jurnal Ilmiah Intech : Information Technology Journal of UMUS*, vol. 03, no. 02, pp. 171–179, 2021.
- [13] S. P. Utami *et al.*, “Rancang Bangun Aplikasi Edukasi Tuberkulosis Menggunakan Metode Scrum,” vol. 7, no. 1, p. 2022.
- [14] A. Andriyadi, R. Rizal Nul Fikri, and E. Friska Saputri, “Evaluasi Sistem Informasi Perpustakaan Institut Informatika Darmajaya dengan WhiteboxTesting,” *Journal of Innovation Reserach and Knowledge*, vol. 1, no. 8, pp. 743–746, Jan. 2022.
- [15] A. Suryanta, L. Wahyu Widiati, dan Mirsyah Ashari, P. Kodiklatad, and S. Jakarta STI, “Pengujian White Box Terhadap Sistem Informasi Akademik (SIKAD) di Politeknik Angkatan Darat,” *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 2, pp. 2384–2390, Apr. 2025.