

BAB III

METODE PENELITIAN

3.1 Jenis Penelitian

Penelitian ini bertujuan untuk merancang sistem informasi *E-commerce* berbasis *platform*, untuk mempermudah transaksi jual beli barang bekas yang masih layak digunakan kembali bagi mahasiswa UPI Purwakarta terutama mereka yang tinggal di kontrakan maupun kos. Dalam pengembangan sistem informasi ini, menggunakan pendekatan *Software Development Life Cycle* (SDLC) model *Waterfall*. Pengembangan sistem informasi *E-commerce* barang bekas menggunakan model *Waterfall* karena mempunyai *workflow* yang jelas dan tahap yang berurutan sehingga dapat mempermudah pengembangan dalam mengontrol dan merencanakan proses proyek. Menurut (Royce, 1970), menggambarkan *Waterfall* sebagai pendekatan sekuensial karena setiap tahap dimulai dari analisis kebutuhan hingga pemeliharaan mampu diselesaikan sebelum memulai tahap berikutnya, dengan adanya penekanan pada dokumentasi untuk meminimalkan risiko kesalahan. Pada penelitian tersebut juga menjelaskan bahwa pendekatan ini cocok bagi sistem dengan kebutuhan yang jelas dan stabil, seperti pengembangan sistem informasi yang memiliki spesifikasi teknis yang terdefinisi dengan baik. Metode *Waterfall* dipilih untuk pengembangan sistem *E-commerce* barang bekas bagi mahasiswa UPI Purwakarta untuk menyesuaikan kebutuhan pengguna, seperti fitur pencarian barang, unggah produk, dan sistem pembayaran, dapat diidentifikasi dengan jelas.

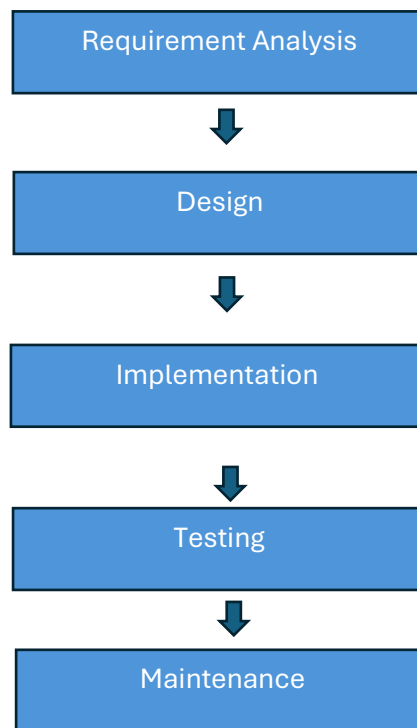
Kelebihan dari model ini adalah adanya tahapan yang lengkap, proses pengembangan yang lebih mudah dikontrol, serta waktu pelaksanaan proyek yang lebih terukur. Disisi lain dengan pendekatan yang runtut, model ini memudahkan tim pengembang untuk memantau progres serta menghindari loncatan tahapan yang dapat mengganggu konsistensi sistem. Namun, model *Waterfall* juga memiliki sejumlah kelemahan, seperti sulitnya melakukan perubahan apabila sistem telah memasuki tahap implementasi. Model ini tidak terlalu fleksibel terhadap perubahan kebutuhan pengguna di tengah proses pengembangan, serta berisiko terhadap kegagalan jika terdapat kesalahan di tahap awal karena tidak adanya proses umpan

balik hingga tahapan akhir. Tetapi, untuk pengembangan sistem informasi *E-commerce* berbasis web seperti dalam penelitian ini, model *Waterfall* tetap relevan digunakan karena proyek bersifat linear, dan kebutuhan pengguna telah didefinisikan secara spesifik sejak awal.

Pada penelitian yang dilakukan oleh (Al Ghani et al., 2022), membahas perancangan sistem informasi *E-commerce* berbasis web dengan menggunakan model pengembangan perangkat lunak *Waterfall*. Dalam penelitian tersebut, model ini dipilih karena terdapat tahapan kerja yang sistematis dan terstruktur, dimulai dari analisis kebutuhan, perancangan sistem, implementasi, pengujian, hingga tahap pemeliharaan. Hasil dari penelitian tersebut menunjukkan bahwa pendekatan *Waterfall* efektif dalam menghasilkan sistem *E-commerce* yang terencana dengan baik, mudah dikembangkan dan dikelola, dan cocok diterapkan pada proyek yang memiliki kebutuhan dan spesifikasi yang jelas sejak awal. Penelitian tersebut sejalan dengan pengembangan dalam penelitian yang akan penulis lakukan. Sistem yang dirancang memiliki kebutuhan fungsional yang relatif stabil, seperti fitur registrasi pengguna, manajemen produk, transaksi barang, dan pelacakan pengiriman. Oleh karena itu, model *Waterfall* menjadi pendekatan yang sesuai karena memungkinkan pengembangan sistem dilakukan secara bertahap dan terkontrol, sehingga setiap tahap dapat diselesaikan secara menyeluruh sebelum melanjutkan ke tahap berikutnya. Dengan struktur yang terorganisir, model ini membantu dalam menciptakan sistem *E-commerce* yang fungsional, efektif, dan sesuai dengan kebutuhan target pengguna, mahasiswa UPI Purwakarta.

3.2 Prosedur Penelitian

Prosedur penelitian ini mengikuti model *Waterfall*, yang merupakan pendekatan pengembangan perangkat lunak secara berurutan dan terstruktur. Tahapan pada model *Waterfall* diantaranya:



Gambar 3. 1 Tahapan *Waterfall*

3.2.1 Analisis Kebutuhan (*Requirement Analysis*)

Tahap analisis kebutuhan merupakan langkah awal dalam model pengembangan perangkat lunak berbasis *Waterfall*, yang bersifat berurutan dan terstruktur. Pada tahap ini, fokus utama adalah mengidentifikasi permasalahan yang akan diselesaikan sistem dan merumuskan kebutuhan sistem secara menyeluruh. Tujuan dari tahap ini adalah untuk menentukan kebutuhan fungsional dan non-fungsional sistem. Menganalisis kebutuhan yang mendalam memberikan kejelasan fitur dan alur sistem yang akan dikembangkan.

3.2.2 Desain Sistem (*Sistem Design*)

Tahap desain sistem meliputi perancangan arsitektur sistem secara keseluruhan berdasarkan hasil analisis kebutuhan. Dalam penelitian ini, desain sistem menggunakan model UML (*Unified Modeling Language*) untuk memastikan representasi yang terstruktur dan sesuai dengan pendekatan berorientasi objek. Penggunaan model *Unified Modeling Language* untuk efisiensi dan kesesuaian dengan standar pengembangan perangkat lunak. Tahap menyusun rancangan sistem yang meliputi berbagai komponen penting sebagai berikut:

1. *Use Case Diagram*

Menggambarkan interaksi pengguna dengan sistem. *Use case diagram* dalam penelitian ini memperlihatkan peran pengguna sebagai penjual dan pembeli, serta tindakan-tindakan yang dapat mereka lakukan seperti menambah barang, mencari barang, dan melakukan transaksi.

2. *Activity Diagram*

Menggambarkan alur aktivitas dari pengguna ketika berinteraksi dengan sistem. Diagram ini menunjukkan bagaimana pengguna melakukan proses login, menelusuri barang, melakukan pembelian, dan menyelesaikan transaksi.

3. *Class Diagram*

Menjelaskan struktur kelas dan hubungan antar kelas dalam sistem. *class diagram* dirancang untuk merepresentasikan elemen utama dalam sistem seperti pengguna, barang, kategori, dan transaksi, termasuk atribut serta metode yang dimiliki oleh setiap kelas. Diagram ini mempermudah pengembang dalam memahami keterkaitan antar objek dan alur logika pemrograman yang digunakan.

4. Desain Antarmuka Pengguna (UI)

Menentukan tampilan visual dan alur interaksi pengguna pada sistem. Desain UI dibuat agar sistem mudah digunakan, responsif, dan menarik. Desain ini mencakup halaman login, registrasi, beranda, daftar barang, formulir unggah barang, dan halaman transaksi.

3.2.3 Implementasi Sistem (*Implementation*)

Tahap implementasi sistem adalah proses realisasi desain untuk menjadi *website E-commerce* fungsional dalam transaksi barang bekas di kalangan mahasiswa UPI Purwakarta. Proses pengembangan tersebut, menggunakan *WordPress* sebagai kerangka dasar pengembangan, karena kemudahan penggunaan dan ketersediaan berbagai *plugin* pendukung. Tema *Woostify* juga digunakan sebagai antarmuka pengguna, dalam memberikan optimasi kecepatan loading dan kompatibilitas penuh dengan *WooCommerce* yang berfungsi sebagai inti pada sistem *E-commerce*.

Pengembangan fitur utama meliputi penerapan *plugin User Frontend* yang memungkinkan mahasiswa mengunggah produk bekas mereka secara mandiri melalui formulir sederhana. *WooCommerce* dapat diintegrasikan sebagai penyedia fungsi dasar seperti manajemen produk, keranjang belanja, dan proses *Checkout*.

Beberapa penyesuaian khusus dilakukan menggunakan *plugin Snippets*, termasuk pengembangan halaman profil penjual yang menampilkan katalog produk, penyempurnaan dashbor pengguna untuk manajemen barang dan transaksi, serta pengoptimalan alur penggunaan sistem. Implementasi ini menghasilkan *website* siap uji yang mendukung transaksi barang bekas secara efisien di lingkungan kampus, sekaligus mempersiapkan sistem untuk tahap pengujian dan pemeliharaan selanjutnya.

3.2.4 Pengujian Sistem (*Testing*)

Tahap pengujian dilakukan untuk memastikan sistem berfungsi sesuai dengan spesifikasi yang telah ditentukan. Setelah sistem dikembangkan, tahap ini bertujuan memastikan fungsi sistem berjalan dengan benar dan sesuai dengan kebutuhan pengguna. Penelitian ini menggunakan metode *Black box Testing*, yang menguji kesesuaian output dengan input tanpa memperhatikan kode program di dalamnya. *Black box Testing* merupakan metode pengujian perangkat lunak yang berfokus pada fungsionalitas sistem tanpa mempertimbangkan struktur atau kode internal, dengan tujuan memastikan bahwa sistem menghasilkan output yang sesuai dengan input yang diberikan berdasarkan spesifikasi kebutuhan. Metode ini mengevaluasi perilaku sistem dari perspektif pengguna akhir melalui skenario pengujian yang mencerminkan penggunaan nyata, seperti memasukkan data yang valid, tidak valid, atau ekstrem untuk menguji ketahanan sistem. Pengujian *BlackBox* efektif dalam mengidentifikasi cacat fungsional pada *website E-commerce*, seperti kesalahan pada proses registrasi akun, pencarian produk, pemesanan, pembayaran, atau pengelolaan notifikasi, yang dapat mengganggu pengalaman pengguna.

Pengujian ini dilakukan dengan skenario pengujian yang mencakup berbagai kasus penggunaan, seperti registrasi pengguna, pencarian produk, dan

pelacakan pesanan. Penelitian ini menegaskan bahwa *BlackBox Testing* meningkatkan kepercayaan pengguna dengan memastikan seluruh fitur yang ada dalam *platform* dapat berfungsi dengan baik, yang sangat penting untuk *website E-commerce* barang bekas yang bergantung pada kepercayaan dalam transaksi. Menggunakan *Black Box Testing* sebagai langkah pengujian suatu *website* juga memiliki kelebihan dan kekurangan diantaranya:

a) Kelebihan *Black Box Testing*

1. Fokus pada Pengalaman Pengguna

Menguji sistem dari perspektif pengguna akhir, memastikan fitur seperti pendaftaran, pencarian barang, dan pembayaran berfungsi sesuai harapan mahasiswa.

2. Tidak Memerlukan Pengetahuan Kode

Penguji tidak perlu memahami struktur kode internal, memungkinkan mahasiswa atau pengguna non-teknis untuk berpartisipasi dalam pengujian.

3. Mencakup Beragam Skenario

Menguji berbagai input, seperti data valid, tidak valid, atau ekstrem, untuk memastikan sistem tahan terhadap kesalahan pengguna, seperti format input yang salah.

b) Kekurangan *BlackBox Testing*

1. Keterbatasan pada Kode Internal

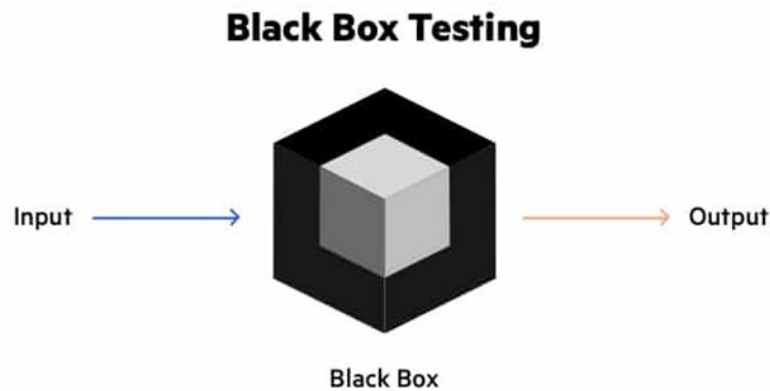
Tidak dapat mendeteksi masalah dalam logika kode atau efisiensi performa internal, seperti bug yang tidak memengaruhi output tetapi memperlambat sistem.

2. Ketergantungan pada Skenario Pengujian

Efektivitas bergantung pada kelengkapan skenario pengujian; jika skenario tidak mencakup kasus tertentu, cacat fungsional dapat terlewat.

3. Tidak Mengatasi Masalah Non-Fungsional Mendalam

Meskipun dapat menguji performa dasar, *Black Box Testing* kurang efektif untuk masalah seperti skalabilitas sistem di bawah beban ekstrem tanpa alat tambahan.



Gambar 3.2 Pengujian *BlackBox*

(Sumber: <https://www.imperva.com/learn/application-security/black-box-testing/>)

Berikut adalah tahapan utama dalam melakukan *Black box Testing* pada pengujian perangkat lunak, khususnya pada sistem *E-commerce* barang bekas berbasis web di lingkungan mahasiswa:

1. Memahami Kebutuhan dan Spesifikasi Sistem
Pahami secara menyeluruh kebutuhan pengguna dan spesifikasi sistem yang akan diuji. Identifikasi bagian aplikasi mana saja yang membutuhkan pengujian dari sisi fungsionalitasnya.
2. Penentuan Input
Tentukan berbagai data input yang akan digunakan untuk menguji setiap fitur. Input bisa berupa data valid, tidak valid, batas minimal/maksimal, kombinasi data, atau skenario khusus lain sesuai fitur yang diuji.
3. Menentukan Output yang Diharapkan
Definisikan output atau hasil yang diharapkan dari setiap input pada tiap fitur. Output ini menjadi acuan dalam menilai apakah aplikasi telah berjalan sesuai requirement.
5. Membuat dan Menyeleksi *Test Case*

Buat daftar test case atau skenario pengujian beserta input-output yang relevan. *Test case* ini akan dijalankan untuk memvalidasi masing-masing fitur.

6. Melakukan Tahap Pengujian

Jalankan aplikasi menggunakan *test case* yang telah disusun. Masukkan input ke sistem, amati output yang dihasilkan, dan dokumentasikan hasil pengujian pada setiap skenario.

7. Membandingkan Output Aktual dengan Harapan

Periksa apakah output yang dihasilkan aplikasi sesuai dengan output yang diharapkan. Catat jika ditemukan ketidaksesuaian, bug, atau error.

8. Evaluasi dan Dokumentasi

Evaluasi hasil pengujian secara keseluruhan. Dokumentasikan setiap hasil uji, baik fitur berjalan normal maupun jika ditemukan masalah, untuk referensi perbaikan pada tahap selanjutnya.

Karakteristik utama *Black box Testing* dalam pengujian perangkat lunak, diantaranya:

1. Fokus pada Fungsionalitas Sistem

Pengujian dilakukan untuk memastikan bahwa setiap fitur sistem bekerja sesuai dengan fungsi yang telah ditentukan. Penguji tidak melihat atau memahami struktur internal kode, melainkan hanya menguji apakah sistem memberikan hasil yang benar berdasarkan input yang diberikan.

2. Berdasarkan Perspektif Pengguna

Pengujian dilakukan dari sudut pandang pengguna akhir, sehingga pendekatan ini merepresentasikan bagaimana pengguna sesungguhnya akan menggunakan sistem. Hal ini penting untuk menilai pengalaman pengguna (*user experience*) dan kemudahan operasional sistem.

3. Validasi Input dan Output

Setiap fitur diuji berdasarkan kombinasi input tertentu, kemudian diverifikasi apakah output yang dihasilkan sesuai dengan yang diharapkan. Misalnya, ketika pengguna mengisi formulir registrasi dan menekan tombol

"Daftar", sistem seharusnya memberikan notifikasi keberhasilan dan menyimpan data ke dalam database.

4. Berorientasi pada Perspektif Pengguna

Black box Testing mensimulasikan cara pengguna nyata berinteraksi dengan sistem. Skenario pengujian disusun mengacu pada kebutuhan fungsional yang telah dirumuskan sebelumnya, seperti pada dokumen *Requirement Analysis*. Dengan kata lain, pengujian memvalidasi bahwa sistem telah dibangun sesuai dengan spesifikasi yang dirancang.

3.2.5 Pemeliharaan (*Maintenance*)

Pemeliharaan bertujuan dalam memastikan sistem tetap stabil, fungsional, dan adaptif terhadap kebutuhan pengguna yang berkembang. Salah satu, aspek utama dari pemeliharaan merupakan perbaikan bug, dengan memperbaiki kesalahan maupun gangguan yang muncul setelah sistem dijalankan secara nyata, seperti kegagalan dalam pengunggahan gambar produk dan lain-lain. Pemeliharaan mencakup penyempurnaan fitur berdasarkan masukan pengguna, seperti penambahan notifikasi otomatis atau integrasi metode pembayaran yang lebih relevan bagi mahasiswa.

Penyesuaian sistem juga diperlukan supaya dapat mengikuti kebutuhan baru, misalnya integrasi dengan sistem kampus atau penambahan fitur filter lokasi. Tahap revisi awal setelah pengujian, yang disebut *early maintenance*, berfungsi dalam menyempurnakan sistem sebelum dirilis secara menyeluruh. Pemeliharaan juga meliputi, upaya dalam menjaga performa jangka panjang, termasuk optimasi kecepatan, stabilitas server, serta pembaruan keamanan data pengguna. Dengan demikian, sistem tetap relevan, aman, dan dapat digunakan secara optimal dalam jangka panjang oleh seluruh pengguna.

3.3 Instrumen Penelitian

Instrumen penelitian sistem *E-commerce* barang bekas berbasis web bagi mahasiswa UPI Purwakarta meliputi lembar observasi, dokumentasi hasil uji coba, dan panduan uji *Black Box*. Lembar observasi mencatat hasil pengujian fitur seperti pendaftaran atau pencarian barang berdasarkan skenario, mencakup input, hasil yang diharapkan, dan hasil aktual. Dokumentasi hasil uji coba berupa screenshot

menunjukkan bukti visual keberhasilan atau kegagalan fitur, seperti halaman unggah produk atau pesan error. Panduan uji *Black box* berisi skenario uji fungsional untuk setiap fitur, seperti menguji input valid dan tidak valid pada pencarian atau pembayaran. Instrumen ini memastikan pengujian dilakukan secara sistematis dan objektif, membantu mengidentifikasi masalah dan memverifikasi fungsionalitas sistem supaya sesuai kebutuhan mahasiswa.

3.4 Subjek Penelitian

Mahasiswa UPI Purwakarta dipilih karena mereka merupakan komunitas yang aktif menggunakan teknologi informasi dan pernah melakukan aktivitas jual beli barang bekas, baik secara langsung maupun melalui media sosial. Subjek dipilih secara *purposive* karena mereka merupakan kelompok yang paling relevan dengan tujuan sistem, yaitu untuk memfasilitasi jual beli barang bekas yang layak pakai di kalangan mahasiswa.