

LAPORAN KERJA PRAKTEK

**PENGUJIAN FUNGSIONAL DENGAN
AUTOMATIC TESTING UNTUK APLIKASI
SAHABATKU**



Oleh:

Yoga Ananta

1462200019

**PROGRAM SARJANA
PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK ELEKTRO
DAN INFORMATIKA CERDAS
UNIVERSITAS 17 AGUSTUS 1945 SURABAYA
2026**

LEMBAR PENGESAHAN

PENGUJIAN FUNGSIONAL DENGAN AUTOMATIC TESTING UNTUK APLIKASI SAHABATKU

Sebagai salah satu syarat untuk melaksanakan Kerja Praktek

Oleh:

Yoga Ananta

1462200019

Surabaya, 06 Januari 2026

Koordinator KP,



Anton Brev Yunanda, S.T., M.MT

NPP. 20450020554

Dosen Pembimbing

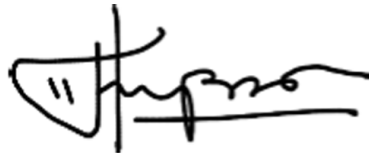


Agus Hermanto, S.Kom., M.MT., ITIL.

NPP. 20460150675

Mengetahui,

Ka, Program Studi Teknik Informatika



Puteri Noraisya Primandari, S.ST., M.IM.

NPP. 20460170736

KATA PENGANTAR

Puji syukur saya panjatkan ke hadirat Tuhan Yang Maha Esa karena atas rahmat dan kesempatan yang diberikan, penulis dapat menyelesaikan laporan penelitian yang berjudul “**Pengujian Fungsional dengan Automatic Testing untuk Aplikasi Sahabatku Menggunakan Selenium pada Website sahabatku.hdnmetatech.com.**” Penelitian ini disusun sebagai salah satu bentuk penerapan pengetahuan yang telah penulis pelajari, sekaligus sebagai bagian dari pemenuhan tugas akademik.

Penyusunan laporan ini tentu tidak terlepas dari bantuan, bimbingan, serta dukungan berbagai pihak. Oleh karena itu, penulis ingin menyampaikan rasa terima kasih yang tulus kepada dosen pembimbing yang telah memberikan arahan selama proses penelitian berlangsung. Ucapan terima kasih juga penulis sampaikan kepada pihak pengelola **Aplikasi Sahabatku**, yang telah memberikan kesempatan kepada penulis untuk melakukan pengujian pada sistem yang mereka kembangkan. Tidak lupa, penulis juga berterima kasih kepada keluarga, teman, dan seluruh pihak yang selalu memberikan semangat, saran, serta bantuan selama proses penyusunan laporan ini.

Penelitian ini berfokus pada bagaimana proses **pengujian fungsional** dapat dilakukan secara otomatis menggunakan **Selenium**, sebagai upaya memastikan bahwa setiap fitur pada aplikasi Sahabatku berjalan sesuai tujuan. Penulis berharap hasil penelitian ini dapat memberikan manfaat, baik sebagai referensi akademik maupun sebagai bahan pertimbangan dalam pengembangan dan peningkatan kualitas perangkat lunak berbasis web.

Penulis menyadari bahwa laporan ini masih memiliki keterbatasan. Oleh karena itu, kritik dan saran sangat penulis harapkan demi penyempurnaan karya ilmiah ini di masa mendatang. Akhir kata, semoga laporan ini dapat memberikan kontribusi dan menjadi bahan pembelajaran bagi pembaca yang tertarik pada topik pengujian perangkat lunak, khususnya pengujian otomatis.

Surabaya, Januari 2026

Penulis

ABSTRAK

Perkembangan teknologi informasi membuat aplikasi berbasis web semakin banyak digunakan untuk mendukung aktivitas kerja, layanan, dan pengelolaan data. Salah satu aplikasi yang dikembangkan untuk tujuan tersebut adalah Aplikasi Sahabatku. Agar aplikasi ini dapat digunakan dengan nyaman dan tidak menimbulkan masalah bagi pengguna, diperlukan proses pengujian yang terstruktur untuk memastikan setiap fungsi berjalan sesuai dengan yang diharapkan. Selama ini, pengujian perangkat lunak sering dilakukan secara manual, yang memerlukan waktu lebih lama dan berisiko menimbulkan kesalahan manusia, terutama ketika pengujian harus dilakukan berulang kali. Kerja praktek ini berfokus pada pengujian fungsional Aplikasi Sahabatku dengan menerapkan metode automatic testing menggunakan Selenium pada website sahabatku.hdnmetatech.com. Pengujian dilakukan dengan menyusun skenario uji (test case) untuk fitur-fitur utama, seperti login, input dan penyimpanan data, navigasi halaman, serta tampilan informasi. Skenario tersebut kemudian diimplementasikan dalam bentuk skrip pengujian otomatis yang dijalankan melalui browser. Dari proses ini disusun dokumentasi hasil pengujian yang memuat kesesuaian antara hasil yang diharapkan dan hasil aktual, serta daftar bug atau ketidaksesuaian fungsi yang ditemukan. Melalui kegiatan ini, diharapkan pengujian fungsional dengan automatic testing dapat membantu pengembang dalam melihat kondisi aplikasi secara lebih objektif, mempercepat proses pengecekan ketika terjadi perubahan pada sistem, serta menjadi dasar untuk perbaikan dan peningkatan kualitas Aplikasi Sahabatku ke depannya.

DAFTAR ISI

LEMBAR PENGESAHAN	ii
KATA PENGANTAR	iii
ABSTRAK	iv
DAFTAR ISI	v
Daftar Gambar	vii
Daftar Tabel	viii
Daftar Lampiran	ix
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan Kerja Praktek	4
1.3 Batasan Kerja Praktek	4
1.4 Tujuan Kerja Praktek	5
1.5 Manfaat Kerja Praktek	6
1.6 Sistematika Penulisan	8
BAB 2 TINJAUAN PUSTAKA	10
2.1 Profil Organisasi	10
2.2 Konsep Kualitas Perangkat Lunak	11
2.3 Metode Pengujian Perangkat Lunak	14
2.4 Perangkat Bantu Pengujian Perangkat Lunak	18
BAB 3 METODOLOGI PELAKSANAAN KERJA PRAKTEK	20
3.1 Tahapan Kerja Praktek	20
3.2 Analisis Kebutuhan Pengujian Perangkat Lunak	21
3.3 Pengembangan Skenario Pengujian Perangkat Lunak	23
BAB 4 PELAKSANAAN KERJA PRAKTEK	25
4.1 Persiapan Pengujian Perangkat Lunak	25
4.2 Implementasi Pengujian Perangkat Lunak	28

4.3 Temuan Dan Rekomendasi	40
BAB 5 KESIMPULAN DAN SARAN.....	42
5.1 Kesimpulan	42
5.2 Saran.....	43
DAFTAR PUSTAKA	45

Daftar Gambar

Gambar 2. 1. Logo Instansi	10
Gambar 4. 1. Install Selenium IDE	25
Gambar 4. 2. Akses Selenium IDE	26
Gambar 4. 3. Membuat Projek Baru	26
Gambar 4. 4. Penamaan Projek	27
Gambar 4. 5. Base URL	27
Gambar 4. 6. Hasil Pengujian Halaman Beranda.....	32
Gambar 4. 7. Hasil Pengujian Halaman Beli produk	33
Gambar 4. 8. Hasil Pengujian Halaman Program Affiliate.....	34
Gambar 4. 9. Hasil Pengujian Halaman Ajukan Klaim	35
Gambar 4. 10. Hasil Pengujian Halaman Artikel.....	36
Gambar 4. 11. Hasil Pengujian Halaman Berita	37
Gambar 4. 12. Hasil Pengujian Halaman Instagram	37
Gambar 4. 13. Hasil Pengujian Halaman WhatsApp.....	38

Daftar Tabel

Table 4.1. Pengujian Navbar Aplikasi Sahabatku.....	30
---	----

Daftar Lampiran

Lampiran 1. Spesifikasi Perangkat Lunak	46
Lampiran 2. Detail Pengujian Perangkat Lunak	50
Lampiran 3. Foto Kegiatan	54
Lampiran 4. Formulir Penilaian	56
Lampiran 5. Formulir Aktivitas	57
Lampiran 6. Formulir Kuesioner	59
Lampiran 7. Lembar Bimbingan.....	62

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi dalam beberapa tahun terakhir telah membawa perubahan besar dalam cara masyarakat menjalankan berbagai aktivitas sehari-hari. Banyak proses yang sebelumnya dilakukan secara manual kini beralih ke sistem digital, mulai dari pelayanan administrasi, pengelolaan data, hingga layanan publik yang berbasis web. Kemajuan ini membuat perangkat lunak menjadi salah satu elemen yang sangat penting dalam mendukung kelancaran berbagai aktivitas, baik di lingkungan kerja maupun kehidupan pribadi. Perangkat lunak yang dirancang dengan baik dapat membantu mempercepat alur kerja, meminimalkan kesalahan, serta meningkatkan efisiensi penggunaan waktu dan tenaga.

Dengan semakin meningkatnya kebutuhan masyarakat terhadap layanan digital, tuntutan akan kualitas perangkat lunak juga semakin besar. Pengguna mengharapkan aplikasi yang mudah digunakan, stabil, cepat, dan bebas dari gangguan. Ketika sebuah aplikasi mengalami error, tidak responsif, atau tampilannya tidak sesuai, hal tersebut dapat menimbulkan ketidaknyamanan bahkan menurunkan tingkat kepercayaan pengguna terhadap aplikasi tersebut. Oleh karena itu, upaya untuk memastikan bahwa perangkat lunak memiliki kualitas yang baik bukan lagi menjadi pilihan, tetapi sebuah keharusan.

Salah satu cara yang digunakan untuk menjaga kualitas perangkat lunak adalah melalui proses pengujian. Pengujian perangkat lunak bertujuan untuk memastikan setiap fungsi berjalan sesuai dengan yang direncanakan, serta membantu menemukan kesalahan atau kekurangan sebelum aplikasi digunakan oleh pengguna secara luas. Pengujian ini tidak hanya mengevaluasi kebenaran sebuah fungsi, tetapi juga memastikan bahwa aplikasi dapat digunakan dalam berbagai kondisi tanpa menimbulkan gangguan. Dengan kata lain, pengujian

menjadi langkah penting untuk memastikan perangkat lunak benar-benar siap digunakan.

Dalam praktiknya, masih banyak pengembang yang melakukan pengujian secara manual. Metode ini memang dapat memberikan gambaran mengenai kondisi aplikasi, tetapi memiliki keterbatasan. Pengujian manual memerlukan waktu yang lama, terlebih jika aplikasi memiliki banyak fitur dan skenario penggunaan. Selain itu, ketika pengujian harus dilakukan berkali-kali—misalnya setelah terjadi perubahan kode atau penambahan fitur—pengujian manual rawan terjadi kesalahan manusia. Kesalahan seperti lupa mengecek satu bagian atau salah memasukkan data saat uji sangat mungkin terjadi. Situasi ini membuat proses pengujian menjadi kurang efisien dan hasilnya tidak selalu konsisten.

Untuk mengatasi kendala tersebut, pengujian otomatis atau automatic testing menjadi salah satu solusi yang banyak digunakan dalam pengembangan perangkat lunak modern. Automatic testing memungkinkan proses pengujian dilakukan secara berulang dengan hasil yang konsisten, cepat, dan lebih akurat. Dengan bantuan tools atau perangkat khusus, pengujian dapat dijalankan secara otomatis tanpa harus melakukan langkah yang sama secara manual. Salah satu tools yang paling banyak digunakan untuk melakukan pengujian otomatis pada aplikasi berbasis web adalah Selenium. Selenium memungkinkan pengujian dilakukan dengan cara mensimulasikan interaksi pengguna pada browser, seperti mengisi formulir, mengeklik tombol, hingga berpindah halaman. Dengan pendekatan ini, pengujian dapat dilakukan pada banyak skenario tanpa memerlukan tenaga dan waktu sebesar pengujian manual.

Kasus yang diangkat dalam kerja praktek ini berkaitan dengan pengujian pada Aplikasi Sahabatku, sebuah aplikasi berbasis web yang dirancang untuk membantu pengguna dalam mengelola aktivitas atau data tertentu secara lebih praktis. Aplikasi ini memiliki beberapa fitur yang penting bagi pengguna, seperti proses login, pengelolaan data, navigasi antarhalaman, dan tampilan informasi. Jika fitur-fitur ini tidak berjalan dengan baik, maka pengguna akan mengalami hambatan dalam menjalankan aktivitas mereka. Oleh karena itu, memastikan bahwa fitur pada

aplikasi Sahabatku berfungsi sebagaimana mestinya menjadi hal yang sangat penting.

Sebagai salah satu platform yang terus berkembang, Aplikasi Sahabatku perlu melalui proses pengujian yang terstruktur dan mendalam. Proses pengujian ini dilakukan untuk memastikan bahwa setiap fungsi pada aplikasi berjalan dengan benar dan stabil sebelum digunakan oleh pengguna secara lebih luas. Dengan menerapkan metode pengujian fungsional melalui automatic testing menggunakan Selenium, diharapkan proses pengujian dapat dilakukan dengan lebih cepat, akurat, dan efisien. Selain itu, penerapan automatic testing ini dapat memberikan gambaran yang jelas tentang bagian mana yang sudah bekerja dengan baik dan bagian mana yang masih memerlukan perbaikan.

Melalui kerja praktek ini, pengujian fungsional dilakukan pada website sahabatku.hdnmetatech.com yang merupakan platform utama dari Aplikasi Sahabatku. Pengujian dilakukan untuk memastikan bahwa keseluruhan fitur inti berjalan normal, mulai dari proses autentikasi, pengolahan data, hingga tampilan antarmuka. Hasil dari pengujian ini nantinya dapat digunakan sebagai dasar untuk memperbaiki aplikasi, meningkatkan kualitas sistem, dan memberikan pengalaman yang lebih baik bagi pengguna.

Berdasarkan uraian tersebut, dapat disimpulkan bahwa pengujian perangkat lunak memainkan peran penting dalam menjaga kualitas aplikasi. Dengan meningkatnya kompleksitas aplikasi dan kebutuhan pengguna, metode automatic testing menjadi solusi yang tepat untuk menghasilkan pengujian yang cepat, konsisten, dan akurat. Pengujian fungsional yang diterapkan pada Aplikasi Sahabatku diharapkan dapat membantu memastikan bahwa aplikasi tersebut sudah siap digunakan secara efektif dan mampu memberikan manfaat bagi penggunanya. Kerja praktek ini juga menjadi pengalaman penting untuk memahami bagaimana proses pengujian dilakukan di dunia nyata dan bagaimana kualitas perangkat lunak dijaga melalui pendekatan yang sistematis.

1.2 Permasalahan Kerja Praktek

Dalam pelaksanaan kerja praktek ini, terdapat beberapa permasalahan utama yang melatarbelakangi perlunya dilakukan pengujian fungsional pada Aplikasi Sahabatku. Adapun rumusan masalah tersebut adalah sebagai berikut:

1. Bagaimana cara memastikan bahwa seluruh fungsi utama pada Aplikasi Sahabatku—seperti login, input data, navigasi, hingga tampilan informasi—berjalan sesuai kebutuhan dan tidak menimbulkan error ketika digunakan?
2. Bagaimana menerapkan metode automatic testing menggunakan Selenium untuk menguji fitur-fitur pada website *sahabatku.hdnmetatech.com* secara lebih cepat, konsisten, dan akurat dibandingkan pengujian manual?

1.3 Batasan Kerja Praktek

Dalam kerja praktek ini, batasan masalah yang digunakan adalah sebagai berikut:

1. Pengujian hanya difokuskan pada fungsi-fungsi utama Aplikasi Sahabatku, yang meliputi proses login, input data, penyimpanan data, navigasi halaman, dan tampilan informasi pada website *sahabatku.hdnmetatech.com*.
2. Jenis pengujian yang dilakukan terbatas pada pengujian fungsional (functional testing) sehingga aspek non-fungsional seperti performa, keamanan, usability, dan load testing tidak menjadi bagian dari analisis.
3. Proses pengujian dilakukan menggunakan metode automatic testing berbasis Selenium, sehingga hasil pengujian sepenuhnya bergantung pada skenario uji (test case) yang dibuat. Fitur yang tidak disertakan dalam skenario uji tidak akan diuji.
4. Lingkungan pengujian hanya dilakukan pada perangkat komputer dan browser tertentu, sehingga hasil pengujian mungkin berbeda jika dijalankan pada perangkat atau browser lain di luar konfigurasi yang digunakan selama kerja praktek.

5. Tidak dilakukan perubahan atau modifikasi terhadap kode sumber aplikasi, sehingga pengujian dilakukan sepenuhnya berdasarkan kondisi aplikasi yang sedang berjalan saat periode kerja praktek berlangsung.

1.4 Tujuan Kerja Praktek

Berdasarkan permasalahan yang telah dirumuskan, kerja praktek ini memiliki beberapa tujuan sebagai berikut:

1. Memastikan bahwa fungsi-fungsi utama pada Aplikasi Sahabatku, seperti proses login, input dan penyimpanan data, navigasi halaman, serta tampilan informasi pada website *sahabatku.hdnmetatech.com*, berjalan sesuai dengan kebutuhan dan alur yang telah ditetapkan.
2. Menerapkan metode pengujian fungsional menggunakan automatic testing berbasis Selenium untuk menguji fitur-fitur Aplikasi Sahabatku, sehingga proses pengujian dapat dilakukan secara lebih cepat, konsisten, dan mengurangi potensi kesalahan yang biasa terjadi pada pengujian manual.
3. Menyusun dan menjalankan skenario pengujian (test case) dalam bentuk skrip otomatis, sehingga setiap alur penggunaan yang diuji terdokumentasi dengan jelas dan dapat dijalankan kembali ketika dibutuhkan, misalnya saat ada perubahan atau penambahan fitur pada aplikasi.
4. Mengidentifikasi dan mendokumentasikan bug, error, atau ketidaksesuaian fungsi yang ditemukan selama proses pengujian otomatis, sebagai bahan evaluasi bagi pengembang dalam melakukan perbaikan dan penyempurnaan aplikasi.
5. Memberikan rekomendasi perbaikan dan peningkatan kualitas Aplikasi Sahabatku berdasarkan hasil pengujian, sehingga aplikasi dapat digunakan secara lebih stabil, nyaman, dan andal oleh pengguna.

1.5 Manfaat Kerja Praktek

Pelaksanaan kerja praktek dengan topik pengujian fungsional menggunakan automatic testing pada Aplikasi Sahabatku memberikan sejumlah manfaat, baik yang bersifat nyata (tangible) maupun tidak berwujud (intangible), bagi berbagai pihak yang terlibat. Adapun manfaat tersebut dapat diuraikan sebagai berikut:

1. Bagi Mahasiswa (Peserta Kerja Praktek)

Tangible:

1. Mahasiswa memperoleh laporan kerja praktek yang terdokumentasi secara sistematis sebagai salah satu syarat akademik.
2. Tersusunnya script automatic testing berbasis Selenium yang dapat digunakan sebagai portofolio teknis maupun bahan rujukan di masa mendatang.
3. Terciptanya dokumen hasil pengujian (test case, hasil eksekusi, dan rekap bug) yang menunjukkan kemampuan mahasiswa dalam melakukan pengujian perangkat lunak secara langsung di lingkungan kerja nyata.

Intangible:

1. Mahasiswa mendapatkan pengalaman praktis dalam menerapkan konsep pengujian fungsional dan automatic testing yang sebelumnya hanya dipelajari secara teori di kelas.
2. Meningkatnya pemahaman dan kepercayaan diri mahasiswa dalam berkomunikasi dengan tim pengembang, menganalisis masalah, serta mengambil keputusan teknis terkait kualitas perangkat lunak.
3. Terbentuknya kebiasaan berpikir kritis dan teliti, terutama dalam menyusun skenario uji dan mengevaluasi hasil pengujian.

2. Bagi Perusahaan – PT Herdina Metatech Sinergi

Tangible:

1. Perusahaan memperoleh hasil pengujian fungsional yang lebih terstruktur terhadap Aplikasi Sahabatku dalam bentuk laporan dan dokumentasi.

2. Adanya daftar bug, error, atau ketidaksesuaian fungsi beserta rekomendasi perbaikan yang dapat langsung ditindaklanjuti oleh tim pengembang.
3. Perusahaan mendapatkan script pengujian otomatis yang dapat dijalankan kembali ketika terjadi pembaruan sistem, sehingga dapat menghemat waktu pada proses pengujian di kemudian hari.

Intangible:

1. Perusahaan mendapatkan masukan objektif mengenai kualitas aplikasi dari sudut pandang pihak eksternal (mahasiswa), sehingga dapat menjadi bahan evaluasi internal.
2. Penguatan budaya kualitas (quality awareness) di lingkungan kerja, khususnya terkait pentingnya pengujian fungsional dan pemanfaatan automatic testing dalam pengembangan perangkat lunak.
3. Terjalannya hubungan kemitraan yang baik dengan perguruan tinggi, yang berpotensi mendukung kerja sama di bidang penelitian atau rekrutmen tenaga kerja di masa depan.

3. Bagi Pengguna Aplikasi Sahabatku

Tangible:

1. Pengguna berpotensi memperoleh versi aplikasi yang lebih stabil, dengan jumlah error yang berkurang setelah hasil pengujian diterapkan dalam proses perbaikan.
2. Meningkatnya kualitas layanan sistem, misalnya proses login yang lebih lancar, data tersimpan dengan benar, dan tampilan informasi yang sesuai.

Intangible:

1. Pengguna akan merasakan kenyamanan dan kepercayaan yang lebih tinggi saat menggunakan Aplikasi Sahabatku karena gangguan teknis dapat diminimalkan.
2. Terbentuk persepsi bahwa aplikasi dikelola secara serius dan profesional karena melalui proses pengujian yang jelas.

4. Bagi Program Studi dan Universitas

Tangible:

1. Program studi memperoleh contoh kerja praktek yang relevan dengan dunia industri, khususnya di bidang pengujian perangkat lunak dan otomatisasi.
2. Hasil kerja praktek dapat dijadikan bahan referensi atau dokumen pendukung dalam pengembangan kurikulum, terutama pada mata kuliah terkait rekayasa perangkat lunak dan software testing.

Intangible:

1. Meningkatnya citra program studi dan universitas di mata mitra industri karena mahasiswa mampu berkontribusi secara nyata dalam proyek pengembangan sistem.
2. Adanya penguatan link and match antara materi perkuliahan dengan kebutuhan dunia kerja, sehingga lulusan diharapkan lebih siap terjun ke industri.

5. Bagi Dunia Pengembangan Perangkat Lunak Secara Umum

Tangible:

1. Menambah contoh penerapan automatic testing menggunakan Selenium pada studi kasus nyata di lingkungan industri, yang dapat dijadikan rujukan bagi pihak lain.

Intangible:

1. Mendorong kesadaran lebih luas tentang pentingnya pengujian otomatis dalam menjamin kualitas aplikasi berbasis web, bukan hanya mengandalkan pengujian manual.

1.6 Sistematika Penulisan

1. Font yang digunakan adalah Times New Roman dengan ukuran 12.
2. Jarak spasi pada laporan KP adalah 1,5.
3. Ukuran kertas yang digunakan adalah A4 dengan ketentuan minimal 50 halaman.

4. Ukuran margin mengikuti aturan, yaitu right, top, dan bottom 3 cm, serta left 4 cm.
5. Halaman Sampul sampai Daftar Isi menggunakan nomor halaman dengan angka romawi kecil (i, ii, iii, ...) dan ditempatkan di sudut kanan bawah.
6. Halaman Pendahuluan sampai Daftar Pustaka menggunakan nomor halaman angka arab (1, 2, 3, ...) dan ditempatkan di sudut kanan atas, kecuali halaman yang memuat judul bab ditempatkan di tengah bagian bawah halaman.

BAB 2

TINJAUAN PUSTAKA

2.1 Profil Organisasi



Gambar 2. 1. Logo Instansi

PT Herdina Metatech Sinergi merupakan sebuah perusahaan yang bergerak di bidang pengembangan teknologi informasi dan layanan digital. Perusahaan ini berdiri dari kebutuhan akan solusi teknologi yang lebih terarah, modern, dan mampu mengikuti perkembangan digital yang semakin cepat. Pada awal berdirinya, perusahaan hanya berfokus pada layanan pengembangan website dan sistem informasi skala kecil. Namun seiring meningkatnya kebutuhan pasar, ruang lingkup layanan perusahaan mulai berkembang.

Dalam perjalanannya, PT Herdina Metatech Sinergi terus menyesuaikan diri dengan tuntutan industri, terutama dalam hal inovasi teknologi dan peningkatan kualitas layanan. Perusahaan mulai memperluas bidang usahanya ke pengembangan aplikasi berbasis web dan mobile, integrasi sistem, hingga layanan konsultasi teknologi. Langkah ini dilakukan sebagai upaya untuk memberikan solusi yang lebih lengkap bagi instansi pemerintah, lembaga pendidikan, maupun sektor swasta yang membutuhkan sistem digital yang efisien.

Perkembangan perusahaan juga ditandai dengan dibentuknya berbagai divisi teknis yang menangani kebutuhan klien secara lebih spesifik, seperti divisi pengembangan sistem, pengujian perangkat lunak, desain antarmuka, dan dukungan teknis. Salah satu produk yang dikembangkan adalah Aplikasi Sahabatku, sebuah platform berbasis web yang dirancang untuk mendukung proses

pelayanan dan manajemen data secara digital. Kehadiran aplikasi ini menjadi bukti komitmen perusahaan untuk menghadirkan teknologi yang mudah digunakan dan dapat membantu proses kerja di berbagai sektor.

Dengan pengalaman yang terus bertambah, PT Herdina Metatech Sinergi berupaya menjaga kualitas layanan melalui pengembangan sumber daya manusia, perbaikan sistem kerja, serta penerapan standar yang lebih profesional dalam setiap proyek. Perusahaan berharap dapat terus berkontribusi dalam mendukung transformasi digital di Indonesia dengan menyediakan solusi teknologi yang relevan, efisien, dan berkelanjutan.

2.2 Konsep Kualitas Perangkat Lunak

Kualitas perangkat lunak pada dasarnya menggambarkan sejauh mana sebuah aplikasi, alat, atau sistem mampu menjalankan fungsi yang telah dirancang secara optimal. Sebuah perangkat lunak dikatakan berkualitas apabila mampu bekerja secara stabil, memberikan kemudahan bagi penggunaanya, serta dapat dipelihara dan dikembangkan dalam jangka waktu yang panjang. Kualitas ini tidak hanya dilihat dari hasil akhirnya saja, tetapi juga dari proses penggunaan, pengujian, serta kemampuan perangkat lunak tersebut dalam beradaptasi terhadap perubahan kebutuhan pengguna.

Dalam konteks pengembangan aplikasi berbasis web, proses pengujian memegang peranan yang sangat penting. Pengujian bertujuan untuk memastikan bahwa sistem berjalan sesuai dengan kebutuhan yang telah ditentukan serta meminimalkan kesalahan sebelum aplikasi digunakan oleh pengguna secara luas. Salah satu pendekatan yang banyak digunakan saat ini adalah pengujian otomatis (automated testing). Pengujian otomatis dianggap lebih efisien dibandingkan pengujian manual karena mampu menghemat waktu, mengurangi kesalahan akibat faktor manusia, serta memungkinkan pengujian dilakukan secara berulang dengan hasil yang konsisten.

Selenium merupakan salah satu alat pengujian otomatis yang paling populer dan banyak digunakan untuk aplikasi berbasis web. Selenium membantu pengembang dan penguji dalam menjalankan skenario pengujian secara otomatis melalui browser, sehingga proses pengujian dapat dilakukan dengan lebih cepat dan sistematis. Untuk memahami kualitas Selenium sebagai perangkat lunak pengujian, diperlukan peninjauan dari beberapa aspek penting yang berkaitan dengan karakteristik kualitas perangkat lunak itu sendiri.

Aspek pertama yang dapat dilihat adalah kecocokan fungsional (functional suitability). Selenium dirancang untuk menjalankan berbagai perintah otomatis yang umum dibutuhkan dalam pengujian aplikasi web, seperti mengisi formulir, menekan tombol, berpindah halaman, serta memeriksa elemen tertentu pada tampilan web. Fitur-fitur tersebut memungkinkan Selenium untuk mensimulasikan aktivitas pengguna secara nyata. Dengan kemampuan ini, Selenium dinilai mampu memenuhi kebutuhan dasar hingga lanjutan dalam pengujian fungsional aplikasi web. Selama skrip pengujian disusun dengan benar dan sesuai dengan alur aplikasi, Selenium dapat menjalankan pengujian secara berulang tanpa mengalami kesalahan yang biasanya sering terjadi pada pengujian manual, seperti kelalaian atau inkonsistensi penguji.

Aspek berikutnya adalah kemudahan penggunaan (usability). Dari sisi ini, Selenium memang menuntut pengguna untuk memiliki pemahaman dasar mengenai bahasa pemrograman tertentu, seperti Java, Python, atau JavaScript. Hal ini menjadi tantangan tersendiri bagi pengguna yang belum memiliki latar belakang teknis. Namun demikian, Selenium didukung oleh dokumentasi yang cukup lengkap, komunitas pengguna yang besar, serta banyak contoh implementasi yang tersedia secara bebas. Kondisi ini membantu pengguna, terutama mahasiswa atau pengembang pemula, untuk mempelajari Selenium secara bertahap. Dengan memahami konsep dasar seperti locator elemen, alur skrip, dan struktur pengujian, pengguna sudah dapat membuat skrip pengujian sederhana sesuai dengan kebutuhan aplikasi yang diuji.

Keandalan dan stabilitas (reliability) juga menjadi aspek penting dalam menilai kualitas perangkat lunak pengujian seperti Selenium. Secara umum, Selenium mampu berjalan dengan stabil selama aplikasi web yang diuji tidak mengalami perubahan besar pada struktur atau tampilan antarmukanya. Namun, dalam praktiknya, aplikasi web sering mengalami pembaruan, baik dari sisi desain maupun struktur elemen. Perubahan tersebut dapat menyebabkan skrip pengujian gagal dijalankan sehingga memerlukan penyesuaian ulang. Oleh karena itu, pemeliharaan skrip menjadi bagian yang tidak terpisahkan dalam penggunaan Selenium. Meskipun demikian, hal ini masih dapat dianggap wajar karena perubahan aplikasi memang menuntut adanya penyesuaian pada proses pengujian.

Dari sisi kinerja (performance efficiency), Selenium memberikan kontribusi yang cukup signifikan dalam mempercepat proses pengujian. Dengan pengujian otomatis, banyak skenario dapat dijalankan dalam waktu yang relatif singkat dibandingkan pengujian manual. Penguji dapat menjalankan pengujian pada berbagai browser dan lingkungan secara bergantian tanpa harus melakukan langkah yang sama berulang kali. Namun, kinerja Selenium juga dipengaruhi oleh beberapa faktor, seperti spesifikasi perangkat yang digunakan, konfigurasi sistem, jenis browser, serta kondisi jaringan. Pada aplikasi web dengan elemen dinamis yang kompleks, proses pengujian dapat berjalan lebih lambat dan membutuhkan penanganan khusus dalam penulisan skrip.

Aspek terakhir yang tidak kalah penting adalah kemudahan pemeliharaan (maintainability). Skrip pengujian yang dibuat menggunakan Selenium relatif mudah untuk diperbarui apabila disusun dengan struktur yang rapi dan konsisten. Pengguna dapat menambahkan skenario baru, memperbaiki langkah pengujian yang bermasalah, atau menghapus bagian tertentu tanpa harus menulis ulang seluruh skrip. Fleksibilitas ini sangat membantu dalam proyek pengembangan aplikasi yang bersifat dinamis dan sering mengalami perubahan. Dengan pola penulisan skrip yang baik, Selenium dapat digunakan dalam jangka panjang sebagai alat bantu pengujian yang andal.

Secara keseluruhan, kualitas perangkat lunak Selenium dapat dikatakan cukup baik dalam mendukung kebutuhan pengujian aplikasi web. Selenium mampu menyediakan pengujian yang cepat, fleksibel, serta dapat dijalankan pada berbagai jenis browser dan platform. Meskipun membutuhkan pemeliharaan skrip secara berkala dan pemahaman teknis tertentu, keunggulan Selenium sebagai alat yang bersifat terbuka, gratis, dan memiliki dukungan komunitas yang luas menjadikannya tetap relevan dan banyak digunakan hingga saat ini. Oleh karena itu, Selenium dapat dianggap sebagai salah satu solusi pengujian otomatis yang efektif dan layak digunakan dalam proses pengembangan perangkat lunak berbasis web.

2.3 Metode Pengujian Perangkat Lunak

Dalam proses memastikan kualitas perangkat lunak, pemilihan metode pengujian yang tepat menjadi salah satu tahapan penting yang tidak dapat diabaikan. Metode pengujian berfungsi sebagai alat untuk menilai apakah sistem yang dikembangkan telah berjalan sesuai dengan kebutuhan yang direncanakan sejak awal. Tanpa adanya metode pengujian yang jelas, pengembang akan kesulitan mengetahui apakah aplikasi sudah layak digunakan atau masih memiliki banyak kekurangan yang dapat mengganggu pengguna. Oleh karena itu, pengujian perangkat lunak tidak hanya bertujuan untuk menemukan kesalahan, tetapi juga untuk memberikan gambaran menyeluruh mengenai tingkat kesiapan sistem sebelum digunakan secara luas.

Metode pengujian juga berperan dalam memastikan bahwa setiap fungsi yang ada di dalam aplikasi dapat berjalan dengan baik dan saling terintegrasi. Pengujian membantu mengidentifikasi kesalahan sejak dini, sehingga perbaikan dapat dilakukan sebelum sistem diterapkan di lingkungan nyata. Selain itu, hasil pengujian dapat menjadi bahan evaluasi bagi pengembang untuk meningkatkan kualitas aplikasi, baik dari sisi fungsionalitas, kemudahan penggunaan, maupun kestabilan sistem. Dalam pengembangan aplikasi berbasis web, pengujian menjadi

semakin penting karena aplikasi harus mampu berjalan pada berbagai kondisi dan perangkat yang berbeda.

Dalam konteks penggunaan Selenium sebagai alat bantu pengujian otomatis, metode pengujian yang diterapkan umumnya berfokus pada cara kerja aplikasi dari sudut pandang pengguna akhir. Selenium dirancang untuk meniru interaksi pengguna dengan aplikasi web, seperti membuka halaman, mengisi data, menekan tombol, hingga memeriksa hasil yang ditampilkan oleh sistem. Dengan pendekatan ini, pengujian yang dilakukan menjadi lebih realistis karena menyerupai kondisi penggunaan sebenarnya. Hal ini membantu pengembang dalam memahami bagaimana aplikasi bekerja ketika digunakan langsung oleh pengguna.

Salah satu metode pengujian yang paling umum diterapkan dengan bantuan Selenium adalah pengujian fungsional (functional testing). Pengujian fungsional bertujuan untuk memastikan bahwa setiap fitur dalam aplikasi telah berjalan sesuai dengan kebutuhan dan spesifikasi yang telah ditentukan. Pada metode ini, pengujian tidak terlalu memperhatikan bagaimana sistem dibangun secara internal, melainkan lebih fokus pada hasil keluaran dari setiap fungsi. Selama hasil yang ditampilkan sesuai dengan yang diharapkan, maka fungsi tersebut dianggap berjalan dengan baik.

Dalam pengujian fungsional menggunakan Selenium, penguji biasanya menyusun beberapa skenario pengujian berdasarkan alur penggunaan aplikasi. Contoh skenario yang sering digunakan antara lain proses login pengguna, pengisian formulir, pengiriman data, serta pengecekan tampilan halaman tertentu. Selenium sangat cocok digunakan untuk metode ini karena mampu menjalankan skrip pengujian secara otomatis dan berulang. Dengan demikian, penguji tidak perlu melakukan langkah yang sama secara manual berkali-kali, sehingga waktu dan tenaga dapat dihemat.

Selain efisiensi waktu, pengujian fungsional dengan Selenium juga memberikan hasil yang lebih konsisten. Kesalahan yang sering terjadi pada pengujian manual, seperti kelalaian penguji atau perbedaan cara pengujian, dapat diminimalkan. Selama skrip pengujian ditulis dengan benar, Selenium akan

menjalankan pengujian dengan cara yang sama setiap kali dijalankan. Hal ini sangat membantu dalam memastikan bahwa fungsi-fungsi utama aplikasi tetap berjalan dengan baik, terutama ketika aplikasi terus dikembangkan.

Metode pengujian lain yang sering digunakan adalah pengujian regresi (regression testing). Pengujian regresi dilakukan ketika sistem mengalami perubahan, baik berupa penambahan fitur baru, perbaikan bug, maupun perubahan tampilan antarmuka. Tujuan utama dari pengujian ini adalah untuk memastikan bahwa perubahan yang dilakukan tidak menimbulkan masalah baru pada fungsi yang sebelumnya sudah berjalan dengan baik. Dalam praktik pengembangan perangkat lunak, perubahan sistem hampir selalu terjadi, sehingga pengujian regresi menjadi bagian yang sangat penting.

Selenium sangat membantu dalam proses pengujian regresi karena skrip pengujian yang telah dibuat sebelumnya dapat digunakan kembali. Penguji tidak perlu menyusun ulang seluruh skenario pengujian dari awal, melainkan cukup menjalankan skrip yang sudah ada untuk memastikan semua fungsi lama masih bekerja dengan baik. Dengan cara ini, pengujian regresi dapat dilakukan dengan lebih cepat dan efisien dibandingkan pengujian manual. Selain itu, hasil pengujian juga lebih mudah dibandingkan karena dijalankan dengan skenario yang sama.

Pengujian kompatibilitas juga menjadi metode penting dalam memastikan kualitas aplikasi web. Pengujian ini bertujuan untuk memastikan bahwa aplikasi dapat berjalan dengan baik pada berbagai browser dan perangkat yang berbeda. Dalam kondisi nyata, pengguna dapat mengakses aplikasi menggunakan browser yang beragam, seperti Google Chrome, Mozilla Firefox, atau Microsoft Edge. Setiap browser memiliki karakteristik dan cara rendering yang berbeda, sehingga aplikasi perlu diuji agar tampil dan berfungsi secara konsisten.

Selenium mendukung pengujian kompatibilitas karena dapat dijalankan pada berbagai browser dengan konfigurasi yang relatif mudah. Penguji dapat menjalankan skrip yang sama pada browser yang berbeda untuk melihat apakah terdapat perbedaan perilaku atau tampilan. Dengan adanya pengujian ini, pengembang dapat mengetahui sejak awal jika terdapat masalah kompatibilitas dan

segera melakukan perbaikan. Hal ini penting untuk menjaga kenyamanan pengguna dan mencegah terjadinya kesalahan saat aplikasi digunakan di lingkungan yang berbeda.

Selain pengujian fungsional, regresi, dan kompatibilitas, Selenium juga dapat dimanfaatkan untuk melakukan pengujian kinerja sederhana (basic performance testing). Meskipun Selenium bukan alat khusus untuk pengujian beban atau stres, skrip Selenium tetap dapat digunakan untuk mengamati respons aplikasi secara umum. Misalnya, penguji dapat menjalankan skenario tertentu secara berulang untuk melihat apakah waktu respons aplikasi tetap stabil atau justru mengalami penurunan.

Pengujian kinerja sederhana ini dapat memberikan gambaran awal mengenai bagian aplikasi yang mungkin memerlukan perbaikan, terutama terkait dengan waktu muat halaman atau proses tertentu yang dirasa lambat. Informasi ini dapat menjadi bahan evaluasi bagi pengembang sebelum melakukan pengujian kinerja yang lebih mendalam menggunakan alat khusus. Dengan demikian, Selenium tetap memiliki peran meskipun bukan alat utama untuk pengujian performa.

Secara keseluruhan, metode pengujian perangkat lunak yang diterapkan dengan bantuan Selenium mendukung tujuan utama dalam memastikan kualitas aplikasi. Pengujian yang dilakukan membantu memastikan bahwa aplikasi berjalan sesuai dengan kebutuhan, stabil digunakan, serta mampu memberikan pengalaman yang baik bagi pengguna. Dengan memanfaatkan Selenium sebagai alat bantu, proses pengujian dapat dilakukan secara lebih terstruktur dan sistematis.

Selain itu, penggunaan Selenium juga meningkatkan efisiensi dan konsistensi dalam proses pengujian. Hasil pengujian yang diperoleh menjadi lebih mudah dianalisis dan dipertanggungjawabkan karena dilakukan berdasarkan skenario yang jelas dan dapat dijalankan ulang. Oleh karena itu, Selenium dapat dianggap sebagai alat yang efektif dalam mendukung berbagai metode pengujian perangkat lunak, khususnya pada aplikasi berbasis web yang terus mengalami perkembangan dan perubahan.

2.4 Perangkat Bantu Pengujian Perangkat Lunak

Dalam proses pengujian perangkat lunak, penggunaan perangkat bantu sangat diperlukan agar hasil pengujian lebih terarah, konsisten, dan mudah dianalisis. Perangkat bantu ini tidak hanya mempermudah pekerjaan penguji, tetapi juga membantu memastikan bahwa setiap skenario pengujian dapat dijalankan secara berulang tanpa mengurangi ketelitian. Mengingat metode pengujian yang digunakan pada penelitian ini melibatkan pengujian fungsional dan regresi, maka keberadaan alat bantu seperti Selenium menjadi bagian yang cukup penting dalam keseluruhan proses.

Selenium sendiri merupakan alat bantu pengujian berbasis web yang memungkinkan penguji menjalankan skenario secara otomatis. Melalui skrip yang disusun, Selenium dapat menirukan tindakan pengguna seperti mengklik tombol, mengisi formulir, hingga memeriksa tampilan halaman. Kemampuan inilah yang membuat Selenium sangat membantu terutama dalam pengujian fungsional dan regresi yang membutuhkan proses berulang. Selain itu, Selenium dapat dijalankan di berbagai browser sehingga sesuai untuk kebutuhan pengujian kompatibilitas yang sebelumnya telah dijelaskan.

Selain Selenium WebDriver sebagai alat utama untuk mengeksekusi perintah otomatis, ada beberapa perangkat pendukung lain yang biasanya ikut digunakan agar proses pengujian menjadi lebih mudah dan terstruktur. Misalnya, browser developer tools seperti yang ada di Chrome atau Firefox. Tools ini sering digunakan untuk melihat struktur elemen HTML, memeriksa respons halaman, hingga mengetahui apakah ada error yang muncul pada sisi klien. Informasi tersebut sangat membantu penguji saat menyusun skrip Selenium supaya lebih tepat mengenai elemen yang ingin diuji.

Perangkat bantu lain yang juga bermanfaat adalah framework pengelola skrip, misalnya TestNG atau JUnit jika pengujian menggunakan bahasa Java. Framework tersebut membantu penguji dalam menyusun, mengatur, dan menjalankan skrip pengujian dalam bentuk yang lebih rapi. Penguji bisa mengatur prioritas pengujian, menandai bagian yang gagal, hingga membuat laporan hasil

pengujian secara otomatis. Walaupun tidak selalu wajib digunakan, keberadaannya dapat mempercepat proses terutama saat jumlah skrip pengujian semakin banyak.

Selain itu, penggunaan version control seperti Git juga sering diterapkan, terutama jika skrip pengujian dikerjakan oleh lebih dari satu orang. Dengan adanya version control, setiap perubahan pada skrip dapat dicatat, dibandingkan, dan dipulihkan jika diperlukan. Hal ini membantu menjaga konsistensi skrip pengujian dan meminimalkan risiko kesalahan akibat modifikasi yang tidak sengaja.

Secara keseluruhan, perangkat bantu pengujian berfungsi untuk memastikan bahwa proses pengujian berjalan dengan lebih efisien dan hasilnya dapat dipertanggungjawabkan. Dalam penelitian ini, kombinasi antara Selenium, browser developer tools, dan framework pendukung menjadi susunan alat yang membantu memperjelas alur pengujian. Dengan demikian, setiap metode pengujian yang digunakan dapat dijalankan secara lebih terkontrol, sesuai dengan tujuan untuk memperoleh kualitas perangkat lunak yang baik dan dapat diandalkan.

BAB 3

METODOLOGI PELAKSANAAN KERJA PRAKTEK

3.1 Tahapan Kerja Praktek

Tahapan kerja praktek merupakan rangkaian langkah yang dilakukan untuk menjalankan kegiatan pengujian perangkat lunak secara terarah dan sesuai dengan tujuan awal yang telah ditetapkan. Setiap tahapan saling berkaitan satu sama lain, mulai dari proses pengumpulan kebutuhan, penyusunan skenario uji, hingga pelaksanaan pengujian menggunakan alat bantu seperti Selenium. Penjelasan berikut menggambarkan alur kerja yang dilakukan selama pelaksanaan kerja praktek.

Tahap pertama adalah pemahaman kebutuhan sistem. Pada tahap ini, penulis mengidentifikasi fungsi-fungsi utama yang harus diuji berdasarkan dokumentasi dan proses bisnis aplikasi. Proses ini penting karena menjadi dasar dalam menentukan skenario pengujian fungsional dan regresi. Tahap ini juga memastikan bahwa apa yang diuji benar-benar mencerminkan kebutuhan pengguna yang sebenarnya.

Tahap kedua yaitu penyusunan skenario pengujian. Skenario ini berisi urutan langkah yang akan dijalankan selama pengujian, seperti proses login, pengecekan halaman tertentu, atau fitur yang memerlukan interaksi pengguna. Skenario disusun dengan memperhatikan metode pengujian yang digunakan sebelumnya, yaitu pengujian fungsional, regresi, serta kompatibilitas. Hasil dari tahapan ini berupa daftar kasus uji yang siap dieksekusi.

Tahap berikutnya adalah penyusunan skrip otomatis. Pada tahapan ini, Selenium digunakan sebagai alat utama untuk membuat instruksi otomatis berdasarkan skenario yang telah disusun. Setiap langkah pada skenario diterjemahkan ke dalam bentuk kode sehingga Selenium dapat menjalankannya secara otomatis. Pada tahap ini biasanya juga melibatkan penggunaan browser developer tools untuk memastikan elemen halaman dapat dikenali dengan tepat oleh skrip.

Tahap keempat yaitu pelaksanaan pengujian. Setelah skrip siap digunakan, proses pengujian mulai dijalankan pada beberapa browser sesuai kebutuhan pengujian kompatibilitas. Eksekusi dilakukan secara berulang untuk memastikan stabilitas hasil, terutama pada pengujian regresi yang memerlukan pengecekan setelah adanya pembaruan sistem. Di tahap ini, hasil pengujian dicatat untuk dianalisis pada tahap selanjutnya.

Tahap kelima adalah analisis hasil pengujian. Hasil yang muncul dari setiap eksekusi skrip dievaluasi untuk mengetahui apakah fungsi berjalan dengan baik atau terdapat bagian yang perlu diperbaiki. Jika ada ketidaksesuaian, skrip atau skenario akan disesuaikan kembali. Analisis ini juga membantu memahami apakah kualitas perangkat lunak sudah memenuhi standar yang diharapkan.

Tahap terakhir adalah pembuatan laporan kerja praktek. Pada tahap ini, seluruh proses yang telah dilakukan—mulai dari pengenalan sistem, metode pengujian, penyusunan skrip, hingga hasil evaluasi—dirangkum dalam bentuk laporan tertulis. Laporan ini menjadi bukti dokumentasi bahwa pengujian telah dilakukan secara sistematis dan dapat dipertanggungjawabkan.

Secara keseluruhan, tahapan kerja praktek ini menjadi kerangka utama dalam memastikan pengujian perangkat lunak berjalan dengan terstruktur, konsisten, dan sesuai dengan konsep kualitas perangkat lunak yang telah dijelaskan pada bab sebelumnya.

3.2 Analisis Kebutuhan Pengujian Perangkat Lunak

Analisis kebutuhan pengujian dilakukan untuk memastikan bahwa proses pengujian yang dijalankan benar-benar sesuai dengan fungsi dan tujuan dari aplikasi yang diuji. Tahap ini penting karena menjadi dasar dalam menentukan skenario uji, memilih metode pengujian, serta menyusun skrip otomatis yang akan dijalankan menggunakan Selenium. Dengan memahami apa saja yang perlu diuji sejak awal, proses pengujian menjadi lebih terarah dan hasilnya dapat dipertanggungjawabkan.

Langkah pertama dalam analisis kebutuhan adalah mengidentifikasi fungsi utama aplikasi. Setiap fitur yang berhubungan langsung dengan aktivitas pengguna perlu diperiksa, seperti proses login, navigasi halaman, pengisian formulir, serta interaksi dengan tombol atau komponen tertentu. Fungsi-fungsi inilah yang umumnya diuji pada pengujian fungsional, seperti yang telah dijelaskan pada bab sebelumnya. Dengan memahami alur kerja aplikasi secara keseluruhan, penguji dapat menentukan bagian mana yang paling krusial dan wajib diuji terlebih dahulu.

Selanjutnya, analisis kebutuhan juga mencakup identifikasi risiko atau perubahan yang mungkin muncul selama pengembangan. Karena aplikasi dapat mengalami pembaruan tampilan atau penambahan fitur, maka kebutuhan pengujian regresi menjadi penting. Kebutuhan ini muncul agar setiap pembaruan tidak mengganggu fungsi lama yang sebelumnya berjalan dengan baik. Dari sini, bagian-bagian aplikasi yang sensitif terhadap perubahan dapat diberi prioritas pengujian yang lebih tinggi.

Kebutuhan lain yang turut dianalisis adalah kompatibilitas penggunaan. Mengingat setiap pengguna dapat mengakses aplikasi melalui browser yang berbeda, maka pengujian pada beberapa browser menjadi bagian dari kebutuhan pengujian. Kebutuhan ini berkaitan langsung dengan kemampuan Selenium untuk menjalankan skrip di browser yang berbeda, sehingga hasil pengujian tetap konsisten meskipun lingkungannya bermacam-macam.

Analisis kebutuhan juga memperhatikan data uji (test data) yang diperlukan. Data uji harus disiapkan sedemikian rupa agar dapat mewakili penggunaan nyata, baik data yang valid maupun data yang tidak valid. Tujuannya untuk memastikan bahwa sistem mampu menangani berbagai kemungkinan masukan dari pengguna. Data uji yang tepat juga mempermudah proses evaluasi karena hasilnya lebih menggambarkan kondisi penggunaan sebenarnya.

Terakhir, analisis kebutuhan mempertimbangkan ketersediaan perangkat bantu seperti Selenium, browser developer tools, serta framework pendukung lainnya. Kebutuhan ini memastikan bahwa alat yang digunakan memang sesuai

dengan metode pengujian yang telah ditetapkan dan dapat mendukung seluruh rangkaian kegiatan pengujian dari awal hingga akhir.

Dari keseluruhan analisis ini, kebutuhan pengujian perangkat lunak dapat dirumuskan menjadi serangkaian prioritas yang meliputi pengujian fungsi utama, pengujian regresi setelah perubahan, pengecekan kompatibilitas, serta penyusunan data uji yang lengkap. Semua kebutuhan tersebut kemudian digunakan sebagai dasar dalam penyusunan skenario uji pada tahapan berikutnya, sehingga proses pengujian berjalan secara terstruktur dan sejalan dengan tujuan kualitas perangkat lunak yang ingin dicapai.

3.3 Pengembangan Skenario Pengujian Perangkat Lunak

Pengembangan skenario pengujian dilakukan berdasarkan kebutuhan pengujian yang sudah dianalisis sebelumnya. Skenario ini berfungsi sebagai panduan praktis dalam menjalankan setiap langkah pengujian, sehingga hasil yang diperoleh dapat lebih terukur, konsisten, dan mudah dievaluasi. Dengan adanya skenario yang jelas, baik pengujian manual maupun otomatis menggunakan Selenium dapat dilakukan secara lebih terarah.

Langkah awal dalam menyusun skenario adalah menentukan fungsi utama yang akan diuji. Fungsi tersebut dipilih berdasarkan prioritas yang telah dirumuskan pada analisis kebutuhan, terutama fitur yang langsung digunakan oleh pengguna. Misalnya, skenario pengujian dapat mencakup proses login, navigasi antar-halaman, pengisian formulir, validasi input, dan pengecekan tampilan halaman. Setiap fungsi ini dijabarkan dalam bentuk langkah-langkah yang mudah diikuti agar proses pengujian tidak menimbulkan bias.

Setelah fungsi utama diidentifikasi, skenario diperluas dengan pengujian regresi, yaitu skenario yang memastikan perubahan atau pembaruan pada sistem tidak menimbulkan masalah baru. Skenario regresi biasanya menggunakan langkah-langkah yang sama dengan skenario fungsi utama, namun dijalankan setelah adanya pembaruan aplikasi. Hal ini penting untuk menjaga kualitas perangkat lunak tetap stabil, seperti yang telah dijelaskan pada bab sebelumnya.

Pengembangan skenario juga mencakup pengujian kompatibilitas. Dalam skenario ini, langkah pengujian disesuaikan untuk dijalankan di beberapa browser berbeda, seperti Chrome, Firefox, dan Edge. Tujuannya adalah memastikan bahwa hasil pengujian tetap konsisten meskipun kondisi lingkungan sistem berubah. Selenium sangat membantu dalam bagian ini karena mampu menjalankan skrip pada beberapa browser tanpa mengubah struktur skenarionya.

Pada tahap berikutnya, skenario disusun dalam bentuk kasus uji (test case) yang lebih rinci. Setiap kasus uji biasanya memuat tujuan pengujian, langkah yang harus dilakukan, data yang digunakan, kondisi yang diharapkan, serta hasil yang diperoleh. Bentuk penyusunan seperti ini membantu penguji mengetahui apakah setiap langkah berhasil atau tidak, sehingga proses evaluasi bisa dilakukan dengan lebih jelas.

Selain itu, pengembangan skenario juga mempertimbangkan kebutuhan data uji. Data uji yang digunakan mencakup data valid dan data tidak valid untuk memastikan sistem mampu menangani berbagai kemungkinan masukan pengguna. Dengan cara ini, hasil pengujian dapat memberikan gambaran lebih lengkap tentang perilaku aplikasi.

Setelah seluruh skenario dirumuskan, langkah terakhir adalah menyesuaikan skenario dengan kemampuan perangkat bantu, terutama Selenium. Setiap langkah dalam skenario diterjemahkan ke dalam bentuk perintah otomatis, sehingga proses pengujian bisa dijalankan secara berulang tanpa harus mengubah skenarionya. Penyelarasan ini penting agar pelaksanaan pengujian berjalan lancar dan hasilnya sesuai dengan tujuan awal.

Secara keseluruhan, pengembangan skenario pengujian menjadi fondasi utama dalam memastikan bahwa seluruh proses pengujian berjalan terstruktur, konsisten, dan mudah ditelusuri. Skenario inilah yang nantinya digunakan pada tahap implementasi untuk memastikan kualitas perangkat lunak dapat dievaluasi secara objektif sesuai standar yang telah ditetapkan.

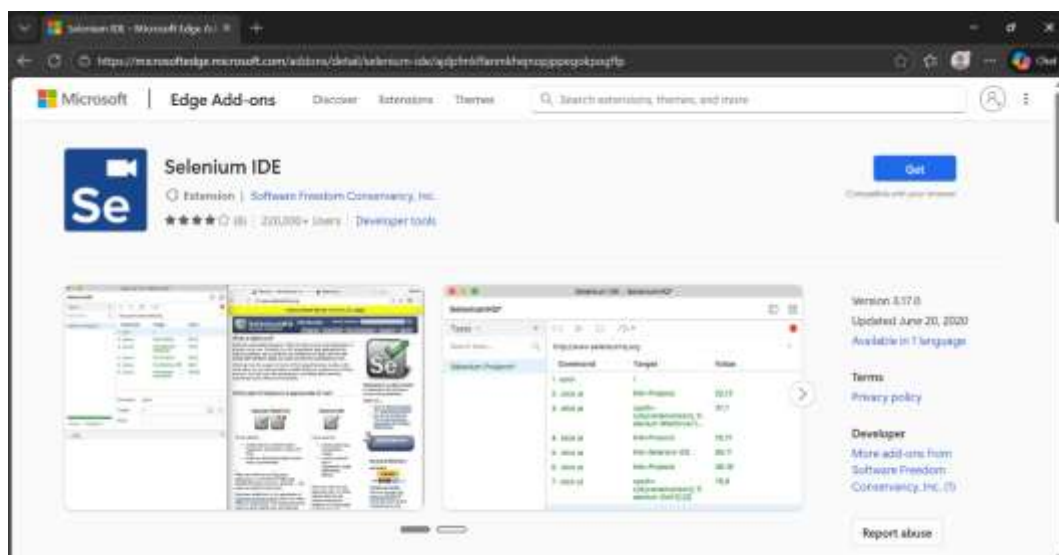
BAB 4

PELAKSANAAN KERJA PRAKTEK

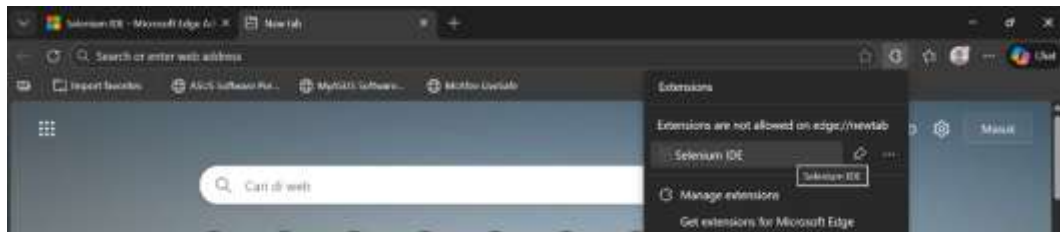
4.1 Persiapan Pengujian Perangkat Lunak

Tahap persiapan pengujian perangkat lunak merupakan langkah awal yang dilakukan sebelum proses pengujian fungsional dijalankan. Persiapan ini bertujuan untuk memastikan bahwa seluruh kebutuhan pengujian telah tersedia sehingga proses pengujian dapat berjalan secara terstruktur, sistematis, dan menghasilkan data yang akurat.

Langkah pertama dalam tahap ini adalah menyiapkan lingkungan pengujian. Pengujian dilakukan menggunakan peramban Microsoft Edge yang telah terpasang ekstensi Selenium IDE. Pemasangan Selenium IDE dilakukan melalui halaman resmi Microsoft Edge Add-ons untuk memastikan ekstensi yang digunakan merupakan versi resmi dan stabil. Setelah ekstensi berhasil dipasang, Selenium IDE dapat diakses melalui menu ekstensi pada browser.

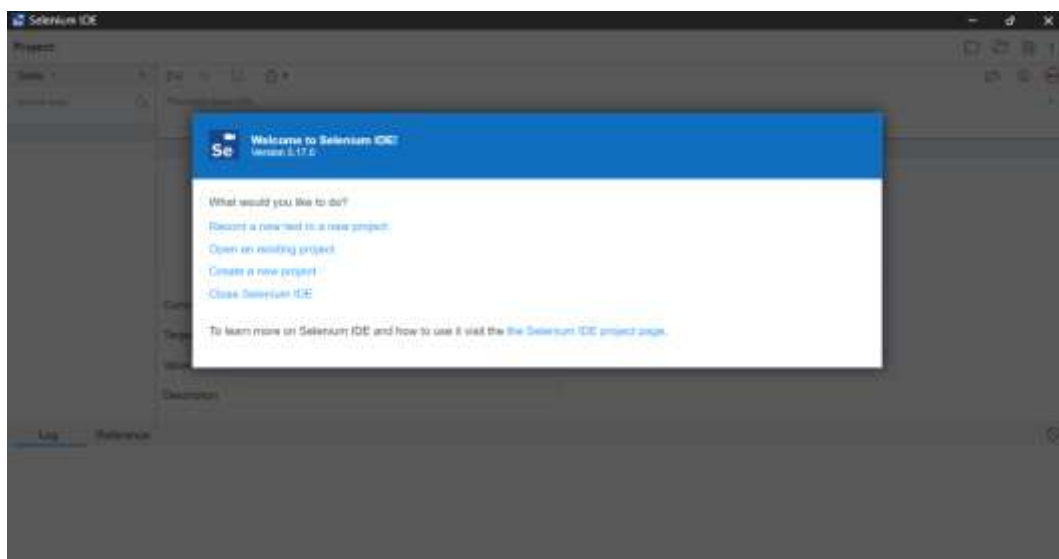


Gambar 4. 1. Install Selenium IDE

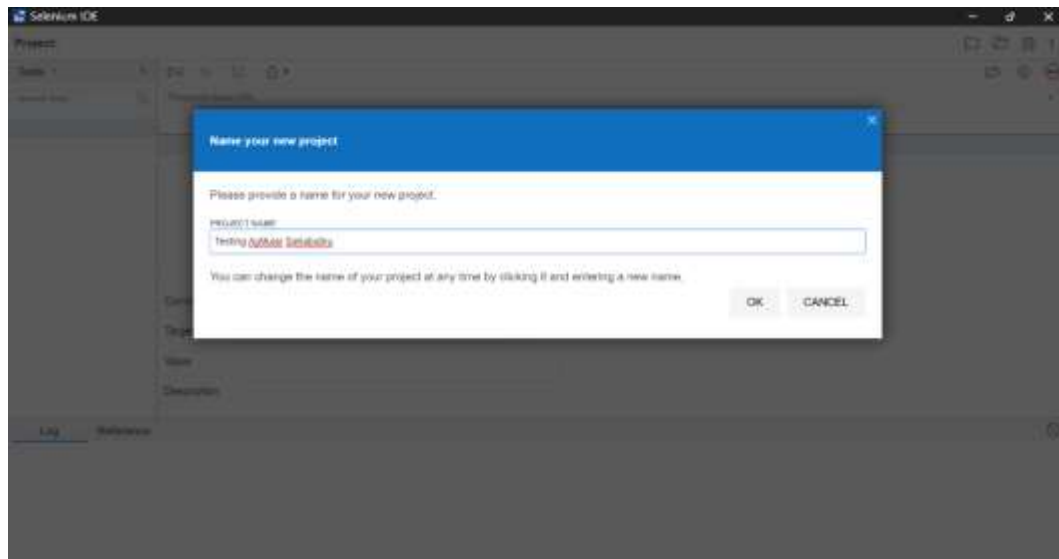


Gambar 4. 2. Akses Selenium IDE

Setelah Selenium IDE dijalankan, langkah berikutnya adalah membuat proyek pengujian baru. Pada tahap ini, pengguna memilih opsi pembuatan proyek baru dan memberikan nama proyek sesuai dengan objek yang diuji, yaitu *“Testing Aplikasi Sahabatku”*. Pemberian nama proyek dilakukan agar proses pengujian terdokumentasi dengan jelas dan mudah dikenali pada tahap analisis maupun pelaporan hasil pengujian.

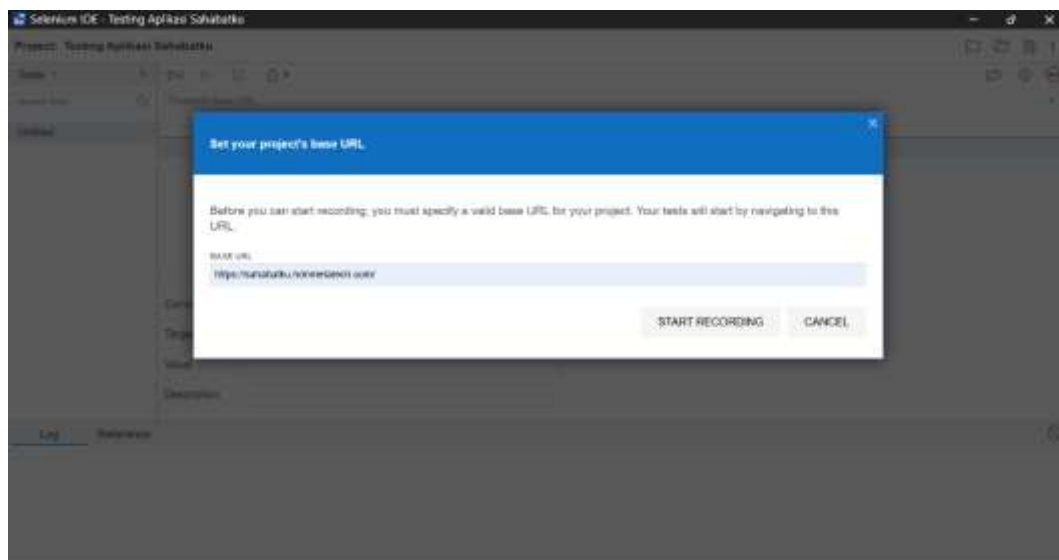


Gambar 4. 3. Membuat Proyek Baru



Gambar 4. 4. Penamaan Projek

Selanjutnya, dilakukan pengaturan awal proyek pengujian. Pengaturan ini meliputi penentuan *base URL* atau alamat dasar aplikasi yang akan diuji. Base URL digunakan sebagai acuan Selenium IDE dalam menjalankan setiap skenario pengujian sehingga seluruh perintah pengujian dapat dieksekusi pada lingkungan aplikasi yang sama dan konsisten.



Gambar 4. 5. Base URL

Setelah proyek dan pengaturan dasar selesai dibuat, tahap berikutnya adalah memastikan aplikasi yang akan diuji berada dalam kondisi siap diuji. Aplikasi Sahabatku diakses melalui browser untuk memastikan tidak terdapat gangguan akses, kesalahan tampilan, maupun error sistem sebelum proses perekaman pengujian dilakukan. Langkah ini penting untuk menghindari kesalahan pengujian yang disebabkan oleh faktor di luar fungsi aplikasi.

Tahap terakhir dalam persiapan pengujian adalah memastikan fitur perekaman (record) pada Selenium IDE dapat berjalan dengan baik. Fitur ini digunakan untuk merekam setiap interaksi pengguna dengan aplikasi, seperti klik menu, perpindahan halaman, dan input data. Dengan memastikan fitur perekaman siap digunakan, proses pengujian fungsional dapat dilakukan secara efektif dan seluruh aktivitas pengujian dapat terdokumentasi secara otomatis.

Melalui tahapan persiapan ini, proses pengujian perangkat lunak dapat dilakukan secara lebih terarah dan sistematis, sehingga hasil pengujian yang diperoleh dapat digunakan sebagai dasar evaluasi terhadap fungsionalitas Aplikasi Sahabatku.

4.2 Implementasi Pengujian Perangkat Lunak

Setelah seluruh persiapan pengujian selesai dilakukan, tahap selanjutnya adalah mengimplementasikan skenario pengujian yang telah disusun sebelumnya. Implementasi ini bertujuan untuk memastikan bahwa setiap fungsi pada aplikasi dapat berjalan sesuai harapan dan memenuhi standar kualitas perangkat lunak yang telah dijelaskan pada bab sebelumnya. Pada tahap ini, Selenium digunakan sebagai alat utama untuk menjalankan skrip otomatis sesuai langkah-langkah yang sudah dirancang.

Proses pelaksanaan dimulai dengan menjalankan skrip pengujian fungsional. Pada bagian ini, setiap skenario diujikan secara berurutan mulai dari fungsi dasar

seperti login, navigasi halaman, hingga proses pengisian formulir. Selenium mengeksekusi setiap perintah sesuai instruksi dalam skrip, sehingga tindakan pengguna dapat ditiru secara otomatis. Selama proses berjalan, penguji mencatat apakah sistem memberikan keluaran yang sesuai dengan kondisi yang diharapkan.

Setelah pengujian fungsional selesai, proses dilanjutkan ke pengujian regresi. Pengujian regresi dilakukan untuk memastikan bahwa pembaruan atau perubahan pada aplikasi tidak menimbulkan gangguan pada fungsi yang sebelumnya sudah berjalan dengan baik. Skrip yang sebelumnya telah digunakan pada pengujian fungsional dijalankan kembali tanpa mengubah struktur skenarionya. Hasil yang muncul dibandingkan dengan hasil sebelumnya untuk melihat apakah ada perbedaan perilaku sistem.

Tahapan berikutnya adalah pengujian kompatibilitas antar-browser. Selenium dijalankan pada browser yang berbeda seperti Chrome, Firefox, dan Edge untuk memastikan bahwa tampilan dan fungsi aplikasi tetap konsisten. Pada bagian ini, penguji mencatat perbedaan yang mungkin muncul, baik dari sisi tampilan maupun proses interaksi. Pengujian kompatibilitas penting karena setiap pengguna mungkin mengakses aplikasi dari browser yang berbeda.

Dalam proses implementasi, penguji juga melakukan verifikasi element locator. Sebagian besar alur pengujian bergantung pada kemampuan Selenium dalam menemukan elemen pada halaman web. Jika struktur elemen berubah, maka skrip perlu disesuaikan kembali. Oleh karena itu, selama pengujian berlangsung, penguji memastikan bahwa setiap tindakan yang dilakukan Selenium benar-benar mengenai elemen yang dimaksud.

Selanjutnya, dilakukan pencatatan hasil secara sistematis. Setiap hasil yang muncul dari eksekusi skrip dicatat dalam bentuk tabel atau log untuk memudahkan proses evaluasi pada tahap berikutnya. Hasil pengujian dapat berupa status berhasil, gagal, atau ditemukan ketidaksesuaian tertentu. Pencatatan yang rapi membantu penguji dalam menarik kesimpulan mengenai kualitas perangkat lunak secara keseluruhan.

Pada beberapa kondisi, penguji juga melakukan pengujian ulang (re-testing). Pengujian ulang dilakukan ketika ada perbaikan atau perubahan setelah ditemukan masalah pada eksekusi pertama. Dengan cara ini, penguji dapat memastikan bahwa masalah tersebut benar-benar telah terselesaikan dan tidak menimbulkan dampak baru pada bagian lain.

Secara keseluruhan, implementasi pengujian perangkat lunak ini merupakan tahap inti dalam menilai kualitas aplikasi. Melalui proses yang terstruktur dan didukung oleh alat bantu seperti Selenium, pengujian dapat dilakukan secara lebih efisien, konsisten, dan objektif. Hasil dari tahap ini kemudian menjadi dasar dalam melakukan evaluasi dan penyusunan laporan akhir mengenai kualitas perangkat lunak.

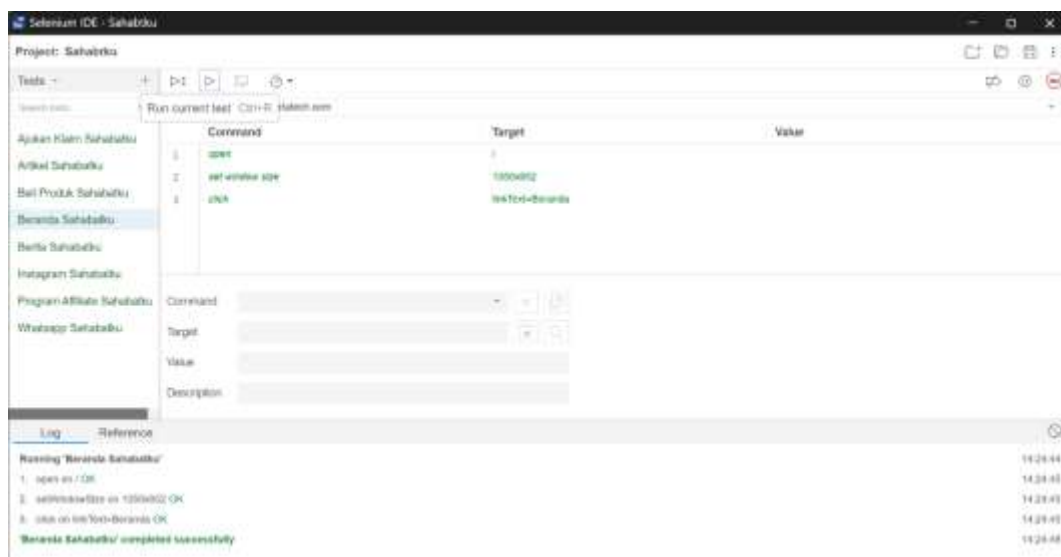
Table 4.1. Pengujian Navbar Aplikasi Sahabatku

No	Menu	ID Test Case	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status	Catatan
1	Beranda	TC-01	Akses halaman Beranda	Klik menu Beranda pada navigasi utama	Sistem menampilkan halaman Beranda tanpa error	Halaman Beranda tampil normal	PASS	—
2	Beli Produk	TC-02	Akses halaman Beli Produk	Klik menu Beli Produk	Sistem menampilkan halaman Beli Produk	Halaman Beli Produk tampil normal	PASS	—
3	Program	TC-03	Akses halaman	Klik menu	Sistem mengara	Halaman	PASS	Navigasi

	Affiliate		Program Affiliate	Program Affiliate	hkan ke halaman Program Affiliate	tidak berpindah namun tidak terjadi error		belum mengarah ke halaman tujuan
4	Ajukan Klaim	TC-04	Akses halaman Ajukan Klaim	Klik menu Ajukan Klaim	Sistem mengarahkan ke halaman Ajukan Klaim	Halaman tidak berpindah namun tidak terjadi error	PASS	Navigasi belum berfungsi maksimal
5	Artikel	TC-05	Akses halaman Artikel	Klik menu Artikel	Sistem menampilkan halaman Artikel	Halaman tidak berpindah namun tidak terjadi error	PASS	Konten belum tampil
6	Berita	TC-06	Akses halaman Berita	Klik menu Berita	Sistem menampilkan halaman Berita	Halaman tidak berpindah namun tidak terjadi error	PASS	Link belum aktif
7	Instagram	TC-07	Akses tautan Instagram	Klik ikon/ menu Instagram	Sistem membuka halaman Instagram	Halaman Instagram terbuka	PASS	—

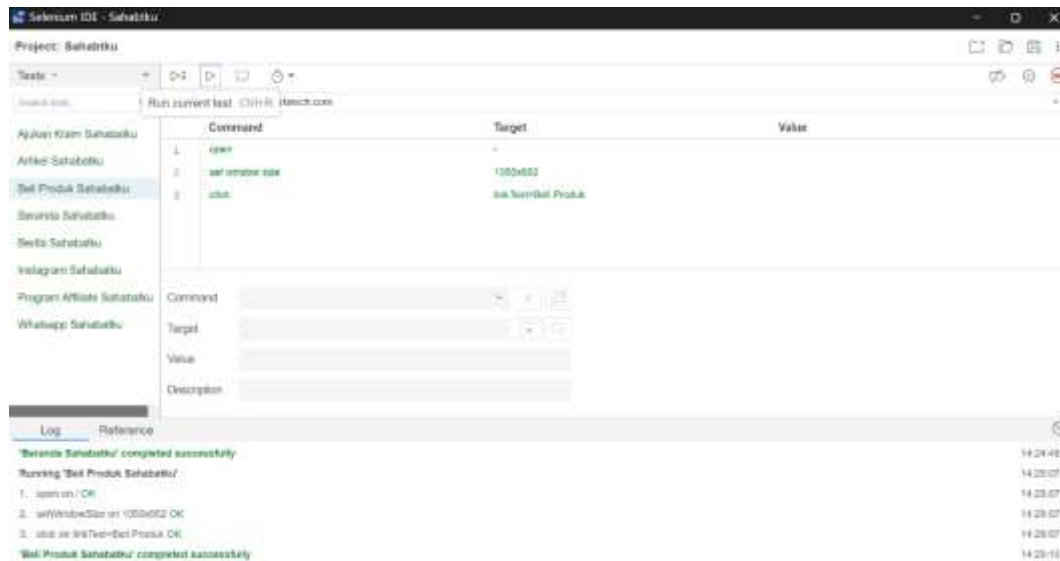
8	Whats App	TC-08	Akses tautan WhatsA pp	Klik ikon/ menu Whats App	Sistem membuka WhatsA pp	Halaman What sApp terbuka	PASS	—
---	-----------	-------	------------------------	---------------------------	--------------------------	---------------------------	------	---

Pengujian Navbar Aplikasi Sahabatku



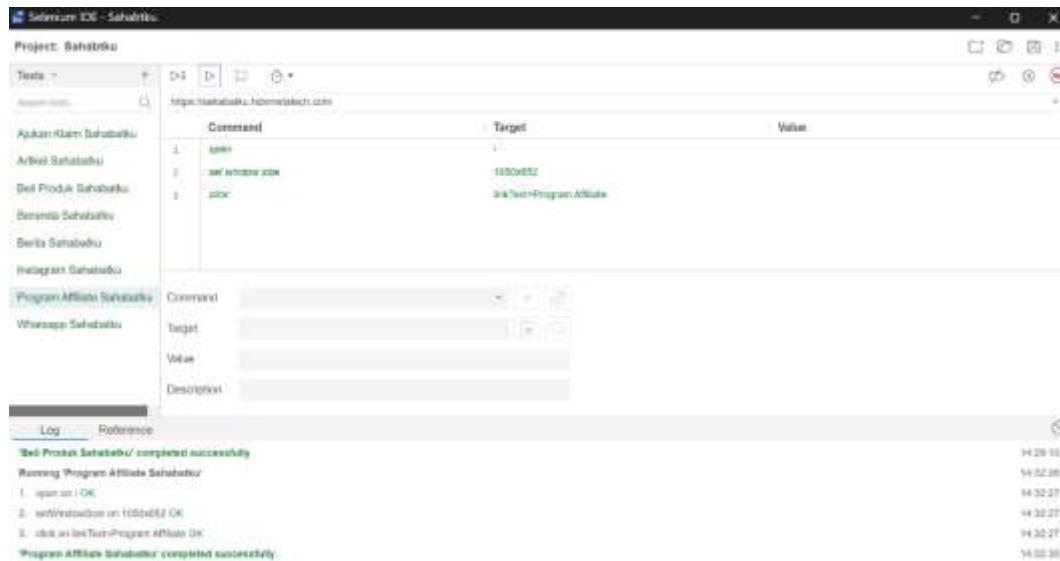
Gambar 4. 6. Hasil Pengujian Halaman Beranda

Gambar di atas menunjukkan hasil pengujian fungsional pada menu Beranda menggunakan Selenium IDE. Proses pengujian diawali dengan membuka halaman utama aplikasi dan menyesuaikan ukuran jendela browser agar tampilan sistem berada pada kondisi yang konsisten. Selanjutnya dilakukan aksi klik pada menu Beranda. Hasil pengujian menunjukkan bahwa halaman Beranda berhasil ditampilkan tanpa kendala dan tidak ditemukan error selama proses eksekusi. Dengan demikian, dapat disimpulkan bahwa fungsi menu Beranda telah berjalan sesuai dengan skenario pengujian yang ditetapkan.



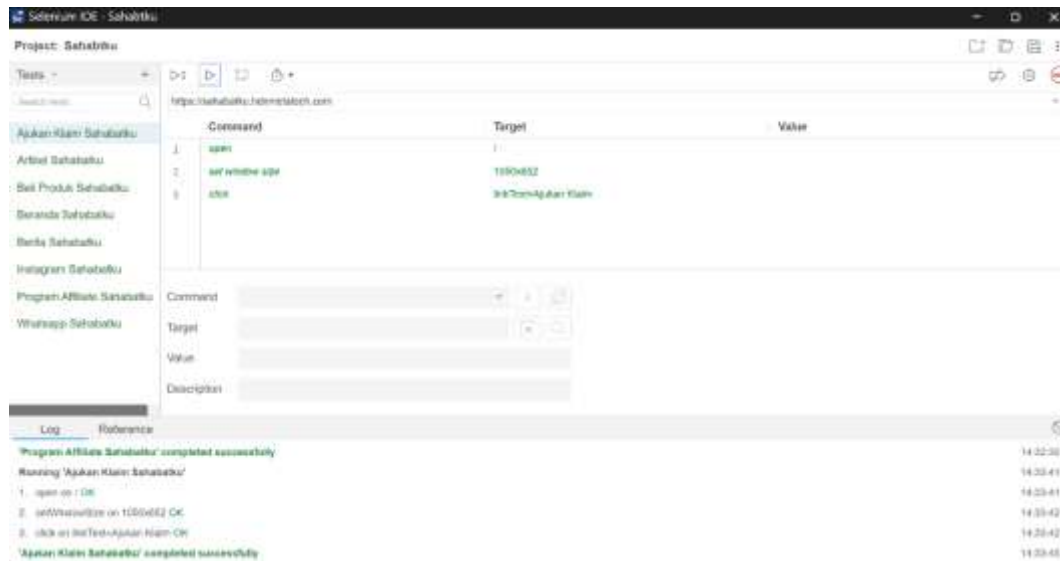
Gambar 4. 7. Hasil Pengujian Halaman Beli produk

Gambar di atas ditampilkan pengujian fungsional terhadap menu Beli Produk. Selenium IDE menjalankan skenario pengujian dengan membuka aplikasi, mengatur ukuran tampilan, dan melakukan klik pada menu Beli Produk. Berdasarkan hasil pengujian, sistem berhasil menampilkan halaman Beli Produk dengan baik tanpa adanya kesalahan. Hal ini menunjukkan bahwa fungsi navigasi menu Beli Produk dapat digunakan oleh pengguna sebagaimana yang direncanakan.



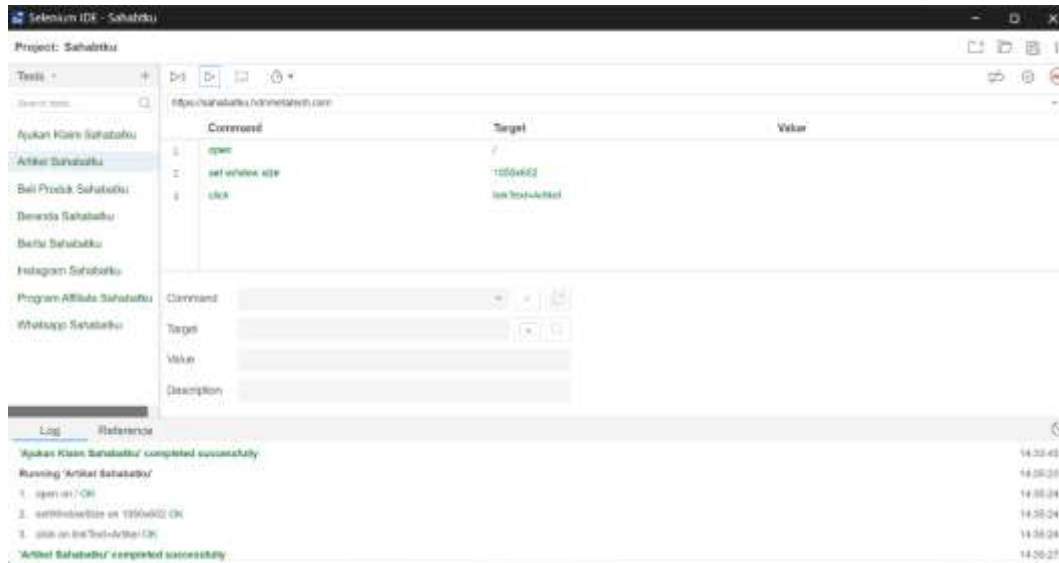
Gambar 4. 8. Hasil Pengujian Halaman Program Affiliate

Gambar di atas memperlihatkan proses pengujian fungsional pada menu Program Affiliate. Pengujian dilakukan dengan menjalankan perintah klik pada menu tersebut setelah halaman utama aplikasi berhasil dibuka. Hasil pengujian menunjukkan bahwa sistem merespons perintah dengan baik dan tidak menampilkan pesan error. Namun demikian, halaman Program Affiliate belum berpindah atau ditampilkan seperti menu lainnya. Kondisi ini mengindikasikan bahwa menu telah terdeteksi oleh sistem, tetapi fungsi navigasi atau pemanggilan halaman masih belum berjalan secara optimal.



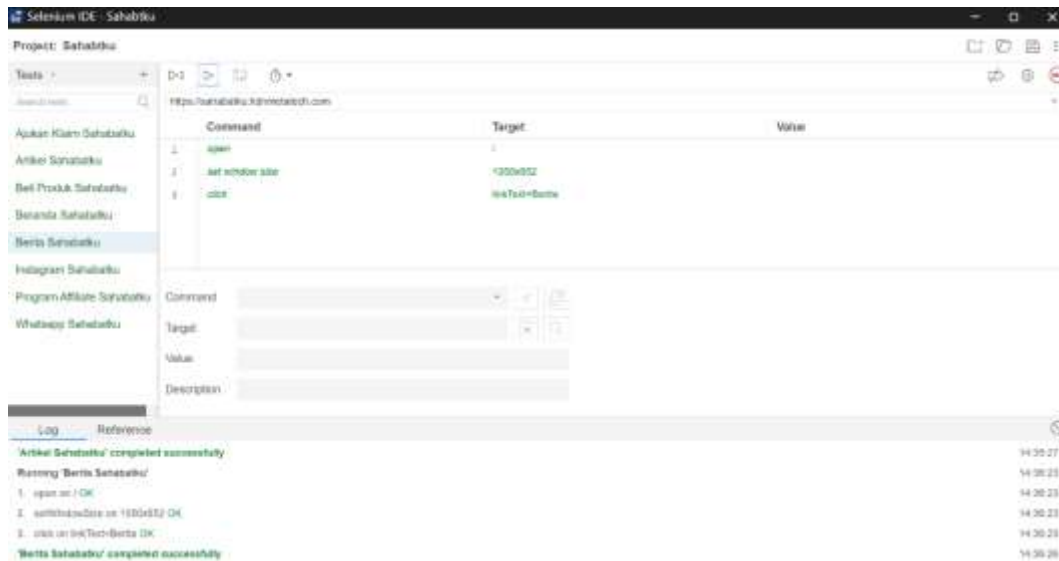
Gambar 4. 9. Hasil Pengujian Halaman Ajukan Klaim

Gambar di atas ditunjukkan hasil pengujian terhadap menu Ajukan Klaim. Proses pengujian dilakukan dengan mengakses halaman utama aplikasi dan mengeksekusi perintah klik pada menu Ajukan Klaim. Berdasarkan hasil pengujian, sistem memberikan respons yang normal tanpa terjadinya error. Akan tetapi, halaman Ajukan Klaim belum ditampilkan atau terjadi perpindahan halaman. Hal ini menunjukkan bahwa fungsi menu telah aktif, namun implementasi tampilan halaman masih memerlukan pengembangan lebih lanjut.



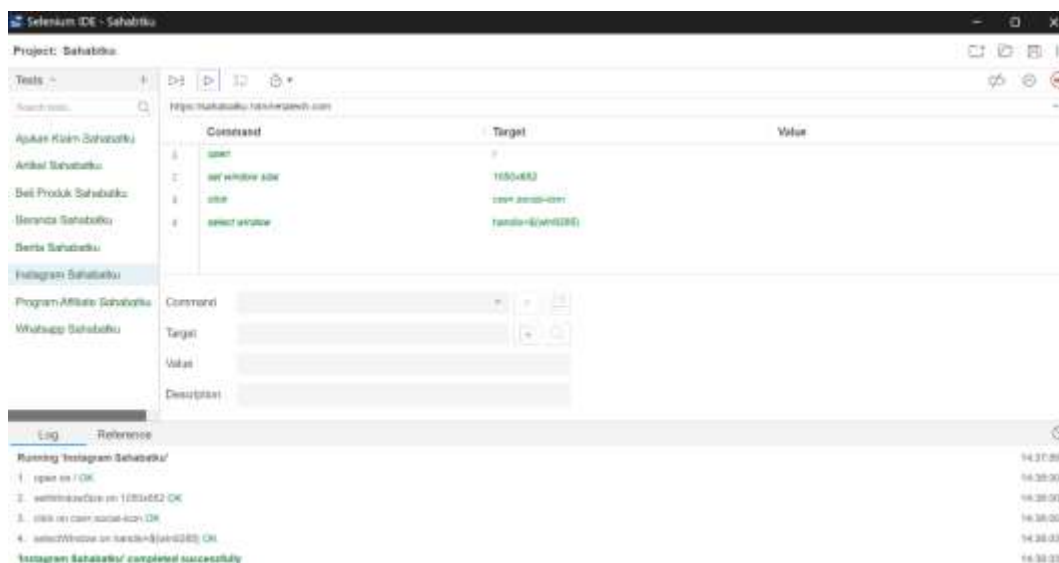
Gambar 4. 10. Hasil Pengujian Halaman Artikel

Gambar di atas menampilkan hasil pengujian fungsional pada menu Artikel. Selenium IDE berhasil menjalankan perintah klik pada menu Artikel setelah aplikasi dibuka. Hasil pengujian menunjukkan bahwa sistem tetap merespons perintah pengguna dan tidak mengalami kegagalan sistem. Namun, halaman Artikel belum muncul atau ditampilkan sebagaimana yang diharapkan. Kondisi ini menunjukkan bahwa menu Artikel telah terhubung pada sistem, tetapi fungsi penampilan konten masih belum berjalan secara maksimal.



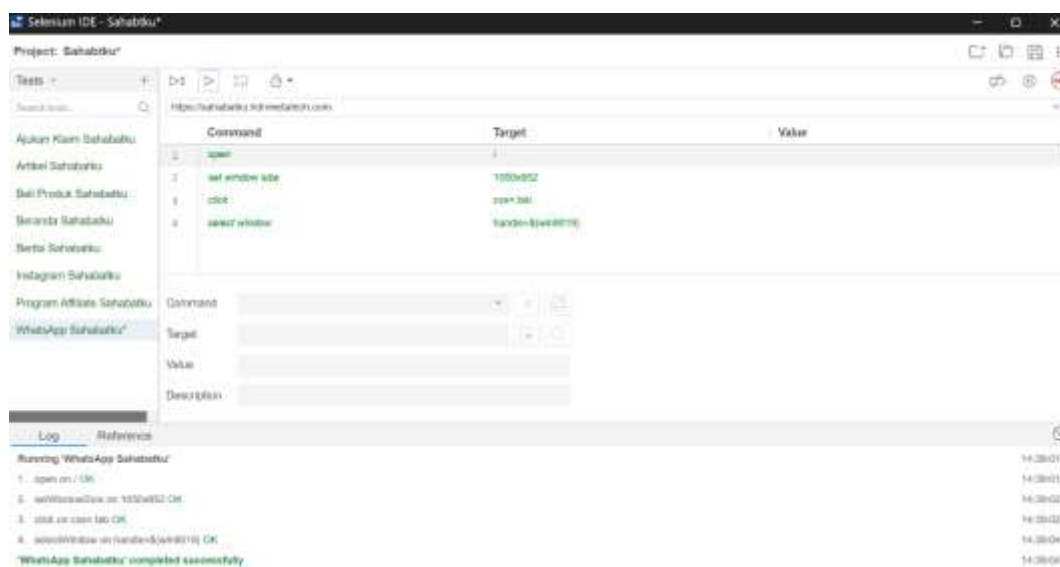
Gambar 4. 11. Hasil Pengujian Halaman Berita

Gambar di atas ditampilkan pengujian fungsional terhadap menu Berita. Pengujian dilakukan dengan mengeksekusi perintah klik pada menu Berita melalui Selenium IDE. Hasil pengujian menunjukkan bahwa sistem tidak mengalami error dan tetap memberikan respons. Akan tetapi, halaman Berita belum ditampilkan atau terjadi perpindahan halaman. Hal ini mengindikasikan bahwa fungsi menu telah tersedia, namun mekanisme pemanggilan halaman masih perlu dilakukan penyempurnaan.



Gambar 4. 12. Hasil Pengujian Halaman Instagram

Gambar di atas menunjukkan hasil pengujian pada ikon Instagram yang terdapat pada aplikasi Sahabatku. Pengujian dilakukan dengan mengklik ikon media sosial tersebut hingga sistem membuka halaman Instagram pada jendela baru. Selenium IDE berhasil melakukan pengalihan jendela (window handling) tanpa kendala. Berdasarkan hasil tersebut, dapat disimpulkan bahwa tautan menuju Instagram berfungsi dengan baik dan integrasi media sosial telah berjalan sesuai dengan skenario pengujian.



Gambar 4. 13. Hasil Pengujian Halaman WhatsApp

Gambar di atas memperlihatkan hasil pengujian fungsional pada ikon WhatsApp. Proses pengujian dilakukan dengan mengakses aplikasi, mengklik ikon WhatsApp, dan melakukan pengalihan jendela ke halaman WhatsApp. Hasil pengujian menunjukkan bahwa sistem berhasil membuka tautan tanpa error dan proses pengalihan jendela berjalan dengan baik. Dengan demikian, fitur komunikasi melalui WhatsApp pada aplikasi Sahabatku dapat dinyatakan berfungsi secara normal.

Pengujian perangkat lunak pada Aplikasi Sahabatku dilakukan dengan pendekatan pengujian fungsional menggunakan metode *automatic testing* berbasis

Selenium. Pengujian ini bertujuan untuk memastikan bahwa setiap fungsi utama pada aplikasi dapat berjalan sesuai dengan alur yang telah dirancang serta tidak menimbulkan kesalahan saat diakses oleh pengguna. Proses pengujian difokuskan pada fitur-fitur yang sering digunakan, seperti navigasi menu, akses halaman, serta tautan menuju platform eksternal.

Pelaksanaan pengujian diawali dengan penyusunan skenario uji berdasarkan kebutuhan sistem dan alur penggunaan aplikasi. Setiap skenario kemudian diterjemahkan ke dalam skrip pengujian otomatis menggunakan Selenium IDE, yang berfungsi untuk mensimulasikan interaksi pengguna pada website sahabatku.hdnmetatech.com. Pengujian dilakukan dengan cara menjalankan skrip pada browser untuk memverifikasi apakah sistem menampilkan halaman yang sesuai, merespons aksi pengguna dengan benar, serta tidak mengalami error selama proses berlangsung.

Berdasarkan hasil pengujian yang telah dilakukan, sebagian besar fungsi navigasi aplikasi menunjukkan hasil yang sesuai dengan harapan. Menu seperti Beranda, Beli Produk, Instagram, dan WhatsApp berhasil diakses dan menampilkan halaman tujuan tanpa adanya error. Hal ini menunjukkan bahwa fungsi dasar navigasi dan integrasi tautan eksternal telah berjalan dengan baik. Selenium mampu menjalankan seluruh skenario pengujian secara konsisten dan menghasilkan keluaran yang stabil.

Namun demikian, pengujian juga menemukan beberapa ketidaksesuaian pada sejumlah menu, seperti Program Affiliate, Ajukan Klaim, Artikel, dan Berita. Pada menu-menu tersebut, sistem tidak mengalami error atau kegagalan fungsi, tetapi halaman tidak berpindah ke tujuan yang semestinya. Kondisi ini mengindikasikan bahwa secara fungsional sistem masih berjalan, namun terdapat masalah pada mekanisme navigasi atau pengaitan halaman yang perlu ditindaklanjuti lebih lanjut oleh pengembang.

Secara keseluruhan, pengujian fungsional menggunakan Selenium memberikan gambaran yang cukup jelas mengenai kondisi aplikasi saat ini. Metode pengujian otomatis terbukti membantu dalam menjalankan pengujian secara

berulang dengan hasil yang konsisten dan terdokumentasi dengan baik. Hasil pengujian ini kemudian digunakan sebagai dasar untuk menyusun temuan serta rekomendasi perbaikan, sehingga dapat mendukung upaya peningkatan kualitas Aplikasi Sahabatku ke depannya.

4.3 Temuan Dan Rekomendasi

Setelah seluruh proses pengujian dijalankan, baik melalui pengujian fungsional, regresi, maupun kompatibilitas, terdapat beberapa temuan yang menjadi dasar dalam memberikan rekomendasi perbaikan. Temuan ini muncul dari hasil pencatatan setiap eksekusi skrip, observasi selama proses pengujian berlangsung, serta analisis perilaku sistem terhadap berbagai data uji yang digunakan.

Temuan pertama berkaitan dengan konsistensi fungsi dasar aplikasi. Dari hasil pengujian fungsional, sebagian besar fitur dapat berjalan sesuai dengan yang diharapkan. Proses login, navigasi halaman, dan interaksi dengan elemen utama aplikasi dapat dieksekusi Selenium tanpa error yang berarti. Namun, terdapat beberapa elemen dinamis yang terkadang membutuhkan waktu lebih lama untuk muncul sehingga menyebabkan skrip tertunda sesaat sebelum melanjutkan ke langkah berikutnya. Hal ini menunjukkan bahwa stabilitas elemen pada halaman masih perlu diperbaiki agar pengujian dapat berjalan lebih lancar.

Temuan berikutnya berasal dari pengujian regresi. Pada bagian ini ditemukan bahwa beberapa pembaruan tampilan pada halaman tertentu menyebabkan skrip pengujian tidak dapat mengenali elemen sebelumnya. Masalah ini bukan disebabkan oleh kesalahan pada fitur, tetapi lebih pada perubahan struktur HTML yang mempengaruhi locator elemen pada Selenium. Hal ini menunjukkan bahwa setiap perubahan tampilan perlu dikoordinasikan dengan tim pengujian agar penyesuaian skrip dapat dilakukan lebih awal.

Dari sisi kompatibilitas antar-browser, hasil pengujian menunjukkan bahwa aplikasi dapat berjalan baik pada Chrome dan Firefox, namun terdapat sedikit perbedaan tampilan pada Edge, terutama pada ukuran margin dan posisi beberapa elemen. Meskipun tidak mengganggu fungsi utama, perbedaan ini dapat

mempengaruhi pengalaman pengguna dan sebaiknya diperhatikan pada pengembangan berikutnya.

Selain itu, pengujian juga menemukan bahwa penanganan input tidak valid belum sepenuhnya konsisten. Pada beberapa form, sistem memberikan pesan kesalahan yang jelas ketika data tidak sesuai, namun pada beberapa form lain, aplikasi hanya melakukan refresh halaman tanpa menampilkan keterangan apapun. Temuan ini perlu diperhatikan karena dapat membingungkan pengguna dalam penggunaan sehari-hari.

Berdasarkan temuan-temuan tersebut, beberapa rekomendasi dapat diberikan untuk meningkatkan kualitas perangkat lunak, yaitu:

1. Meningkatkan stabilitas elemen halaman, terutama elemen dinamis agar waktu loading lebih konsisten dan skrip otomatis tidak tertunda tanpa alasan.
2. Melakukan koordinasi antara pengembang dan penguji setiap kali terjadi perubahan tampilan atau struktur halaman, sehingga skrip Selenium dapat diperbarui sebelum pengujian dijalankan.
3. Meninjau kembali desain antarmuka agar kompatibilitas antar-browser semakin baik dan tidak menimbulkan perbedaan tampilan yang dapat mengganggu pengalaman pengguna.
4. Memperbaiki validasi input, dengan memastikan bahwa setiap formulir memberikan pesan kesalahan yang jelas ketika pengguna memasukkan data yang tidak sesuai.
5. Menambah dokumentasi internal, terutama terkait elemen-elemen yang sering berubah, sehingga proses penyesuaian skrip pengujian dapat dilakukan lebih cepat.

Secara keseluruhan, temuan dan rekomendasi ini menjadi dasar penting untuk meningkatkan kualitas perangkat lunak secara menyeluruh. Dengan melakukan tindak lanjut dari rekomendasi tersebut, aplikasi diharapkan dapat bekerja lebih stabil, mudah dipelihara, dan memberikan pengalaman yang lebih baik bagi pengguna.

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan rangkaian proses pengujian yang telah dilakukan, mulai dari analisis kebutuhan, penyusunan skenario, pelaksanaan pengujian menggunakan Selenium, hingga evaluasi temuan di lapangan, dapat disimpulkan bahwa pengujian perangkat lunak ini berjalan secara terstruktur dan mampu memberikan gambaran nyata mengenai kualitas aplikasi yang diuji. Pendekatan yang digunakan mengutamakan pengujian fungsional, regresi, serta kompatibilitas, sehingga setiap fitur utama dapat diperiksa dengan lebih menyeluruh.

Hasil pengujian menunjukkan bahwa sebagian besar fungsi dasar aplikasi telah berjalan sesuai dengan harapan. Proses login, navigasi, serta interaksi antar-elemen dapat dijalankan dengan stabil oleh Selenium. Hal ini sejalan dengan konsep kualitas perangkat lunak yang menekankan kecocokan fungsi, konsistensi perilaku, serta keandalan sistem dalam menjalankan perintah pengguna.

Meskipun begitu, pengujian juga menemukan beberapa hal yang perlu diperhatikan, seperti elemen halaman yang masih belum stabil, perubahan struktur tampilan yang memengaruhi skrip otomatis, serta perbedaan kecil pada tampilan antar-browser. Temuan tersebut bukan merupakan kegagalan sistem secara keseluruhan, namun lebih kepada aspek teknis yang perlu disempurnakan agar aplikasi dapat bekerja dengan lebih konsisten di berbagai kondisi penggunaan.

Secara keseluruhan, proses pengujian ini membuktikan bahwa penggunaan Selenium sebagai perangkat bantu dapat membantu mempercepat dan mengefisienkan kegiatan pengujian, terutama dalam bagian yang bersifat repetitif dan membutuhkan ketelitian tinggi. Dengan dukungan skenario pengujian yang terencana dan data uji yang terstruktur, hasil yang diperoleh dapat dipertanggungjawabkan dan menjadi dasar yang kuat untuk melakukan perbaikan lebih lanjut.

Dengan demikian, pengujian yang dilakukan tidak hanya memberikan gambaran mengenai kondisi aplikasi saat ini, tetapi juga memberikan arah bagi pengembangan berikutnya agar kualitas perangkat lunak semakin meningkat dan mampu memenuhi kebutuhan pengguna secara optimal.

5.2 Saran

Berdasarkan hasil pengujian dan temuan yang diperoleh selama proses kerja praktek, terdapat beberapa saran yang dapat dipertimbangkan untuk meningkatkan kualitas perangkat lunak serta memperbaiki proses pengujiannya di masa mendatang.

Pertama, stabilitas elemen pada halaman aplikasi perlu ditingkatkan, terutama untuk elemen dinamis yang sering berubah posisi atau membutuhkan waktu lebih lama untuk dimuat. Penyempurnaan ini akan sangat membantu agar proses pengujian otomatis menggunakan Selenium dapat berjalan tanpa hambatan dan hasilnya lebih konsisten.

Kedua, setiap pembaruan tampilan atau penyesuaian struktur halaman sebaiknya dikomunikasikan kepada tim pengujian sejak awal. Hal ini penting agar skrip pengujian dapat disesuaikan lebih cepat dan tidak menimbulkan kegagalan saat pengujian regresi dilakukan. Koordinasi yang lebih baik antara tim pengembang dan penguji akan mempercepat proses perbaikan sekaligus menjaga kualitas aplikasi.

Ketiga, perlu dilakukan peninjauan ulang terhadap desain antarmuka untuk meningkatkan konsistensi tampilan antar-browser. Meskipun perbedaannya tidak terlalu signifikan, tampilan yang seragam dapat memberikan pengalaman yang lebih baik bagi pengguna dan mempermudah proses pengujian di masa mendatang.

Keempat, validasi input pada setiap bagian formulir perlu diperjelas, terutama untuk kasus data tidak valid. Dengan adanya pesan kesalahan yang lebih informatif, pengguna dapat memahami kesalahan yang terjadi, sementara penguji dapat lebih mudah menilai apakah sistem merespons input secara tepat.

Kelima, untuk mendukung keberlanjutan proses pengujian, disarankan agar dokumentasi internal mengenai perubahan elemen dan alur aplikasi diperbarui secara berkala. Dokumentasi yang lengkap akan memudahkan penguji berikutnya dalam memahami struktur aplikasi dan menyesuaikan skrip pengujian tanpa harus mengulang proses dari awal.

Akhirnya, penggunaan Selenium dalam pengujian sebaiknya terus dikembangkan, baik melalui penyusunan skrip yang lebih modular maupun

penggunaan framework pendukung. Dengan cara ini, proses pengujian akan menjadi lebih fleksibel, efisien, dan mudah diperluas jika aplikasi mengalami pengembangan lebih lanjut.

Secara keseluruhan, saran-saran ini diharapkan dapat membantu meningkatkan kualitas sistem yang diuji serta memperbaiki proses pengujian berikutnya agar lebih terarah dan efektif.

DAFTAR PUSTAKA

- Santos, L., 2022. Kualitas Perangkat Lunak: Prinsip Pengujian Fungsional, Regresi, dan Kompatibilitas. Lisbon: Digital Engineering Press.
- Silva, J., 2023. Strategi Pengujian Aplikasi Web Berbasis Selenium: Skenario, Validasi, dan Analisis Hasil. Rio de Janeiro: Software Testing Institute.
- Patel, M., 2024. Optimalisasi Pengujian Perangkat Lunak dengan Pendekatan Skenario Terstruktur dan Otomasi. Mumbai: TechInsight Publishing.

Lampiran 1. Spesifikasi Perangkat Lunak

Selenium IDE merupakan perangkat lunak yang digunakan dalam kegiatan kerja praktek ini sebagai alat bantu utama dalam melakukan pengujian fungsional secara otomatis pada Aplikasi Sahabatku. Penggunaan Selenium IDE dipilih karena kemampuannya dalam membantu proses pengujian aplikasi berbasis web secara lebih sistematis, terstruktur, dan konsisten. Dalam pengujian perangkat lunak, konsistensi hasil menjadi salah satu aspek penting, terutama ketika pengujian harus dilakukan secara berulang pada fitur yang sama. Selenium IDE mampu memenuhi kebutuhan tersebut dengan cara merekam aktivitas pengguna dan menjalankannya kembali tanpa perubahan alur pengujian.

Pada praktik pengujian manual, penguji harus mengulangi langkah yang sama setiap kali melakukan pengujian, seperti membuka halaman, mengklik menu, atau mengakses fitur tertentu. Proses ini tidak hanya memakan waktu, tetapi juga berpotensi menimbulkan kesalahan akibat faktor manusia, misalnya lupa melakukan satu langkah atau melakukan urutan yang tidak sesuai. Dengan adanya Selenium IDE, proses tersebut dapat diminimalkan karena setiap langkah interaksi direkam dan dijalankan kembali secara otomatis sesuai urutan yang telah ditentukan sebelumnya.

Dalam pelaksanaan kerja praktek ini, Selenium IDE dijalankan sebagai ekstensi pada browser Google Chrome. Pemilihan Google Chrome sebagai media pengujian didasarkan pada pertimbangan bahwa browser ini merupakan salah satu browser yang paling banyak digunakan oleh pengguna, sehingga hasil pengujian diharapkan dapat merepresentasikan kondisi penggunaan aplikasi secara umum. Selenium IDE yang terintegrasi langsung dengan browser memungkinkan pengujian dilakukan tanpa perlu konfigurasi yang kompleks, sehingga lebih mudah diterapkan dalam lingkungan kerja praktek.

Fungsi utama Selenium IDE dalam pengujian ini adalah merekam setiap langkah interaksi pengguna dengan sistem. Langkah-langkah tersebut meliputi aktivitas seperti membuka halaman utama, mengklik menu navigasi, mengakses halaman tertentu, hingga membuka tautan eksternal seperti media sosial. Setiap aktivitas yang dilakukan pada browser akan direkam oleh Selenium IDE dan

disimpan dalam bentuk skrip pengujian. Skrip inilah yang kemudian dapat dijalankan kembali untuk melakukan pengujian ulang dengan alur yang sama persis seperti saat perekaman awal.

Keberadaan skrip pengujian ini memberikan keuntungan tersendiri dalam proses pengujian perangkat lunak. Penguji tidak perlu lagi mengulangi langkah yang sama secara manual, sehingga waktu pengujian dapat dipersingkat secara signifikan. Selain itu, skrip pengujian juga dapat digunakan kembali ketika aplikasi mengalami pembaruan atau perubahan tertentu. Dengan menjalankan skrip yang sama, penguji dapat mengetahui apakah perubahan tersebut memengaruhi fungsi yang sebelumnya telah berjalan dengan baik.

Spesifikasi perangkat lunak Selenium IDE yang digunakan dalam kerja praktek ini mencakup versi aplikasi yang telah mendukung pengujian otomatis pada elemen-elemen web modern. Selenium IDE mampu mengenali berbagai komponen halaman web yang bersifat dinamis maupun statis. Dalam proses identifikasi elemen, Selenium IDE menggunakan beberapa metode, seperti atribut *id*, *name*, serta *selector* yang berbasis struktur halaman. Metode identifikasi ini sangat membantu dalam memastikan bahwa perintah pengujian dijalankan pada elemen yang benar sesuai dengan skenario yang telah disusun.

Kemampuan Selenium IDE dalam mengenali elemen halaman juga sangat penting ketika aplikasi memiliki banyak menu dan komponen antarmuka. Dengan identifikasi elemen yang tepat, pengujian dapat dilakukan secara lebih akurat dan terarah. Apabila terjadi perubahan kecil pada tampilan halaman, penguji dapat menyesuaikan elemen yang digunakan dalam skrip tanpa harus membuat ulang seluruh skenario pengujian. Hal ini menjadikan Selenium IDE cukup fleksibel untuk digunakan pada aplikasi yang masih dalam tahap pengembangan.

Selain merekam dan menjalankan skrip pengujian, Selenium IDE juga menyediakan fitur untuk menampilkan hasil eksekusi secara langsung. Setiap langkah pengujian akan ditampilkan dengan status tertentu, seperti berhasil atau gagal. Informasi ini memudahkan penguji dalam mengetahui bagian mana dari pengujian yang berjalan sesuai dengan harapan dan bagian mana yang perlu mendapatkan perhatian lebih lanjut. Dengan adanya informasi hasil ini, proses evaluasi menjadi lebih terstruktur dan mudah dianalisis.

Hasil eksekusi pengujian yang ditampilkan oleh Selenium IDE kemudian digunakan sebagai dasar dalam penyusunan tabel hasil pengujian. Tabel tersebut memuat informasi mengenai skenario pengujian, hasil yang diharapkan, hasil aktual, serta status pengujian. Data ini sangat penting dalam penyusunan laporan kerja praktek karena menjadi bukti bahwa pengujian telah dilakukan secara nyata dan terdokumentasi dengan baik. Selain itu, hasil pengujian juga menjadi dasar dalam penentuan temuan dan rekomendasi perbaikan pada bab selanjutnya.

Dari sisi kemudahan penggunaan, Selenium IDE tergolong sebagai perangkat lunak yang relatif mudah dipahami, terutama bagi pengguna yang baru mengenal konsep pengujian otomatis. Antarmuka yang sederhana dan intuitif memungkinkan penguji untuk memahami alur kerja perangkat lunak tanpa memerlukan waktu pembelajaran yang lama. Penguji dapat langsung merekam, menjalankan, serta mengevaluasi hasil pengujian tanpa harus menulis kode pemrograman yang kompleks.

Kemudahan penggunaan ini menjadi salah satu alasan utama pemilihan Selenium IDE dalam kegiatan kerja praktek. Sebagai mahasiswa, penggunaan alat yang tidak terlalu rumit memungkinkan fokus pengujian diarahkan pada pemahaman konsep pengujian perangkat lunak, bukan pada kesulitan teknis penggunaan alat. Dengan demikian, tujuan kerja praktek untuk memahami proses pengujian secara langsung dapat tercapai dengan lebih optimal.

Selain itu, Selenium IDE juga mendukung pengujian yang bersifat berulang dan konsisten. Pengujian berulang sangat dibutuhkan ketika aplikasi mengalami perubahan atau penyesuaian fitur. Dengan menjalankan kembali skrip pengujian yang sama, penguji dapat memastikan bahwa fungsi yang telah diuji sebelumnya tetap berjalan dengan baik dan tidak mengalami penurunan kualitas. Hal ini sejalan dengan konsep pengujian regresi yang bertujuan menjaga stabilitas sistem.

Dalam konteks pengujian Aplikasi Sahabatku, Selenium IDE berperan sebagai alat bantu yang mendukung proses pengujian fungsional secara menyeluruh. Pengujian dilakukan untuk memastikan bahwa setiap menu dan fitur utama dapat diakses tanpa error serta memberikan respons yang sesuai dengan tindakan pengguna. Dengan dukungan Selenium IDE, seluruh proses pengujian dapat dilakukan secara lebih terarah dan sistematis.

Secara keseluruhan, Selenium IDE memberikan kontribusi yang signifikan dalam pelaksanaan kerja praktek ini. Perangkat lunak ini tidak hanya membantu mempercepat proses pengujian, tetapi juga memastikan bahwa hasil pengujian dapat dicatat dan dianalisis dengan baik. Dokumentasi hasil pengujian yang dihasilkan dari Selenium IDE menjadi bagian penting dalam laporan kerja praktek sebagai bukti pelaksanaan pengujian perangkat lunak secara nyata.

Dengan spesifikasi dan kemampuan yang dimiliki, Selenium IDE dapat digunakan sebagai alat bantu yang efektif dalam menjaga kualitas aplikasi berbasis web. Penggunaan Selenium IDE dalam kerja praktek ini diharapkan dapat memberikan gambaran yang jelas mengenai bagaimana pengujian otomatis dapat diterapkan dalam lingkungan kerja sebenarnya serta menjadi bekal pengetahuan bagi mahasiswa dalam menghadapi dunia kerja di bidang teknologi informasi.

Lampiran 2. Detail Pengujian Perangkat Lunak

Lampiran ini memuat penjelasan secara rinci mengenai keseluruhan proses pengujian perangkat lunak yang dilakukan pada Aplikasi Sahabatku dengan memanfaatkan Selenium IDE sebagai alat bantu pengujian otomatis. Penyusunan lampiran ini bertujuan untuk memberikan gambaran yang jelas dan menyeluruh mengenai bagaimana pengujian dilakukan, mulai dari tahap perencanaan, pelaksanaan, hingga pendokumentasian hasil. Dengan adanya penjelasan detail pada lampiran ini, pembaca diharapkan dapat memahami alur pengujian secara utuh serta menilai bahwa proses pengujian telah dilakukan secara sistematis dan dapat dipertanggungjawabkan.

Pengujian perangkat lunak yang dilakukan pada Aplikasi Sahabatku difokuskan pada pengujian fungsional. Pengujian fungsional merupakan jenis pengujian yang bertujuan untuk memastikan bahwa setiap fungsi dalam sistem berjalan sesuai dengan kebutuhan yang telah ditetapkan sebelumnya. Fokus utama dari pengujian ini adalah menilai apakah sistem mampu merespons tindakan pengguna dengan benar, menampilkan halaman yang sesuai, serta menjalankan fitur tanpa menimbulkan error atau gangguan selama penggunaan. Pendekatan ini dipilih karena aplikasi yang diuji merupakan aplikasi berbasis web yang digunakan langsung oleh pengguna dalam aktivitas sehari-hari.

Tahap awal dalam proses pengujian adalah penentuan ruang lingkup dan fitur yang akan diuji. Penentuan fitur dilakukan dengan mempertimbangkan tingkat kepentingan fitur tersebut dalam penggunaan aplikasi. Fitur yang sering digunakan dan memiliki peran penting dalam alur penggunaan aplikasi menjadi prioritas utama dalam pengujian. Beberapa fitur yang termasuk dalam ruang lingkup pengujian antara lain menu Beranda, menu Beli Produk, menu Program Affiliate, menu Ajukan Klaim, menu Artikel, menu Berita, serta tautan eksternal yang mengarah ke media sosial seperti Instagram dan WhatsApp.

Setelah fitur ditentukan, langkah selanjutnya adalah menyusun skenario pengujian. Skenario pengujian merupakan gambaran langkah-langkah yang akan dilakukan oleh pengguna ketika berinteraksi dengan aplikasi. Penyusunan skenario

dilakukan secara sistematis agar setiap langkah mudah dipahami dan dapat dijalankan secara berurutan. Skenario ini mencakup tindakan seperti membuka halaman utama aplikasi, mengklik menu tertentu, berpindah dari satu halaman ke halaman lain, serta membuka tautan eksternal. Setiap skenario disusun dengan tujuan untuk merepresentasikan kondisi penggunaan aplikasi yang sebenarnya.

Skenario pengujian yang telah disusun kemudian diterjemahkan ke dalam proses perekaman menggunakan Selenium IDE. Pada tahap ini, Selenium IDE digunakan untuk merekam setiap aktivitas yang dilakukan pada browser. Proses perekaman dilakukan dengan menjalankan aplikasi melalui browser Google Chrome, kemudian penguji melakukan interaksi sesuai dengan skenario yang telah ditentukan. Setiap klik, perpindahan halaman, dan akses menu direkam secara otomatis oleh Selenium IDE dan disimpan sebagai rangkaian perintah pengujian.

Hasil dari proses perekaman tersebut berupa skrip pengujian yang berisi urutan langkah pengujian secara lengkap. Skrip ini menjadi komponen penting dalam pengujian otomatis karena dapat dijalankan kembali kapan saja tanpa harus melakukan input secara manual. Dengan adanya skrip pengujian, pengujian dapat dilakukan secara konsisten dengan alur yang sama, sehingga hasil yang diperoleh lebih stabil dan dapat dibandingkan antara satu pengujian dengan pengujian lainnya.

Setelah skrip pengujian terbentuk, tahap berikutnya adalah menjalankan skrip tersebut untuk melihat respons sistem. Pada tahap ini, Selenium IDE mengeksekusi setiap perintah pengujian sesuai dengan urutan yang telah direkam. Selama proses eksekusi berlangsung, penguji melakukan pengamatan terhadap perilaku sistem, terutama apakah halaman yang dituju berhasil ditampilkan, apakah navigasi berjalan dengan baik, serta apakah terdapat pesan error atau gangguan lain yang muncul.

Selenium IDE menampilkan hasil eksekusi pengujian dalam bentuk status pada setiap langkah pengujian. Status ini menunjukkan apakah suatu langkah berhasil dijalankan atau mengalami kegagalan. Informasi tersebut sangat membantu dalam proses evaluasi karena penguji dapat dengan mudah mengetahui

bagian mana dari pengujian yang berjalan sesuai dengan harapan dan bagian mana yang perlu diperhatikan lebih lanjut. Dengan demikian, proses analisis hasil pengujian dapat dilakukan secara lebih terarah.

Berdasarkan hasil pelaksanaan pengujian, dapat diketahui bahwa sebagian besar fitur utama pada Aplikasi Sahabatku dapat dijalankan tanpa kendala yang berarti. Menu Beranda dan menu Beli Produk berhasil menampilkan halaman yang sesuai ketika diakses. Begitu pula dengan tautan eksternal seperti Instagram dan WhatsApp yang dapat dibuka dengan baik melalui aplikasi. Hal ini menunjukkan bahwa fungsi dasar navigasi dan integrasi tautan eksternal telah berjalan dengan cukup baik.

Namun demikian, pengujian juga menemukan beberapa kondisi yang perlu dicatat sebagai bahan evaluasi. Pada beberapa menu, seperti Program Affiliate, Ajukan Klaim, Artikel, dan Berita, sistem tidak menampilkan error atau kegagalan fungsi, tetapi halaman tidak sepenuhnya mengarah ke tujuan yang diharapkan. Meskipun secara teknis sistem tetap berjalan dan tidak mengalami crash, kondisi ini menunjukkan bahwa mekanisme navigasi pada menu tersebut masih perlu diperbaiki agar sesuai dengan alur penggunaan yang direncanakan.

Setiap temuan yang diperoleh selama proses pengujian dicatat secara rinci. Pencatatan dilakukan untuk memastikan bahwa tidak ada informasi penting yang terlewat dan seluruh hasil pengujian terdokumentasi dengan baik. Hasil pengujian kemudian dirangkum dalam bentuk tabel hasil pengujian yang memuat informasi mengenai menu atau fitur yang diuji, skenario pengujian, hasil yang diharapkan, hasil aktual, serta status pengujian. Tabel ini berfungsi sebagai ringkasan hasil pengujian yang mudah dibaca dan dianalisis.

Dokumentasi hasil pengujian memiliki peran yang sangat penting dalam keseluruhan proses kerja praktek. Dokumentasi ini tidak hanya menjadi bukti bahwa pengujian telah dilakukan, tetapi juga menjadi dasar dalam menyusun analisis temuan dan rekomendasi perbaikan. Dengan adanya dokumentasi yang lengkap, pihak pengembang dapat memahami kondisi aplikasi secara lebih jelas dan menentukan langkah perbaikan yang diperlukan.

Selain itu, detail pengujian yang disajikan dalam lampiran ini juga menunjukkan bahwa pengujian dilakukan dengan pendekatan yang terencana dan sistematis. Setiap tahapan pengujian, mulai dari penentuan fitur, penyusunan skenario, perekaman skrip, eksekusi pengujian, hingga pendokumentasian hasil, dilakukan secara berurutan dan saling berkaitan. Pendekatan seperti ini sangat penting dalam menjaga kualitas pengujian perangkat lunak.

Dalam konteks kegiatan kerja praktek, proses pengujian ini juga memberikan pengalaman langsung bagi mahasiswa dalam menerapkan konsep pengujian perangkat lunak di lingkungan kerja nyata. Mahasiswa tidak hanya mempelajari teori pengujian, tetapi juga memahami bagaimana pengujian dilakukan secara praktis dengan memanfaatkan alat bantu seperti Selenium IDE. Pengalaman ini menjadi bekal yang berharga dalam menghadapi dunia kerja di bidang teknologi informasi.

Secara keseluruhan, detail pengujian perangkat lunak yang dijelaskan dalam lampiran ini menunjukkan bahwa pengujian pada Aplikasi Sahabatku telah dilakukan dengan baik dan terstruktur. Penggunaan Selenium IDE sebagai alat bantu pengujian otomatis membantu meningkatkan efisiensi, konsistensi, dan kejelasan hasil pengujian. Dengan adanya lampiran ini, pembaca diharapkan dapat memperoleh pemahaman yang mendalam mengenai proses pengujian yang telah dilakukan serta menilai bahwa hasil pengujian dapat dipertanggungjawabkan secara akademik.

Lampiran ini sekaligus menjadi bukti pendukung bahwa kegiatan kerja praktek tidak hanya berfokus pada penyusunan laporan, tetapi juga pada pelaksanaan pengujian perangkat lunak secara nyata. Dengan demikian, detail pengujian yang disajikan dapat memberikan kontribusi positif dalam upaya peningkatan kualitas Aplikasi Sahabatku serta menjadi referensi bagi pengujian serupa di masa mendatang.

Lampiran 3. Foto Kegiatan





Lampiran 5. Formulir Aktivitas

**AKTIVITAS HARIAN KERJA PRAKTEK
MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS 17 AGUSTUS 1945 SURABAYA**

Nama Mahasiswa : Yoga Ananta
NIM : 1462200019
Judul Kerja Praktek : Pengujian Fungsional Dengan Automatic Testing Untuk Aplikasi Sahabatku

No	Tanggal	Keterangan	TID
1.	15 September	Pengenalan perusahaan, lingkungan kerja, dan penjelasan umum proyek Kerja Praktek	A
2.	16 September	Diskusi awal mengenai aplikasi Sahabatku dan ruang lingkup pengujian aplikasi	A
3.	17 September	Studi literatur terkait automatic testing dan metode pengujian perangkat lunak	A
4.	18 September	Instalasi tools pengujian otomatis dan penyesuaian environment testing	A
5.	19 September	Analisis kebutuhan pengujian dan penentuan skenario test pada aplikasi Sahabatku	A
6.	23 September	Penyusunan test case untuk pengujian fitur utama aplikasi Sahabatku	A
7.	25 September	Implementasi automatic testing pada modul login dan registrasi	A
8.	30 September	Pengujian otomatis dan pencatatan hasil testing	A
9.	10 Oktober	Analisis bug dan error berdasarkan hasil automatic testing	A

No	Tanggal	Keterangan	TTD
10.	20 Oktober	Perbaikan dan pengujian ulang (re-testing) aplikasi	
11.	14 November	Evaluasi hasil pengujian aplikasi Sahabatku	
12.	1 Desember	Penyusunan laporan Kerja Praktek	
13.	15 Desember	Revisi laporan sesuai arahan pembimbing	
14.	24 Desember	Finalisasi laporan dan penutupan kegiatan Kerja Praktek	

Surabaya, 24 Desember 2025

.Pembimbing Lapangan



(Aditya Nanda Utama, S.Kom)

Lampiran 6. Formulir Kuesioner

KUESIONER UNTUK INSTITUSI PENGGUNA MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA UNIVERSITAS 17 AGUSTUS 1945 SURABAYA

Program studi Teknik Informatika Universitas 17 Agustus 1945 Surabaya mengadakan Survei mengenai Profile Mahasiswa Kerja Praktek. Tujuan dari Survei ini untuk mengevaluasi pengembangan kurikulum di Program studi Teknik Informatika Universitas 17 Agustus 1945 Surabaya yang merupakan aktifitas penting untuk meningkatkan program studi. Hasil survei ini akan digunakan untuk bahan evaluasi pengembangan kurikulum di Program studi Teknik Informatika Universitas 17 Agustus 1945 Surabaya. Kami mohon kesediaan Bapak/Ibu untuk menjawab survei ini. Terima kasih.

I. Biodata

Nama Mahasiswa : Yoga Ananta
NIM : 1462200019
Judul Kerja Praktek : Pengujian Fungsional Dengan Automatic
Testing Untuk Aplikasi Sahabatku

II. Profile Umum

Nama Instansi : PT. Herdina Metatech Sinergi
Alamat : Jl. Dr. Ir. Sockarno No. 470, Surabaya
No. Telepon : 082231342515
Homepage : www.hdnmetatech.com
Pembimbing Lapangan : Aditya Nanda Utama, S.Kom
Jabatan : Komisaris PT. Herdina Metatech Sinergi (Co-Founder)
Email : admin@hdmnetatech.com

III. Kompetensi

Berilah tanda ceklis yang paling sesuai untuk menggambarkan kompetensi Mahasiswa selama melaksanakan Kerja Praktek. Kompetensi pada saat mulai melaksanakan Kerja Praktek:

SB: Sangat Baik
B : Baik
C : Cukup
K : Kurang

Kategori	Penilaian			
	SB	B	C	K
1. Motivasi dalam menyelesaikan pekerjaan	✓			
2. Kreativitas dalam menyelesaikan pekerjaan	✓			
3. Motivasi dalam menambah pengetahuan atau keahlian yang dimiliki	✓			
4. Motivasi dalam menambah pengetahuan atau keahlian diluar bidang ilmu yang dimiliki	✓			
5. Kemampuan dalam memecahkan permasalahan	✓			
6. Kemampuan dalam menuangkan ide atau inovasi	✓			
7. Kemampuan dalam berpikir logis	✓			
8. Kemampuan dalam menyelesaikan pekerjaan	✓			
9. Kemampuan dalam melaporkan hasil pekerjaan	✓			
10. Kemampuan dalam menangani permasalahan	✓			
11. Kemampuan dalam memenuhi segala aturan atau petunjuk kerja	✓			
12. Kemampuan dalam bekerja mandiri	✓			
13. Kemampuan dalam mengerjakan pekerjaan yang sesuai bidang ilmu	✓			
14. Kemampuan berkomunikasi dengan pimpinan	✓			
15. Kemampuan berkomunikasi dengan rekan kerja	✓			
16. Etika dan moral di tempat kerja Praktek	✓			
17. Kemampuan dalam menyelesaikan pekerjaan rutin	✓			

Kategori	Penilaian			
	SB	B	C	K
18. Kemampuan dalam membantu rekan kerja	✓			
19. Kemampuan dalam menyelesaikan masalah tim	✓			
20. Kemampuan dalam berkerjasama dalam tim	✓			

Saran-saran terhadap Mahasiswa Kerja Praktek

Saran-saran untuk perbaikan Program Studi Teknik Informatika Universitas 17 Agustus 1945 Surabaya

Terimakasih atas partisipasi Saudara.

Surabaya, 24 Desember 2025

.Pembimbing Lapangan



(Aditya Nanda Utama, S.Kom)

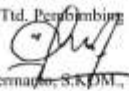
Lampiran 7. Lembar Bimbingan

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS 17 AGUSTUS 1945 SURABAYA

LEMBAR BIMBINGAN KERJA PRAKTEK

Semester Gasal Tahun 2025/2026 Periode : ...

Pas Photo 4 x 6	Nama	: Yoga Ananta
	NBI	: 1462200019
	Alamat Rumah / Kost	: Dusun : Kemayoran, Kembangan, RT/RW : 5/7, Kelurahan : KEMAYORAN, Kecamatan : 052913
	No Telp. / Hp	: 082231709510
	Pembimbing	: Agus Hermanto, S.KOM., M.MT., ITIL, COBIT, SFC (20460150675)
Mulai Bimbingan	Judul KP	: Pengujian Fungsional Dengan Automatic Testing Untuk Aplikasi Sahabatku

PERSETUJUAN DOSEN PEMBIMBING	NILAI
Tanggal : 24, Desember-2025 Ttd. Pembimbing  (Agus Hermanto, S.KOM., M.MT., ITIL, COBIT, SFC)	

LEMBAR BIMBINGAN KERJA PRAKTEK

NO	HARI / TGL	URAIAN MATERI	TT.DOSEN
1	18/Oktober/2025	Pertemuan Bersama dosen pembimbing dan penentuan judul laporan	
2	14/November/2025	Pemaparan Struktur Laporan oleh dosen pembimbing	
3	14/Desember/2025	Pengsheckan Laporan KP dan Bimbingan Revisi Laporan	
4	15/Desember/2025	Revisi Laporan KP	
5	24/Desember/2025	Pengukuhan Laporan KP	

JUDUL KERJA PRAKTEK SETELAH DIREVISI

LEMBAR PENGESAHAN JUDUL KERJA PRAKTEK	
Tanggal : 18, Oktober-2025	
Ttd. Pembimbing	Ttd. Koordinator
	
Agus Hermanto, S.KOM., M.MT., ITIL, COBIT, SFC	Anton Brevia Yumanda S.T., M.MT
NIP : 20460150675	NPP. 20450020554

* Cetak dilembar buffalo kuning

SYARAT MAJU PRESENTASI KERJA PRAKTEK :

1. Bimbingan Kerja Praktek minimal 3x
2. Mengumpulkan Laporan KP beserta Soft Copynya dalam bentuk CD pada saat maju Presentasi.

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS 17 AGUSTUS 1945 SURABAYA

CHECKLIST PROPOSAL KERJA PRAKTEK

Semester Gasal / Genap Tahun 20.../20... Periode : ...

Nama	: Yoga Ananta
NBI	: 1462200019
Alamat Rumah / Kost	: Dusun : Kemayoran, Krembangan, RT/RW : 5/7, Kelurahan : KEMAYORAN, Kecamatan : 052913
No Telp. / Hp	: 082231709510
Pembimbing	: Agus Hermanto, S.KOM., M.MT., ITIL, COBIT, SFC (20460150675)
Judul KP	: Pengujian Fungsional Dengan Automatic Testing Untuk Aplikasi Sahabatku

Dosen Pembimbing wajib memberikan check (✓) untuk tiap point yang telah dipenuhi.

Ketentuan umum yang harus dipenuhi

- ☒ Mahasiswa telah lulus mata kuliah minimal 72 sks
- ☒ Mahasiswa mempunyai IPK minimal 2.50
- ☒ Mahasiswa sudah mencantumkan mata kuliah Kerja Praktek dalam KRS
- ☒ Kerja Praktek sudah sesuai dengan bidang ilmu pada program studi Teknik Informatika
- ☒ Mahasiswa sudah melakukan pembayaran untuk mengikuti mata kuliah Kerja Praktek pada periode saat ini

Sistematika Penulisan Laporan

- ☒ Font yang digunakan adalah Times New Roman dengan ukuran 12
- ☒ Jarak baris pada laporan KP adalah 1.5 spasi
- ☒ Ukuran kertas yang digunakan adalah A4 dengan minimal 50 halaman
- ☒ Ukuran margin yang digunakan sudah sesuai aturan, yaitu right, top, bottom adalah 3 cm, dan left 4 cm
- ☒ Halaman Sampul sampai Daftar Isi diberi nomor halaman dengan huruf: i, ii, iii, dst dan diletakkan pada sudut kanan bawah
- ☒ Halaman Pendahuluan sampai Daftar Pustaka diberi nomor halaman dengan angka arab: 1, 2, 3, ...dst yang diletakkan pada sudut kanan atas, kecuali untuk halaman yang mengandung judul bab diletakkan pada tengah halaman bawah

Surabaya, 24, Desember 2025

Mengetahui,
Koordinator KP

Anton Breva Yunanda S.T., M.MT
NPP. 20450020554

Dosen Pembimbing


Agus Hermanto, S.KOM., M.MT., ITIL, COBIT, SFC
NIP : 20460150675