

PENGGUNAAN *TEST CASE TEMPLATE*
DALAM PROSES *QUALITY*
ASSURANCE



Disusun Oleh:

N a m a : Fadel Muhammad

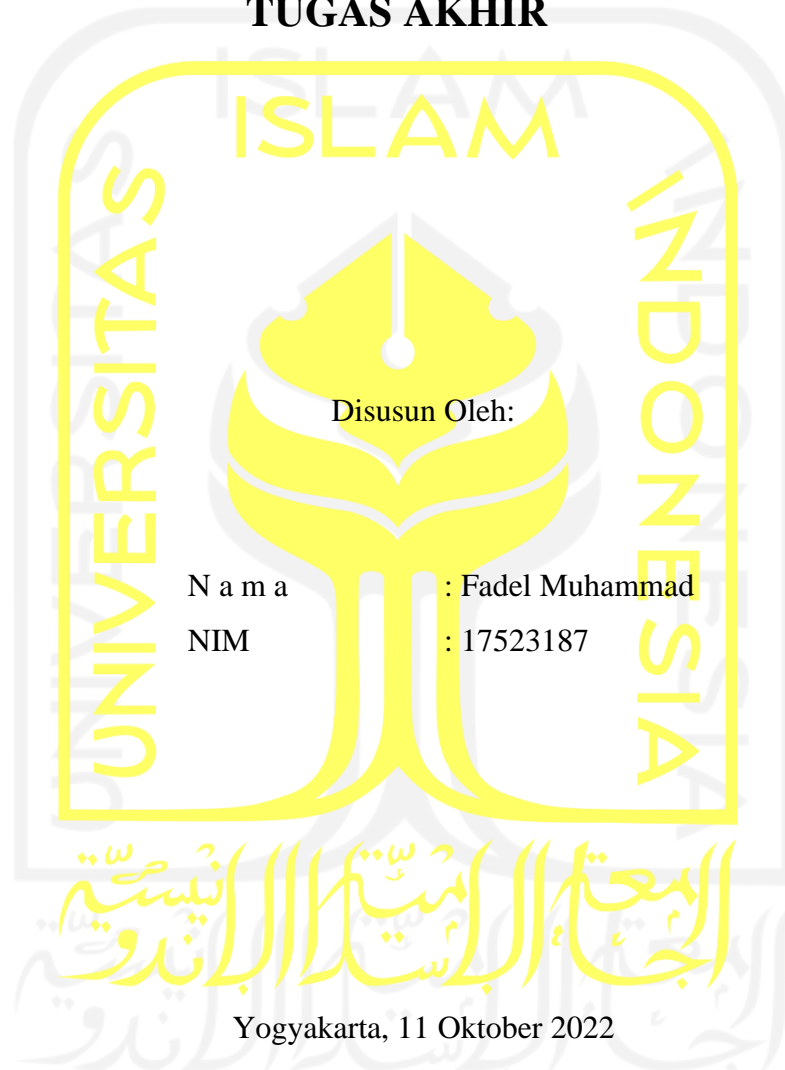
NIM : 17523187

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA

2022

HALAMAN PENGESAHAN DOSEN PEMBIMBING
PENGUNAAN *TEST CASE TEMPLATE*
DALAM PROSES *QUALITY*
ASSURANCE

TUGAS AKHIR



Yogyakarta, 11 Oktober 2022

Pembimbing,

(Hanson Prihantoro Putro, S.T., M.T)

HALAMAN PENGESAHAN DOSEN PENGUJI

PENGUNAAN *TEST CASE TEMPLATE*

DALAM PROSES *QUALITY*

ASSURANCE

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 2 November 2022

Tim Penguji

Hanson Prihantoro Putro, S.T., M.T.

Anggota 1

Nur Wijyaning Rahayu, S.Kom., M.Cs.

Anggota 2

Dr. Syarif Hidayat, S.Kom., M.I.T.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Fadel Muhammad

NIM : 17523187

Tugas akhir dengan judul:

PENGGUNAAN *TEST CASE TEMPLATE*
DALAM PROSES *QUALITY*
ASSURANCE

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 9 Oktober 2022



Fadel Muhammad

HALAMAN PERSEMBAHAN

Laporan Tugas akhir ini saya persembahkan untuk kedua orang tua saya yang selalu mengingatkan dan memberi motivasi untuk menyelesaikan tugas akhir, kepada keluarga saya di Kota Kupang dan Yogyakarta, kepada Bapak Hanson Prihantoro Putro, S.T., M.T., selaku Dosen Pembimbing Akademik dan Dosen Pembimbing Tugas Akhir yang telah membimbing saya sehingga tugas akhir dapat diselesaikan, kepada pengembang aplikasi Ivent yang telah memberi izin untuk menggunakan aplikasi Ivent sebagai objek dalam penelitian.



HALAMAN MOTO

“Skripsi yang bagus bukanlah yang banyak halamannya, skripsi yang bagus adalah yang selesai.”

(Ari Satrio P)

"Beranilah menjalani kehidupan yang anda impikan untuk diri Anda sendiri. Bergeraklah maju dan buatlah impian anda menjadi nyata."

(Ralph Waldo Emerson)

"Jalan awal terbaik untuk mewujudkan segala impian anda adalah bangun dan bangkit dari tempat tidur."

(Paul Valéry)

"Nikmati saja proses psikologis yang dialami sepanjang perjalanan untuk mewujudkan impian."

(Joe Hartanto)

“Dunia itu tempat berjuang, istirahat itu di surga.”

(Syekh Ali Jaber)

“Tidak masalah apabila Anda berjalan lambat, asalkan Anda tidak pernah berhenti berusaha.”

(Confucius)

“Banyak kegagalan yang terjadi dalam hidup, yang mereka tidak sadari betapa dekatnya mereka dengan kesuksesan jika mereka memilih menyerah.”

(Thomas Alva Edison)

“Hidup menjadi berarti ketika Anda menjadi termotivasi, memiliki tujuan, dan mengejarnya dengan cara yang tak bisa terbendung.”

(Les Brown)

KATA PENGANTAR

Segala puji dan syukur saya ucapkan kepada Allah SWT dengan selesainya pembuatan tugas akhir penelitian yang berjudul “Penggunaan *Test Case Template* dalam Proses *Quality Assurance*”.

Tugas akhir ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu (S1) di Jurusan Informatika Fakultas Teknologi Industri UII. Tugas akhir ini membahas tentang penggunaan *test case template* dalam proses *quality assurance* pada aplikasi *E-commerce* berbasis *website* Ivent.

Tanpa dukungan dan bimbingan dari berbagai pihak, penelitian ini tidak akan selesai dengan sebaik ini. Maka saya ingin menyampaikan rasa terima kasih kepada:

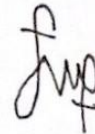
1. Allah SWT yang telah memberikan kehidupan dan kesehatan baik jasmani maupun rohani.
2. Kedua orang tua, saudara-saudara kandung, kakek dan nenek yang berada di kota Kupang yang saya cintai dan hormati. Terima kasih telah memberi dukungan seperti doa, motivasi, perhatian dan materi selama saya melaksanakan kegiatan perkuliahan dan menyelesaikan tugas akhir.
3. Om (Alm) H. Abdur Rahman Arsyad dan tante CH Pujiwiyati beserta keluarga di Yogyakarta yang menerima, menemani, dan selalu memberikan seluruh dukungan sejak awal melaksanakan kegiatan perkuliahan hingga menyelesaikan tugas akhir.
4. Bapak DThomas Hatta Fudholi, S.T., M.Eng., Ph.D., selaku Ketua Program Studi Informatika – Program Sarjana Fakultas Teknologi Industri Universitas Islam Indonesia.
5. Bapak Hanson Prihantoro Putro, S.T., M.T, selaku Dosen Pembimbing Akademik dan Dosen Pembimbing Tugas Akhir yang telah membantu dan membimbing ketika saya mengalami kendala selama menyelesaikan tugas akhir ini.
6. Saudara saya Muhammad Ridwan Firdaus. Terima kasih telah mendengar keluh kesah saya, menyediakan tempat untuk menyelesaikan tugas akhir, selalu memberi dukungan materi dan selalu menyemangati saya dikala sedang terkena masalah.
7. Sahabat-sahabat saya yaitu Sifa Salafiah, Elbo S. Pangestu, Ari Satrio, Galindra Setya, Abdi Humanika, Andres Mondaref J, Ahmad Shofwan Afif, Doddy Ariyansyah, Erwan Dany, Yogi, Muhammad Fahreza, Trenaldy Adisucipto, Prayudha. Terima kasih atas tempat yang diberikan, dukungan ketika ada kendala,

sebagai tempat berbagi keluh kesah dan kenangan indah yang kita jalani bersama selama masa perkuliahan.

8. *Developer* aplikasi Ivent, Aikal Alfrian Mucjal, Muhammad Arif Idris dan Dwiko Nugroho Dani yang telah memberi izin kepada saya untuk menggunakan aplikasi tersebut sebagai objek dalam penelitian.
9. Seluruh Bapak/Ibu dosen Jurusan Informatika Fakultas Teknologi Industri Universitas Islam Indonesia yang telah memberikan ilmu yang sangat bermanfaat selama saya menjalani perkuliahan
10. Dian Ismiyanti. Karena selalu memberi semangat dan dorongan agar segera menyelesaikan tugas akhir.

Harapan penulis adalah laporan tugas akhir ini dapat bermanfaat bagi semua pihak yang membaca dan mempelajarinya terutama dalam bidang *Quality Assurance*.

Yogyakarta, 9 Oktober 2022



Fadel Muhammad

SARI

Aplikasi Ivent merupakan salah satu aplikasi *e-commerce* berbasis *website* yang berfokus pada industri acara seperti *event organizer* dan *vendor*. Dalam proses pengembangannya, pengujian yang dilakukan masih seadanya sehingga masih terdapat beberapa kesalahan dan cacat (*defect*) yang akan mempengaruhi mutu pada aplikasi Ivent. Untuk menyelesaikan permasalahan tersebut dilakukan proses *quality assurance* pada aplikasi Ivent menggunakan *test case template* dan strategi dalam pengujian menggunakan pengujian manual (*manual testing*).

Penggunaan *test case template* dalam pelaksanaan pengujian pada aplikasi Ivent merupakan langkah yang tepat. Karena dengan menggunakan *test case template*, dapat mengetahui fitur-fitur yang masih *error* dengan detail berdasarkan hasil pengujian yang dilakukan. Hasil pengujian yang telah dilakukan akan diserahkan kepada pihak pengembang aplikasi Ivent agar *error* yang terdapat pada aplikasi Ivent segera diperbaiki. Terdapat 5 tahapan dalam proses *quality assurance*, seperti *requirement analysis*, *test planning*, *test case development*, *test execution* dan *closure*.

Hasil dari pengujian yang dilakukan adalah berupa fitur-fitur yang masih terdapat *bug* dan mendapatkan hasil *fail* pada *test case* yang dijalankan. Kode pada fitur tersebut kemudian diperbaiki berdasarkan tingkat prioritas *test case* dan diskusi bersama pihak pengembang aplikasi Ivent lalu dilakukan pengujian lagi. Terdapat 16 fitur pada aplikasi Ivent yang akan dilakukan pengujian. Dari total 45 *test case* yang dijalankan, 43 *test case* mendapatkan hasil *pass* dan 2 *test case* masih mendapatkan hasil *fail*. Penggunaan *test case template* untuk memastikan sistem berjalan dengan baik sesuai dengan kebutuhan awalnya serta sistem dapat memberikan respon ketika diberi masukan yang tidak valid. Penggunaan *test case template* mempermudah dalam pembuatan *test case* sehingga *test case* yang dibuat mudah dipahami baik oleh tester maupun pihak lain serta *test case* yang dibuat menjadi lebih tertata rapi.

Kata kunci: *e-commerce*, *event organizer*, *quality assurance*, *test case template*, Ivent, *test case*

GLOSARIUM

<i>Closure</i>	Tahapan terakhir dari <i>software testing life cycle</i> (STLC) untuk mengevaluasi semua tahapan yang sudah dilakukan mulai dari <i>requirement analysis</i> , <i>test planning</i> , <i>test case development</i> , sampai dengan hasil dari pengujian tiap <i>test case</i> .
<i>Environment</i>	Merupakan hal-hal yang mendukung proses pengujian sebuah aplikasi seperti <i>hardware</i> , <i>software</i> , aplikasi yang diuji dan metode yang akan digunakan.
<i>E-Commerce</i>	Bentuk perdagangan antara konsumen dan penjual secara <i>online</i> dengan pemanfaatan media internet.
<i>Framework</i>	Kerangka kerja yang digunakan untuk membantu proses pengembang sebuah perangkat lunak.
<i>Laravel</i>	Salah satu <i>framework</i> yang membantu pihak pengembang perangkat lunak dalam memaksimalkan penggunaan php pada proses pengembangan sebuah <i>website</i> .
PHP	PHP (<i>Hypertext Preprocessor</i>) salah satu bahasa pemrograman yang digunakan untuk penulis baris kode dalam pembuatan dan pengembang sebuah <i>website</i> .
<i>Stakeholders</i>	Pihak baik individu maupun organisasi yang memiliki kepentingan dalam proses bisnis.
<i>Template</i>	Sebuah dokumen atau berkas yang sudah memiliki format tertentu, digunakan sebagai titik awal dari dokumen baru sehingga tidak harus membuat format baru setiap kali digunakan.
<i>Test Case</i>	Rangkaian tindakan atau aksi yang dilakukan oleh <i>quality assurance tester</i> untuk memverifikasi fitur tertentu dari sebuah perangkat lunak.
<i>Test Suite</i>	Gabungan dari beberapa <i>test case</i> yang akan digunakan di fitur yang sama pada pengujian sebuah perangkat lunak.
<i>Test Plan</i>	Perancangan rencana pengujian seperti langkah-langkah yang akan digunakan, <i>tools</i> dan <i>environment</i> yang akan digunakan selama pengujian pada sebuah perangkat lunak.
<i>Validator</i>	Sebuah fungsi yang akan memvalidasi kondisi yang sudah ditambahkan pada kode program sebuah perangkat lunak.

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING	ii
HALAMAN PENGESAHAN DOSEN PENGUJI	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR	vii
SARI	ix
GLOSARIUM	x
DAFTAR ISI	xi
DAFTAR TABEL	xiii
DAFTAR GAMBAR	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	5
1.3 Batasan Masalah	5
1.4 Tujuan Penelitian	5
1.5 Manfaat Penelitian	6
1.6 Metodologi Penelitian	6
1.7 Sistematika Penulisan	6
BAB II LANDASAN TEORI	8
2.1 Ivent, <i>Event Organizer</i> dan Vendor	8
2.2 <i>Software Testing</i>	9
2.3 <i>Quality Assurance</i>	11
2.4 <i>Test Case Management</i>	14
2.5 <i>Test Documentation</i>	15
2.6 <i>Test Template</i>	18
2.7 <i>Literature Review</i>	23
BAB III ANALISIS DAN PERENCANAAN	27
3.1 <i>Requirement Analysis</i>	27
3.1.1 Analisis Proses Bisnis	27
3.1.2 Analisis Kebutuhan Fungsional	28
3.1.3 Tampilan Antarmuka Aplikasi Ivent	30
3.2 <i>Test Planning</i>	32
BAB IV PENGEMBANGAN DAN EKSEKUSI	34
4.1 <i>Test Case Development</i>	34
4.2 <i>Test Execution</i>	50
4.2.1 Hasil Pengujian Pertama	50
4.2.2 Tindak Lanjut	61
4.2.3 Hasil Pengujian Kedua	63
4.2.4 Waktu Pengujian	67
4.3 <i>Closure</i>	68
4.4 Kontribusi Penelitian	69
BAB V KESIMPULAN DAN SARAN	70
5.1 Kesimpulan	70
5.2 Saran	71
DAFTAR PUSTAKA	72



DAFTAR TABEL

Tabel 1.1 Hasil pengujian sebelumnya pada aplikasi Ivent (Mucjal, 2021).....	3
Tabel 2.1 Model <i>test case</i> (Hasibuan & Dirgahayu, 2021).....	19
Tabel 2.2 <i>Literature review</i>	23
Tabel 3.1 Daftar fitur-fitur aplikasi berbasis <i>website</i> Ivent yang akan diuji.....	33
Tabel 4.1 <i>Test Suites</i> Halaman <i>Register</i>	36
Tabel 4.2 <i>Test Suites</i> Halaman <i>Login</i>	37
Tabel 4.3 <i>Test Suites</i> lupa <i>password</i>	38
Tabel 4.4 <i>Test Suites</i> <i>Menu</i> pada <i>Homepage</i>	38
Tabel 4.5 <i>Test Suites</i> fitur <i>Pencarian</i>	39
Tabel 4.6 <i>Test Suites</i> menambahkan produk.....	39
Tabel 4.7 <i>Test Suites</i> <i>edit</i> produk toko.....	42
Tabel 4.8 <i>Test suites</i> melihat informasi produk.....	43
Tabel 4.9 <i>Test suites</i> memesan produk	44
Tabel 4.10 <i>Test suites</i> melihat pesanan.....	44
Tabel 4.11 <i>Test suites</i> menambahkan alamat <i>user</i> (konsumen).....	45
Tabel 4.12 <i>Test suites</i> menghapus pesanan.....	46
Tabel 4.13 <i>Test suites</i> melihat transaksi	47
Tabel 4.14 <i>Test suites</i> unggah bukti pembayaran	47
Tabel 4.15 <i>Test suites</i> mengecek pesanan penjual.....	48
Tabel 4.16 <i>Test suites</i> verifikasi bukti pembayaran.....	48
Tabel 4.17 Hasil pengujian fitur <i>register</i>	50
Tabel 4.18 Hasil pengujian fitur <i>login</i>	51
Tabel 4.19 Hasil pengujian fitur lupa <i>password</i>	52
Tabel 4.20 Hasil pengujian fitur <i>menu</i>	53
Tabel 4.21 Hasil pengujian fitur <i>pencarian</i>	53
Tabel 4.22 Hasil pengujian fitur <i>tambahkan produk</i>	53
Tabel 4.23 Hasil pengujian fitur <i>edit</i> produk	54
Tabel 4.24 Hasil pengujian fitur <i>melihat produk</i>	55
Tabel 4.25 Hasil pengujian fitur <i>memesan produk</i>	56
Tabel 4.26 Hasil pengujian fitur <i>melihat pesanan</i>	56
Tabel 4.27 Hasil pengujian fitur <i>menambahkan alamat</i>	57
Tabel 4.28 Hasil pengujian fitur <i>menghapus pesanan</i>	57

Tabel 4.29 Hasil pengujian fitur melihat transaksi	58
Tabel 4.30 Hasil pengujian fitur unggah bukti pembayaran.....	59
Tabel 4.31 Hasil pengujian fitur mengecek pesanan untuk penjual	59
Tabel 4.32 Hasil pengujian verifikasi bukti pembayaran oleh admin	60
Tabel 4.33 Ringkasan pengujian pertama.....	61
Tabel 4.34 Daftar <i>test case</i> yang <i>fail</i>	61
Tabel 4.35 Hasil pengujian kedua.....	63
Tabel 4.36 Ringkasan pengujian keseluruhan	67



DAFTAR GAMBAR

Gambar 1.1 Contoh standar <i>field sample test case template</i> (Softwaretestinghelp, 2022).....	4
Gambar 2.1 Proses <i>quality assurance</i>	14
Gambar 2.2 Langkah untuk mengaktifkan makro pada Excel.....	18
Gambar 2.3 Langkah untuk <i>Record Macros</i>	18
Gambar 2.4 Contoh <i>test case template</i> (Neumeyer, 2022).	20
Gambar 3.1 Tampilan halaman <i>login</i> Ivent	30
Gambar 3.2 Tampilan halaman <i>register</i> Ivent	31
Gambar 3.3 <i>Homepage</i> Ivent	31
Gambar 3.4 Tampilan halaman detail produk.....	32
Gambar 4.1 <i>Template</i> yang akan digunakan pada pengujian aplikasi Ivent.....	35
Gambar 4.2 Hasil pengujian fitur <i>register</i>	51
Gambar 4.3 Hasil pengujian fitur lupa <i>password</i>	52
Gambar 4.4 Hasil pengujian fitur tambahkan produk.....	54
Gambar 4.5 Hasil pengujian fitur <i>edit</i> produk	55
Gambar 4.6 Hasil pengujian fitur memesan produk	56
Gambar 4.7 Hasil pengujian fitur menambahkan alamat.....	57
Gambar 4.8 Hasil pengujian fitur menghapus pesanan	58
Gambar 4.9 Hasil pengujian fitur mengunggah bukti pembayaran	59
Gambar 4.10 Hasil pengujian fitur mengecek pesanan untuk penjual.....	59
Gambar 4.11 Hasil pengujian fitur verifikasi bukti pembayaran.....	60
Gambar 4.12 Hasil pengujian kedua fitur <i>edit</i> produk.....	64
Gambar 4.13 Hasil pengujian kedua fitur memesan produk.....	64
Gambar 4.14 Hasil pengujian kedua fitur menambahkan alamat	65
Gambar 4.15 Hasil pengujian kedua fitur mengunggah bukti pembayaran.....	65
Gambar 4.16 Hasil pengujian kedua fitur verifikasi bukti pembayaran	66
Gambar 4.17 Waktu pengujian setiap <i>test case</i>	68

BAB I PENDAHULUAN

1.1 Latar Belakang

Teknologi informasi pada masa kini telah berkembang dengan sangat pesat dan menghasilkan perubahan kehidupan masyarakat dalam berbagai bidang. Perkembangan teknologi ini memaksa kita untuk bertindak secara efisien dan efektif dalam berbagai bidang terutama dalam proses transaksi. Kebutuhan masyarakat tersebut kemudian diwujudkan oleh perusahaan dan berbagai penyedia jasa layanan untuk memberikan akses informasi serta menghubungkan penyedia jasa dan barang dengan konsumen melalui penerapan aplikasi yang bersifat modern seperti teknologi *e-commerce* (Firmansyah, 2018). Dengan adanya *e-commerce*, konsumen tidak perlu lagi datang mengunjungi toko dan mengantri ketika akan melakukan transaksi. Pemanfaatan media komputer maupun *smartphone* yang tersambung dengan internet dapat membantu masyarakat melakukan transaksi pada *e-commerce* secara cepat dan mudah. Hal tersebut merupakan kelebihan dari aktivitas ekonomi digital dan menjadi salah satu daya tarik yang besar bagi konsumen untuk melakukan transaksi pembelian pada aplikasi *e-commerce*.

Aplikasi *e-commerce* sendiri tersedia pada perangkat *smartphone* (*mobile apps*) dan juga *e-commerce* berbasis *website*. Aplikasi berbasis *website* sendiri memiliki peran yang penting dalam perkembangan era digital. Sehingga perlu dipastikan *website* tersebut aman, tepat dan efisien dengan cara melakukan pengujian pada *website* tersebut. Karena pentingnya peran aplikasi *website* ini, maka cacat yang terdapat pada perangkat lunak tersebut dapat memberikan dampak yang signifikan bagi pengguna dan juga penyedia layanan. Fakta menunjukkan bahwa pengujian perangkat lunak merupakan komponen penting dalam proses *software development life cycle* (SDLC) (Bahaweres et al., 2017). Sebanyak 40% dari waktu pengembangan perangkat lunak digunakan untuk melakukan pengujian demi memastikan kualitas dari perangkat lunak yang dikembangkan (Pamungkas & Rochimah, 2019). Studi menunjukkan bahwa peningkatan biaya dalam proses pengujian perangkat lunak lebih dari 50%. Strategi dalam pengujian bisa menggunakan pengujian secara manual (*manual testing*) atau pengujian secara otomatis (*automated testing*). *Manual testing* dilakukan dengan cara membuat *test case* secara manual. Pengujian secara manual rentan terhadap kesalahan yang dilakukan manusia (*human errors*). *Automated testing* menggunakan alat (*tools*) yang dapat mengeksekusi *test case* berdasarkan input yang dilakukan oleh pengguna.

Aplikasi berbasis *website* Ivent merupakan aplikasi *e-commerce* yang menghubungkan konsumen dengan pemilik usaha *event organizer* atau vendor (Mucjal, 2021). Dengan adanya aplikasi Ivent, konsumen lebih mudah untuk menentukan pilihan karena vendor dan juga *event organizer* terdapat dalam satu *platform*. Ivent merupakan aplikasi penyewaan barang atau jasa yang berfokus pada industri acara seperti *event organizer* dan vendor. Dalam proses pengembangan aplikasi Ivent, pengujian yang dilakukan sebelumnya oleh pihak developer menggunakan metode *black box testing* dan *user acceptance testing*. Metode *black box testing* dilakukan dengan cara menampilkan perangkat lunak untuk melihat antarmuka dan memberikan aksi ke perangkat lunak untuk melihat respon dari perangkat lunak tersebut dan membuat skenario sebagai konsumen, penyedia jasa *event organizer* dan vendor serta bagian admin dari aplikasi Ivent. Metode *user acceptance testing* dilakukan oleh pengguna dengan cara berinteraksi langsung kepada perangkat lunak yang dikembangkan dengan tujuan memeriksa keseluruhan fungsi sudah berjalan sesuai spesifikasi atau tidak. Pada pengujian *black box* aplikasi Ivent berhasil menjalankan keseluruhan fungsi dengan baik, sedangkan pengujian *user acceptance testing* mendapatkan hasil dengan total presentasi 94,4% berdasarkan bobot penilaian yang telah ditentukan dengan skala 0%-100%.

Saat ini aplikasi Ivent sudah selesai dibuat namun belum sampai ke tahap *deployment* untuk *website*. Hasil pengujian yang dilakukan dapat dilihat pada Tabel 1.1. Aplikasi Ivent sudah dilakukan pengujian namun belum dilakukan proses *quality assurance*. Hasil dari pengujian sebelumnya yang dilakukan developer berbeda dengan hasil ketika penulis menggunakan aplikasi Ivent, masih terdapat beberapa kesalahan yang ditemukan. Kesalahan yang ditemukan seperti, produk yang ingin diubah malah terhapus ketika menekan tombol ubah produk, fitur untuk mengubah kata sandi masih belum bisa digunakan, dan masih terdapat permasalahan lain yang tidak ditemukan langsung oleh pengguna. Untuk menyelesaikan permasalahan tersebut, pada penelitian ini akan dilakukan proses *quality assurance* pada aplikasi Ivent menggunakan *test case template*.

Quality assurance merupakan serangkaian proses sistematis untuk memastikan bahwa aplikasi yang dibuat sesuai dengan ketentuan yang telah ditetapkan (Hosal et al., 2021). *Test case template* merupakan proses perencanaan serangkaian tes yang digunakan oleh *user* untuk melakukan verifikasi terhadap suatu fitur atau fungsi dari perangkat lunak (Kaner, 2003). *Test case* memiliki beberapa komponen seperti *Test Case ID*, *test case description*, *pre-condition*, *test steps*, *test data*, *Expected Result*, *Actual Result* dan *status*. Penggunaan *test case management* untuk memastikan sistem berjalan dengan baik dan sesuai dengan kebutuhan

awalnya serta sistem dapat memberikan respon ketika diberi masukan yang tidak valid. *Test case* sendiri bertindak sebagai titik mula dari sebuah pengujian perangkat lunak. Dengan menggunakan *test case*, *user* dapat mengetahui apakah fitur pada sistem berjalan dengan normal atau tidak.

Tabel 1.1 Hasil pengujian sebelumnya pada aplikasi Ivent (Mucjal, 2021).

No	Pertanyaan	1	2	3	4	5	Total
1	Apakah keseluruhan tampilan dari <i>website</i> Ivent sudah menarik dan sesuai kebutuhan?	0	0	2	4	4	42
2	Apakah menu pada <i>website</i> Ivent mudah digunakan?	0	0	0	0	10	50
3	Apakah terdapat kesulitan saat menggunakan menu pada <i>website</i> Ivent?	0	0	0	0	10	50
4	Apakah terdapat kesalahan pada saat menjalankan menu tertentu?	0	0	0	0	10	50
5	Apakah terjadi kesulitan dalam melakukan pemesanan produk vendor dan jasa <i>event organizer</i> ?	0	0	0	0	10	50
6	Apakah <i>website</i> Ivent ini dapat menjadi referensi anda untuk mencari <i>event organizer</i> dan vendor?	0	0	0	4	6	46
7	Apakah terjadi kesalahan saat melakukan pencarian <i>event organizer</i> dan vendor?	0	0	0	0	10	50
8	Apakah <i>website</i> Ivent berhasil menampilkan produk sesuai kategorinya (<i>event organizer</i> atau vendor)?	0	0	0	4	6	46
9	Apakah <i>website</i> Ivent dapat mempermudah anda mencari <i>event organizer</i> dan vendor?	0	0	0	7	3	43
10	Apakah sistem transaksi dan pembayaran pada <i>website</i> Ivent mudah dilakukan?	0	0	3	4	3	40
TOTAL		0	0	5	23	72	467

Tersedia beberapa *test case template* di internet yang dapat digunakan dalam proses *quality assurance*. *Template* yg tersedia dapat digunakan dan disesuaikan dengan tim pengujian proyek serta aplikasi yang akan diuji (Softwaretestinghelp, 2022).

Seperti yang dapat dilihat pada Gambar 1.1, terdapat beberapa atribut yang digunakan pada *sample test case template* seperti *Project Name*, *Test Priority (Low/Medium/High)*, *Module Name*, *Test Title*, *Description*, *Test Design by*, *Test Design date*, *Test Execute by*, *Test Execute date*, *Pre-Condition*, *Dependencies*, *Step*, *Test Steps*, *Test Data*, *Expected Result*, *Actual Result*, *Status (Pass/Fail)* dan *Notes*.

Test Case Template						
Project Name:						
Test Case ID: Fun_10			Test Designed by: <Name>			
Test Priority (Low/Medium/High): Med			Test Designed date: <Date>			
Module Name: Google login screen			Test Executed by: <Name>			
Test Title: Verify login with valid username and password			Test Execution date: <Date>			
Description: Test the Google login page						
Pre-conditions: User has valid username and password						
Dependencies:						
Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to login page	User= smangin@gmail.com	User should be able to login	User is navigated to	Fail	
2	Provide valid username	Password: 1234		dashboard with successful		
3	Provide valid password			login		
4	Click on Login button					
Post-conditions:						
User is validated with database and successfully login to account. The account session details are logged in database.						

Gambar 1.1 Contoh standar *field sample test case template*
(Softwaretestinghelp, 2022)

Selain atribut yang sudah disebutkan, dapat juga menambahkan beberapa atribut berikut sesuai dengan kebutuhan. Atribut lain yang dapat ditambahkan yaitu, 1. *Defect ID/Link*, berisi tautan ke log kerusakan jika status fail, 2. *Test Type / Keywords*, *field* ini dapat digunakan untuk mengklasifikasikan tes berdasarkan jenis tes, 3. *Requirements*, persyaratan untuk yang digunakan untuk *test case*. Persyaratan disesuaikan dengan *test case* mana saja yang akan digunakan, 4. *Attachment / Reference*, *field* ini berguna untuk skenario pengujian yang kompleks untuk menjelaskan langkah pengujian atau hasil yang diharapkan menggunakan diagram visio sebagai referensi. Jika menggunakan *field* ini, sertakan tautan atau lokasi ke jalur sebenarnya dari diagram atau dokumen dan yang terakhir 5. *Automation (Yes/No)*, apakah *test case* ini dijalankan secara otomatis atau tidak. *Field* ini berguna untuk melacak status otomatisasi ketika *test case* diotomatisasi.

Ivent merupakan aplikasi *e-commerce* berbasis *website* yang bergerak di sektor industri acara. Berbagai acara *event* seperti konser, pernikahan dan lain sebagainya saat ini sudah diperbolehkan untuk dilakukan dan merupakan peluang bisnis yang besar bagi pihak pengembang. Aplikasi Ivent dipilih sebagai objek penelitian karena masih terdapat beberapa

kesalahan dan belum pernah dilakukan proses *quality assurance* dibandingkan dengan aplikasi serupa yang sudah rilis seperti eventbrite.com. Penggunaan *test case template* dalam pelaksanaan pengujian pada aplikasi berbasis *website* Ivent merupakan langkah yang tepat. Karena dengan menggunakan *test case template*, dapat mengetahui fitur-fitur yang masih *error* dengan detail berdasarkan hasil pengujian yang dilakukan. Kemudian hasil pengujian yang dilakukan akan diserahkan kepada developer aplikasi berbasis *website* Ivent agar *error* yang terdapat pada aplikasi tersebut segera diperbaiki.

Untuk meningkatkan mutu pada aplikasi Ivent maka dilakukan proses *quality assurance* menggunakan *test case template*. *Test case template* dalam proses *quality assurance* digunakan untuk membantu pengujian dalam mencari *error* yang masih terdapat pada aplikasi Ivent, menganalisis hasil dari pengujian yang dilakukan untuk mengetahui apakah sistem sudah berjalan sesuai dengan *software requirement* yang sudah ada. Diharapkan penelitian ini dapat menemukan *error* yang tidak terlihat pada pengujian sebelumnya, membantu pihak pengembang untuk memperbaiki aplikasi Ivent berdasarkan *error* yang ditemukan pada pengujian ini serta untuk meningkatkan mutu pada aplikasi Ivent.

1.2 Rumusan Masalah

Berdasarkan permasalahan yang terdapat pada latar belakang di atas maka rumusan masalah yang diangkat yaitu bagaimana melakukan proses *quality assurance* pada aplikasi *website e-commerce* Ivent menggunakan *test case template* agar berjalan sesuai dengan *software requirement* yang sudah ada.

1.3 Batasan Masalah

Batasan masalah yang perlu diperhatikan pada penelitian ini sebagai berikut:

- a. Perangkat lunak yang akan diuji merupakan aplikasi *e-commerce* berbasis *website* Ivent.
- b. Pengujian perangkat lunak berdasarkan *test case* yang dikaji dalam penelitian ini.
- c. Pengujian perangkat lunak akan menggunakan pengujian manual (*manual testing*).

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah melakukan proses *quality assurance* pada aplikasi berbasis *website* Ivent dengan menggunakan *test case template* serta menganalisis hasil dari pengujian yang dilakukan untuk mengetahui apakah sistem sudah berjalan sesuai dengan *software requirement* yang sudah ada.

1.5 Manfaat Penelitian

Manfaat dari penggunaan *test case template* dalam proses *quality assurance* pada penelitian ini sebagai berikut:

- a. Menemukan kesalahan (*errors*) dan cacat (*defect*) yang tidak terlihat pada pengujian perangkat lunak aplikasi berbasis *website* Ivent sebelumnya.
- b. Membantu pengembang aplikasi untuk memperbaiki aplikasi Ivent berdasarkan kesalahan (*errors*) dan cacat (*defect*) yang telah ditemukan.
- c. Meningkatkan jaminan mutu pada aplikasi berbasis *website* Ivent.

1.6 Metodologi Penelitian

Terdapat 5 tahapan yang akan dilakukan dalam proses *quality assurance* pada aplikasi berbasis *website* Ivent. Tahapan yang akan dilakukan sebagai berikut:

1. *Requirement analysis*, tahapan ini akan menganalisis *software requirements* yang didapat dari *stakeholders*.
2. *Test planning*, tahapan ini melakukan perencanaan seperti teknik pengujian, *tools* dan *environment* yang akan digunakan.
3. *Test case development*, tahapan ini akan membuat *test case*, *test data* dan *test script*.
4. *Test execution*, tahapan ini merupakan pelaksanaan pengujian berdasarkan *test plan* dan *test case* yang sudah dibuat pada tahapan sebelumnya serta mendokumentasikan hasil dari pengujian.
5. *Closure*, tahapan ini akan mengevaluasi semua tahapan yang sudah dilakukan sebelumnya.

1.7 Sistematika Penulisan

Sistematika penulisan dibagi ke dalam beberapa bab untuk memberikan gambaran terhadap permasalahan yang diangkat. Beberapa bab tersebut yaitu:

BAB I PENDAHULUAN

Bab pertama akan dijelaskan masalah umum yang diangkat seperti: latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan.

BAB II LANDASAN TEORI

Pada bab kedua ini, berisi tentang penjelasan dasar teori yang berkaitan dengan *software testing*, *quality assurance*, *test case management* serta *test template*.

BAB III ANALISIS DAN PERENCANAAN

Pada bab ketiga ini, berisi tentang analisis *software requirement* dan perencanaan pengujian pada aplikasi berbasis *website* Ivent.

BAB IV PENGEMBANGAN DAN EKSEKUSI

Pada bab ke empat ini, berisi penjelasan tentang tahapan pembuatan *test suites*, pengujian yang sudah dilakukan serta hasil dari proses *quality assurance* pada aplikasi berbasis *website* Ivent.

BAB V KESIMPULAN DAN SARAN

Pada bab kelima ini, berisi tentang kesimpulan dan saran dari proses *quality assurance* pada aplikasi berbasis *website* Ivent.

BAB II LANDASAN TEORI

2.1 Ivent, *Event Organizer* dan Vendor

Menurut (Whinston, 1997) *e-commerce* atau *electronic commerce* adalah kegiatan perdagangan (penjualan, pembelian, pertukaran produk, layanan dan informasi) baik jasa dan barang yang dilakukan menggunakan sarana dan teknologi elektronik, termasuk proses transaksi *business to business*, *business to consumer* dan *consumer to business* (Ramadhan et al., 2019). Pengiriman produk ataupun layanan terjadi menggunakan internet. Menurut (Hartman, 2000) *e-commerce* adalah jenis dari mekanisme bisnis secara elektronik yang berfokus pada transaksi bisnis berbasis individu dengan menggunakan internet sebagai media pertukaran barang atau jasa antara dua institusi (*business to business*) maupun antara institusi dan konsumen (*business to consumer*) (Romindo, 2019). Adapun keuntungan dari penggunaan *e-commerce* seperti (Anjali, 2014):

1. Pengurangan biaya pencarian karena pembeli tidak melalui banyak perantara untuk mencari informasi mengenai produk dan harga seperti pada rantai pasokan tradisional.
2. Pengurangan biaya pemrosesan transaksi karena memungkinkan otomatisasi proses transaksi.
3. Pemrosesan *online* meningkatkan manajemen inventaris dan logistik. Pemasok dapat berinteraksi langsung dengan pembeli sehingga menghilangkan perantara dan distributor.

Event organizer adalah profesi yang berfokus pada bidang industri acara. Di dalam EO terdapat banyak pembagian tugas untuk setiap anggota yang mengacu kepada seluruh tujuan yang sama dalam suatu acara (Dani, 2021). Vendor adalah pihak yang memiliki tanggung jawab sebagai penjual ataupun penyewaan barang baik itu sebuah institusi atau badan usaha. (Afifah & Setyantoro, 2021). *E-commerce* di bidang acara baik untuk *event organizer* maupun vendor sudah mulai dikembangkan. *E-commerce* berbasis *website* terbukti dapat digunakan untuk menyelesaikan permasalahan pemasaran atau penjualan suatu produk barang melalui jalur komunikasi digital. Sistem penjualan berbasis *website* mempermudah dalam pemasaran suatu jasa/produk yang ditawarkan pada bidang acara. Selain itu *e-commerce* juga dapat digunakan untuk berbagai transaksi dalam suatu pemesanan produk / jasa (Dwi Juniansyah et al., 2020). Beberapa aplikasi *E-commerce* yang bergerak di bidang acara serupa dengan aplikasi Ivent yaitu Akusewa.com yang berfokus pada penyedia vendor, Bridestory.com yang

berfokus pada *wedding organizer* dan Evenbrite.com yang berfokus pada penjualan tiket untuk acara.

Ivent merupakan aplikasi *e-commerce* yang mendukung pergerakan bisnis pada bidang industri acara dengan cara menghubungkan antara konsumen dengan pemilik usaha *event organizer* dan vendor yang memudahkan konsumen untuk lebih bebas menentukan pilihan mereka karena berbagai *eo* dan vendor terdapat dalam satu *platform* (Mucjal, 2021). Ivent merupakan *e-commerce* yang menyediakan proses penyewaan jasa / barang yang berfokus pada bidang industri acara seperti *event organizer* dan vendor. Aplikasi Ivent sudah selesai dikembangkan namun belum sampai ke tahap *deployment* ke *website*. Aplikasi Ivent masih harus dilakukan proses *quality assurance* sebelum dideploy ke *website*. Dalam proses *quality assurance* akan dilakukan pengujian perangkat lunak yang berfungsi untuk mendeteksi adanya kesalahan pada perangkat lunak yang menyebabkan kegagalan perangkat lunak tersebut. Pengujian perangkat lunak juga didefinisikan sebagai proses verifikasi dan validasi terhadap perangkat lunak untuk memastikan bahwa perangkat lunak sudah memenuhi persyaratan teknis dan bisnis seperti yang diharapkan pihak pengembang perangkat lunak (Irawan, 2017).

2.2 Software Testing

Pengujian perangkat lunak (*software testing*) merupakan komponen penting *software quality assurance*. Berbagai organisasi yang bergerak di bidang perangkat lunak menghabiskan sumber daya sebesar 40% untuk melakukan pengujian perangkat lunak. Biaya pengujian perangkat lunak bisa beragam, tergantung dengan seberapa penting perangkat lunak tersebut. Contohnya perangkat lunak yang digunakan pada penerbangan, biaya untuk pengujiannya lebih mahal dibandingkan pengujian perangkat lunak lainnya (István & Simon, 2009).

Pengujian merupakan aktivitas untuk mengevaluasi dan meningkatkan kualitas perangkat lunak tersebut. Tujuan dilakukan pengujian adalah untuk menemukan kesalahan (*error*) dan juga cacat (*defect*) pada perangkat lunak. Jika perangkat lunak tersebut sudah memenuhi persyaratan yang diinginkan, maka dapat dikatakan perangkat lunak tersebut sudah memenuhi keinginan pengguna.

Kualitas sebuah perangkat lunak bergantung pada kepuasan pengguna dan kualitas tersebut harus dijaga dengan alasan sebagai berikut:

1. Dapat bertahan hidup di dunia bisnis
2. Dapat bersaing dengan kompetitor lainnya
3. Kualitas perangkat lunak penting untuk pemasaran secara global

4. Menghemat biaya agar tidak banyak membuang perangkat lunak yang gagal dipasarkan atau diproduksi
5. Mempertahankan pengguna dan meningkatkan keuntungan (Cholifah et al., 2018).

Pengembang atau penguji perangkat lunak harus menyediakan sesi untuk menguji program yang sudah dibuat agar kesalahan (*error*) dan cacat (*defect*) dapat diidentifikasi dan diperbaiki secepat mungkin. Pengujian itu sendiri merupakan bagian penting dari *software quality assurance* dan merupakan bagian dari siklus hidup pengembangan perangkat lunak (Sebastianus, 2021).

Penelitian berikut mengkaji perancangan pengujian perangkat lunak pada aplikasi pendaftaran kp dan ta menggunakan metode *black box* (Novianingsih, 2022). Tujuan pada penelitian ini adalah menghasilkan dokumentasi perancangan pengujian yang mendetail untuk aplikasi pendaftaran KP-TA di Teknik Informatika Universitas Pasundan. Pada penelitian ini menggunakan metode *black box testing* berbasis *equivalence partitioning* dan *boundary value analysis*. Hasil dari penelitian ini adalah kesimpulan penyusunan dokumen pengujian pada saat melakukan perancangan *test case* dengan menetapkan nilai input dari pengujian, sehingga terdapat batasan-batasan input serta output pada pengujian yang dilakukan.

Jurnal berikutnya mengkaji perbandingan 2 teknik *white box testing* pada studi kasus sistem informasi *reporting community TB-HIV care* (Yunisa & Putro, 2019). Jurnal ini menerapkan *white box testing* dengan cara melakukan pengecekan terhadap source code program pada sistem informasi *reporting community TB-HIV care*. Teknik dari pengujian *white box* yang digunakan adalah *statement coverage* dan *branch coverage*. Hasil identifikasi pengujian yang dilakukan memperoleh 7 tahapan dalam melakukan pengujian. Perbandingan yang lain memperoleh terakit objek eksekusi, kebutuhan data uji dan kelengkapan *coverage*, cakupan *branch testing* lebih luas dari *statement coverage testing*.

Dari hasil wawancara saya dengan seorang *software engineer* yg berinteraksi dengan QA dalam proses *development software* di salah satu perusahaan SaaS di Indonesia. maka diketahui tahapan dalam pengujian aplikasi seperti:

1. Membuat *requirements features / bugfix* yang dibutuhkan
2. *Testing Staging* dan *production*.
3. Pembuatan *test case (Manual testing & automation)*.
4. Verifikasi dan mendapatkan persetujuan dari pihak QA untuk dilanjutkan ke tahap *production*.

Hasil wawancara saya dengan pihak lain yang bekerja sebagai QA dalam sebuah perusahaan, maka diketahui pengujian yang dilakukan seperti:

1. Pengujian *prototype* (html).
2. Pengujian Fungsionalitas (*backend*).
3. Pengujian API menggunakan Insomnia.
4. Pembuatan skenario pengujian menggunakan Google Sheet dan *board project management* (Trello).

Adapun hasil wawancara dengan pihak yang sama dan bekerja sebagai QA tetapi dalam sebuah perusahaan yang berbeda, diketahui pengujian yang dilakukan seperti:

1. *Replicate* dan *feedback* permasalahan *user* dilakukan dengan menggunakan *tools* Postman.
2. *Testing Fungsional & Code* (pada fitur yang sudah *release*, memerlukan perbaikan secepatnya).
3. Pembuatan skenario pengujian menggunakan aplikasi Qase.io.

2.3 *Quality Assurance*

Penjamin mutu perangkat lunak atau biasa *software* disebut *quality assurance* adalah proses mengevaluasi dan mendokumentasikan yang telah dibuat selama siklus pengembangan perangkat lunak (SDLC) (Hariyanto et al., 2020). Tujuan utama dilakukan *quality assurance* untuk memastikan produk yang telah dibuat telah sesuai dengan persyaratan yang telah ditetapkan.

Software testing dan *quality assurance* merupakan bagian yang tidak terpisahkan dari siklus pengembangan perangkat lunak (SDLC) seperti halnya desain, *code* dan analisis. *Software testing* sendiri merupakan elemen kritis dari *software quality assurance* (Mustaqbal et al., 2015). *Software testing* sendiri fokus pengujiannya pada produknya atau *software*, *quality assurance* berfokus pada keseluruhan proses yang digunakan dalam pengembangan perangkat lunak termasuk *testing* itu sendiri. *Quality assurance* adalah pendekatan manajemen secara keseluruhan untuk memastikan keberhasilan penerapan tujuan kualitas perusahaan. *Quality assurance* mencakup semua proses, kebijakan, standar, alat, pelatihan dan lainnya yang diperlukan untuk mendukung keberhasilan pencapaian sasaran mutu. *Software testing* adalah bagian yang sangat kecil dari strategi *quality assurance* yang terintegrasi dengan baik. Strategi *quality assurance* yang baik harus proaktif dan reaktif.

Quality assurance dapat meminimalisir risiko fungsional dan non-fungsional yang ada pada tahapan pengembangan perangkat lunak. Risiko fungsional seperti perangkat lunak yang tidak berfungsi seperti yang diharapkan dan perangkat lunak yang sulit digunakan pengguna. Adapun risiko non-fungsional seperti, biaya pengembangan perangkat lunak yang melebihi anggaran. Kegagalan pengembangan perangkat lunak diakibatkan karena tidak adanya proses *quality assurance*. Dengan adanya proses *quality assurance*, diharapkan dapat membantu memenuhi kebutuhan perangkat lunak yang diharapkan. *Software quality assurance* juga mendokumentasikan setiap proses yang dilakukan sehingga dapat membuktikan pengembangan perangkat lunak telah dilakukan dengan baik.

Penelitian yang dilakukan berikutnya mengkaji *Software Quality Assurance* pada Perusahaan Pengembang Perangkat Lunak Skala Kecil dan Menengah (Hariyanto et al., 2020). Penelitian ini menerapkan SQA (*Software Quality Assurance*) pada perusahaan perangkat lunak skala kecil dan menengah. Penelitian ini bertujuan untuk melihat hasil penerapan SQA dari perspektif skala perusahaan yang beragam. Metode yang digunakan dalam penelitian ini adalah kualitatif yang meliputi wawancara dengan berbagai pihak penting dan juga studi dokumen terkait SQA. Hasil dari penelitian ini menunjukkan bahwa perusahaan skala kecil dan menengah telah menerapkan SQA dan tiap perusahaan mempunyai standar tertentu untuk metode pengembangan perangkat lunak. Kualitas hasil akhir dari pengembangan perangkat lunak bergantung pada kapasitas sumber daya alam setiap perusahaan.

Jurnal yang dilakukan berikutnya mengkaji *Quality Assurance Testing* Aplikasi *Human Resource Splendid* Berbasis *Website* Pada PT. Garasilabs Manivesta (Jamali, 2020). Jurnal ini menerapkan SQA (*Software Quality Assurance*) *testing* dalam proses pembuatan aplikasi *splendid* untuk melakukan pengecekan terhadap kualitas aplikasi tersebut. Proses *quality assurance* menggunakan metode *black box testing*, *user acceptance testing* dan *load testing*. Proses *quality assurance* juga menggunakan *tools* yaitu Apache Benchmark. Hasil dari proses *quality assurance* pada aplikasi *splendid* berupa desain *test case* dan laporan implementasi *testing*. Jumlah keseluruhan *test case* yang dihasilkan sebesar 62 *test case* yang mencakup 9 modul dan 2 kebutuhan non fungsional. Dari total 62 *test case*, 50 *test case* sudah memenuhi persyaratan aplikasi dan 12 *test case* masih perlu dilakukan perbaikan.

Adapun tahapan-tahapan dalam melakukan proses *quality assurance* dapat dilihat pada Gambar 2.1 dan untuk penjelasan setiap tahapan (Anwar & Kurniawan, 2019).

1. *Requirement analysis*

Requirement analysis merupakan tahapan pertama dari proses *quality assurance*. Pada tahapan ini menganalisis *software requirement* yang didapat dari *stakeholders* untuk mengetahui detail dari *software* yang akan diuji seperti fitur dan fungsi yang tersedia, ketepatan sistem dalam menjalankan fitur yang tersedia dan seberapa efisien sistem dalam membantu *user*.

2. *Test planning*

Tahapan berikutnya melakukan perencanaan berdasarkan hasil analisis yang didapat dari tahapan pertama (*Requirement analysis*). Pada tahapan ini akan menghasilkan spesifikasi pengujian yang berisi teknik pengujian, *tools* dan *environment* yang akan digunakan.

3. *Test case development*

Pada tahapan ini akan membuat *test case*, *test data*, *test script* dan *automation test* (jika diperlukan) berdasarkan fitur-fitur yang tersedia pada *software* yang akan diuji.

4. *Test Execution*

Pada tahapan ini akan melakukan pengujian berdasarkan *test plan* dan *test case* yang sudah dibuat di tahapan sebelumnya. Jika hasil dari fitur yang diuji sesuai dengan *software requirement*, maka status pada fitur tersebut adalah *pass* atau berhasil. Jika hasil dari fitur yang diuji tidak sesuai dengan *software requirement* maka status pada fitur tersebut adalah *fail* atau gagal. Pada tahapan juga mendokumentasikan hasil pengujian yang dilakukan dan mencatat *bug* pada *test case* yang gagal, kemudian *bug* tersebut akan diperbaiki dan *software* diuji lagi berdasarkan *test case* yang telah ada.

5. *Closure*

Tahapan terakhir dari *quality assurance* adalah *closure*. Pada tahapan ini akan didiskusikan hasil dari pengujian yang dilakukan dan mengevaluasi semua tahapan yang sudah dilakukan mulai dari *requirement analysis*, *test planning*, *test case development*, sampai dengan hasil dari pengujian tiap *test case*.



Gambar 2.1 Proses *quality assurance*

2.4 Test Case Management

Test case merupakan rancangan skenario tugas yang dilakukan oleh *user* untuk memverifikasi fitur dari perangkat lunak. Pengujian perangkat lunak dapat membantu pengembang aplikasi untuk menemukan dan menghapus kesalahan (*error*) maupun cacat (*defect*) sebelum perangkat lunak tersebut dirilis. Pengujian yang dilakukan pada perangkat lunak yang sudah dirilis bertujuan untuk menjaga mutu dari aplikasi tersebut. Hasil dari pengujian menggunakan *test case* akan dibandingkan dengan hasil yang telah ditetapkan pengembang aplikasi (Hasibuan & Dirgahayu, 2021). Jika terdapat perbedaan dari hasil pengujian dan hasil yang telah ditetapkan pengembang aplikasi, maka akan dilakukan perbaikan pada kode program aplikasi tersebut.

Beberapa faktor berikut juga menentukan apakah *test case* akan efektif digunakan selama pengujian. Faktor-faktor tersebut yaitu, aplikasi yang diuji memiliki fungsionalitas yang tidak lengkap, logika desain pengujian yang buruk dan kesalahpahaman spesifikasi pengujian yang dilakukan oleh penguji. *Test case* dapat dikatakan efektif apabila semakin banyak kesalahan (*error*) dan cacat (*defect*) yang ditemukan saat pengujian. Tetapi, jika *test case* hanya menemukan kesalahan (*error*) dan cacat (*defect*) dalam jumlah yang sedikit, maka *test case* tersebut belum efektif (Chernak, 2001).

Penelitian berikut mengkaji *Test case management tools* untuk aksesibilitas pengujian (Petrie, 2006). Penelitian ini menggunakan 2 alat (*tools*) yang mendukung *test case management* untuk aksesibilitas *test suites*. *Test suites* merupakan wadah yang memiliki serangkaian pengujian yang membantu penguji dalam melaksanakan dan melaporkan status pelaksanaan pengujian. *Test case* dapat ditambahkan ke beberapa *test suites* dan *test plan*. Alat

(tools) yang digunakan pada penelitian ini adalah Parsifal dan Amfortas. Parsifal merupakan aplikasi berbasis *desktop* yang memungkinkan mengubah berkas deskripsi pengujian (*test descriptive files*) dengan mudah. *Test descriptive files* menyusun lapisan (*layer*) XML yang berisi informasi deskriptif tentang *test cases* tertentu. Amfortas adalah aplikasi berbasis *website* yang memungkinkan evaluasi secara terkontrol dari *test suites* oleh pengguna. Aplikasi ini juga menyimpan hasil evaluasi dan juga mengetahui kondisi fisik dan teknis dari evaluator.

Penelitian berikut mengkaji studi kasus dalam *test management* (Parveen et al., 2007). Penelitian ini membandingkan teori-teori dari *test case management* dengan keadaan yang biasanya terjadi di industri sehubungan dengan *manual test case*. Penelitian ini menggambarkan studi kasus dari pengalaman penulis dalam menerapkan praktik akademik untuk *software test case management* dan hasil dari praktik tersebut. Penelitian ini juga menguraikan pendekatan untuk merampingkan *test case management* dan bagaimana meningkatkan produktivitas melalui *test case management*. Penelitian ini juga menjelaskan tentang teknik-teknik terkenal yang sering digunakan industri dalam proses pengujian perangkat lunak (*software testing*). Teknik-teknik yang sering digunakan yaitu, *black-box testing*, *white-box testing* dan *gray-box testing*.

Penelitian ini juga menggunakan *TestDirector* (TD). *TestDirector* merupakan aplikasi berbasis *website* yang diterbitkan oleh Mercury Interactive. Aplikasi ini dapat mengelola aspek-aspek penting dari *test management*, *requirement management* dan *defect management*. *TestDirector* merupakan *test management tools* yang bersifat global karena sudah berbasis *website*. Karena sudah berbasis *website*, *TestDirector* mendukung kolaborasi antar tim yang terdistribusi di berbagai belahan dunia atau di tempat yang berbeda dalam organisasi yang sama. Dengan *TestDirector*, penguji dapat merancang rencana pengujian (*test plan*), mengembangkan kasus uji (*test case*) dan bisa menjalankan kasus uji (*test case*) secara local maupun jarak jauh. Hasil dari penelitian ini dimaksudkan untuk mengatasi beberapa kekurangan *test case management* dalam pengaturan industri seperti variasi dalam terminologi, masalah pertukaran file, dan sentralisasi keseluruhan melalui otomatisasi.

2.5 Test Documentation

Salah satu definisi dari dokumentasi adalah 'Setiap informasi tertulis atau gambar yang menjelaskan, mendefinisikan, menentukan, melaporkan, atau mengesahkan aktivitas, persyaratan, prosedur, atau hasil' (IEEE, 1990). Dokumentasi sendiri mencakup panduan pengguna (*user manuals*), bantuan daring (*online helps*), desain dan spesifikasi fitur, komentar

kode program (*source code comment*), rencana dan laporan pengujian (*test plan and test report*), dan hal lainnya yang menjelaskan tentang cara kerja atau penggunaan sesuatu (Mamone, 2000). Dokumen harus diuji karena dalam banyak kasus, dokumentasi merupakan kesan pertama yang dimiliki pelanggan (*customer*) terhadap suatu produk. Jika dokumentasi dirancang atau ditulis dengan buruk, tidak jelas dan kurang lengkap, maka pelanggan akan tidak puas dengan produk yang dihasilkan. Dokumentasi yang baik dapat membuat suatu produk lebih mudah untuk digunakan dan juga mempengaruhi biaya perawatan produk tersebut. Dokumentasi juga dapat mempengaruhi nilai jual produk tersebut. Produk dengan dokumentasi yang lebih mudah dibaca, akan lebih diminati dibandingkan dengan produk serupa dengan dokumentasi yang buruk.

Terdapat beberapa standar yang dapat digunakan dalam pengujian perangkat lunak sebagai contoh adalah IEEE. IEEE atau *Institute of Electrical and Electronics Engineers* adalah institusi yang membuat standar dalam pengujian perangkat lunak (IEEE, 2012). IEEE sendiri memiliki banyak standar contohnya seperti IEEE 829 yang menangani proses pengujian perangkat lunak dan IEEE 1012 yang menangani pembuatan dokumen validasi dan verifikasi sistem.

IEEE 829 *Standard for Software Testing* merupakan standar untuk proses pengujian perangkat lunak yang dikeluarkan oleh organisasi internasional IEEE (Yuana, 2021). Semua tahapan untuk pengujian dan dokumentasi perangkat lunak sudah ditetapkan oleh IEEE. IEEE 829 juga menetapkan berbagai aturan dan regulasi terkait setiap langkah dalam proses pengujian perangkat lunak dan cara menulis dokumen yang benar pada setiap langkah. Terdapat beberapa tahapan dalam pengujian dan pelaporan perangkat lunak berdasarkan IEEE 829 seperti rencana pengujian, uji spesifikasi desain, spesifikasi kasus uji, spesifikasi prosedur pengujian, log uji, laporan insiden pengujian dan laporan ringkasan pengujian. Aturan dokumentasi yang berbeda untuk setiap tahapan telah disusun oleh IEEE, kesalahan yang ditemukan selama pengujian dapat memberikan penilaian sertifikasi IEEE ke dalam perangkat lunak yang diuji.

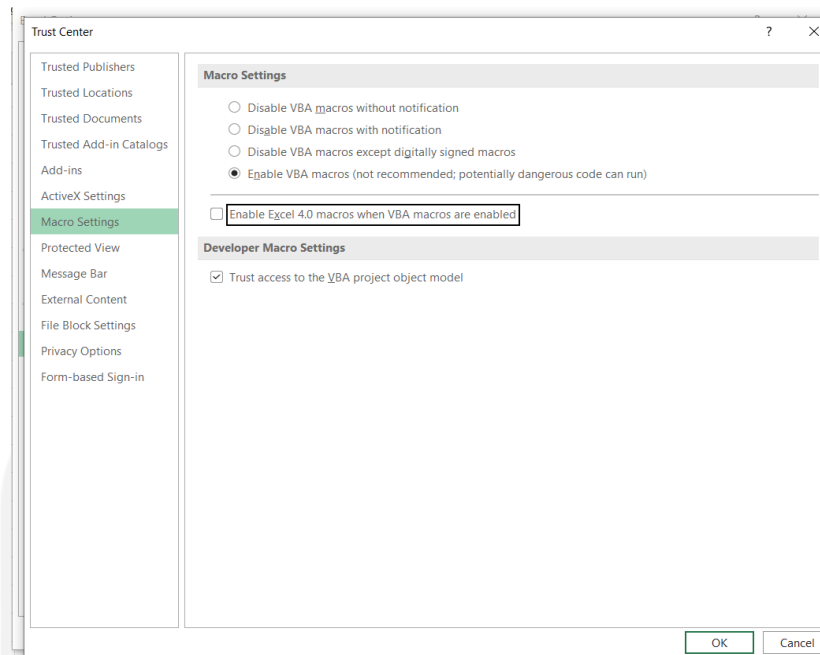
IEEE 1012 *Standard for Software Verification and Validation* merupakan standar yang digunakan sebagai dasar untuk SVV atau *Software Verification and Validation*. IEEE 1012 dikeluarkan oleh organisasi internasional bernama IEEE dan telah digunakan semenjak tanggal 20 juli 1998 (The Institute of Electrical and Electronic Engineers, 2017). IEEE 1012 membahas semua proses siklus sistem, perangkat keras dan perangkat lunak. IEEE 1012 merupakan standar yang sesuai dengan semua model *lifecycle* tetapi, tidak semua model *lifecycle*

menggunakan semua proses yang tercantum dalam IEEE 1012. Proses V&V atau *Validation & Verification* dapat menentukan apakah produk tersebut telah memenuhi tujuan penggunaan produk dan kebutuhan dari pengguna. Penentuan tersebut mencakup beberapa proses, seperti analisi, evaluasi, tinjauan, inspeksi, penilaian serta pengujian produk. IEEE 1012 memiliki tujuan seperti,

- Menetapkan kerangka kerja umum pada proses V&V, seluruh siklus sistem, perangkat lunak dan perangkat keras.
- Menentukan tugas V&V, memberikan *input* serta *output* yang diperlukan dalam proses *lifecycle*.
- Mengidentifikasi tugas V&V yang sesuai dengan skema integritas empat tingkat.
- Menentukan konten rencana untuk proses verifikasi dan validasi.

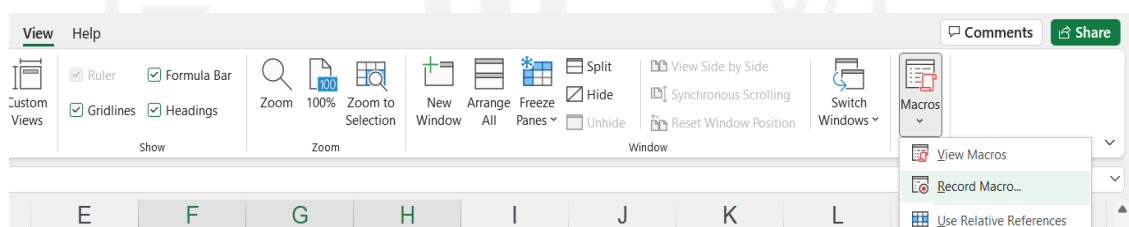
Kajian berikut membahas bagaimana cara mengotomatiskan tugas *quality assurance* secara berulang menggunakan makro Excel (Dee, 2022). Kebutuhan untuk automation saat ini merupakan tuntutan yang penting dan terus berkembang. Pada masa sekarang, semua orang mencari sesuatu yang memudahkan pekerjaannya atau kesibukan aktivitas yang terlibat dalam pekerjaan rutin. *Manual testers* biasanya memiliki pola pikir bahwa *manual testers* tidak ada hubungannya dengan hal teknis seperti *coding*. Ini merupakan pola pikir yang salah dan dapat menghentikan pertumbuhan kita sebagai *engineers*. Proses QA secara manual memiliki tugas harian yang dapat berulang mengikuti alur kerja yang sama. QA dapat mengidentifikasi tugas-tugas tersebut dengan sangat baik dan membuatnya otomatis menggunakan makro dan rumus dari Excel. Penggunaan *automation* dapat membantu dua hal seperti, mengurangi waktu yang digunakan dalam pengujian manual dan dapat menginspirasi QA untuk terus belajar, beradaptasi dengan pemikiran logis dan peningkatan keterampilan di bidang teknis. Hal-hal yang dapat diotomatisasi dalam tugas QA, seperti *test planning*, *test design*, *test execution* dan *test report*. Penulis juga memberikan *quick context setting* untuk *automation* menggunakan makro Excel. Langkah langkah untuk *quick context setting* seperti berikut:

1. Simpan berkas Excel dengan ekstensi *xlsxm*. Ekstensi tersebut merupakan ekstensi Excel yang mendukung makro.
2. Untuk mengaktifkan langkah di Excel, pilih *File* → *Opsi* → *Trust Center Setting* → *Macro Setting* → *Select Radio Button* “*Enable All Macros*” dan *check the checkbox* “*Trust access to the VBA Project object model*” → *klik oke*. Contohnya dapat dilihat pada Gambar 2.2.



Gambar 2.2 Langkah untuk mengaktifkan makro pada Excel

3. Kemudian pilih menu *View* → *Macros* → *Record Macros* → *Perform some activity and Stop Macros*. Makro dapat diubah dan dapat melihat jendela kode VBA yang direkam di jendela *editor* VBA. Kemudian dapat menambahkan kode secara eksplisit untuk membuat makro. Untuk langkah ketiga dapat dilihat pada Gambar 2.3.



Gambar 2.3 Langkah untuk *Record Macros*

4. Langkah terakhir adalah dengan melihat contoh tertentu dari tugas-tugas berulang yang bisa menjadi kandidat yang baik untuk *automation* melalui makro/rumus Excel.

2.6 Test Template

Test case merupakan pengujian *software* secara manual sehingga pengujian dilakukan pada setiap fitur atau fungsionalitas pada program yang akan diuji (Rahayu, 2020). Dalam pembuatan *test case* terdapat beberapa atribut yang dibutuhkan beserta keterangannya, yaitu:

1. *Fitur*: Merupakan nama fitur atau fungsionalitas yang akan diuji.
2. *Test Case ID*: Merupakan identitas dari fitur atau fungsionalitas yang akan diuji.
3. *Test case description*: Merupakan identitas dari fitur atau fungsionalitas yang akan diuji.
4. *Pre-condition*: Merupakan kondisi sebelum sistem dilakukan pengujian.
5. *Test steps*: Merupakan langkah-langkah yang akan dilakukan saat pengujian.
6. *Test data*: Merupakan data yang akan dimasukkan pada *test case* ini.
7. *Expected Result*: Merupakan hasil yang diharapkan dari fitur atau fungsionalitas pada program yang akan diuji.
8. *Actual Result*: Merupakan hasil sebenarnya yang terjadi pada fitur atau fungsionalitas pada program yang diuji.
9. *Status*: Merupakan status dari fitur yang diuji apakah berhasil (*pass*) atau gagal (*fail*).
Jika hasil tes tidak sesuai dengan hasil yang diharapkan (*Expected Result*) maka status pada test tersebut akan diisi gagal (*fail*), begitu juga sebaliknya.

Setelah memahami dan menyusun atribut yang dibutuhkan, langkah selanjutnya adalah membuat model pengujian manual dengan *test case* dalam bentuk tabel sesuai dengan atribut yang telah ditetapkan sebelumnya. Model *test case* dalam bentuk tabel dapat dilihat pada Tabel 2.1 (Hasibuan & Dirgahayu, 2021).

Tabel 2.1 Model *test case* (Hasibuan & Dirgahayu, 2021)

Fitur	<i>Test Case ID</i>	<i>Test Case Description</i>	<i>Pre-Condition</i>	<i>Test Steps</i>	<i>Expected Result</i>	<i>Actual Result</i>	<i>Status</i>

Kajian berikut membahas penggunaan *test case template* sederhana untuk pengujian perangkat lunak menggunakan aplikasi Microsoft Excel (Neumeyer, 2022). Dalam beberapa skenario, *template test case* dapat digunakan untuk *software development*, *IT projects*, *ERP rollouts / migrations*, dan *manufacturing*. Terdapat beberapa atribut yang selalu digunakan dalam *template* seperti:

1. *Test case*
2. *Steps*

3. *Steps description*

4. *Status*

5. *Expected Result*

6. *Actual Result*

Selain atribut yang sudah disebutkan di atas, dapat disertakan beberapa atribut tambahan yang berguna seperti:

1. *Process*

2. *Comment*

3. *Company*

4. *Department*

Jika pengujian dilakukan oleh perusahaan atau departemen lain, dapat menambahkan beberapa atribut tambahan yang sudah disebutkan dan dapat menambahkan beberapa atribut lainnya yang dibutuhkan. Atribut *comment* dapat membantu memberikan informasi tambahan terutama ketika *test steps* gagal. *Test cases* harus selalu ditentukan oleh pelanggan karena hanya pelanggan yang dapat mengkonfirmasi apakah suatu produk memenuhi persyaratannya atau tidak. Meminta seseorang yang tidak familiar dengan proses untuk menentukan tes, dapat menyebabkan konsekuensi yang tidak diinginkan seperti *test case* yang salah atau tidak lengkap. Hasil dari konsekuensi tersebut adalah pengujian yang dilakukan tidak akan akurat dan harus menerima resiko proyek dari masalah yang tidak terdeteksi. Karena itu, harus tetap memberikan bantuan kepada pelanggan. Beberapa orang mengalami kesulitan memahami konsep pengujian perangkat lunak sistematis dan kesulitan dengan penggunaan *spreadsheet* yang benar. Penulis juga memberikan *screenshot* dari *template* yang digunakan dan dapat dilihat pada Gambar 2.4.

PROCESS	NO.	TEST CASE	STEP	DESCRIPTION	STATUS	EXPECTED RESULT	ACTUAL RESULT	Comment
Logistics	1	Delivery	1	Create delivery	passed	delivery is created without errors		
			2	Print delivery form	failed	form is printed correctly	The logo is missing on the form	
			3	Print shipping label	in work	label is printed correctly		
			4	Notify freight forwarder	open	freight forwarder informed		
			5	Freight forwarder picks up delivery	open	delivery picked up successfully		
Accounting	2	Billing	1	Select customer delivery	failed	delivery can be selected	unable to find delivery	
			2	Print invoice	in work	invoice is printed correctly		
			3	Send invoice to customer	open	invoice sent successfully		
			4	Create open item in AR	open	new open item is created without problems		
Sales	3	Customer master data	1	Create new customer	passed	new customer is created		
			2	Change customer data	passed	customer changed		
			3	Block customer	failed	customer blocked	I'm getting a 'missing authorization' error	
			4	Delete customer	in work	customer deleted		

Gambar 2.4 Contoh *test case template* (Neumeyer, 2022).

Adapun beberapa *test scenario* yang dapat digunakan seperti (Ihsan, 2022):

1. *Test scenario* halaman *login*

Pada halaman *login* akan memverifikasi apakah akan muncul pesan *error* jika *user* memasukan *username* atau *password* yang salah, verifikasi apakah *user* dapat melihat *passwordnya* jika menekan *icon* mata pada *field password* dan lain sebagainya.

2. *Test scenario* halaman *register*

Pada halaman ini akan memverifikasi apakah *password* yang diinputkan sudah terenkripsi atau tidak, verifikasi apakah data yang diinputkan pada alamat *email* sudah sesuai format *email* atau tidak dan lain sebagainya.

3. *Test scenario* halaman produk

Pada halaman ini akan memverifikasi apakah gambar produk yang ditampilkan pada halaman *website* pecah (*broken*) / buram (*blurred*), verifikasi informasi pengiriman detail produk ditampilkan atau tidak, verifikasi angka ketersediaan produk ditampilkan atau tidak, verifikasi bahwa produk dikategorisasikan atau tidak dan lain sebagainya.

4. *Test scenario* *upload file* / foto

Pada bagian ini akan memverifikasi apakah tombol untuk *upload file* / foto dapat diklik, memverifikasi apakah *user* dapat memilih dan mengunggah *file* / foto lebih dari 1, memverifikasi apakah *user* hanya dapat mengunggah *file* dengan format *extension* yang diizinkan saja, memverifikasi apakah nama *file* sebelum diunggah sama dengan nama *file* setelah diunggah, memverifikasi apakah muncul pesan *error* ketika *file* yang diunggah melebihi ukuran yang ditentukan dan memverifikasi apakah fitur *drag & drop* berfungsi ketika digunakan

5. *Test scenario* halaman lupa *password*

Pada halaman ini akan memverifikasi bahwa terdapat *forget password link* pada halaman *login*, verifikasi bahwa terdapat *field email* pada halaman *forget password*, memverifikasi bahwa *user* akan mendapatkan *email* setelah memasukkan alamat *email* yang *valid* pada halaman *forget password*, verifikasi bahwa pesan kesalahan harus ditampilkan saat memasukkan spasi di *field* dan tombol kirim diklik dan verifikasi bahwa *user* dapat login menggunakan *password* yang baru, bukan *password* yang lama.

6. *Test scenario mobile number field*

Pada bagian ini akan memverifikasi ketika format untuk nomor ponsel ketika ditambahkan nomor ponsel pada *field*, verifikasi bahwa *field* hanya menerima *input numeric value*, verifikasi bahwa muncul pesan ‘format tidak sesuai’ ketika memasukkan selain *numeric value* ke *field*, verifikasi dengan memasukkan nomor yang lebih sedikit dari nomor ponsel yang sebenarnya (11 hingga 12 digit nomor) dan verifikasi nomor ponsel yang benar tersimpan pada database atau tidak ketika tombol *save* diklik.

7. *Test scenario* untuk fungsi pencarian

Pada bagian ini akan memverifikasi bahwa terdapat *icon* pencarian pada *field*, verifikasi bahwa pencarian berfungsi dan memberikan hasil yang benar untuk kata kunci yang benar atau tidak dengan menambahkan pencarian yang valid, verifikasi bahwa pencarian berfungsi dengan memasukan kata kunci dan memencet tombol pencarian, verifikasi bahwa user dapat *paste* kata kunci menggunakan *mouse*, verifikasi pesan kesalahan harus ditampilkan untuk *input* kosong, verifikasi *pagination* ditambahkan jika hasil pencarian sesuai dengan jumlah halaman, verifikasi apakah *pagination* dapat diakses atau tidak dengan mengklik *next*, *previous*, dan *number*, verifikasi apakah saran ditampilkan atau tidak saat menambahkan kata kunci ke pencarian dan memverifikasi apakah kata kunci terkait ditambahkan ke hasil pencarian atau tidak.

8. *Test scenario* untuk *email field*

Pada bagian ini akan memverifikasi bahwa *field* alamat *email* dapat diakses ketika diklik, verifikasi bahwa *user* dapat mengetik alamat *email* dalam *email field*, validasi *email field* dengan memasukan alamat email yang benar dan verifikasi *email* yang berisi karakter khusus (seperti +, -, _ dan lain sebagainya) yang dianggap valid.

9. *Cart button test case*

Pada bagian ini akan memastikan bahwa *cart button* tersedia pada halaman *website* untuk produk, memastikan *user* dapat memilih produk untuk ditambahkan ke halaman keranjang, memastikan apakah *user* dapat menambahkan lebih dari 1 item, verifikasi jumlah produk yang ditambahkan terlihat jelas pada halaman keranjang atau tidak, verifikasi total harga produk terlihat pada halaman *website* atau tidak, verifikasi bahwa *user* dapat menghapus produk dari keranjang atau tidak dan

verifikasi dengan menambahkan atau menghapus produk yang sama, apakah harga total terbaru produk diperbarui pada halaman keranjang atau tidak.

10. *Test case* untuk pembayaran

Pada bagian ini akan memverifikasi apakah terdapat tombol *checkout* atau pembayaran pada halaman keranjang, memastikan apakah halaman pembayaran menggunakan berbagai metode pembayaran tersedia atau tidak untuk *user*, memastikan pembayaran telah berhasil dilakukan dengan menambahkan kartu *credit* / *debit* yang valid, verifikasi apakah alamat pengiriman yang benar ditampilkan atau tidak, verifikasi apakah *user* mendapatkan *email* notifikasi atau tidak dan verifikasi apakah *email* yang dikirim terdapat nomor pemesanan atau tidak.

2.7 Literature Review

Pada penelitian ini, mengkaji beberapa referensi yang digunakan sebagai dasar teori dari proses *quality assurance* yang akan dilakukan. Referensi yang digunakan dapat dilihat pada Tabel 2.2.

Tabel 2.2 *Literature review*

Penulis	Metode yang digunakan / cara yang dilakukan	Hasil
(Novianingsih, 2022)	metode <i>black box testing</i> berbasis <i>equivalence partitioning</i> dan <i>boundary value analysis</i> .	kesimpulan penyusunan dokumen pengujian pada saat melakukan perancangan <i>test case</i> dengan menetapkan nilai input dari pengujian, sehingga terdapat batasan-batasan input serta output pada pengujian yang dilakukan.
(Yunisa & Putro, 2019)	Metode <i>white box testing</i> dengan teknik pengujian <i>statement coverage</i> dan <i>branch coverage</i>	Hasil yang didapat berupa identifikasi pengujian yang dilakukan memperoleh 7 tahapan dalam melakukan pengujian.
(Hariyanto et al., 2020)	Penelitian ini menerapkan SQA (<i>Software Quality Assurance</i>) dengan metode yang digunakan	Hasil dari penelitian ini menunjukkan bahwa perusahaan skala kecil dan

	<p>dalam penelitian ini adalah kualitatif yang meliputi wawancara dengan berbagai pihak penting dan juga studi dokumen terkait SQA.</p>	<p>menengah telah menerapkan SQA dan tiap perusahaan mempunyai standar tertentu untuk metode pengembangan perangkat lunak. Kualitas hasil akhir dari pengembangan perangkat lunak bergantung pada kapasitas sumber daya alam setiap perusahaan.</p>
(Jamali, 2020)	<p>Proses <i>quality assurance</i> menggunakan metode <i>black box testing</i>, <i>user acceptance testing</i> dan <i>load testing</i> dan <i>tools</i> yaitu Apache Benchmark.</p>	<p>Hasil dari proses <i>quality assurance</i> pada aplikasi splendid berupa desain <i>test case</i> dan laporan implementasi <i>testing</i>. Jumlah keseluruhan <i>test case</i> yang dihasilkan sebesar 62 <i>test case</i> yang mencakup 9 modul dan 2 kebutuhan non fungsional. Dari total 62 <i>test case</i>, 50 <i>test case</i> sudah memenuhi persyaratan aplikasi dan 12 <i>test case</i> masih perlu dilakukan perbaikan.</p>
(Petrie, 2006)	<p>Penelitian ini menggunakan 2 alat (<i>tools</i>) yang mendukung <i>test case management</i> untuk aksesibilitas <i>test suites</i>. Alat (<i>tools</i>) yang digunakan pada penelitian ini adalah Parsifal dan Amfortas.</p>	<p>Parsifal merupakan aplikasi berbasis <i>desktop</i> yang memungkinkan mengubah berkas deskripsi pengujian (<i>test descriptive files</i>) dengan mudah. <i>Test descriptive files</i> menyusun lapisan (<i>layer</i>) XML yang berisi informasi deskriptif tentang <i>test cases</i> tertentu. Amfortas adalah aplikasi berbasis <i>website</i> yang memungkinkan evaluasi secara terkontrol dari <i>test suites</i> oleh</p>

		<p>pengguna. Aplikasi ini juga menyimpan hasil evaluasi dan juga mengetahui kondisi fisik dan teknis dari evaluator.</p>
(Parveen et al., 2007)	<p>Penelitian ini membandingkan teori-teori dari <i>test case management</i> dengan keadaan yang biasanya terjadi di industri sehubungan dengan <i>manual test case</i>, menggambarkan studi kasus dari pengalaman penulis dalam menerapkan praktik akademik untuk <i>software test case management</i> dan hasil dari praktik tersebut.</p>	<p>Penelitian ini juga menguraikan pendekatan untuk merampingkan <i>test case management</i> dan bagaimana meningkatkan produktivitas melalui <i>test case management</i>. Penelitian ini juga menjelaskan tentang teknik-teknik terkenal yang sering digunakan industri dalam proses pengujian perangkat lunak (<i>software testing</i>). Teknik-teknik yang sering digunakan yaitu, <i>black-box testing</i>, <i>white-box testing</i> dan <i>gray-box testing</i>.</p>
(Yuana, 2021)	<p>Penelitian dilakukan untuk implementasi dokumen dengan standar IEEE-829 dan IEEE-1012. Proses implementasi dokumen pengujian dengan standar IEEE-829 dilakukan dari pengantar pengujian, pengujian menggunakan <i>blackbox</i> dan <i>whitebox</i> serta hasil uji.</p> <p>IEEE-1012 dilakukan dari penggambaran dokumentasi, proses verifikasi dan validasi SDLC hingga persyaratan dalam pembuatan dokumentasi</p>	<p>Standar IEEE-829 lebih populer, lebih banyak referensi yang bisa digunakan dengan kompleksitas yang lebih rendah sehingga dapat digunakan dalam pengembangan sistem skala kecil</p>

(Dee, 2022)	Proses QA secara manual memiliki tugas harian yang dapat berulang mengikuti alur kerja yang sama. QA dapat mengidentifikasi tugas-tugas tersebut dengan sangat baik dan membuatnya otomatis menggunakan makro dan rumus dari Excel.	Penulis memberikan <i>quick context setting</i> untuk <i>automation</i> menggunakan makro Excel.
(Neumeyer, 2022)	Penggunaan <i>test case template</i> sederhana untuk pengujian perangkat lunak menggunakan aplikasi Microsoft Excel	Atribut-atribut yang digunakan dalam <i>test case template</i> serta contoh <i>test case template</i> menggunakan aplikasi Microsoft Excel
(Ihsan, 2022)		<i>Test case</i> dan <i>test scenario</i> yang dapat digunakan dalam pengujian perangkat lunak berbasis <i>website</i> .

BAB III

ANALISIS DAN PERENCANAAN

Proses *quality assurance* pada penelitian ini terdiri dari beberapa tahapan. Pada bab ini akan membahas tahapan pertama dan kedua. Tahapan yang pertama yaitu mengumpulkan dan menganalisis *software requirements* terkait aplikasi berbasis *website Ivent* yang didapat dari *stakeholders*. Tahapan kedua yaitu melakukan perencanaan berdasarkan *software requirements* yang didapat dari tahapan pertama. Perencanaan yang dilakukan berupa mempersiapkan teknik pengujian yang akan dilakukan, mempersiapkan alat yang akan digunakan, mempersiapkan aplikasi berbasis *website Ivent* sebagai objek pengujian dan mempersiapkan *hardware* dan *software* yang akan digunakan selama pengujian.

3.1 Requirement Analysis

Ivent merupakan aplikasi *e-commerce* berbasis *website* yang dikembangkan untuk mendukung pergerakan bisnis di sektor industri acara dengan cara menghubungkan antara konsumen dengan para pemilik usaha *event organizer* dan vendor sehingga lebih memudahkan konsumen untuk menentukan pilihan mereka karena *event organizer* dan vendor dikumpulkan dalam satu *platform* (Mucjal, 2021). *Ivent* sendiri dikembangkan oleh 3 orang dengan peran yang berbeda-beda dalam tim. Dwiko Nugroho Dani sebagai *huslter* memiliki peran dalam membuat rancangan bisnis data yang sesuai dengan spesifikasi kebutuhan pengguna (Dani, 2021). Muhammad Arif Idris sebagai *hipster* memiliki peran dimulai dari proses perancangan *wireframe* hingga pembuatan antarmuka dari *Ivent* yang dikembangkan menggunakan Figma (Idris, 2021). Aikal Alfrian Mucjal sebagai *hacker* memiliki peran dalam proses implementasi dan pengembangan perangkat lunak menggunakan Bahasa pemrograman PHP (*Hypertext Preprocessor*) dan *framework* Laravel. Aplikasi *e-commerce* berbasis *website Ivent* dikembangkan menggunakan metode *waterfall* karena metode tersebut mampu meminimalisir kesalahan pada pengembangan perangkat lunak karena model alur pengembangannya linear serta struktur atau rangkaiannya yang jelas.

3.1.1 Analisis Proses Bisnis

Banyak sektor bisnis yang telah berkembang menjadi digital dengan memanfaatkan penerapan dari teknologi informasi. *Event organizer* atau penyelenggara acara merupakan salah satu sektor di industri acara yang telah berkembang dengan pesat. Perkembangan *event*

organizer di Indonesia yang berkembang dengan pesat, membuat kebutuhan masyarakat pada pembuatan acara semakin bertambah kuat dan membutuhkan peran *event organizer* untuk mengorganisir acara yang hendak dibuat. Vendor yang berperan sebagai penyedia peralatan dan pendukung acara juga semakin diminati berkat besarnya kebutuhan masyarakat pada pembuatan acara. Karena ketergantungan masyarakat yang semakin besar, bisnis *event organizer* dan vendor terus berkembang.

Sektor industri acara merupakan peluang bisnis yang besar. Beberapa *startup* di Indonesia sudah mengembangkan *e-commerce* di sektor acara ini. Contoh *startup* yang mengembangkan *e-commerce* seperti Akusewa.com yang berfokus pada penyedia vendor dan Bridestory.com yang berfokus pada *wedding organizer*. *E-commerce* pada sektor ini sudah cukup banyak, namun yang berfokus pada *event organizer* dan vendor masih sedikit jumlahnya. Beberapa penyedia *event organizer* masih memasarkan jasa mereka menggunakan media sosial. Hal tersebut masih menyulitkan konsumen untuk mencari dan membandingkan setiap *event organizer* dan vendor. Kesulitan utama konsumen ketika harus mencari setiap akun media sosial dari *event organizer* dan vendor untuk membandingkan sesuai yang diminati konsumen. Solusi yang ditawarkan tim pengembang aplikasi Ivent adalah mengembangkan aplikasi *e-commerce* pada sektor acara dengan cara menghubungkan antara konsumen dengan pemilik usaha *event organizer* dan vendor sehingga lebih memudahkan konsumen untuk menentukan pilihan mereka karena berbagai *event organizer* dan vendor berada pada satu *platform*. Pengembangan aplikasi Ivent diharapkan dapat membuat peluang baru dan membantu pemilik usaha di sektor ini untuk menjangkau konsumen secara luas dalam penyewaan barang dan juga jasa, serta membantu konsumen untuk memilih sendiri berbagai *event organizer* dan vendor yang sesuai dengan keinginan dan kebutuhan konsumen.

3.1.2 Analisis Kebutuhan Fungsional

Kebutuhan fungsional adalah jenis kebutuhan yang berisi tentang proses-proses yang akan dilakukan oleh sistem. Kebutuhan fungsional mempunyai beberapa syarat yaitu aktivitas-aktivitas yang harus dilakukan dalam sistem, berdasarkan prosedur dan fungsi-fungsi bisnis, serta didokumentasikan dalam model. Fitur-fitur yang terdapat pada aplikasi Ivent seperti (Mucjal, 2021):

1. Fitur *Register account*, dapat digunakan oleh konsumen maupun penjual untuk mendaftarkan diri ke sistem.

2. Fitur *Login account*, digunakan oleh *user* untuk masuk ke sistem menggunakan data-data yang sudah didaftarkan sebelumnya.
3. Fitur *Lupa password*, digunakan oleh *user* ketika lupa *password*. Fitur ini dapat mengirimkan *link* kepada *email user* yang bersangkutan untuk dapat melakukan *reset password*, sehingga *user* dapat melakukan proses *login* dengan menggunakan *password* yang baru.
4. Fitur Menu pada *homepage*, dapat digunakan pengguna untuk berpindah dari *homepage* ke halaman yang lain. Menu yang dapat dipilih *user* seperti sewa, buka toko, pesanan (*cart*) dan profil akun.
5. Fitur Pencarian, fitur ini terdapat pada *homepage* aplikasi Ivent. Dapat digunakan untuk mencari EO dan vendor menggunakan kata kunci yang mendekati nama EO dan vendor yang dicari.
6. Fitur Tambahkan produk, Di dalam fitur ini terdapat beberapa *field* kosong yang akan digunakan sebagai input untuk produk baru. *Field* pada fitur ini seperti nama produk, detail produk, gambar produk, harga, kategori produk (EO/vendor), stok produk dan lokasi.
7. Fitur *Edit* produk, pada fitur ini dapat mengubah informasi detail produk dengan mengubah isi pada *field* yang sesuai.
8. Fitur Melihat produk, fitur ada dapat digunakan *user* dengan cara klik pada produk yang ingin dilihat. Kemudian sistem akan menampilkan informasi detail mengenai produk yang diklik.
9. Fitur Memesan produk, fitur ini digunakan *user* untuk memesan produk dengan cara klik produk yang ingin dipesan, lalu klik tombol 'tambahkan pesanan'.
10. Fitur Melihat pesanan, fitur ini dapat digunakan dengan cara klik mana pesanan pada *homepage* aplikasi Ivent. Kemudian akan berpindah ke halaman pesanan yang berisi daftar produk yang sudah ditambahkan ke pesanan.
11. Fitur Menambahkan alamat, digunakan untuk menambahkan alamat *user* ketika melakukan *checkout* suatu produk. *User* dapat menambahkan informasi alamat pengiriman seperti nama pembeli, alamat pengiriman secara detail dan nomor telepon.
12. Fitur Menghapus pesnana, *user* dapat menghapus pesanan pada halaman pesanan.
13. Fitur Melihat transaksi, *user* dapat melihat transaksi pada halaman status transaksi. Informasi yang tertera pada halaman tersebut seperti total harga yang harus dibayar, status pembayaran, status pengiriman dan detail barang yang dipesan.

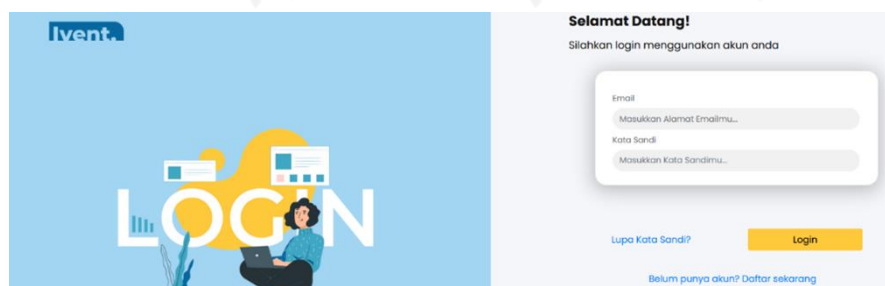
14. Fitur Mengunggah bukti pembayaran, *user* dapat mengunggah bukti pembayaran pada halaman pembayaran, lalu klik tombol '*choose file*', pilih gambar bukti pembayaran yang mau diunggah lalu klik tombol 'kirim'.
15. Fitur Mengecek pesanan untuk penjual, terdapat pada halaman buka toko, lalu pilih menu transaksi. Informasi yang ditampilkan seperti nama pembeli, harga satuan produk, harga total produk, status pembayaran, status pengiriman dan bukti pembayaran.
16. Fitur Verifikasi bukti pembayaran oleh admin, untuk fitur ini berada pada halaman *dashboard* admin. Admin dapat memverifikasi bukti pembayaran yang diunggah oleh *user* dengan cara klik tombol 'detail' pada halaman *dashboard* admin.

3.1.3 Tampilan Antarmuka Aplikasi Ivent

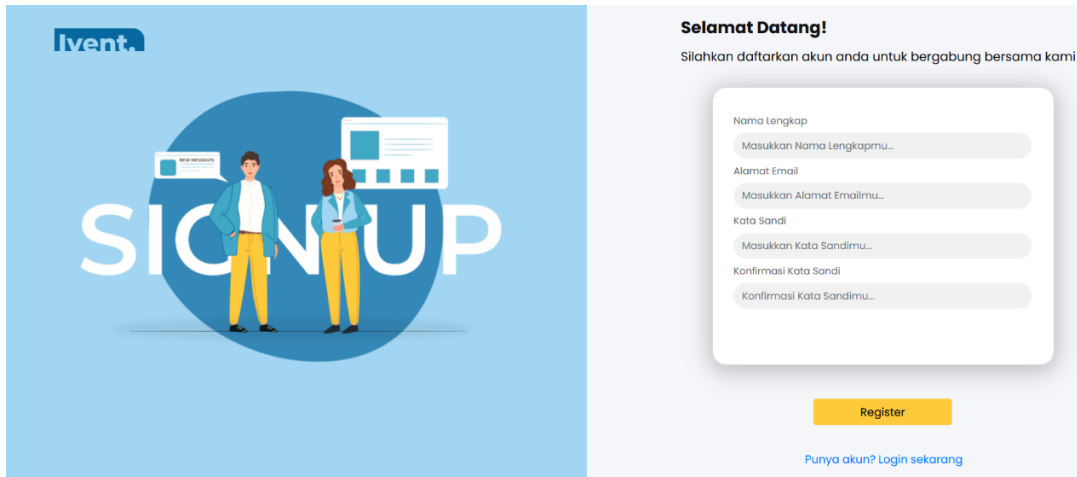
Proses implementasi pengembangan aplikasi Ivent dilakukan oleh pengembang menggunakan Bahasa pemrograman PHP dan *framework* Laravel. Beberapa tampilan antarmuka yang ditangkap aplikasi Ivent diperlihatkan sebagai berikut:

1. Halaman *Login & Register*

User dapat melakukan transaksi pada *website* Ivent dengan cara melakukan *login* terlebih dahulu. Halaman ini dapat diakses oleh konsumen, penjual (EO dan vendor) dan admin. Pada halaman ini *user* harus memasukkan *email* dan kata sandi yang sudah terdaftar pada sistem. Pada halaman ini *user* juga terdapat fitur untuk lupa *password* dan fitur untuk *register* bagi *user* baru pada *website* Ivent. Untuk tampilan halaman *login* dapat dilihat pada Gambar 3.1 dan halaman *register* terdapat beberapa *field* yang harus diisi *user* untuk mendaftarkan akunnya. *Field* yang harus diisi seperti nama lengkap, alamat *email*, kata sandi, konfirmasi kata sandi. *Hypertext* untuk *login* juga tersedia pada halaman ini. Tampilan pada halaman *register* dapat dilihat pada Gambar 3.2.



Gambar 3.1 Tampilan halaman *login* Ivent



Gambar 3.2 Tampilan halaman *register* Ivent

2. Halaman *Homepage* Ivent

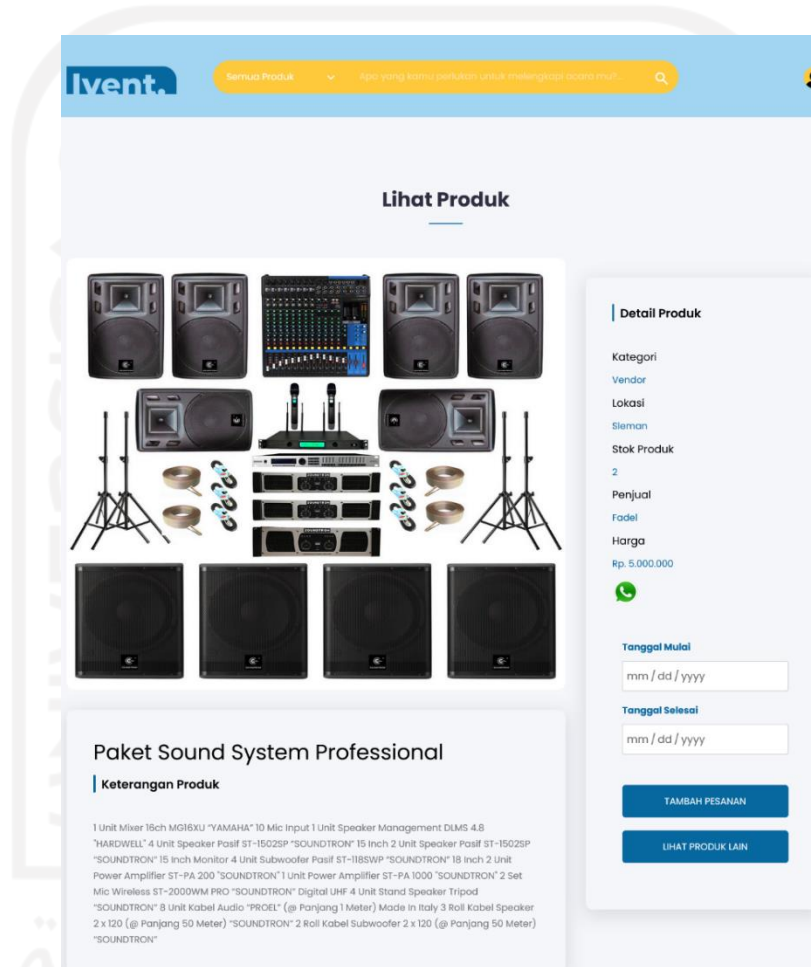
Homepage Ivent terdapat 4 menu yaitu sewa, buka toko, pesanan dan juga profil *user*. Pada *homepage* terdapat fitur pencarian yang dapat digunakan untuk mencari keperluan *user* menggunakan kata kunci yang dimasukkan. Pada *homepage* juga terdapat EO atau vendor yang dapat dipilih oleh *user*. Tampilan *homepage* dapat dilihat pada Gambar 3.3.



Gambar 3.3 *Homepage* Ivent

3. Halaman Detail Produk

Pada halaman ini ditampilkan detail produk seperti gambar produk, keterangan produk, kategori produk, lokasi produk, stok yang tersedia, nama penjual produk, nomor penjual produk, serta pada halaman ini user dapat menambahkan pesanan dengan cara memilih tanggal mulai dan tanggal selesai untuk sewa kemudian klik tombol pesanan. Untuk tampilan halaman detail produk dapat dilihat pada Gambar 3.4.



Gambar 3.4 Tampilan halaman detail produk

3.2 Test Planning

Berdasarkan hasil analisis untuk *software requirements* yang didapat, setelah itu akan dilakukan perencanaan untuk pengujian pada aplikasi berbasis *website* Ivent. Perencanaan pengujian yang dilakukan seperti berikut:

1. Peneliti akan menggunakan laptop pribadi sebagai alat pengujian.
2. Aplikasi berbasis *website* Ivent sebagai objek yang diuji.
3. Penguji menentukan fitur-fitur apa saja yang akan diuji.

4. Pengujian dilakukan secara manual / *manual testing*.
5. Pengujian dilakukan berdasarkan *test case* yang dibuat.

Cara pembuatan *Test Case*:

1. Mempelajari aplikasi Ivent.
2. Memilih atribut-atribut pada *template test case* yang sudah dikaji pada penelitian ini sesuai kebutuhan.
3. Memastikan penggunaan bahasa dan langkah-langkah pengujian yang harus jelas.
4. Mendokumentasikan setiap *test case* yang dibuat dan hasil pengujian setiap *test case*.
5. Memastikan konsisten dalam penamaan *test case*.
6. Membuat *test case* berdasarkan *software requirement* sistem aplikasi Ivent.
7. Membuat *test suites* untuk setiap fitur.
8. Membuat minimal 2 *test case* untuk setiap *test suites* (kasus normal & alternatif).

Adapun fitur-fitur pada aplikasi berbasis *website Ivent* yang akan diuji terdapat pada Tabel 3.1.

Tabel 3.1 Daftar fitur-fitur aplikasi berbasis *website Ivent* yang akan diuji

No	Fitur-fitur yang akan diuji
1	Fitur <i>Register account user</i>
2	Fitur <i>Login account user</i>
3	Fitur Lupa <i>password</i> untuk user
4	Fitur Menu pada <i>homepage</i>
5	Fitur Pencarian pada <i>homepage</i>
6	Fitur Tambahkan produk toko untuk penjual
7	Fitur <i>Edit</i> produk toko untuk penjual
8	Fitur Melihat produk untuk <i>user</i>
9	Fitur Memesan produk untuk <i>user</i>
10	Fitur Melihat pesanan untuk <i>user</i>
11	Fitur Menambahkan alamat untuk <i>user</i>
12	Fitur Menghapus pesanan untuk <i>user</i>
13	Fitur Melihat transaksi untuk <i>user</i> (penjual)
14	Fitur Mengunggah bukti pembayaran untuk <i>user</i>
15	Fitur Mengecek pesanan untuk penjual
16	Fitur Verifikasi bukti pembayaran untuk admin

Template yang akan digunakan seperti pada Tabel 2.1 dengan menambahkan 1 kolom atribut. Atribut -atribut pada *template* yang akan digunakan adalah fitur, *test case id*, *test case description*, *pre-condition*, *test steps*, *test data*, *expected result*, *actual result* dan status.

BAB IV

PENGEMBANGAN DAN EKSEKUSI

Pada bab ini akan membahas tahapan ketiga sampai kelima dalam proses *quality assurance*. Tahapan ketiga yaitu pembuatan dan pengelompokkan desain *test case* berdasarkan fitur-fitur yang akan diuji pada aplikasi berbasis *website Ivent*. Tahapan yang keempat yaitu tahapan pengujian. Pada tahapan ini akan dilakukan pengujian *software* berbasis *website Ivent* menggunakan *test case* yang dibuat pada tahapan sebelumnya serta melakukan pencatatan proses dan hasil yang dilakukan selama pengujian. Tahapan yang terakhir yaitu melakukan penyusunan dokumentasi pengujian yang berisi kesimpulan dan kelayakan *software* berbasis *website Ivent* yang diuji.

4.1 Test Case Development

Test case development merupakan tahapan ketiga dari proses *quality assurance* membuat *test case* yang akan digunakan saat *pengujian software* berbasis *website Ivent*. Setelah *test case* dibuat, kemudian akan dibuat dalam tabel dan dikelompokkan berdasarkan fitur-fitur yang terdapat pada aplikasi tersebut. *Template* yang dijadikan referensi seperti pada Tabel 2.1, kemudian dimodifikasi dengan menambahkan kolom atribut *test data* karena terdapat *data* yang harus diinputkan pada pengujian yang akan dilakukan. Setelah menambahkan kolom atribut *test data*, *template* yang akan digunakan seperti pada Gambar 4.1. *Template* seperti pada Gambar 4.1 juga dapat digunakan untuk pengujian pada aplikasi serupa dengan memodifikasi dan menyesuaikan atribut-atribut yang digunakan.

Template test case yang digunakan terdiri dari 9 kolom yaitu Fitur, *Test Case ID*, *Test Case Description*, *Pre-Condition*, *Test Steps*, *Test Data*, *Expected Result*, *Actual Result* dan Status. Penjelasan untuk isi pada setiap kolom seperti:

1. Kolom Fitur berisi fitur-fitur pada aplikasi Ivent yang akan diuji.
2. Kolom *Test Case ID* berisi urutan identitas *test case* yang digunakan untuk mempermudah pengujian. Identitas test case diberi angka sesuai dengan urutan fitur yang diuji.
3. Kolom *Test Case Description* berisi deskripsi atau fungsionalitas yang akan diuji pada aplikasi Ivent.
4. Kolom *Pre-Condition* berisi kondisi sistem sebelum *test case* dilakukan pengujian.

Tabel 4.1 *Test Suites* Halaman *Register*

Fitur	Test Case ID	Test Case Description	Pre-Condition	Test Steps	Test Data	Expected Result
1.Register	TC1.1	Salah satu kolom ada yang kosong	Sudah berada di halaman <i>Register user</i>	<ol style="list-style-type: none"> 1. Klik <i>field email</i> 2. Masukkan <i>email</i> dengan format yang sesuai 3. Klik <i>field password</i> 4. Masukkan <i>password</i> yang ingin digunakan 5. klik <i>field konfirmasi password</i> 6. Masukkan <i>password</i> yang sama 7. klik tombol <i>register</i> 	<i>Email:</i> cobamasuk1@gmail.com <i>Password:</i> 12345678 <i>Konfirmasi password:</i> 12345678	Tetap berada di halaman <i>register</i> dan muncul <i>alert</i> karena salah satu <i>field</i> belum terisi
	TC1.2	Memasukkan konfirmasi <i>password</i> yang tidak sesuai	Sudah berada di halaman <i>Register user</i>	<ol style="list-style-type: none"> 1. Klik <i>field</i> nama lengkap 2. Masukkan nama lengkap dengan format yang sesuai 3. Klik <i>field email</i> 4. Masukkan <i>email</i> dengan format yang sesuai 5. Klik <i>field password</i> 6. Masukkan <i>password</i> yang ingin digunakan 7. klik <i>field konfirmasi password</i> 8. Masukkan <i>password</i> yang berbeda 9. klik tombol <i>register</i> 	Nama lengkap: Tester QA <i>Email:</i> cobamasuk1@gmail.com <i>Password:</i> 12345678 <i>Konfirmasi password:</i> 123456789	Tetap berada di halaman <i>register</i> dan muncul <i>alert</i> karena <i>password</i> dan konfirmasi <i>password</i> dimasukkan tidak sama
	TC1.3	Memasukkan email yang sudah pernah terdaftar pada sistem	Sudah berada di halaman <i>Register user</i>	<ol style="list-style-type: none"> 1. Klik <i>field</i> nama lengkap 2. Masukkan nama lengkap dengan format yang sesuai 3. Klik <i>field email</i> 4. Masukkan <i>email</i> dengan format yang sesuai 5. Klik <i>field password</i> 6. Masukkan <i>password</i> yang ingin digunakan 7. klik <i>field konfirmasi password</i> 8. Masukkan <i>password</i> yang berbeda 9. klik tombol <i>register</i> 	Nama lengkap: Tester QA <i>Email:</i> cobamasuk@gmail.com <i>Password:</i> 12345678 <i>Konfirmasi password:</i> 12345678	Tetap berada di halaman <i>register</i> dan muncul <i>alert</i> karena <i>email</i> yang dimasukkan sudah pernah terdaftar pada sistem

	TC1.4	Mendaftarkan diri sebagai <i>user</i> di sistem	Sudah berada di halaman <i>Register user</i>	<ol style="list-style-type: none"> 1.Klik <i>field</i> nama lengkap 2.Masukkan nama lengkap dengan format yang sesuai 3. Klik <i>field email</i> 4.Masukkan <i>email</i> dengan format yang sesuai 5.Klik <i>field password</i> 6.Masukkan <i>password</i> yang ingin digunakan 7.klik <i>field</i> konfirmasi <i>password</i> 8.Masukkan <i>password</i> yang sama 9.klik tombol <i>register</i> 	<p>Nama lengkap: Tester QA</p> <p>Email: cobamasuk1@gmail.com</p> <p>Password: 12345678</p>	Berhasil mendaftar sebagai <i>user</i> dan berpindah ke <i>homepage</i> Ivent
--	-------	---	--	--	---	---

Tabel 4.2 Test Suites Halaman Login

Fitur	Test Case ID	Test Case Description	Pre-Condition	Test Steps	Test Data	Expected Result
2.Login	TC2.1	Memasukkan <i>email</i> & <i>password</i> pengguna yang sudah terdaftar	Sudah berada di halaman <i>login user</i>	<ol style="list-style-type: none"> 1.Klik <i>field email</i> 2.Masukkan <i>email</i> yang sudah terdaftar 3.Klik <i>field password</i> 4.Masukkan <i>password</i> yang sesuai 5.Klik tombol masuk 	<p>Email: user@Ivent.com</p> <p>Password: 123456</p>	Berhasil <i>login</i> dan berpindah halaman ke <i>homepage</i>
	TC2.2	Melakukan <i>login</i> dengan mengosongkan salah satu <i>field</i>	Sudah berada di halaman <i>login user</i>	<ol style="list-style-type: none"> 1.Klik <i>field email</i> 2.Masukkan <i>email</i> yang sudah terdaftar 3.Klik tombol masuk 	Email: user@Ivent.com	Tetap berada di <i>login page</i> dan muncul <i>alert</i> karena salah satu <i>field</i> belum terisi
	TC2.3	Melakukan <i>login</i> dengan mengosongkan satu <i>field</i> ada	Sudah berada di halaman <i>login user</i>	<ol style="list-style-type: none"> 1.Klik <i>field password</i> 2.Masukkan <i>password</i> yang sudah terdaftar 3.Klik tombol masuk 	Password: 123456	Tetap berada di <i>login page</i> dan muncul <i>alert</i> karena salah satu <i>field</i> belum terisi
	TC2.4	Memasukkan <i>email</i> & <i>password</i> pengguna yang tidak terdaftar pada database sistem	Sudah berada di halaman <i>login user</i>	<ol style="list-style-type: none"> 1.Klik <i>field email</i> 2.Masukkan <i>email</i> yang belum terdaftar 3.Klik <i>field password</i> 4.Masukkan <i>password</i> yang belum terdaftar 5.Klik tombol masuk 	<p>Email: cobamasuk2@gmail.com</p> <p>Password: 12345678</p>	Tetap berada di <i>login page</i> dan muncul <i>alert</i> karena <i>email</i> & <i>password</i> yang dimasukkan salah

Tabel 4.3 *Test Suites* lupa password

Fitur	Test Case ID	Test Case Description	Pre-Condition	Test Steps	Test Data	Expected Result
3.Lupa Password	TC3.1	Masukkan <i>email</i> yang sudah terdaftar untuk mendapatkan <i>link reset password</i>	Sudah berada pada halaman <i>login user</i>	1.Klik “Lupa Kata Sandi“ 2.Klik <i>field email</i> 3.Masukkan email yang sudah terdaftar 4.klik tombol “Send Password Link”	Email: cobamasuk@gmail.com	Mendapatkan <i>email</i> dari sistem untuk <i>reset password</i>
	TC3.2	Masukkan <i>email</i> yang belum terdaftar pada sistem	Sudah berada pada halaman <i>login user</i>	1.Klik “Lupa Kata Sandi“ 2.Klik <i>field email</i> 3.Masukkan email yang sudah terdaftar 4.klik tombol “Send Password Link”	Email: cobamasuk2@gmail.com	Tidak mendapatkan <i>email</i> dari sistem untuk <i>reset password</i> karena email yang diinputkan belum terdaftar di sistem.

Tabel 4.4 *Test Suites* Menu pada Homepage

Fitur	Test Case ID	Test Case Description	Pre-Condition	Test Steps	Test Data	Expected Result
4.Menu	TC4.1	Verifikasi fungsionalitas menu pada <i>homepage</i>	Sudah berada di <i>homepage</i>	1.Klik Menu Sewa	-	Berpindah dari <i>homepage</i> ke halaman pencarian produk
	TC4.2	Verifikasi fungsionalitas menu pada <i>homepage</i>	Sudah berada di <i>homepage</i>	1.Klik Menu buka toko	-	Berpindah dari <i>homepage</i> ke halaman buka toko
	TC4.3	Verifikasi fungsionalitas menu pada <i>homepage</i>	Sudah berada di <i>homepage</i>	1.Klik Menu pesanan	-	Berpindah dari <i>homepage</i> ke daftar pesanan / keranjang
	TC4.4	Verifikasi fungsionalitas menu pada <i>homepage</i>	Sudah berada di <i>homepage</i>	1.Klik Menu Profil 2.Klik Keluar	-	User keluar dari sistem
	TC4.5	Verifikasi fungsionalitas menu pada <i>homepage</i>	Sudah berada di <i>homepage</i>	1.Klik Profil 2.Klik Akun	-	Berpindah dari <i>homepage</i> ke akun user

Tabel 4.5 *Test Suites* fitur Pencarian

Fitur	Test Case ID	Test Case Description	Pre-Condition	Test Steps	Test Data	Expected Result
5.Pencarian	TC5.1	Mencari produk dengan menggunakan nama produk tanpa memilih kategori produk	Sudah berada pada halaman produk / <i>homepage</i>	1.Klik kolom pencarian 2.Masukkan nama produk 3.Klik tombol cari / <i>icon</i> pencarian	Panggung	Sistem menampilkan produk berdasarkan kata kunci yang dimasukkan
	TC5.2	Mencari produk dengan menggunakan nama produk dan kategori produk	Sudah berada pada halaman produk / <i>homepage</i>	1.Pilih kategori pada kolom pencarian 2.Klik kolom pencarian 3.Masukkan nama produk 4.Klik tombol cari / <i>icon</i> pencarian	Cajon Kategori: vendor	Sistem menampilkan produk berdasarkan kata kunci dan kategori yang dipilih
	TC5.3	Mencari produk dengan dengan menggunakan kata kunci	Sudah berada pada halaman produk / <i>homepage</i>	1.Pilih kategori pada kolom pencarian 2.Klik kolom pencarian 3.Masukkan nama produk dengan kata kunci yang tidak sama 4.Klik tombol cari / <i>icon</i> pencarian	Kata kunci: "So", untuk mencari produk bernama <i>sound</i> sistem	Sistem menampilkan produk dengan nama yang mendekati kata kunci yang dimasukkan

Tabel 4.6 *Test Suites* menambahkan produk

Fitur	Test Case ID	Test Case Description	Pre-Condition	Test Steps	Test Data	Expected Result
6.Tambahkan produk	TC6.1	Menambahkan produk dengan mengosongkan salah satu <i>field</i>	Sudah berada pada halaman buka toko	1.Klik tombol tambahkan produk 2.Klik <i>field</i> detail produk 3.Tambahkan detail produk	<i>email:</i> cobamasuk1@gmail.com , <i>password:</i> 12345678	Tidak dapat menambahkan produk dan muncul pesan bahwa salah

				<p>4.Klik tombol 'browse' pada <i>field</i> gambar</p> <p>5.Pilih gambar produk lalu tambahkan</p> <p>6. Klik <i>field</i> harga</p> <p>7.Tambahkan harga produk</p> <p>8.Pilih kategori sesuai produk yang ditambahkan</p> <p>9.Klik <i>field</i> stok produk</p> <p>10.Tambahkan stok produk yang tersedia</p> <p>11.Pilih lokasi</p> <p>12.Klik tombol 'submit'</p>	<p>detail produk: menyesuaikan produk gambar produk: menyesuaikan produk harga produk: 22.000.000 kategori produk: Vendor stok produk: 2 lokasi produk: Yogyakarta</p>	<p>satu <i>field</i> kosong (<i>field</i> nama produk)</p>
TC6.2	Menambahkan produk dengan mengisi <i>field</i> harga produk dengan kalimat	Sudah berada pada halaman buka toko	<p>1.Klik tombol tambahkan produk</p> <p>2.Klik <i>field</i> detail produk</p> <p>3.Tambahkan detail produk</p> <p>4.Klik tombol 'browse' pada <i>field</i> gambar</p> <p>5.Pilih gambar produk lalu tambahkan</p> <p>6. Klik <i>field</i> harga</p> <p>7.Tambahkan harga produk</p> <p>8.Pilih kategori sesuai produk yang ditambahkan</p> <p>9.Klik <i>field</i> stok produk</p> <p>10.Tambahkan stok produk yang tersedia</p> <p>11.Pilih lokasi</p> <p>12.Klik tombol 'submit'</p>	<p><i>email:</i> cobamasuk1@gmail.com <i>password:</i> 12345678 nama produk: <i>Cathering Makanan Acara Nikah</i> detail produk: menyesuaikan Gambar produk: menyesuaikan produk harga produk: Dua puluh dua juta rupiah kategori produk: Vendor stok produk: 2 lokasi produk: Yogyakarta</p>	<p>Tidak dapat menambahkan produk dan muncul pesan bahwa <i>field</i> harga produk harus diisi dengan angka</p>	
TC6.3	Menambahkan produk dengan tidak menambahkan gambar produk	Sudah berada pada halaman buka toko	<p>1.Klik tombol tambahkan produk</p> <p>2.Klik <i>field</i> detail produk</p> <p>3.Tambahkan detail produk</p> <p>4.Klik tombol 'browse' pada <i>field</i> gambar</p>	<p><i>email:</i> cobamasuk1@gmail.com <i>password:</i> 12345678 nama produk: <i>Cathering Makanan Acara Nikah</i></p>	<p>Tidak dapat menambahkan produk dan muncul pesan bahwa <i>harus</i></p>	

			<ul style="list-style-type: none"> 5. Klik <i>field</i> harga 6. Tambahkan harga produk 7. Pilih kategori sesuai produk yang ditambahkan 8. Klik <i>field</i> stok produk 9. Tambahkan stok produk yang tersedia 10. Pilih lokasi 11. Klik tombol '<i>submit</i>' 	<p>detail produk: menyesuaikan harga produk: 22.000.000 kategori produk: Vendor stok produk: 2 lokasi produk: Yogyakarta</p>	menambahkan gambar produk
TC6.4	Menambahkan produk yang ingin dijual pada aplikasi Ivent	Sudah berada pada halaman buka toko	<ul style="list-style-type: none"> 1. Klik tombol tambahkan produk 2. Klik <i>field</i> nama produk 3. Tambahkan nama produk 4. Klik <i>field</i> detail produk 5. Tambahkan detail produk 6. Klik tombol '<i>browse</i>' pada <i>field</i> gambar 7. Pilih gambar produk lalu tambahkan 8. Klik <i>field</i> harga 9. Tambahkan harga produk 10. Pilih kategori sesuai produk yang ditambahkan 11. Klik <i>field</i> stok produk 12. Tambahkan stok produk yang tersedia 13. Pilih lokasi 14. Klik tombol '<i>submit</i>' 	<p><i>email:</i> cobamasuk1@gmail.com, <i>password:</i> 12345678 nama produk: <i>Cathering Makanan Acara Nikah</i> detail produk: menyesuaikan harga produk: 22.000.000 kategori produk: Vendor stok produk: 2 lokasi produk: Yogyakarta</p>	Berhasil menambahkan produk dan berpindah ke halaman buka toko

Tabel 4.7 *Test Suites edit produk toko*

Fitur	Test Case ID	Test Case Description	Pre-Condition	Test Steps	Test Data	Expected Result
7. Edit produk toko	TC 7.1	Mengubah informasi pada produk dengan mengisi semua <i>field</i> pada halaman <i>edit</i> produk	Berada pada halaman produk	<ol style="list-style-type: none"> 1.Pilih produk yang ingin diubah 2.Klik tombol <i>edit</i> produk 3.Mengisi <i>field</i> nama produk 4.Mengisi <i>field</i> rincian produk 5.Mengunggah gambar produk 6.Mengisi harga produk 7.Memilih kategori produk 8.Mengisi stok produk 9. Memilih lokasi produk 10.Klik tombol 'submit ' 	<i>email:</i> cobamasuk1@gmail.com , <i>password:</i> 12345678 nama produk: <i>Cathering</i> Makanan acara nikah 2 detail produk: menyesuaikan harga produk: 33.000.000 kategori produk: Vendor stok produk: 4 lokasi produk: Yogyakarta	Berhasil mengubah produk dan berpindah ke halaman produk
	TC.7.2	Mengubah informasi pada produk dengan mengisi <i>field</i> harga dengan format kata	Berada pada halaman produk	<ol style="list-style-type: none"> 1.Pilih produk yang ingin diubah 2.Klik tombol <i>edit</i> produk 3.Mengisi <i>field</i> nama produk 4.Mengisi <i>field</i> rincian produk 5.Mengunggah gambar produk 6.Mengisi harga produk 7.Memilih kategori produk 8.Mengisi stok produk 9. Memilih lokasi produk 10.Klik tombol 'submit ' 	<i>email:</i> cobamasuk1@gmail.com , <i>password:</i> 12345678 nama produk: <i>Cathering</i> Makanan acara nikah 2 detail produk: menyesuaikan harga produk: Tiga puluh tiga juta rupiah kategori produk: Vendor stok produk: 4 lokasi produk: Yogyakarta	Tetap berada pada halaman <i>edit</i> produk dan muncul notifikasi bahwa format yang dimasukkan salah.
	TC7.3	Mengubah informasi pada produk dengan mengisi <i>field</i> stok produk dengan format kata		<ol style="list-style-type: none"> 1.Pilih produk yang ingin diubah 2.Klik tombol <i>edit</i> produk 3.Mengisi <i>field</i> nama produk 4.Mengisi <i>field</i> rincian produk 5.Mengunggah gambar produk 6.Mengisi harga produk 7.Memilih kategori produk 	<i>email:</i> cobamasuk1@gmail.com , <i>password:</i> 12345678 nama produk: <i>Cathering</i> Makanan acara nikah 2 detail produk: menyesuaikan harga produk: 33.000.000 kategori produk: Vendor	Tetap berada pada halaman <i>edit</i> produk dan muncul notifikasi bahwa format yang

				8.Mengisi stok produk 9. Memilih lokasi produk 10.Klik tombol ‘submit ‘	stok produk: Empat lokasi produk: Yogyakarta	dimasukkan salah.
	TC7.4	Mengubah informasi pada produk dengan mengosongkan salah satu <i>field</i>		1.Pilih produk yang ingin diubah 2.Klik tombol <i>edit</i> produk 3.Mengisi <i>field</i> nama produk 4.Mengunggah gambar produk 5.Mengisi harga produk 6.Memilih kategori produk 7.Mengisi stok produk 8. Memilih lokasi produk 9.Klik tombol ‘submit ‘	<i>email:</i> cobamasuk1@gmail.com , <i>password:</i> 12345678 nama produk: Cathering Makanan acara nikah 2 harga produk: 33.000.000 kategori produk: Vendor stok produk: 4 lokasi produk: Yogyakarta	Tetap berada pada halaman <i>edit</i> produk dan muncul notifikasi bahwa semua <i>field</i> harus diisi / tidak boleh ada <i>field</i> yang kosong.

Tabel 4.8 *Test suites* melihat informasi produk

Fitur	<i>Test Case ID</i>	<i>Test Case Description</i>	<i>Pre-Condition</i>	<i>Test Steps</i>	<i>Test Data</i>	<i>Expected Result</i>
8.Melihat detail produk	TC8.1	Melihat detail gambar produk	Sudah melakukan <i>login</i> dan berada pada halaman sewa <i>website</i> Ivent	1.Pilih produk yang ingin dilihat 2.Klik nama produk atau gambar produk 3.Pada halaman detail produk, klik gambar produk	<i>email:</i> cobamasuk1@gmail.com , <i>password:</i> 12345678	Gambar produk bisa di klik dan muncul <i>pop up</i> berisi gambar produk secara lebih detail.
	TC8.2	Melihat informasi detail produk		1.Pilih produk yang ingin dilihat 2.Klik nama produk atau gambar produk 3.Verifikasi informasi produk yang ditampilkan	<i>email:</i> cobamasuk1@gmail.com , <i>password:</i> 12345678	Semua informasi produk dapat ditampilkan dengan jelas

Tabel 4.9 *Test suites* memesan produk

Fitur	Test Case ID	Test Case Description	Pre-Condition	Test Steps	Test Data	Expected Result
9.Memesan Produk	TC9.1	Melakukan pemesanan produk	Sudah melakukan <i>login</i> dan berada pada halaman sewa <i>website</i> Ivent	<ol style="list-style-type: none"> 1.Pilih produk yang ingin dilihat 2.Klik nama produk atau gambar produk 3.klik <i>field</i> tanggal mulai 4.Masukkan tanggal mulai 5.klik <i>field</i> tanggal selesai 6.Masukkan tanggal selesai 7.Klik tombol ‘tambah pesanan’ 	<i>email:</i> cobamasuk1@gmail.com , <i>password:</i> 12345678 produk yang dipilih: <i>Event organizer</i> konser Tanggal mulai: 20 September 2022 Tanggal selesai: 21 September 2022	Berhasil menambahkan pesanan dan berpindah pada halaman pesanan
	TC9.2	Melakukan pemesanan produk dengan tanggal mulai pada tahun 2022 dan tanggal selesai pada tahun 2000	Sudah melakukan <i>login</i> dan berada pada halaman sewa <i>website</i> Ivent	<ol style="list-style-type: none"> 1.Pilih produk yang ingin dilihat 2.Klik nama produk atau gambar produk 3.klik <i>field</i> tanggal mulai 4.Masukkan tanggal mulai 5.klik <i>field</i> tanggal selesai 6.Masukkan tanggal selesai 7.Klik tombol ‘tambah 	<i>email:</i> cobamasuk1@gmail.com , <i>password:</i> 12345678 produk yang dipilih: <i>Event organizer</i> konser Tanggal mulai: 20 September 2022 Tanggal selesai: 21 September 2000	Tetap pada halaman detail produk dan muncul notifikasi bahwa format yang diinputkan salah.

Tabel 4.10 *Test suites* melihat pesanan

Fitur	Test Case ID	Test Case Description	Pre-Condition	Test Steps	Test Data	Expected Result
10.Melihat pesanan	TC10.1	Melihat pesanan yang sudah ditambahkan	Sudah menambahkan pesanan produk	1.Pilih menu profil dan muncul menu <i>dropdown</i> .	<i>email:</i> cobamasuk1@gmail.com , <i>password:</i> 12345678	Berpindah ke halaman pesanan dan menampilkan produk yang sudah

			dan berada pada <i>homepage</i> Ivent	2.Pilih menu pesanan anda pada <i>dropdown</i> .		ditambahkan ke pesanan beserta harga sewa produk.
	TC10.2	Melihat pesanan yang sudah ditambahkan		1.Klik menu pesanan pada <i>homepage</i> Ivent	<i>email:</i> cobamasuk1@gmail.com , <i>password:</i> 12345678	Berpindah ke halaman pesanan dan menampilkan produk yang sudah ditambahkan ke pesanan beserta harga sewa produk.

Tabel 4.11 *Test suites* menambahkan alamat *user* (konsumen)

Fitur	<i>Test Case ID</i>	<i>Test Case Description</i>	<i>Pre-Condition</i>	<i>Test Steps</i>	<i>Test Data</i>	<i>Expected Result</i>
11.Menambahkan alamat pemesan	TC11.1	Menambahkan alamat pemesan	Sudah menambahkan pesanan dan berada pada halaman pesanan	1.Klik tombol <i>checkout</i> . 2.klik tombol tambahkan alamat 3.Mengisi <i>field</i> nama penerima 4.Mengisi <i>field</i> alamat 5.Mengisi <i>field</i> nomor telepon 6.Mengisi <i>field</i> provinsi 7.Mengisi <i>field</i> kota 8.Mengisi <i>field</i> kecamatan 9.Mengisi <i>field</i> kelurahan 10.Mengisi <i>field</i> kode pos 11.Klik tombol simpan	Nama penerima: Fadel Alamat: Perumahan agatama c18 Nomor telepon: 081317111999 Provinsi: DI Yogyakarta Kota: Yogyakarta Kecamatan: Umbulharjo Kelurahan: Giwangan Kodepos: 55163	Berhasil menambahkan alamat pemesan.
	TC11.2	Menambahkan alamat pemesan dengan mengisi <i>field</i> nomor telepon dengan format kata	Sudah berada pada halaman <i>checkout</i>	1.klik tombol tambahkan alamat 2.Mengisi <i>field</i> nama penerima 3.Mengisi <i>field</i> alamat	Nama penerima: Fadel Alamat: Perumahan agatama c18 Nomor telepon: satuduatigaempat	Tetap berada pada halaman menambahkan alamat pengiriman dan

				4.Mengisi <i>field</i> nomor telepon 5.Mengisi <i>field</i> provinsi 6.Mengisi <i>field</i> kota 7.Mengisi <i>field</i> kecamatan 8.Mengisi <i>field</i> kelurahan 9.Mengisi <i>field</i> kode pos 10.Klik tombol simpan	Provinsi: DI Yogyakarta Kota: Yogyakarta Kecamatan: Umbulharjo Kelurahan: Giwangan Kodepos: 55163	muncul notifikasi bahwa format yang diinputkan tidak sesuai.
	TC11.3	Menambahkan alamat pemesan dengan mengosongkan salah satu <i>field</i>	Sudah berada pada halaman <i>checkout</i>	1.klik tombol tambahkan alamat 2.Mengisi <i>field</i> nama penerima 3.Mengisi <i>field</i> alamat 4.Mengisi <i>field</i> nomor telepon 5.Mengisi <i>field</i> provinsi 6.Mengisi <i>field</i> kota 7.Mengisi <i>field</i> kecamatan 8.Mengisi <i>field</i> kelurahan 9.Klik tombol simpan	Nama penerima: Fadel Alamat: Perumahan agatama c18 Nomor telepon: 081317111999 Provinsi: DI Yogyakarta Kota: Yogyakarta Kecamatan: Umbulharjo Kelurahan: Giwangan	Muncul notifikasi tidak dapat menambahkan alamat karena salah satu <i>field</i> kosong

Tabel 4.12 *Test suites* menghapus pesanan

Fitur	<i>Test Case ID</i>	<i>Test Case Description</i>	<i>Pre-Condition</i>	<i>Test Steps</i>	<i>Test Data</i>	<i>Expected Result</i>
12.Menghapus pesanan	TC12.1	Menghapus salah satu item di pesanan	Sudah menambahkan pesanan dan berada <i>homepage</i> Ivent	1.Klik menu pesanan pada <i>homepage</i> Ivent 2.Pilih produk yang ingin dihapus 3.Klik tombol hapus pada produk yang dipilih		Muncul notifikasi berhasil menghapus produk yang dipilih

	TC12.2	Menghapus keseluruhan item yang ada pada halaman pesanan		1..Klik menu pesanan pada <i>homepage</i> Ivent 2.Klik tombol kosongkan untuk menghapus		Muncul notifikasi berhasil menghapus seluruh item pada halaman pesanan
--	--------	--	--	--	--	--

Tabel 4.13 *Test suites* melihat transaksi

Fitur	Test Case ID	Test Case Description	Pre-Condition	Test Steps	Test Data	Expected Result
13.Melihat transaksi	TC13.1	Melihat keseluruhan transaksi yang dilakukan <i>user</i>	Sudah menambahkan pesanan dan berada <i>homepage</i> Ivent	1.Klik menu transaksi pada <i>homepage</i> Ivent		Berpindah ke halaman transaksi dan menampilkan keseluruhan transaksi
	TC13.2	Melihat detail salah satu transaksi <i>user</i>		1.Klik menu transaksi pada <i>homepage</i> Ivent 2.Klik tombol detail pada salah satu transaksi		Berpindah ke halaman pembayaran dan menampilkan detail dari transaksi yang dipilih

Tabel 4.14 *Test suites* unggah bukti pembayaran

Fitur	Test Case ID	Test Case Description	Pre-Condition	Test Steps	Test Data	Expected Result
14.Unggah bukti pembayaran	TC14.1	Mengunggah bukti pembayaran pada salah satu transaksi	Berada pada halaman transaksi	1.Klik tombol detail pada salah satu transaksi 2.klik tombol ' <i>choose file</i> ' pada bagian upload bukti pembayaran. 3.Pilih gambar bukti pembayaran dengan ekstensi .jpg / .png 4.Klik Open 5.Lalu klik tombol 'Kirim'	Gambar bukti pembayaran	Berhasil melakukan pembayaran dan mendapatkan bukti pembayaran
	TC14.2	Mengunggah bukti pembayaran pada salah satu transaksi		1.Klik tombol detail pada salah satu transaksi	File yang berekstensi	Gagal mengunggah bukti

		menggunakan file yang berekstensi selain .jpg, .jpeg, .png dan .pdf.		2.klik tombol ' <i>choose file</i> ' pada bagian upload bukti pembayaran. 3.Pilih bukti pembayaran dengan file selain gambar 4.Klik Open 5.Lalu klik tombol 'Kirim'	selain .jpg, .jpeg, .png dan .pdf.	pembayaran karena file ekstensi tidak sesuai format
--	--	--	--	--	------------------------------------	---

Tabel 4.15 *Test suites* mengecek pesanan penjual

Fitur	Test Case ID	Test Case Description	Pre-Condition	Test Steps	Test Data	Expected Result
15.Mengecek pesanan untuk penjual	TC15.1	Mengecek apakah data transaksi ditampilkan atau tidak	Berada pada <i>homepage</i> Ivent	1.Klik menu buka toko. 2.Pada halaman buka toko, klik menu transaksi.		Menampilkan data transaksi
	TC15.2	Mengecek apakah data transaksi yang ditampilkan sudah benar atau belum		1.Klik menu buka toko. 2.Pada halaman buka toko, klik menu transaksi.		Menampilkan semua transaksi yang masuk pada penjual

Tabel 4.16 *Test suites* verifikasi bukti pembayaran

Fitur	Test Case ID	Test Case Description	Pre-Condition	Test Steps	Test Data	Expected Result
16.Verifikasi bukti pembayaran oleh Admin	TC16.1	Melihat bukti pembayaran yang sudah diunggah oleh <i>user</i>	Berada pada halaman <i>login</i> Ivent	1.Memasukkan <i>email</i> admin pada <i>field email</i> 2.Memasukkan <i>password</i> admin pada <i>field password</i> 3.Klik menu <i>dashboard</i> pada halaman admin 4.Klik menu <i>dropdown</i> transaksi 5.Klik salah satu transaksi yang sudah mengunggah bukti pembayaran	<i>Email:</i> admin@Ivent.com <i>Password:</i> 123456	Menampilkan transaksi <i>user</i> beserta bukti pembayaran yang sudah diunggah

			6.Klik tombol ‘detail‘		
	TC16.2	Memverifikasi salah satu bukti pembayaran yang telah diunggah oleh <i>user</i>	1.Memasukkan <i>email</i> admin pada <i>field email</i> 2.Memasukkan <i>password</i> admin pada <i>field password</i> 3.Klik menu <i>dashboard</i> pada halaman admin 4.Klik menu <i>dropdown</i> transaksi 5.Klik salah satu transaksi yang sudah mengunggah bukti pembayaran 6.Klik tombol ‘Verifikasi pembayaran‘	<i>Email:</i> admin@Ivent.com <i>Password:</i> 123456	Menampilkan transaksi <i>user</i> yang memiliki bukti pembayaran yang sudah diverifikasi admin

Test case yang sudah dibuat menggunakan *template* yang dimodifikasi dari Tabel 2.1, dapat digunakan untuk pengujian aplikasi lain pada sektor industri acara maupun sektor lainnya dengan memodifikasi dan menyesuaikan atribut-atribut pada *test case* yang akan digunakan. Terdapat beberapa *test case* yang tidak perlu dimodifikasi untuk digunakan pada aplikasi lain seperti contoh *test case* fitur lupa *password* pada Tabel 4.3 dan *test case* fitur pencarian pada Tabel 4.5. *Test case* tersebut dapat digunakan di aplikasi berbasis *website* Bridestory.com dengan penyesuaian pada kolom atribut *pre-condition*, *test steps* dan *test data*.

Terdapat beberapa skenario dalam pengujian aplikasi Ivent yang menggunakan *template* seperti yang sudah dikaji pada sub-bab 2.6 (Ihsan, 2022). Beberapa *test scenario* yang digunakan seperti pada Tabel 4.2 fitur *login* untuk memverifikasi apakah muncul pesan *error* jika memasukan *username* atau *password* yang salah. Pada Tabel 4.3 fitur lupa *password* untuk memverifikasi bahwa *user* akan mendapatkan *email* setelah memasukkan alamat *email* yang valid pada halaman lupa *password*.

Pada Tabel 4.8 fitur melihat detail produk untuk memverifikasi apakah stok produk terlihat atau tidak dan memverifikasi apakah produk dikategorisasikan atau tidak. Pada Tabel 4.14 fitur mengunggah bukti pembayaran untuk memverifikasi apakah *user* hanya dapat mengunggah file dengan format ekstensi yang diizinkan saja.

4.2 Test Execution

Pengujian pada aplikasi Ivent dilakukan 2 kali. Pengujian pertama adalah pengujian setiap *test suites* yang sudah dibuat pada subbab 4.1 untuk mencari hasil pengujian berupa *actual result*, status dan rata-rata waktu yang diperlukan untuk menjalankan setiap *test case*. Jika terdapat fitur *test case* yang memiliki status *fail*, maka fitur tersebut akan diperbaiki lalu diuji kembali untuk mendapatkan status *pass* / berhasil dan rata-rata waktunya. Dalam *test execution* akan menggunakan tabel dengan atribut Fitur, *Test Case ID*, *Actual Result* dan Status. Gambar dan waktu hasil pengujian setiap *test case* juga ditambahkan.

4.2.1 Hasil Pengujian Pertama

Tujuan dilakukan pengujian pertama untuk mengetahui apakah fitur-fitur yang tersedia pada aplikasi Ivent sudah sesuai dengan *software requirement* yang ada atau belum. Fitur-fitur yang akan diuji dapat dilihat pada Tabel 3.1 dan menggunakan langkah-langkah yang sudah dibuat dalam *test suite* pada subbab 4.1.

Pengujian Fitur Register

Fitur *register* merupakan fitur pertama yang akan diuji. Hasil pengujian fitur *register* tertera pada Tabel 4.17 dan Gambar 4.2. Rata-rata waktu yang dibutuhkan untuk menjalankan setiap *test case* pada fitur *register* adalah 14.75 detik. Dari empat *test case* yang dijalankan pada fitur *register*, semua berstatus *pass* karena *Expected Result* sama dengan *Actual Result* yang didapat.

Tabel 4.17 Hasil pengujian fitur *register*

Fitur	Test Case ID	Actual Result	Status
Register	TC1.1	Tetap berada pada halaman <i>register</i> dan muncul <i>alert</i> bahwa salah satu <i>field</i> belum diisi.	Pass
	TC1.2	Tetap berada di halaman <i>register</i> dan muncul <i>alert</i> karena <i>password</i> dan konfirmasi <i>password</i> dimasukkan tidak sama	Pass
	TC1.3	Tetap berada di halaman <i>register</i> dan muncul <i>alert</i> karena <i>email</i> yang dimasukkan sudah pernah terdaftar pada sistem	Pass

	TC1.4	Berhasil mendaftar sebagai <i>user</i> dan berpindah ke <i>homepage</i> Ivent	Pass
--	-------	---	------



Gambar 4.2 Hasil pengujian fitur *register*

Pengujian Fitur *Login*

Pengujian berikutnya dilakukan pada fitur *login*. Rata-rata waktu yang dibutuhkan untuk menjalankan setiap *test case* pada fitur *login* adalah 5,5 detik. Dari 4 *test case* yang dijalankan pada fitur ini, semua *test case* mendapatkan hasil *pass*. Hasil pengujian fitur *login* disajikan dalam Tabel 4.18.

Tabel 4.18 Hasil pengujian fitur *login*

Fitur	Test Case ID	Actual Result	Status
<i>Login</i>	TC2.1	Berhasil <i>login</i> dan berpindah halaman ke <i>homepage</i>	Pass

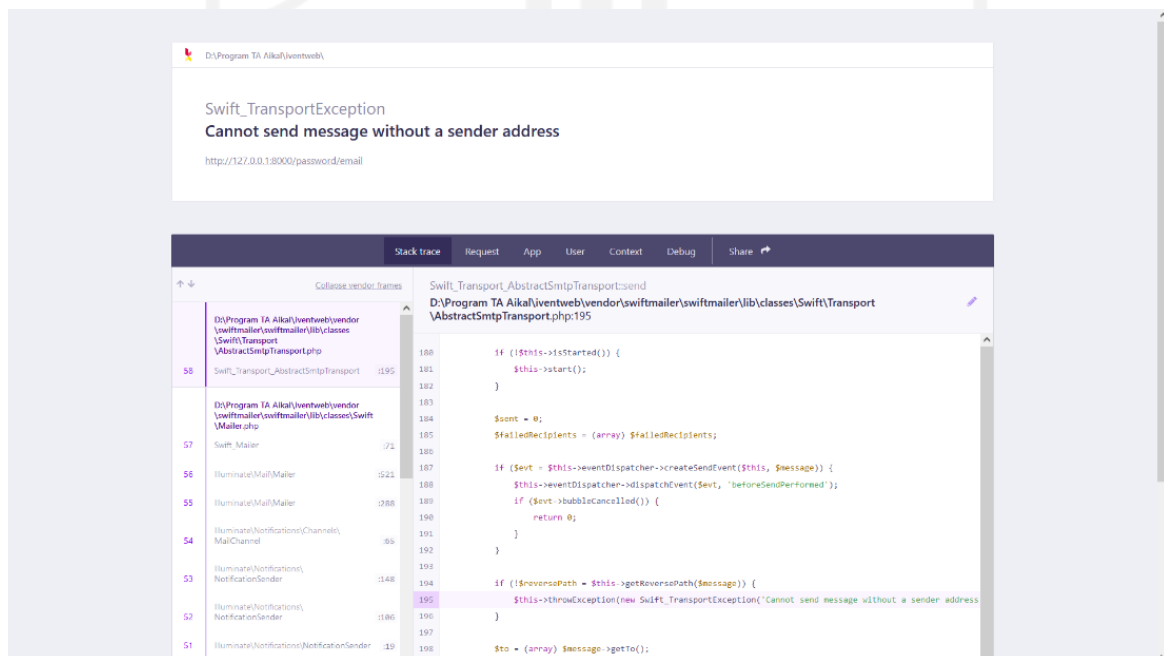
	TC2.2	Tetap berada di <i>login page</i> dan muncul <i>alert</i> karena salah satu <i>field</i> belum terisi	Pass
	TC2.3	Tetap berada di <i>login page</i> dan muncul <i>alert</i> karena salah satu <i>field</i> belum terisi	Pass
	TC2.4	Tetap berada di <i>login page</i> dan muncul <i>alert</i> karena salah satu <i>field</i> belum terisi	Pass

Pengujian Fitur Lupa password

Pengujian yang dilakukan pada fitur ketiga ini mendapatkan hasil *fail* pada 2 *test case* yang dijalankan. Ketika memasukkan *email* pada *field email address* dan klik tombol ‘*send password reset link*’, muncul *error* bahwa tidak terdapat *email* pengirim yang berfungsi untuk mengirimkan *link reset password* kepada *email* yang diinputkan oleh *user*. Pengujian yang dilakukan *fail*, sehingga tidak dapat menghitung waktu rata-rata untuk menjalankan setiap *test case* pada fitur ini. Hasil pengujian disajikan dalam Tabel 4.19 dan Gambar 4.3.

Tabel 4.19 Hasil pengujian fitur lupa *password*

Fitur	Test Case ID	Actual Result	Status
Lupa Password	TC3.1	muncul <i>error</i> bahwa tidak terdapat <i>email</i> pengirim yang berfungsi untuk mengirimkan <i>link reset password</i> kepada <i>email</i> yang diinputkan oleh <i>user</i> .	Fail
	TC3.2	muncul <i>error</i> bahwa tidak terdapat <i>email</i> pengirim yang berfungsi untuk mengirimkan <i>link reset password</i> kepada <i>email</i> yang diinputkan oleh <i>user</i> .	Fail



Gambar 4.3 Hasil pengujian fitur lupa *password*

Pengujian Fitur Menu

Pengujian pada fitur menu mendapatkan hasil *pass* pada 5 *test case* yang dijalankan. Rata-rata waktu yang dibutuhkan untuk menjalankan setiap *test case* adalah 4,2 detik. Hasil pengujian fitur menu disajikan dalam Tabel 4.20.

Tabel 4.20 Hasil pengujian fitur menu

Fitur	Test Case ID	Actual Result	Status
Menu	TC4.1	Berpindah dari <i>homepage</i> ke halaman pencarian produk	<i>Pass</i>
	TC4.2	Berpindah dari <i>homepage</i> ke halaman buka toko	<i>Pass</i>
	TC4.3	Berpindah dari <i>homepage</i> ke daftar pesanan / keranjang.	<i>Pass</i>
	TC4.4	<i>User</i> keluar dari sistem	<i>Pass</i>
	TC4.5	Berpindah dari <i>homepage</i> ke halaman akun <i>user</i>	<i>Pass</i>

Pengujian Fitur Pencarian

Pengujian berikutnya pada fitur pencarian. Pada fitur ini mendapatkan hasil *pass* pada 3 *test case* yang dijalankan. Rata-rata waktu yang dibutuhkan adalah 7,6 detik. Hasil pengujian fitur ini disajikan dalam Tabel 4.21.

Tabel 4.21 Hasil pengujian fitur pencarian

Fitur	Test Case ID	Actual Result	Status
Pencarian	TC5.1	Sistem menampilkan produk berdasarkan kata kunci yang dimasukkan	<i>Pass</i>
	TC5.2	Sistem menampilkan produk berdasarkan kata kunci dan kategori yang dipilih	<i>Pass</i>
	TC5.3	Sistem menampilkan produk dengan nama yang mendekati kata kunci yang dimasukkan	<i>Pass</i>

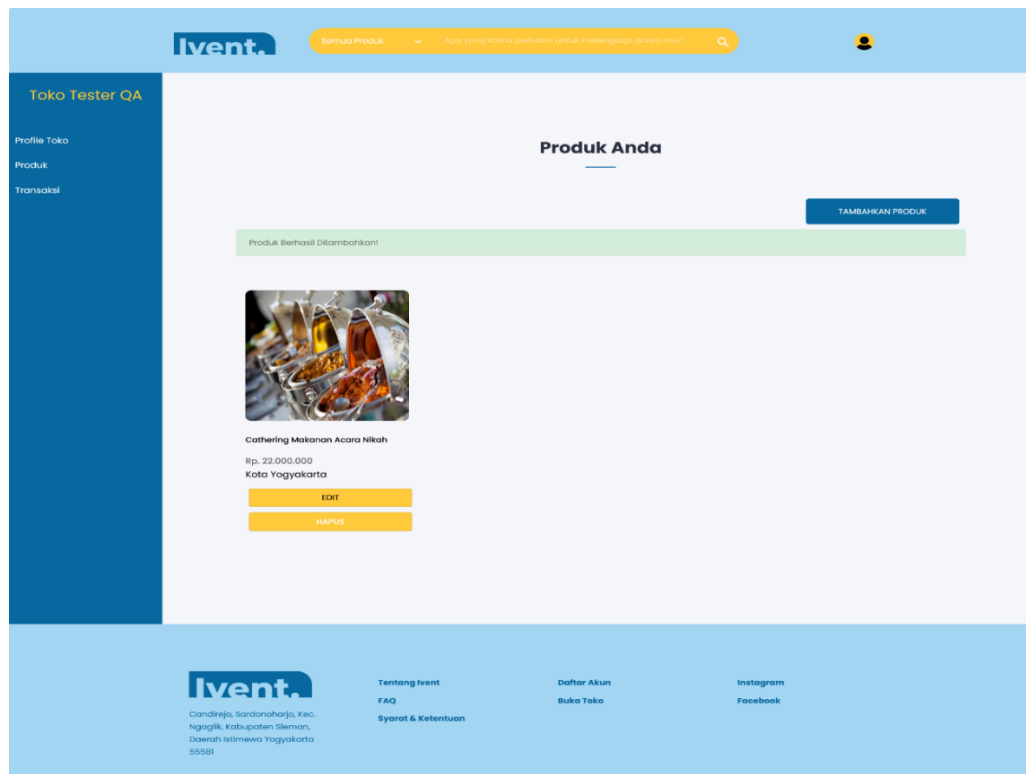
Pengujian Fitur Tambahkan Produk

Pengujian pada fitur tambahkan produk mendapatkan hasil *pass* dari total 5 *test case* ketika dijalankan. Rata-rata waktu yang dibutuhkan dari total 5 *test case* adalah 23,25 detik. Hasil pengujian fitur ini disajikan dalam Tabel 4.22 dan Gambar 4.4.

Tabel 4.22 Hasil pengujian fitur tambahkan produk

Fitur	Test Case ID	Actual Result	Status
Tambahkan Produk	TC6.1	Tidak dapat menambahkan produk dan muncul pesan bahwa salah satu <i>field</i> kosong (<i>field</i> nama produk)	<i>Pass</i>
	TC6.2	Tidak dapat menambahkan produk dan muncul pesan bahwa <i>field</i> harga produk harus di isi dengan angka	<i>Pass</i>

	TC6.3	Tidak dapat menambahkan produk dan muncul pesan bahwa <i>harus</i> menambahkan gambar produk	Pass
	TC6.4	Berhasil menambahkan produk dan berpindah ke halaman buka toko	Pass



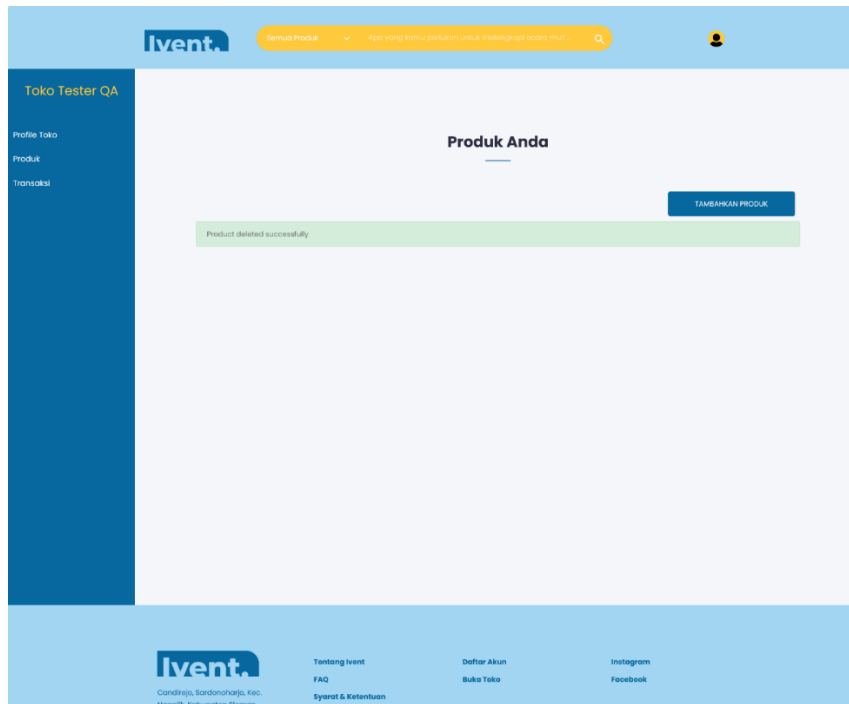
Gambar 4.4 Hasil pengujian fitur tambahkan produk

Pengujian Fitur *Edit* Produk

Hasil eksekusi pengujian fitur *edit* produk pada dapat dilihat pada Tabel 4.23 dan Gambar 4.5. Total 4 *test case* yang dijalankan *fail* / gagal karena *Expected Result* dan *Actual Result* berbeda dan tidak dapat menghitung waktu rata-rata. Ketika klik tombol '*edit* produk', tidak berpindah ke halaman *edit produk* dan produk yang dipilih terhapus sehingga tidak dapat menjalankan 4 *test case* pada fitur ini.

Tabel 4.23 Hasil pengujian fitur *edit* produk

Fitur	Test Case ID	Actual Result	Status
Edit Produk	TC7.1	Ketika klik tombol <i>edit</i> produk, tidak berpindah ke halaman produk dan produk yang dipilih terhapus	Fail
	TC7.2		Fail
	TC7.3		Fail
	TC7.4		Fail



Gambar 4.5 Hasil pengujian fitur *edit* produk

Pengujian Fitur Melihat Produk

Pada fitur ini, total 2 *test case* yang dijalankan mendapatkan hasil *pass*. Rata-rata waktu yang dibutuhkan adalah 3,5 detik. Hasil pengujian pada fitur ini disajikan dalam Tabel 4.24.

Tabel 4.24 Hasil pengujian fitur melihat produk

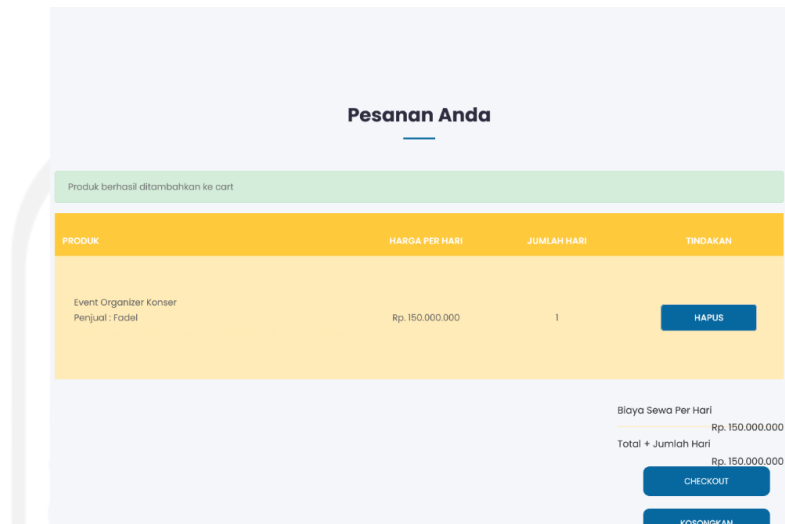
Fitur	Test Case ID	Actual Result	Status
Melihat Produk	TC8.1	Gambar produk bisa diklik dan muncul <i>pop up</i> berisi gambar produk secara lebih detail.	<i>Pass</i>
	TC8.2	Semua informasi produk dapat ditampilkan dengan jelas	<i>Pass</i>

Pengujian Fitur Memesan Produk

Pada fitur ini terdapat 1 *test case* yang mendapatkan hasil *fail* dari total 2 *test case*. *Test case* tersebut *fail* karena *Expected Result* pada *test case* tersebut adalah tetap pada halaman produk dan muncul *alert* bahwa tanggal selesai tidak boleh lebih kecil dari tanggal mulai dan *Actual Result* yang didapatkan adalah berhasil menambahkan pesanan dan berpindah pada halaman pesanan. Hasil pengujian disajikan dalam Tabel 4.25 dan Gambar 4.6

Tabel 4.25 Hasil pengujian fitur memesan produk

Fitur	Test Case ID	Actual Result	Status
Memesan Produk	TC9.1	Berhasil menambahkan pesanan dan berpindah pada halaman pesanan	Pass
	TC9.2		Fail



Gambar 4.6 Hasil pengujian fitur memesan produk

Pengujian Fitur Melihat Pesanan

Fitur ini dapat diakses dengan 2 cara yaitu, pada menu profil dan pada menu pesanan. Total 2 *test case* yang dijalankan dan mendapatkan hasil *pass*. Rata-rata waktu yang dibutuhkan adalah 3 detik. Hasil pengujian pada fitur melihat pesanan disajikan dalam Tabel 4.26.

Tabel 4.26 Hasil pengujian fitur melihat pesanan

Fitur	Test Case ID	Actual Result	Status
Melihat Pesanan	TC10.1	Berpindah ke halaman pesanan dan menampilkan produk yang sudah ditambahkan ke pesanan beserta harga sewa produk.	Pass
	TC10.2	Berpindah ke halaman pesanan dan menampilkan produk yang sudah ditambahkan ke pesanan beserta harga sewa produk.	Pass

Pengujian Fitur Menambahkan Alamat

Terdapat 1 *test case* yang mendapatkan hasil *fail* dari total 3 *test case* yang dijalankan pada pengujian fitur ini. *Expected Result* pada TC11.2 adalah tetap berada pada halaman menambahkan alamat pengiriman dan muncul notifikasi bahwa format yang diinputkan tidak sesuai ketika pada *field* nomor telepon diinput dengan format kata dan *Actual Result* yang

didapatkan adalah berhasil menambahkan alamat pemesan. Hasil pengujian pada fitur ini disajikan dalam Tabel 4.27 dan Gambar 4.7.

Tabel 4.27 Hasil pengujian fitur menambahkan alamat

Fitur	Test Case ID	Actual Result	Status
Menambahkan Alamat	TC11.1	Berhasil menambahkan alamat pemesan.	Pass
	TC11.2	Berhasil menambahkan alamat pemesan.	Fail
	TC11.3	Muncul notifikasi tidak dapat menambahkan alamat karena salah satu <i>field</i> kosong	Pass

The image shows a screenshot of a web application interface for managing shipping addresses. It consists of two main sections: 'Alamat Pengiriman' (Shipping Addresses) and 'Form Alamat Pengiriman' (Shipping Address Form). The 'Alamat Pengiriman' section displays a list of existing addresses with columns for 'Nama Penerima', 'Alamat', and 'No tlp'. There are buttons for 'KEMBALI KE CHECKOUT', 'UBAH KE UTAMA', and 'ALAMAT UTAMA'. The 'Form Alamat Pengiriman' section contains a form with various input fields: 'Nama Penerima', 'Kota', 'Alamat', 'Kecamatan', 'Perumahan', 'Umbulharjo', 'No Tip', 'Kelurahan', 'Giwangan', 'Provinsi', 'Kodepos', and 'DI'. A validation message 'The kodepos field is required.' is shown above the form. A 'SIMPAN' button is located at the bottom right of the form.

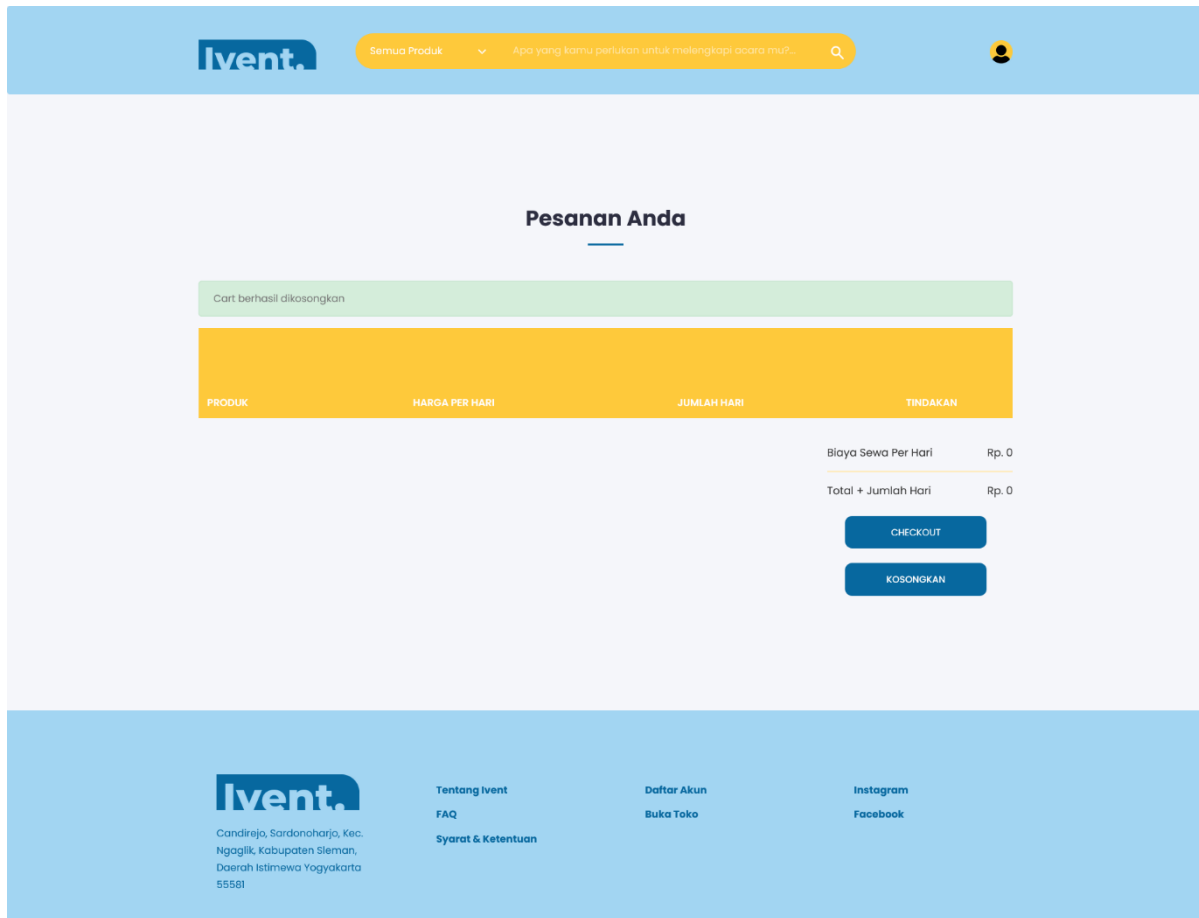
Gambar 4.7 Hasil pengujian fitur menambahkan alamat

Pengujian Fitur Menghapus Pesanan

Pada fitur menghapus pesanan, rata-rata waktu yang dibutuhkan adalah 7 detik. Total 2 *test case* yang dijalankan pada fitur ini mendapatkan hasil *pass*. Hasil pengujian pada fitur menghapus pesanan disajikan dalam Tabel 4.28 dan Gambar 4.8.

Tabel 4.28 Hasil pengujian fitur menghapus pesanan

Fitur	Test Case ID	Actual Result	Status
Menghapus Pesanan	TC12.1	Muncul notifikasi berhasil menghapus produk yang dipilih	Pass
	TC12.2	Muncul notifikasi berhasil menghapus seluruh item pada halaman pesanan	Pass



Gambar 4.8 Hasil pengujian fitur menghapus pesanan

Pengujian Fitur Melihat transaksi

Total 2 *test case* yang dijalankan pada fitur ini mendapatkan hasil *pass*. Hasil pengujian fitur melihat transaksi disajikan dalam Tabel 4.29.

Tabel 4.29 Hasil pengujian fitur melihat transaksi

Fitur	Test Case ID	Actual Result	Status
Melihat Transaksi	TC13.1	Berpindah ke halaman transaksi dan menampilkan keseluruhan transaksi	Pass
	TC13.2	Berpindah ke halaman pembayaran dan menampilkan detail dari transaksi yang dipilih	Pass

Pengujian Fitur Mengunggah Bukti Pembayaran

Pengujian yang dilakukan pada fitur ini mendapatkan *hasil fail* dari total 2 *test case*. Karena ketika mengunggah bukti pembayaran, terjadi *error 404 'not found'* sehingga ke 2 *test case*

yang dijalankan mendapatkan hasil *fail*. Hasil pengujian fitur ini disajikan dalam Tabel 4.30 dan Gambar 4.9.

Tabel 4.30 Hasil pengujian fitur unggah bukti pembayaran

Fitur	Test Case ID	Actual Result	Status
Mengunggah Bukti Pembayaran	TC14.1	Terjadi <i>error</i> 404 ' <i>not found</i> '	<i>Fail</i>
	TC14.2		<i>Fail</i>

404 | Not Found

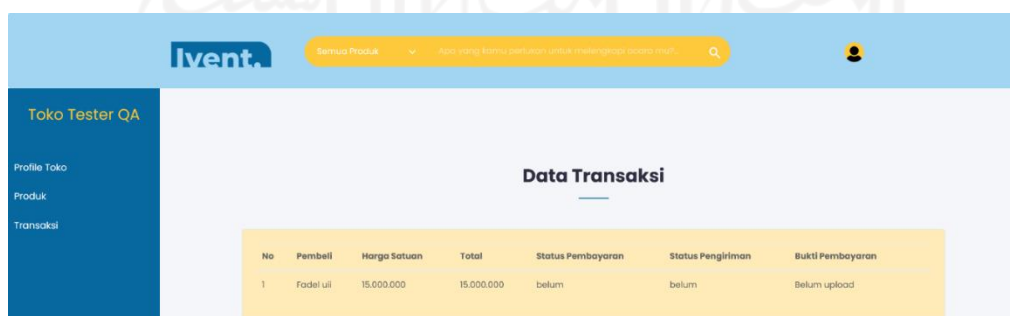
Gambar 4.9 Hasil pengujian fitur mengunggah bukti pembayaran

Pengujian Fitur Mengecek Pesanan untuk Penjual

Pada fitur ini membutuhkan rata-rata waktu 10 detik. Total 2 *test case* yang dijalankan dan semua *test case* mendapatkan hasil *pass*. Hasil dari pengujian pada fitur ini disajikan dalam Tabel 4.31 dan Gambar 4.10.

Tabel 4.31 Hasil pengujian fitur mengecek pesanan untuk penjual

Fitur	Test Case ID	Actual Result	Status
Mengecek Pesanan untuk Penjual	TC15.1	Menampilkan data transaksi	<i>Pass</i>
	TC15.2	Menampilkan semua transaksi yang masuk pada penjual.	<i>Pass</i>



No	Pembeli	Harga Satuan	Total	Status Pembayaran	Status Pengiriman	Bukti Pembayaran
1	Fadel Iul	15.000.000	15.000.000	belum	belum	Belum upload

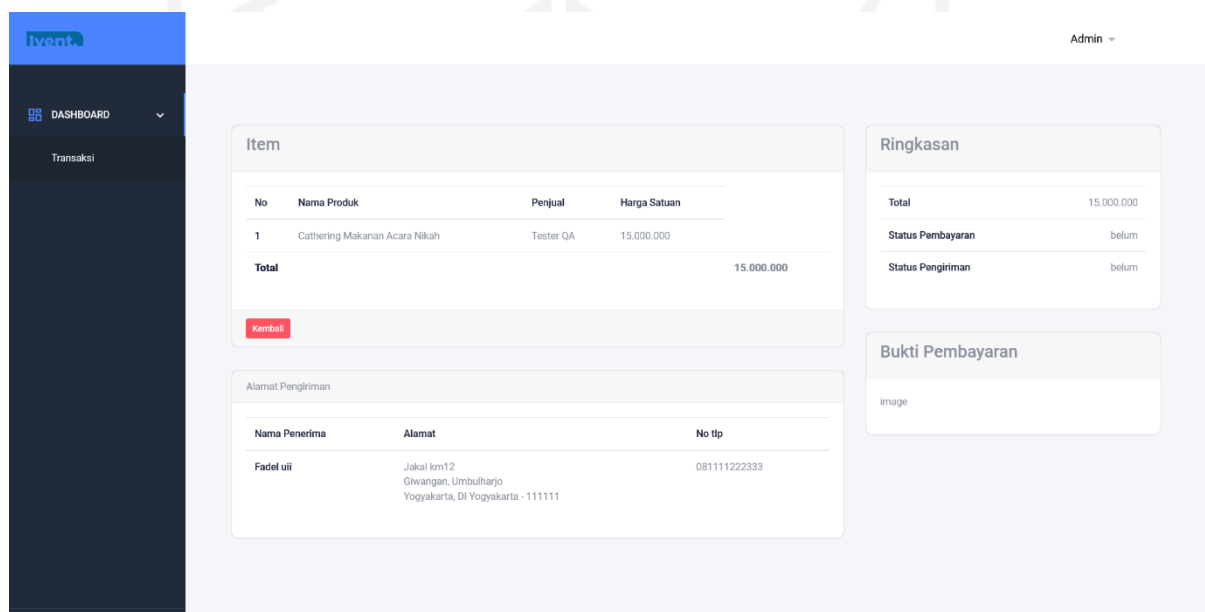
Gambar 4.10 Hasil pengujian fitur mengecek pesanan untuk penjual

Pengujian Fitur Verifikasi Bukti Pembayaran

Pengujian pada fitur ini mendapatkan hasil *fail* dari total 2 *test case*. Karena fitur mengunggah bukti pembayaran masih terjadi *error*, sehingga tidak adanya bukti pembayaran untuk diverifikasi pada halaman *dashboard* admin. Untuk hasil pengujian verifikasi bukti pembayaran disajikan dalam Tabel 4.32 dan Gambar 4.11.

Tabel 4.32 Hasil pengujian verifikasi bukti pembayaran oleh admin

Fitur	Test Case ID	Actual Result	Status
Verifikasi Bukti Pembayaran oleh Admin	TC16.1	Tidak terdapat bukti pembayaran karena pada fitur unggah bukti pembayaran terjadi <i>error</i> .	<i>Fail</i>
	TC16.2		<i>Fail</i>



Gambar 4.11 Hasil pengujian fitur verifikasi bukti pembayaran

Dari pengujian 16 fitur yang telah dilakukan, diperoleh kesimpulan seperti diperlihatkan pada Tabel 4.33. Total fitur yang diuji pada aplikasi Ivent sebanyak 16 fitur dan total *test case* yang digunakan dalam pengujian sebanyak 45 *test case*. Dari total 45 *test case* yang digunakan dalam pengujian, 33 berhasil dan 12 *test case* gagal. Terdapat 10 fitur yang mendapatkan 100% presentasi keberhasilan dan 4 fitur mendapatkan 0% presentasi keberhasilan. Total presentasi keberhasilan seluruh *test case* sebesar 73,33%. Pengujian yang dilakukan menggunakan proses quality assurance dapat menyelesaikan permasalahan pada aplikasi Ivent

Tabel 4.33 Ringkasan pengujian pertama

No	Tabel Fitur	Jumlah Test Case	Jumlah Berhasil	Jumlah Gagal	Presentasi Berhasil
1	<i>Register</i>	4	4	0	100%
2	<i>Login</i>	4	4	1	100%
3	<i>Lupa Password</i>	2	0	2	0%
4	Menu	5	5	0	100%
5	Pencarian	3	3	0	100%
6	Tambahkan Produk Toko	4	4	0	100%
7	<i>Edit Produk Toko</i>	4	0	4	0%
8	Melihat Produk untuk <i>user</i>	2	2	0	100%
9	Memesan Produk untuk <i>user</i>	2	1	1	50%
10	Melihat Pesanan untuk <i>user</i>	2	2	0	100%
11	Menambahkan Alamat untuk <i>user</i>	3	2	1	66,6%
12	Menghapus Pesanan untuk <i>user</i>	2	2	0	100%
13	Melihat Transaksi untuk penjual	2	2	0	100%
14	Mengunggah Bukti Pembayaran untuk <i>user</i>	2	0	2	0%
15	Mengecek Pesanan untuk penjual	2	2	0	100%
16	Verifikasi Bukti Pembayaran untuk Admin	2	0	2	0%
Total		45	33	12	73,33%

4.2.2 Tindak Lanjut

Setelah melakukan pengujian pertama, terdapat beberapa *test case* yang berstatus *fail* sehingga akan dilakukan perbaikan pada fitur tersebut. Setelah fitur tersebut diperbaiki, akan dilakukan pengujian lagi. Sebagai batasan penelitian kali ini, fitur yang *fail* akan diperbaiki berdasarkan tingkat prioritas tertinggi (1). Fitur-fitur *test case* yang berstatus *fail* seperti disajikan pada Tabel 4.34.

Tabel 4.34 Daftar *test case* yang *fail*

Test Case ID	Ringkasan Scenario Test Case	Status	Prioritas (1-3)
TC3.1	Memasukkan <i>email</i> yang sudah terdaftar pada aplikasi Ivent di halaman lupa <i>password</i> untuk mendapatkan <i>link reset password</i>	<i>Fail</i>	1
TC3.2	Memasukkan <i>email</i> yang belum terdaftar di aplikasi Ivent yang bertujuan untuk mengecek apakah sistem tetap mengirimkan <i>link reset password</i> atau tidak	<i>Fail</i>	1
TC7.1	Mengubah informasi pada suatu produk dengan mengisi semua <i>field</i> pada halaman <i>edit produk</i>	<i>Fail</i>	1

TC7.2	Mengubah informasi pada suatu produk dengan mengisi <i>field</i> harga produk dengan format kata untuk mengecek apakah ada notifikasi yang muncul bahwa format pada <i>field</i> harga yang diinputkan salah.	<i>Fail</i>	2
TC7.3	Mengubah informasi pada suatu produk dengan mengisi <i>field</i> stok produk dengan format kata untuk mengecek apakah ada notifikasi yang muncul bahwa format pada <i>field</i> stok produk yang diinputkan salah.	<i>Fail</i>	2
TC7.4	Mengubah informasi pada suatu produk dengan mengosongkan salah satu <i>field</i> untuk mengecek apakah muncul notifikasi bahwa ada <i>field</i> yang kosong / notifikasi bahwa semua <i>field</i> harus diisi.	<i>Fail</i>	2
TC9.2	Melakukan pemesanan dengan tanggal mulai tahun 2022 dan tanggal selesai tahun 2000 untuk mengecek apakah muncul notifikasi bahwa tanggal selesai tidak boleh lebih kecil dari tanggal mulai dan tidak dapat memasukkan produk tersebut ke pesanan.	<i>Fail</i>	1
TC11.2	Menambahkan alamat pemesan dengan mengisi <i>field</i> nomor telepon dengan format kata untuk mengecek apakah muncul notifikasi bahwa format yang diinputkan pada <i>field</i> nomor telepon tidak sesuai.	<i>Fail</i>	1
TC14.1	Mengunggah bukti pembayaran pada salah satu transaksi	<i>Fail</i>	1
TC14.2	Mengunggah bukti pembayaran pada salah satu transaksi dengan file yang berekstensi selain .pdf, .jpg, .png untuk mengecek apakah dapat mengunggah bukti pembayaran dengan format tersebut atau tidak	<i>Fail</i>	1
TC16.1	Melihat bukti pembayaran yang sudah diunggah oleh <i>user</i>	<i>Fail</i>	1
TC16.2	Memverifikasi salah satu bukti pembayaran yang telah diunggah oleh <i>user</i>	<i>Fail</i>	1

Keterangan:

1. Prioritas 1: Merupakan prioritas *High*, *bug* yang terjadi pada fitur tersebut harus segera diperbaiki.
2. Prioritas 2: Merupakan prioritas *Medium*, prioritas ini satu tingkatan di bawah prioritas 1, *bug* yang terjadi pada fitur tersebut dapat dipertimbangkan apakah harus segera diperbaiki atau tidak.
3. Prioritas 3: Merupakan prioritas *Low*, prioritas ini berada pada tingkatan paling bawah, *bug* yang terjadi pada fitur ini tidak harus segera diperbaiki.

Tindak lanjut fitur lupa *password* TC3.1 dan TC3.2

Setelah berdiskusi langsung dengan pihak pengembang aplikasi Ivent, pada fitur *lupa password* yang memiliki prioritas (1) tidak dilakukan perbaikan karena kode pada fitur *lupa password* masih dalam tahap pengembangan dan belum diselesaikan oleh pihak pengembang aplikasi Ivent.

4.2.3 Hasil Pengujian Kedua

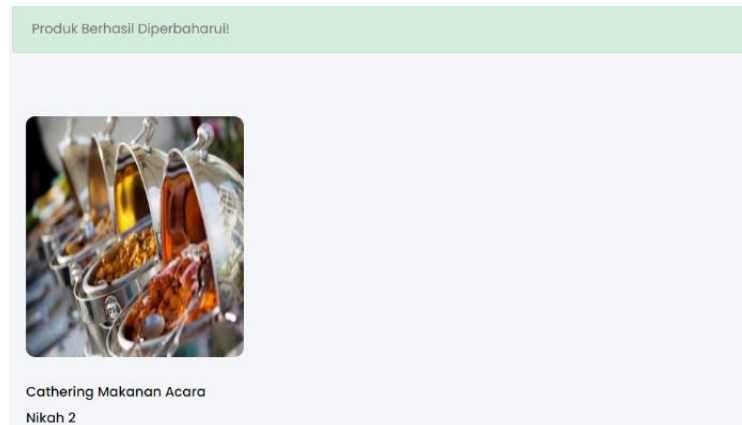
Pengujian kedua dilakukan setelah perbaikan kepada fitur yang mengalami *error* pada pengujian pertama. Perbaikan dilakukan berdasarkan hasil diskusi dengan pihak pengembang aplikasi Ivent dan tingkat prioritas *test case* pada Tabel 4.34. Hasil pengujian kedua disajikan dalam Tabel 4.35 .

Tabel 4.35 Hasil pengujian kedua

Fitur	<i>Test Case ID</i>	<i>Actual Result</i>	Status
<i>Edit Produk</i>	TC7.1	Berhasil mengubah produk dan berpindah ke halaman produk	<i>pass</i>
Memesan Produk	TC9.2	Tetap pada halaman detail produk dan muncul notifikasi bahwa format yang diinputkan salah	<i>pass</i>
Menambahkan Alamat Pemesanan	TC11.2	Tetap pada halaman detail produk dan muncul notifikasi bahwa format yang diinputkan salah.	<i>pass</i>
Mengunggah Bukti Pembayaran	TC14.1	Berhasil melakukan pembayaran dan mendapatkan bukti pembayaran	<i>pass</i>
	TC14.2	Gagal mengunggah bukti pembayaran karena file ekstensi tidak sesuai format	<i>pass</i>
Verifikasi Bukti Pembayaran	TC16.1	Menampilkan transaksi <i>user</i> beserta bukti pembayaran yang sudah diunggah	<i>pass</i>
	TC16.2	Menampilkan transaksi <i>user</i> yang dengan bukti pembayaran yang sudah diverifikasi dan status pembayaran yang berubah menjadi 'sudah dibayar'	<i>pass</i>

Pengujian kedua fitur *edit produk* TC7.1

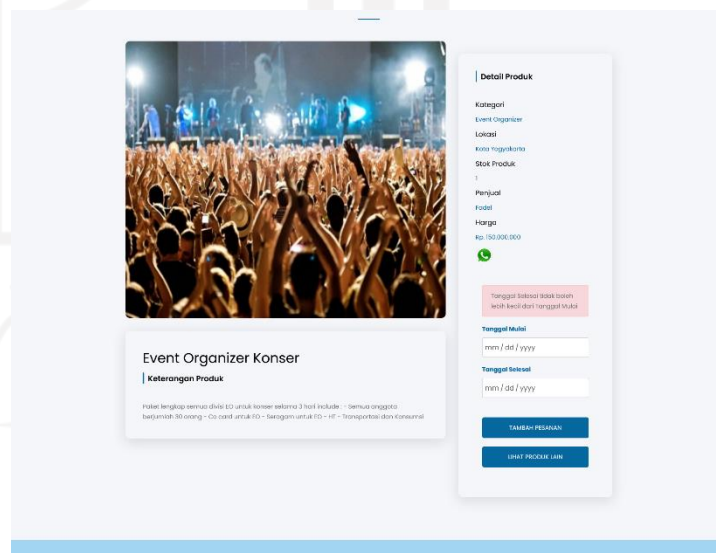
Perbaikan yang dilakukan pada fitur *edit produk* TC7.1 yang memiliki prioritas 1 juga berdampak pada TC7.2 - TC7.4 yang memiliki prioritas 2. Setelah dilakukan perbaikan, TC7.2 - TC7.4 dapat dijalankan. Pengujian fitur *edit produk* mendapatkan hasil *pass* dari total 4 *test case* yang dijalankan. Rata-rata waktu yang diperlukan adalah 20,25 detik. Ketika klik tombol '*edit produk*', berpindah ke halaman *edit produk* dan dapat menjalankan 4 *test case* pada fitur ini. Hasil pengujian pada fitur ini dapat dilihat pada Gambar 4.5 dibandingkan dengan pengujian sebelumnya pada Gambar 4.12.



Gambar 4.12 Hasil pengujian kedua fitur *edit* produk

Pengujian kedua fitur memesan produk TC9.2

Pengujian fitur memesan produk pada TC9.2 mendapatkan hasil *pass*. Rata-rata waktu yang diperlukan adalah 17 detik. Setelah dilakukan perbaikan, *user* tidak dapat memesan produk apabila tanggal selesai < tanggal mulai dan akan muncul *alert* bahwa format yang diinputkan salah atau tidak sesuai. Hasil pengujian pada fitur ini dapat dilihat pada Gambar 4.13 dibandingkan dengan pengujian sebelumnya pada Gambar 4.6.



Gambar 4.13 Hasil pengujian kedua fitur memesan produk

Pengujian kedua fitur menambahkan alamat TC11.2

Pengujian fitur menambahkan alamat pemesan pada TC11.2 mendapatkan hasil *pass*. Waktu yang diperlukan adalah 28 detik. Setelah dilakukan perbaikan, *user* tidak dapat

menginputkan kata pada *field* nomor telepon. Hasil pengujian pada fitur ini dapat dilihat pada Gambar 4.14 dibandingkan dengan pengujian sebelumnya pada Gambar 4.7.

Form Alamat Pengiriman

The no tip must be at least 8.

The no tip may not be greater than 12.

Nama Penerima	Kota
Fadel	Yogyakarta
Alamat	Kecamatan
Perumahan	Umbulharjo
No Tlp	Kelurahan
XXXXXXXXXXXX	Giwangan
Provinsi	Kodepos
DI	55163

Gambar 4.14 Hasil pengujian kedua fitur menambahkan alamat

Pengujian kedua fitur mengunggah bukti pembayaran TC14.1-TC14.2

Pengujian yang dilakukan pada fitur ini mendapatkan hasil pass dari total 2 *test case* yang dijalankan. Rata-rata waktu yang dibutuhkan adalah 8 detik. Setelah dilakukan perbaikan, *user* dapat menambahkan bukti pembayaran dengan ekstensi .jpg, .jpeg, .png dan .pdf. Jika *user* menambahkan bukti pembayaran selain ekstensi yang sudah ditetapkan, maka akan muncul *alert* bahwa hanya dapat menambahkan sesuai ekstensi yang sudah ditetapkan. Hasil pengujian pada fitur ini dapat dilihat pada Gambar 4.15 dibandingkan dengan pengujian sebelumnya pada Gambar 4.9.

Pembayaran

127.0.0.1:8000 says
Please upload file having extensions .jpg/.jpeg/.png/.pdf only.

Daftar Pesanan				Ringkasan	
No	Nama Produk	Harga Per Hari	Jumlah Hari	Total	
1	Paket Lampu Ponggung	Rp. 4.000.000	1	Rp. 4.000.000	Status Pembayaran: belum
Total:				Rp. 4.000.000	Status Pengiriman: belum

Alamat Pengiriman

Nama Penerima	Alamat Penerima	Nomor Telepon
Fadel	Perumahan aadama cIB	satuduatuaseamat

Upload Bukti Pembayaran

Choose file... Browse

KIRIM

Gambar 4.15 Hasil pengujian kedua fitur mengunggah bukti pembayaran

Pengujian kedua fitur verifikasi bukti pembayaran oleh admin TC16.1-TC16.2

Perbaikan yang dilakukan pada fitur mengunggah bukti pembayaran TC14.1 - TC14.2 juga berdampak pada fitur verifikasi bukti pembayaran TC16.1 - TC16.2. Setelah dilakukan

perbaikan, *user* dapat mengunggah bukti pembayaran dan bukti tersebut akan muncul ke halaman *dashboard* admin dan dapat dilakukan verifikasi bukti pembayaran. Pengujian yang dilakukan pada fitur ini mendapatkan hasil pass dari total 2 *test case* yang dijalankan. Rata-rata waktu yang dibutuhkan adalah 10 detik. Setelah dilakukan perbaikan pada fitur mengunggah bukti pembayaran, bukti pembayaran yang sudah diunggah dapat ditampilkan dan dilakukan verifikasi pada halaman *dashboard* admin. Hasil pengujian pada fitur ini dapat dilihat pada Gambar 4.16 dibandingkan dengan pengujian sebelumnya pada Gambar 4.11.

The screenshot displays the Ivent Admin Dashboard. On the left is a dark sidebar with 'Ivent' at the top and 'DASHBOARD' and 'Transaksi' below. The main content area is light blue and contains several sections:

- Item:** A table with columns 'No', 'Nama Produk', 'Penjual', and 'Harga Satuan'. It lists one item: 'Event Organizer Korser' by 'Fadel' for '150.000.000'. A 'Total' row shows '150.000.000'. A red 'Kembali' button is below.
- Ringkasan:** A summary box showing 'Total: 150.000.000', 'Status Pembayaran: belum', and 'Status Pengiriman: belum'.
- Alamat Pengiriman:** A table with columns 'Nama Penerima', 'Alamat', and 'No tlp'. It lists 'Fadel' with address 'Perumahan agestama c18, Slewang, Limbuharjo, Yogyakarta, Di Yogyakarta - 55163' and phone '08131711999'.
- Verifikasi Pembayaran:** A box with a blue 'Verifikasi Pembayaran' button.
- Bukti Pembayaran:** A BNI transaction receipt showing 'Transaksi Berhasil' with details like account numbers and amounts.

Gambar 4.16 Hasil pengujian kedua fitur verifikasi bukti pembayaran

Total 10 *test case fail* yang diperbaiki seperti pada Tabel 4.34, kemudian dilakukan pengujian kedua untuk mendapatkan hasil dari fitur yang sudah diperbaiki. Kesimpulan dari keseluruhan pengujian yang dilakukan pada aplikasi Ivent disajikan pada Tabel 4.36. Dari total 16 fitur yang diuji, 15 fitur mendapatkan 100% presentasi keberhasilan, 1 fitur mendapatkan 0% presentasi keberhasilan. Fitur yang mendapatkan 0% keberhasilan adalah fitur lupa *password*. Setelah berdiskusi dengan pihak pengembang aplikasi Ivent, fitur lupa *password* masih dalam tahap pengembangan dan belum selesai sehingga tidak dilakukan perbaikan.

Tabel 4.36 Ringkasan pengujian keseluruhan

No	Tabel Fitur	Jumlah <i>Test Case</i>	Jumlah Berhasil	Jumlah Gagal	Presentasi Berhasil
1	<i>Register</i>	4	4	0	100%
2	<i>Login</i>	4	4	0	100%
3	<i>Lupa Password</i>	2	0	2	0%
4	Menu	5	5	0	100%
5	Pencarian	3	3	0	100%
6	Tambahkan Produk Toko	4	4	0	100%
7	<i>Edit Produk Toko</i>	4	4	0	100%
8	Melihat Produk untuk <i>user</i>	2	2	0	100%
9	Memesan Produk untuk <i>user</i>	2	2	0	100%
10	Melihat Pesanan untuk <i>user</i>	2	2	0	100%
11	Menambahkan Alamat untuk <i>user</i>	3	3	0	100%
12	Menghapus Pesanan untuk <i>user</i>	2	2	0	100%
13	Melihat Transaksi untuk penjual	2	2	0	100%
14	Mengunggah Bukti Pembayaran untuk <i>user</i>	2	2	0	100%
15	Mengecek Pesanan untuk penjual	2	2	0	100%
16	Verifikasi Bukti Pembayaran untuk Admin	2	2	0	100%
Total		45	43	2	95,56%

Kualitas perangkat lunak pada aplikasi Ivent meningkat setelah dilakukan pengujian kedua, lebih banyak *test case* yang mendapatkan hasil *pass*. Peningkatan kualitas pada aplikasi Ivent dapat dilihat pada Tabel 4.33 dibandingkan dengan Tabel 4.35. Presentasi keberhasilan sebelum dilakukan perbaikan sebesar 73,33% dan setelah perbaikan, peningkatan presentasi keberhasilan menjadi 95,56%. Tabel 4.36 menunjukkan bahwa metode yang digunakan pada pengujian aplikasi Ivent dapat mengukur kesesuaian antara aplikasi Ivent dengan *software requirement* yang sudah ditentukan.

4.2.4 Waktu Pengujian

Pengujian dilakukan dengan menggunakan *test case template* kemudian waktu yang dibutuhkan untuk menjalankan setiap *test case* dicatat. Waktu hasil dari pengujian yang dilakukan dapat dilihat pada Gambar 4.17. Pada fitur melihat pesanan *user* dengan 2 *test case* yang dijalankan mendapat hasil rata-rata waktu 3 detik dan pada fitur menambahkan pesanan dengan 3 *test case* yang dijalankan mendapat hasil rata-rata waktu 28 detik.

No	Fitur	Test Case ID	Waktu	Rata-rata	NO	Fitur	Test Case ID	Waktu	Rata-rata
1	Register	TC1.1	15	14.75	9	Memesan Produk untuk user	TC9.1	17	17
		TC1.2	16				TC9.2	17	
		TC1.3	12		10	Melihat Pesanan untuk user	TC10.1	3	3
		TC1.4	16				TC10.2	3	
2	Login	TC2.1	6	5.5	11	Menambahkan Alamat untuk user	TC11.1	28	28
		TC2.2	5				TC11.2	28	
		TC2.3	5				TC11.3	28	
		TC2.4	6		12	Menghapus Pesanan untuk user	TC12.1	7	7
TC3.1		TC12.2	7						
3	Lupa Password	TC3.2		7.67	13	Melihat Transaksi untuk penjual	TC13.1	8	8.5
		TC4.1	5				TC13.2	9	
4	Menu	TC4.2	5	4.2	14	Mengunggah bukti pembayaran	TC14.1	8	8
		TC4.3	3				TC14.2	8	
		TC4.4	4		15	Mengecek pesanan untuk penjual	TC15.1	10	10
		TC4.5	4				TC15.2	10	
		TC5.1	8				16	Verifikasi bukti pembayaran oleh admin	
TC5.2	8	TC16.2	10						
6	Menambahkan Produk Toko	TC5.3	7	23.25	7	Edit Produk Toko	TC6.2	21	20.25
		TC6.3	22				TC7.1	21	
		TC6.4	22				TC7.2	20	
		TC6.5	28				TC7.3	20	
7	Melihat produk untuk user	TC7.4	20	3.5	8	Melihat produk untuk user	TC8.1	4	3
		TC8.2	3				TC8.2	3	

Gambar 4.17 Waktu pengujian setiap *test case*

4.3 Closure

Proses *quality assurance* pada aplikasi Ivent menggunakan *test case template* telah selesai dilakukan. Proses *quality assurance* menggunakan *test case template* membantu pengujian dalam mencari *error* sehingga dapat dicatat dan diperbaiki. Pada pengujian pertama, 33 *test case* mendapatkan hasil *pass* dan 12 *test case* mendapatkan hasil *fail* dari total 45 *test case* yang dijalankan. Pada pengujian kedua, dari total 45 *test case* yang dijalankan, 43 *test case* mendapatkan hasil *pass* dan 2 *test case* masih mendapatkan hasil *fail*. Presentasi hasil keberhasilan pada pengujian pertama sebesar 73,33% seperti pada Tabel 4.33. Setelah memperbaiki *error* yang ditemukan, kemudian dilakukan pengujian kedua dan mendapatkan presentasi keberhasilan sebesar 95,56% seperti pada Tabel 4.36. *Test case* yang *fail* adalah TC3.1 dan TC3.2 pada fitur lupa *password*.

Perbaikan *test case* yang *fail* pada hasil pengujian pertama dilakukan setelah menyerahkan bukti hasil pengujian pertama dan mendapatkan izin dari pengembang aplikasi Ivent yang dapat dilihat pada LAMPIRAN A. Waktu pengujian setiap *test case* yang dijalankan terdapat pada LAMPIRAN B. *Template* pengujian dan dokumen pengujian dapat dilihat pada LAMPIRAN C. Hasil dan ringkasan pengujian aplikasi Ivent dapat dilihat pada LAMPIRAN D.

4.4 Kontribusi Penelitian

Penerapan *test case template* dalam pengujian perangkat lunak dapat menjamin bahwa semua proses bisnis yang ada dalam aplikasi tersebut dapat teruji tanpa ada satupun yang terlewatkan, karena pengujian perangkat lunak menggunakan *test case template* sendiri lebih berfokus pada *test case* yang dibuat untuk setiap fungsionalitas berdasarkan *software requirements* dari aplikasi yang diuji dan dapat mengetahui fitur-fitur yang masih *error* dengan detail berdasarkan hasil pengujian yang dilakukan. Pada implementasinya *test case template* dilakukan dengan membandingkan antara hasil yang sebenarnya terjadi dan hasil yang diharapkan. Hal tersebut akan mempermudah tim pengembang menemukan dan memperbaiki kesalahan yang terdapat pada perangkat lunak.



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan dari kegiatan penelitian implementasi penggunaan *test case template* dalam proses *quality assurance* pada aplikasi *e-commerce* berbasis *website* Ivent seperti berikut:

- a. Pengujian pada aplikasi Ivent dalam proses *quality assurance* menggunakan *test case template* telah berhasil dilakukan.
- b. Proses *quality assurance* dengan menggunakan *test case template* dilakukan dengan cara menganalisis *software requirements* pada aplikasi Ivent, menentukan *environment* dan rencana pengujian, membuat *test case* sesuai dengan *test case template* yang digunakan, melakukan pengujian menggunakan *test case* yang sudah dibuat dan mengevaluasi terhadap seluruh tahapan yang sudah dilakukan.
- c. Proses *quality assurance* dapat mengetahui detail *software*, modul, fitur yang akan dilakukan pengujian, mendapatkan hasil dari pengujian yang dilakukan berupa *bug* pada fitur yang diuji dan waktu yang diperlukan selama pengujian.
- d. Proses *quality assurance* dengan menggunakan *test case template* dapat menjadi solusi untuk mengetahui apakah aplikasi Ivent sudah berjalan sesuai dengan *software requirements* yang sudah ada.
- e. Pengujian dilakukan pada 16 fitur aplikasi Ivent. 15 fitur telah berjalan sesuai dengan *software requirements* yang ada.
- f. Terdapat 1 fitur yang masih terjadi *error* karena kode pada fitur tersebut masih belum diselesaikan oleh pihak pengembang aplikasi Ivent.
- g. Total 46 *test case* yang digunakan dalam pengujian. 43 *test case* mendapatkan hasil *pass* dan 2 *test case* masih mendapatkan hasil *fail*.
- h. Proses *quality assurance* pada aplikasi Ivent dengan menggunakan *test case template* membantu pengujian dalam mencari *bug* yang belum ditemukan sebelumnya.
- i. Manfaat penggunaan *test case template* untuk mempermudah penguji dalam pembuatan *test case* pada setiap fitur yang akan diuji, *test case* yang dibuat lebih mudah dipahami, dibaca oleh pihak lain dan terlihat lebih rapi.

5.2 Saran

Peningkatan jaminan mutu pada aplikasi Ivent setelah dilakukan pengujian menggunakan *test case template* sudah meningkat namun belum maksimal. Masih terdapat beberapa kendala seperti pada fitur lupa *password* yang tidak dapat diperbaiki karena kode pada fitur tersebut belum diselesaikan oleh pihak pengembang aplikasi Ivent. Berdasarkan dari keseluruhan pengujian pada aplikasi Ivent, maka pemberian saran dimaksudkan kepada pihak pengembang aplikasi Ivent dan kepada peneliti selanjutnya. Kepada pihak pengembang aplikasi Ivent, aplikasi dapat dikembangkan lebih lanjut, seperti menyelesaikan fitur lupa *password* karena fitur tersebut penting bagi *user*, menambahkan fitur untuk mengubah alamat pemesan yang sudah ada, menghapus alamat pemesan, dan menambahkan fitur untuk pengiriman pesan antara *user* dan penjual.

Kemudian pada peneliti selanjutnya, perlu untuk lebih mengkaji penelitian maupun referensi yang berkaitan dengan *quality assurance* agar penelitian yang akan dilakukan selanjutnya lebih baik dan lebih terperinci. Peneliti selanjutnya juga perlu mengkaji referensi terkait *automated testing* beserta *tools* yang mempermudah proses *quality assurance* seperti Qase, Katalon Studio, Selenium, Cypress.io dan lain sebagainya untuk lebih mempermudah pengujian dan mengurangi *human errors* yang terjadi. Selain itu, perlu juga dilakukan pengujian non-fungsional untuk mengukur akurasi, keamanan, daya tahan, kebenaran dan lain sebagainya.

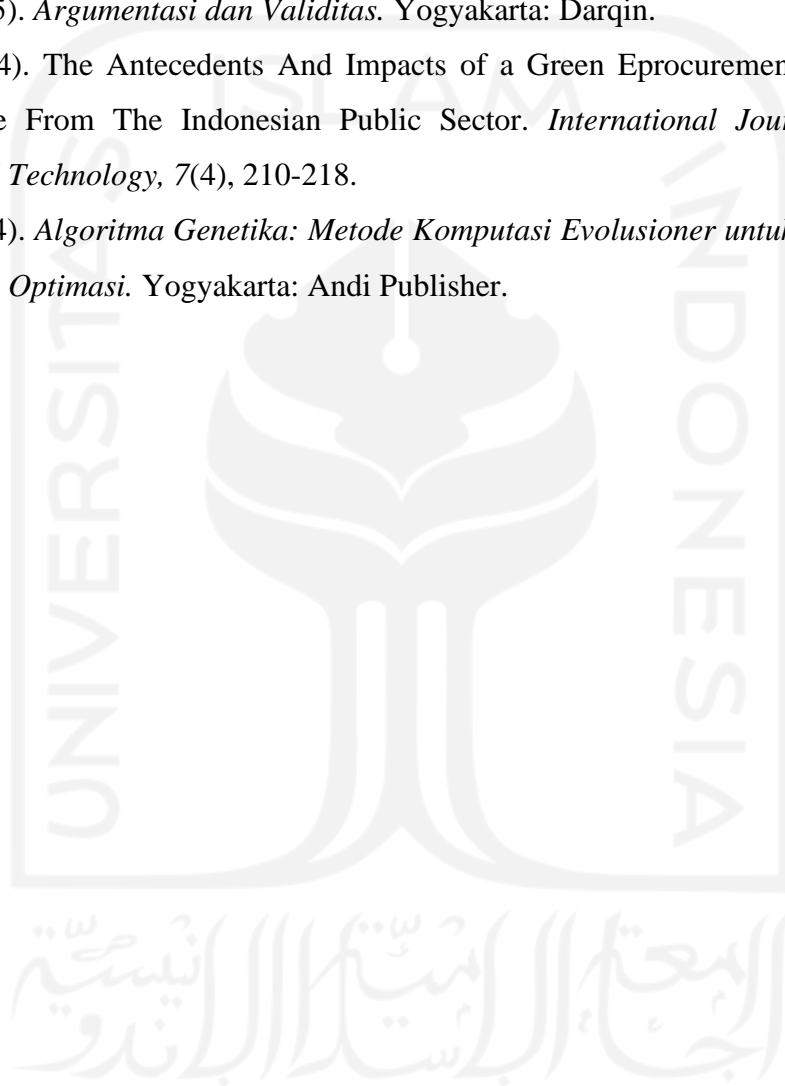
DAFTAR PUSTAKA

- Afifah, V., & Setyantoro, D. (2021). Rancangan Sistem Pemilihan dan Penetapan Harga dalam Proses Pengadaan Barang dan Jasa Logistik Berbasis Web. *Jurnal IKRA-ITH INFORMATIKA*, 5(2), 108–117.
- Anjali, G. (2014). *E-COMMERCE : ROLE OF E-COMMERCE IN TODAY'S BUSINESS*. 4(1).
- Anwar, M. A. H., & Kurniawan, Y. (2019). Dokumentasi Software Testing Berstandar Ieee 829-2008 Untuk Sistem Informasi Terintegrasi Universitas. *Kurawal - Jurnal Teknologi, Informasi Dan Industri*, 2(2), 118–125. <https://doi.org/10.33479/kurawal.2019.2.2.118-125>
- Bahaweres, R. B., Zawawi, K., Khairani, D., & Hakiem, N. (2017). Analysis of statement branch and loop coverage in software testing with genetic algorithm. *International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), 2017-Decem*(September), 19–21. <https://doi.org/10.1109/EECSI.2017.8239088>
- Chernak, Y. (2001). *software testing Test-Case Effectiveness*. 81–86.
- Cholifah, W. N., Yulianingsih, Y., & Sagita, S. M. (2018). Pengujian Black Box Testing pada Aplikasi Action & Strategy Berbasis Android dengan Teknologi Phonegap. *STRING (Satuan Tulisan Riset Dan Inovasi Teknologi)*, 3(2), 206. <https://doi.org/10.30998/string.v3i2.3048>
- Dani, D. N. (2021). *Strategi Ide Bisnis Startup E-Commerce Ivent Menggunakan Metode Design Thinking*. <https://dspace.uui.ac.id/handle/123456789/36011>
- Dee, S. (2022). *How To Automate Repetitive QA Task Using Excel Macros*. <https://www.softwaretestinghelp.com/automate-repetitive-qa-tasks-using-excel-macros/>
- Dwi Juniansyah, B., Redy Susanto, E., & Deni Wahyudi, A. (2020). Pembuatan E-Commerce Pemesanan Jasa Event Organizer Untuk Zero Seven Entertainment. *Jurnal Tekno Kompak*, 14(1), 41–46. <https://ejurnal.teknokrat.ac.id/index.php/teknokompak/article/view/499>
- Firmansyah, A. (2018). Kajian Kendala Implementasi E-Commerce Di Indonesia. *Masyarakat Telematika Dan Informasi : Jurnal Penelitian Teknologi Informasi Dan Komunikasi*, 8(2), 127. <https://doi.org/10.17933/mti.v8i2.107>
- Hariyanto, Teduh Dirgahayu, & Hanson Prihantoro P. (2020). Software Quality Assurance pada Perusahaan Pengembang Perangkat Lunak Skala Kecil dan Menengah. *JARTIKA Jurnal Riset Teknologi Dan Inovasi Pendidikan*, 3(2), 283–294. <https://doi.org/10.36765/jartika.v3i2.265>

- Hasibuan, A. N., & Dirgahayu, T. (2021). Pengujian dengan Unit Testing dan Test case pada Proyek Pengembangan Modul Manajemen Pengguna. *Automata*, 2(1). <https://journal.uui.ac.id/AUTOMATA/article/view/17367>
- Hosal, V., Angriani, H., & Muawwal, A. (2021). Implementasi Software Testing Dalam Quality Assurance Pada Learning Management System Website Classes. *KHARISMA Tech*, 16(2), 156–168.
- Idris, M. A. (2021). *PERANCANGAN USER INTERFACE & USER EXPERIENCE PADA APLIKASI E-COMMERCE IVENT MENGGUNAKAN METODE USER CENTERED DESIGN (UCD) PERANCANGAN UI / UX PADA APLIKASI BERBASIS WEBSITE E-COMMERCE NT DENGAN METODE USER CENTERED DESIGN PERANCANGAN USER INTERFACE & US.*
- IEEE. (1990). *IEEE_SoftwareEngGlossary.pdf* (p. 84).
- IEEE. (2012). *IEEE Standard for System and Software Verification and Validation, in IEEE Std 1012-2012 (Revision of IEEE Std 1012-2004)*. vol., no., <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6204026&isnumber=6204025>
- Ihsan, S. (2022). *Test case for few common modules of website testing*. https://www.linkedin.com/posts/subhan-ihsan_website-testing-activity-6935247660878565376-bPW_?utm_source=linkedin_share&utm_medium=member_desktop_web
- Irawan, Y. (2017). Pengujian Sistem Informasi Pengelolaan Pelatihan Kerja UPT BLK Kabupaten Kudus dengan Metode Whitebox Testing. *Sentra Penelitian Engineering Dan Edukasi*, 9(3), 59–63.
- István, M., & Simon, J. (2009). The IPSI BgD Transactions on Internet Research. *Ipsi J Tir*, 6(2), 35–38.
- Jamali, N. (2020). *QUALITY ASSURANCE TESTING APLIKASI HUMAN RESOURCE SPLENDID BERBASIS WEBSITE PADA PT. GARASILABS MANIVESTA*. 21(1), 1–45.
- Kaner, C. (2003). What Is a Good Test Case? *Software Testing Analysis & Review Conference (STAR East)*, 1–16.
- Mamone, S. (2000). Documentation testing. *ACM SIGSOFT Software Engineering Notes*, 25(2), 26–29. <https://doi.org/10.1145/346057.346066>
- Mucjal, A. A. (2021). *PENGEMBANGAN E-COMMERCE VENDOR DAN EVENT ORGANIZER BERBASIS WEBSITE DENGAN METODE WATERFALL.*

- Mustaqbal, M. S., Firdaus, R. F., & Rahmadi, H. (2015). *PENGUJIAN APLIKASI MENGGUNAKAN BLACK BOX TESTING BOUNDARY VALUE ANALYSIS (Studi Kasus : Aplikasi Prediksi Kelulusan SNMPTN)*. I(3), 31–36.
- Neumeyer, A. (2022). *A Proven Test Plan Template for Software Testing (Excel)*. <https://www.tacticalprojectmanager.com/test-case-template-excel-with-example/>
- Novianingsih, E. (2022). *Perancangan Pengujian Perangkat Lunak Pada*.
- Pamungkas, D. W. L., & Rochimah, S. (2019). Pengujian Aplikasi Web - Tinjauan Pustaka Sistematis. *Jurnal IPTEK*, 23(1), 17–24. <https://doi.org/10.31284/j.ipitek.2019.v23i1.459>
- Parveen, T., Tilley, S., & Gonzalez, G. (2007). A case study in test management. *Proceedings of the Annual Southeast Conference, 2007*, 82–87. <https://doi.org/10.1145/1233341.1233357>
- Petrie, H. (2006). *Test case management tools for accessibility testing*. T Cdl.
- Rahayu, D. K. P. (2020). Unit Testing Pada Aplikasi Web Mobile (Studi Kasus Bisnis Jasa Laundry). *Skripsi*. <https://dspace.uui.ac.id/handle/123456789/28341>
- Ramadhan, G., Kusumaningtyas, R. H., & Hasan Taufik, A. S. (2019). Analisis Pengaruh Biaya Terhadap Loyalitas Pengguna Mobile Banking Jenius Menggunakan Model Neural Networks. *Applied Information System and Management (AISM)*, 2(2). <https://doi.org/10.15408/aism.v2i2.20163>
- Romindo. (2019). *E-Commerce: Implementasi, Strategi dan Inovasinya*. [https://books.google.co.id/books?hl=en&lr=&id=PirGDwAAQBAJ&oi=fnd&pg=PR5&dq=\(Romindo,+2019\).&ots=OxZOu03dmL&sig=a6LKO0tXUafHJv7-2RIUiirpSM&redir_esc=y#v=onepage&q=\(Romindo%2C%202019\).&f=false](https://books.google.co.id/books?hl=en&lr=&id=PirGDwAAQBAJ&oi=fnd&pg=PR5&dq=(Romindo,+2019).&ots=OxZOu03dmL&sig=a6LKO0tXUafHJv7-2RIUiirpSM&redir_esc=y#v=onepage&q=(Romindo%2C%202019).&f=false)
- Sebastianus, A. (2021). *PENGUJIAN QUALITY ASSURANCE APLIKASI INVENTORI BARANG BERBASIS WEBSITE PADA PT PETROKIMIA GRESIK*. 6.
- Softwaretestinghelp. (2022). *Sample Test Case Template With Test Case Examples*. <https://www.softwaretestinghelp.com/test-case-template-examples/>
- The Institute of Electrical and Electronic Engineers, I. (2017). *Standard for System, Software and Hardware Verification and Validation - Corrigendum*. <https://standards.ieee.org/standard/1012-2016-Cor1-2017.html>
- Yuana, K. C. (2021). *STANDAR IEEE-829 DAN IEEE-1012*.
- Yunisa, R., & Putro, H. P. (2019). Perbandingan 2 Teknik White Box Testing: Statement Coverage dan Branch Coverage Testing. *The 5th Annual Research Seminar, 5.1*(1), 197–206.

- Hendrik, Anjomshooa, A., & Tjoa, A. M. (2014). Towards Semantic Mashup Tools For Big Data Analysis. *Proceeding of the Information & Communication Technology-EurAsia Conference 2014*, (pp. 100-145). Bali.
- Setiawan, A. M. (2013). *Integrated Framework For Business Process Complexity Analysis*. Retrieved from ECIS 2013 Completed Research: http://aisel.aisnet.org/ecis2013_cr/49
- Taufiq, H. (2015). *Argumentasi dan Validitas*. Yogyakarta: Darqin.
- Wahid, F. (2014). The Antecedents And Impacts of a Green Eprocurement Infrastructure: Evidence From The Indonesian Public Sector. *International Journal of internet Protocol Technology*, 7(4), 210-218.
- Zukhri, Z. (2014). *Algoritma Genetika: Metode Komputasi Evolusioner untuk Menyelesaikan Masalah Optimasi*. Yogyakarta: Andi Publisher.



LAMPIRAN

LAMPIRAN A. PERMOHONAN IZIN UNTUK MENGGUNAKAN APLIKASI IVENT

SURAT PERMOHONAN IZIN

Kepada : Yth. Bapak/Ibu Pengembang Aplikasi Ivent

Dengan hormat, saya yang bertanda tangan dibawah ini:

Nama : Fadel Muhammad

NIM : 17523187

Jurusan : Informatika

Dengan ini mengajukan Permohonan Izin untuk menggunakan, melakukan pengujian dan melakukan perbaikan pada aplikasi *e-commerce* berbasis *website* Ivent dalam penelitian yang berjudul "*Penggunaan Test Case Template dalam Proses Quality Assurance*"

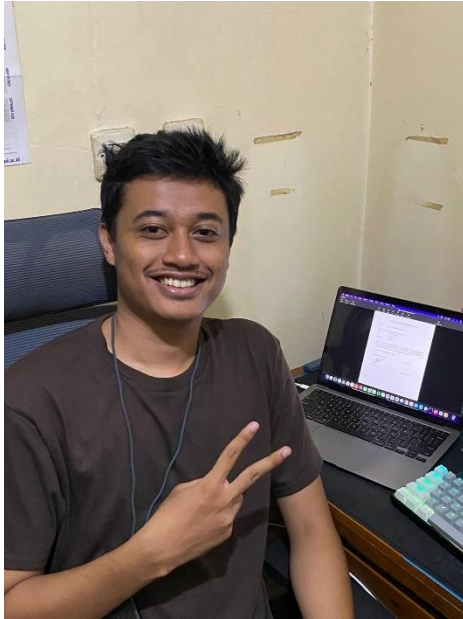
Demikian surat ini disampaikan, Terima kasih.

Menyetujui / ~~Tidak Menyetujui~~
Pengembang Aplikasi Ivent,


Aikal/Alfrian Mucjal

Hormat Saya,


Fadel Muhammad



LAMPIRAN B. WAKTU HASIL PENGUJIAN SETIAP *TEST CASE*

(Link file:

<https://docs.google.com/spreadsheets/d/1Oen2Hez3lh36mRcHAGheGLHSQoHXn5-p/edit?usp=sharing&ouid=108479432665439606335&rtfpof=true&sd=true>)

Waktu Pengujian Yang didapat Pada Setiap *Test Case*

No	Fitur	Test Case ID	Waktu	Rata-rata
1	Register	TC1.1	15	14.75
		TC1.2	16	
		TC1.3	12	
		TC1.4	16	
2	Login	TC2.1	6	5.5
		TC2.2	5	
		TC2.3	5	
		TC2.4	6	
3	Lupa Password	TC3.1		
		TC3.2		
4	Menu	TC4.1	5	4.2
		TC4.2	5	
		TC4.3	3	
		TC4.4	4	
		TC4.5	4	
5	Pencarian	TC5.1	8	7.67
		TC5.2	8	
		TC5.3	7	
6	Menambahkan Produk Toko	TC6.2	21	23.25
		TC6.3	22	
		TC6.4	22	
		TC6.5	28	
		TC6.1	21	
7	Edit Produk Toko	TC7.2	20	20.25
		TC7.3	20	
		TC7.4	20	
		TC7.1	20	
8	Melihat produk untuk user	TC8.1	4	3.5
		TC8.2	3	
9	Memesan Produk untuk user	TC9.1	17	17
		TC9.2	17	
10	Melihat Pesanan untuk user	TC10.1	3	3
		TC10.2	3	
11	Menambahkan Alamat untuk user	TC11.1	28	28
		TC11.2	28	
		TC11.3	28	
12	Menghapus Pesanan untuk user	TC12.1	7	7
		TC12.2	7	
13	Melihat Transaksi untuk penjual	TC13.1	8	8.5
		TC13.2	9	
		TC13.3	9	
14	Mengunggah bukti pembayaran	TC14.1	8	8
		TC14.2	8	
15	Mengecek pesanan untuk penjual	TC15.1	10	10
		TC15.2	10	
16	Verifikasi bukti pembayaran oleh admin	TC16.1	10	10
		TC16.2	10	

LAMPIRAN C. TEMPLATE SETIAP TEST CASE EXCEL

(Link file:

https://docs.google.com/spreadsheets/d/1qwHp6mPa0knjABrD5QtApuFo6u2_QHSY/edit?usp=sharing&oid=108479432665439606335&rtpof=true&sd=true)

Test Case ID	Test Case Description	Test Condition	Test Steps	Test Data	Expected Result
TC1.1	Salah satu kolom ada yang kosong	Sudah berada di halaman Register user	1. Klik field email 2. Masukkan email dengan format yang sesuai 3. Klik field password 4. Masukkan password yang ingin digunakan 5. Klik field konfirmasi password 6. Masukkan password yang sama 7. Klik tombol register	Email: cohansani1@gmail.com Password: 12345678 Konfirmasi password: 12345678	Tetap berada di halaman register dan muncul alert karena salah satu field belum terisi
TC1.2	Menasukkan konfirmasi password yang tidak sesuai	Sudah berada di halaman Register user	1. Klik field nama lengkap 2. Masukkan nama lengkap dengan format yang sesuai 3. Klik field email 4. Masukkan email dengan format yang sesuai 5. Klik field password 6. Masukkan password yang ingin digunakan 7. Klik field konfirmasi password 8. Masukkan password yang berbeda 9. Klik tombol register	Nama lengkap: Tester QA Email: cohansani1@gmail.com Password: 12345678 Konfirmasi password: 123456789	Tetap berada di halaman register dan muncul alert karena password dan konfirmasi password dimasukkan tidak
TC1.3	Menasukkan email yang sudah pernah terdaftar pada sistem	Sudah berada di halaman Register user	1. Klik field nama lengkap 2. Masukkan nama lengkap dengan format yang sesuai 3. Klik field email 4. Masukkan email dengan format yang sesuai 5. Klik field password 6. Masukkan password yang ingin digunakan 7. Klik field konfirmasi password 8. Masukkan password yang berbeda 9. Klik tombol register	Nama lengkap: Tester QA Email: cohansani1@gmail.com Password: 12345678	Tetap berada di halaman register dan muncul alert karena email yang dimasukkan sudah pernah terdaftar pada
TC1.4	Mendaftarkan diri sebagai user di sistem	Sudah berada di halaman Register user	1. Klik field nama lengkap 2. Masukkan nama lengkap dengan format yang sesuai 3. Klik field email 4. Masukkan email dengan format yang sesuai 5. Klik field password 6. Masukkan password yang ingin digunakan 7. Klik field konfirmasi password 8. Masukkan password yang sama 9. Klik tombol register	Nama lengkap: Tester QA Email: cohansani1@gmail.com Password: 12345678	Berhasil mendaftar sebagai user dan berpindah ke homepage Event

LAMPIRAN D. HASIL DAN RINGKASAN PENGUJIAN

(Link file:

https://docs.google.com/spreadsheets/d/1heXo7oMqkztsCIzNDvq1_E4LGOHnOwnX/edit?usp=sharing&oid=108479432665439606335&rtpof=true&sd=true)

Fitur	Test Case ID	Actual Result	Status
Register	TC1.1	Tetap berada pada halaman register dan muncul alert bahwa salah satu field belum diisi	Pass
	TC1.2	Tetap berada di halaman register dan muncul alert karena password dan konfirmasi password dimasukkan tidak sama	Pass
	TC1.3	Tetap berada di halaman register dan muncul alert karena email yang dimasukkan sudah pernah terdaftar pada sistem	Pass
	TC1.4	Berhasil mendaftar sebagai user dan berpindah ke homepage Event	Pass
Login	TC2.1	Berhasil login dan berpindah halaman ke homepage	Pass
	TC2.2	Tetap berada di login page dan muncul alert karena salah satu field belum terisi	Pass
	TC2.3	Tetap berada di login page dan muncul alert karena salah satu field belum terisi	Pass
Lupa Password	TC3.1	Muncul notifikasi bahwa tidak dapat mengirim pesan tanpa alamat email pengirim.	Fail
	TC3.2		Fail
Menu	TC4.1	Berpindah dari homepage ke halaman pencarian produk	Pass
	TC4.2	Berpindah dari homepage ke halaman buka toko	Pass
	TC4.3	Berpindah dari homepage ke daftar pesanan / keranjang	Pass
	TC4.4	User keluar dari sistem	Pass
	TC4.5	Berpindah dari homepage ke halaman akun user	Pass
Pencarian	TC5.1	Sistem menampilkan produk berdasarkan kata kunci yang dimasukkan	Pass
	TC5.2	Sistem menampilkan produk berdasarkan kata kunci dan kategori yang dipilih	Pass
	TC5.3	Sistem menampilkan produk dengan nama yang mendekati kata kunci yang dimasukkan	Pass
	TC5.4	Tidak dapat menambahkan gambar dengan cara drag & drop	Fail
Tambahkan Produk	TC6.2	Tidak dapat menambahkan produk dan muncul pesan bahwa salah satu field kosong (field nama produk)	Pass
	TC6.3	Tidak dapat menambahkan produk dan muncul pesan bahwa field harga produk harus diisi dengan angka	Pass
	TC6.4	Tidak dapat menambahkan produk dan muncul pesan bahwa harus menambahkan gambar produk	Pass