

MANAJEMEN PENGUJIAN PERANGKAT LUNAK MENGUNAKAN QASE IO



Disusun Oleh:

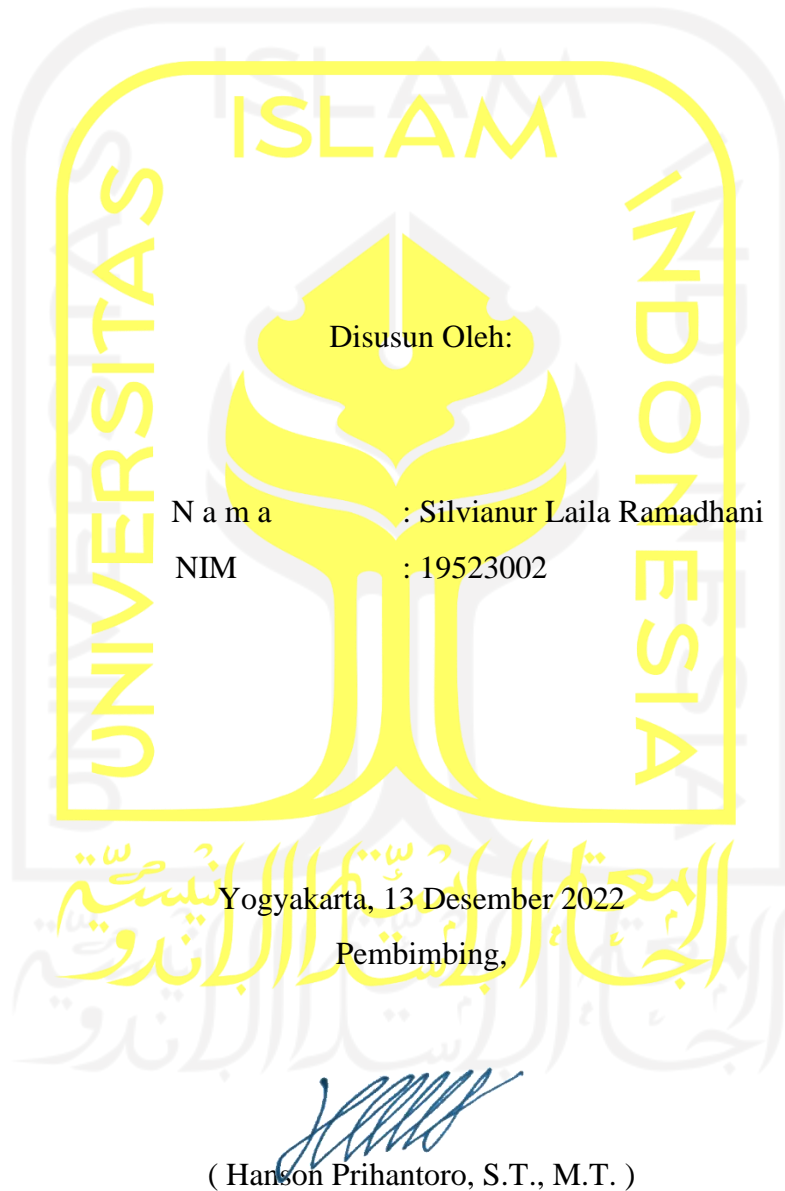
N a m a : Silvianur Laila Ramadhani
NIM : 19523002

**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
2023**

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**MANAJEMEN PENGUJIAN PERANGKAT LUNAK
MENGUNAKAN QASE IO**

TUGAS AKHIR



HALAMAN PENGESAHAN DOSEN PENGUJI

**MANAJEMEN PENGUJIAN PERANGKAT LUNAK
MENGUNAKAN QASE IO****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, 6 Januari 2023

Tim Penguji

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

Anggota 1

Kholid Haryono, S.T., M.Kom.

Anggota 2

Sri Mulyati, S.Kom., M.Kom.

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 : Silvianur Laila Ramadhani

NIM : 19523002

Tugas akhir dengan judul:

**MANAJEMEN PENGUJIAN PERANGKAT LUNAK
MENGUNAKAN QASE IO**

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, 13 Desember 2022



(Silvianur Laila Ramadhani)

المكتبة الإسلامية الإلكترونية

HALAMAN PERSEMBAHAN

Alhamdulillah rabbil'alamin, puji syukur kusembahkan kepadaMu ya Allah, Tuhan yang Maha Pengasih dan Penyayang. Segala syukur kuucapkan kepadaMu Ya Rabb, karena sudah menghadirkan orang-orang berarti yang selalu mendukungku. Ku persembahkan tugas akhir ini untuk kedua orang tua dan adik kesayanganku yang selalu memberikan dukungan agar selalu semangat dalam menjalani perkuliahan.



HALAMAN MOTO

“Sesungguhnya Bersama kesulitan itu ada kemudahan, Karena itu bila kau telah selesai (dari suatu urusan), kerjakanlah dengan sungguh-sungguh (urusan yang lain)”
{QS. Al- Insyirah: 6-7}

"Tujuan pendidikan itu untuk mempertajam kecerdasan, memperkuat kemauan serta memperhalus perasaan" - Tan Malaka

“Barang siapa keluar untuk mencari sebuah ilmu, maka ia akan berada di jalan Allah hingga ia kembali.” – HR Tirmidzi



KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Alhamdulillah, segala puji syukur saya panjatkan kehadiran Allah subhanahu wata'ala, yang telah memberikan ridho dan rahmatnya. Sehingga penulis dapat menyelesaikan pembuatan Tugas Akhir yang berjudul “Manajemen Pengujian Perangkat Lunak menggunakan Qase IO” dengan baik.

Tujuan Tugas Akhir ini dibuat sebagai salah satu syarat memperoleh gelar sarjana Informatika di Universitas Islam Indonesia. Tanpa adanya doa, dukungan, dan bimbingan dari berbagai pihak, penelitian ini tidak akan selesai.

Oleh karena itu penulis mengucapkan terimakasih sebanyak-banyaknya kepada:

1. Allah SWT yang telah memberikan kesehatan, rahmat, dan hidayahnya sehingga penulis dapat menyelesaikan Tugas Akhir ini.
2. Kedua orang tua dan adik tercinta yang selalu memberi motivasi, doa, dan dukungan semangat untuk mengerjakan Tugas Akhir.
3. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc., selaku ketua Jurusan Informatika dan Dosen Pembimbing Akademik.
4. Bapak Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D., selaku ketua Prodi Informatika.
5. Bapak Hanson Prihantoro Putro, S.T, M.T., selaku Dosen Pembimbing yang sangat sabar dalam membimbing dan memberikan ilmu baru selama proses pembuatan Tugas Akhir.
6. Bapak Ibu dosen yang tidak dapat penulis sebutkan seluruhnya. Terimakasih telah memberikan banyak ilmu selama penulis berkuliah.
7. Seluruh developer aplikasi Ivent, terimakasih telah berkenan mengizinkan aplikasinya menjadi objek penelitian Tugas Akhir ini.
8. Sahabat-sahabat penulis terutama Adis, Rania, Jasmine, Pinta, Thalia, Chika, Elsa, Riko, Alafta, Dea dan lain-lain yang tidak dapat penulis sebutkan seluruhnya. Terimakasih atas pengalaman hidup di Jogja, tempat berkeluh kesah, dan kenangan indah selama kuliah.

Penulis menyadari masih banyak kekurangan dalam pembuatan Tugas Akhir ini, baik kendala dalam ilmu maupun dari pribadi penulis. Oleh karena itu mohon untuk kritik dan saran yang membangun agar penelitian ini dapat diteruskan dan berguna di masa mendatang. Sekali lagi terima kasih sebanyak-banyaknya.

Wassalamu'alaikum warahmatullahi wabarakatuh

Yogyakarta, 13 Desember 2022



(Silvanur Laila Ramadhani)

الْحَمْدُ لِلَّهِ الَّذِي

SARI

Pada proses pengembangan dilakukan peningkatan kualitasnya dengan menguji, menemukan kecacatan, dan memperbaiki kecacatan tersebut hingga produk dianggap baik untuk dirilis. Permasalahan kegagalan fitur selalu muncul ketika perusahaan telah merilis produk perangkat lunak tersebut. Selain itu banyaknya waktu yang dibutuhkan dalam melakukan pengujian, membuat perusahaan perlu mengeluarkan biaya lebih banyak dengan lamanya estimasi waktu pengerjaan. Oleh karena itu perlu dilakukan manajemen dalam pengujian agar terstruktur dan dapat menghemat waktu. Manajemen pengujian perangkat lunak memiliki bagian penting untuk membantu mengatur proses pengujian perangkat lunak sehingga terstruktur dan sesuai dengan rencana. Bagian penting pada manajemen pengujian yaitu bagian perencanaan dan bagian eksekusi. Qase merupakan aplikasi manajemen pengujian berbasis *cloud* yang dapat diakses secara gratis dan menjadi alat bantu dalam manajemen pengujian perangkat lunak sebelum dirilis. Qase memiliki beberapa fitur pendukung manajemen pengujian yaitu *test case management*, *test plans and runs*, *integration*, dan *team management*. Qase dipilih sebagai alat bantu manajemen pengujian karena dapat menyimpan defect dari *test case* yang mengalami kegagalan. Selain itu, Qase juga membuat data pengujian lebih terstruktur dan mudah dipahami oleh pihak lain. Kasus yang dipilih adalah aplikasi Ivent, dimana aplikasi Ivent sendiri merupakan *e-commerce* berbasis web yang dapat menghubungkan antara pemilik usaha dengan pencari *event organizer* atau vendor. Pengujian aplikasi Ivent dilakukan berdasarkan 11 *usecase* yang digunakan untuk membuat 28 *test case*. Manfaat yang didapat yaitu Qase mampu membantu dalam mengatur dan mengelola pengujian agar lebih terstruktur.

Kata kunci: Manajemen pengujian perangkat lunak, Blackbox testing, Qase, e-Commerce, Aplikasi Ivent.

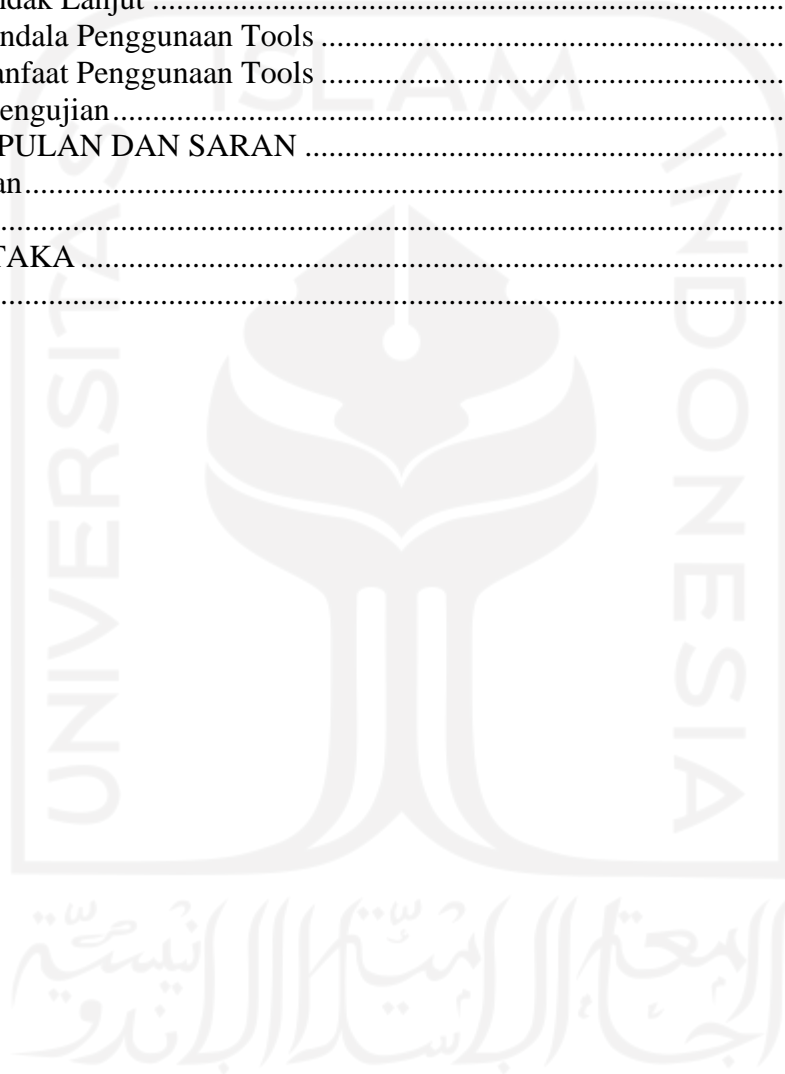
GLOSARIUM

<i>Blackbox</i>	Metode pengujian yang sering disebut pengujian fungsionalitas, karena hanya menguji fitur-fitur fungsional pada perangkat lunak.
<i>Developer</i>	Pihak pengembang atau yang bertugas untuk menciptakan suatu perangkat lunak sesuai dengan kebutuhan pengguna dan menganalisis persyaratan serta menentukan rancangan perangkat lunak.
<i>Test Case</i>	Sebuah masukan, kondisi, dan ekspektasi hasil yang digunakan untuk menguji sebuah perangkat lunak.
<i>Test Plan</i>	Dokumen hasil perencanaan yang berisi sumber daya, jadwal pengujian, dan tujuan atau sasaran pengujian dalam lingkup proyek pengujian. Mengidentifikasi fitur yang akan diuji, pembagian tugas dan lain sebagainya.
<i>Test Suite</i>	Gabungan serangkaian <i>test case</i> yang digunakan untuk membantu penguji dalam melaksanakan pengujian pada suatu fitur yang sama.
<i>Test Run</i>	Fitur pada Qase untuk melakukan serangkaian pengujian pada kasus uji.

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.....	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Batasan	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat.....	3
1.6 Metode Penelitian.....	4
1.7 Sistematika Laporan	4
BAB II KAJIAN PUSTAKA.....	6
2.1 E-commerce.....	6
2.2 Ivent.....	6
2.3 Pengujian Perangkat Lunak	7
2.4 Manajemen Pengujian Perangkat Lunak	8
2.4.1 Analisis Risiko dan Pengujian (<i>Risk Analysis and Testing</i>)	9
2.4.2 Perencanaan dan Estimasi pengujian (<i>Test Planning and Estimation</i>)	10
2.4.3 Pendekatan Pengujian (<i>Test Approches/Strategies</i>)	10
2.4.4 Organisasi Pengujian (<i>Test Organisation</i>)	11
2.4.5 Pemantauan dan Pengendalian Pengujian (<i>Monitoring and Controlling Test</i>) ..	12
2.4.6 Manajemen Insiden (<i>Incident Management</i>).....	12
2.4.7 Laporan Pengujian (<i>Test Report</i>).....	13
2.5 Alat Pengujian Perangkat Lunak	14
2.5.1 Alat Manajemen Pengujian	14
2.5.2 Qase Tools	15
BAB III ANALISIS DAN PERENCANAAN.....	20
3.1 Analisis Risiko dan Pengujian (<i>Risk and Testing Analysis</i>).....	20
3.2 Perencanaan dan Estimasi Pengujian (<i>Test Planning and Estimation</i>)	22
3.3 Pendekatan Pengujian (<i>Test Approches/Strategies</i>)	28
BAB IV ORGANISASI, PEMANTAUAN DAN PENGENDALIAN	30
4.1 Organisasi Pengujian (<i>Test Organisation</i>)	30
4.1.1 Persiapan.....	30
4.1.2 Pengujian	31
4.1.3 Fitur Tambahan.....	36
4.2 Pemantauan dan Pengendalian Pengujian (<i>Monitoring and Controlling Test</i>).....	38
4.2.1 Fitur Registrasi	39
4.2.2 Fitur Login.....	39

4.2.3	Fitur Tambah Produk.....	40
4.2.4	Fitur Edit Produk	41
4.2.5	Fitur Melihat Produk	42
4.2.6	Fitur Membuat Pesanan	43
4.2.7	Fitur Melihat Pesanan	44
4.2.8	Fitur Melihat Transaksi	45
4.2.9	Fitur Melakukan Pembayaran.....	45
4.2.10	Fitur Memeriksa Status Pembayaran	46
4.2.11	Fitur Verifikasi Pembayaran.....	47
4.3	Manajemen Insiden	48
4.3.1	Tindak Lanjut	48
4.3.2	Kendala Penggunaan Tools	50
4.3.3	Manfaat Penggunaan Tools	50
4.4	Laporan Pengujian.....	52
BAB V KESIMPULAN DAN SARAN		59
5.1	Kesimpulan.....	59
5.2	Saran.....	60
DAFTAR PUSTAKA		61
LAMPIRAN.....		64



DAFTAR TABEL

Tabel 3.1 Skenario Pengujian Aplikasi Ivent	23
Tabel 4.1 Hasil pengujian fitur <i>register</i>	39
Tabel 4.2 Hasil Pengujian Fitur Login.....	40
Tabel 4.3 Hasil pengujian fitur tambah produk	41
Tabel 4.4 Hasil pengujian fitur edit produk	41
Tabel 4.5 Hasil pengujian fitur melihat produk	42
Tabel 4.6 Hasil pengujian fitur membuat pesanan.....	43
Tabel 4.7 Hasil pengujian fitur melihat pesanan	44
Tabel 4.8 Hasil pengujian fitur melihat transaksi	45
Tabel 4.9 Hasil pengujian fitur melakukan pembayaran	46
Tabel 4.10 Hasil pengujian fitur memeriksa status pembayaran	46
Tabel 4.11 Hasil pengujian fitur verifikasi pembayaran.....	47
Tabel 4.12 Hasil keseluruhan pengujian <i>test case</i>	48
Tabel 4.13 Daftar <i>test case</i> berstatus <i>fail</i>	49
Tabel 4.14 Kelebihan dan Kekurangan Qase.....	51
Tabel 4.15 Hasil pengujian kedua.....	52
Tabel 4.16 Rangkuman hasil keseluruhan pengujian	56

DAFTAR GAMBAR

Gambar 2.1 Proses Manajemen Pengujian	9
Gambar 2.2 Fitur <i>Test Case Management</i>	16
Gambar 2.3 Fitur <i>test plan</i>	16
Gambar 2.4 Fitur <i>test run</i>	17
Gambar 2.5 Fitur <i>team management</i>	17
Gambar 2.6 Fitur <i>invite new member</i>	17
Gambar 2.7 Fitur <i>integration</i>	18
Gambar 2.8 Fitur <i>import test cases</i>	18
Gambar 2.9 Fitur <i>export test cases</i>	19
Gambar 3.1 <i>Use Case Diagram</i> dari Ivent.....	22
Gambar 4.1 Halaman beranda Qase.....	30
Gambar 4.2 Halaman <i>login</i> Qase	31
Gambar 4.3 Halaman daftar akun Qase	31
Gambar 4.4 <i>Form project</i> baru	32
Gambar 4.5 Halaman <i>dashboard project</i>	32
Gambar 4.6 Halaman <i>repository</i>	32
Gambar 4.7 Halaman membuat <i>test case</i>	33
Gambar 4.8 <i>Form detail test plan</i>	33
Gambar 4.9 Halaman <i>test plan</i>	34
Gambar 4.10 <i>Form detail test run</i>	34
Gambar 4.11 Halaman eksekusi <i>test case</i>	34
Gambar 4.12 <i>Test run</i> dengan status <i>fail</i>	35
Gambar 4.13 <i>Test run</i> dengan status <i>pass</i>	35
Gambar 4.14 Halaman hasil <i>test run</i>	36
Gambar 4.15 Fitur <i>export data</i> pada <i>test plan</i>	36
Gambar 4.16 Mengundang anggota baru.....	37
Gambar 4.17 Halaman <i>workspace</i> anggota	37
Gambar 4.18 <i>Form</i> membuat <i>milestone</i>	38
Gambar 4.19 Halaman <i>milestones</i>	38
Gambar 4.20 Hasil pengujian fitur <i>register</i>	39
Gambar 4.21 Hasil pengujian fitur <i>login</i>	40
Gambar 4.22 Hasil pengujian fitur tambah produk	41

Gambar 4.23 Hasil pengujian fitur edit produk	42
Gambar 4.24 Hasil pengujian fitur melihat produk	43
Gambar 4.25 Hasil pengujian fitur membuat pesanan.....	44
Gambar 4.26 Hasil pengujian fitur melihat pesanan.....	44
Gambar 4.27 Hasil pengujian fitur melihat transaksi	45
Gambar 4.28 Hasil pengujian fitur melakukan pembayaran	46
Gambar 4.29 Hasil pengujian fitur memeriksa status pembayaran	47
Gambar 4.30 Hasil pengujian fitur verifikasi pembayaran	47
Gambar 4.31 <i>Upgrade</i> langganan Qase	50
Gambar 4.32 Hasil pengujian TC 4.1 fitur edit produk	52
Gambar 4.33 <i>Defect</i> TC 4.1 pada Qase.....	53
Gambar 4.34 Hasil pengujian TC 4.2 fitur edit produk	53
Gambar 4.35 <i>Defect</i> TC 4.2 pada Qase.....	54
Gambar 4.36 Hasil pengujian TC 7.1 fitur melihat pesanan.....	54
Gambar 4.37 <i>Defect</i> TC 7.1 pada Qase.....	55
Gambar 4.38 Hasil pengujian TC 7.2 fitur melihat pesanan.....	55
Gambar 4.39 <i>Defect</i> TC 7.2 pada Qase.....	56

BAB I PENDAHULUAN

1.1 Latar Belakang

Penjaminan kualitas perangkat lunak dapat dilakukan dengan pengujian oleh *Quality Assurance* (QA). Tujuan dari QA adalah menghadirkan perangkat lunak yang minim cacat, memenuhi tingkatan fungsi, keandalan, dan kinerja yang telah ditentukan (Ioan Mihnea Iacob, 2008). QA biasanya terikat dengan pengujian perangkat lunak atau dikenal dengan *software testing* yang dilakukan oleh *software tester*. Menciptakan sebuah perangkat lunak yang handal dan berkualitas diperlukan pengujian perangkat lunak, di mana pengujian perangkat lunak secara singkat adalah proses menjalankan program perangkat lunak untuk menemukan kesalahan (Hariyanto, 2020).

Pengujian perangkat lunak adalah proses verifikasi untuk melakukan penilaian dan peningkatan kualitas perangkat lunak. Selama pengembangan dilakukan peningkatan kualitas produk perangkat lunak dengan melakukan pengujian-menemukan kecacatan-memperbaiki kecacatan, hingga dinilai bahwa produk telah baik untuk dirilis (Kshirasagar Naik, 2008). Pengujian juga bertujuan untuk menunjukkan kesesuaian fungsi-fungsi perangkat lunak terhadap spesifikasinya. Perangkat lunak dinyatakan gagal apabila tidak memenuhi spesifikasi yang telah ditentukan (Fournier, 2008).

Pada setiap perusahaan yang mengembangkan perangkat lunak, diharuskan untuk dapat meningkatkan kualitas produk perangkat lunak yang dimiliki sebelum digunakan oleh pengguna. Permasalahan sering terjadi ketika perusahaan telah meluncurkan perangkat lunak kepada pengguna. Kurang maksimalnya fitur yang dimiliki perangkat lunak membuat kualitas yang dimiliki oleh produk menurun. Dengan demikian, diperlukan evaluasi terhadap kualitas produk kualitas yang bermasalah (Kartiko, 2019). Pengujian perangkat lunak menjadi hal terpenting dalam proses pengembangan. Akan tetapi, biaya yang dikeluarkan untuk melakukan pengujian cukup besar yakni sekitar 20% hingga 80% selama siklus hidup perangkat lunak (Hesham Alaqail, 2018).

Besarnya biaya pengujian perangkat lunak membuat perusahaan perlu melakukan pengelolaan manajemen pengujian dengan baik untuk menghindari *over budget* pada anggaran pengembangan perangkat lunak. Manajemen pengujian perangkat lunak memiliki beberapa bagian penting untuk membantu mengatur segala proses pengujian perangkat lunak agar terstruktur dan sesuai dengan rencana. Bagian yang ada pada manajemen pengujian

yaitu bagian perencanaan yang terdapat organisasi pengujian, perencanaan dan estimasi pengujian, serta analisis risiko dan pengujian. Kemudian terdapat bagian eksekusi yang terdiri dari manajemen konfigurasi, monitoring dan kontroling pengujian, manajemen permasalahan atau insiden (Brian Hambling, 2010).

Terdapat alat bantu (*tools*) yang dapat digunakan dalam pengujian. Alat tersebut diklasifikasikan berdasarkan aktivitas pengujian yang dilakukan, misalnya adalah alat untuk mendukung manajemen pengujian perangkat lunak, alat untuk mendukung pengujian otomatis, dan lain sebagainya. Beberapa alat dapat melakukan fungsi yang spesifik (khusus) dan terbatas, tetapi banyak juga alat yang dapat melakukan beberapa fungsi berbeda secara bersamaan. Dalam permasalahan manajemen pengujian maka diperlukan alat manajemen pengujian perangkat lunak yang biasanya mampu melakukan pengelolaan pengujian, manajemen insiden, manajemen persyaratan, dan lain sebagainya (Dorothy Graham, 2006).

Salah satu alat bantu Qase dapat digunakan dalam manajemen pengujian perangkat lunak sebelum diluncurkan kepada pengguna. Qase adalah sistem manajemen kasus uji (*test case management*) berbasis cloud yang digunakan oleh pengembang dan penguji (Rajkumar, 2022). Qase memiliki tampilan UI dan UX yang mudah digunakan untuk pemula dalam manajemen pengujian perangkat lunak. Selain itu Qase dapat diakses secara gratis melalui *website* dengan melakukan pendaftaran akun terlebih dahulu. Dengan kelebihan fitur yang dimiliki Qase mampu membantu QA untuk mengikuti kemajuan perangkat lunak yang ada, terutama dalam menangani permasalahan menyimpan dan tracking bug.

Pada penelitian ini akan dilakukan manajemen terkait proses pengujian perangkat lunak yang dilakukan pada sebuah pengembangan aplikasi. Aplikasi yang dipilih untuk dilakukan manajemen pengujian adalah aplikasi *e-commerce* Ivent. Ivent merupakan *website e-commerce* dengan model bisnis yang menghubungkan antara pemilik bisnis dengan pencari *event organizer* atau vendor. Aplikasi Ivent ada disebabkan permasalahan ekonomi pada sektor industri kreatif yang mengalami kemunduran dan kesusahan dalam memasarkan produk akibat Covid19 (Mucjal, 2021).

Adanya aplikasi Ivent memudahkan pemilik bisnis bidang *event organizer* atau vendor acara untuk menawarkan jasa maupun produk yang mereka miliki kepada penyewa melalui internet. Penyewa *event organizer* atau vendor lebih mudah menentukan pilihan jasa atau produk mana yang akan dipilih karena banyaknya pilihan yang ditawarkan. Terlebih aplikasi ini memanfaatkan adanya kemajuan teknologi untuk memasarkan produknya secara luas.

Oleh karena itu harapannya aplikasi Ivent dapat dikembangkan dan diluncurkan untuk mempermudah kebutuhan banyak orang.

Proses manajemen dilakukan dengan menggunakan alat bantu Qase yang mampu membuat perencanaan pengujian, baik ditambahkan secara manual maupun otomatis. Pengerjaan pada Qase dapat dilakukan bersama tim dengan mengundang tim ke dalam *workspace project* yang dapat diberi label sesuai dengan kebutuhan. Apabila nanti ditemukan *bug* pada proses pengujian, Qase akan menyimpan semua *bug* atau *defect* untuk diserahkan kepada developer. Sehingga diharapkan dengan menggunakan alat manajemen Qase dapat membantu penulis untuk melakukan pengujian pada perangkat lunak agar kualitas perangkat lunak dapat lebih baik.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat berdasarkan latar belakang di atas adalah bagaimana melakukan manajemen pengujian perangkat lunak menggunakan alat Qase?

1.3 Batasan

Sesuai dengan judul yang dibuat, Qase dipilih sebagai *tools* yang digunakan untuk manajemen pengujian. Harapannya dapat dilakukan pada pengembangan perangkat lunak semua aplikasi. Namun sebagai kasus yang diangkat, *e-commerce* Ivent dipilih untuk menjadi aplikasi yang diuji dalam penelitian ini.

1.4 Tujuan Penelitian

Adapun tujuan yang ingin dicapai pada penelitian ini adalah melakukan manajemen pengujian menggunakan Qase pada proyek pengembangan perangkat lunak, terutama pada aplikasi *e-commerce* Ivent. Dengan manajemen ini akan diketahui *issue* pada perangkat lunak yang diuji yang perlu diselesaikan sebagai bentuk penjaminan mutu perangkat lunak (*software quality assurance*).

1.5 Manfaat

Manfaat yang didapat dari penelitian ini adalah:

- a. Mengetahui cara kerja, pengelolaan, dan proses pada alat manajemen pengujian Qase yang berbasis cloud untuk mengelola kasus pengujian.

- b. Membantu pengembang dalam mengetahui kesalahan apa saja yang ada pada perangkat lunak untuk dilakukan perbaikan.

1.6 Metode Penelitian

Pada penelitian ini pengujian yang dilakukan menggunakan metode manajemen pengujian (Brian Hambling, 2010):

- a. Analisis risiko dan Pengujian (*Risk Analysis and Testing*)
Tahap ini dilakukan analisis risiko apa saja yang dapat mempengaruhi pengujian perangkat lunak, seperti dampak atau kemungkinan yang akan terjadi.
- b. Perencanaan dan Estimasi Pengujian (*Test Planning and Estimation*)
Tahap yang mendefinisikan lingkup kegiatan pengujian dan batasan dari pengujian perangkat lunak.
- c. Pendekatan Pengujian (*Test Approaches/Strategies*)
Pada tahap ini desain pengujian dimulai dari penerapan strategi tentang bagaimana pengujian akan dilakukan.
- d. Organisasi Pengujian (*Test Organisation*)
Tahap ini berisikan skenario pengujian yang akan membantu meningkatkan efektivitas pengujian perangkat lunak.
- e. Pemantauan dan Kontrol Pengujian (*Monitoring and Controlling Test*)
Tahap untuk melakukan peninjauan terhadap perkembangan rencana pengujian dan perubahan apa saja yang terjadi terhadap risiko pengujian.
- f. Manajemen Insiden (*Incident Management*)
Tahap ini memuat cacat (*defect*), perubahan spesifikasi, atau permasalahan yang terjadi saat pengujian dilakukan.
- g. Laporan Pengujian (*Test Report*)
Tahap terakhir yang berisi kesimpulan dari seluruh metrik pengujian dan analisis informasi untuk mendukung rencana pengujian yang akan dilakukan kedepannya secara ringkas.

1.7 Sistematika Laporan

Sistematika penulisan pada laporan Tugas Akhir yang akan dijelaskan secara singkat sebagai berikut:

a. BAB I PENDAHULUAN

Bab ini berisi latar belakang penulisan penelitian, rumusan masalah, batasan dalam penelitian, tujuan dari penelitian, manfaat adanya penelitian, dan sistematika penulisan pada laporan penelitian yang digunakan menjadi Tugas Akhir.

b. BAB II KAJIAN PUSTAKA

Bab kajian pustaka memuat teori dan definisi yang terkait dengan penelitian baik dari buku, jurnal, maupun website. Selain itu bab ini juga berisi referensi artikel penulisan yang mengimplementasikan setiap sub bab.

c. BAB III ANALISIS DAN PERENCANAAN

Bab ini berisi tentang tahap analisis dan rencana berdasarkan metode yang digunakan dalam proses manajemen pengujian perangkat lunak. Terdapat beberapa tahapan untuk melakukan analisis dan perencanaan sebelum tahap eksekusi pengujian terhadap aplikasi Ivent.

d. BAB IV ORGANISASI, PEMANTAUAN DAN PENGENDALIAN

Pada bab ini menjelaskan tentang proses eksekusi dari pengujian yang telah dilakukan terhadap aplikasi Ivent, baik dari pengujian tampilan maupun pengujian proses penjalanan aplikasi. Selain itu juga berisi laporan hasil keseluruhan pengujian yang telah dilakukan.

e. BAB V KESIMPULAN DAN SARAN

Bab ini memuat kesimpulan yang didapatkan setelah menyelesaikan pengujian perangkat lunak. Selain itu juga berisi saran yang didapatkan untuk penelitian terkait pengujian perangkat lunak berikutnya.

BAB II

KAJIAN PUSTAKA

2.1 E-commerce

Perusahaan bisnis perlu memiliki sebuah situs web penjualan atau disebut *e-commerce*. *E-commerce* secara umum digambarkan sebagai proses jual beli barang atau jasa di *website* (Bayu Dwi Juniansyah, 2020). *E-commerce* adalah perangkat teknologi yang mencakup aplikasi dan proses bisnis dengan menghubungkan bisnis, konsumen, dan suatu komunitas melalui transaksi elektronik serta terjadi pertukaran barang secara elektronik (Handayani, 2018).

Ada tiga kategori utama *e-commerce* berdasarkan jenis pelakunya. Pertama, *Business to consumer* (B2C) *e-commerce* yang dilakukan oleh bisnis dan konsumen. Contoh dari B2C adalah Tiket.com, Berrybenka.com, dan lain lain. Kategori kedua, *Business to Business* (B2B) *e-commerce* yang melibatkan penjualan barang dan jasa antar bisnis. Contohnya adalah Bizzy dan Ralali. Lalu ketiga, *Consumer to Consumer* (C2C) *e-commerce*. *E-commerce* ini melibatkan konsumen yang menjual langsung kepada konsumennya. Contoh dari *e-commerce* ini adalah Tokopedia, Shopee, Bukalapak, dan lain lain (Sandra Ayu, 2020).

Event Organizer (EO) adalah perusahaan dalam bidang jasa yang secara resmi ditunjuk oleh pelanggan dalam penyelenggaraan rangkaian acara, mulai dari pembuatan konsep, perencanaan, persiapan, pelaksanaan dan akhir acara (Eka Chandra Ramdhani, 2018). Setiap kegiatan acara yang diselenggarakan bertujuan untuk memberikan manfaat bagi kedua belah pihak, baik penyelenggara dan maupun peserta selama acara berlangsung.

Pada pelaksanaanya, EO membutuhkan pihak lain untuk melaksanakan sebuah acara. Oleh karena itu dibutuhkanlah vendor acara untuk membantu dalam menyediakan alat dan peralatan terkait penyelenggaraan acara. Vendor adalah pihak, lembaga, badan usaha, atau perusahaan yang menyediakan barang dan jasa dalam jumlah besar yang dibutuhkan oleh perusahaan lain. (Vivi Afifah, 2021)

2.2 Ivent

Ivent adalah aplikasi *e-commerce* dengan memanfaatkan model *business to customer* (B2C) yang menghubungkan pemilik bisnis dengan konsumen. Konsumen disini adalah pihak yang mencari *event organizer* atau vendor untuk suatu acara. Ivent merupakan *e-commerce* yang menyediakan penyewaan barang atau jasa pada industri acara (Mucjal, 2021).

Ivent dikembangkan berbasis *website* yang nantinya dapat dengan mudah diakses melalui *browser* pada *desktop*. Namun, karena aplikasi Ivent masih pada versi 01 dan belum dilakukan *deployment* ke website sehingga tidak ada *url* secara umum yang dapat diakses orang lain. Sebelum dilakukan *deployment* ke *website*, aplikasi Ivent perlu dilakukan peningkatan kualitas dengan menguji fungsionalitas fitur yang telah diimplementasikan.

2.3 Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah proses verifikasi untuk melakukan penilaian dan peningkatan kualitas perangkat lunak. Pengujian memiliki peran penting dalam mencapai dan menilai kualitas dari suatu produk perangkat lunak. Selama pengembangan dilakukan peningkatan kualitas produk perangkat lunak dengan melakukan pengujian-menemukan kecacatan-memperbaiki kecacatan, hingga dinilai bahwa produk telah baik untuk dirilis (Kshirasagar Naik, 2008). Umumnya pengujian perangkat lunak dibagi menjadi dua kategori yaitu analisis statis dan analisis dinamis.

Analisis statis didasarkan pada pemeriksaan dokumen persyaratan, model perangkat lunak, dokumen desain dan kode sumber. Analisis statis tradisional termasuk tinjauan kode, inspeksi, penelusuran, dan analisis algoritma. Pengujian dilakukan untuk menilai kemungkinan-kemungkinan yang muncul selama waktu pengembangan berjalan.

Analisis dinamis dari sistem perangkat lunak melibatkan eksekusi program untuk menilai kemungkinan kegagalan program, sifat perilaku dan kinerja program juga diamati. Program dijalankan dengan nilai input yang khas dan dipilih dengan cermat. Lalu, proses dilanjutkan dengan mengamati beberapa perilaku program yang representatif dan mencapai kesimpulan tentang kualitas sistem. Pilihan yang tepat dari satu set pengujian sangat penting untuk mencapai kesimpulan yang andal.

Di dalam literatur tentang pengujian perangkat lunak, ditemukan referensi untuk istilah *failure* (kegagalan), *error* (kesalahan), *fault* (kesalahan), dan *defect* (kecacatan) yang memiliki kemiripan arti dengan tetap memiliki perbedaan khas masing-masing.

Failure : Kegagalan dikatakan terjadi ketika perilaku eksternal dari sistem tidak sesuai dengan yang ditentukan dalam spesifikasi sistem.

Error: Kesalahan adalah keadaan sistem dengan tidak adanya tindakan korektif, status kesalahan dapat menyebabkan kegagalan.

Fault: Kesalahan sistem yang merupakan penyebab adanya *error*.

Defect: Kecacatan adalah ketidaksempurnaan dari sebuah sistem, yang dapat mengurangi kualitas ataupun kegunaan sistem.

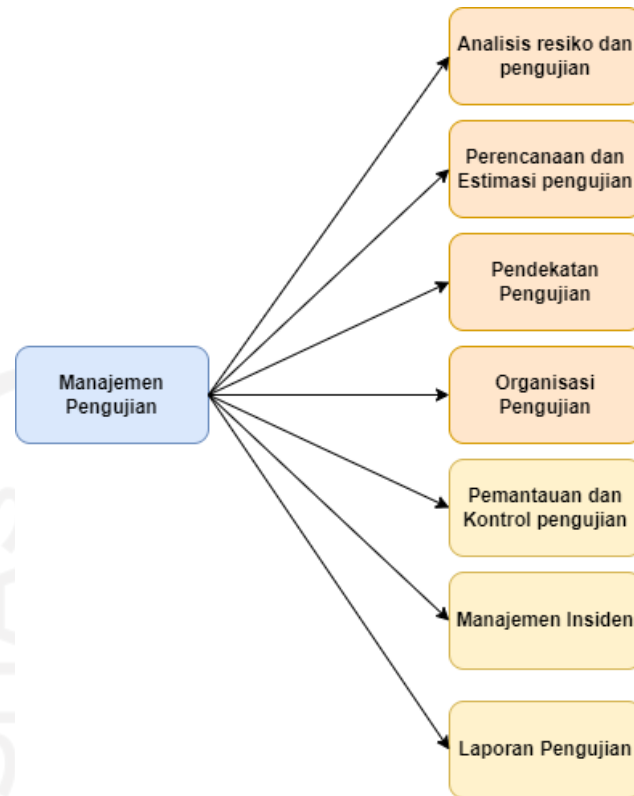
Pengujian perangkat lunak diperlakukan sebagai proses yang berbeda dari proses yang lain, karena pada pengujian perangkat lunak melibatkan berbagai aktivitas, teknik, strategi, dan kebijakan seperti:

1. Pengujian dilakukan untuk mencapai tujuan berikut:
 - a. Mengungkapkan kecacatan produk perangkat lunak.
 - b. Menunjukkan kualitas sistem perangkat lunak pada bagian keandalan, kinerja, stabilitas, dan skalabilitas.
2. Pengujian dimulai bersamaan dengan konsep pengerjaan produk mulai dikembangkan, dan pengujian berlangsung selama produk masih dalam proses pengembangan.
3. Pengujian dilakukan oleh orang yang berbeda-beda atau umumnya dikenal sebagai pengujian unit, pengujian integrasi, pengujian sistem, dan pengujian akhir.
4. Pengujian dipengaruhi oleh kebijakan yang dibuat seperti tingkat kualitas yang ingin dicapai, durasi pengujian, anggaran untuk pengujian.
5. Pengujian dapat dilakukan dengan kombinasi mode manual dan otomatis.

Adapun sebuah artikel (Abdulrahman Mahmoud, 2019) mengembangkan Minotaur sebagai peralatan baru yang digunakan untuk meningkatkan kecepatan dan akurasi analisis ketahanan (*endurance analysis*). Ide utama dibalik Minotaur adalah adanya kesamaan konseptual antara analisis ketahanan modern dengan pengujian perangkat lunak. Di dalam artikel dijelaskan bahwa pada pengujian perangkat lunak, kasus pengujian adalah input dan output yang diharapkan. Hal tersebut digunakan untuk menentukan apakah sistem yang diuji memenuhi beberapa tujuan pengujian perangkat lunak.

2.4 Manajemen Pengujian Perangkat Lunak

Manajemen pengujian perangkat lunak dibagi menjadi beberapa bagian penting dalam manajemen pengujian. Pembagian proses manajemen pengujian berupa bagian perencanaan yang terdapat organisasi pengujian, perencanaan dan estimasi pengujian, serta analisis risiko dan pengujian. Kemudian terdapat bagian eksekusi yang terdiri dari manajemen konfigurasi, monitoring dan kontroling pengujian, manajemen permasalahan atau insiden (Brian Hambling, 2010). Proses ini diperlihatkan pada Gambar 2.1.



Gambar 2.1 Proses Manajemen Pengujian

2.4.1 Analisis Risiko dan Pengujian (*Risk Analysis and Testing*)

Sebelum melakukan pengujian terlebih dahulu dilakukan analisis risiko apa saja yang dapat mempengaruhi proses pengujian perangkat lunak. Jika tidak ada risiko yang dianalisis, maka tidak akan ada proses pengujian perangkat lunak di masa mendatang. Risiko adalah faktor yang dapat mengakibatkan konsekuensi negatif di masa depan, biasanya dinyatakan sebagai dampak dan kemungkinan.

Risiko digunakan untuk memutuskan di mana harus memulai pengujian dalam siklus pengembangan perangkat lunak. Risiko produk juga memberikan informasi yang membantu membuat keputusan mengenai bagaimana pengujian harus dilakukan pada komponen atau sistem tertentu. Di dalam pengujian, cara-cara seperti ini digunakan untuk mengurangi risiko cacat yang terjadi.

Dalam sebuah proyek, seorang pemimpin penguji (*test lead*) akan menggunakan risiko dalam dua bagian yang berbeda yaitu risiko proyek dan risiko produk. Saat mengelola proyek, pemimpin penguji akan menggunakan risiko proyek. Sedangkan saat merencanakan dan mendefinisikan proyek, pemimpin pengujian akan menggunakan risiko produk.

2.4.2 Perencanaan dan Estimasi pengujian (*Test Planning and Estimation*)

Perencanaan pengujian adalah aktivitas paling penting yang perlu dilakukan oleh pemimpin pengujian (*test leader*) dalam setiap proyek uji. Hal ini untuk memastikan bahwa terdapat daftar tugas dan batasan dalam melaksanakan proses pengujian. Perencanaan pengujian digunakan dalam pengembangan dan implementasi proyek (pengujian) serta pemeliharaan (*maintenance*) kegiatan.

Dokumen yang dihasilkan dalam perencanaan pengujian sering disebut dengan project test plan. Dokumen tersebut mendefinisikan lingkup kegiatan pengujian yang direncanakan, biasanya diproduksi selama awal fase proyek dan nantinya akan memberikan informasi yang cukup untuk proses pengujian. Aktivitas yang lebih rinci terkait perkiraan tingkat pengujian akan didokumentasikan dalam *test-level plan* atau *system test plan*.

Perencanaan pengujian merupakan aktivitas yang berkelanjutan selama proses pengujian berlangsung. Ketika risiko atau perubahan terjadi, maka perencanaan juga akan diubah agar proses pengujian berikutnya dapat menyesuaikan dengan baik.

Estimasi pengujian memiliki dua pendekatan yaitu berbasis metrik dan berbasis ahli. Kedua pendekatan tersebut sangat berbeda karena pendekatan berbasis metrik berdasarkan atas data, sedangkan pendekatan berbasis ahli cukup subjektif.

Pendekatan berbasis metrik bergantung pada data yang dikumpulkan dari proyek sebelumnya atau yang serupa. Jenis data yang digunakan termasuk jumlah kasus uji yang ditulis, jumlah kasus uji yang telah dieksekusi, waktu yang dibutuhkan untuk mengembangkan kasus uji, jumlah cacat (*defect*) yang ditemukan, dan lain sebagainya. Sehingga data yang ada dapat digunakan untuk memperkirakan biaya dan waktu yang dibutuhkan proyek.

Pendekatan berbasis ahli adalah pendekatan yang menggunakan pengalaman ahli yang relevan untuk mendapatkan perkiraan, biasanya dikenal dengan istilah Wideband Delphi. Ahli yang dimaksud merupakan seorang pakar bisnis, konsultan proses pengujian, dan siapapun yang memiliki pengetahuan tentang aplikasi yang akan diuji atau tugas yang terlibat dalam proses pengujian.

2.4.3 Pendekatan Pengujian (*Test Approches/Strategies*)

Pendekatan pengujian adalah penerapan strategi pengujian pada proyek tertentu, mencakup semua keputusan tentang bagaimana pengujian akan dilaksanakan untuk keseluruhan organisasi, program kerja atau proyek individu berdasarkan tujuan dan sasaran

pengujian serta penilaian risiko. Biasanya dikembangkan diawal *life cycle* yang dikenal sebagai pencegahan. Pada pendekatan ini proses desain pengujian dimulai sedini mungkin untuk menghentikan cacat yang dibangun.

Pendekatan pengujian berisi semua keputusan yang dibuat tentang pelaksanaan pengujian berdasarkan tujuan dan target, serta penilaian risiko proyek (pengujian). Ada banyak pendekatan atau strategi yang dapat digunakan oleh tim *tester* tergantung dengan situasi dan kondisi pengujian. Biasanya *tester* akan mempertimbangkan risiko, bahaya, keselamatan, sumber daya dan keterampilan, serta teknologi yang tersedia. Pendekatan yang berbeda dapat digabungkan, sesuai dengan keadaan manajemen pengujian yang terjadi saat itu.

2.4.4 Organisasi Pengujian (*Test Organisation*)

Pada pengujian terdapat istilah pengujian independen, biasanya dilakukan perusahaan atau organisasi yang sistem perangkat lunaknya sesuai dengan persyaratan yang diinginkan. Pengujian independen adalah pengujian yang dilakukan oleh orang lain selain *developer* atau pengembang dari kode yang sedang diuji. Hal ini akan meningkatkan efektivitas pengujian apabila dilakukan dengan benar.

Sebagai penulis kode seringkali kurang dapat melihat kesalahan sendiri, seperti kesalahan ejaan atau penggunaan sintaks yang salah, hingga kesalahan mendasar pada inti dokumen. Sehingga penulis kode merasa yakin bahwa telah menghasilkan apa yang dibutuhkan dengan baik. Sebaliknya, penguji akan menganggap bahwa dokumen yang dilakukan pengujian kemungkinan mengandung kesalahan dan akan mengidentifikasi serta menemukan kesalahan tersebut. Oleh karena itu pengujian independen penting karena sulitnya penulis kode mengidentifikasi kesalahannya sendiri, tetapi lebih mudah apabila orang lain yang melihatnya.

Secara umum semakin jauh penguji dari produksi dokumen, semakin besar tingkat independensinya. Semakin tinggi tingkat independensinya, semakin besar kemungkinan kesalahan dalam pengujian yang timbul. Tingkat independensi juga akan tergantung pada ukuran organisasi, dalam organisasi yang lebih kecil di mana setiap orang berkontribusi pada setiap aktivitas maka lebih sulit untuk membedakan peran penguji dari peran lain. Hal tersebut cenderung membuat penguji tidak memiliki pikiran yang independen, sedangkan dalam organisasi di mana ada peran yang jelas jauh lebih mudah bagi penguji untuk memiliki pikiran yang independen.

2.4.5 Pemantauan dan Pengendalian Pengujian (*Monitoring and Controlling Test*)

Pemantauan pengujian adalah peninjauan terhadap apa yang sebenarnya terjadi setelah mengembangkan rencana pengujian, kegiatan dan rentang waktu yang telah ditentukan. Tujuan utama pemantauan pengujian adalah untuk memberikan umpan balik dan visibilitas kemajuan kegiatan pengujian.

Data yang diperlukan untuk memantau kemajuan dapat dikumpulkan secara manual atau menggunakan alat manajemen pengujian yang canggih. Metrik pengujian lebih sering digunakan untuk memantau kemajuan terhadap penyelesaian pengujian biasanya berbentuk grafik. Sekarang terdapat tren penggunaan *dashboard* yang menggambarkan semua metrik relevan dalam satu halaman untuk memastikan pengaruh yang terjadi.

Kontrol pengujian menggunakan referensi pada laporan data kemajuan pengujian untuk memutuskan tindakan pengendalian aktivitas pengujian dan memastikan *exit criteria* terpenuhi. Hal seperti itu perlu dilakukan ketika aktivitas pengujian tidak sesuai dengan jadwal yang telah direncanakan. Tindakan yang diambil dapat berdampak pada aktivitas pengujian dan berpengaruh pada aktivitas siklus hidup perangkat lunak lainnya.

Hal yang dilakukan dalam kontrol pengujian biasanya berupa membuat ulang prioritas pengujian ketika risiko proyek yang diidentifikasi terjadi, mengubah jadwal pengujian karena ketersediaan (*availability*) lingkungan pengujian yang berubah-ubah. Selain itu juga meninjau risiko produk dan mungkin mengubah peringkat (urutan) risiko agar memenuhi target.

2.4.6 Manajemen Insiden (*Incident Management*)

Insiden adalah setiap peristiwa yang terjadi dan tidak direncanakan. Hal ini di dalam pengujian diartikan menjadi hal di mana hasil yang ada berbeda dengan yang diharapkan. Suatu insiden ketika diselidiki dapat berupa cacat (*defect*), perubahan spesifikasi, atau permasalahan pada pengujian yang sedang dijalankan. Setelah ditemukan suatu insiden, sangat penting untuk melakukan pelacakan hingga mendalam. Karena insiden dapat muncul kapan saja sepanjang siklus hidup (*life cycle*) pengembangan perangkat lunak.

Insiden harus ditulis di dalam laporan insiden baik melalui sistem manajemen insiden atau ditulis di atas kertas. Laporan insiden memiliki tujuan untuk memberikan umpan balik kepada pengembang dan pihak lain tentang masalah yang ada serta memungkinkan dilakukan identifikasi, isolasi, dan koreksi seperlunya. Selain itu juga bertujuan untuk memberikan pemimpin pengujian (*leader test*) sarana untuk menilai kualitas sistem yang sedang diuji dan

kemajuan pengujian. Hal tersebut dilakukan berdasarkan jumlah insiden yang terjadi, prioritas insiden, dan jumlah insiden yang telah diperbaiki.

2.4.7 Laporan Pengujian (*Test Report*)

Laporan pengujian adalah proses di mana metrik pengujian dilaporkan dalam format ringkasan yang mampu memberi informasi kepada pembaca mengenai tugas pengujian yang telah dilakukan. Informasi yang dilaporkan berisi hal yang terjadi selama periode tertentu, tingkatan pengujian, dan seluruh upaya dalam pengujian.

Pada tahap laporan juga berisi tentang analisis informasi dan metrik yang diperlukan untuk mendukung rencana atau keputusan yang akan dilakukan dimasa mendatang. Sehingga pada proyek selanjutnya terdapat peluang untuk meningkatkan proses pengujian. Karena laporan memuat informasi terkait tujuan yang telah ditetapkan apakah berjalan dengan benar, pendekatan atau strategi pengujian yang digunakan, dan lain sebagainya.

Sebuah artikel (Rajeshwari Sissodia, 2017) mengevaluasi sistem dengan cara manual atau otomatis untuk memverifikasi bahwa sistem memenuhi persyaratan yang telah ditentukan sebelum pengujian. Manajemen dan kontrol pengujian pada makalah ini melibatkan tiga aktivitas utama yaitu perencanaan pengujian, pelaksanaan pengujian, dan manajemen cacat. Aktivitas pengujian perlu direncanakan dan dan dipantau secara terstruktur untuk memastikan produk berkualitas sesuai dengan semua jadwal proyek. Tujuan dari artikel ini adalah untuk menyajikan metode, prosedur, dan pendekatan yang akan digunakan dalam manajemen dan kontrol pengujian. Terdiri dari aktivitas penting seperti mendefinisikan metodologi pengujian secara rinci untuk perencanaan pengujian dan menetapkan proses serta prosedur yang akan dijalankan selama eksekusi pengujian.

Artikel (Michal Pawlak, 2018) menganalisis tentang Kungfu Testing sebagai pendekatan pengujian berdasarkan saran dan praktik terbaik yang dianjurkan oleh para ahli di bidang pengujian. Pendekatan Kungfu Testing memberikan langkah-langkah untuk mengelola kegiatan pengujian pada layanan online teladan 'Erasmus Booking'. Tahapan yang digunakan dimulai dari *risk analysis* (analisis risiko), *test planning* (perencanaan pengujian), *test estimation* (estimasi pengujian), *test organization* (organisasi pengujian), *test monitoring and controlling* (monitoring dan kontroling pengujian), *issue management* (manajemen masalah) atau *incident management* (manajemen masalah), dan terakhir adalah *test report* (laporan pengujian).

2.5 Alat Pengujian Perangkat Lunak

Pengujian perangkat lunak dibagi menjadi dua kategori yaitu analisis statis dan analisis dinamis (Kshirasagar Naik, 2008). Alat analisis statis membantu pengujian dalam fase pengkodean, sedangkan alat analisis dinamis digunakan dalam fungsionalitas, integrasi, dan pengujian unit. Alat-alat tersebut menilai sistem ketika perangkat lunak berjalan dan dikelompokkan berdasarkan aktivitas pengujian. Misalnya adalah alat pengujian yang mendukung aktivitas manajemen, alat untuk mendukung pengujian statis, dan lain sebagainya.

2.5.1 Alat Manajemen Pengujian

Alat manajemen pengujian dapat digunakan diseluruh siklus hidup pengembangan perangkat lunak. Pada penggunaannya, manajer pengujian akan menggunakan alat manajemen pengujian untuk mengelola pengujian pada sistem (Dorothy Graham, 2006). Fitur atau karakteristik alat manajemen pengujian mencakup dukungan untuk:

1. Manajemen pengujian, misalnya mengetahui nomor urut pengujian, ditulis, dijalankan, lulus atau gagal, dan lain sebagainya.
2. Melakukan penjadwalan pengujian yang akan dieksekusi, baik secara manual maupun menggunakan alat.
3. Manajemen aktivitas pengujian, misalnya waktu yang dihabiskan dalam desain pengujian, pelaksanaan pengujian, dan lain sebagainya.
4. Mengenali dan menelusuri proses pengujian (pada hasil pengujian dan cacat pada persyaratan).
5. Mencatat hasil pengujian, menyiapkan laporan berdasarkan matriks yang telah dibuat. Terkait proses pengujian, hasil yang diterima, insiden yang muncul, cacat yang telah diperbaiki maupun yang di luar perkiraan.

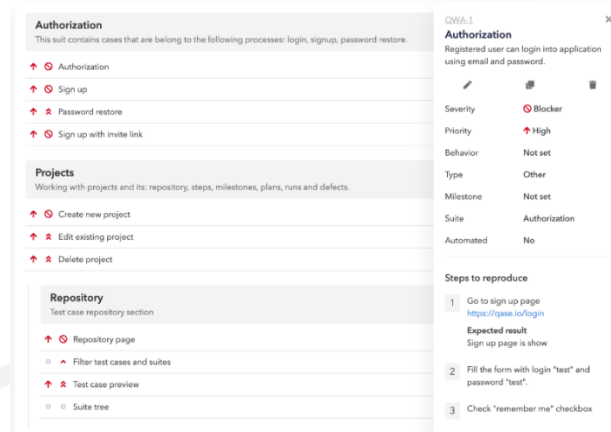
Sebuah artikel (Rajeshwari Sissodia, 2017) menjelaskan bahwa pengujian pada dasarnya digunakan untuk menemukan bug, error dari sistem, buat laporan dan kirim ke pengembang. Pengujian dapat dilakukan dengan dua jenis baik pengujian manual maupun otomatisasi. Artikel ini menunjukkan pentingnya pengujian otomatisasi dan kasus uji, perbedaan pengujian otomatisasi dan pengujian manual serta penggunaan alat QTP (*Quicktest Professional*) sebagai alat untuk pengujian. QTP adalah produk dari HP (Hewlett Packard) yang membantu penguji untuk melakukan pengujian fungsional otomatis dengan mulus tanpa pemantauan setelah pengembangan skrip selesai.

Artikel (Baidhya, 2019) bertujuan untuk mengembangkan sistem manajemen pengujian yang dapat menyelesaikan permasalahan dan sesuai dengan kriteria perusahaan. Peneliti mengawali artikel dengan pengetahuan teori tentang topik permasalahan yang ada, memeriksa proses pengujian yang sedang berlangsung, memilih alat manajemen pengujian yang sesuai dan akhirnya membangun sistem di alat yang dipilih. Alat manajemen pengujian yang dipilih adalah TestLink, karena setelah dilakukan peninjauan (*review*) ditemukan bahwa TestLink lebih baik daripada sistem lama yang digunakan oleh perusahaan. TestLink adalah salah satu sistem atau alat manajemen pengujian perangkat lunak open source berbasis web yang sering digunakan untuk membantu menjaga kualitas perangkat lunak. Setiap catatan tahapan *Software Testing Life Cycle* (STLC) mulai dari tahap proyek uji hingga ke tahap pembuatan laporan akan dikelola oleh TestLink.

2.5.2 Qase Tools

Qase adalah sistem manajemen kasus uji (*test case management*) berbasis *cloud* yang digunakan oleh pengembang dan penguji (Rajkumar, 2022). Qase bisa diakses secara gratis di link <https://app.qase.io/> melalui pencarian Google. Qase membantu untuk meningkatkan produktivitas pengujian dan memungkinkan untuk mengelola kasus pengujian. Selain itu, Qase juga menyusun rencana pengujian dan melakukan pengujian dengan cara yang cukup mudah.

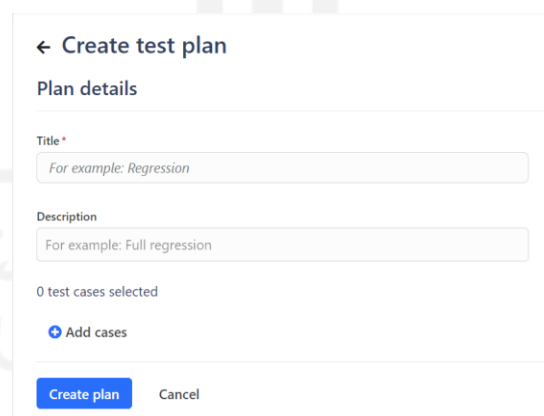
Qase memiliki beberapa fitur yang mendukung penggunaannya yaitu *test case management*, *test plans and runs*, *team management*, dan *integration*. Fitur pertama yaitu pengelolaan kasus uji (*test case management*) yang dapat membantu mengatur semua kasus pengujian ke dalam grup logis (*logical groups*). Fitur ini memungkinkan untuk menentukan tingkat keparahan kasus uji dan prioritasnya; menjelaskan prakondisi dan pascakondisi, serta langkah-langkah untuk mereproduksi kasus uji ditampilkan pada Gambar 2.2.



Gambar 2.2 Fitur *Test Case Management*

Fitur kedua pada Qase adalah rencana pengujian dan penjalanannya (*test plans and runs*). Fitur ini dapat membantu penguji menyusun rencana pengujian dan menjalankan pengujian, serta membantu memeriksa semua kasus sekaligus. Bagian fitur kedua ini memiliki pelacak waktu (*time tracker*) yang akan menampilkan informasi terperinci tentang waktu yang dihabiskan untuk setiap kasus yang diuji. Penguji dapat dengan mudah membagikan laporan pengujian dengan informasi mendetail pada kasus pengujian yang dijalankan kepada manajer. Ditampilkan pada

Gambar 2.3 yang menunjukkan fitur test plan dan Gambar 2.4 menunjukkan fitur test run Qase.



Gambar 2.3 Fitur *test plan*

← Start test run

Run title *

Test run 2022/06/16

Description

For example: Full regression

Plan

Select...

Environment

Not set

Milestone

Not set

Default assignee

Select...

Tags

Select...

Test cases

0 test cases selected

Start run Cancel

Gambar 2.4 Fitur *test run*

Selanjutnya fitur ketiga pada Qase yaitu manajemen tim (*team management*), pengujian perangkat lunak dalam sebuah tim. Seperti pada Gambar 2.5 dan Gambar 2.6, fitur ini dapat mengundang rekan tim untuk bergabung dan menggunakan semua alat yang ada pada manajemen kasus uji bersama-sama seperti menjalankan pengujian, menulis dokumentasi pengujian, dan menyusun rencana pengujian. Tim yang menggunakan Qase 50% lebih sedikit menghabiskan waktu untuk mengelola kasus uji dan 30% lebih sedikit menghabiskan waktu untuk eksekusi pengujian.

Your team

Invite new member

All members (2) Invited

User	Status	Role	Role title	Last action
Rajakumari arivurajee95@gmail.com	✓	Owner	Team owner	36 minutes ago
Venkey venkeycoofun@gmail.com	✓	Member	QA	2 months ago

Gambar 2.5 Fitur *team management*

Invites

Invite new member Search for existing

Invite new user

Single Bulk

Email *

example@example.com

Name

For example: John Smith

Role title *

For example: Senior QA

Role

Member (default)

☐ Read only

☒ Access

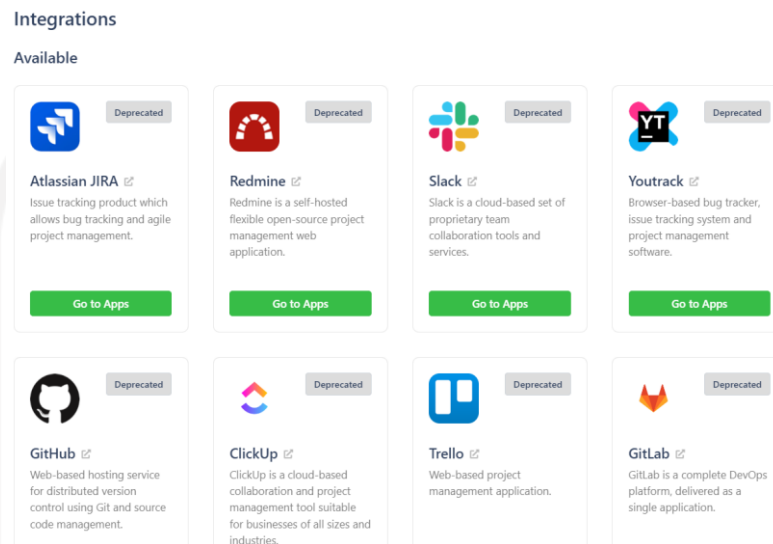
Read-only users don't count as regular users (are not billed) and have only a read access (based on the role settings).

A user will receive access to all private projects in your workspace.

Cancel Invite

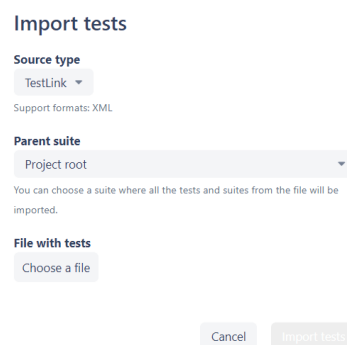
Gambar 2.6 Fitur *invite new member*

Fitur keempat yang ada pada Qase adalah integrasi (*integration*), di mana penguji dapat terintegrasi dengan alat yang digunakan seperti Jira, GitHub, Trello, dan lain sebagainya seperti pada Gambar 2.7. Penguji dapat membuat catatan atau tugas pada pelacak masalah (*issue trackers*) langsung dari sistem manajemen kasus uji. Penguji juga dapat mengirimkan laporan bug ke tim pengembang dengan instruksi mendetail terkait cara mereproduksinya.

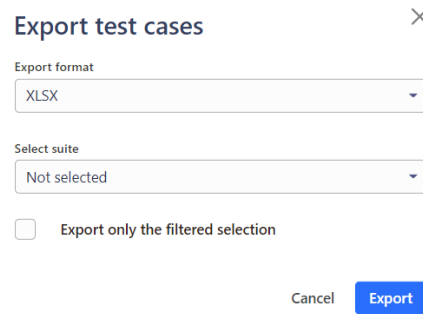


Gambar 2.7 Fitur *integration*

Qase memudahkan penguji untuk mengimpor kasus uji dari alat manajemen pengujian populer seperti TestLink, Jira, Testrail dan lain sebagainya serta mengekspor kasus uji dari Qase ke berbagai format seperti XLSX, CSV, dan lain sebagainya (Arivazhagan, 2022). Sehingga dalam penggunaan Qase penguji dapat menyampaikan masalah penting yang ditemukan selama pengujian secara langsung ke pengembang dalam waktu singkat untuk penyelesaiannya. Tampilan fitur impor kasus uji Qase ada pada Gambar 2.8 dan fitur ekspor kasus uji ada pada Gambar 2.9.



Gambar 2.8 Fitur *import test cases*



Export test cases

Export format

XLSX

Select suite

Not selected

☐ Export only the filtered selection

Cancel Export

Gambar 2.9 Fitur *export test cases*

Artikel (Devaldi Akbar Suryadi, 2022) bertujuan untuk mengetahui kualitas produk aplikasi Menu Harianku dan memberikan rekomendasi peningkatan kualitas aplikasi berdasarkan model ISO/IEC 25010:2011. Standar ISO/IEC 25010:2011 dipilih karena dianggap sesuai untuk mengukur kualitas perangkat lunak. Pengujian diawali dengan mengidentifikasi masalah, kualitas desain, menguji kualitas performa, dan membuat rekomendasi perbaikan. Pengujian karakteristik fungsional dilakukan menggunakan metode black box dengan bantuan alat manajemen pengujian Qase untuk mengelola kasus uji dan eksekusi pengujian.

BAB III

ANALISIS DAN PERENCANAAN

Proses manajemen pengujian perangkat lunak memiliki beberapa tahapan yang harus dilakukan. Pada bab ini berisi tentang tahap analisis dan rencana dari pengujian berdasarkan metode yang digunakan dalam proses manajemen pengujian perangkat lunak. Tahap pertama dilakukan analisis terkait *use case* dari aplikasi Ivent yang memberikan gambaran risiko apabila fungsi tersebut tidak dilakukan. Tahap kedua adalah perencanaan yang berisi tentang skenario pembuatan *test case* dan estimasi waktu perjalanan dari pengujian aplikasi Ivent. Berikutnya tahap ketiga menentukan metode yang digunakan untuk pengujian. Aplikasi Ivent diuji menggunakan metode *Blackbox* atau dengan kata lain menguji fungsionalitas fitur yang ada.

3.1 Analisis Risiko dan Pengujian (*Risk and Testing Analysis*)

Aplikasi yang akan dilakukan pengujian adalah aplikasi Ivent yang berbasis web (Mucjal, 2021) (Dani, 2021) (Idris, 2021). Ivent merupakan *e-commerce* dengan model bisnis yang menghubungkan antara pemilik bisnis dengan pencari *event organizer* atau vendor. Terdapat dua pengguna utama yang akan menggunakan aplikasi ini yaitu pengguna sebagai pemilik vendor atau *event organizer* dan pengguna sebagai pencari vendor. Pemilik vendor dapat memposting foto atau portofolio pada aplikasi Ivent sehingga pencari vendor dapat melihat produk yang ditawarkan.

Terdapat 3 aktor yang dapat menggunakan *website* aplikasi Ivent yaitu penjual (vendor atau *event organizer*), konsumen, dan admin seperti pada tampilan *use case* diagram Gambar 3.1 dengan modifikasi dan penjelasan sebagai berikut (Mucjal, 2021):

1. Register

Pada tahap ini penjual (vendor atau *event organizer*), konsumen, dan admin akan melakukan pendaftaran akun sebagai penjual atau sebagai konsumen untuk menggunakan *website* pada aplikasi Ivent. Sehingga nantinya pengguna tidak perlu berkali-kali mendaftar ketika menggunakan *website* Ivent, karena data pengguna telah tersimpan. Jika tidak melakukan registrasi atau pendaftaran, pengguna tidak akan memiliki akun untuk beraktivitas pada aplikasi.

2. Login

Tahap *login* merupakan bagian dari keamanan sebuah aplikasi untuk penggunanya. Setelah *login* nantinya admin dapat melihat daftar transaksi, mengecek pembayaran dari pembeli dan verifikasi pembayaran konsumen. Penjual (vendor atau *event*

organizer) dapat membuat produk, mengedit produk, dan melihat produk yang dipesan. Lalu konsumen dapat melihat dan memesan produk, melihat pesanan, serta melakukan pembayaran. Jika pengguna tidak melakukan login maka fitur aplikasi tidak dapat digunakan secara maksimal dan aktivitas yang dilakukan tidak akan tersimpan pada *database*.

3. Membuat Produk

Penjual (*vendor* atau *event organizer*) dapat menambahkan produk yang akan disewakan berupa jasa atau barang di *website* Ivent. Penjual akan mengisi *form* produk untuk melengkapi spesifikasi produk. Jika penjual tidak mengisi *form* produk maka spesifikasi produk tidak akan muncul pada beranda, sehingga pembeli tidak akan tahu spesifikasi produk yang dijual.

4. Edit Produk

Penjual (*vendor* atau *event organizer*) dapat melakukan edit produk untuk mengubah spesifikasi dari produk yang akan disewakan. Penjual tidak akan dapat mengganti keterangan produk ketika terdapat kesalahan input atau penulisan jika mengabaikan kegunaan edit produk.

5. Melihat Produk

Berikutnya penjual (*vendor* atau *event organizer*) dan konsumen dapat melihat produk-produk yang akan disewakan. Penjual dapat melihat produk miliknya pada halaman toko, sedangkan konsumen dapat melihat pilihan produk pada halaman Sewa.

6. Memesan produk

Apabila konsumen telah melihat produk yang diinginkan, konsumen dapat memesan produk pada halaman detail produk untuk memasukkan produk yang dipilih.

7. Melihat Pesanan

Setelah memasukkan pesanan produk yang diinginkan, konsumen dapat melihat pesanan produk pada halaman *cart*. Jika tidak memeriksa pesanan pada halaman *cart*, dikhawatirkan terjadi kesalahan pemesanan produk yang tidak dapat ditukarkan kembali.

8. Melihat Transaksi

Konsumen dapat melihat transaksi yang telah dilakukan dan melakukan pembayaran pesanan produk yang disewa, serta melakukan pengiriman bukti transfer. Admin akan melihat bukti transfer dari pembayaran yang dilakukan konsumen.

9. Membayar Produk

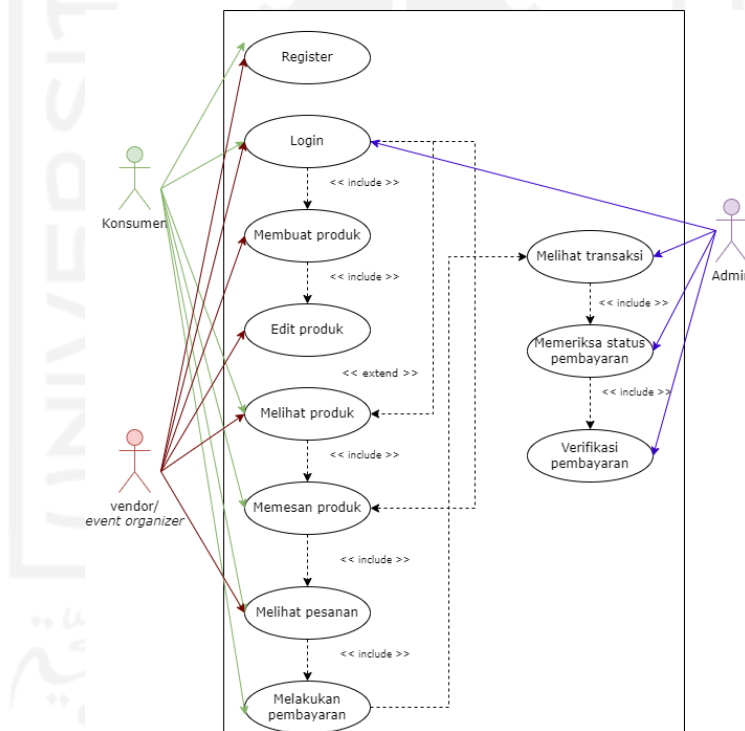
Pada tahap ini konsumen dapat membayar pesanan produk dengan cara transfer yang dapat dilihat pada halaman Pembayaran. Pembeli harus membayar produk pesannya agar pesanan dapat segera diproses.

10. Memeriksa Status Pembayaran

Memeriksa status pembayaran dilakukan oleh admin pada halaman *dashboard* admin dan dapat mengubah status dari pembayaran yang dilakukan konsumen.

11. Verifikasi Pembayaran

Setelah konsumen mengirimkan bukti pembayaran, admin akan melakukan verifikasi pembayaran pada halaman *dashboard* admin dan mengubah status pembayaran yang telah dilakukan konsumen. Sehingga pemesanan produk dapat segera diproses kepada penjual.



Gambar 3.1 Use Case Diagram dari Ivent

Pengujian dilakukan terhadap seluruh *use case* yang diimplementasikan pada fitur aplikasi Ivent. Berdasarkan risiko yang telah dijelaskan, terkait apabila *use case* tidak dilakukan nantinya akan memberikan gambaran pembuatan *test case* pada tahap berikutnya.

3.2 Perencanaan dan Estimasi Pengujian (*Test Planning and Estimation*)

Sebelum pengujian dilakukan, diperlukan perencanaan dalam membuat *test case*, agar proses pengujian lebih terstruktur dan mudah untuk dijalankan. Dikarenakan setiap *use case*

memiliki beberapa aktivitas yang tidak semua aktor dapat melakukan aktivitas tersebut. Pengujian akan menggunakan beberapa *use case* sesuai dengan batasan yang akan dibuat untuk melihat kelayakan fungsional pada perangkat lunak.

Berdasarkan perbedaan kegunaan setiap *use case*, maka dibuatlah batasan pengujian hanya sampai pada tahap fitur yang telah diimplementasi. Hal tersebut dikarenakan keterbatasan pada sumber yang digunakan, yaitu terkait keberlanjutan aplikasi yang belum selesai dalam pembuatannya. *Use case* yang akan diuji mulai dari *registrasi*, *login*, membuat produk, edit produk, dan melihat produk.

Skenario pengujian nantinya akan menggunakan metode *blackbox* untuk melakukan pengujian fungsionalitas. Setiap *use case* akan diuji untuk melihat kesesuaian fungsinya, apakah keluarannya sesuai dengan masukan yang dilakukan penguji. Setiap *use case* akan dikodekan dengan awalan TC ketika pengujian dan nantinya akan memiliki kode angka sesuai dengan urutan *use case* yang telah dibuat sebelumnya.

Berdasarkan aktivitas yang ada pada setiap *use case*, akan dibuat beberapa skenario pengujian seperti diperlihatkan pada Tabel 3.1.

Tabel 3.1 Skenario Pengujian Aplikasi Ivent

Id	Use Case	Skenario Pengujian	Tahapan Pengujian	Keluaran yang Diharapkan
TC 1.1	Register	Mengisikan data diri sesuai <i>form</i> pendaftaran.	<ol style="list-style-type: none"> 1. Membuka halaman beranda Ivent 2. Memilih opsi <i>register</i> atau daftar 3. Masuk ke dalam halaman <i>register</i> 4. Mengisi <i>form</i> dengan benar 5. Klik “<i>Register</i>” 	Pengguna dapat mendaftarkan data dirinya dan data tersimpan.
TC 1.2		Mengosongkan bagian nama lengkap pada <i>form</i> pendaftaran.	<ol style="list-style-type: none"> 1. Membuka halaman beranda Ivent 2. Memilih opsi <i>register</i> atau daftar 3. Masuk ke dalam halaman <i>register</i> 4. Mengisi <i>form</i> dan mengosongkan bagian nama lengkap 5. Klik “<i>Register</i>” 	Pengguna tetap berada pada halaman <i>register</i> dan muncul peringatan kesalahan.
TC 1.3		Mengosongkan bagian alamat email pada <i>form</i> pendaftaran.	<ol style="list-style-type: none"> 1. Membuka halaman beranda Ivent 2. Memilih opsi <i>register</i> 	Pengguna tetap berada pada halaman <i>register</i> dan muncul peringatan

			atau Daftar 3. Masuk ke dalam halaman <i>register</i> 4. Mengisi <i>form</i> dan mengosongkan bagian alamat email 5. Klik “Register”	kesalahan.
TC 2.1	Login Pengguna	Mengisikan <i>email</i> dan kata sandi yang sesuai.	1. Membuka halaman beranda Ivent 2. Memilih opsi <i>login</i> atau masuk 3. Berada dalam halaman login 4. Mengisi <i>form</i> dengan data yang telah didaftarkan 5. Klik “Login”	Pengguna dapat masuk ke halaman utama Ivent
TC 2.2		Mengosongkan bagian kata sandi pada <i>form</i> .	1. Membuka halaman beranda Ivent 2. Memilih opsi login atau masuk 3. Berada dalam halaman <i>login</i> 4. Mengisi <i>form</i> dan mengosongkan bagian kata sandi 5. Klik “Login”	Pengguna tidak dapat masuk dan tetap pada halaman login
TC 2.3		Mengosongkan bagian <i>email</i> pada <i>form</i>	1. Membuka halaman beranda Ivent 2. Memilih opsi <i>login</i> atau masuk 3. Berada dalam halaman <i>login</i> 4. Mengisi <i>form</i> dan mengosongkan bagian email 5. Klik “Login”	Pengguna tidak dapat masuk dan tetap pada halaman login
TC 2.4	Login Admin	Mengisikan email dan kata sandi yang sesuai.	1. Membuka halaman beranda Ivent 2. Memilih opsi <i>login</i> atau masuk 3. Berada dalam halaman <i>login</i> 4. Mengisi <i>form</i> dengan data yang telah didaftarkan 5. Klik “Login”	Admin dapat masuk ke halaman beranda admin.
TC 3.1	Tambah	Mengisikan <i>form</i>	1. Membuka halaman	Pengguna berhasil

	Produk	detail produk dengan lengkap.	Buka Toko 2. Klik “Tambah Produk” 3. Mengisi form detail produk dengan lengkap 4. Klik “Submit”	menambahkan produk.
TC 3.2		Tidak mengisi <i>form</i> detail produk pada bagian harga produk.	1. Membuka halaman Buka Toko 2. Klik “Tambah Produk” 3. Mengisi <i>form</i> detail produk, kecuali bagian harga produk 4. Klik “Submit”	Pengguna tetap pada halaman tambah produk dan muncul peringatan bahwa <i>form</i> ada yang kosong.
TC 4.1	Edit Produk	Melakukan edit detail produk.	1. Membuka halaman Produk 2. Pilih produk yang akan diubah 3. Klik “Edit Produk” 4. Mengisi <i>form</i> detail produk 5. Klik “Submit”	Pengguna berhasil mengubah produk.
TC 4.2		Menghapus harga produk dan mengganti dengan huruf	1. Membuka halaman Produk 2. Pilih produk yang akan diubah 3. Klik “Edit Produk” 4. Mengisi form detail produk, mengganti harga produk dengan huruf 5. Klik “Submit”	Pengguna tetap pada halaman edit dan mengisi kolom harga dengan angka
TC 5.1	Melihat Produk	Memilih antara topik “Event Organizer” atau “Vendor Acara”	1. Membuka halaman beranda Ivent 2. Klik halaman Sewa 3. Terdapat dua pilihan topik, klik topik “Event Organizer”	Pengguna berhasil Masuk pada halaman topik yang dipilih yaitu <i>Event Organizer</i>
TC 5.2		Mengisikan suatu kata pada bar pencarian	1. Membuka halaman beranda Ivent 2. Klik bar pencarian 3. Masukkan kata “speaker” 4. Klik tombol cari	Hasil pencarian pengguna muncul sesuai dengan kata yang dicari
TC 5.3		Mengisikan suatu angka pada bar pencarian	1. Membuka halaman beranda Ivent 2. Klik bar pencarian	Hasil pencarian pengguna muncul sesuai dengan angka yang

			<ol style="list-style-type: none"> Masukkan angka "100" Klik tombol cari 	dicari
TC 6.1	Membuat Pesanan	Melakukan Checkout dan mengisikan tanggal dengan benar.	<ol style="list-style-type: none"> Pilih produk Klik "Lihat produk" Mengisikan tanggal mulai dan tanggal selesai Klik "Tambah Pesanan" 	Berhasil menambahkan pesanan pada halaman Pesanan
TC 6.2		Melakukan Checkout dan tidak mengisikan tanggal dengan benar.	<ol style="list-style-type: none"> Pilih produk Klik "Lihat produk" Mengisikan tanggal mulai dan mengosongkan tanggal selesai Klik "Tambah Pesanan" 	Tetap berada pada halaman Produk dan terdapat peringatan kesalahan
TC 6.3		Melakukan Checkout dengan mengisikan alamat	<ol style="list-style-type: none"> Klik "Checkout" Klik "Tambahkan alamat" Mengisi form detail alamat Klik "Simpan" 	Berhasil membuat alamat pesanan untuk pengiriman
TC 6.4		Melakukan Checkout dengan mengosongkan kolom kode pos alamat	<ol style="list-style-type: none"> Klik "Checkout" Klik "Tambahkan alamat" Mengisi form detail alamat dengan mengosongkan kolom kode pos Klik "Simpan" 	Tetap berada pada halaman Pesanan dan terdapat peringatan kesalahan
TC 7.1	Melihat Pesanan	Melakukan klik pada halaman Pesanan	<ol style="list-style-type: none"> Klik <i>dropdown</i> pada profil Pilih "Pesanan Saya" 	Pengguna berhasil melihat pesanan dan deskripsi lainnya
TC 7.2		Melakukan klik pada halaman Pesanan dan batalkan salah satu pesanan	<ol style="list-style-type: none"> Klik <i>dropdown</i> pada profil Pilih "Pesanan Saya" Pilih salah satu pesanan, batalkan pesanan 	Pengguna berhasil membatalkan pesanan
TC 8.1	Melihat Transaksi	Memeriksa halaman transaksi	<ol style="list-style-type: none"> Masuk ke beranda Ivent menggunakan akun admin Klik "Transaksi" 	Berhasil masuk halaman transaksi dan melihat keseluruhan transaksi

TC 8.2		Melihat detail transaksi pembelian	<ol style="list-style-type: none"> 1. Masuk ke beranda Ivent menggunakan akun admin 2. Klik “Transaksi” 3. Pilih salah satu transaksi 	Ditampilkan detail transaksi yang dipilih
TC 9.1	Melakukan Pembayaran	Melakukan pembayaran dan mengunggah bukti transfer.	<ol style="list-style-type: none"> 1. Klik salah satu pesanan 2. Pilih “Bayar” 3. Klik “<i>Choose file</i>” dan upload bukti pembayaran (.jpg atau .png) 4. Lalu klik “Kirim” 	Pengguna berhasil melakukan pembayaran dan mendapatkan bukti bayar
TC 9.2		Melakukan pembayaran tanpa mengunggah bukti transfer.	<ol style="list-style-type: none"> 1. Klik salah satu pesanan 2. Pilih “Bayar” 3. Mengabaikan unggah bukti pembayaran 4. Lalu klik “Kirim” 	Pengguna gagal melakukan pembayaran dan tetap pada halaman pembayaran
TC 10.1	Memeriksa Status Pembayaran	Membuka halaman dashboard admin dan cek status pembayaran pembeli	<ol style="list-style-type: none"> 1. Masuk ke beranda Ivent menggunakan akun admin 2. Klik “Transaksi” 	Admin melihat seluruh daftar transaksi konsumen
TC 10.2		Membuka halaman Transaksi dengan melihat salah satu transaksi	<ol style="list-style-type: none"> 1. Masuk ke beranda Ivent menggunakan akun admin 2. Klik “Transaksi” 3. Pilih salah satu transaksi 	Admin melihat detail salah satu transaksi
TC 11.1	Verifikasi Pembayaran	Melakukan verifikasi pembayaran konsumen.	<ol style="list-style-type: none"> 1. Masuk ke beranda Ivent menggunakan akun admin 2. Klik “Transaksi” 3. Pilih salah satu transaksi 4. Klik “Verifikasi Pembayaran” 	Transaksi pesanan konsumen telah diverifikasi
TC 11.2		Melakukan kirim produk setelah verifikasi	<ol style="list-style-type: none"> 1. Masuk ke beranda Ivent menggunakan akun admin 2. Klik “Transaksi” 3. Pilih salah satu transaksi 4. Klik “Verifikasi Pembayaran” 5. Klik “Kirim Produk” 	Transaksi pesanan konsumen dikirim

Estimasi waktu pengujian dilakukan selama 3 bulan, dimulai bulan september hingga november. Pembagian estimasi proses analisis dan rencana pengujian dilakukan selama 1 bulan pertama. Lalu dilanjutkan proses eksekusi dilakukan selama 1 bulan secara manual, baik melalui Qase maupun *running* aplikasi Ivent. Berikutnya ada tahap pembuatan laporan akhir pengujian yang dilakukan selama 1 bulan. Untuk itu estimasi biaya selama pengujian adalah 0 rupiah. Hal tersebut dikarenakan penggunaan Qase gratis dan alat pendukung milik sendiri sehingga tidak mengeluarkan biaya.

3.3 Pendekatan Pengujian (*Test Approches/Strategies*)

Pengujian dilakukan dengan menggunakan metode *blackbox* yang dapat membantu penguji untuk membuat perangkat lunak menjadi lebih baik sebelum diberikan kepada pengguna. Metode *blackbox* dapat diartikan menjadi sebuah metode yang digunakan oleh penguji untuk menguji sebuah perangkat lunak tanpa harus memperhatikan *source code* dan detail lainnya. *Blackbox* sering disebut dengan pengujian fungsionalitas yang hanya menguji fitur-fitur fungsional pada perangkat lunak (Bagus Bayu Sasongko, 2021).

Pengujian menggunakan *blackbox* hanya memeriksa bagian hasil dari nilai keluaran berdasarkan nilai masukan. Proses tersebut dilakukan dengan mencoba memasukkan beberapa data pada program perangkat lunak yang telah dibuat. Berdasarkan hal tersebut dapat diketahui program tersebut berjalan sesuai dengan rencana yang dibutuhkan oleh perusahaan (Fadhila Cahya Ningrum, 2019).

Blackbox memiliki beberapa jenis pengujian yaitu *Equivalence Partitions*, *Boundary Value Analysis*, *Sample Testing*, *Comparison Testing*, *Robustness Testing*, *Behavior Testing*, *Requirement Testing*, *Performance Testing*, *Endurance Testing* dan *Cause* (Fadhila Cahya Ningrum, 2019). Pengujian *blackbox* pada penelitian biasanya menggunakan metode *Equivalence Partitions* yang dilakukan berdasarkan klasifikasi masukan yang telah dikelompokkan sesuai dengan fungsinya (Taufik Hidayat, 2020).

Pengujian fungsional adalah proses penjaminan kualitas yang pengujiannya berdasarkan pada spesifikasi perangkat lunak yang diuji. Pengujian dilakukan dengan memberikan *input* data dan memeriksa *output*nya (Riyanthi Angrainy Sianturi, 2021). Pendapat lain mengatakan pengujian fungsional merupakan pengujian perangkat lunak terkait dengan layanan yang disediakan oleh sistem (Adila Chusnul Fatiyah, 2019).

Pengelompokan diperlukan saat menggunakan pendekatan metrik yang biasanya digunakan sebagai alat untuk mengukur fungsionalitas, *source code*, arsitektur, dan kinerja

dari perangkat lunak (Muhammad Andryan Wahyu Saputra, 2021). Pendekatan metrik adalah tindakan mengevaluasi dan menilai pelaksanaan dan menentukan tingkat produktivitas perangkat lunak (Rizky Parlika, 2020). Pendekatan metrik dapat membantu meningkatkan proses pengembangan perangkat lunak dan memberi informasi terkait prediksi yang nantinya digunakan untuk masa depan.



BAB IV

ORGANISASI, PEMANTAUAN DAN PENGENDALIAN

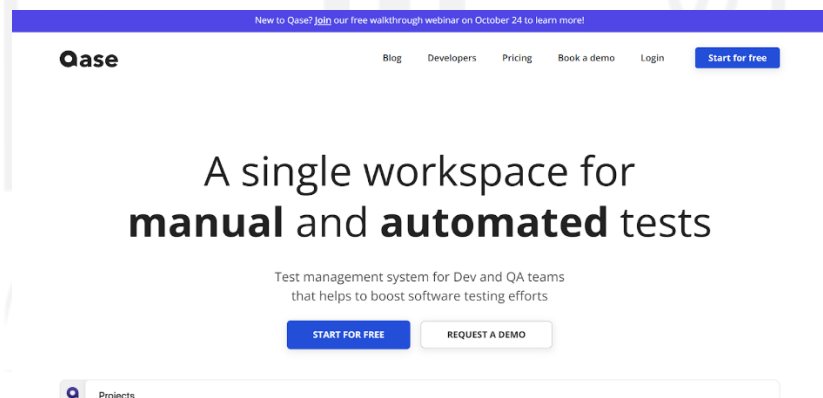
Pada bab ini akan membahas tahap selanjutnya dari metode manajemen pengujian perangkat lunak, yaitu tahap keempat sampai ketujuh. Tahap keempat terdapat pengorganisasian atau pengelompokan pengujian. Lalu tahap kelima menjadi tahap hasil dari eksekusi pengujian pada *test case* yang telah dibuat. Tahap keenam berisikan *defect* atau kegagalan dan solusi dari pengujian yang telah dilakukan. Terakhir adalah tahap pembuatan laporan hasil dari keseluruhan pengujian aplikasi Ivent yang berhasil dilakukan.

4.1 Organisasi Pengujian (*Test Organisation*)

Qase membantu untuk meningkatkan produktivitas pengujian dan memungkinkan untuk mengorganisasikan atau mengelola kasus pengujian. Selain itu, Qase juga menyusun rencana pengujian dan melakukan eksekusi pengujian dengan cara yang cukup mudah. Qase memiliki berbagai fitur seperti *test case management*, *test plans and runs*, *team management*, dan lain lain. Sehingga dengan kata lain Qase membantu dalam manajemen segala hal terkait pengujian agar terorganisir dengan baik.

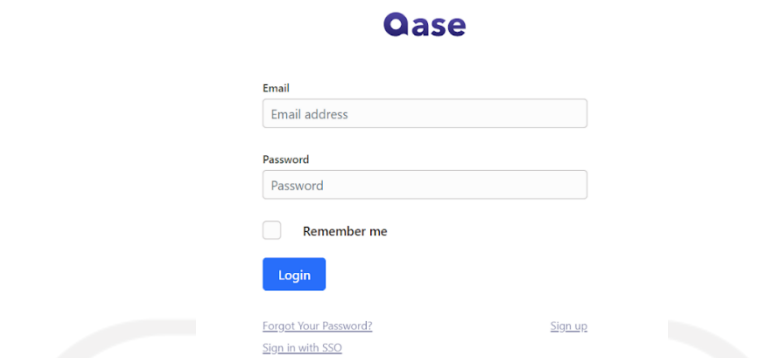
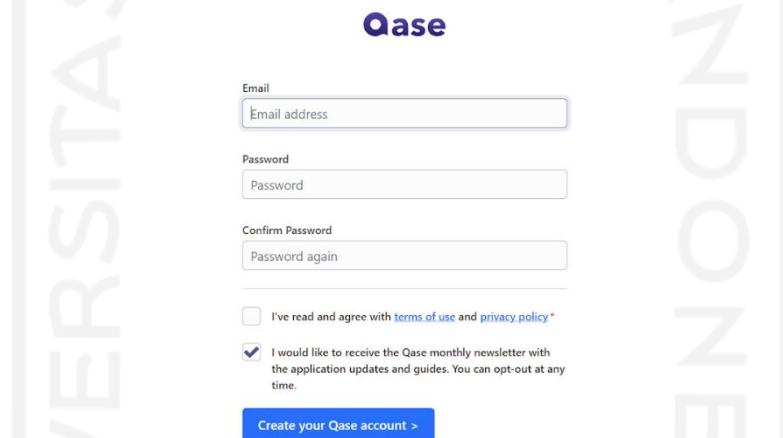
4.1.1 Persiapan

Langkah persiapan dalam penggunaan Qase dapat dilakukan dengan mengakses Qase melalui <https://app.qase.io/> pada pencarian Google Chrome. Setelah mengakses *url* tersebut, nantinya akan muncul halaman beranda seperti pada Gambar 4.1.



Gambar 4.1 Halaman beranda Qase

Pengguna perlu melakukan *login* terlebih dahulu untuk menggunakan fitur Qase secara lengkap seperti pada Gambar 4.2. Apabila belum memiliki akun, maka pengguna perlu mendaftarkan diri terlebih dahulu dengan mengisi *form* seperti pada Gambar 4.3. Nantinya pengguna tidak perlu mendaftar berulang kali ketika akan menggunakan Qase sehingga hanya dengan klik *login* maka sistem akan otomatis masuk ke dalam halaman utama Qase.


Gambar 4.2 Halaman *login* Qase


Gambar 4.3 Halaman daftar akun Qase

4.1.2 Pengujian

Terdapat beberapa langkah yang dapat dilakukan dalam pengujian perangkat lunak menggunakan Qase yaitu:

1. Membuat *project*
2. Membuat *test suite*
3. Membuat *test case*
4. Membuat *test plan*
5. Menjalankan *test run*

Pertama yang dilakukan adalah login, pengguna akan masuk ke dalam halaman *Project* setelah *login* menggunakan akun terdaftar. Di mana pada halaman ini dapat membuat *project* baru untuk dilakukan pengujian dengan klik *create new project* untuk mengisi form detail project seperti pada tampilan Gambar 4.4. Project baru akan muncul pada *dashboard Project*, Gambar 4.5 menunjukkan project Demo (template) dan *project* baru.

Create new project

Project name *
For example: Web Application

Project code *
For example: WA

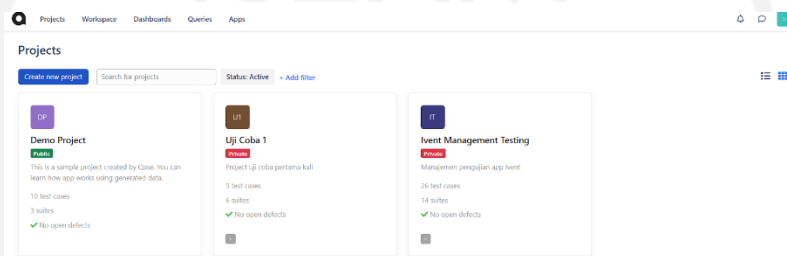
Description
Write a few sentences about your project

Project access type
☒ Private
☐ Public

Member access
☒ Add all members to this project
☐ Add members from specific group
☐ Don't add members

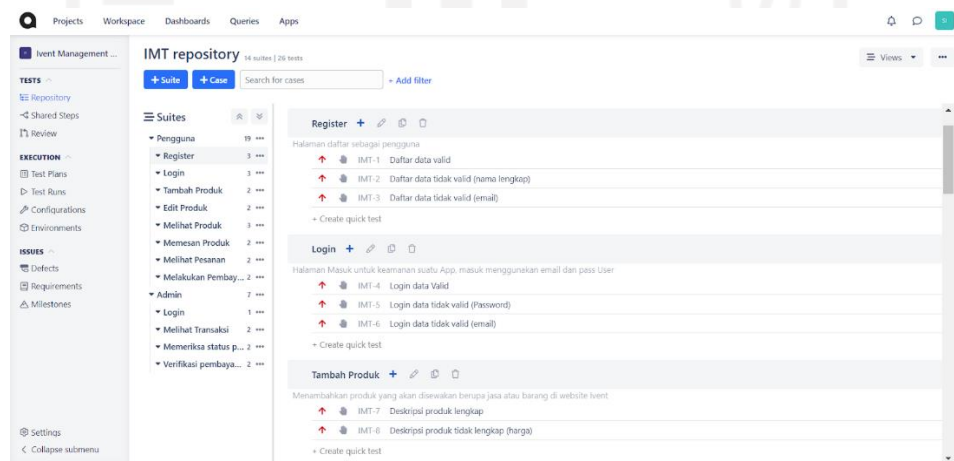
Cancel Create project

Gambar 4.4 Form project baru



Gambar 4.5 Halaman dashboard project

Langkah berikutnya pada penggunaan Qase dilanjutkan dengan membuat rangkaian *test suite* yang akan tersusun rapi sesuai urutan pembuatan pada halaman *repository*. Gambar 4.6 menunjukkan susunan *test suite* yang telah berisi *test case*, di mana terdapat dua *test suite* yaitu pengguna dan admin.



Gambar 4.6 Halaman repository

Langkah selanjutnya yaitu membuat *test case*. Membuat *test case* di dalam test suite cukup mudah dengan mengisi form detail test case pada Gambar 4.7 secara lengkap sesuai kebutuhan. Prioritas *test case* dalam pengujian dapat diatur pada tingkat *high*, *medium*, dan *low*. Selain itu, *test case* juga dapat dikategorikan sebagai perilaku *test case positive*,

negative, dan *destructive*. Hal tersebut nantinya dapat mempermudah penguji dalam mengatur *test case* apabila penguji ingin menampilkan klasifikasi *test case* sesuai kategori maupun seluruhnya pada halaman repository.

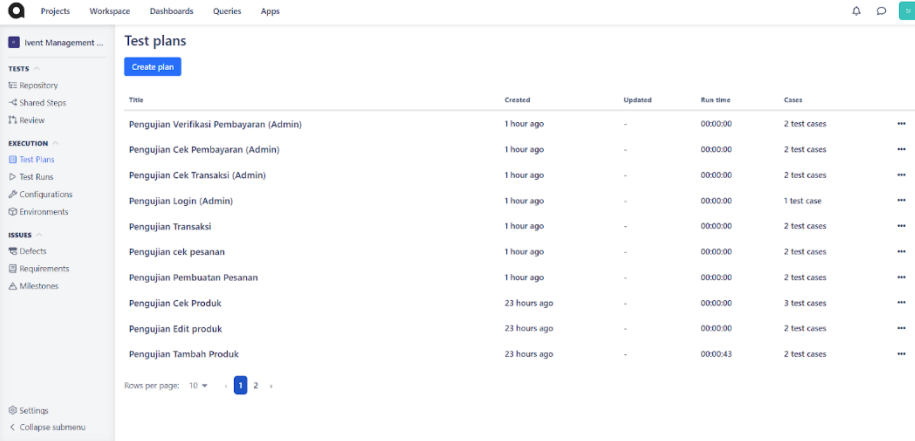
The screenshot shows the 'Create test case' form in a software testing tool. The form is divided into sections: Basic, Conditions, and Pre-conditions. The Basic section includes fields for Title, Status, Description, Suite, Severity, Priority, Type, Layer, Is flaky, Milestone, Behavior, and Automation status. The Conditions section includes Pre-conditions and Post-conditions. The form has a 'Save' button and a 'Save and create another' button.

Gambar 4.7 Halaman membuat *test case*

Setelah langkah pembuatan *test case* dilakukan, berikutnya adalah pembuatan *test plan* dari *test case* yang telah dibuat pada repository. *Test plan* dapat dibuat dengan mengisi *form* detail rencana pada Gambar 4.8 dengan menambahkan *test case* mana saja yang akan diuji. Setelah mengisi detail rencana, *test plan* akan muncul pada halaman *Test Plan* sesuai dengan urutan pembuatan seperti pada Gambar 4.9.

The screenshot shows the 'Create test plan' form in a software testing tool. The form is titled 'Plan details' and includes fields for Title, Description, and a section for selecting test cases. There is an 'Add cases' button and a 'Create plan' button.

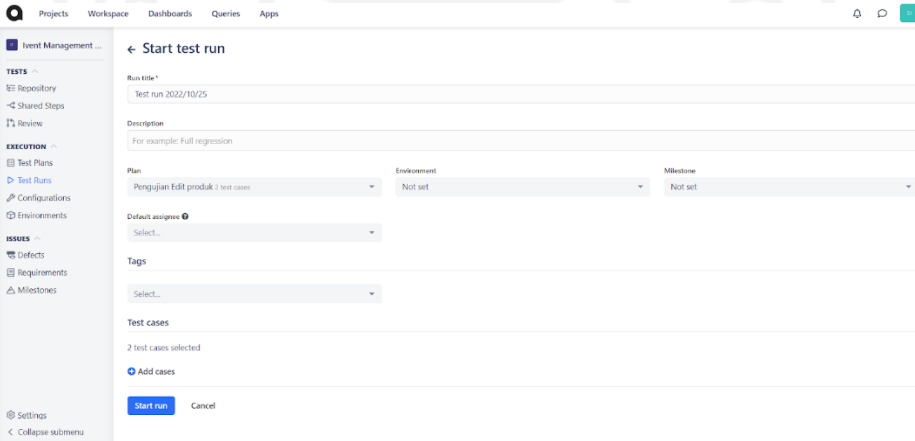
Gambar 4.8 Form detail *test plan*



Title	Created	Updated	Run time	Cases
Pengujian Verifikasi Pembayaran (Admin)	1 hour ago	-	00:00:00	2 test cases
Pengujian Cek Pembayaran (Admin)	1 hour ago	-	00:00:00	2 test cases
Pengujian Cek Transaksi (Admin)	1 hour ago	-	00:00:00	2 test cases
Pengujian Login (Admin)	1 hour ago	-	00:00:00	1 test case
Pengujian Transaksi	1 hour ago	-	00:00:00	2 test cases
Pengujian cek pesanan	1 hour ago	-	00:00:00	2 test cases
Pengujian Pembuatan Pesanan	1 hour ago	-	00:00:00	2 test cases
Pengujian Cek Produk	23 hours ago	-	00:00:00	3 test cases
Pengujian Edit produk	23 hours ago	-	00:00:00	2 test cases
Pengujian Tambah Produk	23 hours ago	-	00:00:43	2 test cases

Gambar 4.9 Halaman *test plan*

Test run dapat dilakukan dengan memilih salah satu *test plan* yang akan diuji dan mengisi form kelengkapannya. Pada *form test run* Gambar 4.10 pengguna dapat mengatur anggota tim yang mengerjakan bagian tersebut. Pengujian dapat dilakukan setelah mengisi *form* dengan klik *start run*.



Start test run

Run title *
Test run 2022/10/25

Description
For example: Full regression

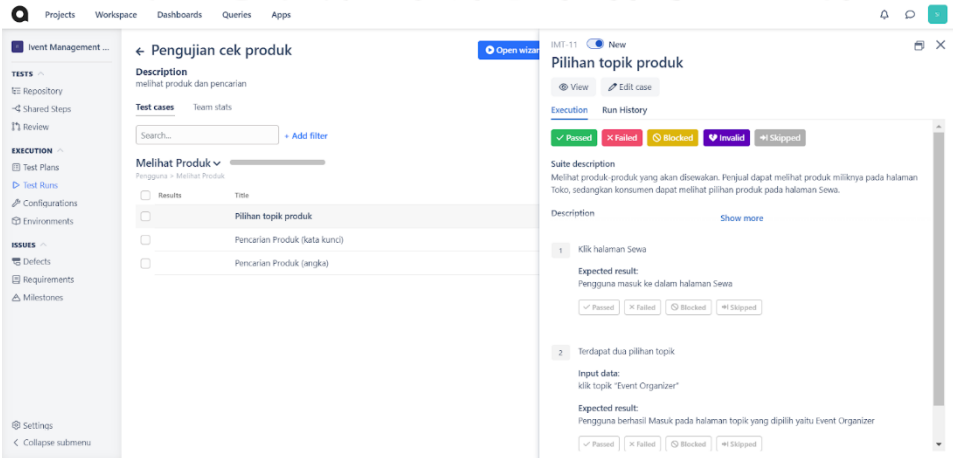
Plan: Pengujian Edit produk 2 test cases | Environment: Not set | Milestone: Not set

Default assignee
Select...

Tags
Select...

Test cases
2 test cases selected
Add cases

Start run Cancel

Gambar 4.10 *Form detail test run*


Penguian cek produk

Description: melihat produk dan pencarian

Test cases: Team stats

Melihat Produk
Pilih topik produk
Pencarian Produk (kata kunci)
Pencarian Produk (angka)

Execution: Run History

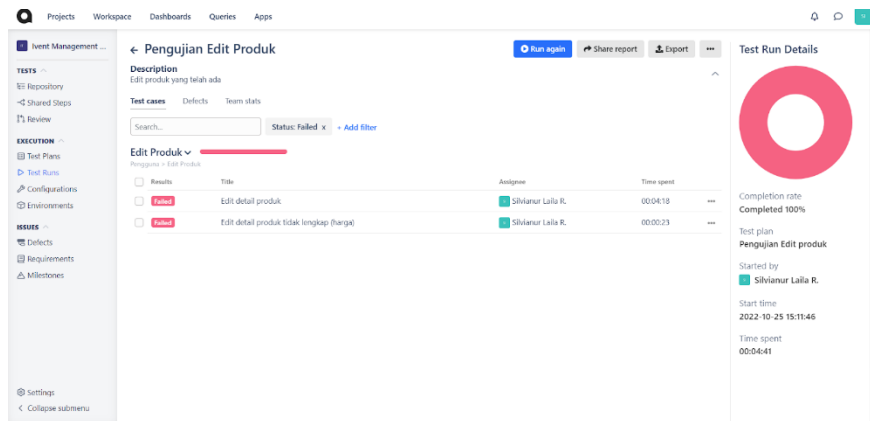
Suite description: Melihat produk-produk yang akan diowakan. Penjual dapat melihat produk miliknya pada halaman Toko, sedangkan konsumen dapat melihat pilihan produk pada halaman Sewa.

1. Klik halaman Sewa
Expected result: Pengguna masuk ke dalam halaman Sewa
Passed

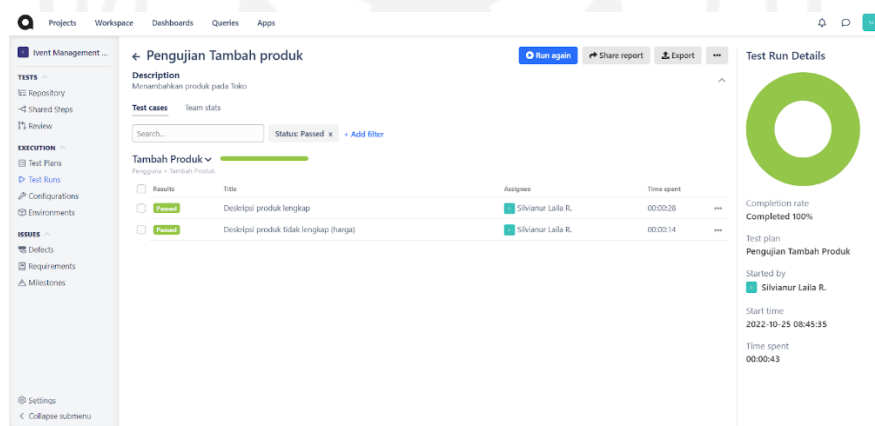
2. Terdapat dua pilihan topik
Input data: klik topik "Event Organizer"
Expected result: Pengguna berhasil Masuk pada halaman topik yang dipilih yaitu Event Organizer
Passed

Gambar 4.11 Halaman eksekusi *test case*

Pengujian dimulai dengan klik *open wizard* agar halaman eksekusi *test case* seperti pada Gambar 4.11 muncul. Pada halaman eksekusi *test case* terdapat beberapa pilihan status test case seperti *pass*, *fail*, *block*, *invalid*, dan *skip*. Setiap step memiliki penilaian statusnya sendiri, *pass* apabila step dapat dilakukan dengan benar dan berhasil. *Fail* apabila step tidak dapat dilakukan atau memiliki hasil yang tidak sesuai dengan yang direncanakan.

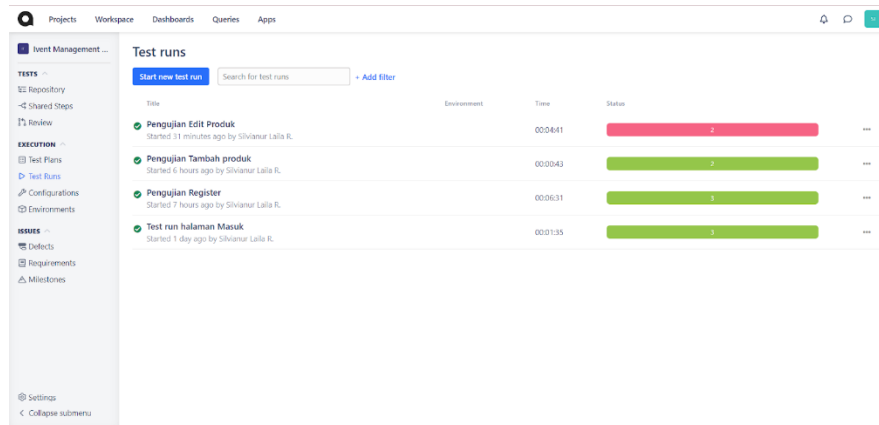


Gambar 4.12 Test run dengan status *fail*



Gambar 4.13 Test run dengan status *pass*

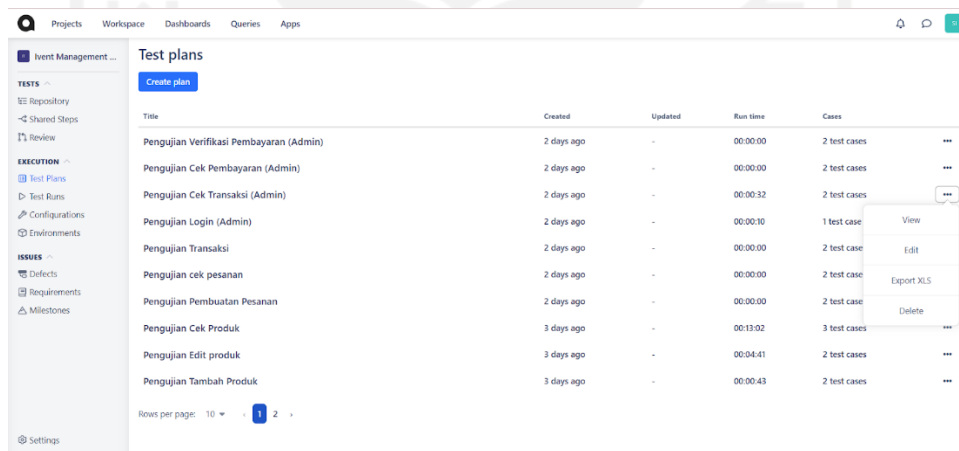
Setelah eksekusi dilakukan pada setiap *test case*, hasil dari eksekusi akan muncul dengan informasi waktu dan status *test case*. Tampilan status *test case* berupa *fail* akan seperti Gambar 4.12 berwarna merah, sedangkan status *test case* berupa *pass* akan seperti Gambar 4.13 berwarna hijau. Selain itu nama penguji yang melakukan pengujian juga muncul pada laporan *test run*. Makna *time spent* pada *test run* adalah ringkasan waktu yang telah dihabiskan untuk *test case* dalam proses *test run*.



Title	Environment	Time	Status
Pengujian Edit Produk Started 31 minutes ago by Silvanur Laila R.		00:04:41	2
Pengujian Tambah produk Started 6 hours ago by Silvanur Laila R.		00:00:43	2
Pengujian Register Started 7 hours ago by Silvanur Laila R.		00:06:31	2
Test run halaman Masuk Started 1 day ago by Silvanur Laila R.		00:01:35	2

Gambar 4.14 Halaman hasil *test run*

Halaman *test run* setelah dilakukan eksekusi pada *test case* yang ada sesuai *test plan* masing-masing dapat dilihat pada Gambar 4.14. *Time* pada halaman bermakna berapa banyak waktu yang telah dihabiskan sejauh ini untuk melakukan eksekusi test case dalam proses ini. Jumlah waktu eksekusi tiap *test run* ditampilkan beserta status pengujian *pass* atau *fail* dengan keterangan jumlah *test case* di dalamnya. Hasil dari eksekusi test case dapat di *export* atau unduh dengan format xls melalui halaman *test plans* Gambar 4.15 dengan memilih salah satu *test plan*.



Title	Created	Updated	Run time	Cases
Pengujian Verifikasi Pembayaran (Admin)	2 days ago	-	00:00:00	2 test cases
Pengujian Cek Pembayaran (Admin)	2 days ago	-	00:00:00	2 test cases
Pengujian Cek Transaksi (Admin)	2 days ago	-	00:00:32	2 test cases
Pengujian Login (Admin)	2 days ago	-	00:00:10	1 test case
Pengujian Transaksi	2 days ago	-	00:00:00	2 test case
Pengujian cek pesanan	2 days ago	-	00:00:00	2 test case
Pengujian Pembuatan Pesanan	2 days ago	-	00:00:00	2 test case
Pengujian Cek Produk	3 days ago	-	00:13:02	3 test cases
Pengujian Edit produk	3 days ago	-	00:04:41	2 test cases
Pengujian Tambah Produk	3 days ago	-	00:00:43	2 test cases

Gambar 4.15 Fitur *export* data pada *test plan*

4.1.3 Fitur Tambahan

Qase memiliki fitur lain yang menjadi pendukung proses manajemen pengujian dan mempermudah penguji yaitu:

1. Mengundang anggota tim
2. Membuat milestones

Manajemen pengujian pada Qase tidak hanya dapat dilakukan sendiri tetapi Qase memiliki fitur untuk mengundang anggota tim baru untuk melakukan eksekusi *test case* bersama. Fitur manajemen tim ini berada pada halaman *workspace* dengan klik *invite* dan mengisi detail anggota baru seperti Gambar 4.16. Undangan akan dikirimkan Qase melalui email yang telah didaftarkan dengan peran yang berbeda. Selain itu anggota juga dapat diatur hanya melihat *project* atau dapat mengakses keseluruhan *project*.

Gambar 4.16 Mengundang anggota baru

User	Status	Type	Role	Role title	Last action
Silvanur Laila R. silvanur.laila@gmail.com	Active	Regular	Owner	Student	5 minutes ago
Fadel 17523187@student.usd.ac.id	Active	Regular	Member	Developer	2 days ago

Gambar 4.17 Halaman *workspace* anggota

Halaman *workspace* akan bertambah anggota apabila email yang dikirimkan telah diterima dan diakses oleh anggota baru. Tampilan halaman pada Gambar 4.17 memuat nama anggota beserta peran anggota dalam *project* Qase. Sehingga pada *test run* anggota baru dapat ditugaskan dalam eksekusi pengujian *test case*.

Berikutnya adalah fitur untuk membuat *milestone* atau dengan kata lain adalah *timeline* yang membantu dalam mengatur estimasi waktu pekerjaan secara umum. Pada Qase *milestone* dibuat sesuai kebutuhan dengan mengisi form seperti pada Gambar 4.18. Setelah membuat *milestones*, nantinya pada halaman fitur *milestones* yang disajikan pada Gambar

4.19 akan muncul sesuai dengan urutan tenggat waktu. *Milestones* dapat ditambahkan pada data pendukung *test case* di halaman *repository*.

Gambar 4.18 *Form* membuat *milestone*

Title	Status	Description	Cases	Due date
Pengujian 2 Created 48 seconds ago Updated 1 second ago	Active	Melakukan pengujian setelah defect diperbaiki	0 test cases	2022-11-13
Pengujian 1 Created 1 minute ago Updated 13 seconds ago	Completed	Melakukan pengujian pertama	0 test cases	2022-11-08

Gambar 4.19 Halaman *milestones*

4.2 Pemantauan dan Pengendalian Pengujian (*Monitoring and Controlling Test*)

Memantau pengujian perlu dilakukan agar pengujian tetap sesuai dengan rencana yang telah dibuat. Hasil pengujian pada aplikasi Ivent menggunakan test case pada Tabel 3.1 yang disesuaikan dengan *use case* masing-masing. Pengujian dilakukan secara manual dengan bantuan Qase dalam manajemen pengujian. Pengujian bertujuan untuk mengetahui apakah fitur yang ada pada aplikasi Ivent telah sesuai dengan *use case* yang direncanakan. Pengendalian hasil pengujian menjadi *feedback* yang berguna untuk memperbaiki kesalahan atau kegagalan fitur. Terdapat 11 fitur yang akan dilakukan pengujian yaitu:

1. Fitur Registrasi
2. Fitur Login
3. Fitur Tambah Produk
4. Fitur Edit Produk
5. Fitur Melihat Produk
6. Fitur Membuat Pesanan
7. Fitur Melihat Pesanan
8. Fitur Melihat Transaksi

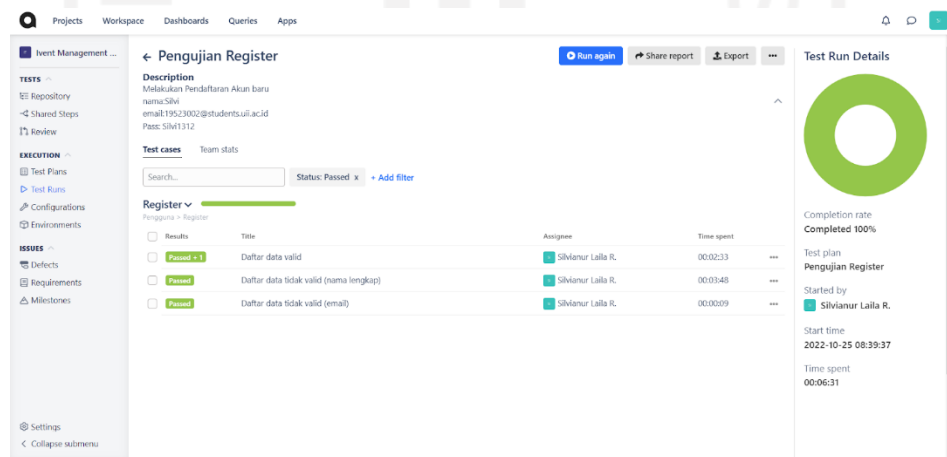
9. Fitur Melakukan Pembayaran
10. Fitur Memeriksa Status Pembayaran
11. Fitur Verifikasi Pembayaran

4.2.1 Fitur Registrasi

Fitur *register* merupakan fitur pertama dalam aplikasi Ivent yang berguna untuk membantu pengguna dalam mendaftarkan akun. Hasil dari pengujian fitur *register* terdapat pada Tabel 4.1 dan Gambar 4.20 yang menggunakan Qase. Waktu yang dihabiskan untuk menguji test case adalah 6 menit 31 detik. Keseluruhan *test case* pada fitur register berstatus *pass* karena keluaran yang diharapkan dengan keluaran sebenarnya sama.

Tabel 4.1 Hasil pengujian fitur *register*

Fitur	ID	Keluaran Sebenarnya	Status
Register	TC 1.1	Pengguna dapat mendaftarkan data dirinya dan data tersimpan.	<i>pass</i>
	TC 1.2	Pengguna tetap berada pada halaman register dan muncul peringatan kesalahan.	<i>pass</i>
	TC 1.3	Pengguna tetap berada pada halaman register dan muncul peringatan kesalahan	<i>pass</i>



Gambar 4.20 Hasil pengujian fitur *register*

4.2.2 Fitur Login

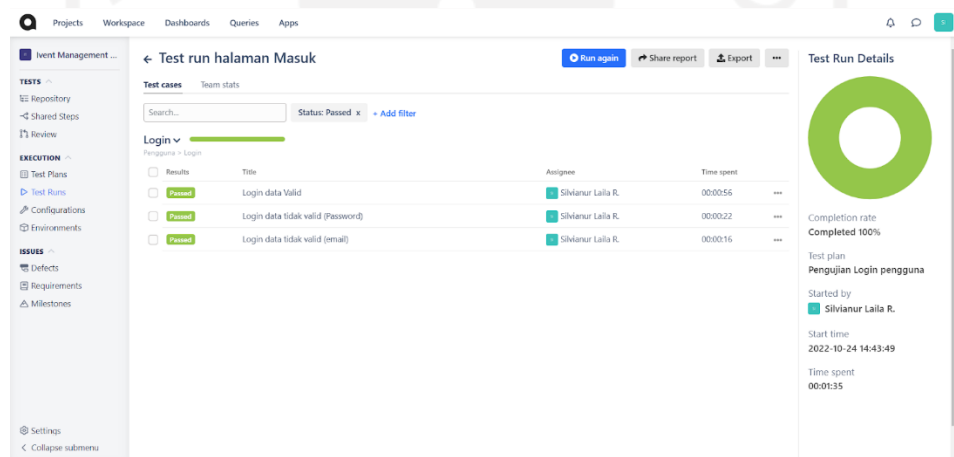
Fitur *login* menjadi fitur kedua pada aplikasi Ivent yang digunakan sebagai fitur keamanan dan pintu masuk pengguna untuk mengakses fitur aplikasi secara menyeluruh. Waktu yang dihabiskan untuk menguji test case adalah 1 menit 35 detik. Keluaran yang diharapkan dari

fitur *login* sama dengan keluaran sebenarnya, sehingga hasil pengujian pada fitur *login* keseluruhannya berstatus *pass* dapat dilihat pada

Tabel 4.2 dan penggunaan Qase di Gambar 4.21.

Tabel 4.2 Hasil Pengujian Fitur Login

Fitur	ID	Keluaran Sebenarnya	Status
Login Pengguna	TC 2.1	Pengguna dapat masuk ke halaman utama Ivent.	<i>pass</i>
	TC 2.2	Pengguna tidak dapat masuk dan tetap pada halaman Login.	<i>pass</i>
	TC 2.3	Pengguna tidak dapat masuk dan tetap pada halaman Login.	<i>pass</i>
Login Admin	TC 2.4	Admin dapat masuk ke halaman beranda admin.	<i>pass</i>



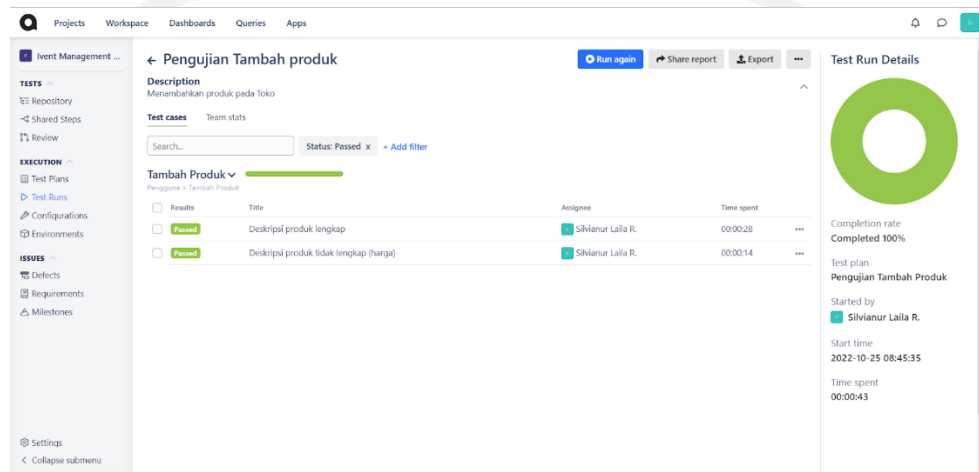
Gambar 4.21 Hasil pengujian fitur *login*

4.2.3 Fitur Tambah Produk

Fitur tambah produk adalah fitur ketiga yang dilakukan pada pengujian aplikasi Ivent. Terdapat 2 *test case* yang memiliki status *pass* dapat dilihat pada Tabel 4.3 dan Gambar 4.22 yang menggunakan Qase tools. Selain itu waktu yang dihabiskan untuk menguji test case adalah 43 detik.

Tabel 4.3 Hasil pengujian fitur tambah produk

Fitur	ID	Keluaran Sebenarnya	Status
Tambah produk	TC 3.1	Pengguna berhasil menambahkan produk.	<i>pass</i>
	TC 3.2	Pengguna tetap pada halaman tambah produk dan muncul peringatan bahwa form ada yang kosong.	<i>pass</i>



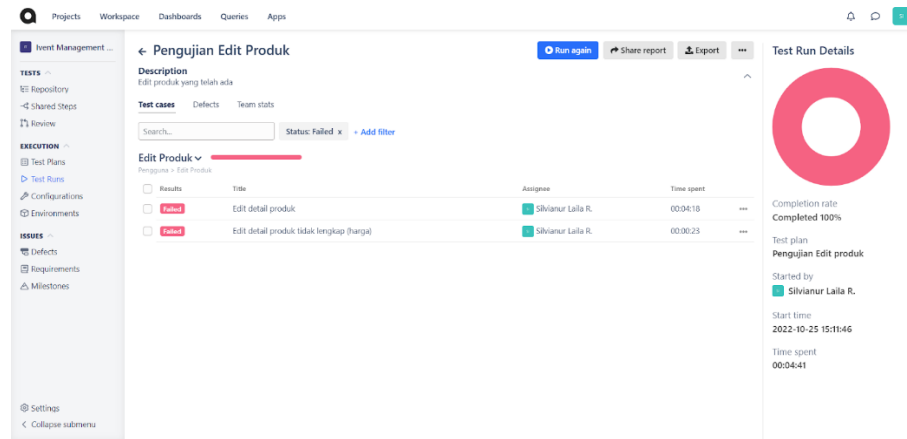
Gambar 4.22 Hasil pengujian fitur tambah produk

4.2.4 Fitur Edit Produk

Berikutnya adalah fitur edit produk pada aplikasi Ivent yang memiliki 2 *test case* untuk dilakukan pengujian. Hasil pengujian berstatus *fail* karena edit produk tidak dapat dilakukan, terdapat kesalahan yang seharusnya pengguna dapat berpindah ke halaman edit produk tetapi produk justru terhapus. Sehingga dapat dilihat pada Tabel 4.4 dan Gambar 4.23 hasil keluaran yang diharapkan tidak sama dengan hasil sebenarnya. Waktu yang dihabiskan untuk menguji test case pada fitur edit produk adalah 4 menit 41 detik.

Tabel 4.4 Hasil pengujian fitur edit produk

Fitur	ID	Keluaran Sebenarnya	Status
Edit produk	TC 4.1	Tidak dapat berpindah ke halaman edit produk, produk terhapus dari sistem.	<i>fail</i>
	TC 4.2		<i>fail</i>



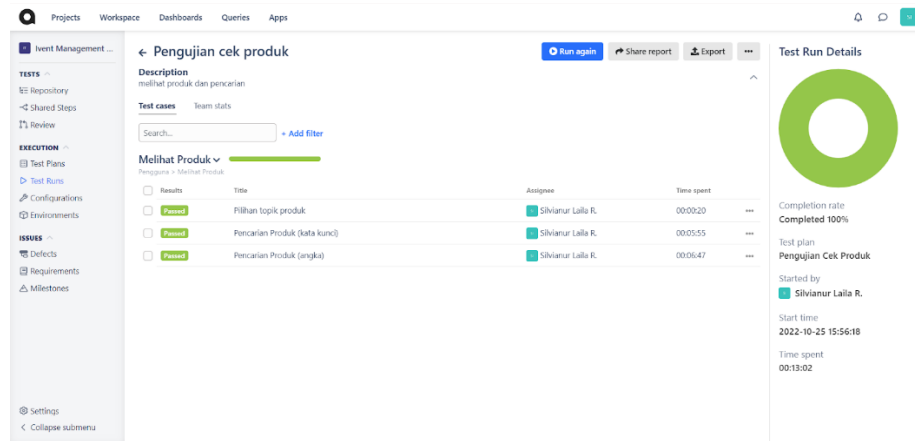
Gambar 4.23 Hasil pengujian fitur edit produk

4.2.5 Fitur Melihat Produk

Fitur melihat produk memiliki 3 test case yang telah dilakukan pengujian. Hasil pengujian ketiga *test case* berstatus *pass*, dapat dilihat pada Tabel 4.5 dan Gambar 4.24 yang menunjukkan hasil pada Qase. Status *pass* pada *test case* dikarenakan hasil keluaran sebenarnya saat pengujian sama dengan hasil yang diharapkan. Waktu yang dihabiskan untuk menguji test case fitur melihat produk adalah 13 menit 2 detik.

Tabel 4.5 Hasil pengujian fitur melihat produk

Fitur	ID	Keluaran Sebenarnya	Status
Melihat Produk	TC 5.1	Pengguna berhasil Masuk pada halaman topik yang dipilih yaitu Event Organizer.	<i>pass</i>
	TC 5.2	Hasil pencarian pengguna muncul sesuai dengan kata yang dicari.	<i>pass</i>
	TC 5.3	Hasil pencarian pengguna muncul sesuai dengan angka yang dicari.	<i>pass</i>



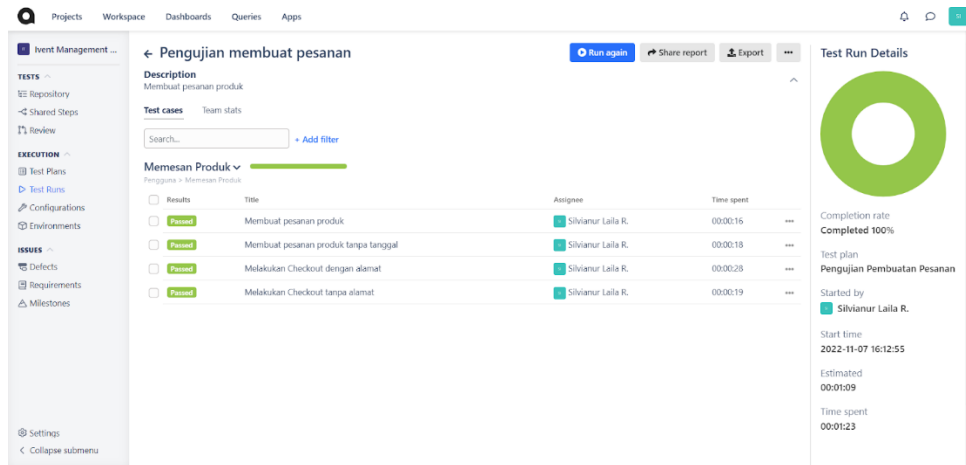
Gambar 4.24 Hasil pengujian fitur melihat produk

4.2.6 Fitur Membuat Pesanan

Selanjutnya adalah fitur membuat pesanan yang memiliki 4 *test case* untuk dilakukan pengujian. Hasil dari pengujian pada keempat *test case* berstatus *pass* karena keluaran yang diharapkan sama dengan keluaran sebenarnya. Hal tersebut dapat dilihat pada Tabel 4.6 dan Gambar 4.25. Waktu yang dihabiskan untuk menguji test case fitur membuat pesanan adalah 1 menit 23 detik.

Tabel 4.6 Hasil pengujian fitur membuat pesanan

Fitur	ID	Keluaran Sebenarnya	Status
Membuat Pesanan	TC 6.1	Berhasil menambahkan pesanan pada halaman Pesanan.	<i>pass</i>
	TC 6.2	Tetap berada pada halaman Produk dan terdapat peringatan kesalahan.	<i>pass</i>
	TC 6.3	Berhasil membuat alamat pesanan untuk pengiriman.	<i>pass</i>
	TC 6.4	Tetap berada pada halaman Pesanan dan terdapat peringatan kesalahan	<i>pass</i>



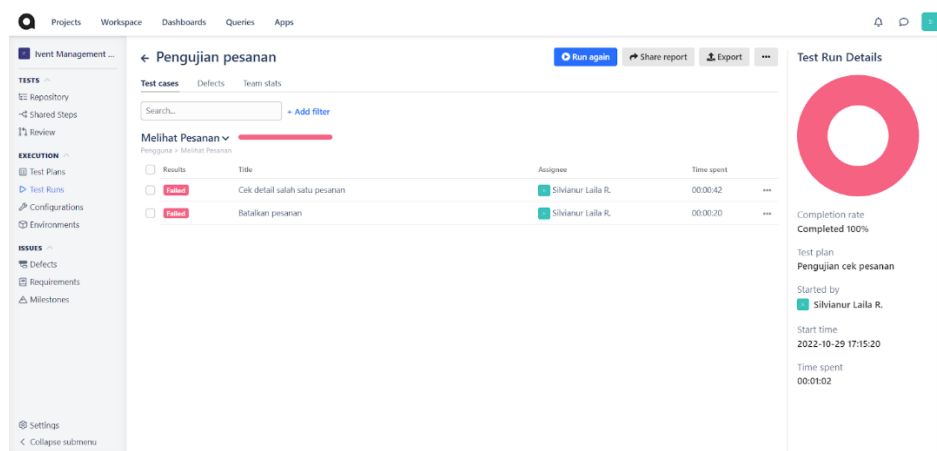
Gambar 4.25 Hasil pengujian fitur membuat pesanan

4.2.7 Fitur Melihat Pesanan

Fitur melihat pesanan pada aplikasi Ivent memiliki 2 *test case* yang berstatus *fail*. Hal tersebut dikarenakan tidak dapat berpindah ke halaman *cart* pesanan, baik melalui bar menu atau *dropdown* pada profil. Hasil pengujian dapat dilihat pada Tabel 4.7 dan Gambar 4.26. Waktu yang dihabiskan untuk menguji test case fitur melihat pesanan adalah 1 menit 2 detik.

Tabel 4.7 Hasil pengujian fitur melihat pesanan

Fitur	ID	Keluaran Sebenarnya	Status
Melihat Pesanan	TC 7.1	Tidak dapat berpindah ke halaman Cart Pesanan, halaman tidak tersedia muncul 404 <i>not found</i> .	<i>fail</i>
	TC 7.2		<i>fail</i>



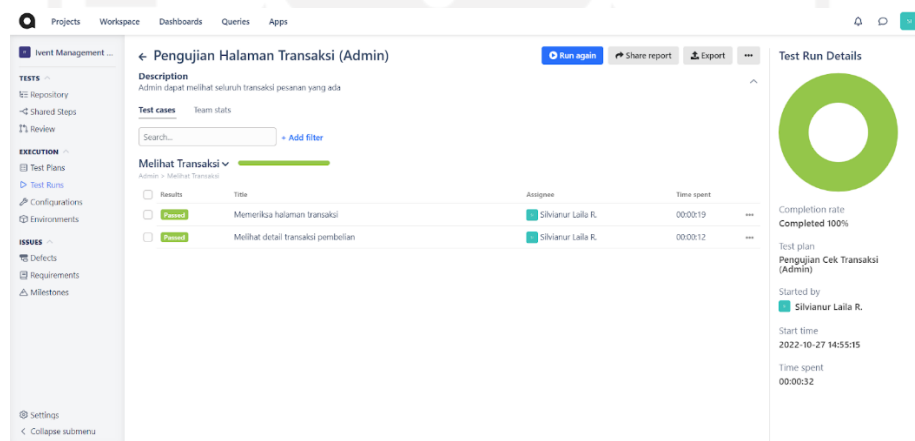
Gambar 4.26 Hasil pengujian fitur melihat pesanan

4.2.8 Fitur Melihat Transaksi

Fitur melihat transaksi pada aplikasi Ivent yang dilakukan pengujian ini adalah sudut pandang sebagai admin, sehingga perlu *login* menggunakan akun admin. Fitur ini memiliki 2 *test case* berstatus *pass* dapat dilihat pada Tabel 4.8 dan Gambar 4.27, di mana hasil keluaran sebenarnya sama dengan hasil keluaran yang diharapkan. Waktu yang dihabiskan untuk menguji test case fitur melihat transaksi adalah 32 detik.

Tabel 4.8 Hasil pengujian fitur melihat transaksi

Fitur	ID	Keluaran Sebenarnya	Status
Melihat Transaksi	TC 8.1	Berhasil masuk halaman transaksi dan melihat keseluruhan transaksi.	<i>pass</i>
	TC 8.2	Ditampilkan detail transaksi yang dipilih.	<i>pass</i>



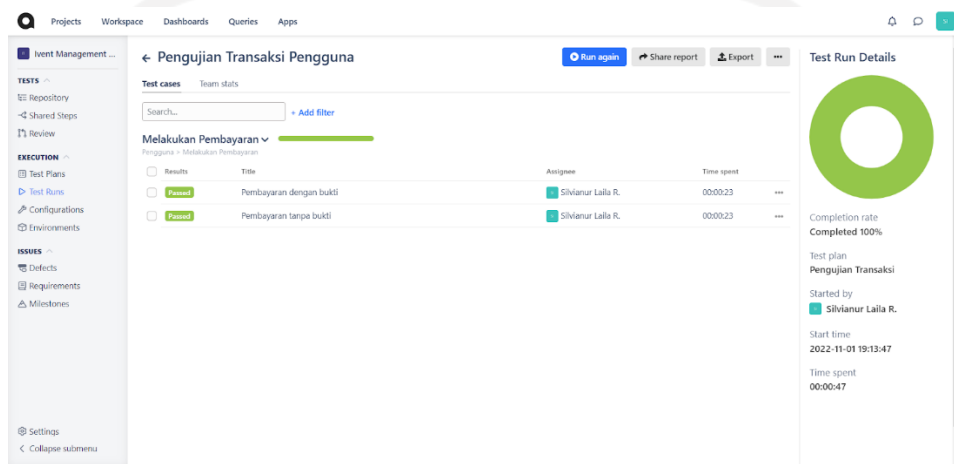
Gambar 4.27 Hasil pengujian fitur melihat transaksi

4.2.9 Fitur Melakukan Pembayaran

Selanjutnya fitur yang dilakukan pengujian adalah fitur melakukan pembayaran. Fitur ini memiliki 2 *test case* dengan status *pass* dapat dilihat pada Tabel 4.9 dan Gambar 4.28 yang menggunakan Qase. Waktu yang dihabiskan untuk menguji test case fitur melakukan pembayaran adalah 47 detik.

Tabel 4.9 Hasil pengujian fitur melakukan pembayaran

Fitur	ID	Keluaran Sebenarnya	Status
Melakukan Pembayaran	TC 9.1	Pengguna berhasil melakukan pembayaran dan mendapatkan bukti bayar.	<i>pass</i>
	TC 9.2	Pengguna gagal melakukan pembayaran dan tetap pada halaman pembayaran.	<i>pass</i>



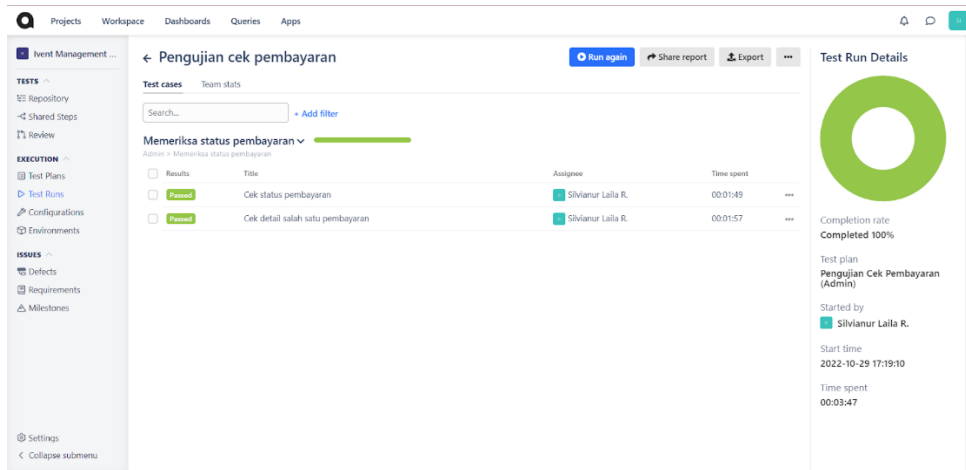
Gambar 4.28 Hasil pengujian fitur melakukan pembayaran

4.2.10 Fitur Memeriksa Status Pembayaran

Fitur memeriksa status pembayaran yang dilakukan oleh admin memiliki 2 *test case* dengan status *pass* karena hasil yang diharapkan sama dengan hasil sebenarnya. Hasil pengujian dapat dilihat pada Tabel 4.10 dan Gambar 4.29. Waktu yang dihabiskan untuk menguji test case fitur memeriksa status pembayaran adalah 3 menit 47 detik.

Tabel 4.10 Hasil pengujian fitur memeriksa status pembayaran

Fitur	ID	Keluaran Sebenarnya	Status
Memeriksa Status Pembayaran	TC 10.1	Admin melihat seluruh daftar transaksi konsumen.	<i>pass</i>
	TC 10.2	Admin melihat detail salah satu transaksi	<i>pass</i>



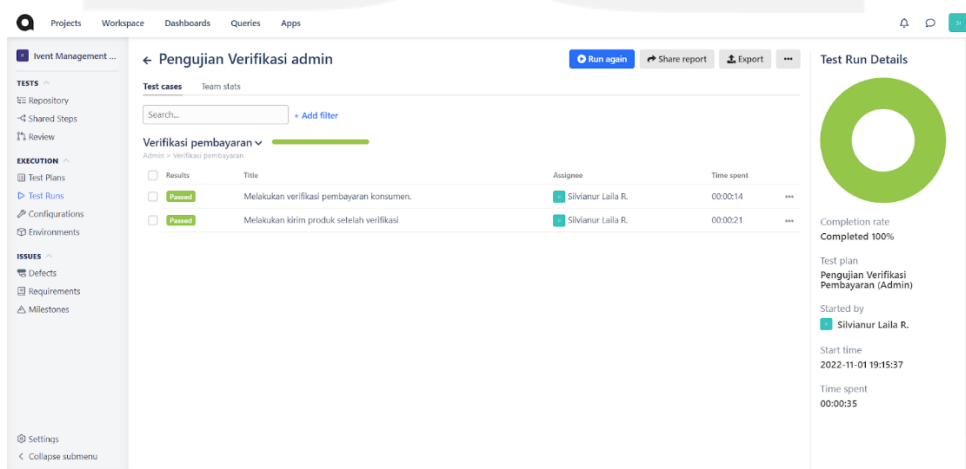
Gambar 4.29 Hasil pengujian fitur memeriksa status pembayaran

4.2.11 Fitur Verifikasi Pembayaran

Pengujian fitur terakhir adalah verifikasi pembayaran dari pembayaran yang telah dilakukan pembeli. Fitur ini memiliki 2 *test case* dengan hasil keluaran sebenarnya sama dengan keluaran yang diharapkan, sehingga fitur verifikasi pembayaran berstatus *pass*. Hal tersebut dapat dilihat pada Tabel 4.11 dan Gambar 4.30. Waktu yang dihabiskan untuk menguji test case fitur verifikasi pembayaran adalah 35 detik.

Tabel 4.11 Hasil pengujian fitur verifikasi pembayaran

Fitur	ID	Keluaran Sebenarnya	Status
Verifikasi Pembayaran	TC 11.1	Transaksi pesanan konsumen telah diverifikasi.	<i>pass</i>
	TC 11.2	Transaksi pesanan konsumen dikirim.	<i>pass</i>



Gambar 4.30 Hasil pengujian fitur verifikasi pembayaran

4.3 Manajemen Insiden

Pada pengujian pertama yang dilakukan pada 28 *test case* dari 11 fitur aplikasi Ivent dihasilkan 9 fitur berstatus *pass* dan 2 fitur berstatus *fail*. Fitur yang berstatus *fail* yaitu fitur edit produk dan fitur melihat pesanan. Hal tersebut dikarenakan terjadi kesalahan seperti pada fitur edit produk justru menjadikan produk terhapus dan fitur melihat pesanan tidak dapat berpindah halaman atau muncul 404 *Not found*. Dari 28 *test case* terdapat 4 *test case fail* dan 24 *test case pass*, dapat dilihat pada Tabel 4.12 yang merangkum keseluruhan pengujian.

Tabel 4.12 Hasil keseluruhan pengujian *test case*

No.	Fitur	Jumlah Test Case	Jumlah <i>pass</i>	Jumlah <i>fail</i>
1	Register	3	3	0
2	Login Pengguna	3	3	0
	Login Admin	1	1	0
3	Tambah Produk	2	2	0
4	Edit Produk	2	0	2
5	Melihat Produk	3	3	0
6	Membuat Pesanan	4	4	0
7	Melihat Pesanan	2	0	2
8	Melihat Transaksi	2	2	0
9	Melakukan Pembayaran	2	2	0
10	Transaksi pesanan konsumen dikirim	2	2	0
11	Verifikasi Pembayaran	2	2	0
Total		27	24	4

4.3.1 Tindak Lanjut

Berdasarkan hasil pengujian pertama, akan dilakukan perbaikan pada *test case* yang berstatus *fail*. Perbaikan dilakukan sesuai dengan prioritas pada *test case* dengan kategori *high*, *medium*, dan *low*. Daftar prioritas *test case* yang berstatus *fail* pada pengujian aplikasi Ivent disajikan pada Tabel 4.13 dengan skenario singkat.

Keterangan:

High: Prioritas tinggi, *bug* yang harus segera diperbaiki. Biasanya disebabkan *requirement* yang tidak tampil pada web browser.

Medium: Prioritas sedang, *bug* yang dapat dipertimbangkan dalam perbaikannya (segera atau tidak). Biasanya disebabkan oleh *alert* saat pengguna melakukan kesalahan tidak muncul atau sejenisnya.

Low: Prioritas rendah, *bug* yang tidak harus segera diperbaiki atau perbaikannya dapat ditunda karena kesalahan kecil pada *user interface*.

Tabel 4.13 Daftar *test case* berstatus *fail*

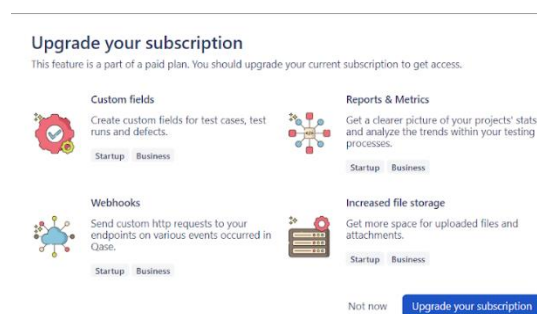
ID	Fitur	Skenario	Prioritas
TC 4.1	Edit Produk	Melakukan edit detail produk dengan mengisi form detail produk yang ingin diganti, hasilnya pengguna berhasil melakukan edit produk.	<i>High</i>
TC 4.2		Menghapus harga produk dan mengganti dengan huruf, hasilnya pengguna tetap pada halaman edit dan muncul peringatan kesalahan isi form.	<i>Medium</i>
TC 7.1	Melihat Pesanan	Melakukan klik halaman Pesanan pada drop down profil atau navbar menu, sehingga pengguna berhasil melihat pesanan dan detail deskripsi.	<i>High</i>
TC 7.2		Membatalkan pesanan pada salah satu pesanan yang ada, hasilnya pengguna berhasil membatalkan pesanan yang dipilih.	<i>High</i>

Perbaikan pada fitur edit produk dilakukan karena saat pengujian pertama fitur edit produk justru menghapus produk yang akan diedit. Perbaikan dilakukan dengan cara mengganti *source code* yang mengalami kesalahan. Setelah *code* diperbaiki dan dijalankan kembali, menunjukkan bahwa fitur edit produk berhasil untuk melakukan edit produk.

Perbaikan pada fitur melihat pesanan dilakukan dengan membuat akun baru sebanyak 5 kali dengan menambahkan beberapa produk tambahan yang berbeda ke dalam *database*. Kesalahan terjadi karena *value* pada *database* tidak terdeteksi sehingga muncul *404 not found* pada saat ingin melihat pesanan. Setelah *database* terisi produk dan beberapa pengguna, fitur melihat pesanan sudah dapat diakses baik melalui bar menu atau *dropdown* profil.

4.3.2 Kendala Penggunaan Tools

Kendala dalam penggunaan Qase adalah tutorial *step by step* penggunaan fitur pada Qase yang masih sedikit. Selain itu dalam pencarian artikel maupun jurnal yang berkaitan dengan Qase *tools* cukup sulit, karena Qase merupakan alat manajemen kasus uji yang belum populer untuk digunakan. Penggunaan Qase yang gratis membuat beberapa fitur pada Qase tidak dapat digunakan seperti *requirement*, *review*, *reports* dan *metrics*. Perlu dilakukan *upgrade* berlangganan agar bisa memanfaatkan fitur tersebut seperti diperlihatkan pada Gambar 4.31, di mana setiap kelebihan fitur Qase yang ditawarkan memiliki harga yang berbeda-beda.



Gambar 4.31 Upgrade langganan Qase

4.3.3 Manfaat Penggunaan Tools

Manfaat penggunaan Qase selain gratis menurut penulis adalah fitur manajemen pengujian yang dimiliki Qase cukup lengkap untuk mengatur dan mengelola pengujian menjadi lebih terstruktur. Selain itu Qase menyediakan fitur untuk mengundang anggota tim baru pada *project* yang dibuat. Sehingga anggota baru tersebut dapat ditugaskan pada beberapa *test plan* untuk melakukan pengujian. Kemudian Qase juga memberikan fitur penyimpanan *defect* atau *bug* untuk memudahkan saat melakukan *tracking* pada saat terjadi *retest*.

Manfaat tersebut telah dikonfirmasi kepada praktisi suatu perusahaan yang menggunakan Qase dalam proses pengujian sebagai QA. Bahwa manfaat penggunaan Qase yang didapat penulis sama dengan yang didapatkan praktisi. Akan tetapi terdapat perbedaan tambahan yaitu praktisi dapat menjalankan seluruh fitur Qase karena praktisi menggunakan Qase versi berbayar.

Berdasarkan hal tersebut, walaupun fitur Qase belum sepenuhnya dapat digunakan akibat berbayar. Qase tetap mampu mengatur dan mengelola pengujian agar terstruktur serta data yang tersimpan memudahkan acuan pengujian berikutnya. Disajikan pada Tabel 4.14 kelebihan dan kekurangan Qase secara menyeluruh.

Tabel 4.14 Kelebihan dan Kekurangan Qase

Kelebihan	Kekurangan	Solusi
Qase sendiri merupakan sistem manajemen pengujian modern (<i>test management system</i>) yang memang digunakan oleh QA.	Terdapat beberapa fitur yang penggunaannya perlu upgrade dan dilakukan pembayaran seperti <i>requirement</i> , <i>report</i> , dan <i>metrics</i> .	Membuat secara manual untuk laporan keseluruhan. Setiap fitur masing-masing hasilnya dapat diexport dengan format csv atau pdf.
Fitur utama dapat diakses secara gratis melalui website. Pengguna tidak perlu menginstall apapun di dekstop.	Mengisi banyak detail test case secara manual (apabila menggunakan pengujian manual)	Fitur <i>repository</i> dapat <i>import</i> data <i>test case</i> dari berbagai alat bantu.
Mampu menyimpan banyak <i>test case</i> dengan detail pada <i>repository</i> .		
Fitur utama yang memudahkan <i>developer</i> atau QA yaitu <i>test case management</i> , <i>test plan</i> , <i>test run</i> , <i>dashboards</i> , <i>team management</i> , <i>integration</i> , <i>export-import</i> data, dan lain lain.		
Memiliki kemudahan untuk melakukan <i>tracking</i> pada <i>test case</i> yang sama apabila terjadi <i>re test</i> . Karena terdapat fitur <i>defect</i> yang menyimpan kesalahan atau kegagalan pada <i>test case</i> .		
Terdapat fitur <i>team management</i> , sehingga memudahkan anggota tim diassign pada setiap aktivitas.		
Dapat berintegrasi dengan alat populer seperti Jira, Trello, Github, dan lain lain.		
Qase mampu digunakan dalam pengujian otomatis maupun manual.		

4.4 Laporan Pengujian

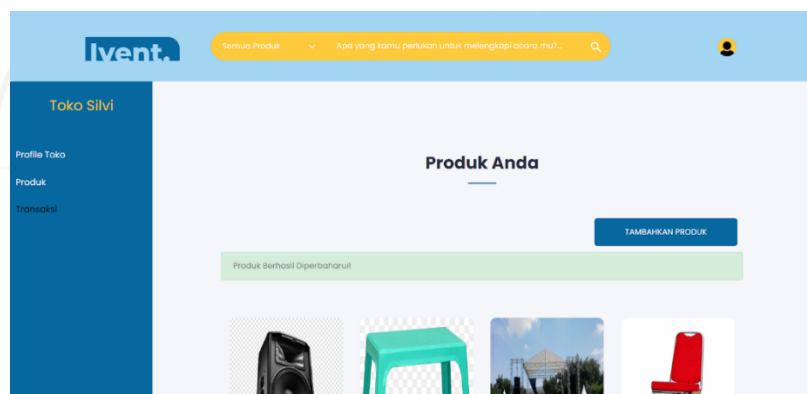
Pengujian kedua dilakukan setelah perbaikan pada test case yang mengalami kegagalan atau *fail* saat pengujian pertama. Perbaikan dilakukan oleh pengembang agar fungsi fitur aplikasi Ivent dapat berjalan sesuai dengan kebutuhan yang telah direncanakan. Hasil pengujian kedua dapat dilihat pada Tabel 4.15.

Tabel 4.15 Hasil pengujian kedua

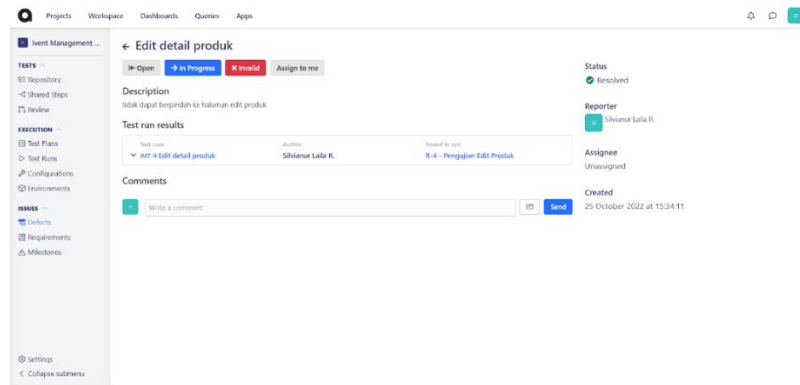
Fitur	ID	Keluaran Sebenarnya	Status
Edit produk	TC 4.1	Pengguna berhasil mengubah produk.	<i>pass</i>
	TC 4.2	Pengguna tetap pada halaman edit dan kolom harga dibuat khusus angka saja	<i>pass</i>
Melihat Pesanan	TC 7.1	Pengguna berhasil melihat pesanan dan detail pesanan lainnya	<i>pass</i>
	TC 7.2	Pengguna dapat menghapus salah satu pesanan yang dipilih	<i>pass</i>

Perbaikan hasil pengujian kedua fitur edit produk menghasilkan status *pass* karena hasil yang diharapkan sesuai dengan hasil sebenarnya. Keberhasilan pengujian pada TC 4.1 dapat dilihat pada

Gambar 4.32, di mana produk meja telah berhasil diperbaharui dan tidak mengalami kegagalan lagi. Waktu yang dihabiskan untuk menguji test case fitur edit produk adalah 1 menit 42 detik.

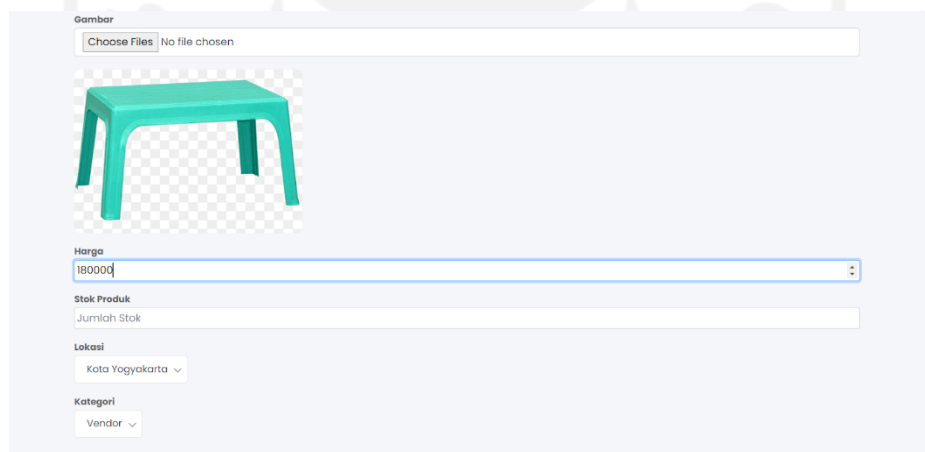


Gambar 4.32 Hasil pengujian TC 4.1 fitur edit produk

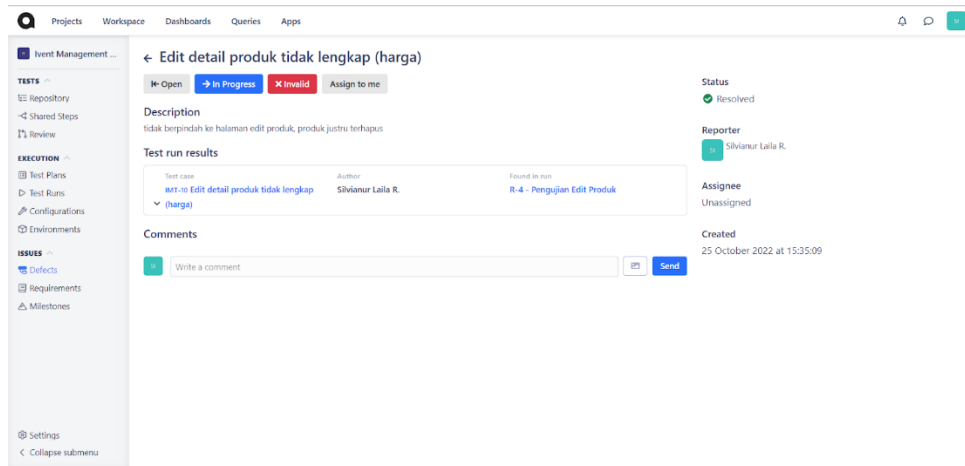


Gambar 4.33 *Defect* TC 4.1 pada Qase

Selanjutnya keberhasilan pengujian pada TC 4.2 disajikan pada Gambar 4.34 yang menunjukkan kolom harga hanya dapat diisi angka, sehingga pengguna tidak dapat mengisi menggunakan huruf. Penyelesaian *defect* fitur edit produk di Qase, diselesaikan dengan status *resolved* ditampilkan pada Gambar 4.33 dan Gambar 4.35.

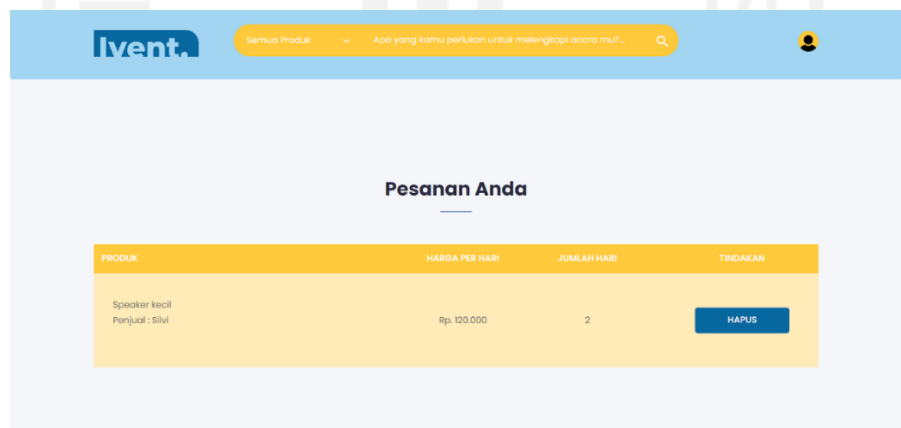


Gambar 4.34 Hasil pengujian TC 4.2 fitur edit produk

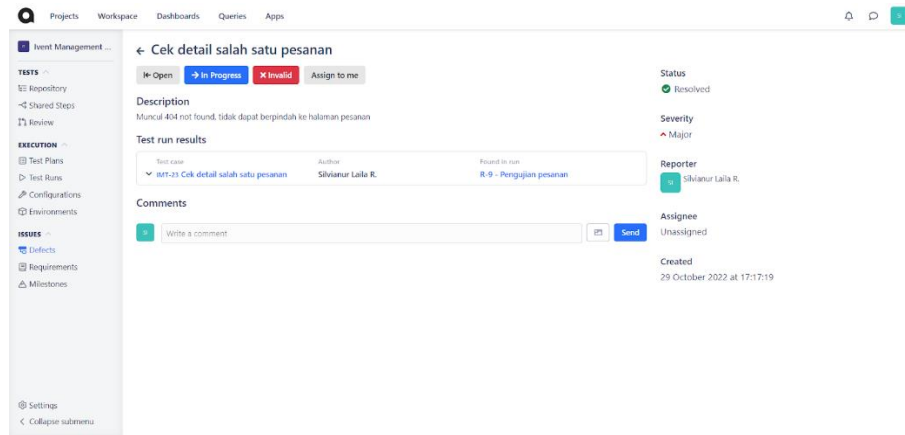


Gambar 4.35 Defect TC 4.2 pada Qase

Perbaikan fitur melihat pesanan dilakukan karena saat pengujian pertama halaman pesanan tidak dapat diakses dengan muncul pesan *error 404 not found*. Sehingga dilakukan perbaikan agar pengguna dapat mengakses halaman pesanan untuk melihat pesanan yang telah masuk ke keranjang. Keberhasilan pengujian pada TC 7.1 dapat dilihat pada Gambar 4.36 yang menunjukkan halaman pesanan. Penyelesaian *defect* TC 7.1 di Qase dengan status *resolved* yang telah disajikan pada Gambar 4.37. Waktu yang dihabiskan untuk menguji test case fitur melihat pesanan adalah 22 detik.

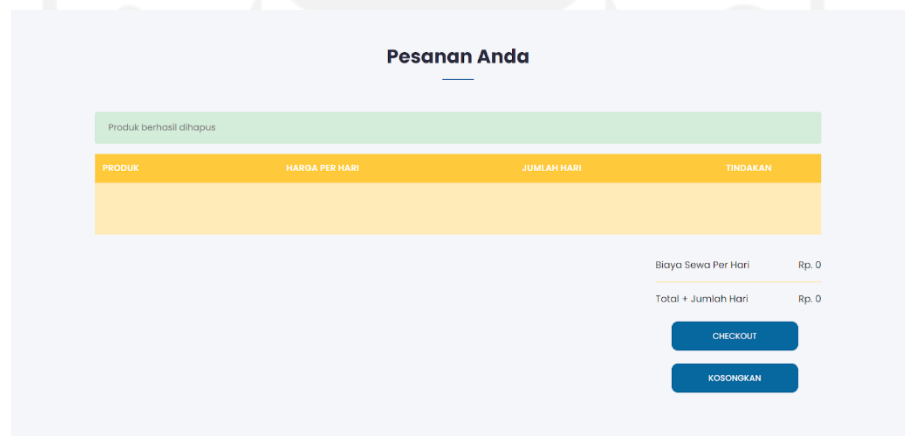


Gambar 4.36 Hasil pengujian TC 7.1 fitur melihat pesanan

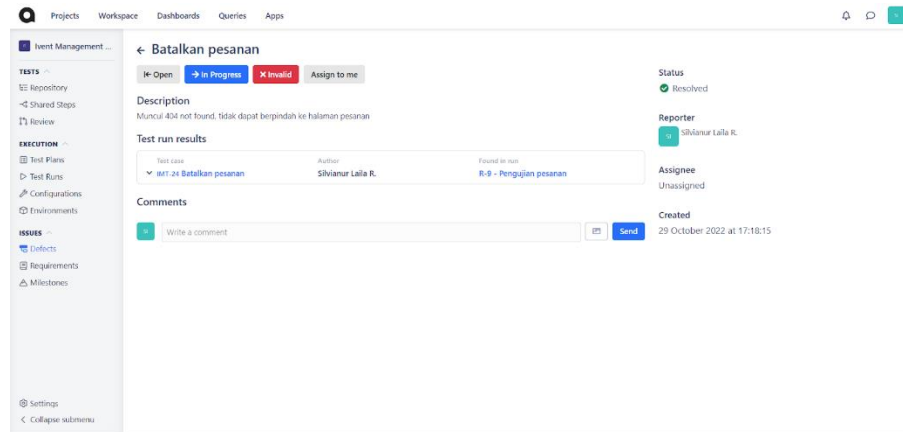


Gambar 4.37 Defect TC 7.1 pada Qase

Berikutnya keberhasilan pengujian pada TC 7.2 ditunjukkan pada Gambar 4.38 bahwa pesanan yang dipilih dapat dihapus dari halaman pesanan. Penyelesaian *defect* TC 7.2 di Qase disajikan pada Gambar 4.39 dengan status *resolved* karena telah berhasil dilakukan perbaikan.



Gambar 4.38 Hasil pengujian TC 7.2 fitur melihat pesanan



Gambar 4.39 Defect TC 7.2 pada Qase

Setelah dilakukan pengujian keseluruhan fitur terutama fitur yang mengalami kegagalan atau *fail* yaitu fitur edit produk dan melihat pesanan, maka hasil pengujian dirangkum pada Tabel 4.16. Sehingga dapat dilihat terdapat 11 fitur berhasil dilakukan pengujian dengan status *pass*.

Laporan Hasil Pengujian

Nama Project: Ivent Management Testing

Jumlah Test Suite: 14

Jumlah Test Case: 28

Total Pengujian: 2 kali

Tanggal Mulai Pengujian: 10 Oktober 2022

Tanggal Selesai Pengujian: 12 November 2022

Tabel 4.16 Rangkuman hasil keseluruhan pengujian

No.	Test Suite	Jumlah Test Case	Pengujian Pertama	Pengujian kedua
Pengguna				
1	Register	3	3	3
2	Login Pengguna	3	3	3
	Login Admin	1	1	1
3	Tambah Produk	2	2	2

4	Edit Produk	2	2	2
5	Melihat Produk	3	3	3
6	Membuat Pesanan	4	4	4
7	Melihat Pesanan	2	2	2
8	Melakukan Pembayaran	2	2	2
Admin				
9	Melihat Transaksi	2	2	2
10	Transaksi pesanan konsumen dikirim	2	2	2
11	Verifikasi Pembayaran	2	2	2
Total		28	28	28

No.	Pengujian Test Suite	Jumlah Test Case	Time Spent
1	Pengujian Register	3	6 menit 31 detik
2	Pengujian Login Pengguna	3	1 menit 35 detik
	Pengujian Login Admin	1	10 detik
3	Pengujian Tambah Produk	2	43 detik
4	Pengujian Edit Produk (1)	2	4 menit 41 detik
	Pengujian Edit Produk (2)	2	1 menit 42 detik
5	Pengujian Melihat Produk	3	13 menit 2 detik
6	Pengujian Membuat Pesanan	4	1 menit 23 detik
7	Pengujian Melihat Pesanan (1)	2	1 menit 2 detik
	Pengujian Melihat Pesanan (2)	2	22 detik
8	Pengujian Pembayaran	2	47 detik
9	Pengujian Melihat Transaksi	2	32 detik
10	Pengujian Cek Transaksi	2	3 menit 47 detik
11	Pengujian Verifikasi Pembayaran	2	35 detik
Total Waktu yang Dhabiskan			35 menit 52 detik

Keterangan:

(1) adalah pengujian pertama

(2) adalah pengujian kedua

Pengujian pada aplikasi Ivent telah selesai dilakukan dengan bantuan Qase io. Qase io sangat membantu dalam manajemen pengujian dengan beberapa fitur yang disediakan secara gratis seperti fitur *repository*, *test plan*, *test run*, *defect*, dan *invite team*. Sehingga keseluruhan tahap pengujian dapat dilakukan menjadi satu di Qase dengan memanfaatkan fitur yang tersedia tersebut.

Pengujian diawali dengan membuat *test case* pada *repository* terlebih dahulu, lalu membuat *test plan* dengan menyesuaikan *test case* yang telah dibuat. *Test plan* digunakan untuk melakukan *test run* atau bagian eksekusi pengujian, bagian ini dapat menunjukkan *test case* berstatus *pass* atau *fail*. Apabila terdapat *test case* dengan status *fail* secara otomatis akan masuk ke dalam fitur *defect* untuk dilakukan perbaikan agar berstatus *resolved*.

Keseluruhan 28 *test case* berhasil dilakukan dengan 2 kali pengujian dan terdapat tahap perbaikan. Pada pengujian pertama terdapat 24 *test case* yang berstatus *pass* dan 4 *test case* berstatus *fail*. *Test case* yang berstatus *fail* dilakukan perbaikan berdasarkan prioritas *high* dan *medium*. Setelah dilakukan perbaikan, dilanjutkan dengan melakukan pengujian kedua dan hasil yang didapatkan keseluruhan 28 *test case* berstatus *pass*.

Total waktu yang dihabiskan untuk menguji keseluruhan *test case* adalah 35 menit 52 detik. Berdasarkan status setiap fitur dan total waktu yang dihabiskan, terdapat beberapa fitur yang masih mengalami *loading* cukup lama. Namun tetap dapat digunakan dan diakses sebagaimana mestinya fitur tersebut telah direncanakan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan dari penelitian yang telah dilakukan terkait manajemen pengujian perangkat lunak pada aplikasi Ivent adalah sebagai berikut:

- a. Pelaksanaan pengujian aplikasi Ivent telah berhasil dilakukan dengan menggunakan alat bantu manajemen pengujian Qase.
- b. Manajemen pengujian perangkat lunak dilakukan dengan melalui 7 tahap, yaitu analisis risiko dan pengujian, perencanaan dan estimasi pengujian, pendekatan pengujian, organisasi pengujian, pemantauan dan kontrol pengujian, manajemen insiden, dan pembuatan laporan pengujian.
- c. Qase digunakan dalam manajemen pengujian melalui beberapa langkah yaitu membuat *project*, membuat *test suite*, membuat *test case*, membuat *test plan*, dan menjalankan *test run*.
- d. Alat bantu Qase membantu manajemen pengujian menjadi lebih terstruktur dan mudah dilakukan. Selain itu Qase membantu menyimpan *defect* atau *bug* pada tiap fitur yang mengalami kegagalan. Sehingga hasil temuan tersebut dapat diserahkan ke pengembang untuk diperbaiki. Proses pengujian membuat penulis mengetahui cara kerja, pengelolaan, dan proses pengerjaan pada alat manajemen pengujian Qase yang diakses secara gratis untuk mengelola pengujian perangkat lunak.
- e. Pengujian dilakukan terhadap 11 *usecase* dengan membuat 28 skenario *test case* untuk menguji fungsionalitas dari *usecase* yang diimplementasikan pada fitur aplikasi Ivent.
- f. Pengujian dilakukan 2 kali dengan pengujian pertama menghasilkan 24 *test case* yang berstatus pass atau berhasil. Lalu pengujian kedua dilakukan setelah proses perbaikan dan menghasilkan 4 *test case* berstatus pass atau berhasil. Sehingga keseluruhan 28 *test case* berstatus *pass* dengan waktu total yang dihabiskan 35 menit 52 detik.
- g. Berdasarkan hasil keseluruhan pengujian membantu pengembang untuk meningkatkan kualitas dari perangkat lunak yang dikembangkan dengan memperbaiki *defect* yang ditemukan.

5.2 Saran

Manajemen pengujian menggunakan Qase masih belum maksimal karena aplikasi Ivent masih dalam tahap pengembangan dan belum sepenuhnya selesai. Sehingga *test case* yang dapat dibuat hanya sedikit dan masih terdapat fitur Qase yang belum digunakan karena berbayar. Diharapkan penelitian di masa mendatang dapat menggunakan studi kasus yang lebih rumit agar penggunaan Qase lebih maksimal dan hasil terlihat nyata. Selain itu perlu lebih banyak mengkaji referensi tentang pengujian perangkat lunak, alat bantu pengujian, dan metode manajemen pengujian yang digunakan.



DAFTAR PUSTAKA



- Abdulrahman Mahmoud, R. V. (2019). Minotaur: Adapting Software Testing Techniques For Hardware Errors. *ASPLOS '19: Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, 1087-1103.
- Adila Chusnul Fatiyah, S. F. (2019). Pengujian Fungsional dan Non Fungsional Aplikasi Web BorongAjaYuk. *e-Proceeding of Engineering*, 8415-8421.
- Bagus Bayu Sasongko, F. M. (2021). Pengujian Blackbox Menggunakan Teknik Equivalence. *Jurnal ICTEE*, 10-16.
- Baidhya, E. (2019). An Optimum Test Management System. *TURKU UNIVERSITY OF APPLIED SCIENCES*, 1-50.
- Bayu Dwi Juniansyah, E. R. (2020). Pembuatan E-Commerce Pemesanan Jasa Event Organizer Untuk Zero Seven Entertainment. *Jurnal Tekno Kompak*, 41-46.
- Brian Hambling, P. M. (2010). *Software Testing - An ISTQB-ISEB Foundation Guide Second Edition*. Swindon, Uk: British Informatics Society Limited.
- Dani, D. N. (2021). Strategi Ide Bisnis Startup E-commerce Ivent Menggunakan Metode Design Thinking. *Universitas Islam Indonesia*.
- Devaldi Akbar Suryadi, E. S. (2022). Evaluation of Information Quality Using ISO/IEC 25010:2011 (Case Research: Menu Harianku Application). *International Journal Of Innovation In Enterprise System*, 06, No. 02, 119-132.
- Dorothy Graham, E. v. (2006). *Foundations of Software Testing : ISTQB Certification*. Thomson.
- Eka Chandra Ramdhani, J. E. (2018). Sistem Informasi Penyewaan Peralatan Event Organizer Berbasis Web pada PT. Adecon Jakarta. *Jurnal Pengabdian Kepada Masyarakat BSI*, 390-397.
- Fadhila Cahya Ningrum, D. S. (2019). Pengujian Black Box pada Aplikasi Sistem Seleksi Sales Terbaik. *Jurnal Informatika Universitas Pamulang*, 125-130.
- Fournier, G. (2008). *Essential Software Testing A Use Case Approach*. New York: CRC Press.
- Handayani, S. (2018). Perancangan Sistem Informasi Penjualan Berbasis E-Commerce Studi Kasus Toko Kun Jakarta. *ILKOM Jurnal Ilmiah*, 182-189.

- Hariyanto, T. D. (2020). Software Quality Assurance pada Perusahaan Pengembang Perangkat Lunak Skala Kecil dan Menengah. *Jurnal Riset Teknologi dan Inovasi Pendidikan (JARTIKA)*, 283-294.
- Hesham Alaqail, S. A. (2018). Overview of Software Testing Standard ISO/IEC/IEEE 29119. *International Journal of Computer Science and Network Security*, Vol. 18 No.2.
- Idris, M. A. (2021). Perancangan User Interface & User Experience Pada Aplikasi E-Commerce Ivent Menggunakan Metode User Centere Design (Ucd). *Universitas Islam Indonesia*.
- Ioan Mihnea Iacob, R. C. (2008). Testing: First Step Towards Software Quality. *Applied Quantitative Methods*(Quality in Statistic), 241-253.
- Kartiko, C. (2019). Evaluasi Kualitas Aplikasi Web Pemantau Menggunakan Model Pengujian Perangkat Lunak ISO/IEC 9126. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, 8(No. 1), 16-21.
- Kshirasagar Naik, P. T. (2008). *Software Testing and Quality Assurance : Theory and Practice*. Hoboken, New Jersey: y John Wiley & Sons, Inc.
- Michal Pawlak, A. P.-M. (2018). Software Test Management Approach for Agile Environments. *Information Systems in Management Vol. 7 (1)*, 47–58.
- Mucjal, A. A. (2021). Pengembangan e-commerce vendor dan event organizer berbasis website dengan metode waterfall. *Universitas Islam Indonesia*.
- Muhammad Andryan Wahyu Saputra, W. A. (2021). Survei Teknik-Teknik Pengukuran Kualitas Perangkat Lunak. *Journal of Computer Science and Applied Informatics*, 11-29.
- Rajeshwari Sissodia, A. K. (2017). Analysis of Automation and Manual Testing. *International Journal of Scientific & Engineering Research*, 8(10), 355-357.
- Rajkumar. (2022, June 11). *Qase Test Case Management Tool Review Tutorial 2022*. Diambil kembali dari Software Testing Material: <https://www.softwaretestingmaterial.com/qase-test-case-management-tool/>
- Riyanthi Angrainy Sianturi, A. M. (2021). Perancangan Pengujian Fungsional dan Non Fungsional Aplikasi. *Jurnal Komputer dan Informatika*, 133-141.
- Rizky Parlika, D. C. (2020). Studi Literatur Perbandingan Antara Metode LOC, COCOMO, FPA dalam Ranah Software Metric. *Jurnal Pendidikan Informatika dan Sains*, 66-74.
- Sandra Ayu, A. L. (2020). Peran e-commerce terhadap perekonomian Indonesiaselama pandemi Covid-19. *Jurnal Kajian Manajemen Bisnis*, 114-123.

- Taufik Hidayat, M. M. (2020). Pengujian Sistem Informasi Pendaftaran dan Pembayaran Wisuda Online menggunakan Black Box Testing dengan Metode Equivalence Partitioning dan Boundary Value Analysis. *Jutis (Jurnal Teknik Informatika)*, 25-29.
- Vivi Afifah, D. S. (2021). Rancangan Sistem Pemilihan dan Penetapan Harga dalam Proses Pengadaan Barang dan Jasa Logistik Berbasis Web. *Jurnal IKRA-ITH INFORMATIKA*, Vol. 5 No. 2.



LAMPIRAN

Surat Permohonan Izin	
<p>Kepada Yth:</p> <p>Bapak/Ibu Pengembang Aplikasi Ivet</p> <p>Di Tempat</p> <p>Dengan hormat,</p> <p>Saya yang bertanda tangan dibawah ini:</p> <p>Nama : Silvianur Laila Ramadhani</p> <p>NIM : 19523002</p> <p>Jurusan : Informatika</p> <p>Fakultas : Teknologi Industri</p> <p>Mengajukan permohonan izin untuk menggunakan, melakukan pengujian dan melakukan perbaikan pada aplikasi Ivet berbasis website dalam penelitian saya berjudul “Manajemen Pengujian Perangkat Lunak”</p> <p>Demikian surat permohonan ini saya buat, atas perhatiannya saya ucapkan terima kasih.</p> <p>Menyetujui/Tidak menyetujui (dapat dicoret salah satu)</p> <div style="display: flex; justify-content: space-between; align-items: flex-end;"> <div style="text-align: center;"> <p>Pengembang Aplikasi Ivet,</p>  <p>Aikal Alfrian Mucjal</p> </div> <div style="text-align: center;"> <p>Hormat saya,</p>  <p>Silvianur Laila Ramadhani</p> </div> </div>	

الجامعة الإسلامية
الاستاذ الدكتور