

# Manajemen Pengujian Perangkat Lunak Menggunakan Aplikasi Qase

Silvianur Laila Ramadhani  
Jurusan Informatika Fakultas Teknologi Industri  
Universitas Islam Indonesia  
Yogyakarta  
19523002@students.uui.ac.id

Hanson Prihantoro Putro  
Jurusan Informatika Fakultas Teknologi Industri  
Universitas Islam Indonesia  
Yogyakarta  
hanson@uui.ac.id

**Abstract**—Pada proses pengembangan dilakukan peningkatan kualitasnya dengan menguji, menemukan kecacatan, dan memperbaiki kecacatan tersebut hingga produk dianggap baik untuk dirilis. Permasalahan fitur yang gagal selalu muncul ketika perusahaan telah merilis produk perangkat lunak tersebut. Apalagi banyaknya waktu yang dibutuhkan dalam melakukan pengujian, membuat banyak perusahaan mengeluarkan biaya lebih mahal dengan lamanya estimasi waktu pengerjaan. Perlu dilakukan manajemen dalam pengujian agar terstruktur dan menghemat waktu. Manajemen pengujian perangkat lunak memiliki bagian penting untuk membantu dalam mengatur segala proses pengujian perangkat lunak sehingga terstruktur dan sesuai dengan rencana. Bagian penting yang ada pada manajemen pengujian yaitu bagian perencanaan dan bagian eksekusi. Qase merupakan aplikasi manajemen pengujian berbasis *cloud* secara gratis yang menjadi alat bantu dalam manajemen pengujian perangkat lunak sebelum dirilis. Qase memiliki beberapa fitur pendukung yaitu *test case management*, *test plans and runs*, *team management*, dan *integration*. Qase dipilih karena dapat menyimpan defect dari *test case* yang mengalami kegagalan. Selain itu, Qase membuat data pengujian lebih terstruktur dan mudah dipahami. Aplikasi Ivent 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*. Setelah menggunakan Qase manfaat yang didapat yaitu Qase mampu membantu dalam mengatur dan mengelola pengujian agar lebih terstruktur. Walaupun terdapat beberapa fitur pada Qase yang belum dapat digunakan akibat berbayar.

**Keywords**—Manajemen pengujian perangkat lunak, Blackbox testing, Qase, e-Commerce, Aplikasi Ivent

## I. PENDAHULUAN

Melalui pengujian *Quality Assurance* (QA), dapat dilakukan penjaminan kualitas perangkat lunak. Tujuan dari jaminan kualitas adalah untuk menyediakan perangkat lunak yang memiliki *bug* minimum dan memenuhi tingkat fungsionalitas, keandalan, dan kinerja yang telah ditentukan [1] Pengujian perangkat lunak diperlukan untuk membuat perangkat lunak yang andal dan berkualitas tinggi. Pengujian perangkat lunak adalah proses menjalankan program perangkat lunak untuk menemukan kesalahan [2].

Ivent adalah aplikasi *e-commerce* berbasis *website* dengan menggunakan model bisnis yang menghubungkan pemilik bisnis dengan penyewa vendor atau *event organizer*. Aplikasi ini memiliki dua kelompok pengguna yaitu pemilik bisnis dan penyewa *event organizer* atau vendor. Aplikasi Ivent ada karena solusi dari permasalahan

ekonomi pada sektor industri kreatif yang mengalami kemunduran dan kesusahan dalam memasarkan produk akibat Covid19.

Aplikasi Ivent memudahkan pemilik bisnis *event organizer* atau vendor untuk menawarkan produk maupun jasa kepada penyewa secara luas melalui internet. Aplikasi Ivent memiliki beberapa fitur seperti tampilan berbagai macam produk, pemesanan produk, tambah produk, dan lain lain. Berdasarkan fitur tersebut penyewa dapat memesan produk atau jasa yang ditawarkan pemilik bisnis dari mana saja dengan mudah. pengujian terlebih dahulu untuk menemukan *bug*, agar aplikasi Ivent dapat diluncurkan dengan hasil yang baik.

Perusahaan yang mengembangkan perangkat lunak, diharuskan untuk melakukan tahap peningkatan kualitas produk perangkat lunak sebelum digunakan pengguna. Permasalahan sering muncul saat perusahaan telah merilis perangkat lunak, kurang maksimalnya fitur yang ada membuat kualitas yang dimiliki oleh produk menurun. Oleh karena itu, kualitas produk yang mengalami masalah perlu dilakukan evaluasi dengan pengujian [3].

Saat melakukan pengujian perangkat lunak, perusahaan perlu melakukan manajemen pengujian untuk menghindari *over budget* akibat proses panjang tahap pengujian. Manajemen pengujian memiliki dua bagian penting yaitu bagian perencanaan yang berisi tahap perencanaan dan estimasi pengujian, tahap organisasi pengujian, serta tahap analisis risiko dan pengujian. Lalu bagian eksekusi yang berisikan tahap *monitoring* dan *controlling*, manajemen pengujian diperlukan alat manajemen pengujian perangkat lunak yang dapat melakukan pengelolaan pengujian, manajemen insiden, dan lain sebagainya [4].

Qase menjadi salah satu alat bantu yang digunakan dalam manajemen pengujian perangkat lunak. Qase merupakan sebuah sistem manajemen kasus uji berbasis *cloud* [5]. Tampilan UI sederhana yang dimiliki Qase mampu memudahkan pemula dalam melakukan manajemen pengujian perangkat lunak dan dapat diakses secara gratis. Pengerjaan pada Qase dapat dilakukan bersama-sama dengan mengundang tim dan dapat diberi label sesuai dengan posisi.

Qase dipilih karena memiliki perbedaan cukup signifikan dibandingkan dengan alat bantu manajemen yang umum digunakan seperti Trello. Selain itu, Qase memiliki fitur untuk menambahkan jenis prioritas dan data penting lainnya pada *test case*. Qase dapat menyimpan *defect*, disesuaikan dengan *test case* mana yang mengalami

kegagalan. Sehingga dapat dikatakan Qase membantu pengujian menjadi lebih terstruktur dan mudah untuk dipahami orang lain.

Dalam penelitian ini akan dilakukan manajemen proses pengujian perangkat lunak yang dilakukan pada sebuah aplikasi. Proses manajemen pengujian dilakukan dengan alat bantu Qase yang nantinya membantu dalam pembuatan rencana pengujian. Apabila nanti ditemukan bug pada proses pengujian, Qase menyimpan semua *bug* atau *defect* untuk diserahkan kepada *developer* agar diperbaiki. Sehingga diharapkan dengan menggunakan Qase dapat membantu untuk melakukan manajemen pengujian pada perangkat lunak agar kualitas perangkat lunak yang dikembangkan menjadi lebih baik.

## II. KAJIAN PUSTAKA

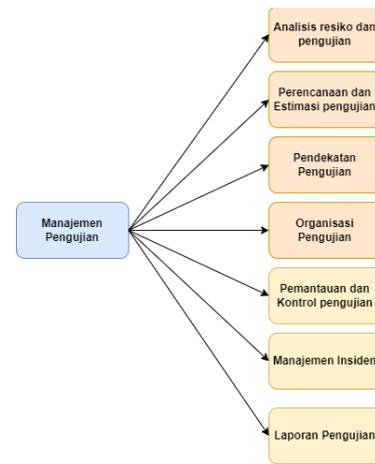
### A. Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah proses untuk mengevaluasi dan meningkatkan kualitas perangkat lunak. Pengujian menjadi peran penting dalam mencapai kualitas perangkat lunak yang direncanakan. Selama dalam tahap pengembangan, kualitas produk perangkat lunak ditingkatkan melalui pengujian-penemuan *bug*-perbaikan *bug* hingga produk dianggap baik untuk dirilis. Terdapat berbagai istilah yang sering muncul di dalam literatur tentang pengujian perangkat lunak, seperti *failure* (kegagalan), *error* (kesalahan), *fault* (kesalahan), dan *defect* (kecacatan) yang memiliki arti serupa tetapi tetap memiliki perbedaan tersendiri [6].

Sebagai kajian pustaka, sebuah artikel mengembangkan Minotaur sebagai alat baru untuk analisis ketahanan dan meningkatkan kecepata [7]. Adanya kesamaan konseptual antara analisis ketahanan modern dan pengujian perangkat lunak menjadi ide adanya Minotaur. Artikel tersebut menjelaskan bahwa kasus uji pada pengujian perangkat lunak merupakan input dan *output* yang direncanakan. Berdasarkan hal tersebut dapat digunakan dalam menentukan sistem yang diuji apakah telah memenuhi beberapa tujuan dari pengujian perangkat lunak yang dikembangkan.

### B. Manajemen Pengujian Perangkat Lunak

Manajemen pengujian perangkat lunak dibagi menjadi beberapa bagian penting dalam manajemen pengujian. Pembagian proses manajemen pengujian terlihat pada Gambar 1 berupa bagian perencanaan yang terdapat organisasi pengujian, perencanaan dan estimasi pengujian, serta analisis resiko dan pengujian. Kemudian terdapat bagian eksekusi yang terdiri dari manajemen konfigurasi, *monitoring* dan *controlling* pengujian, manajemen permasalahan atau insiden [8].



Gambar 1. Proses Manajemen Pengujian

Sebuah artikel mengevaluasi sistem dengan cara manual atau otomatis untuk memverifikasi bahwa sistem memenuhi persyaratan yang telah ditentukan sebelum pengujian [9]. Manajemen dan kontrol pengujian yang ada pada artikel tersebut menggunakan tiga tahap utama yaitu perencanaan pengujian, pelaksanaan pengujian, dan manajemen cacat. Tahap pengujian perlu dilakukan perencanaan dan secara terstruktur dipantau untuk memastikan sesuai dengan semua jadwal proyek. Tujuan dari artikel ini adalah menunjukkan prosedur, metode, dan pendekatan yang digunakan dalam manajemen dan kontrol pengujian. Hal tersebut terdiri dari tahap penting seperti mendeskripsikan metodologi pengujian secara rinci untuk perencanaan pengujian dan menetapkan prosedur yang akan dilakukan dalam proses eksekusi pengujian.

Sebuah artikel menganalisis tentang *Kungfu Testing* yaitu pendekatan pengujian berdasarkan saran dan praktik terbaik yang dikemukakan oleh para ahli di bidang pengujian [10]. Pendekatan *Kungfu Testing* memberikan langkah-langkah dalam melaksanakan kegiatan pengujian layanan *online* teladan 'Erasmus Booking'. Tahapan yang digunakan dimulai dari analisis risiko, perencanaan pengujian, estimasi pengujian, organisasi pengujian, *monitoring* dan *controlling* pengujian, manajemen masalah atau manajemen masalah, dan terakhir adalah laporan pengujian.

### C. Alat pengujian Perangkat Lunak

Pengujian perangkat lunak dibagi menjadi dua kategori yaitu analisis statis dan analisis dinamis [6]. Alat analisis statis mendukung pengujian dalam fase pengkodean. Alat analisis dinamis digunakan dalam proses pengujian fungsional, pengujian integrasi, dan pengujian unit. Alat-alat tersebut mengevaluasi sistem ketika perangkat lunak berjalan dan dikelompokkan berdasarkan kegiatan pengujian. Contohnya adalah alat pengujian yang mendukung aktivitas manajemen, alat untuk mendukung pengujian statis, dan lain sebagainya.

Alat manajemen pengujian dapat digunakan di seluruh siklus hidup pengembangan perangkat lunak. Pada penggunaannya, manajer pengujian akan menggunakan alat manajemen pengujian untuk mengelola pengujian pada sistem [4]. Fitur atau karakteristik alat manajemen pengujian mencakup dukungan untuk melakukan manajemen pengujian, membuat jadwal pengujian,

mengatur aktivitas pengujian, menelusuri proses pengujian dan mencatat hasil pengujian.

Sebuah artikel menjabarkan tujuan untuk mengembangkan sistem manajemen pengujian yang dapat menyelesaikan permasalahan dan sesuai dengan kriteria perusahaan [11]. Artikel diawali dengan pengetahuan teori tentang topik permasalahan, lalu memeriksa proses pengujian, menentukan alat manajemen pengujian yang sesuai dan akhirnya membuat sistem pada alat yang dipilih. Alat manajemen pengujian yang dipilih adalah TestLink, dikarenakan setelah melakukan peninjauan menyeluruh ditemukan bahwa TestLink lebih baik daripada menggunakan sistem lama perusahaan. TestLink merupakan salah satu alat manajemen pengujian perangkat lunak *open source* berbasis web yang sering digunakan oleh penguji.

#### D. Qase Tools

Qase adalah aplikasi manajemen kasus uji (*test case management*) berbasis *cloud* yang digunakan oleh pengembang dan penguji [5]. 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*.

Qase memudahkan penguji untuk mengimpor kasus uji dari alat manajemen pengujian populer seperti TestLink, Jira, Testrail dan lain sebagainya serta mengeksport kasus uji dari Qase ke berbagai format seperti XLSX, CSV, dan lain sebagainya [12]. Sehingga dalam penggunaan Qase penguji dapat menyampaikan masalah penting yang ditemukan selama pengujian secara langsung ke pengembang dalam waktu singkat untuk penyelesaiannya.

Sebuah artikel yang menggunakan model ISO/IEC 25010:2011, bertujuan untuk mengetahui kualitas produk aplikasi Menu harianku dan memberikan rekomendasi peningkatan kualitas aplikasi [13]. Model tersebut memanfaatkan 6 karakteristik yaitu *Usability, Performance Efficiency, Functional Suitability, Reliability, Portability, dan Security*. Pengujian dilakukan dengan metode *blackbox* dengan menggunakan alat bantu pengelolaan pengujian Qase. Pengujian dilakukan dengan mengidentifikasi permasalahan yang ada, mengkaji kualitas desain, menguji kualitas performa, dan membuat rekomendasi perbaikan apabila terdapat kecacatan.

### III. METODE PENELITIAN

Penelitian ini dilakukan dengan menggunakan metode manajemen pengujian yang melalui 7 tahapan berurutan. Gambar 1 menunjukkan gambar metode penelitian. Pertama, sebelum menjalankan pengujian terlebih dahulu dilakukan analisis resiko apa saja yang dapat mempengaruhi proses pengujian perangkat lunak. Risiko digunakan untuk memutuskan di mana harus memulai pengujian dalam siklus pengembangan perangkat lunak dan mengurangi risiko cacat yang terjadi.

Dalam langkah selanjutnya, perencanaan pengujian menjadi tahap penting yang perlu dilakukan dalam setiap proyek uji. Hal ini berisi daftar tugas dan batasan dalam

melaksanakan proses pengujian. Perencanaan pengujian dilakukan saat pengembangan dan implementasi pengujian serta pemeliharaan kegiatan. Dokumen yang dihasilkan dari perencanaan pengujian disebut dengan *project test plan*. Estimasi pengujian memiliki dua pendekatan yaitu berbasis ahli dan metrik. Pendekatan ahli bersikap subjektif, sedangkan pendekatan metrik berdasarkan data yang ada.

Langkah ketiga yaitu menentukan pendekatan pengujian. Pendekatan pengujian merupakan implementasi strategi pengujian pada proyek, termasuk semua keputusan tentang bagaimana pengujian akan dilaksanakan untuk seluruh program kerja berdasarkan tujuan, sasaran pengujian dan penilaian risiko. Pada tahap ini proses desain pengujian mulai dibuat untuk menghentikan cacat yang mungkin terjadi.

Pada langkah keempat dilakukan pekerjaan untuk pengorganisasian pengujian. Dalam pengujian terdapat istilah pengujian independen, biasanya dilakukan perusahaan atau organisasi yang sistem perangkat lunaknya memenuhi persyaratan yang dimaksudkan. Semakin jauh penguji dari penulisan dokumen, maka semakin besar tingkat independensinya. Semakin besar tingkat independensinya, semakin besar pula kemungkinan kesalahan dalam pengujian yang timbul. Begitu pula tingkat independensi juga tergantung pada ukuran organisasi. Pada organisasi yang lebih kecil di mana setiap orang berkontribusi dalam segala aktivitas, akan lebih sulit untuk membedakan peran penguji dengan peran lain.

Langkah selanjutnya, pemantauan pengujian akan memeriksa terhadap apa yang sebenarnya terjadi setelah rencana pengujian, kegiatan dan rentang waktu dibuat. Tujuan pemantauan pengujian agar memberikan umpan balik terkait kegiatan pengujian. Kontrol pengujian dilakukan menggunakan referensi laporan data pengujian, untuk memastikan *exit criteria* terpenuhi dan memutuskan tindakan pengendalian untuk aktivitas pengujian berikutnya. Hal seperti itu perlu dilakukan ketika muncul ketidaksesuaian pada aktivitas pengujian dengan rencana.

Langkah keenam yaitu manajemen insiden. Insiden di dalam pengujian diartikan menjadi hal di mana hasil pengujian yang ada, berbeda dengan hasil yang diharapkan. Suatu insiden dapat berupa cacat atau permasalahan yang terjadi pada pengujian. Laporan insiden memiliki tujuan untuk memberikan umpan balik kepada pengembang tentang masalah yang ada saat proses pengujian agar dilakukan identifikasi dan perbaikan seperlunya. Penilaian kualitas sistem yang sedang diuji dan kemajuan pengujian dilakukan berdasarkan jumlah insiden, kategori prioritas insiden, dan jumlah insiden yang telah diperbaiki.

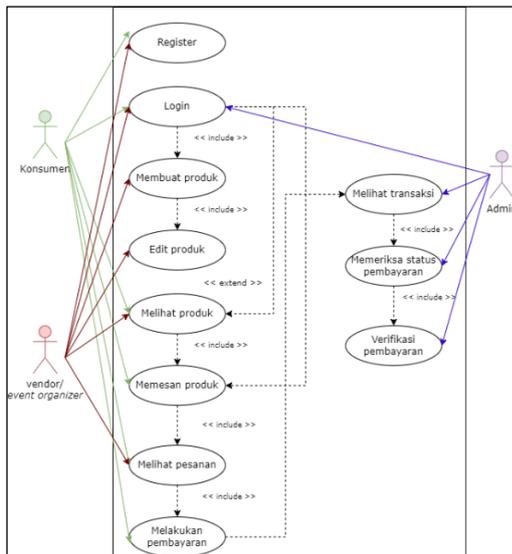
Terakhir, dilakukan tahap pelaporan pengujian. Laporan pengujian adalah proses laporan metrik pengujian secara ringkas yang memberikan informasi kepada pembaca mengenai pengujian yang telah dilakukan selesai. Informasi yang dilaporkan berisi hal yang terjadi selama periode tertentu, tingkatan pengujian, dan seluruh upaya dalam pengujian. Selain itu laporan juga berisi tentang analisis informasi dan metrik untuk mendukung rencana yang akan dilakukan pengujian dimasa mendatang.

#### IV. HASIL DAN PEMBAHASAN

##### A. Analisis Risiko dan Pengujian

Aplikasi yang akan dilakukan pengujian adalah aplikasi Ivent berbasis web [14] [15] [16]. Ivent merupakan *e-commerce* yang menghubungkan antara pemilik usaha dengan pencari *event organizer* atau vendor. Pengguna utama yang akan menggunakan aplikasi ini yaitu pemilik vendor atau *event organizer* dan pencari vendor. Pemilik vendor dapat mengunggah portofolio pada aplikasi Ivent, sehingga pencari vendor dapat melihat penawaran produk. Aktor yang berpartisipasi dalam aplikasi Ivent dapat dilihat pada tampilan *use case diagram* Gambar 2 dengan modifikasi [15].

Ketika dijalankan setiap *use case* diidentifikasi memiliki risiko yang dapat mempengaruhi fungsionalitasnya. Contoh risiko pada tahap *register* yaitu apabila pengguna tidak mendaftar maka tidak dapat menggunakan aplikasi Ivent. Sedangkan pada tahap memesan produk, pengguna perlu mencantumkan tanggal penyewaan agar pemesanan dapat masuk ke keranjang. Begitu pula dengan risiko yang dimiliki *use case* yang lainnya. Oleh karena itu berdasarkan risiko yang ada, pengujian memiliki gambaran teknis pengujian yang akan dilakukan.



Gambar 2. Use Case Diagram dari Ivent

##### B. Perencanaan dan Estimasi Pengujian

Berdasarkan perbedaan kegunaan dari setiap *use case*, dibuatlah batasan pengujian sampai tahap fitur yang telah diimplementasi. Hal ini dikarenakan keterbatasan pada sumber yang digunakan, yaitu terkait aplikasi yang belum selesai dalam proses pembuatannya. Fitur yang dapat dilakukan pengujian seperti *registrasi*, *login*, tambah produk, edit produk, melihat produk, dan lain lain.

Skenario pengujian yang direncanakan akan menggunakan metode *blackbox*, karena dianggap sesuai untuk melakukan pengujian fungsionalitas. Setiap fitur berdasarkan *use case* dengan kode TC akan diuji untuk melihat kesesuaian fungsinya, contoh dapat dilihat pada Tabel I. Skenario tersebut nantinya akan menghasilkan jawaban terkait apakah keluarannya sesuai dengan masukan yang dilakukan pengujian. Sehingga dengan

skenario yang dibuat diestimasikan bahwa aplikasi dapat diuji sesuai dengan kebutuhan yang ditentukan.

Tabel I. Contoh skenario pengujian aplikasi Ivent

Id	Use Case	Skenario Pengujian	Tahapan Pengujian	Keluaran yang Diharapkan
TC 1.1	Register	Mengisikan data diri sesuai form pendaftaran.	1. Membuka halaman berada Ivent 2. Memilih opsi Register atau Daftar 3. Masuk ke dalam halaman Register 4. Mengisi form dengan benar 5. Klik "Register"	Pengguna dapat mendaftarkan data dirinya dan data tersimpan.
TC 1.2		Mengosongkan bagian nama lengkap pada form pendaftaran.	1. Membuka halaman berada Ivent 2. Memilih opsi Register atau Daftar 3. Masuk ke dalam halaman Register 4. Mengisi form dan mengosongkan bagian nama lengkap 5. Klik "Register"	Pengguna tetap berada pada halaman register dan muncul peringatan kesalahan.
TC 1.3		Mengosongkan bagian alamat email pada form pendaftaran.	1. Membuka halaman berada Ivent 2. Memilih opsi Register atau	Pengguna tetap berada pada halaman register dan muncul peringatan kesalahan.

Estimasi waktu yang dibutuhkan pengujian aplikasi Ivent adalah 3 bulan. Pada 1 bulan pertama dilakukan analisis dan perencanaan terkait pengujian, lalu bulan berikutnya dilakukan tahap eksekusi. Berikutnya pembuatan laporan akhir dari pengujian yang dilakukan selama 1 bulan terakhir. Sedangkan estimasi biaya yang diperlukan adalah 0 atau gratis. Hal tersebut dikarenakan penggunaan Qase yang gratis dan penulis memanfaatkan peralatan milik sendiri selama proses pengujian.

##### C. Pendekatan Pengujian (Test Approaches/Strategies)

Pengujian dilakukan dengan menggunakan metode Blackbox. Metode Blackbox diartikan menjadi sebuah metode yang digunakan tanpa harus memperhatikan source code untuk menguji sebuah perangkat lunak. Blackbox hanya menguji fitur-fitur fungsional pada perangkat lunak sehingga sering disebut dengan pengujian fungsionalitas. [17]

Pengujian yang menerapkan metode 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 [18].

Pendekatan metrik memerlukan pengelompokan dalam prosesnya, biasanya digunakan sebagai alat untuk mengukur fungsionalitas, *source code*, arsitektur, dan kinerja dari perangkat lunak [19]. Pendekatan metrik adalah tindakan mengevaluasi dan menentukan tingkat produktivitas perangkat lunak [20].

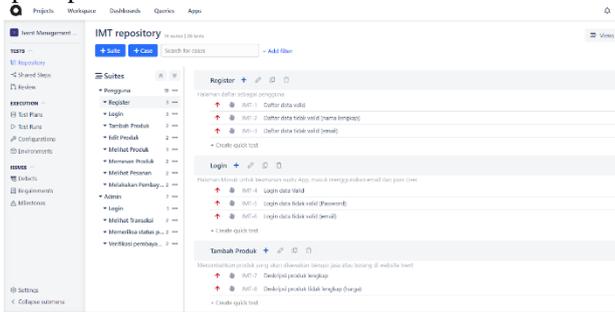
##### D. Organisasi Pengujian

Qase memungkinkan untuk mengorganisasikan kasus pengujian dengan memanfaatkan berbagai fitur seperti *test plans and runs*, *test case management*, *team management*, dan lain lain.

Dalam langkah persiapan penggunaan Qase, dimulai dari mengakses Qase melalui <https://app.qase.io/>. Lalu pengguna mendaftarkan diri dengan mengisi *form* yang telah disediakan pada halaman *register*. Akun yang telah terdaftar nantinya akan digunakan untuk *login* Qase tanpa perlu mendaftar berulang kali. Setelah *login* dengan akun yang dibuat, pengguna akan masuk ke halaman *project*. Di

mana pada halaman ini perlu membuat *project* baru untuk melakukan pengujian dengan klik *create new project* lalu mengisi *form* detail *project*.

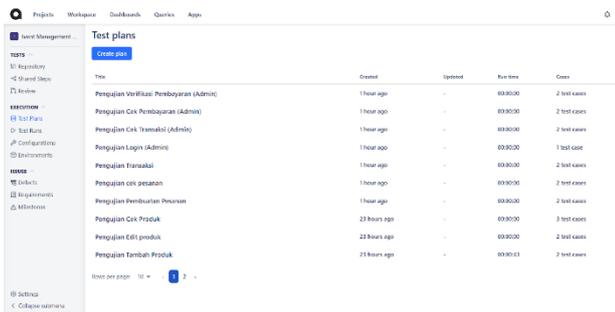
Langkah berikutnya adalah membuat rangkaian *test suite* sesuai urutan pembuatan pada halaman *repository* seperti pada Gambar 3.



Gambar 3. Halaman Repository

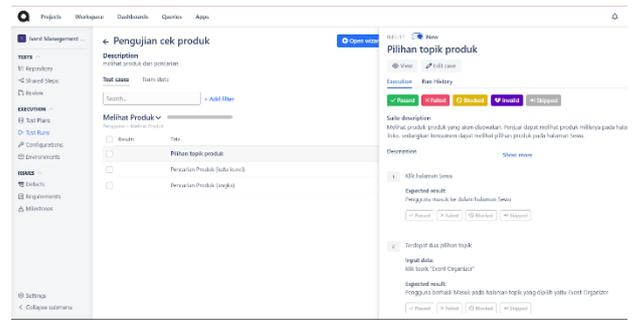
Membuat *test case* di dalam *test suite* dilakukan dengan mengisi *form* detail *test case* sesuai kebutuhan. Prioritas *test case* saat pengujian dapat diatur pada tingkat *high*, *medium*, dan *low*. Selain itu, perilaku *test case* juga dapat dikategorikan *test case positive*, *negative*, dan *destructive*.

Berikutnya adalah pembuatan *test plan* berdasarkan *test case* yang telah dibuat di *repository*. *Test plan* dibuat dengan mengisi *form* detail rencana dan menambahkan *test case* yang akan diuji. Nantinya *test plan* akan muncul pada halaman *Test Plan* sesuai dengan urutan pembuatan seperti Gambar 4.



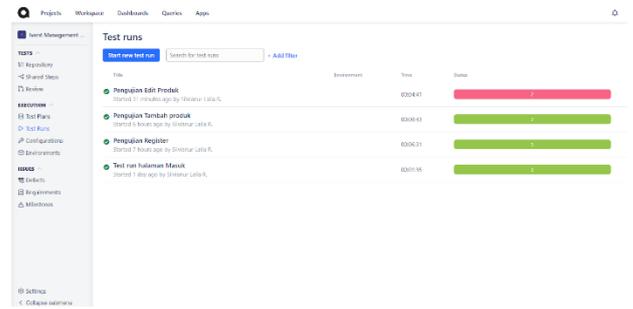
Gambar 4. Halaman Test Plan

*Test run* dapat dijalankan dengan memilih salah satu *test plan* yang akan diuji dan mengisi *form* detail lalu klik *start run* untuk berpindah ke halaman eksekusi. Pengujian dimulai dengan klik *open wizard* agar muncul halaman eksekusi *test case* seperti pada Gambar 5. 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 sesuai dengan harapan. *Fail* apabila *step* tidak dapat dilakukan karena tidak sesuai dengan yang direncanakan.



Gambar 5. Halaman Eksekusi Test Case

Gambar 6 menunjukkan halaman *test run* setelah dilakukan eksekusi, akan ditampilkan jumlah waktu eksekusi tiap *test run* beserta status pengujian.



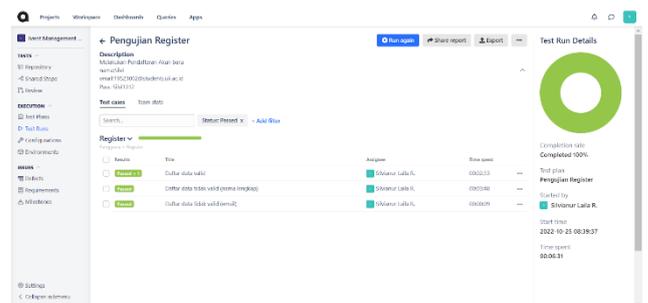
Gambar 6. Halaman Test Run

Qase memiliki fitur untuk mengundang anggota tim untuk melakukan eksekusi *test case* bersama. Fitur manajemen tim ini berada pada halaman *Workspace* dengan klik *invite* dan mengisi detail anggota baru. Undangan akan dikirimkan melalui email yang telah didaftarkan dengan peran yang berbeda.

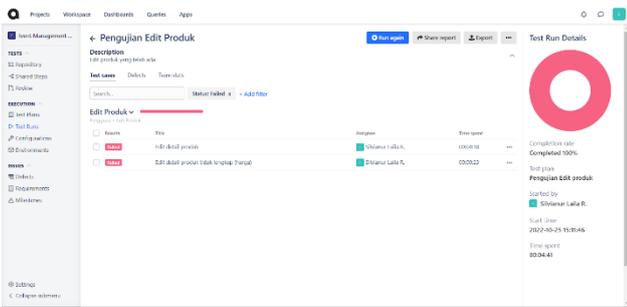
### E. Pemantauan dan Kontrol Pengujian

Memantau pengujian dilakukan agar pengujian yang berjalan sesuai dengan rencana dibuat. Pengujian aplikasi *Event* menggunakan *test case* pada Tabel 1 secara keseluruhan. Pengujian dilakukan secara manual dengan alat bantu Qase untuk manajemen pengujian. Pengujian dilakukan untuk mengetahui apakah fitur pada aplikasi *Event* sesuai dengan *use case* yang direncanakan. Pengendalian hasil dari pengujian menjadi umpan balik yang digunakan untuk memperbaiki kegagalan fitur.

Fitur dikatakan memiliki status *pass*, apabila hasil sebenarnya sama dengan hasil yang diharapkan. Apabila fitur berstatus *fail*, maka hasil yang diharapkan tidak sesuai dengan hasil sebenarnya karena terdapat *error* pada fitur.



Gambar 7. Hasil pengujian berstatus *pass*



Gambar 8. Hasil Pengujian Berstatus *fail*

Pengujian pada Qase tidak jauh berbeda terkait penentuan status *pass* dan *fail* pada fitur. Hanya saja pada Qase status terlihat melalui warna dari hasil pengujian, *pass* berwarna hijau seperti Gambar 7 dan *fail* berwarna merah seperti Gambar 8.

#### F. Manajemen Insiden

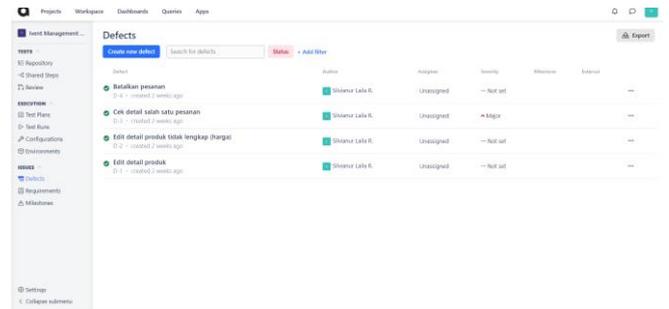
Pengujian pertama yang dilakukan pada 11 fitur aplikasi Ivent dihasilkan 9 fitur berstatus *pass* dan 2 fitur berstatus *fail*. Fitur yang berstatus *fail* yaitu fitur edit produk dan fitur melihat pesanan, dikarenakan terjadi kesalahan saat eksekusi. Dari 28 test case terdapat 4 test case *fail* dan 24 test case *pass*. Dalam manajemen insiden, dilakukan tindak lanjut dan pencatatan kendala pengujian.

Tindak lanjut yang dilakukan berdasarkan hasil pengujian pertama yaitu perbaikan test case berstatus *fail*. Perbaikan dilakukan sesuai dengan prioritas test case dengan kategori *high*, *medium*, dan *low*. Prioritas kategori *high* disebabkan *requirement* yang tidak dapat tampil. Prioritas kategori *medium* berdasarkan *alert* yang tidak muncul atau sejenisnya. Sedangkan prioritas kategori *low* dikarenakan kesalahan kecil yang dapat ditunda perbaikannya.

Daftar prioritas test case yang berstatus *fail* pada pengujian aplikasi Ivent disajikan pada Tabel III. *Defect* yang ada pada Qase diselesaikan dengan status *resolved* apabila fitur telah diperbaiki, dapat dilihat pada Gambar 9.

Tabel III. 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>



Gambar 9. Hasil Penyelesaian *Defect* di Qase

Perbaikan pada fitur edit produk dilakukan dengan mengganti *code* dan menjalankan ulang aplikasi. Setelah *code* yang mengalami kesalahan diganti, fitur edit produk dapat digunakan dengan baik.

Perbaikan pada fitur melihat pesanan dilakukan dengan menambah beberapa akun dan memperbanyak produk berbeda yang ada pada database. Hal ini dilakukan karena *value* pada *database* tidak terdeteksi apabila data masih sedikit. Setelah diperbaiki, fitur melihat pesanan dapat digunakan baik melalui *dropdown* profil atau menu.

Selanjutnya, ditemui kendala dalam penggunaan Qase seperti tutorial penggunaan Qase yang masih sedikit dilakukan oleh orang lain. Selain itu pencarian artikel maupun jurnal yang berkaitan dengan Qase *tools* juga cukup sulit. Karena Qase masih menjadi alat manajemen kasus uji yang belum populer untuk digunakan. Beberapa fitur pada Qase tidak dapat digunakan karena perlu biaya berlangganan seperti *requirement*, *reports*, *review*, dan *metrics*.

Manfaat menurut penulis dalam menggunakan aplikasi Qase selain dapat digunakan secara gratis, yaitu Qase memiliki fitur yang membantu dalam mengatur dan mengelola pengujian agar lebih terstruktur. Qase juga memberikan fitur mengundang anggota baru untuk melakukan pengujian bersama. Lalu Qase memberikan kemudahan saat melakukan *tracking* apabila terjadi *re test* pada test case yang sama.

Manfaat tersebut telah dikonfirmasi kepada praktisi yang menggunakan Qase dalam pengujian sebagai *Quality Assurance*. Bahwa ternyata manfaat Qase yang dirasakan penulis juga sama dengan yang dirasakan praktisi. Namun karena praktisi menggunakan Qase versi berbayar, seluruh fitur dapat digunakan untuk proses manajemen pengujian.

Oleh karena itu, walaupun terdapat beberapa fitur pada Qase yang belum dapat digunakan penulis akibat berbayar. Qase tetap mampu membantu dalam mengatur dan mengelola pengujian agar lebih terstruktur, hal itu berdasarkan tahap pengujian yang telah dilakukan. *Defect* dan riwayat dari pengujian yang telah dilakukan dapat tersimpan dan menjadi kemudahan acuan penguji berikutnya.

#### G. 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.

Setelah dilakukan pengujian keseluruhan fitur terutama pada fitur yang mengalami kegagalan, maka hasil pengujian dirangkum pada Gambar 10. 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				
Tanggal Mulai Pengujian: 10 Oktober 2022				
Tanggal Selesai Pengujian: 12 November 2022				
Tabel 3.16 Rangkuman Hasil Keseluruhan Pengujian				
No.	Test Suite	Jumlah Test Case	Jumlah <i>pass</i>	Jumlah <i>fail</i>
<b>Pengguna</b>				
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	2	0
5	Melihat Produk	3	3	0
6	Membuat Pesanan	4	4	0
7	Melihat Pesanan	2	2	0
8	Melakukan Pembayaran	2	2	0
<b>Admin</b>				
9	Melihat Transaksi	2	2	0
10	Transaksi pesanan konsumen dikirim	2	2	0
11	Verifikasi Pembayaran	2	2	0
<b>Total</b>		<b>28</b>	<b>28</b>	<b>0</b>

Gambar 10. Hasil Laporan Pengujian

Pengujian pada aplikasi Ivent telah selesai dilakukan dengan bantuan Qase. Qase 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.

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*.

## V. KESIMPULAN DAN SARAN

Penelitian telah berhasil dilakukan dengan melaksanakan proses manajemen pada pengujian aplikasi Ivent menggunakan aplikasi Qase. Manajemen pengujian dilakukan berdasarkan 7 tahapan yaitu analisis risiko dan pengujian, perencanaan dan estimasi pengujian, pendekatan pengujian, organisasi pengujian, pemantauan dan kontrol pengujian, manajemen insiden, serta pembuatan laporan pengujian. Dari 11 *usecase* dibuatlah 28 skenario *test case* untuk menguji fitur fungsionalitas aplikasi dengan metode *blackbox*. Pengujian dilakukan 2 kali dengan hasil pengujian pertama 24 *test case* berstatus *pass* dan pada

pengujian kedua menguji 4 *test case* yang berstatus *fail* sehingga menghasilkan 28 *test case* berstatus *pass*.

Manfaat dari penggunaan Qase adalah Qase memiliki fitur untuk mengatur dan mengelola pengujian, serta mampu mengundang anggota baru. Selain itu Qase memberikan kemudahan *tracking* saat *re test* pada *test case* yang sama. Walaupun terdapat beberapa fitur pada Qase yang belum dapat digunakan akibat berbayar. Qase tetap mampu membantu dalam mengatur dan mengelola pengujian agar lebih terstruktur.

Saran yang dapat diberikan adalah penelitian di masa mendatang yaitu dalam pembuatan *test case* dapat diperbanyak dengan memanfaatkan studi kasus yang lebih rumit agar penggunaan fitur Qase menjadi lebih maksimal dan terlihat nyata hasilnya. Lalu perlu dikaji lebih banyak referensi terkait pengujian perangkat lunak dan alat bantu serta metode manajemen pengujian yang akan digunakan.

## REFERENSI

- [1] R. C. Ioan Mihnea IACOB, "Testing: First Step Towards Software Quality," *Journal of Applied Quantitative Methods*, vol. 3, pp. 241-253, 2008.
- [2] T. D. H. P. Hariyanto Hariyanto, "Software Quality Assurance pada Perusahaan Pengembang Perangkat Lunak Skala Kecil dan Menengah," *Jurnal Riset Teknologi dan Inovasi Pendidikan (JARTIKA)*, pp. 283-294, 2020.
- [3] C. Kartiko, "Evaluasi Kualitas Aplikasi Web Pemantau Menggunakan Model Pengujian Perangkat Lunak ISO/IEC 9126," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, vol. 8 No. 1, pp. 16-23, 2019.
- [4] E. v. V. I. E. R. B. Dorothy Graham, *Foundations of Software Testing : ISTQB Certification*, Thomson, 2006..
- [5] Rajkumar, "Qase Test Case Management Tool Review Tutorial 2022," 5 October 2022. [Online]. Available: <https://www.softwaretestingmaterial.com/qase-test-case-management-tool/>.
- [6] P. T. Kshirasagar Naik, *Software Testing and Quality Assurance : Theory and Practice*, New Jersey: John Wiley & Sons, Inc, 2008.
- [7] R. V. K. A. Abdulrahman Mahmoud, "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*, pp. 1087-1103, 2019.
- [8] P. M. A. G. T. a. P. W. Brian Hambling, *Software Testing An ISTQB-ISEB Foundation Guide Second Edition*, Swindon, UK: British Informatics Society Limited, 2010.
- [9] A. K. V. Rajeshwari Sissodia, "Analysis of Automation and Manual Testing," *International Journal of Scientific & Engineering Research*, vol. 8, no. 10, pp. 355-357, 2017.
- [10] A. P.-M. Michal Pawlak, "Software Test Management Approach for Agile Environments," *Information Systems in Management Vol. 7 (1)*, p. 47-58, 2018.
- [11] E. Baidhya, "An Optimum Test Management System," *Turku University Of Applied Sciences*, pp. 1-50, 2019.
- [12] R. Arivazhagan, "MST Solutions," 11 June 2022. [Online]. Available: <https://www.mstsolutions.com/technical/qase-is-a-cloud-based-test-case-management-tool/>.
- [13] E. S. Devaldi Akbar Suryadi, "Evaluation of Information Quality Using ISO/IEC 25010:2011 (Case Research: Menu Harianku Application)," *International Journal Of Innovation In Enterprise System*, Vols. 06, No. 02, pp. 119-132, 2022.
- [14] D. N. Dani, "Strategi Ide Bisnis Startup E-commerce Ivent Menggunakan Metode Design Thinking," *Universitas Islam Indonesia*, 2021.
- [15] A. A. Mucjal, "Pengembangan e-commerce vendor dan event organizer berbasis website dengan metode waterfall," *Universitas Islam Indonesia*, 2021.

- [16] M. A. Idris, "Perancangan User Interface & User Experience Pada Aplikasi E-Commerce Ivent Menggunakan Metode User Centere Design (Ucd)," *Universitas Islam Indonesia*, 2021
- [17] F. M. F. A. A. F. R. F. D. A. D. P. R. Bagus Bayu Sasongko, "Pengujian Blackbox Menggunakan Teknik Equivalence," *Jurnal ICTEE*, pp. 10-16, 2021.
- [18] D. S. S. A. e. Fadhila Cahya Ningrum, "Pengujian Black Box pada Aplikasi Sistem Seleksi Sales Terbaik," *Jurnal Informatika Universitas Pamulang*, pp. 125-130, 2019.
- [19] W. A. R. H. S. M. A. Y. Muhammad Andryan Wahyu Saputra, "Survei Teknik-Teknik Pengukuran Kualitas Perangkat Lunak," *Journal of Computer Science and Applied Informatics*, pp. 11-29, 2021.
- [20] D. C. M. W. d. Rizky Parlita, "Studi Literatur Perbandingan Antara Metode LOC, COCOMO, FPA dalam Ranah Software Metric," *Jurnal Pendidikan Informatika dan Sains*, pp. 66-74, 2020