



Pengembangan Sistem *Dashboard* untuk Mengevaluasi *Platform E-Marketplace* dengan Metode *Scrum* pada *Startup GRAVIS*

Zakky Al Attar^{#1}, Tien Fabrianti Kusumasari^{#2}, Rahmat Fauzi^{#3}

[#]Program Studi Sistem Informasi, Universitas Telkom

Jl. Telekomunikasi Terusan Buah Batu 40257 Bandung, Indonesia

¹zakkyalattar@student.telkomuniversity.ac.id

²tienkusumasari@telkomuniversity.ac.id

³rahmatfauzi@telkomuniversity.ac.id

Abstrak— Dalam pengelolaan sebuah bisnis, penjual perlu mengetahui kinerja tokonya dengan mengolah dan menganalisis data transaksi dengan pembeli menjadi sebuah informasi menggunakan *Dashboard*. Penggunaan *dashboard* cukup penting sehingga butuh studi lanjutan terkait evaluasi penggunaannya. Penelitian sebelumnya lebih banyak membahas pembuatan dan pengembangan aplikasi *dashboard* berbasis website dan mendiskusikan operasionalisasi *dashboard* pada lembaga pendidikan. Oleh karena itu penelitian ini berupaya memfokuskan pada aspek perusahaan rintisan. Penelitian ini akan difokuskan pengembangan *dashboard* menggunakan framework *Laravel* dengan metode pengembangan *Scrum*. Penelitian ini menggunakan model konseptual dan sistematika penyelesaian masalah sehingga tercipta sebuah aplikasi yang membantu *e-Marketplace*. Tujuan penelitian ini adalah (1) memahami KPI evaluasi platform *GRAVIS*. (2) memahami konsep arsitektur aplikasi *dashboard* platform *GRAVIS*. (3) mengembangkan *dashboard* untuk menyajikan informasi dalam bentuk visualisasi data. Hasil penelitian ini menunjukkan (1) *Key performance indicator* (KPI) yang digunakan meliputi enam belas poin utama. (2) *Dashboard* akan menampilkan informasi dengan *gauge chart*, *line chart* dan *bar chart*. (3) Hasil dari enam scenario *black-box testing* yang diujikan mendapatkan total sukses sebanyak enam scenario dan nol scenario gagal, maka dari itu hasil dari pengujian *black-box testing* selesai pada iterasi pertama.

Kata kunci— Evaluasi, *E-Marketplace*, *Scrum*, Sistem *Dashboard*, *Startup GRAVIS*

I. PENDAHULUAN

Dalam era digitalisasi saat ini, banyak masyarakat Indonesia yang ingin mengembangkan bisnisnya menjadi bisnis digital karena kemudahan menjangkau konsumen tanpa harus memiliki kantor atau toko di lokasi yang dekat dengan konsumen. Karenanya banyak bermunculan *startup* maupun *UMKM* (Usaha Mikro Kecil Menengah) yang berorientasi pada ekonomi digital dan salah satunya adalah inovasi *e-Marketplace* yang berfokus hanya pada satu bidang untuk menjangkau para konsumen yang mencari

produk atau jasa pada bidang tertentu. *E-Marketplace* merupakan media pemasaran produk secara elektronik untuk mempertemukan berbagai penjual dan pembeli sehingga dapat saling bertransaksi [1-2]. *E-Marketplace* telah menyediakan wadah untuk berjualan online. Hanya dibutuhkan pendaftaran saja untuk memulai berjualan. Semua proses telah ditanggung oleh tim *marketplace*. Penjual hanya butuh untuk meningkatkan pelayanan dan promo. Kemungkinan penjualan barang lebih banyak, karena pada platform *e-Marketplace* telah bergabung banyaknya para penjual dan pembeli [2].

Dalam pengelolaan sebuah bisnis, penjual perlu mengetahui kinerja toko agar dapat mengambil strategi pengembangan produknya. Kinerja toko dapat diketahui dengan mengolah dan menganalisis data transaksi dengan pembeli menjadi sebuah informasi dengan menggunakan *Dashboard*. *Dashboard* merupakan aplikasi sistem informasi untuk menampilkan informasi terkait indikator utama dari aktivitas organisasi atau perusahaan secara sekilas dalam layar tunggal [3]. *Dashboard* digunakan oleh para manajer untuk memantau penjualan, keuntungan, dan analisis bisnis yang lainnya [4].

Dashboard terdiri dari banyak indikator visual dimana setiap indikatornya menampilkan satu set KPI (*Key Performance Indicator*). KPI merupakan ukuran keuangan dan non-keuangan yang digunakan organisasi untuk mengungkapkan seberapa sukses mereka dalam mencapai tujuan jangka panjang [5]. KPI merupakan indikator utama yang digunakan untuk mengevaluasi dan menganalisis data pada suatu organisasi dan dengan adanya KPI perusahaan dapat menghasilkan *reward* dan *punishment* [6].

Telah banyak penelitian terdahulu yang mencoba untuk mengkaji tentang penggunaan *dashboard* untuk perusahaan. Ruiperez-Valiente et al., [7] menemukan bahwa guru dapat melakukan monitoring langsung murid mereka selama kelas sehingga penggunaan *dashboard* cukup penting untuk diimplementasikan. Wibowo et al., [8] menemukan bahwa penggunaan *decision support system dashboard*

dapat memprediksi kelulusan mahasiswa menggunakan metode Pureshare. Saputra et al., [9] menemukan bahwa *dashboard* dapat digunakan untuk monitoring performa mahasiswa dengan menyediakan tampilan *dashboard* berbasis *web* yang dapat diakses oleh dosen sebagai indikator kinerja monitoring performa mahasiswa berbasis *website*. Iqbal et al., [10] menemukan tampilan *dashboard website* Egarage.id dengan menggunakan metode agile iterative dan incremental dapat digunakan untuk melakukan monitoring suatu organisasi secara efektif.

Dari studi yang dilakukan oleh Ruiperez-Valiente et al., Wibowo et al., Saputra et al., dan Iqbal et al., [7–10] dapat disimpulkan bahwa penggunaan *dashboard* cukup penting untuk monitor kinerja sebuah organisasi atau perusahaan sehingga butuh studi lanjutan terkait evaluasi penggunaan *dashboard* pada suatu organisasi. Penelitian oleh Ruiperez-Valiente et al., Wibowo et al., Saputra et al., dan Iqbal et al., [7–10] telah membahas mengenai pembuatan dan pengembangan aplikasi *dashboard* berbasis *website* dan telah banyak mendiskusikan operasionalisasi *dashboard* dengan tujuan kinerja lembaga pendidikan. Berdasarkan penelitian yang pernah dilakukan, maka penelitian ini berupaya memfokuskan pada aspek selain pendidikan, yaitu pada perusahaan rintisan sehingga fokus pada komponen yang akan ditampilkan pada *dashboard* juga akan berbeda. Karena adanya fenomena teori dan empiris ini maka penelitian ini menjadi penting untuk dilakukan. Penelitian ini akan difokuskan pengembangan *dashboard* menggunakan framework Laravel dengan metode pengembangan Scrum.

Adapun tujuan penelitian untuk (1) memahami KPI yang digunakan untuk monitoring dan evaluasi platform GRAVIS. (2) memahami konsep arsitektur aplikasi yang digunakan dalam membangun *dashboard* platform GRAVIS. (3) mengembangkan *dashboard* untuk menyajikan informasi dalam bentuk visualisasi data.

Pada penelitian ini, *dashboard* digunakan sebagai sistem monitoring dan evaluasi platform e-Marketplace GRAVIS. Startup Digital Graphic Visual (GRAVIS) merupakan startup e-Marketplace bidang jasa desain digital dengan tujuan menghubungkan antara freelancer desain dengan para UMKM atau korporasi yang membutuhkan jasa desain.

II. TINJAUAN PUSTAKA

A. Dashboard

Dashboard merupakan sekumpulan data yang secara visual menampilkan indikator kinerja bisnis. *Dashboard* digunakan oleh para manajer untuk memantau penjualan, keuntungan, dan analisis bisnis yang lainnya [4]. Kebanyakan *dashboard* dibangun berbasis *website* dan dapat diakses melalui *web browser* yang terhubung dengan *database* yang memungkinkan data laporan dan tabel diperbarui setiap saat. Penggunaan *dashboard* bertujuan untuk mempengaruhi pilihan dalam desain visual dan kemampuan fungsionalnya. Namun Tujuan *dashboard* telah diperluas secara substansial dari *dashboard* "operasional" sehari-hari untuk pengambilan keputusan di

tingkat yang lebih tinggi, dan bahkan mungkin tidak mendukung pengambilan keputusan sama sekali [11]. Terdapat 6 kategori bentuk penyajian dalam *dashboard* antara lain: grafik, ikon, teks, gambar, *drawing object*, dan *organizer* [12].

B. Balance Scorecard

Balance scorecard (sistem akuntansi pertanggung jawaban berbasis strategis) adalah tujuan dan ukuran sistem akuntansi pertanggung jawaban untuk empat perspektif yang berbeda: perspektif keuangan, perspektif pelanggan, perspektif proses, dan perspektif pembelajaran dan pertumbuhan (infrastruktur) [15]. Premis dasarnya adalah untuk menetapkan seperangkat indikator yang dapat digunakan untuk memantau kemajuan kinerja dan kemudian membandingkan tujuan yang ditetapkan dengan hasil. *Balanced scorecard* merupakan suatu sistem manajemen strategis atau lebih tepat dinamakan suatu "Strategic based responsibility accounting system" yang menjabarkan misi dan strategi suatu organisasi kedalam tujuan operasional dan tolak ukur kinerja untuk empat perspektif yang berbeda, yaitu perspektif keuangan (*financial perspective*), perspektif pelanggan (*customer perspective*), perspektif proses usaha internal (*internal business process perspective*), dan perspektif pembelajaran dan pertumbuhan (*learning and growth (infrastructure) perspective*) [16].

C. Key Performance Indicator

Key Performance Indicators (KPI) ialah suatu pernyataan tentang suatu posisi atau jabatan yang terukur. Berasal dari visi perusahaan, cara pencapaian visi melalui misi. Kemudian dijabarkan oleh setiap departemen. Dari misi kemudian dapat menghadirkan sistem kerja yang kemudian dilanjutkan dengan sasaran (*outcome*) dari tugas, dengan memiliki nilai ambang batas (*threshold*) untuk membedakan antara nilai target dengan nilai aktual. Dengan adanya KPI perusahaan dapat menghasilkan *reward* dan *punishment* [6]. KPI dibedakan menjadi *lagging indicator* dan *leading indicator*. *leading indicator* (indikator periode mendatang) adalah suatu rangkaian data statistik periode lalu yang menunjukkan kecenderungan yang mencerminkan perubahan - perubahan pada waktu mendatang dalam beberapa sektor ekonomi terkait atau sebagai sinyal kejadian di masa depan. Singkatnya, *leading indicator* merupakan beberapa variabel ekonomi yang bergerak mendahului pergerakan variabel utama ekonomi. *Lagging indicator* atau yang disebut juga sebagai indikator periode lalu adalah suatu rangkaian data statistik pada periode lalu dan telah menunjukkan kecenderungan yang mencerminkan perubahan-perubahan pada waktu lalu dalam beberapa sektor ekonomi yang saling berkaitan [17].

D. Perancangan Aplikasi

Proses pengembangan perangkat lunak dirancang untuk menghasilkan produk perangkat lunak dengan cepat dan kualitas yang lebih baik. Dalam perancangan sistem perangkat lunak, terdapat banyak metode yang dapat

digunakan salah satunya adalah metode *Agile*. Metode *Agile* berfokus pada kepuasan pengguna, dan mencapai tujuan ini dengan cara yang efisien. *Agile* menekankan kerja sama antar individu untuk menciptakan perubahan yang cepat dan murah, dan berfokus pada fungsionalitas produk dibandingkan dengan pedoman yang ada [18]. Metodologi pengembangan *Agile* saat ini sudah diadopsi secara luas oleh perusahaan software. Namun, mengelola *Agile projects* merupakan sebuah tantangan, terutama dalam kasus Usaha Kecil dan Menengah (UKM). Tantangan bagi Pemilik Produk dan *Scrum Master* setidaknya ada dua: untuk memastikan kualitas produk perangkat lunak dan untuk memfasilitasi efektivitas tim dan proses [19].

E. Metode Pengembangan Sistem

1. Metode Pengembangan *Prototype*

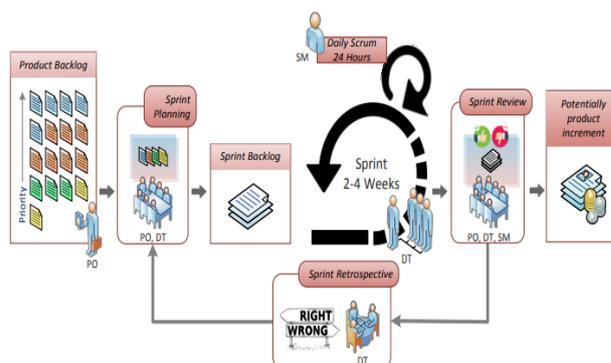
Prototyping merupakan metode pengembangan perangkat lunak dengan membuat rancangan model yang merupakan versi awal dari sebuah perangkat lunak dengan tujuan pengujian produk tersebut. Metode *prototyping* adalah cara yang efektif dan efisien untuk melakukan validasi kebutuhan perangkat lunak untuk menghindari kesalahan pada tahap awal pengembangan perangkat lunak. Selain implementasi fungsionalitas sistem utama, *prototype* juga memiliki *user interface* (UI) yang memungkinkan klien memvalidasi kebutuhan sistem secara visual, memudahkan dalam menemukan kesalahan dan memperbaikinya [20]. Model *prototype* merupakan salah satu model pengembangan perangkat lunak dimana pengembang program dan objek penelitian dapat saling berkomunikasi dan memberikan informasi yang terdiri dari mendengarkan pelanggan atau analisa kebutuhan, membuat rancangan (*mockup*) dan pengujian rancangan [21].

2. Metode Pengembangan *Scrum*

Scrum merupakan Metodologi *agile* yang berfokus pada manajemen proyek dan memungkinkan pengembangan iteratif. *Scrum* adalah siklus pengembangan *incremental*, terdiri dari *short-time deliveries* secara berkala yang menggunakan iterasi untuk mendapatkan produk perangkat lunak akhir (*final*) [22]. *Scrum* dimulai dengan sebuah tim kecil yang terdiri dari sekitar tujuh pengembang perangkat lunak dan bergerak sendiri atau *self-organizing*. Manajemen dalam tim merupakan *Scrum Master* yang bertanggung jawab untuk menghilangkan hambatan bagi tim dan memastikan bahwa praktik *Scrum* berjalan dan diikuti. Setiap proyek memiliki suatu *Product Backlog*, yang merupakan daftar prioritas dari semua kebutuhan perangkat lunak yang harus dicapai. *Product Backlog* selalu berubah sesuai dengan perubahan kebutuhan dan perubahan pemahaman tentang permasalahan. Hanya *Product Owner* yang memiliki kewenangan untuk memprioritaskan *Product Backlog*.

Scrum memecahkan proyek menjadi beberapa *increments* yang disebut *Sprint*. *Sprint* merupakan iterasi

tunggal, biasanya 30 hari, dimana tim *Scrum* menambahkan fungsionalitas baru kepada produk dan menentukan bagian mana dari *Product Backlog* yang harus dikembangkan. Bagian dari *Product Backlog* ini diterjemahkan ke dalam tujuan yang disebut *Sprint Goal*. Kemudian, Tim *Scrum* bertemu secara internal dan menggunakan *Sprint Goal* untuk mendapatkan daftar kebutuhan yang diperlukan untuk mencapai tujuan tersebut. Kebutuhan ini ditempatkan ke dalam *Sprint Backlog*. Keberhasilan dari suatu *Sprint* ditentukan dengan capaian pada *Sprint Goal* [23].



Gambar 1. Kerangka kerja *scrum* [23]

F. Framework Laravel

Laravel adalah kerangka kerja yang menggunakan pola desain MVC (*Model View Controller*). *Framework* ini menempatkan *level view* dan *controller* di direktori "*Resources*" dan "*Routes*". Pada *level model*, *framework* Laravel menyediakan direktori "*database*" yang digunakan untuk manipulasi data. Laravel merupakan salah satu *framework* PHP yang paling banyak digunakan. Kerangka kerja PHP menyediakan kode yang telah dibuat sebelumnya. Kita hanya perlu mengatur modul ini sesuai dengan kebutuhan kita [24]. Dalam pemanfaatannya Laravel memiliki beberapa kelemahan, salah satunya adalah ukuran dokumen yang sangat besar. Terlebih lagi, ini memerlukan asosiasi web dengan memperkenalkan dan mengunduh pustaka Laravel dan varian dasar PHP minimal versi 5.4 untuk menjalankannya [25].

G. MySQL

MySQL adalah *Relational Database Management System* (RDBMS) yang digunakan sebagai SaaS (*Software as a Service*) melalui *cloud*. MySQL merupakan sistem manajemen *database* yang paling populer untuk penyedia layanan *hosting*. Sistem manajemen basis data MySQL memiliki kelebihan dan keterbatasan dalam hal keamanan, keandalan, kinerja, dan kemudahan penggunaan [26]. MySQL memiliki beberapa keistimewaan, antara lain: dapat berjalan stabil pada kerangka kerja yang berbeda, bersifat *open source*, dapat digunakan oleh beberapa *user* secara bersamaan, memiliki kecepatan yang luar biasa dalam menangani *query* langsung, memiliki jenis segmen yang sangat kompleks, memiliki beberapa lapisan keamanan, mampu menangani basis data dalam skala besar, dapat mengidentifikasi pesan kesalahan pada klien dengan

menggunakan lebih dari dua puluh bahasa, memiliki *interface* (antar muka) ke berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*), dilengkapi dengan berbagai instrumen (*tool*), dan memiliki struktur tabel yang lebih fleksibel dalam menangani *ALTER TABLE* [26].

H. Black-Box Testing

Black-box testing (BBT) adalah pendekatan pengujian fungsionalitas perangkat yang sedang diuji tanpa mengetahui struktur internal perangkat untuk mengetahui

hasil yang diharapkan dari input yang diberikan. *Black-box testing* melibatkan pengujian perangkat lunak untuk fungsionalitas yang digunakan untuk mengetahui kesalahan dalam struktur data, fungsi yang salah, kesalahan *interface*, dll. *Black-box testing* mengabaikan mekanisme internal suatu sistem untuk mengidentifikasi *bug* yang terjadi sesuai dengan malfungsi *software* seperti yang terungkap dalam *output* yang salah. Ini digunakan untuk menemukan fungsi yang salah yang menyebabkan *output* yang tidak sesuai saat *software* dijalankan [27].

I. Penelitian Terdahulu

TABEL I
PENELITIAN SEBELUMNYA

No.	Jenis Penelitian	Judul Jurnal	Peneliti	Keterangan
1.	Jurnal Internasional	Analysis and Design of Decision Support System Dashboard for Predicting Student Graduation Time (Tahun 2018)	Satrio Wibowo, Rachmadita Andreswari, Muhammad Azani Hasibuan	Pada penelitian ini penulis membahas tentang analisis dan perancangan <i>decision support system dashboard</i> untuk memprediksi kelulusan mahasiswa menggunakan metode Pureshare.
2.	Jurnal Nasional	Perancangan Dashboard untuk Monitoring Performa Mahasiswa D3 Sistem Informasi Fakultas Ilmu Terapan Berbasis Web (Tahun 2021)	Firman Aldo Saputra, Inne Gartina Husein, Mutia Qana'a	Penelitian ini menghasilkan aplikasi perancangan <i>dashboard</i> untuk monitoring performa mahasiswa menyediakan tampilan <i>dashboard</i> berbasis <i>web</i> yang dapat diakses oleh Kaprodi dan dosen prodi D3 Sistem Informasi sebagai indikator kinerja monitoring performa mahasiswa berbasis <i>website</i> .
3.	Jurnal Nasional	Perancangan Dashboard Aplikasi Rental Kendaraan pada Start-Up Egarage.id dengan Menggunakan Metode Agile Iterative dan Incremental (Tahun 2019)	Teuku Muhammad Raza Iqbal, Soni Fajar Gumilang, Putra Fajar Alam	Penelitian menghasilkan tampilan <i>dashboard website</i> Egarage.id dengan menggunakan metode <i>agile iterative</i> dan <i>incremental</i> .

Berdasarkan Tabel 1. diketahui bahwa pada penelitian sebelumnya telah membahas mengenai pembuatan dan pengembangan aplikasi dashboard berbasis website. Pada penelitian sebelumnya membahas tentang pengaplikasian dashboard dengan tujuan yang berbeda sehingga fokus

pada komponen yang akan ditampilkan pada dashboard juga akan berbeda. Maka dari itu pada penelitian ini akan difokuskan pengembangan dashboard menggunakan framework Laravel dengan metode pengembangan Scru

J. Pemilihan Metode

TABEL III
PERBANDINGAN METODE

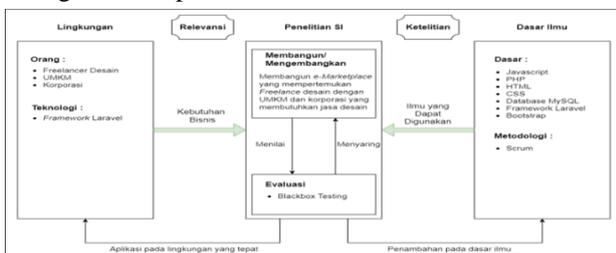
Karakteristik metode	Metode Waterfall	Metode Scrum	Penelitian ini
Fleksibilitas metode	Rigid	Fleksibel	Membutuhkan metode yang fleksibel
Pendefinisian <i>requirement</i>	Harus sudah jelas	Fleksibel, dapat berubah - ubah	Membutuhkan pendefinisian <i>requirement</i> yang dapat berubah - ubah
Perubahan <i>requirement</i> pada saat pengembangan	Sulit untuk diubah	Mudah dalam mengubah <i>requirement</i>	Membutuhkan metode yang mudah dalam mengubah <i>requirement</i> dalam pengembangan
Waktu yang dibutuhkan dalam mengimplementasikan metode saat pengembangan	Lambat	Cepat	Membutuhkan metode pengembangan yang cepat
Tingkat kemudahan dalam mengimplementasikan metode pengembangan	Mudah	Sulit	Membutuhkan metode yang mudah untuk diimplementasikan
Pemilihan Metode	Metode yang dipilih pada penelitian ini adalah metode <i>Framework Scrum</i>		

Berdasarkan tabel perbandingan metode diatas, metode *Scrum* cocok untuk diimplementasikan pada pengembangan *website* GRAVIS. Metode *Scrum* memiliki keunggulan waktu pengembangan yang relatif cepat dan fleksibel sehingga *requirement* pada pengembangan *website* dapat diubah seiring dengan berjalannya pengembangan. Hal tersebut dapat memudahkan pengembangan *website* GRAVIS karena dapat meminimalisir risiko dan kesalahan pada saat pengembangan.

III. METODE

A. Model Konseptual

Model konseptual merupakan suatu diagram yang menunjukkan hubungan secara logis antara variabel yang sudah diidentifikasi. Dalam penelitian ini akan digunakan model konseptual *design science research* [11-13] sebagai kerangka dalam pemecahan masalah.



Gambar 2. Model konseptual

Model Konseptual dimulai dengan mengidentifikasi lingkungan penelitian. Terdapat komponen orang dimana freelancer menyediakan jasa desain, sedangkan UMKM dan korporasi membutuhkan jasa desain. Kemudian pada komponen teknologi digunakan framework Laravel yang dibangun untuk menghubungkan freelancer dengan UMKM melalui platform e-Marketplace desain.

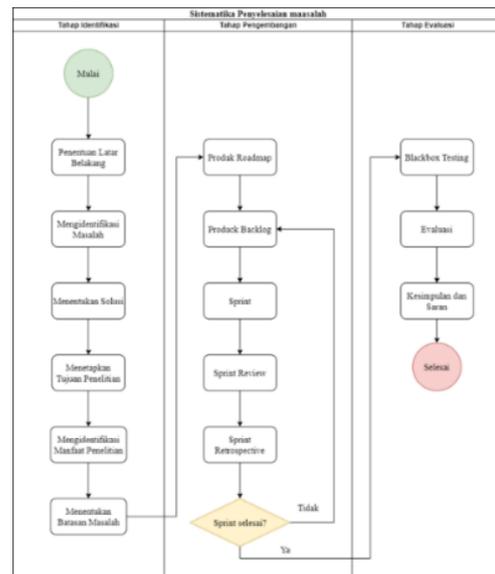
Metode yang digunakan dalam membangun website e-Marketplace GRAVIS menggunakan Scrum. Pengujian atau evaluasi menggunakan Black-box Testing. Pada tahap evaluasi ini para calon pengguna diminta untuk menjalankan fitur pada website GRAVIS sesuai dengan skenario yang telah dibuat.

Penelitian ini akan menghasilkan sebuah aplikasi berbasis web e-Marketplace yang akan digunakan untuk menghubungkan para freelancer desain dengan UMKM atau korporasi yang membutuhkan jasa desain mudah agar format makalah Anda sesuai dengan format makalah yang kami perlukan, gunakan dokumen ini sebagai *template* dan ketik teks Anda di dalamnya.

B. Sistematika Penyelesaian Masalah

Menjelaskan tentang langkah-langkah yang dilalui untuk menyelesaikan masalah menggunakan metodologi Scrum. Metode Scrum adalah metode yang termasuk dalam *agile software development*. Scrum dianggap dapat menciptakan kualitas perangkat lunak yang baik sesuai dengan keinginan pengguna dan mudah untuk mengadopsi perubahan. Tahapan aktifitas scrum meliputi produk

backlog, sprint backlog, daily scrum, sprint review, dan sprint retrospective [14].



Gambar 3. Sistematika penyelesaian masalah

Pengembangan aplikasi *e-Marketplace* GRAVIS dalam penelitian ini memiliki tiga tahap yaitu tahap identifikasi, tahap pengembangan sistem, dan tahap kesimpulan dan saran.

1. Tahapan Identifikasi

Pada tahap identifikasi akan ditentukan permasalahan yang melatarbelakangi mengapa dibutuhkannya *dashboard* bagi pengelola *website*. Tahapan ini dilakukan dengan cara melakukan riset dan analisis terhadap penelitian sebelumnya dan juga kondisi fakta empiris kasus.

2. Tahapan Pengembangan

Pengembangan produk yang berupa aplikasi *e-Marketplace* berbasis *website* GRAVIS ini digunakan metodologi *framework* Scrum. Dalam pengimplementasian *framework* Scrum memiliki beberapa tahapan [15]. Tahapan implementasi *framework* Scrum diantaranya:

- Product Backlog.** Pada tahap ini dilakukan perancangan dan analisa tentang kebutuhan aplikasi terhadap masalah yang akan dipecahkan atau diberikan suatu solusi. Mengidentifikasi permasalahan yang ada pada calon pengguna seperti *freelancer* desain dan korporasi dalam mencari jasa desain sehingga dibutuhkan aplikasi *e-Marketplace* GRAVIS. Dengan melakukan pendekatan terhadap pengguna terlebih dahulu maka kita dapat menentukan *product backlog* yang sesuai untuk aplikasi *e-Marketplace* GRAVIS.
- Sprint Planning.** Pada tahap ini dilakukan perencanaan pemenuhan kebutuhan sesuai dengan *backlog items* yang sudah ditentukan pada tahap sebelumnya. Perencanaan ini meliputi penyusunan prioritas *backlog items* dan jadwal pengerjaan *sprint*.
- Sprint.** Pada tahap ini bertujuan meningkatkan kualitas komunikasi dengan tim dalam mengerjakan *backlog items* dengan durasi waktu yang sudah ditentukan

berdasarkan tahap sebelumnya. *Daily scrum* akan dilakukan selama durasi *sprint* berlangsung.

- d) *Sprint Review*. Pada ini dilakukan pada akhir *sprint* untuk meninjau kembali fitur yang sudah diselesaikan dan melakukan pembaharuan terhadap *product backlog* bila diperlukan. Pada tahap ini, tim scrum akan meninjau apa yang sudah diselesaikan pada tahap *sprint* sehingga *backlog items* dapat dinyatakan terpenuhi.
- e) *Sprint Retrospective*. Pada tahap ini dilakukan untuk mengevaluasi *sprint* serta membuat perencanaan mengenai peningkatan yang akan dilakukan pada *sprint* berikutnya. Apabila terdapat *backlog items* yang belum terselesaikan, maka akan *backlog items* tersebut akan dilanjutkan pada *sprint* selanjutnya dan dilakukan hingga fitur tersebut terselesaikan.

3. Tahapan Evaluasi

Metode yang digunakan pada tahap evaluasi adalah *black-box testing*. Metode *black box testing* digunakan untuk menguji kesesuaian antara komponen dan fungsi pada rancangan dengan fungsi pada aplikasi yang telah dibangun [16]. Penelitian ini menggunakan metode *black*

box testing karena metode ini dianggap dapat menyediakan fasilitas validasi terhadap sudut pandang *user* terhadap aplikasi secara langsung dan cepat. Dengan metode ini dapat mengungkapkan ketidaksesuaian dan ambiguitas pengguna selama menggunakan aplikasi. Tujuan utama dalam evaluasi aplikasi ini adalah untuk memastikan komponen pada sistem aplikasi telah berfungsi sesuai dengan hasil yang diharapkan. Hal tersebut berguna agar aplikasi yang dihasilkan merupakan aplikasi yang optimal.

IV. HASIL DAN PEMBAHASAN

A. Analisis Perancangan Dashboard

1. Analisis Balance Scorecard

Balance Scorecard digunakan untuk pengukuran dan penilaian kinerja suatu perusahaan dengan mengukur empat perspektif yaitu perspektif *finance*, perspektif *customer*, perspektif *internal business process*, perspektif *innovation learning dan growth* untuk mengukur apakah aktivitas-aktivitas operasional pada suatu perusahaan dalam skala yang lebih kecil sejalan dengan sasaran yang lebih besar dalam hal visi dan strategi

TABEL IIIII
BALANCE SCORECARD

Perspektif	Target	Deskripsi	Indikator
<i>Finance</i>	Peningkatan pendapatan	Pendapatan didapat dari 10% komisi yang diambil dari transaksi <i>freelancer</i>	10% dari penghasilan <i>freelancer</i>
		Banyaknya total transaksi berhasil yang dilakukan dalam <i>website</i>	Total banyaknya transaksi yang dilakukan
<i>Customer</i>	Peningkatan jumlah <i>user</i> pada <i>website</i>	Banyaknya <i>freelancer</i> atau <i>client</i> yang mendaftar pada <i>website</i>	Jumlah <i>user</i> yang terdaftar
	Kepuasan dan kepercayaan <i>customer</i>	<i>User</i> yang melakukan paling banyak transaksi	<i>User</i> dengan jumlah transaksi paling banyak
	Peningkatan jumlah produk	Banyaknya jumlah produk yang dibuat	Jumlah produk yang dibuat <i>user</i>
<i>Internal business process</i>	Meningkatkan efisiensi kerja admin	Banyaknya jumlah transaksi yang belum di <i>approve</i> oleh admin	Jumlah transaksi yang belum di <i>approve</i>
<i>Innovation learning dan growth</i>	Mengetahui trend desain terkini	Kategori dengan produk yang paling banyak	Jumlah produk dalam suatu kategori

Tabel 1 menunjukkan terdapat 6 target untuk mengukur kinerja dari *website*. Untuk mencapai target-target tersebut maka dibutuhkan data kredibel, data-data tersebut merupakan *key performance indicator* yang akan diambil langsung dari *website* GRAVIS dengan secara *real time*.

2. Analisis Key Performance Indicator

Key Performance Indicator (KPI) adalah parameter yang digunakan untuk mengukur performa suatu organisasi, perusahaan atau individu untuk memudahkan dalam mengidentifikasi tujuannya telah tercapai atau gagal. Adapun KPI yang digunakan dalam *dashboard e-marketplace* adalah sebagai berikut

TABEL IVV
KEY PERFORMANCE INDICATOR

No	KPI	Deskripsi
1	Total Users	Total pengguna jasa <i>website</i> GRAVIS
2	Total Product	Jumlah <i>product</i> yang ada di dalam <i>website</i> GRAVIS
3	Total Value Transactions	Total nilai transaksi
4	Pending Transaction	Transaksi tertunda
5	Ongoing Transaction	Transaksi berjalan
6	Complete Transaction	Transaksi selesai
7	Revenue	Arus pemasukan yang diperoleh
8	Total Freelance Transaction	Jumlah transaksi yang dilakukan oleh <i>freelance</i>
9	Total Client Transaction	Jumlah transaksi yang dilakukan oleh <i>client</i>
10	New Users	Pengguna baru dalam kurun waktu tertentu
11	Product per Category	Jumlah produk yang ditawarkan berdasarkan kategori
12	Best Selling Product	Produk paling sering dibeli
13	Reporting	Memantau jumlah nilai transaksi yang terjadi pada <i>website</i> , melihat pendapatan dari <i>website</i> , dan melihat total pendapatan <i>freelancer</i>
14	User List	Melihat dan mengatur <i>user</i> yang ada pada <i>website</i>
15	Product List	Melihat dan mengatur produk yang ada pada <i>website</i>
16	Category List	Melihat dan mengatur kategori produk yang ada pada <i>website</i>

TABEL V
AKTOR

No.	Aktor	Deskripsi
1.	Admin	Aktor ini memiliki wewenang untuk mengubah menambah dan menghapus kategori pada <i>website</i> aktor ini juga dapat menambah <i>user</i> baru dan mengubah data <i>user</i> jika di perlukan dan juga memberikan akses admin kepada <i>user</i> . Aktor ini juga dapat membuat dan menghapus produk pada <i>website</i> . Admin juga bertanggung jawab dengan transaksi yang berjalan didalam <i>website</i> .
2.	Member	Aktor ini dapat disebut sebagai prosumer yaitu memiliki dua peran sekaligus pada <i>website</i> ini. Peran pertama yaitu sebagai pemilik proyek atau sebagai konsumen. Peran yang kedua yaitu sebagai <i>freelancer</i> atau produsen. Aktor ini dapat melakukan pemesanan jasa <i>freelancer</i> maupun menyediakan jasa sebagai <i>freelancer</i> . Aktor ini tidak memiliki akses ke halaman admin.

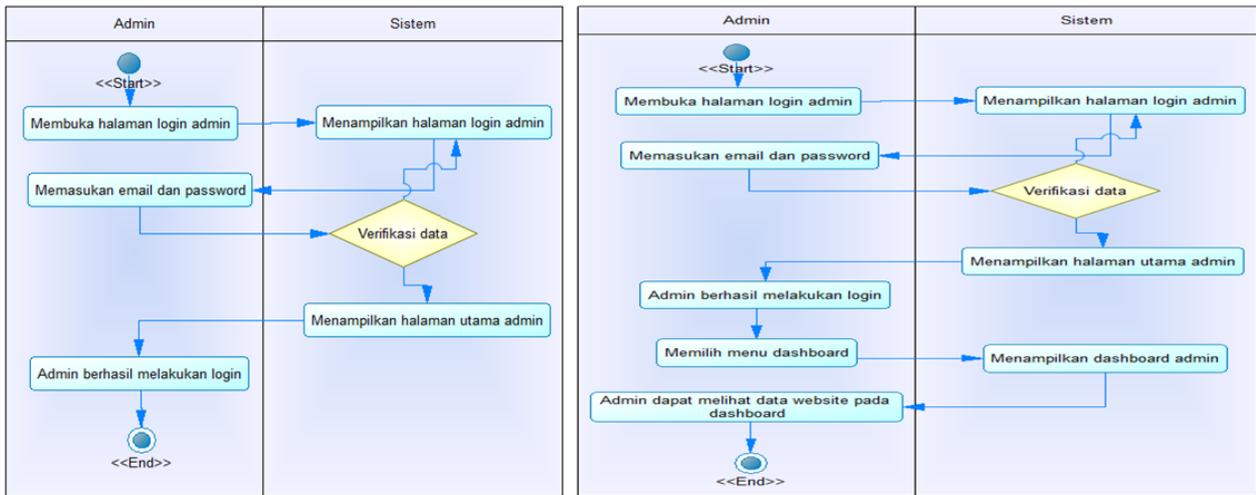
TABEL VI
KEBUTUHAN FITUR

Halaman Admin			
No.	Kebutuhan Fitur	Deskripsi	Aktor
1.	Login	Fitur yang digunakan admin untuk masuk kedalam halaman admin. Halaman admin hanya bisa diakses oleh peran admin	Admin
2.	Logout	Fitur yang digunakan admin untuk keluar dari halaman admin	Admin
3.	Dashboard Admin	Halaman berisi informasi penting yang akan digunakan oleh admin maupun informasi yang digunakan untuk pengembangan <i>startup</i>	Admin
4.	Transaksi	Halaman ini digunakan untuk mengelola transaksi mulai dari tahap <i>approve</i> hingga transaksi selesai dilakukan.	Admin
5.	User	Halaman ini digunakan untuk mengelola <i>user</i> yang terdaftar pada <i>website</i> GRAVIS	Admin
6.	Kategori	Halaman ini digunakan untuk mengelola kategori yang akan ditampilkan pada halaman utama <i>website</i> GRAVIS	Admin
7.	Produk	Halaman ini digunakan untuk mengelola seluruh produk yang ada pada <i>website</i> GRAVIS	Admin
8.	Report	Halaman ini berisikan data transaksi yang sudah selesai dilakukan. Halaman ini juga memiliki data pendapatan dari <i>startup</i> GRAVIS	Admin

B. Activity Diagram

Terdapat 4 activity diagram diantaranya yaitu login

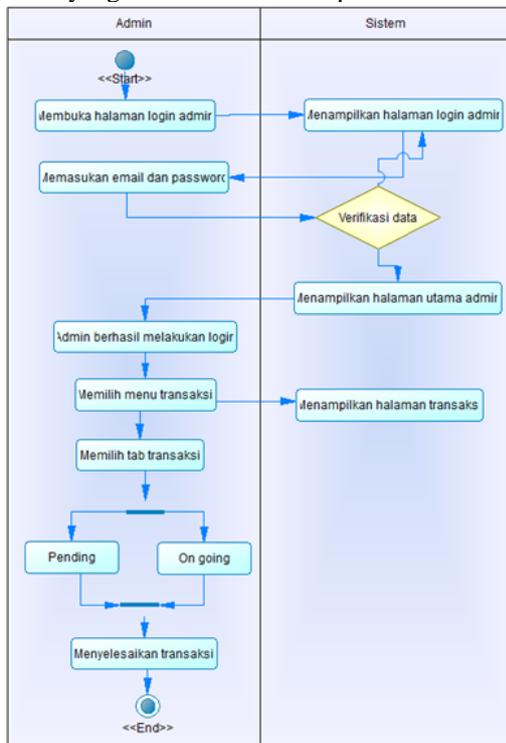
admin, melihat dashboard admin, konfirmasi transaksi, dan melihat report.



Gambar 4. Activity diagram login dan melihat dashboard admin

Aktivitas dimulai ketika admin mengakses halaman login dan memasukkan data email dan password yang sebelumnya telah terdaftar pada sistem lalu sistem akan melakukan validasi data apakah data tersebut sesuai dengan database, jika sesuai maka sistem akan menampilkan halaman utama admin. Setelah admin berhasil melakukan login admin akan memilih menu dashboard pada side bar maka sistem akan menampilkan dashboard yang berisikan data-data pada website GRAVI

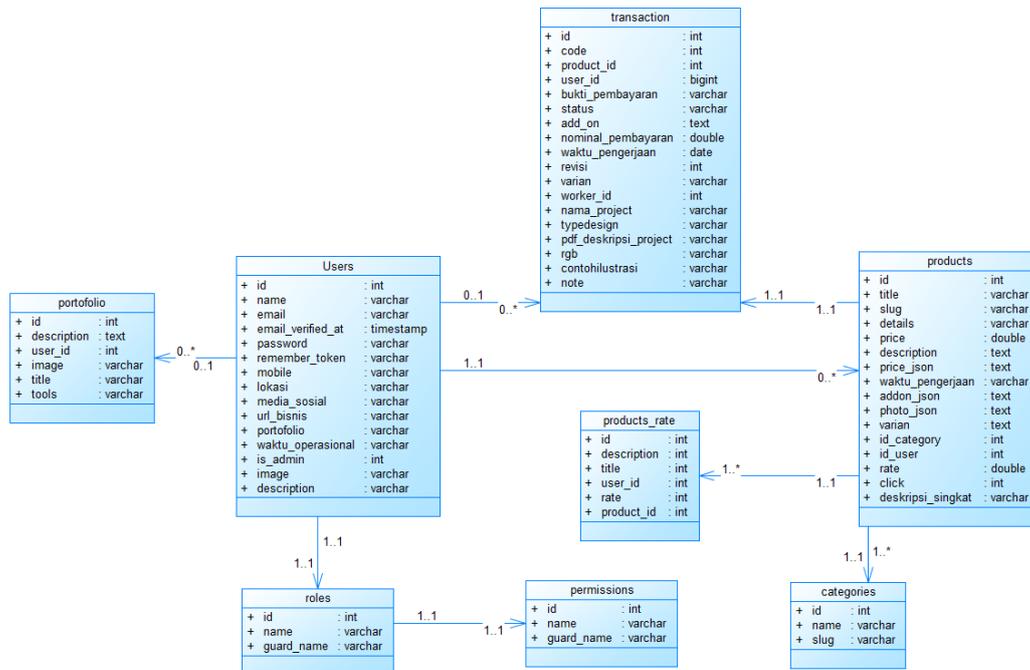
Aktivitas dimulai ketika admin mengakses halaman login dan memasukkan data email dan password. Setelah berhasil melakukan login admin akan memilih menu transaksi pada side bar sistem akan menampilkan halaman transaksi. Untuk menerima transaksi, admin akan masuk ke tab pending dan menerima transaksi yang masih pending. Lalu pada tab ongoing admin dapat melihat data transaksi yang telah selesai dilakukan dan admin dapat menyelesaikan transaksi dengan mengirimkan dana freelancer kepada freelancer yang bersangkutan.



Gambar 5. Activity diagram konfirmasi transaksi

C. Entity Relationship Diagram

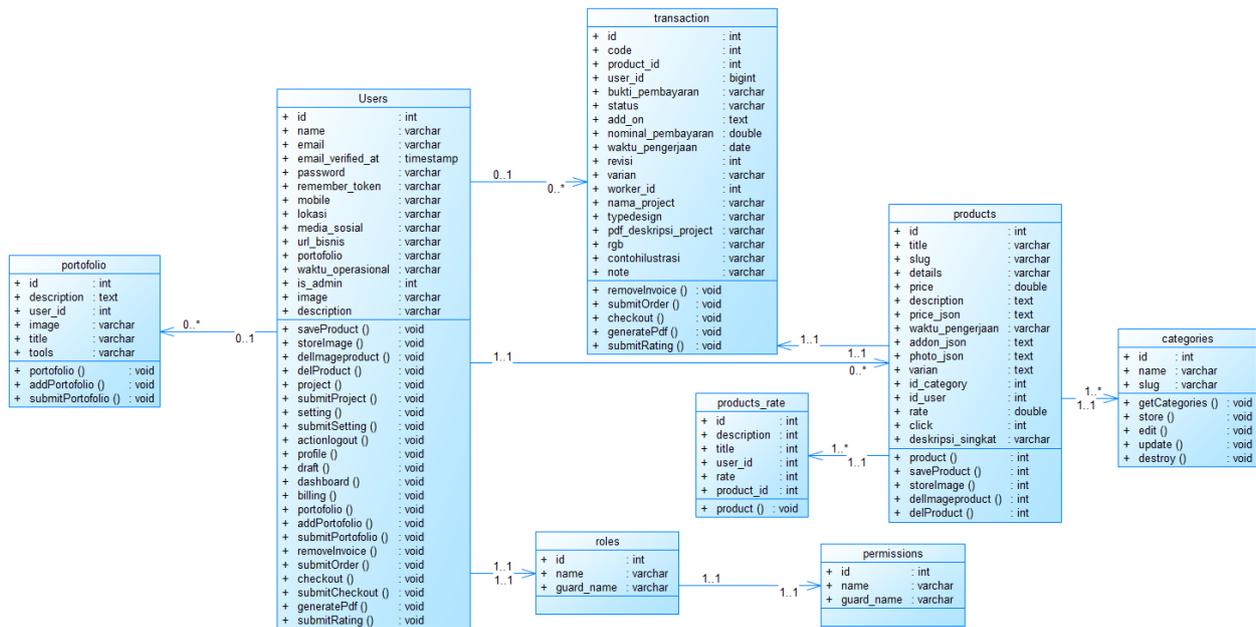
Entity Relationship Diagram digunakan untuk menggambarkan interkoneksi antara tabel individu sehingga terbentuk database



Gambar 6. Entity relationship diagram

D. Class Diagram

Class diagram adalah Diagram kelas UML, representasi visual dari kelas sistem dan hubungannya.

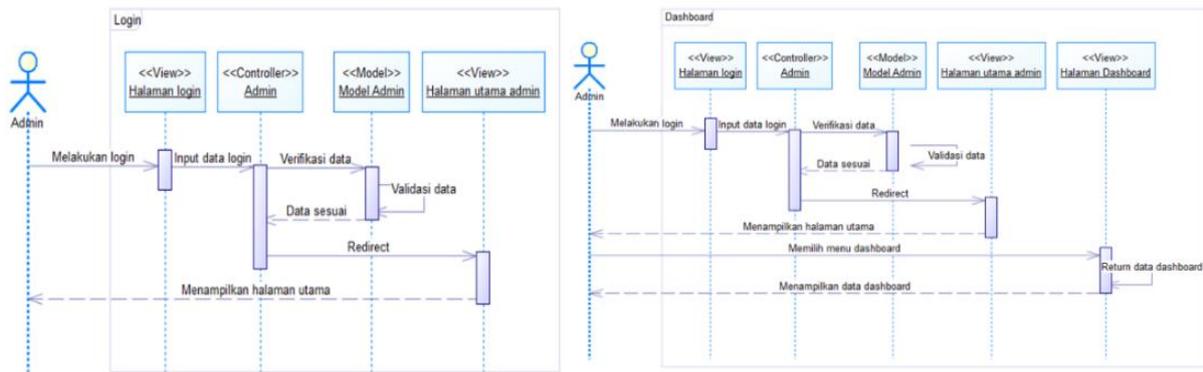


Gambar 7. Class diagram

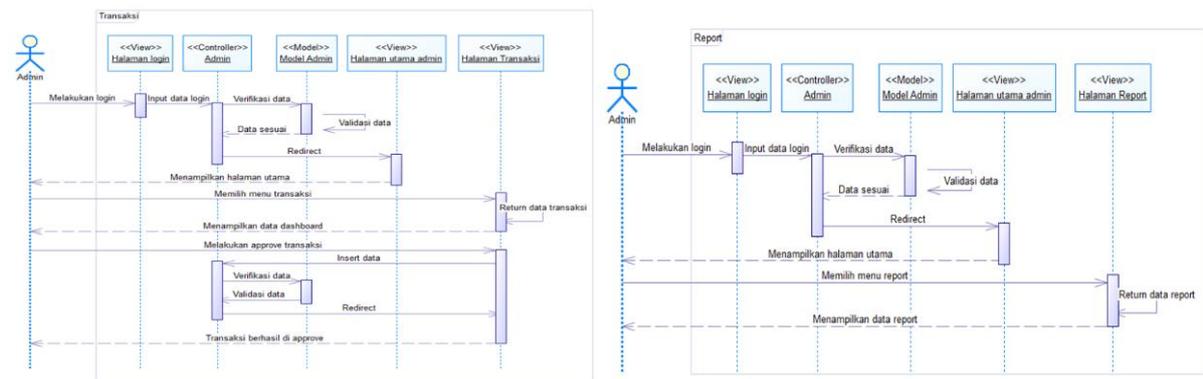
Gambar 7 menunjukkan diagram kelas dari situs GRAVIS, dan di dalam gambar tersebut terdapat hubungan antar diagram kelas; koneksi ini ada karena tindakan tertentu termasuk dalam setiap diagram kelas.

E. Sequence Diagram

Sequence diagram adalah sebuah diagram yang digunakan untuk menjelaskan dan menampilkan interaksi antar objek-objek dalam sebuah sistem secara terperinci. Selain itu *sequence diagram* juga akan menampilkan pesan atau perintah yang dikirim, beserta waktu pelaksanaannya. Terdapat 4 *sequence diagram* sebagai berikut:



Gambar 8. Sequence diagram login dan dashboard

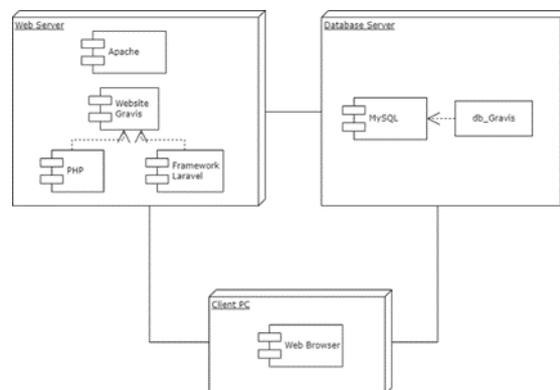


Gambar 9. Sequence diagram transaksi dan report

Perancangan *sequence diagram* menggunakan penerapan *model, view, controller* (MVC). *Model* merupakan struktur yang bertugas dalam membantu setiap proses yang akan berhubungan dengan *database* seperti memasukkan data, mengedit data dan menghapus data dalam sebuah *database*. Sedangkan *view* merupakan desain tampilan yang ditampilkan sesuai dengan data yang sudah diproses pada *model*, dan *controller* berfungsi sebagai pengatur segala data yang diproses pada *model* untuk ditampilkan pada *view* sehingga *controller* juga merupakan sebuah perantara antara *model* dan *view*.

F. Deployment Diagram

Deployment Diagram adalah sejenis model diagram UML yang digunakan untuk menggambarkan proses memasukkan objek ke dalam *node*. Ada tiga *node* dalam diagram *Deployment*, dan garis yang menghubungkan mereka semua menjelaskan bagaimana mereka berbicara satu sama lain. Peramban *web* dipasang pada simpul klien, dan simpul klien memiliki koneksi satu-ke-satu ke *server* basis data dan *server web*. MySQL DBMS berada pada *node server database*, yang juga terhubung ke *node web server* dan *node klien*. *Node* terakhir adalah *node server web*, dan berisi komponen *server* Apache. *Node* ini juga berkomunikasi dengan *server database* dan *node klien*



Gambar 10. Deployment diagram

G. Implementasi Scrum

1. Sprint 1

Pada *sprint* pertama ini akan mulai dirancang halaman utama *dashboard* admin sesuai dengan urutan prioritas yang telah ditentukan sebelumnya pada produk *backlogitem*.

2. Sprint 1 Planning

Pada *sprint planning* ini akan disusun *backlog* yang akan dikerjakan dan rentang waktu *sprint* akan dilakukan. Pada *sprint* yang pertama terdapat 2 *backlog* yang akan

dikembangkan yaitu total *user*, total transaksi, dan jumlah produk pada halaman *dashboard* admin.

TABEL VII
SPRINT 1 PLANNING

No.	Backlog	Deskripsi	Periode
1.	Dashboard admin total <i>user</i> dan transaksi	Digunakan untuk memantau pengguna dan jumlah transaksi yang dilakukan pengguna pada <i>website</i> GRAVIS	07-11 Maret 2022
2.	Dashboard admin jumlah produk	Digunakan untuk memantau produk yang ada pada <i>website</i> GRAVIS dan menambahkan kategori produk jika diperlukan	

3. Daily Scrum 1

Pada tahap *daily scrum* akan dilakukan evaluasi setiap harinya untuk mengetahui apa yang telah dikerjakan pada hari ini, apa saja kendala yang dialami, dan apa yang akan dikerjakan di hari berikutnya. Berikut ini merupakan hasil dari pengerjaan *backlog* pada *sprint* pertama.

TABEL VIII
DAILY SCRUM 1

Backlog	Keterangan
Dashboard admin total <i>user</i> dan transaksi	Selesai
Dashboard admin jumlah produk	Selesai

4. Sprint 1 Review

Pada tahap *sprint review* akan dilakukan evaluasi terhadap seluruh *backlog* yang telah dikerjakan. Berikut merupakan hasil dari *sprint review* pertama.

TABEL IX
SPRINT 1 REVIEW

Backlog	Keterangan
Dashboard admin total <i>user</i> dan transaksi	Sesuai dengan kebutuhan
Dashboard admin jumlah produk	Sesuai dengan kebutuhan

5. Sprint 1 Retrospective

Pada tahap *sprint retrospective* dilakukan untuk mengevaluasi terhadap kinerja team dalam melaksanakan *sprint*. Tahap ini dilakukan dengan menjawab 3 pertanyaan penting yaitu *what went well?*, *what went wrong?*, *what can be improved?*. Berikut ini merupakan hasil dari *sprint retrospective* yang pertama.

TABEL X
SPRINT 1 RETROSPECTIVE

Pertanyaan	Jawaban
<i>What went well?</i>	Seluruh <i>backlog</i> terselesaikan
<i>What went wrong?</i>	-
<i>What can be improved?</i>	-

6. Sprint 2

Sprint kedua akan berfokus pada halaman transaksi yang akan digunakan admin untuk memonitoring transaksi yang dilakukan oleh pengguna *website*.

7. Sprint 2 Planning

Pada *sprint* yang kedua terdapat 2 *backlog* yang akan dikembangkan pada halaman transaksi.

TABEL XI
SPRINT 2 PLANNING

No.	Backlog	Deskripsi	Periode
1.	Halaman transaksi <i>pending</i>	Digunakan untuk memantau transaksi antar pengguna dan melakukan <i>approve</i> transaksi yang masuk	22-25 Maret 2022
2.	Halaman transaksi <i>ongoing</i>	Digunakan untuk memantau transaksi antar pengguna dan menyelesaikan transaksi yang sedang berlangsung	

8. Daily Scrum 2

Berikut ini merupakan hasil dari pengerjaan *backlog* pada *sprint* kedua.

TABEL XII
DAILY SCRUM 2

Backlog	Keterangan
Halaman transaksi <i>pending</i>	Tidak selesai
Halaman transaksi <i>ongoing</i>	Selesai

9. Sprint 2 Review

Berikut merupakan hasil dari *sprint review* kedua.

TABEL XIII
SPRINT 2 REVIEW

Backlog	Hasil Review
Halaman transaksi <i>pending</i>	Belum sesuai dengan kebutuhan
Halaman transaksi <i>ongoing</i>	Sesuai dengan kebutuhan

10. Sprint 2 Retrospective

Berikut ini merupakan hasil dari *sprint retrospective* yang kedua.

TABEL XIV
SPRINT 2 RETROSPECTIVE

Pertanyaan	Jawaban
<i>What went well?</i>	Ada <i>backlog</i> yang tidak terselesaikan
<i>What went wrong?</i>	Terdapat kesalahan pada halaman transaksi <i>pending</i>
<i>What can be improved?</i>	Komunikasi dengan tim harus ditingkatkan

11. Sprint 3

Sprint ketiga masih berfokus pada halaman transaksi yang akan digunakan admin untuk memonitoring transaksi yang dilakukan oleh pengguna *website*. Pada *sprint* kedua terdapat *product backlog* yang tidak terselesaikan maka akan dilanjutkan pada *sprint* ketiga.

12. Sprint 3 Planning

Pada *sprint* yang ketiga terdapat 2 *backlog* baru dan 1 *backlog* turunan dari *sprint* sebelumnya.

TABEL XV
SPRINT 3 PLANNING

No.	Backlog	Deskripsi	Periode
1.	Memantau transaksi yang sudah selesai dilakukan oleh pengguna	Melakukan pengembangan untuk melihat list transaksi yang telah diselesaikan oleh pengguna	28 Maret – 01 April 2022
2.	Memantau transaksi yang ditolak oleh admin	Melakukan pengembangan untuk melihat list transaksi yang ditolak oleh admin	
3.	Perbaikan halaman transaksi <i>pending</i>	Melakukan perbaikan kesalahan pada halaman transaksi <i>pending</i>	

13. Daily Scrum 3

Berikut ini merupakan hasil dari pengerjaan *backlog* pada *sprint* ketiga.

TABEL XVI
DAILY SCRUM 3

Backlog	Keterangan
Memantau transaksi yang sudah selesai dilakukan oleh pengguna	Selesai
Memantau transaksi yang ditolak oleh admin	Selesai
Perbaikan halaman transaksi <i>pending</i>	Selesai

14. Sprint 3 Review

Berikut merupakan hasil dari *sprint review* ketiga.

TABEL XVII
SPRINT 3 REVIEW

Backlog	Hasil Review
Memantau transaksi yang sudah selesai dilakukan oleh pengguna	Sesuai dengan kebutuhan
Memantau transaksi yang ditolak oleh admin	Sesuai dengan kebutuhan
Perbaikan halaman transaksi <i>pending</i>	Sesuai dengan kebutuhan

15. Sprint 3 Retrospective

Berikut ini merupakan hasil dari *sprint retrospective* yang ketiga.

TABEL XVIII
SPRINT 3 RETROSPECTIVE

Pertanyaan	Jawaban
What went well?	Seluruh <i>backlog</i> terselesaikan, Komunikasi tim membaik
What went wrong?	-
What can be improved?	-

16. Sprint 4

Sprint keempat akan berfokus pada pengembangan halaman *report* yang akan digunakan admin untuk mengevaluasi *revenue* transaksi pada *website*.

17. Sprint 4 Planning

Pada *sprint* yang keempat ini terdapat 2 *backlog* yang akan dikembangkan.

TABEL XIX
SPRINT 4 PLANNING

No.	Backlog	Deskripsi	Periode
1.	Halaman <i>report</i>	Memantau jumlah nilai transaksi yang terjadi pada <i>website</i> , melihat pendapatan dari <i>website</i> , dan melihat total pendapatan <i>freelancer</i>	04-08 April 2022
2.	<i>Chart Revenue</i>	Membuat <i>chart</i> pendapatan yang didapat oleh <i>website</i> GRAVIS	

18. Daily Scrum 4

Berikut ini merupakan hasil dari pengerjaan *backlog* pada *sprint* keempat.

TABEL XX
DAILY SCRUM 4

Backlog	Keterangan
Halaman <i>report</i>	Selesai
<i>Chart Revenue</i>	Tidak selesai

19. Sprint 4 Review

Berikut merupakan hasil dari *sprint review* keempat.

TABEL XXI
SPRINT 4 REVIEW

Backlog	Hasil Review
Halaman <i>report</i>	Sesuai dengan kebutuhan
<i>Chart Revenue</i>	Belum sesuai dengan kebutuhan

20. Sprint 4 Retrospective

Berikut ini merupakan hasil dari *sprint retrospective* yang keempat.

TABEL XXII
SPRINT 4 RETROSPECTIVE

Pertanyaan	Jawaban
What went well?	Komunikasi tim berjalan dengan baik
What went wrong?	Pembuatan <i>chart</i> tertunda
What can be improved?	-

21. Sprint 5

Sprint kelima ini akan berfokus pada pengembangan *chart* pada halaman utama *dashboard* admin yang akan digunakan admin untuk mengevaluasi kinerja dari *website*.

22. Sprint 5 Planning

Pada *sprint* yang kelima ini terdapat 2 *backlog* yang akan dikembangkan.

TABEL XXIII
SPRINT 5 PLANNING

No.	Backlog	Deskripsi	Periode
1.	<i>Chart Revenue</i>	Membuat <i>chart</i> pendapatan dari <i>website</i> GRAVIS	11-15 April 2022
2.	<i>Chart</i> total transaksi <i>user</i>	Membuat <i>chart</i> total transaksi <i>user</i> pada <i>website</i> GRAVIS	

23. Daily Scrum 5

Berikut ini merupakan hasil dari pengerjaan *backlog* pada *sprint* kelima.

TABEL XXIV
DAILY SCRUM 5

Backlog	Keterangan
Chart Revenue	Selesai
Chart total transaksi user	Selesai

24. Sprint 5 Review

Berikut merupakan hasil dari *sprint review* kelima.

TABEL XXV
SPRINT 5 REVIEW

Backlog	Hasil Review
Chart Revenue	Sesuai dengan kebutuhan
Chart total transaksi user	Sesuai dengan kebutuhan

25. Sprint 5 Retrospective

Berikut ini merupakan hasil dari *sprint retrospective* yang kelima.

TABEL XXVI
SPRINT 5 RETROSPECTIVE

Pertanyaan	Jawaban
What went well?	Semua backlog terselesaikan
What went wrong?	Komunikasi tim yang kurang
What can be improved?	Daily scrum dilakukan tepat waktu

26. Sprint 6

Sprint keenam akan berfokus pada pengembangan *chart* pada halaman utama *dashboard* admin yang akan digunakan admin untuk mengevaluasi kinerja dari *website*.

27. Sprint 6 Planning

Pada *sprint* yang keenam ini terdapat 2 *backlog* yang akan dikembangkan.

TABEL XXVII
SPRINT 6 PLANNING

No.	Backlog	Deskripsi	Periode
1.	Chart total nilai transaksi user	Membuat <i>chart</i> total nilai transaksi user pada <i>website</i> GRAVIS	18-22 April 2022
2.	Chart jumlah user	Membuat <i>chart</i> total jumlah user pada <i>website</i> GRAVIS	

28. Daily Scrum 6

Berikut ini merupakan hasil dari pengerjaan *backlog* pada *sprint* keenam.

TABEL XXVIII
DAILY SCRUM 6

Backlog	Keterangan
Chart total nilai transaksi user	Selesai
Chart jumlah user	Selesai

29. Sprint 6 Review

Berikut merupakan hasil dari *sprint review* keenam.

TABEL XXIX
SPRINT 6 REVIEW

Backlog	Hasil Review
Chart total nilai transaksi user	Sesuai dengan kebutuhan
Chart jumlah user	Sesuai dengan kebutuhan

30. Sprint 6 Retrospective

Berikut ini merupakan hasil dari *sprint retrospective* yang keenam.

TABEL XXX
SPRINT 6 RETROSPECTIVE

Pertanyaan	Jawaban
What went well?	Semua backlog terselesaikan
What went wrong?	-
What can be improved?	-

31. Sprint 7

Sprint ketujuh berfokus pada pengembangan *chart* pada halaman utama *dashboard* admin yang akan digunakan admin untuk mengevaluasi kinerja dari *website*.

32. Sprint 7 Planning

Pada *sprint* yang ketujuh ini terdapat 2 *backlog* yang akan dikembangkan.

TABEL XXXI
SPRINT 7 PLANNING

No.	Backlog	Deskripsi	Periode
1.	Chart jumlah produk pada setiap kategori	Membuat <i>chart</i> jumlah produk pada setiap kategori	25-29 April 2022
2.	Chart produk yang paling sering dibeli	Membuat <i>chart</i> pada produk yang paling sering dibeli	

33. Daily Scrum 7

Berikut ini merupakan hasil dari pengerjaan *backlog* pada *sprint* pertama.

TABEL XXXII
DAILY SCRUM 7

Backlog	Keterangan
Chart jumlah produk pada setiap kategori	Selesai
Chart produk yang paling sering dibeli	Selesai

34. Sprint 7 Review

Berikut merupakan hasil dari *sprint review* ketujuh.

TABEL XXXIII
SPRINT 7 REVIEW

Backlog	Hasil Review
Chart jumlah produk pada setiap kategori	Sesuai dengan kebutuhan
Chart produk yang paling sering dibeli	Sesuai dengan kebutuhan

35. Sprint 7 Retrospective

Berikut ini merupakan hasil dari *sprint retrospective* yang ketujuh.

TABEL XXXIV
SPRINT 7 RETROSPECTIVE

Pertanyaan	Jawaban
What went well?	Semua <i>backlog</i> terselesaikan. Komunikasi berjalan dengan baik sehingga <i>bug</i> dapat langsung diatasi
What went wrong?	-
What can be improved?	-

36. *Sprint 8*

Sprint kedelapan ini akan berfokus pada perbaikan fitur dari *sprint* – *sprint* sebelumnya agar meminimalisir kesalahan pengguna selama menggunakan *website*.

37. *Sprint 8 Planning*

Pada *sprint* yang kedelapan ini terdapat 1 *backlog* sebagai berikut.

TABEL XXXV
SPRINT 8 PLANNING

No.	Backlog	Deskripsi	Periode
1.	Melakukan <i>bug fixing</i>	Memperbaiki <i>bug</i> atau <i>typo</i> pada fitur yang telah dibuat	09-13 Mei 2022

38. *Daily Scrum 8*

Berikut ini merupakan hasil dari pengerjaan *backlog* pada *sprint* kedelapan.

TABEL XXXVVI
DAILY SCRUM 8

Backlog	Keterangan
<i>Bug fixing</i>	Selesai

39. *Sprint 8 Review*

Berikut merupakan hasil dari *sprint review* kedelapan.

TABEL XXXVII
SPRINT 8 REVIEW

Backlog	Hasil Review
<i>Bug fixing</i>	Sesuai dengan kebutuhan

40. *Sprint 8 Retrospective*

Berikut ini merupakan hasil dari *sprint retrospective* yang kedelapan.

TABEL XXXVIII
SPRINT 8 RETROSPECTIVE

Pertanyaan	Jawaban
What went well?	Semua <i>backlog</i> terselesaikan
What went wrong?	-
What can be improved?	-

H. *Proses Pengujian Perangkat Lunak*

Tujuan pengujian perangkat lunak adalah untuk memberikan jawaban pasti atas pertanyaan tentang kelangsungan hidup program dan perilaku yang diharapkan serta menemukan *bug* dalam perangkat lunak.

1. *Black-box Testing*

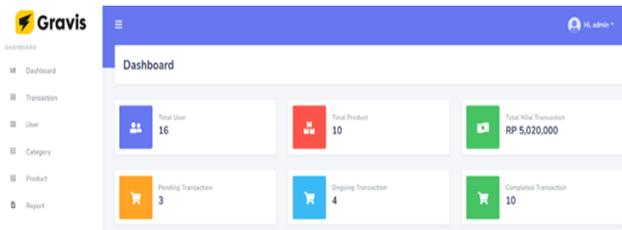
Pengujian menggunakan *black-box testing* untuk menguji hasil dari *input* dan *output* dari fitur yang telah dibuat. Pengujian ini dilakukan dengan mengasumsikan sudut pandang pengguna yaitu pengelola *website* atau admin

TABEL XXXIX
HASIL *BLACK-BOX TESTING*

No	Skenario yang diujikan	Langkah pengujian	Hasil yang diharapkan	Hasil pengujian	Status
1	Admin melakukan login	1. Admin masuk ke halaman <i>login</i> admin 2. Admin memasukan <i>email</i> dan <i>password</i> yang sudah terdaftar sebagai admin 3. Admin menekan tombol <i>login</i>	Sistem menampilkan halaman admin	Setelah menekan tombol <i>login</i> admin berhasil masuk ke halaman admin	Sukses
2	Dashboard admin	1. Sistem menampilkan <i>dashboard</i> admin 2. Admin dapat melihat data-data yang ada pada <i>website</i> 3. Admin dapat melihat <i>graph</i> kesimpulan data-data	Sistem akan menampilkan data-data dan <i>chart</i>	Setelah berhasil melakukan <i>login</i> sistem menampilkan halaman <i>dashboard</i>	Sukses
3	Transaksi	1. Admin masuk ke menu transaksi 2. Sistem menampilkan halaman transaksi 3. Admin dapat melihat data transaksi <i>pending</i> dan <i>ongoing</i>	Sistem akan menampilkan data transaksi <i>pending</i> dan <i>ongoing</i>	Setelah memilih menu transaksi sistem berhasil menampilkan data transaksi <i>pending</i> dan <i>ongoing</i>	Sukses
4	Transaksi Pending	1. Admin masuk ke halaman transaksi 2. Admin masuk ke tab <i>pending</i> 3. Admin dapat melihat data transaksi yang perlu di konfirmasi 4. Admin dapat melakukan konfirmasi transaksi	Admin berhasil melakukan konfirmasi transaksi	Setelah menekan tombol <i>approve</i> sistem berhasil menerima transaksi	Sukses
5	Transaksi Ongoing	1. Admin masuk ke halaman transaksi 2. Admin masuk ke tab <i>ongoing</i> 3. Admin menekan tombol <i>set complete</i> 4. Admin berhasil menyelesaikan transaksi	Admin berhasil menyelesaikan transaksi	Setelah menekan tombol <i>set complete</i> sistem berhasil menyelesaikan transaksi lalu data masuk ke halaman <i>report</i>	Sukses
6	Report	1. Admin masuk ke halaman <i>report</i> 2. Admin dapat melihat data <i>report</i> yang dimiliki <i>website</i>	Sistem akan menampilkan data <i>reporting</i>	Setelah masuk ke menu <i>report</i> sistem berhasil menampilkan data yang dibutuhkan	Sukses

I. Visualisasi Perancangan Dashboard

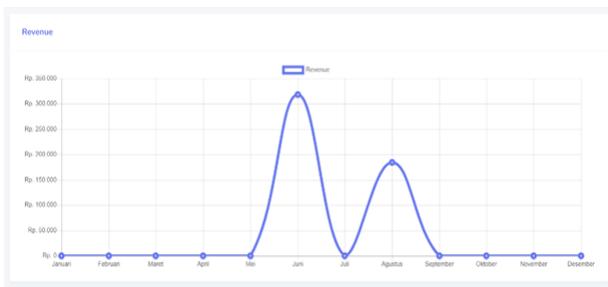
Hasil akhir perancangan dashboard dalam penelitian ini akan menampilkan informasi dengan line chart dan bar chart



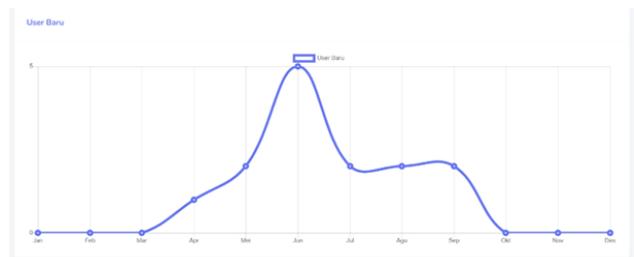
Gambar 11. Hasil akhir tampilan menu *dashboard*



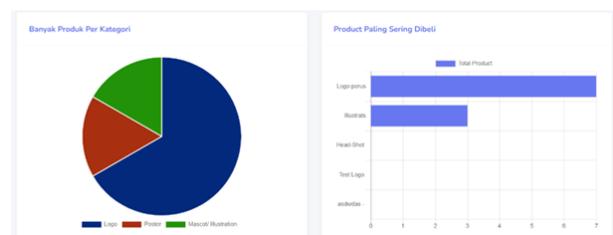
Gambar 13. Hasil akhir tampilan transaksi berupa *bar chart*



Gambar 12. Hasil akhir tampilan revenue berupa *line chart*



Gambar 14. Hasil akhir data pengguna baru berupa *bar chart*



Gambar 15. Hasil akhir data product baru berupa *pie chart* dan *bar chart*

V. KESIMPULAN

Penelitian ini menghasilkan beberapa temuan untuk menjawab tujuan penelitian yaitu :

1. *Key performance indicator* (KPI) yang digunakan meliputi 16 aspek: (1) *Total Product*, (2) *Total Value Transactions*, (3) *Pending Transaction* (4) *Ongoing Transaction*, (5) *Complete Transaction*, (6) *Revenue*, (7) *Total User*, (8) *Total Freelance Transaction*, (9) *Total Client Transaction*, (10) *New Users*, (11) *Product per Category*, (12) *Best Selling Product*, (13) *Reporting*, (14) *User List*, (15) *Product List*, (16) *Category List*
2. *Dashboard* dirancang dengan menampilkan 16 bagian yaitu, *total users*, *total product*, *total value transactions*, *pending transaction*, *ongoing transaction*, *complete transaction*, *revenue*, *total freelance transaction*, *total client transaction*, *new users*, *product per category*, *best-selling product*, *reporting*, *user list*, *product list*, *category list*. *Dashboard* akan menampilkan informasi dengan *line chart* dan *bar chart*.
3. Pada pengujian *dashboard* admin menggunakan *black-box testing* dengan 6 *scenario testing*. Hasil dari 6 *scenario black-box testing* mendapatkan total sukses sebanyak 6 dan 0 *scenario* gagal, maka dari itu hasil dari pengujian *black-box testing* selesai pada iterasi pertama.

REFERENSI

- [1] Apriadi D, Saputra AY. Jurnal Resti. Resti. 2017;1(1):19–25.
- [2] Wahyuni N, Irman A, Mutaqin S, Gunawan A. Pengenalan Dan Pemanfaatan Marketplace E-Commerce. J Pengabd Din. 2019;6(1).
- [3] Ilhamsyah, Rahmayudha S. Perancangan Model Dashboard Untuk Monitoring Evaluasi Mahasiswa. J Inform Pengemb IT. 2017;2(1):13–7.
- [4] Mahajan S, Parekh M, Patel H, Patil S. BRB dashboard: A web-based statistical dashboard. Proc 2017 Int Conf Innov Information, Embed Commun Syst ICIIECS 2017. 2018;2018-Janua:1–6.
- [5] Velimirovi D, Velimirovi M, Stankovi R. Role and Importance of Key Performance. Vol. 6, Serbian Journal of Management 6 (1) (2011) 63 - 72. 2011. p. 63–72.
- [6] Yuniarsih T, Suwatno. Manajemen Sumber Daya Manusia. Cetakan ke. Bandung: Alfabeta; 2011. 226 p.
- [7] RUIPEREZ-VALIENTE JA, GOMEZ MJ, MARTINEZ PA, KIM YJ. Ideating and Developing a Visualization Dashboard to Support Teachers Using Educational Games in the Classroom. IEEE Access. 2021;9:83467–81.
- [8] Wibowo S, Andreswari R, Hasibuan MA. Analysis and design of decision support system dashboard for predicting student graduation time. Int Conf Electr Eng Comput Sci Informatics. 2018;2018-Octob:684–9.
- [9] Saputra FA, Gartina Husein I, Qana'a M. Perancangan Dashboard Untuk Monitoring Performa Mahasiswa D3 Sistem Informasi Fakultas Ilmu Terapan Berbasis Web. Proceeding Appl Sci. 2021;7(6):2713.
- [10] Iqbal TMR, Gumilang SF, Alam PF. Perancangan Aplikasi Start-Up Rental Kendaraan Egarage . Id Berbasis Android Platform Untuk Solusi Transportasi Dengan Metode Agile Iterative Dan Incremental Design of Start-Up Egarage . Id a Vehicle Rental Application Based on Android Platform for Transp. e-Proceeding Eng. 2019;6(2):8131–5.
- [11] Sarikaya A, Correll M, Bartram L, Tory M, Fisher D. What Do We Talk About When We Talk About Dashboards? IEEE Trans Vis Comput Graph. 2019;25(1):682–92.
- [12] Few S. Information dashboard design: the effective visual communication of data [Internet]. First edition. Beijing ; Cambridge [MA]: O'Reilly, [2006] ©2006; Available from: <https://search.library.wisc.edu/catalog/9910016831702121>
- [13] Mowen MM, Hansen DR, Heitger DL. Managerial accounting : the cornerstone of business decision making (6th ed.). United States: Cengage Learning; 2016. 784 p.
- [14] Mariza I. Pengukuran Kinerja Dengan Balanced Scorecard. The Winners [Internet]. 2003 Sep 30;4(2):127. Available from: <http://journal.binus.ac.id/index.php/winners/article/view/3815>
- [15] Laksmi VA, Januarti I. ANALISIS PENGUKURAN KINERJA PERUSAHAAN DENGAN METODE BALANCED SCORECARD (Studi Kasus Pada PT. Bank Jateng Cabang Utama Semarang) [Internet]. Diponegoro; 2011. Available from: <https://repofeb.undip.ac.id/id/eprint/5253>
- [16] Aslam W, Ijaz F. A Quantitative Framework for Task Allocation in Distributed Agile Software Development. IEEE Access. 2018;6:15380–90.
- [17] Choraś M, Springer T, Kozik R, López L, Martínez-Fernández S, Ram P, et al. Measuring and Improving Agile Processes in a Small-Size Software Development Company. IEEE Access. 2020;8:78452–66.
- [18] Yang Y, Li X, Ke W, Liu Z. Automated Prototype Generation From Formal Requirements Model. IEEE Trans Reliab [Internet]. 2020 Jun;69(2):632–56. Available from: <https://ieeexplore.ieee.org/document/8822502/>
- [19] Ahmad S, Harris T, Limb E, Kerry S, Victor C, Ekelund U, et al. Evaluation of reliability and validity of the General Practice Physical Activity Questionnaire (GPPAQ) in 60–74 year old primary care patients. BMC Fam Pract [Internet]. 2015;16(1):113. Available from: <https://doi.org/10.1186/s12875-015-0324-8>
- [20] Bica DAB, Silva CAG d. Learning Process of Agile Scrum Methodology With Lego Blocks in Interactive Academic Games: Viewpoint of Students. IEEE Rev Iberoam Tecno del Aprendiz. 2020;15(2):95–104.
- [21] Harvie DP, Agah A. Targeted Scrum: Applying Mission Command to Agile Software Development. IEEE Trans Softw Eng. 2016;42(5):476–89.
- [22] Yadav N, Rajpoot DS, Dhakad SK. LARAVEL: A PHP Framework for E-Commerce Website. In: 2019 Fifth International Conference on Image Information Processing (ICIIP). 2019. p. 503–8.
- [23] Naista D. Codeigniter Vs Laravel: Kasus Membuat Website Pencari Kerja. Yogyakarta: Lokomedia; 2017. 226 p.
- [24] Dawodi M, Hedayati MH, Baktash JA, Erfan AL. Facebook MySQL Performance vs MySQL Performance. In: 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON). 2019. p. 103–9.
- [25] Khan MA, Sadiq M. Analysis of black box software testing techniques: A case study. In: The 2011 International Conference and Workshop on Current Trends in Information Technology (CTIT 11). 2011. p. 1–5.
- [26] Hevner AR, March ST, Park J, Ram S. Essay in Information Design Science systems. Manag Inf Syst. 2004;28(1):75–105.
- [27] Bækgaard L. Conceptual Model of Artifacts for Design Science Research. In: AMCIS. 2015.
- [28] Hadji S, Taufik M, Mulyono S. Implementasi Metode Scrum Pada Pengembangan Aplikasi Delivery Order Berbasis Website (Studi Kasus Pada Rumah Makan Lombok Idjo Semarang). Konf Ilm Mhs Unissula 2. 2019;32–43.
- [29] Blankenship J, Bussa M, Millett S. Managing Agile Projects with Scrum. Pro Agil NET Dev with Scrum. 2011;13–27.
- [30] Setiyani L. Pengujian Sistem Informasi Inventory Pada Perusahaan Distributor Farmasi Menggunakan Metode Black Box Testing. Techno Xplore J Ilmu Komput dan Teknol Inf. 2019;4(1):1–9.