



Artikel

Pengembangan aplikasi *web tes RIASEC online* pada bakat.uam.ac.id menggunakan metode Agile Scrum

Warna Agung Cahyono 

Rekayasa Perangkat Lunak, Universitas Anwar Medika, Sidoarjo, 61263, Indonesia

Abstrak—Dinamika pasar kerja yang terus berubah menuntut instrumen penilaian minat karir yang adaptif dan mudah diakses. Tes RIASEC, merupakan salah satu alat yang telah teruji untuk pemetaan karir, dan digitalisasinya melalui aplikasi web menawarkan banyak keuntungan. Namun, metode pengembangan perangkat lunak linier sering kali terbukti kaku dalam merespons perubahan kebutuhan pengguna yang dinamis. Penelitian ini bertujuan untuk mengembangkan aplikasi web tes RIASEC online di bakat.uam.ac.id dengan mengimplementasikan metode Agile Scrum untuk meningkatkan fleksibilitas dan akselerasi proses pengembangan. Metode ini diterapkan melalui tiga siklus *sprint*. *Product backlog* diprioritaskan menggunakan metode MoSCoW (*Must-have, Should-have, Could-have, Won't-have*) untuk menentukan item yang akan dikerjakan dalam setiap *sprint*. Setiap siklus *sprint* mencakup tahapan *sprint task* (eksekusi harian) dan *sprint review*. Evaluasi menggunakan *sprint retrospective* menunjukkan bahwa pendekatan ini berhasil mengurangi waktu penyelesaian fitur-fitur inti. Durasi *sprint* 1, 2, dan 3 adalah 2 minggu, 3 minggu, dan 3 minggu, dengan total waktu pengerjaan 7 jam lebih cepat dari estimasi awal. Kerangka kerja Scrum memberikan otonomi dan ruang kreativitas bagi tim pengembang dalam menentukan *Sprint Backlog Item* (SBI) pada tahap *sprint task*, namun tetap terkendali oleh *Product Backlog Item* (PBI) dan estimasi waktu yang ditetapkan oleh *product owner*.

Kata kunci—*riasec; scrum; user story; moscow rules; aplikasi web;*

1. Pendahuluan

Era digital modern ditandai oleh pergeseran lanskap pekerjaan yang dipicu beragam faktor, seperti inovasi teknologi, dinamika geopolitik, kebijakan pemerintah, dan perubahan sosial. Fenomena ini menyebabkan disrupsi karir, menciptakan pasar kerja yang semakin kompleks. Dalam konteks ini, instrumen asesmen minat dan bakat menjadi fondasi krusial bagi individu untuk mengambil keputusan karir yang tepat. Tes RIASEC (*Realistic, Investigative, Artistic, Social, Enterprising, Conventional*), yang dikembangkan berdasarkan teori John L. Holland, telah diakui secara luas sebagai kerangka kerja efektif untuk memetakan karakteristik personal dengan lingkungan kerja yang sesuai (Armstrong et al., 2003; Pramudawardani & Adiati, 2024).

Namun, perkembangan kebutuhan pengguna dan transformasi digital menuntut penyempurnaan instrumen ini secara berkelanjutan, yang belum sepenuhnya terakomodasi dalam metode pengembangan konvensional. Platform web menawarkan keunggulan strategis dibandingkan metode konvensional karena dapat mengurangi konsumsi waktu penghitungan para psikolog dan mengurangi kesalahan hasil tes manual (Farid et al., 2021). Dengan demikian platform tersebut menarik untuk diterapkan pada pengembangan aplikasi tes RIASEC, yang berisi kemampuan interaksi beserta riwayat tes, adaptasi item-item metode psikotes, dan pelaporan *real-time*. Meski demikian, pengembangannya memerlukan sinergi multidisipliner antara psikometer, desainer sistem, dan pengembang *backend*.

Terdapat beragam pendekatan dalam proses pengembangan

* Penulis korespondensi.

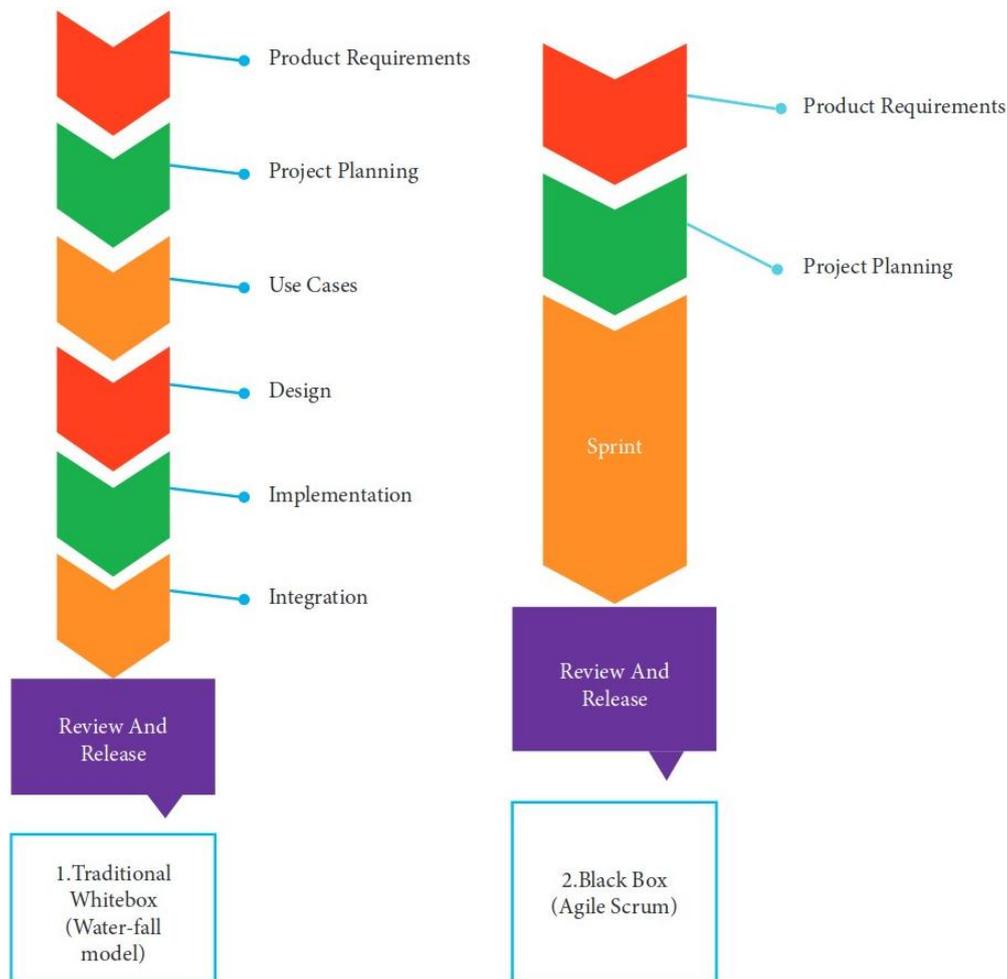
Alamat E-mail: warna.agung@uam.ac.id (W.A. Cahyono)

Email para penulis: WAC (warna.agung@uam.ac.id)

Digital Object Identifier 10.32815/jitika.v19i2.1185

Manuskrip dikirim 24 Juli 2025; direvisi 6 Agustus 2025; diterima 6 Agustus 2025.

ISSN: 2580-8397(O), 0852-730X(P).



Gambar 1. Analogi perbandingan Waterfall dan Agile Scrum

web, yang secara umum dapat diklasifikasikan menjadi dua kategori utama: linear seperti model *Waterfall*, dan adaptif seperti metodologi *Agile*. Pendekatan *Waterfall* cenderung bersifat kaku, terutama dalam konteks pengembangan web tes psikologi, di mana setiap tahapan—mulai dari desain, pengkodean, hingga validasi—dilaksanakan secara terpisah dan tanpa ruang untuk iterasi. Konsekuensinya, perubahan kebutuhan dari pengguna yang muncul di tengah proses pengembangan seringkali baru dapat diakomodasi pada siklus berikutnya (Istiqomah et al., 2024). Hal ini tidak hanya memperpanjang waktu peluncuran produk ke pasar (*time-to-market*), tetapi juga meningkatkan risiko bahwa produk yang dihasilkan sudah tidak relevan saat dirilis. Selain itu, ketidakselarasan antara kebutuhan psikometrik dengan implementasi teknis dapat menghasilkan aplikasi yang tidak valid atau kurang ramah pengguna (*user-friendly*).

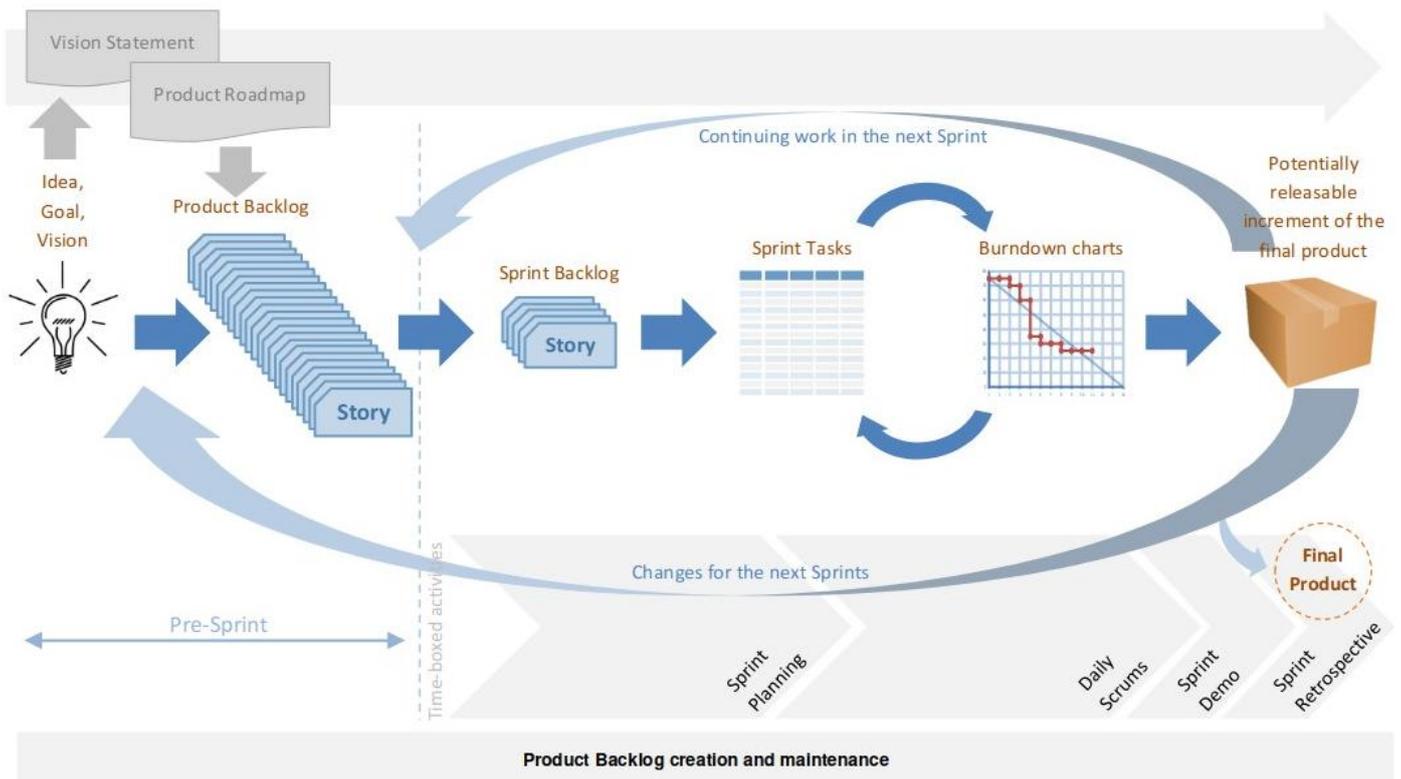
Sebagai alternatif, pendekatan *Agile* seperti *Scrum* menawarkan kerangka kerja yang revolusioner, memadukan fleksibilitas dengan kontrol yang ketat melalui siklus pengembangan yang disebut *sprint*. Setiap *sprint* direalisasikan dalam periode singkat, antara 2 hingga 4 minggu (Nugraha et al., 2023), yang dalam konteks pengembangan web tes RIASEC ini memungkinkan beberapa hal: (1) pengujian fitur spesifik, (2) validasi psikometrik parsial, dan (3) perbaikan pengalaman pengguna (UX) berdasarkan umpan balik langsung. Komunikasi lintas disiplin dalam tim dijamin melalui praktik

seperti *daily stand-up* dan *sprint review*.

Prinsip-prinsip ini berakar pada “*Manifesto for Agile Software Development*,” yang mengutamakan kepuasan pelanggan melalui perilsan perangkat lunak yang bernilai secara rutin dan berkelanjutan, keterbukaan terhadap perubahan kebutuhan bahkan di tahap akhir, serta kolaborasi intensif antara pemangku kepentingan bisnis dan tim pengembang. Metodologi ini juga menekankan pentingnya membangun lingkungan kerja yang mendukung dan memotivasi developer, serta mengutamakan diskusi tatap muka sebagai metode komunikasi yang lebih efektif dibandingkan hanya sepenuhnya mengandalkan dokumentasi (Cohn, 2011).

Dengan mengadopsi *Agile Scrum*, penelitian ini bertujuan untuk (1) membangun sebuah kerangka kerja *Scrum* yang fleksibel untuk pengembangan web tes RIASEC dan (2) mengukur dampaknya terhadap kecepatan pengembangan. Implikasi praktis yang diharapkan adalah reduksi waktu pengembangan dan peningkatan akurasi dalam memenuhi kebutuhan pengguna melalui umpan balik berkelanjutan. Secara teoretis, penelitian ini memperluas penerapan *Scrum* ke domain pengembangan instrumen ilmiah berbasis web.

Keberhasilan proyek ini diharapkan dapat memfasilitasi pelaksanaan tes karier secara digital bagi institusi pendidikan dan membuktikan bahwa pendekatan *Agile* dapat diadopsi untuk aplikasi psikodiagnostik. Fleksibilitas ini menjadi krusial, mengingat karakteristik tes RIASEC yang memerlukan



Gambar 2. Tahapan pelaksanaan scrum

penyesuaian dengan karakter local (Tasrif, 2022), sehingga parameter pertanyaan dan jawaban harus dapat diubah dengan cepat tanpa melalui proses cetak ulang berkas tes.

2. Metode

Metode pengembangan non-*Agile*, seperti *Waterfall*, model spiral, *win-win*, dan *unified process*, seringkali dianggap tidak efisien dari segi waktu (Alshammari, 2022). Pendekatan ini memiliki sifat sekuensial yang kaku, di mana tahapan seperti identifikasi kebutuhan, perencanaan proyek, desain sistem, implementasi, dan integrasi harus diselesaikan secara berurutan. Sebaliknya, metodologi *Agile* menawarkan pendekatan yang lebih adaptif. Tim pengembang tidak perlu menunggu seluruh perencanaan selesai, melainkan dapat fokus mengerjakan tugas-tugas spesifik yang datang secara bertahap dan melakukan evaluasi produk dalam siklus pendek (*sprint*).

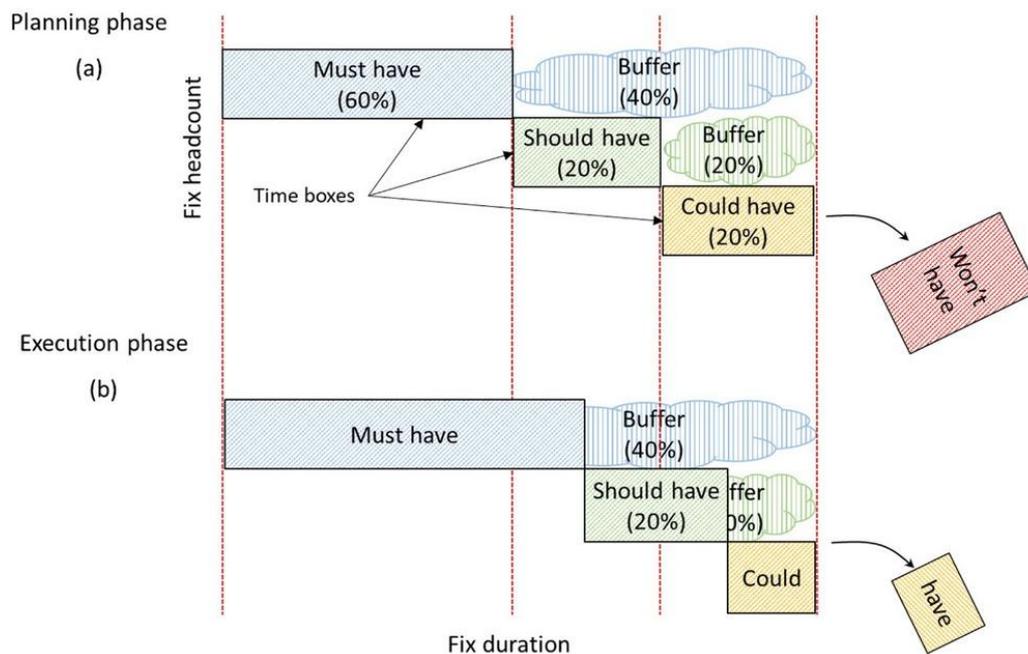
Seperti yang diilustrasikan pada Gambar 1, metode *Agile Scrum* memiliki siklus iterasi yang lebih pendek, mencakup *product requirement*, *product planning* (dalam bentuk *product backlog*), *sprint*, serta *review/release*. *Agile* adalah sebuah pendekatan pengembangan perangkat lunak yang berfokus pada penyampaian nilai kepada pelanggan secara iteratif dan inkremental, serta sangat responsif terhadap perubahan kebutuhan (Suwarno & Jaya, 2022). Intinya, *Agile* menekankan kolaborasi erat dengan pelanggan, pengiriman fitur bernilai tinggi secara berkala, dan kemampuan beradaptasi terhadap perubahan pasar, berbeda dengan pendekatan tradisional yang berpegang teguh pada rencana jangka panjang yang kaku. Tujuannya adalah untuk mengurangi risiko, meningkatkan kualitas produk, dan memastikan solusi yang dikembangkan benar-benar menjawab permasalahan pengguna. Oleh karena

itu, penelitian ini mengadopsi metode *Agile Scrum* dengan model iteratif yang diimplementasikan melalui tiga *sprint*, masing-masing berdurasi 2-3 minggu.

Kerangka kerja *Scrum* itu sendiri dibangun di atas tiga pilar empiris: Transparansi, di mana seluruh aspek proses harus dapat dilihat dan dipahami bersama; Inspeksi, yaitu pemeriksaan kemajuan secara berkala untuk mendeteksi penyimpangan; dan Adaptasi, yang berarti penyesuaian segera dilakukan jika terjadi penyimpangan atau perubahan kebutuhan (Rodriguez et al., 2016). Prinsip utama *Scrum* mencakup kerja dalam iterasi berbatas waktu (*Sprints*), pengelolaan kebutuhan melalui *Product Backlog* yang diprioritaskan oleh *Product Owner*, tim yang kecil, lintas-fungsi, dan mandiri (*self-organizing*), serta peran *Scrum Master* yang memfasilitasi proses dan menghilangkan hambatan bagi tim.

Secara garis besar, tahapan *Scrum* dalam penelitian ini diilustrasikan pada Gambar 2 dan terbagi menjadi tiga bagian utama: tahap pra-*sprint*, tahap *sprint*, dan tahap evaluasi berupa *sprint retrospective* (Rad, 2021). Siklus pengembangan yang berulang terjadi pada fase *sprint*.

Tahap pertama, atau pra-*sprint*, dimulai dengan perumusan *vision statement* dari sudut pandang pengguna dan penyusunan *product roadmap* oleh *Product Owner* sebagai linimasa pengembangan fitur. Setelah itu, dibentuklah *Scrum Team* yang terdiri dari peran multidisiplin, yaitu *Scrum Master*, *Product Owner*, dan *Developer*. Tim kemudian menyusun *Product Backlog*, yang merupakan daftar kebutuhan produk. Kebutuhan ini digali dari pengguna dan dirumuskan ke dalam format *user story* oleh *Product Owner* (Cohn, 2011). *Backlog* untuk aplikasi ini mencakup fitur-fitur utama seperti autentikasi, bank soal, model skoring RIASEC, dasbor hasil, dan modul administrasi. Salah satu keunggulan *Scrum* adalah fleksibilitasnya, yang tidak



Gambar 3. Pola MoSCoW rules dengan should-have sebagai buffer

mengharuskan semua *user story* diselesaikan sebelum melanjutkan ke tahap berikutnya.

Sebelum memasuki tahap *sprint*, perlu dilakukan penentuan prioritas terhadap item-item dalam *Product Backlog*. Penelitian ini menggunakan Metode MoSCoW (*Must-have, Should-have, Could-have, Won't-have*) untuk memprioritaskan item yang akan dimasukkan ke dalam *sprint* berdasarkan analisis risiko dan nilai bisnisnya (Hermans, 2023; Miranda, 2022). Dalam konteks ini, fitur dengan kategori *Must-have* difokuskan pada validitas konten dan keamanan data, baik untuk sisi admin maupun peserta.

Tahap berikutnya adalah eksekusi *sprint*, yang dimulai dengan *sprint planning* untuk menguraikan setiap *product backlog* yang dipilih menjadi tugas-tugas yang lebih kecil. Proses implementasi melibatkan pengembangan berbasis Git dan sinkronisasi harian melalui *daily stand-up* selama 15 menit untuk membahas progres dan hambatan. Di akhir setiap *sprint*, *increment* produk yang dihasilkan akan divalidasi melalui dua mekanisme (Agustini, 2023). Pertama, *Sprint Review* yang melibatkan para pemangku kepentingan (seperti humas institusi dan calon pengguna) untuk menguji fungsionalitas dan *user experience*. Kedua, *Psychometric Sprint Review* yang dilakukan oleh psikometris atau dokter untuk memverifikasi validitas butir soal dan konsistensi hasil tes.

Terakhir, masukan yang diperoleh dari kedua sesi *review* tersebut menjadi bahan utama untuk *Sprint Retrospective*. Dalam forum ini, tim melakukan evaluasi untuk menyempurnakan proses kerja pada *sprint* berikutnya (Setiawan et al., 2024).

Berdasarkan Gambar 3, penerapan aturan MoSCoW memungkinkan tim untuk mengelola risiko waktu pengerjaan. Kategori *Should-have* dapat difungsikan sebagai penyangga (*buffer*) untuk mengantisipasi keterlambatan (*delivery*) pada pengerjaan item *Must-have* yang tidak sesuai dengan prediksi awal. Selanjutnya, seperti yang diilustrasikan pada Gambar 2, fase *sprint* mencakup serangkaian aktivitas yang terstruktur.

Proses ini diawali dengan *sprint planning*, sebuah rapat di mana tim merencanakan pekerjaan yang akan dilakukan dalam satu siklus *sprint* (misalnya, durasi tetap 2 minggu) berdasarkan prioritas yang telah dihitung menggunakan aturan MoSCoW. Dalam rapat ini, *Product Owner* menentukan kebutuhan (*requirement*) esensial yang berasal dari *user story* untuk dimasukkan ke dalam *Sprint Backlog*. Kemudian, *story-story* tersebut dipecah menjadi tugas-tugas (*task*) yang lebih teknis dan terperinci.

Pada akhir *sprint*, tim melakukan *sprint demo* untuk mendemonstrasikan fungsionalitas yang telah berhasil dikembangkan sesuai dengan *user story* yang ditargetkan. Tahap ini juga mencakup *sprint review*, yang menjadi dasar untuk diskusi dalam *sprint retrospective*. Dalam forum retrospektif, tim akan mengevaluasi hasil *review* untuk memutuskan apakah diperlukan improvisasi pada proses kerja di *sprint* berikutnya. Keseluruhan proses pengembangan perangkat lunak ini dipandu dan diarahkan oleh *Scrum Master*, yang memastikan semua pihak yang terlibat menjalankan perannya secara efektif.

3. Hasil

Bagian ini akan memaparkan hasil dan pembahasan dari penerapan metodologi *Scrum* dalam pengembangan aplikasi tes RIASEC yang dapat diakses di <https://bakat.uam.ac.id>. Sesuai dengan alur pada Gambar 2, proses *Agile Scrum* dalam penelitian ini distrukturkan ke dalam tiga fase utama: pra-*sprint*, pelaksanaan *sprint*, dan evaluasi final. Sistematika penelitiannya meliputi:

- a. tahap pra-*sprint* yang mencakup perumusan *vision statement*, penyusunan *product roadmap*, pengembangan *user story*, dan pembuatan *product backlog*;
- b. tahap *sprint* yang terdiri dari perencanaan *sprint backlog*, pengerjaan *sprint task*, serta pelaksanaan *sprint demo* dan *review*; dan

Tabel 1. Roadmap produk

Tahap	Epic	Benefit/Tujuan
2-minggu pertama	Tes online siswa	Siswa dapat melakukan tes, jika koneksi terputus siswa tetap melanjutkan tanpa mengulang dari awal.
2-minggu kedua	Hasil Tes	Siswa dapat melihat hasil tes, sedangkan guru BK juga dapat melihat hasil tes siswa-siswa di sekolahnya.
2-minggu ketiga	Riwayat tes	Siswa dan admin dapat melihat Riwayat tes yang pernah dilakukan.

- c. diakhiri dengan tahap evaluasi melalui *sprint retrospective*, yang datanya diambil dari *sprint task* untuk menyusun *burndown chart*.

3.1. Pre-sprint

Tahap pra-*sprint* merupakan fase persiapan awal dalam proses *Scrum*. Fase ini dimulai dengan perumusan *vision statement*, yang merupakan penjabaran definisi produk dari sudut pandang pengguna. Untuk proyek ini, definisi produk dan *vision statement*-nya adalah sebagai berikut:

- Produk: Aplikasi tes bakat minat RIASEC yang ditujukan untuk siswa jenjang atas.
- *Vision Statement*: Menyediakan platform tes daring yang dapat diakses di mana saja, yang memastikan progres tes tetap tersimpan meskipun koneksi internet terputus. Aplikasi ini bertujuan untuk memberikan gambaran peta karakteristik diri bagi siswa, sekaligus menyajikan gambaran potensi karier di masa depan bagi guru Bimbingan Konseling (BK).

Berlandaskan *vision statement* tersebut, disusunlah sebuah *product roadmap* yang strategi globalnya dirinci pada Tabel 1. Perlu dicatat bahwa linimasa dan fase dalam *roadmap* ini bersifat fleksibel. Penyesuaian dapat terjadi seiring dengan proses penghalusan *Product Backlog* menjadi *user story*, dan linimasanya dapat kembali berubah setelah dilakukan perhitungan rasio nilai (*value*) dan upaya (*effort*) seperti yang akan dijelaskan pada Tabel 4.

3.2. User story

Proses perancangan sistem diawali dengan identifikasi kebutuhan fungsional, yang pada penelitian ini menghasilkan 14 kebutuhan terstruktur. Kebutuhan tersebut dirumuskan ke dalam format *user story* setelah melalui proses observasi dan wawancara dengan lima kelompok pemangku kepentingan (*stakeholder*): psikometris (dokter), humas Universitas Anwar Medika (UAM), tim IT yayasan, kepala Lembaga Pengembangan Teknologi Informasi dan Komunikasi (LPTIK), *developer*, dan siswa.

3.3. Product backlog

User story yang telah terkumpul kemudian digunakan untuk menghaluskan dan membentuk *Product Backlog*. Format *user*

Tabel 2. Daftar *user story*

Kode	Uraian <i>user story</i>
Pengguna dengan peran sebagai siswa:	
US1	Sebagai siswa, saya ingin buat akun supaya bisa <i>login</i> .
US2	Sebagai siswa, saya ingin <i>login</i> supaya bisa masuk sistem.
US3	Sebagai <i>user</i> , saya ingin memulai tes supaya mengetahui karakteristik pribadi.
US4	Sebagai <i>user</i> , saya ingin melihat hasil tes supaya dapat terpetakan kelebihan kekurangan.
US5	Sebagai <i>user</i> , saya ingin melihat riwayat tes supaya bisa cek apakah ada perbedaan karakteristik pada waktu yang berbeda.
US6	Sebagai <i>user</i> , saya ingin <i>logout</i> supaya hp/laptop saya dapat dipakai <i>user</i> yang lain.
US6b	Sebagai <i>user</i> , saya dapat mengklik deskripsi jurusan sehingga saya mendapat pengetahuan tentang jurusan tersebut.
Pengguna dengan peran sebagai admin:	
US7	Sebagai admin, saya ingin <i>login</i> supaya bisa masuk.
US8	Sebagai admin, saya ingin melihat daftar siswa supaya tahu apakah siswa sudah bikin akun.
US9	Sebagai admin, saya ingin melihat riwayat tes per siswa supaya tahu potensi pada siswa tersebut.
US10	Sebagai admin, saya ingin dapat menambahkan data sekolah baru yang belum terdaftar di dalam sistem tes RIASEC <i>online</i> .
US11	Sebagai admin, saya ingin membuat <i>link share</i> riwayat tes pada humas sekolah supaya pihak sekolah dapat melihat laporan tes anak-anak didiknya pada periode tertentu.
US12	Sebagai admin, saya ingin <i>logout</i> supaya setelah pakai tidak ada orang lain yang masuk tanpa <i>login</i> .
US12b	Sebagai seorang admin, saya ingin <i>membroadcast</i> hasil tes RIASEC kepada pengguna tertentu.

story (“Sebagai [peran], saya ingin [tujuan], sehingga [manfaat]”) terbukti sangat efektif untuk menggali kebutuhan pengguna secara detail, meskipun dalam penelitian ini pembahasan *user story card* tidak diperdalam. Kumpulan *user story* pada Tabel 2 kemudian diolah menjadi *Product Backlog Item* (PBI) yang lebih konkret, seperti yang akan dirinci pada Tabel 3.

3.4. Sprint backlog

Setelah daftar PBI tersusun, tahap selanjutnya adalah perencanaan *sprint* untuk membentuk *Sprint Backlog*. Dalam fase ini, dilakukan proses prioritas untuk menentukan PBI mana yang akan dimasukkan ke dalam *sprint* berikutnya. Penentuan prioritas ini didasarkan pada perhitungan skor rasio antara nilai (*value*) dari sudut pandang bisnis dan pengguna dengan estimasi upaya (*effort*) dari sisi tim pengembang, seperti pada Tabel 4.

Setelah setiap PBI diberikan skor awal (seperti pada Tabel 4), langkah selanjutnya adalah melakukan perhitungan rasio antara *Value* dan *Effort* untuk memprioritaskan PBI secara objektif. Dalam konteks ini, *Value* didefinisikan sebagai nilai suatu fitur dari sudut pandang pengguna, sedangkan *Effort* adalah estimasi usaha yang dibutuhkan oleh tim pengembang untuk merealisasikannya (Del Sagrado & Del Águila, 2021). Prinsip

Tabel 3. PBI pada tes RIASEC <https://bakat.uam.ac.id>

Kode	Deksripsi PBI	Keterangan
PBI.1	<i>Login/Register/Logout user</i>	Gabungan dari US1, US2, dan US6.
PBI.2	Tes soal	US3 merupakan fitur untuk tes pilihan ganda ya atau tidak.
PBI.3	Grafik Hasil tes	US4 berupa grafik radar sirkular mengenai dominansi karakteristik diri.
PBI.4	Riwayat tes	US5 berupa daftar/ <i>list</i> riwayat berapa kali melakukan tes serupa pada aplikasi ini(bakat.uam.ac.id).
PBI.5	<i>Login/logout admin</i>	Gabungan dari US7 dan US12.
PBI.6	<i>List siswa dan Riwayat tes siswa</i>	US9 Admin bakat.uam.ac.id yang merupakan humas pihak Universitas Anwar Medika dapat cek siswa tersebut sudah membuat akun atau belum, sekaligus dapat melihat apakah siswa tersebut sudah tes ataukah belum beserta potensi dari siswa tersebut.
PBI.7	Menambah sekolah	US10 jika ada siswa yang tidak dapat registrasi akun karena sekolahnya belum ada, maka admin dapat menambahkan data sekolah yang belum tersedia.
PBI.8	<i>Create link</i> riwayat tes periode tertentu per sekolah	US11 admin dapat <i>generate link</i> khusus pada sekolah yang membutuhkan hasil tes pada rentang periode tertentu.
PBI.9	Deskripsi jurusan hasil tes RIASEC	Setelah <i>user</i> melakukan tes, maka dari halaman hasil terdapat <i>link</i> menuju deskripsi jurusan yang tepat untuk siswa tersebut.
PBI.10	<i>Broadcast</i> hasil via WhatsApp	Admin ingin mengirimkan hasil pada beberapa atau semua hasil tes yang telah dilakukan oleh pengguna.

Tabel 4. Skor sisi *value* bisnis dan sisi *effort development* (1-5)

PBI	Value (V)	Effort (E)	Estimasi (jam)
PBI.1	5	1	12
PBI.2	5	3	36
PBI.3	5	4	48
PBI.4	3	3	36
PBI.5	4	1	12
PBI.6	3	4	48
PBI.7	5	2	12
PBI.8	5	4	48
PBI.9	1	2	24
PBI.10	1	5	60
Total			336

Tabel 5. Pola MoSCoW

Kategori	Deskripsi
M	<i>Must Have</i> Value tinggi vs Effort rendah
S	<i>Should have</i> Value tinggi vs Effort tinggi
C	<i>Could have</i> Value rendah vs Effort rendah
W	<i>Won't have</i> Value rendah vs Effort tinggi

Tabel 6. rasio *value* dan *effort* yang telah diurutkan

PBI	Value (V)	Effort (E)	Rasio	Estimasi (jam)
PBI.1	5	1	5	12
PBI.5	4	1	4	12
PBI.7	5	2	2,5	12
PBI.2	5	3	1,67	36
PBI.3	5	4	1,25	48
PBI.8	5	4	1,25	48
PBI.4	3	3	1	36
PBI.6	3	4	0,75	48
PBI.9	1	2	0,5	24
PBI.10	1	5	0,2	60

dasarnya adalah bahwa prioritas sebuah fitur berbanding lurus dengan nilainya dan berbanding terbalik dengan upaya pengembangannya, seperti pada Tabel 5. Hubungan ini dapat dirumuskan sebagai berikut:

$$rasio_{value_{effort}} = \frac{value}{effort} \tag{1}$$

Dengan menggunakan formula tersebut, rasio untuk setiap PBI dihitung dan hasilnya disajikan pada Tabel 6. Berdasarkan skor rasio ini, setiap *Product Backlog Item* kemudian diklasifikasikan menggunakan aturan MoSCoW (*Must-have, Should-have, Could-have, Won't-have*), seperti pada Tabel 9.

Must-have yaitu fitur yang wajib ada agar sistem dapat berfungsi secara minimal. Tanpa fitur ini, produk dianggap tidak berjalan. Contoh dalam proyek ini adalah fungsionalitas inti dari soal tes RIASEC itu sendiri.

Should-have, merupakan fitur yang sangat penting dan seharusnya ada, namun tidak bersifat kritis. Aplikasi tetap dapat berjalan tanpanya, meskipun fungsionalitasnya menjadi kurang lengkap. Contohnya adalah grafik radar hasil tes yang

memetakan karakteristik pengguna.

Could-have yaitu fitur yang diharapkan ada karena dapat memberikan nilai tambah, tetapi tidak esensial. Contohnya adalah fitur untuk melihat riwayat tes.

Won't-have merupakan fitur yang pada tahap ini diputuskan untuk tidak dikembangkan, meskipun mungkin akan dibutuhkan di masa depan. Contohnya adalah fitur *broadcast* hasil tes via WhatsApp. Fitur ini tidak diprioritaskan karena berdasarkan masukan, guru BK lebih menginginkan siswa menerima hasil tes secara langsung dari mereka, bukan melalui notifikasi otomatis dari sistem.

Klasifikasi inilah yang menjadi dasar untuk merumuskan PBI mana yang akan dimasukkan ke dalam *Sprint Backlog* pada tahap

Tabel 7. MoSCoW rules untuk penggolongan produk

Klasifikasi	Daftar PBI
M	PBI.1, PBI.2, PBI.5, PBI.7
S	PBI.3 dan PBI.8
C	PBI.4, PBI.6, PBI.9
W	PBI.10

Tabel 8. Total estimasi setelah mengeliminasi *won't have*

Daftar PBI	Estimasi	Persentase
M PBI.1, PBI.2, PBI.5, PBI.7	72 jam	26 %
S PBI.3 dan PBI.8	96 jam	35 %
C PBI.4, PBI.6, PBI.9	108 jam	39 %
Total	276 jam	100 %

Tabel 9. Sprint ke-1 durasi 2 minggu estimasi 72 jam

PBI	Item	Estimasi	SBI	Item	Run
PBI.1	Autentikasi peserta (<i>register, login, logout</i>)	12	01	Implementasi autentikasi menggunakan <i>scaffolding</i> Laravel/Breeze.	1
			02	Penyesuaian tata letak halaman registrasi pengguna.	3
			03	Pembuatan halaman <i>landing page</i> dan login untuk peserta.	4
			04	Implementasi fungsionalitas <i>logout</i> peserta.	1
Estimasi		12	Total		9
PBI.5	Autentikasi admin (<i>login & logout dengan role guard</i>) Berbeda dengan peserta, <i>guard role</i> berfungsi untuk mengambil data dari tabel <i>user database</i> yang berbeda	12	05	Konfigurasi <i>guard</i> (<i>auth.php</i>) dan <i>middleware</i> (<i>admin.php</i>) untuk autentikasi peran admin.	5
			06	Pembuatan halaman <i>dashboard</i> untuk admin.	5
			07	Implementasi fungsionalitas <i>logout</i> khusus untuk peran admin.	2
Estimasi		12	Total		12
PBI.7	Manajemen data sekolah	12	08	Pembuatan tabel data untuk mengelola data sekolah se-Indonesia.	7
			09	Implementasi formulir untuk menambah data sekolah baru.	5
Estimasi		12	Total		12
PBI.2	Tes pilihan ganda, soal-soal jenis RIASEC	36	10	Pembuatan halaman panduan pelaksanaan tes.	2
			11	Implementasi penanganan sesi (<i>session handling</i>) untuk menjaga progres tes saat koneksi terputus.	4
			12	Pembuatan model data untuk soal dan kategori RIASEC.	5
			13	Pengembangan API untuk pengacakan soal tes.	6
			14	Desain tata letak antarmuka untuk menampilkan soal.	5
			15	Pengembangan API untuk menyimpan jawaban dan mengambil soal berikutnya (SBI 13).	5
16	Implementasi fungsionalitas untuk menyimpan hasil akhir tes dan mengakhiri sesi.	6			
Estimasi		36	Total		33
Total Sprint 1		72	Total Sprint 1		66

perencanaan *sprint*, seperti pada Tabel 7.

Strategi perencanaan *sprint* dilakukan sesuai dengan aturan MoSCoW (Gambar 3), di mana item berkategori *Must-have* menjadi prioritas utama untuk dikerjakan. Pendekatan ini juga memanfaatkan kategori lainnya sebagai zona penyangga (*buffer*) untuk mitigasi risiko keterlambatan. Secara praktis, item *Should-have* berfungsi sebagai penyangga bagi *Must-have*, dan item *Could-have* menjadi penyangga bagi *Should-have*. Struktur ini dirancang agar jika terjadi penundaan, dampaknya akan terlebih dahulu mengenai fitur-fitur dengan prioritas lebih rendah.

Setelah klasifikasi MoSCoW diterapkan pada kesepuluh PBI, total estimasi waktu pengerjaan disusun ulang dengan mengeliminasi item dari kategori *Won't Have*, sebagaimana dirinci pada Tabel 8. Berdasarkan tabel tersebut, alokasi pengerjaan terdistribusi sebagai berikut: 26% untuk PBI *Must-have* (PBI.1, PBI.2, PBI.5, dan PBI.7) dan 35% untuk PBI *Should-have* yang berfungsi sebagai zona penyangga utama (PBI.3 dan PBI.8). Kategori *Could-have* menjadi penyangga terakhir.

Dengan demikian, jika terjadi keterlambatan dalam proses pengiriman, fitur yang paling mungkin terdampak adalah item *Could-have* yang tidak bersifat kritis, sehingga masih ada ruang untuk negosiasi tanpa mengorbankan fungsionalitas inti produk.

3.5. Sprint task

Setelah prioritas PBI ditetapkan menggunakan MoSCoW, proses pengembangan memasuki tahap eksekusi, yang merupakan inti dari kerangka kerja *Scrum*. Tahap ini berpusat pada pelaksanaan *sprint*, yaitu siklus kerja dengan durasi tetap yang fleksibel, antara 2 hingga 4 minggu, disesuaikan dengan kebutuhan proyek.

Untuk menentukan kapasitas kerja setiap *sprint*, dilakukan perhitungan berdasarkan jam kerja efektif tim. Dengan asumsi, Universitas Anwar Medika memiliki 40 jam kerja per minggu dengan waktu istirahat 1 jam per hari. Sehingga, total efektif kerja yaitu 35 jam per minggu (atau 40 jam kerja dikurangi 5 jam

Tabel 10. Sprint ke-2 durasi 3 minggu estimasi 96 jam

PBI	Item	Estimasi	SBI	Item	Run
PBI.3	Visualisasi grafik hasil tes	48	17	Kalkulasi skor tes per kategori menjadi data statistik (JSON).	20
			18	Pembuatan <i>card</i> label untuk keterangan kategori dan rekomendasi jurusan.	15
			19	Pembuatan grafik radar untuk visualisasi klasifikasi hasil tes.	15
Estimasi		48	Total		50
PBI.8	Fitur berbagi laporan untuk guru BK	48	20	Implementasi filter laporan per sekolah dan rentang periode.	10
			21	Implementasi fungsionalitas untuk memilih data (SBI 20) dan menghasilkan tautan (<i>link</i>) laporan.	17
			22	Pembuatan halaman daftar <i>link</i> yang menampilkan hasil tes siswa (termasuk 3 karakteristik utama dan jurusan yang cocok).	10
			23	Pembuatan halaman detail laporan yang menampilkan penjelasan 6 karakteristik beserta grafik radar.	10
Estimasi		48	Total		47
Total Sprint 2		96	Total Sprint 2		97

Tabel 11. Sprint ke-3 durasi 3 minggu estimasi 108 jam

PBI	Item	Estimasi	SBI	Item	Run
PBI.4	Riwayat tes sisi peserta	36	34	Pembuatan halaman detail riwayat tes (termasuk 3 karakteristik utama dan jurusan).	18
			25	Implementasi grafik radar pada halaman riwayat tes.	17
Estimasi		36	Total		35
PBI.6	Daftar siswa dan riwayat siswa sisi administrator	48	26	Pembuatan <i>datatable</i> untuk daftar siswa dengan <i>paging</i> , dibatasi 500 untuk <i>paging</i> besar dan 10/25/50/100 untuk <i>paging</i> kecil.	14
			27	Implementasi fungsionalitas edit, hapus, sembunyikan, dan nonaktifkan akun siswa.	9
			28	Pembuatan daftar riwayat tes per siswa di <i>dashboard</i> admin.	8
			29	Implementasi <i>card</i> label keterangan dan rekomendasi jurusan di halaman admin.	8
			30	Implementasi grafik radar klasifikasi hasil tes di halaman admin.	10
Estimasi		48	Total		49
PBI.9	Deskripsi jurusan/prodi hasil tes	24	8	Pembuatan model data relasional antara kategori RIASEC dan jurusan.	6
			11	Desain antarmuka pengguna (UI) untuk menampilkan deskripsi jurusan.	10
			5	Implementasi <i>badge hyperlink</i> dari hasil tes menuju halaman deskripsi jurusan.	6
Estimasi		24	Total		22
Total Sprint 3		108	Total Sprint 3		106

istirahat per minggu). Maka, satu *sprint* yang berjalan selama 2 minggu memiliki kapasitas sekitar 70 jam. Dalam durasi ini, PBI yang telah diprioritaskan (dengan mengeliminasi kategori *Won't Have* untuk efisiensi pada Tabel 8) dipecah menjadi unit-unit tugas yang lebih kecil dan dapat dikelola, yang dalam penelitian ini disebut sebagai *Sprint Backlog Item* (SBI).

Setiap penyimpangan antara realisasi pengerjaan *sprint task* dengan estimasi yang telah ditetapkan dalam *Sprint Backlog* akan menjadi bahan evaluasi utama dalam sesi *Sprint Retrospective*. Analisis ini sangat berguna untuk perbaikan proses perencanaan pada siklus *sprint* berikutnya dan proyek di masa mendatang. Berdasarkan prioritas PBI yang telah final, maka disusunlah SBI seperti pada Tabel 9 hingga Tabel 11.

Selanjutnya, *burndown charts* akan ditampilkan di bab pembahasan (Gambar 13).

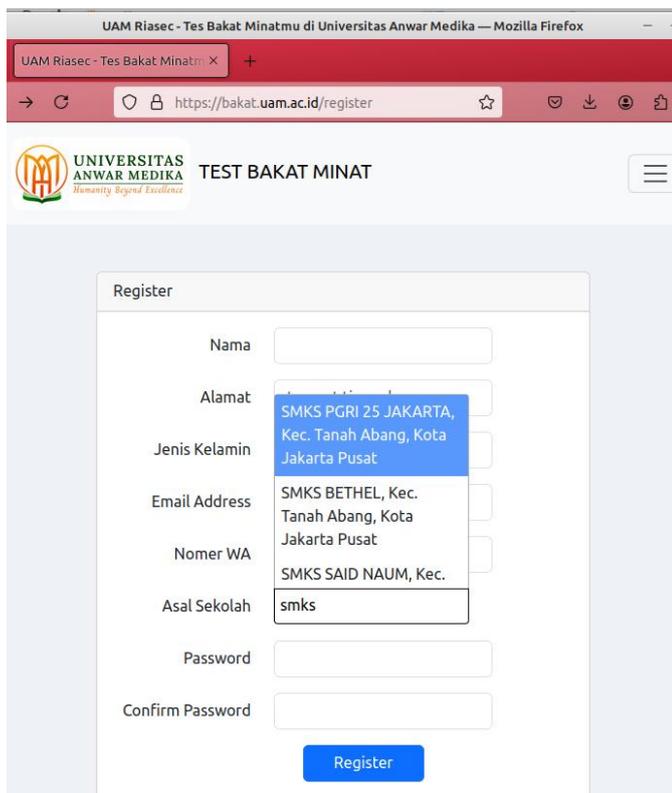
3.6. *Sprint demo*

Tahap *Sprint Review* merupakan sesi evaluasi yang diadakan di akhir setiap *sprint* untuk mendemonstrasikan hasil kerja yang telah diselesaikan oleh tim pengembang (Pratama et al., 2022). Tujuan utama dari fase ini adalah untuk mengidentifikasi *defect* atau ketidaksesuaian fungsionalitas, di mana setiap temuan akan menjadi masukan untuk dicari solusi pada sesi *Sprint Retrospective* (Lee & Chen, 2023).

Hasil dari proses *review* yang dilakukan pada ketiga *sprint*

Tabel 12. Review untuk menemukan defect

PBI	Deskripsi	Hasil Review
PBI.1	Login/Register/Logout user	Revisi minor
PBI.2	Tes soal	diterima
PBI.3	Grafik Hasil tes	Defect minor
PBI.4	Riwayat tes	diterima
PBI.5	Login/logout admin	diterima
PBI.6	List siswa dan Riwayat tes siswa	diterima
PBI.7	Menambah sekolah	diterima
PBI.8	Create link riwayat tes periode tertentu per sekolah	Diterima
PBI.9	Deskripsi jurusan/prodi hasil tes RIASEC	Diterima

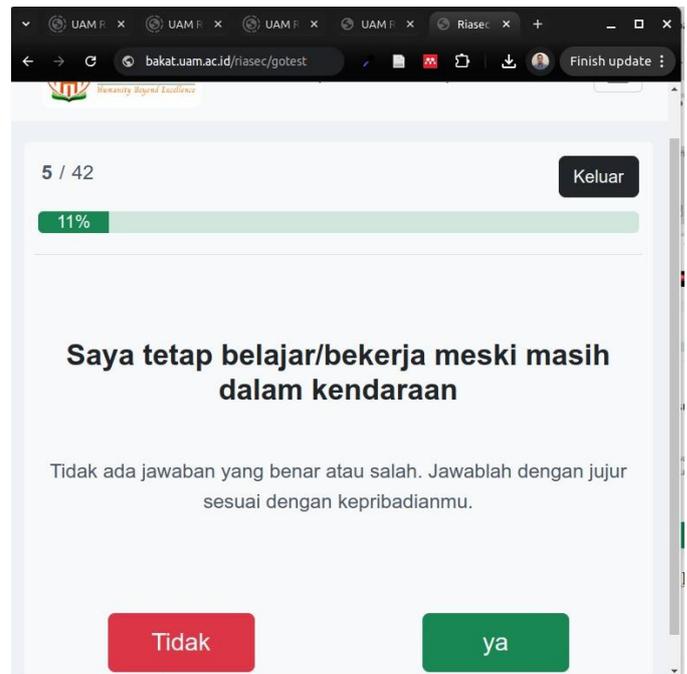


```

96
97 public function authenticate(Request $request)
98
99     $credentials = $request->validate([
100         'email' => 'required|email',
101         'password' => 'required'
102     ]);
103
104     if (Auth::attempt($credentials)) {
105         $request->session()->regenerate();
106         return redirect()->route('dashboard')
107             ->withSuccess('You have successfully logged in!');
108     }
109     if (Auth::guard('admin')->attempt(['username' => $request->input('email'), 'password' =>
110         $request->input('password')], $request->input('remember'))) {
111         return redirect('/admin/dashdash');//redirect()->intended(route('admin.dash'));
112     }
113
114     return back()->withErrors([
115         'email' => 'Your provided credentials do not match in our records.',
116     ])->onlyInput('email');
    
```

Gambar 4. PBI.1 register dan login

dalam penelitian ini dirangkum pada Tabel 12. Salah satu temuan signifikan dari *sprint* pertama adalah terkait proses registrasi pengguna. Ditemukan bahwa jumlah kolom isian yang bersifat wajib terlalu banyak, sehingga berpotensi menurunkan minat pengguna untuk mendaftar. Sebagai solusinya, tim pengembang menyederhanakan formulir pendaftaran dengan



```

96 private function getriasec($id)
97 {
98     $sases = array_keys(Session::get('riasec')[1]);
99     $soal = (new SoalRia)::whereNotIn('id_soal', $sases)->pluck('id_soal');
100     $idx = rand(0, count($soal) - 1);
101     $soal = (new SoalRia)::where("id_soal", count($soal) == 0 ?
102         $sases[count($sases) - 1] : $soal[$idx])->first();
103     return response(json_encode(["code" => 0, "alert" => "Berhasil",
104         "message" => "Get Riasec", "showalert" => false,
105         "data" => json_encode(array("soal" => $soal,
106         "progress" => count($soal) == 0 ? count($sases) : count($sases) + 1,
107         "total" => count($sases) + count($soal)])),
108         201)->header('Content-Type', 'application/json');
109 }
    
```

Gambar 5. PBI.2 tes RIASEC

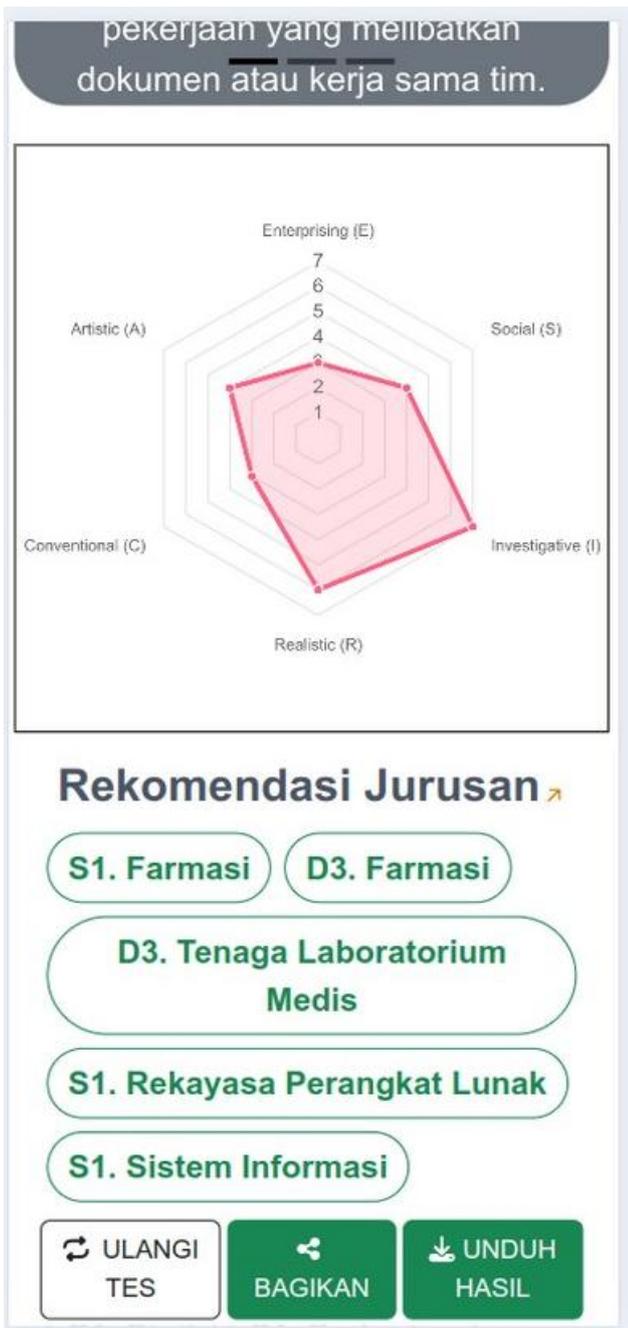
hanya mewajibkan pengisian data esensial, yaitu email, nama, asal sekolah, dan nomor WhatsApp.

Gambar 4 hingga Gambar 12 menampilkan ulasan fungsional dari sembilan PBI yang telah diimplementasikan, beserta beberapa detail teknis yang relevan.

Pada fitur registrasi (Gambar 4), daftar sekolah yang dapat dipilih oleh pengguna dibatasi hanya pada sekolah yang telah didaftarkan. Sementara itu, proses login pengguna memanfaatkan autentikasi *default* dari *framework* Laravel.

Untuk fungsionalitas inti, yaitu pelaksanaan tes, diimplementasikan sebuah mekanisme untuk menjaga integritas data dan kenyamanan pengguna. Sebagaimana ditunjukkan pada Gambar 5, sebuah cuplikan kode pada baris 98 memastikan bahwa progres pengerjaan soal tidak akan hilang jika koneksi internet terputus selama sesi tes masih aktif. Selain itu, metode *whereNotIn* yang diterapkan pada baris 99 menjamin bahwa tidak ada soal yang sama akan ditampilkan berulang kali kepada pengguna.

Tampilan hasil tes, seperti yang diilustrasikan pada Gambar 6, telah disempurnakan berdasarkan masukan dari tahap *review*. Halaman ini menyajikan deskripsi karakteristik psikologis siswa di bagian atas dan visualisasi data berupa grafik radar di bagian bawah. Fitur riwayat tes juga telah berhasil diimplementasikan dan berfungsi dengan baik, seperti yang terlihat pada Gambar 7. Halaman ini menampilkan daftar tes yang pernah dilakukan dan dilengkapi tombol untuk melihat detail hasil, yang terhubung langsung dengan visualisasi grafik pada PBI.3.



Gambar 7. PBI.4 riwayat tes



Gambar 6. PBI.3 grafik hasil tes

```

97 public function authenticate(Request $request)
98 {
99     $credentials = $request->validate([
100         'email' => 'required|email',
101         'password' => 'required'
102     ]);
103
104     if (Auth::attempt($credentials)) {
105         $request->session()->regenerate();
106         return redirect()->route('dashboard')
107             ->withSuccess('You have successfully logged in!');
108     }
109
110     if (Auth::guard('admin')->attempt(['username' => $request->input('email'), 'password' =>
111         $request->input('password')], $request->input('remember'))) {
112         return redirect('/admin/dashdash');//redirect()->intended(route('admin.dash'));
113     }
114
115     return back()->withErrors([
116         'email' => 'Your provided credentials do not match in our records.',
117     ])->onlyInput('email');
118 }
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153 public function logout(Request $request)
154 {
155     // dd("hhhhh");
156     if(Auth::guard('admin')->check()) // this means that the admin was logged in.
157     {
158         Auth::guard('admin')->logout();
159         return redirect()->route('login')
160             ->withSuccess('You have logged out successfully!');
161     }
162     Auth::logout();
163     $request->session()->invalidate();
164     $request->session()->regenerateToken();
165     return redirect()->route('login')
166         ->withSuccess('You have logged out successfully!');
167 }
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
    
```

Gambar 8. PBI.5 login dan logout admin

registered/index

Daftar Pengguna Tes

10 entries per page section 1-500

No	Sekolah	Nama	Phone	Alamat	E-mail	SPP
11	SMAN 1 BOJONEGORO	Fathoni	0936392046	Bojonegoro	fathoniimam1999@gmail.com	100000.00
12	SMAN 1 KRIAN	Firda	087788778787	Krian	gh@gmail.com	0.00
13	SMA Al Islam Islamic Boarding School	Firdha	0877666776676	Krian	g@gmail.com	300000.00
14	SMAS MUHAMMADIYAH 1 TAMAN	gaswul fikri	08958030374	sidoarjo	gofik@gmail.com	0.00
15	SMKS MUHAMMADIYAH 1 TAMAN	Imam Fathoni	085808710503	Bojonegoro	fathoniimam199@gmail.com	0.00
16		rtv	234324		wer@aw	0
17	SMKS MUHAMMADIYAH 1 TAMAN	Rulita	081231769663	Surabaya	nurrazizah385@gmail.com	0.00
18	SMAN 1 SIDOARJO	SATRIA	081234556677	MALANG	wahyuagungsatria20@gmail.com	500.00
19	SMAS ISLAM SAID NA UM JAKARTA	Satria	085393838399	Sidoarjo	wahyu@gmail.com	1.00
20	SMKS MUHAMMADIYAH 1 TAMAN	Vicky Dwi Oktavianto, S.Pd., M.AP	082231436433	Krian	vickyoktavianto30@guru.smk.belajar.id	0.00

Showing 11 to 20 of 22 entries

Alvaro Recoba
 Jl Raya Bypass Krian 30
 alvaro_recoba@gmail.com
 08512345625

SMKS HARVARD JAKARTA
 SPP: Rp 25.000.000,00

Edit

Hapus

enabled

Hide

Riwayat Tes

No	Tgl	Prodi Sesuai Karakter Belajar/Karir
1	September 19, 2024 at 11:03 AM	<p>1. Enterprising</p> <p>SI. Bisnis Digital SI. Kewirausahaan SI. ekonomi SI. Bahasa</p> <p>2. Artistic</p> <p>SI. Bahasa Digital</p> <p>3. Social</p> <p>SI. Fisioterapi SI. Administrasi Kesehatan D3. Tenaga Laboratorium Medis SI. Ilmu sosial SI. psikologi</p>

Gambar 9. PBI.6 daftar siswa beserta riwayat tes

Untuk mengakomodasi kebutuhan administrasi, dikembangkan serangkaian fitur khusus pada sisi admin. Proses autentikasi admin dibuat terpisah dari pengguna biasa dengan menambahkan *guard* khusus pada *framework* Laravel untuk menangani lokasi tabel pengguna yang berbeda, sebagaimana

diilustrasikan pada Gambar 8.

Dari *dashboard* admin, fungsionalitas manajemen pengguna dan data dapat diakses. Admin dapat melihat daftar siswa beserta riwayat tesnya melalui antarmuka yang interaktif (Gambar 9). Selain itu, terdapat pula fitur untuk mengelola data

Daftar Sekolah
+ Add Sekolah

10 entries per page
section -999--500 < >

No	NPSN	Nama Sekolah	Propinsi	Kota
491	70029751	MA AL-GHUROBA	Papua Barat Daya	Kab. Sorong
492	70028107	MA AL-HABIBI	Sumatera Utara	Kab. Deli Serdang
493	70044621	MA AL-HADY	Kalimantan Barat	Kab. Sambas
494	69994698	MA AL-HAFIDHAH	Jawa Timur	Kab. Sumenep
495	69994698	MA AL-HAFIDHAH	Jawa Timur	Kab. Sumenep
496	70027821	MA AL-HAFIDZIYAH	Jawa Timur	Kab. Lumajang
497	70027821	MA AL-HAFIDZIYAH	Jawa Timur	Kab. Lumajang
498	70013778	MA AL-HANNAN DDI SEPPANGE	Sulawesi Selatan	Kab. Bone
499	70033547	MA AL-HAROKAH DARUNNAJAH 12	Riau	Kota Dumai
500	69993783	MA AL-HAROMAIN	Jawa Timur	Kab. Jember

Showing 491 to 500 of 500 entries

<< < 1 ... 46 47 48 49 50 > >>

☰

Nama

Jenjang

NPSN

Propinsi

Kota

SAVE

Gambar 10. PBI.7 daftar sekolah dan formulir pengelolaan daftar sekolah

sekolah yang akan ditampilkan pada halaman registrasi (Gambar 10).

Salah satu fitur kunci adalah kemampuan untuk menghasilkan tautan (*link*) laporan yang dapat dibagikan kepada guru BK. Seperti yang ditunjukkan pada Gambar 11, admin dapat membuat URL unik yang memberikan akses kepada pihak sekolah untuk melihat rekapitulasi minat bakat siswa-siswanya. Untuk melengkapi hasil tes, aplikasi juga menyediakan halaman deskripsi untuk setiap jurusan yang direkomendasikan. Halaman ini dapat diakses melalui *hyperlink* yang tersedia di bawah grafik radar hasil tes (Gambar 12), memberikan informasi mendalam bagi pengguna untuk eksplorasi karier lebih lanjut.

4. Pembahasan

Tahap *Sprint Retrospective* bertujuan untuk mengevaluasi keseluruhan proses *Scrum* yang telah dijalankan. Sesi ini

berfokus pada analisis hasil *review* dari setiap *sprint* dengan tujuan untuk meningkatkan kinerja tim di masa mendatang. Salah satu alat evaluasi utama yang digunakan adalah *sprint burndown chart*, yang membandingkan estimasi waktu pengerjaan dengan realisasinya.

Secara keseluruhan, data menunjukkan bahwa total estimasi waktu pengerjaan untuk PBI yang diselesaikan adalah 276 jam. Namun, realisasi pengerjaan dari *sprint* 1 hingga 3 tercatat hanya 269 jam, atau 7 jam lebih cepat dari perkiraan awal, seperti yang terlihat pada Tabel 9 hingga Tabel 11. Meskipun demikian, analisis lebih dalam pada *burndown chart* (Gambar 13) menunjukkan adanya dinamika kecepatan yang berbeda di setiap *sprint*.

Pada *sprint* pertama, tim pengembang bekerja 6 jam lebih cepat dari yang direncanakan. Perbedaan ini terutama disebabkan oleh estimasi waktu yang kurang akurat pada PBI.1 (fitur *login*, *logout*, dan registrasi). Temuan ini menjadi pelajaran berharga untuk proyek serupa di masa depan, di mana estimasi

Daftar Hasil Per Sekolah

[New Link](#)

No	Sekolah	Tanggal Mulai	Tanggal Akhir	Detail
1	SMKS MUHAMMADIYAH 5 JAKARTA	01 Januari 2024	30 September 2024	Share Link klik saya
2	SMKS MUHAMMADIYAH 5 JAKARTA	02 Januari 2024	30 September 2024	Share Link klik saya
3	SMKS DHARMA AGUNG	30 Januari 2025	19 Februari 2025	Share Link klik saya

Tambah Link Sekolah

range tanggal: 05 / 01 / 2023 - 07 / 14 / 2025

10 entries per page section 1-500 [Save Link](#)

No	Tanggal Tes	Karakteristik Bakat	Prodi Yang Cocok
Nama & Tab; : Agung Cahyono., S.Kom., MT & Tab; , -			
12	2024-08-02 14:33:35	Conventional	SI. Administrasi Kesehatan, SI. Sistem Informasi, SI. Hukum, SI. Akuntansi
13	2024-08-02 14:32:20	Enterprising, Realistic, Social	SI. Bisnis Digital, SI. Kewirausahaan, SI. ekonomi, SI. Bahasa, SI. Farmasi, D3. Farmasi, D3. Tenaga Laboratorium Medis, SI. Rekayasa Perangkat Lunak, SI. Sistem Informasi, SI. Administrasi Kesehatan, SI. Sipil, SI. Pertanian, SI. Kedokteran, SI. Informatika, SI. Elektro, SI. Mesin, SI. Fisioterapi, SI. Ilmu sosial, SI. psikologi
Nama & Tab; : dewi & Tab; , -			
1	2025-04-12 06:36:22	Conventional, Realistic	SI. Administrasi Kesehatan, SI. Sistem Informasi, SI. Hukum, SI. Akuntansi, SI. Farmasi, D3. Farmasi, D3. Tenaga Laboratorium Medis, SI. Rekayasa Perangkat Lunak, SI. Sipil, SI. Pertanian, SI. Kedokteran, SI. Informatika, SI. Elektro, SI. Mesin
Nama & Tab; : qqq & Tab; , qqq			
2	2024-09-26 09:42:11	Social, Realistic, Investigative, Conventional	SI. Kewirausahaan, SI. Fisioterapi, SI. Administrasi Kesehatan, D3. Tenaga Laboratorium Medis, SI. Bisnis Digital, SI. Ilmu sosial, SI. psikologi, SI. Farmasi, D3. Farmasi, SI. Rekayasa Perangkat Lunak, SI. Sistem Informasi, SI. Sipil, SI. Pertanian, SI. Kedokteran, SI. Informatika, SI. Elektro, SI. Mesin, SI. Hukum, SI. Akuntansi
3	2024-09-26 09:41:07	Investigative, Enterprising	SI. Farmasi, D3. Farmasi, D3. Tenaga Laboratorium Medis, SI. Administrasi Kesehatan, SI. Rekayasa Perangkat Lunak, SI. Sistem Informasi, SI. Bisnis Digital, SI. Kewirausahaan, SI. ekonomi, SI. Bahasa

Riwayat Tes

No	Tgl	Prodi Sesuai Karakter Belajar/Karir
1	Agung Cahyono., S.Kom., MT 2 Agustus 2024, 14:33	1. Conventional SI. Administrasi Kesehatan SI. Sistem Informasi SI. Hukum SI. Akuntansi
2	Imam Fatotni 19 September 2024, 06:34	1. Enterprising SI. Bisnis Digital SI. Kewirausahaan SI. ekonomi SI. Bahasa 2. Realistic SI. Farmasi D3. Farmasi D3. Tenaga Laboratorium Medis SI. Rekayasa Perangkat Lunak SI. Sistem Informasi SI. Administrasi Kesehatan SI. Sipil SI. Pertanian SI. Kedokteran SI. Informatika SI. Elektro SI. Mesin 3. Social SI. Fisioterapi SI. Ilmu sosial SI. psikologi

Gambar 11. PBI.8 share link untuk guru BK

menyukai bidang pekerjaan yang melibatkan dokumen atau kerja sama tim.

Rekomendasi Jurusan

- S1. Farmasi
- D3. Farmasi
- D3. Tenaga Laboratorium Medis
- S1. Rekayasa Perangkat Lunak
- S1. Sistem Informasi
- S1. Administrasi Kesehatan
- S1. Sipil

ULANGI TES BAGIKAN UNDUH HASIL

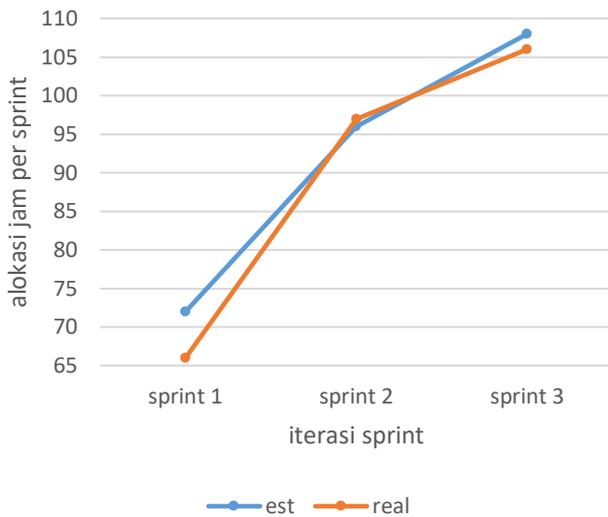
UNIVERSITAS ANWAR MEDIKA
Remaining Beyond Excellence

Tes RIASEC (Tes Minat Bakat)

Teknik Sipil

salah satu cabang ilmu teknik yang berfokus pada perencanaan, perancangan, konstruksi, pemeliharaan, dan pengelolaan infrastruktur serta lingkungan buatan manusia. Jurusan ini mempelajari bagaimana merancang dan membangun berbagai struktur yang mendukung kehidupan modern, seperti jalan, jembatan, gedung, bendungan, bandara, sistem transportasi, jaringan air bersih, dan sistem pengelolaan limbah.

Gambar 12. PBI.9 deskripsi jurusan/prodi sesuai kategori RIASEC



Gambar 13. Grafik burndown keseluruhan sprint

waktu untuk PBI.1 dapat diturunkan menjadi 9 jam dan PBI.2 menjadi 33 jam.

Sebaliknya, pada *sprint* kedua, *burndown chart* (Gambar 13) mengindikasikan bahwa tim pengembang bekerja lebih lambat dari perencanaan. Hal ini dapat disebabkan oleh ketidakakuratan dalam estimasi skor *effort* pada tahap sebelumnya. Secara spesifik, pengerjaan PBI.3 memakan waktu 2 jam lebih lama dari perkiraan, meskipun PBI.8 berhasil diselesaikan 1 jam lebih cepat. Akibatnya, terjadi keterlambatan bersih selama 1 jam pada *sprint* ini. Temuan ini menjadi bahan evaluasi untuk proyek dengan tingkat kesulitan serupa di masa mendatang. Lebih lanjut, analisis pada sesi *daily scrum* mengidentifikasi bahwa keterlambatan utama terjadi pada SBI.17 (kalkulasi nilai tes), yang realisasi pengerjaannya mencapai 15 jam. Penyebabnya adalah tingkat kesulitan teknis pada algoritma perhitungan skor dan misinterpretasi kebutuhan—terutama terkait aturan bisnis mengenai jumlah maksimal karakter dominan yang harus ditampilkan dan bagaimana sistem harus merespons jika terdapat lebih dari tiga karakter. Insiden ini menyoroti bahwa meskipun telah dibantu oleh *user story*, tingkat kedetailan penerjemahan kebutuhan pengguna menjadi PBI oleh *Product Owner* masih dapat ditingkatkan.

Pada *sprint* ketiga, grafik realisasi kembali berada di bawah estimasi, yang mengindikasikan tidak adanya keterlambatan. Kinerja efisien ini salah satunya disebabkan oleh penyelesaian perbaikan *error* dari PBI.1 dan PBI.3 yang ditemukan pada *review sprint* sebelumnya. Dengan mengejar target tersebut, tim berhasil mengkompensasi keterlambatan minor yang terjadi pada *sprint* kedua.

5. Kesimpulan

Berdasarkan hasil penelitian, metode *Agile Scrum* terbukti efektif dalam memandu tim selama proses pengembangan perangkat lunak aplikasi tes RIASEC. Kerangka kerja ini memungkinkan *Product Owner* untuk merinci *Product Backlog Item* (PBI) secara efektif menggunakan *user story* dan memprioritaskannya melalui metode MoSCoW. Selanjutnya, pada tahap *sprint task*, tim pengembang dapat fokus

menguraikan PBI tersebut menjadi tugas-tugas teknis yang lebih kecil (*Sprint Backlog Item* atau SBI).

Salah satu keunggulan utama yang teridentifikasi adalah bagaimana *Scrum* memberikan ruang otonomi kepada pengembang melalui SBI, yang terbukti mendorong munculnya ide-ide kreatif. Meskipun demikian, otonomi ini tetap berada dalam kerangka kerja yang terkendali, karena seluruh rangkaian proses terpandu oleh PBI yang telah ditetapkan. Hal ini menciptakan lingkungan di mana pengembang merasa bebas untuk mengatur pekerjaan teknis mereka, namun tetap selaras dengan tujuan besar proyek.

Penggunaan metode MoSCoW juga berhasil memitigasi risiko keterlambatan. Menariknya, *buffer zone* yang secara teoretis dialokasikan pada kategori *should-have* tidak perlu dimanfaatkan, karena tim pengembang mampu menyelesaikan item-item *must-have* lebih cepat dari estimasi awal. Dengan demikian, proyek ini dapat disimpulkan berhasil, yang dibuktikan dengan selesainya pengembangan 7 jam lebih cepat dari total waktu yang diestimasi, sebagaimana ditunjukkan pada Gambar 13.

Ketersediaan data

Semua data yang dihasilkan atau dianalisis selama penelitian tersedia dalam artikel ini.

Deklarasi konflik kepentingan

Para penulis menyatakan tidak mempunyai kepentingan finansial yang bersaing atau koneksi pribadi yang dapat dianggap mempengaruhi penelitian yang disajikan dalam artikel ini.

Kontribusi Penulis

Tulisan ini berdasarkan proyek yang dikerjakan oleh peneliti. Sehingga penulis yang telah melaksanakan pengembangan proyek berbasis Agile. Berlanjut pada pelaksanaan penulisan artikel ini.

Ucapan Terima Kasih

Penulis ingin menyampaikan ucapan terima kasih kepada Anwar Medika Corporate dokter Nungky Taniyasi MARS, Martina Kurnia Rohmah selaku rektor Universitas Anwar Medika, dan Muhammad Shodikin.

Daftar rujukan

- Agustini, F. (2023). Implementasi Metode Scrum Pada Aplikasi Penjualan Peta Dan Buku (Studi Kasus Pada CV Ubo Rampe Palwoko). *Artikel Ilmiah Sistem Informasi Akuntansi*, 3(1), 36–41. <https://doi.org/10.31294/akasia.v3i1.1900>
- Alshammari, F. H. (2022). Analytical Evaluation of SOA and SCRUM Business Process Management Approaches for IoT-Based Services Development. *Scientific Programming*, 2022, 1–14. <https://doi.org/10.1155/2022/3556809>
- Armstrong, P. I., Hubert, L., & Rounds, J. (2003). Circular unidimensional scaling: A new look at group differences in interest structure. *Journal of Counseling Psychology*, 50(3), 297–308. <https://doi.org/10.1037/0022-0167.50.3.297>
- Cohn, Mike. (2011). *Succeeding with agile : software development using Scrum*. Addison-Wesley.
- Del Sagrado, J., & Del Águila, I. M. (2021). Assisted requirements

- selection by clustering. *Requirements Engineering*, 26(2), 167–184. <https://doi.org/10.1007/s00766-020-00341-1>
- Farid, M. T., Adi Nugroho, H., & Hidayah, I. (2021). Sistem Rekomendasi Minat Anak Menggunakan Alat Tes Rothwell Miller Interest Blank Berbasis Web. *Jurnal Ilmiah ILKOMINFO - Ilmu Komputer & Informatika*, 4(1). <https://doi.org/10.47324/ilkominfo.v4i1.116>
- Hermans, K. (2023). *Mastering Agile: A Comprehensive Guide to Learn Agile*. Cybellium Ltd.
- Istiqomah, D. A., Vikky Aprelia Windarni, Surya Tri Atmaja Ramadhani, & Zein Nur Ichsan. (2024). Peningkatan Pengetahuan Agile Scrum dalam Pengembangan Perangkat Lunak pada Startup Kala Kreatif Indonesia. *JURPIKAT (Jurnal Pengabdian Kepada Masyarakat)*, 5(3), 915–924. <https://doi.org/10.37339/jurpikat.v5i3.1777>
- Lee, W.-T., & Chen, C.-H. (2023). Agile Software Development and Reuse Approach with Scrum and Software Product Line Engineering. *Electronics*, 12(15), 3291. <https://doi.org/10.3390/electronics12153291>
- Miranda, E. (2022). *Moscow Rules: A Quantitative Exposé* (pp. 19–34). https://doi.org/10.1007/978-3-031-08169-9_2
- Nugraha, D., Nur, I. L., Hidayatuloh, M. T., Laluma, R. H., & Gunawa (2023). Implementasi Sistem Informasi Manajemen Kantor Menggunakan Scrum Framework Di Desa Wangunsari. *Jurnal Ilmiah Media Sisfo*, 17(1), 116–124. <https://doi.org/10.33998/mediasisfo.2023.17.1.740>
- Pramudawardani, N. K. P., & Adiati, R. P. (2024). Holland's RIASEC Model: Career and Job Development Assessment for Personnel Administration Staff. *Psikostudia : Jurnal Psikologi*, 13(3), 361. <https://doi.org/10.30872/psikostudia.v13i3.14915>
- Pratama, S., Ibrahim, S., & Reybaharsyah, M. A. (2022). Jurnal Penggunaan Metode Scrum Dalam Membentuk Sistem Informasi Penyimpanan Gudang Berbasis Web. *INTECH*, 3(1), 27–35. <https://doi.org/10.54895/intech.v3i1.1192>
- Rad, N. K. . (2021). *Agile Scrum Foundation Courseware English: Based on the 3rd Edition of the Agile Scrum Handbook*. Van Haren Publishing.
- Rodriguez, G., Soria, A., & Campo, M. (2016). Measuring the Impact of Agile Coaching on Students' Performance. *IEEE Transactions on Education*, 59(3), 202–209. <https://doi.org/10.1109/TE.2015.2506624>
- Setiawan, R., Mulyani, A., Ramdan, G. M., & Roji, F. F. (2024). Aplikasi Resep Makanan Bergizi Membantu Pencegahan Stunting Menggunakan Metodologi Agile Framework Scrum. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 11(4), 761–770. <https://doi.org/10.25126/jtiik.1148384>
- Suwarno, S., & Jaya, W. S. (2022). Design and Development of Software Project Management System using Scrum. *Journal of Informatics and Telecommunication Engineering*, 5(2), 483–493. <https://doi.org/10.31289/jite.v5i2.6412>
- Tasrif, E. (2022). RIASEC Holland's reliability and validity on personality of informatics engineering education students in higher education. *JPPI (Jurnal Penelitian Pendidikan Indonesia)*, 8(1), 11. <https://doi.org/10.29210/020221602>



Warna Agung Cahyono adalah seorang dosen pada prodi rekayasa perangkat lunak pada Universitas Anwar Medika. Saat ini sedang aktif dalam kegiatan pengembangan aplikasi terutama berbasis web. Bidang penelitian pengembangan software terutama web.