

Pengembangan Aplikasi Mobile untuk Sistem Penjualan Produk Smart Farming di Rooftop Farming Center Telkom University Surabaya Menggunakan Metode Design Thinking dan Scrum

1st Teguh Sambada Abriel Karisma
Rekayasa Perangkat Lunak
Universitas Telkom
Surabaya, Indonesia

abrielkarisma@student.telkomuniversity.ac.id

2nd Dewi Rahmawati
Rekayasa Perangkat Lunak
Universitas Telkom
Surabaya, Indonesia

dewirahmawati@telkomuniversity.ac.id

3rd Achmad Muzakki
Rekayasa Perangkat Lunak
Universitas Telkom
Surabaya, Indonesia

achmadmuzakki@telkomuniversity.ac.id

Abstrak — *Rooftop Farming Center* (RFC) di Universitas Telkom Surabaya, sebuah inisiatif ketahanan pangan perkotaan, menghadapi inefisiensi operasional akibat proses penjualan yang manual dan dari orang ke orang, yang membatasi jangkauan pasar serta mempersulit manajemen stok. Untuk mengatasi masalah ini, dikembangkan "RFC Store," sebuah aplikasi *mobile commerce* berbasis *Android*. Laporan ini merinci pengembangan aplikasi menggunakan metodologi gabungan yaitu *DT@Scrum*, yang mengintegrasikan *Design Thinking* untuk penemuan masalah yang berpusat pada pengguna dengan *Scrum* untuk implementasi yang tangkas. Sistem ini dibangun menggunakan beberapa teknologi, termasuk *Flutter* untuk *frontend*, *Node.js* untuk *backend*, dan terintegrasi dengan *payment gateway* *Midtrans*. Evaluasi triangulasi yang ketat dilakukan, mencakup pengujian *black-box*, *System Usability Scale* (*SUS*), dan *User Acceptance Testing* (*UAT*). Hasilnya menunjukkan kualitas dan kesiapan yang tinggi: seluruh 120 kasus uji fungsional berhasil, aplikasi mencapai skor *SUS* "Excellent" sebesar 81,55, dan semua 23 skenario *UAT* berhasil diselesaikan oleh pengguna akhir. Penelitian ini menyajikan model tervalidasi untuk digitalisasi penjualan dalam pertanian perkotaan berbasis komunitas dan memberikan studi kasus praktis tentang penerapan kerangka kerja *DT@Scrum* yang berhasil.

Kata kunci — *Design Thinking*, *Scrum*, *DT@Scrum*, Pertanian Cerdas

I. PENDAHULUAN

Indonesia menghadapi tantangan signifikan di sektor pertanian dan peternakan, terutama di kota-kota besar di mana populasi terus bertambah dan terjadi alih fungsi lahan yang masif. Urbanisasi telah menggeser praktik pertanian konvensional, sehingga mendorong munculnya solusi alternatif seperti pertanian modern untuk mendukung ketahanan pangan perkotaan. Salah satu inisiatif yang menjawab tantangan ini adalah *Rooftop Farming Center* (RFC) di Universitas Telkom Surabaya, yang memanfaatkan atap gedung sebagai lahan pertanian dan peternakan berbasis teknologi *Internet of Things* (*IoT*) untuk sistem pengairan dan

pemantauan. Inisiatif ini tidak hanya menjadi solusi atas keterbatasan lahan, tetapi juga berfungsi sebagai sarana edukasi bagi generasi muda mengenai smart farming dan praktik pertanian berkelanjutan.

Meskipun RFC telah mengadopsi teknologi canggih di sisi produksi, proses bisnis penjualannya masih menghadapi kendala fundamental. Penjualan hasil panen dilakukan secara konvensional dari orang ke orang, sistem pembayaran masih manual, dan tidak ada sistem terintegrasi untuk manajemen stok dan pemantauan penjualan. Keterbatasan ini secara langsung menghambat jangkauan pasar dan efisiensi operasional. Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk menghasilkan aplikasi *mobile* untuk sistem penjualan *smart farming* di RFC Universitas Telkom Surabaya, menerapkan kombinasi metode *Design Thinking* dan *Scrum* untuk mendukung proses pengembangan, dan melakukan evaluasi komprehensif terhadap aplikasi menggunakan *System Usability Scale* (*SUS*), *black-box testing*, dan *User Acceptance Testing* (*UAT*) untuk mengukur kepuasan pengguna, fungsionalitas, dan penerimaan oleh pengguna akhir.

II. KAJIAN TEORI

A. Sistem Penjualan

Sistem penjualan modern mencakup rangkaian aktivitas terintegrasi untuk mengelola stok, transaksi, dan layanan pelanggan. Dengan dukungan teknologi digital (misalnya aplikasi *e-commerce*), sistem penjualan dapat menjangkau pasar lebih luas, mengotomatisasi pencatatan transaksi dan analisis data penjualan secara *real-time*. Hal ini tidak hanya meningkatkan efisiensi operasional penjual (misalnya pemantauan stok otomatis), tetapi juga meningkatkan kepuasan pelanggan melalui kemudahan akses dan kecepatan layanan[1].

B. Design Thinking

Design Thinking adalah pendekatan pemecahan masalah yang berpusat pada pengguna (*human-centered*)[2], dengan

tujuan menghasilkan solusi inovatif dan relevan. Pendekatan ini menggabungkan elemen desain dan strategi bisnis untuk menciptakan solusi yang sesuai kebutuhan spesifik pengguna. *Design Thinking* terdiri dari lima tahapan iteratif: *empathize* (memahami masalah pengguna melalui wawancara/observasi), *define* (merumuskan permasalahan utama), *ideate* (menghasilkan berbagai ide kreatif), *prototype* (membuat model solusi awal), dan *test* (menguji coba prototipe kepada pengguna untuk mendapatkan umpan balik). Metode ini membantu identifikasi kekurangan dan memperbaiki desain sebelum implementasi akhir[2].

C. Scrum

Scrum adalah kerangka kerja *Agile* pengembangan perangkat lunak yang iteratif dan inkremental[3]. *Scrum* membagi kerja ke dalam siklus *sprint* berdurasi pendek (biasanya 1-4 minggu) dengan tujuan menghasilkan produk fungsional pada akhir setiap *sprint*. Terdapat tiga peran utama dalam *Scrum*: *Product Owner* (bertanggung jawab visi produk dan prioritas tugas), *Scrum Master* (memastikan kerangka kerja *Scrum* dijalankan dengan benar), dan *Development Team* (tim pengembang fitur). Proses *Scrum* melibatkan kegiatan perencanaan *sprint*, *daily Scrum*, *sprint review*, dan *retrospective* untuk menjaga kolaborasi tim dan adaptasi terhadap perubahan. Pendekatan *Scrum* yang iteratif memungkinkan tim mengadaptasi perubahan kebutuhan pengguna dengan cepat, sedangkan integrasi prinsip *Agile* memastikan pengiriman perangkat lunak berkualitas secara cepat dan berkelanjutan[4].

D. DT@Scrum

Kombinasi *design thinking* dan *scrum* (*DT@Scrum*) merupakan pendekatan efektif dalam pengembangan perangkat lunak yang berorientasi pada inovasi dan kebutuhan pengguna. *Design thinking* fokus pada eksplorasi masalah dan penciptaan solusi berbasis prototipe, sementara *scrum* menekankan efisiensi dalam iterasi pengembangan. *DT@Scrum* mencakup tiga mode: eksplorasi masalah dan prototipe awal (*Design Thinking Mode*), pengembangan dan pengujian prototipe resolusi tinggi (*Initial Development Mode*), serta implementasi penuh *scrum* (*Fully Integrated Mode*)[5]. Meskipun mampu menghasilkan solusi inovatif dan adaptif, pendekatan ini juga menghadapi tantangan koordinasi tim dan keterbatasan sumber daya. Secara keseluruhan, *DT@Scrum* menjadi kerangka kerja yang cocok untuk pengembangan produk dalam lingkungan yang dinamis[6].

III. METODE

Penelitian ini menggunakan metode pengembangan aplikasi dengan pendekatan *DT@Scrum*, yaitu menggabungkan tahapan *Design Thinking* dalam kerangka kerja *Scrum*. Proyek dilakukan oleh tim pengembang yang memulai fase *design thinking* (*Sprint 1*) dengan menggalang kebutuhan pengguna melalui observasi langsung dan wawancara dengan pengelola RFC (fase *empathize*). Data kualitatif dari observasi/wawancara selanjutnya dianalisis untuk merumuskan kebutuhan (fase *define*) dan menghasilkan ide desain awal. Hasil awal berupa user stories dan prototipe *wireframe low-fidelity* dikaji bersama pengguna melalui diskusi kelompok (FGD). Selanjutnya dilanjutkan *initial development phase* (*Sprint 2*), yaitu

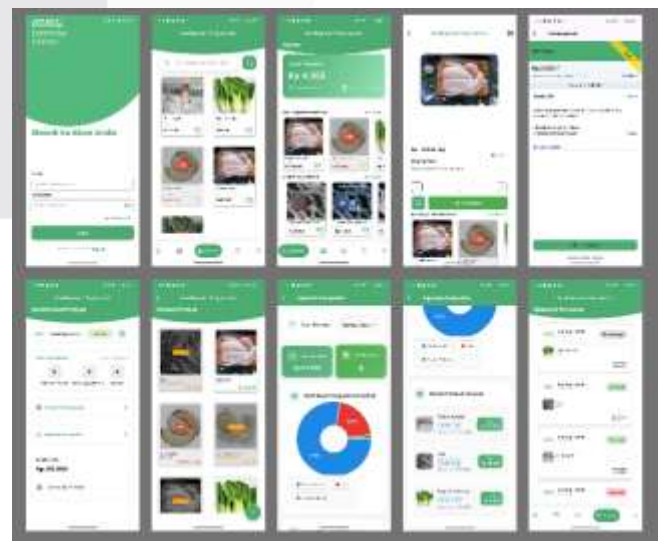
pengembangan prototipe antarmuka *high-fidelity* dan pengujian awal menggunakan *System Usability Scale* (*SUS*) untuk mendapatkan umpan balik desain. Fase terakhir adalah *fully integrated phase* (*Sprint 3-5*), di mana fungsi sistem diimplementasikan secara bertahap dengan *Scrum* untuk mengakomodasi perubahan. Setiap *sprint* mencakup penentuan *backlog* item, pengkodean fitur, dan pengujian secara berkala[5],[6]. Sumber data meliputi catatan wawancara pengguna, studi literatur (teknologi mobile, *framework Flutter*, *Node.js*, integrasi *Midtrans*), serta hasil pengujian aplikasi.



GAMBAR 1
(ARSITEKTUR SISTEM RFC STORE)

Aplikasi *RFC Store* dirancang dengan arsitektur seperti yang ditunjukkan pada Gambar 1. *Flutter* dipilih sebagai *framework mobile* untuk antarmuka, sedangkan *Node.js* digunakan untuk logika *backend* dan integrasi basis data. Aplikasi juga terintegrasi dengan layanan *payment gateway Midtrans* untuk mendukung transaksi pembayaran digital. Kemudian terdapat layanan lain seperti *fonnte* yaitu layanan *whatsapp gateway* untuk *OTP* dan *cloudinary* sebagai layanan *cloud* penyimpanan gambar. Untuk verifikasi sistem, dilakukan *black-box testing* pada seluruh fungsionalitas utama sesuai skenario pengujian, serta *User Acceptance Testing* (*UAT*) melibatkan pengguna akhir untuk validasi fungsionalitas. Pengujian *System Usability Scale* (*SUS*) digunakan untuk mengukur kepuasan dan kemudahan penggunaan antarmuka aplikasi. Semua tahapan di atas dilaksanakan secara iteratif dan terdokumentasi sebagai bagian dari siklus *DT@Scrum*.

IV. HASIL DAN PEMBAHASAN



GAMBAR 2
(ANTARMUKA APLIKASI RFC STORE)

Hasil dari metodologi di atas adalah sebuah aplikasi *mobile e-commerce* pertanian yang disesuaikan dengan kebutuhan *Rooftop Farming Center*. Aplikasi ini memiliki dua modul utama sesuai peran pengguna, yakni modul *User* (Pembeli) dan modul *Penjual*. Pada sisi pengguna akhir (pembeli), tersedia fitur katalog produk yang menampilkan daftar produk pertanian dan peternakan yang dijual, lengkap dengan foto, deskripsi, harga, dan informasi stok secara *real-time*. Pengguna dapat menambahkan item ke keranjang belanja, mengatur jumlah pembelian, kemudian melakukan *checkout*. Saat *checkout*, pengguna memasukkan detail pengiriman dan memilih metode pembayaran elektronik yang tersedia. Aplikasi akan memproses pesanan dan terhubung ke *gateway Midtrans* untuk menyelesaikan pembayaran. Proses pembayaran berlangsung aman melalui antarmuka *Midtrans* yang tertanam di aplikasi, mendukung berbagai metode (transfer bank, *e-wallet*, dll.). Setelah pembayaran terverifikasi sukses, status pesanan pengguna akan otomatis diperbarui menjadi “dibayar” dan pengguna mendapatkan konfirmasi (notifikasi) bahwa order akan diproses oleh penjual. Di sisi penjual, aplikasi menyediakan fitur manajemen produk yang memungkinkan menambah produk baru, mengubah detail produk (misalnya saat ada panen baru atau perubahan harga), serta memperbarui jumlah stok seiring terjadinya penjualan atau produksi. Tersedia pula fitur untuk memperbarui status pesanan (misal: dari “diproses” menjadi “siap dikirim” atau “selesai”) sehingga pembeli dapat melacak progres order mereka. Seluruh histori transaksi tersimpan dan dapat diakses sebagai laporan penjualan, membantu pengelola menganalisis penjualan per periode.

Hasil Pengujian Fungsional (*Black-box*): Pengujian *black-box* atas aplikasi menunjukkan seluruh fitur utama berfungsi sesuai ekspektasi dan bebas kesalahan fatal. Setiap skenario uji yang dirancang berhasil dijalankan dengan output yang benar.. Dengan hasil ini, dapat disimpulkan aplikasi telah lulus uji fungsional, artinya siap digunakan dari perspektif teknis. Ringkasan uji *blackbox* bisa dilihat pada Tabel 1

TABEL 1
(RINGKASAN *BLACKBOX*)

<i>Sprint</i>	Jumlah Kasus Uji	Hasil (<i>Passed</i>)	Tingkat Keberhasilan
<i>Sprint 3</i>	45	45	100%
<i>Sprint 4</i>	53	53	100%
<i>Sprint 5</i>	22	22	100%
Total	120	120	100%

Hasil *User Acceptance Testing* melibatkan 3 orang pengguna perwakilan: 1 mahasiswa sebagai pembeli, 1 pengelola *Rooftop* sebagai penjual, dan 1 staf admin. Ketiganya mencoba aplikasi dalam skenario nyata selama beberapa hari. Secara umum, pengguna menyatakan puas dengan fungsi-fungsi aplikasi karena mempermudah proses yang sebelumnya manual. Beberapa masukan *minor* dari pengguna turut dicatat selama *UAT*, masukan ini sangat berharga sebagai pertimbangan pengembangan lanjutan. Namun, secara keseluruhan tidak ada keluhan mengenai fungsi utama – semua skenario uji user berhasil dijalankan. Para pengguna menyepakati aplikasi ini layak dioperasikan dalam lingkungan *Rooftop Farming Center*. Dengan

demikian, aplikasi dinyatakan lulus *UAT* tanpa perlu perubahan. Hasil SUS Akhir dilakukan dengan menyebarkan kuesioner SUS kepada 10 setelah mereka mencoba aplikasi. Didapat skor SUS rata-rata 81.55. Menurut skala penilaian standar SUS, nilai tersebut berada di atas rata-rata industri (68) yang berarti usability aplikasi tergolong baik dan pengguna merasa cukup nyaman menggunakan aplikasi. Tidak ada pernyataan pada kuesioner yang mendapat skor sangat rendah; mayoritas responden memberi penilaian positif terutama pada aspek kemudahan belajar (*learnability*) dan integrasi fitur. Artinya, pendekatan perancangan yang fokus pada pengguna sejak awal memang berkontribusi pada tingginya tingkat *usability*.

TABEL 2
(HASIL SUS)

Q 1	Q 2	Q 3	Q 4	Q 5	Q 6	Q 7	Q 8	Q 9	Q 10	Jumlah	Nilai (Jumlah x 2.5)
3	4	3	4	3	4	4	3	3	3	35	87.5
3	4	3	3	3	3	4	3	2	2	30	75
3	4	3	4	3	3	4	3	3	2	32	80
3	4	3	4	3	4	3	3	4	3	34	85
3	4	3	4	3	4	3	3	4	2	33	82.5
3	4	3	3	3	4	3	3	3	3	32	80
3	4	3	4	3	4	3	3	3	2	32	80
4	4	3	4	3	4	3	3	3	2	33	82.5
3	4	3	4	3	4	3	3	3	2	32	80
3	4	3	4	3	4	3	3	2	3	32	80
Skor rata rata											81.5

V. KESIMPULAN

Pengembangan aplikasi *mobile* untuk sistem penjualan Aplikasi *mobile* yang dikembangkan dalam penelitian ini berhasil menyediakan sistem penjualan produk smart farming yang terintegrasi bagi *Rooftop Farming Center* Tel-U Surabaya. Semua fitur utama – mulai dari manajemen produk hingga pembayaran digital – telah berjalan sesuai spesifikasi dengan hasil pengujian fungsional 100% lulus. *UAT* melibatkan pengguna akhir juga menunjukkan keberhasilan penuh pada setiap skenario uji. Pengukuran usability (SUS) menempatkan aplikasi pada kategori “Excellent” dengan skor 81,55, mengindikasikan antarmuka yang mudah digunakan. Metodologi gabungan Design Thinking dan Scrum (DT@Scrum) terbukti mendukung pengembangan yang adaptif dan user-centric, sehingga aplikasi yang dihasilkan sesuai dengan kebutuhan pengguna. Dengan demikian, aplikasi ini diharapkan dapat mendigitalisasi proses penjualan produk pertanian dan peternakan di RFC Surabaya, memperluas jangkauan pasar, mempermudah pemantauan stok, dan memberikan fondasi bagi pengembangan sistem yang lebih holistik di masa depan.

LAMPIRAN

- [1] Silmi Nurul Utami and Serafica Gischa, “Penjualan: Pengertian Para Ahli, Bentuk, Jenis, dan Tujuannya,” <https://www.kompas.com/skola/read/2021/08/27/131616269/penjualan-pengertian-para-ahli-bentuk-jenis-dan-tujuannya>.
- [2] Michael. Lewrick, Patrick. Link, L. J. . Leifer, and Nadia. Langensand, *The design thinking playbook : mindful digital transformation of teams, products,*

services, businesses and ecosystems. John Wiley & Sons, Inc., 2018.

- [3] H. Noprisson, "Implementasi Metodologi Agile Software Development pada Proyek Perangkat Lunak," vol. 5, no. 2, 2023, [Online]. Available: <https://jurnal.ikhafi.or.id/index.php/jusibi>

- [4] K. Schwaber and J. Sutherland, "The Scrum Guide The Definitive Guide to Scrum: The Rules of the Game," 2020.

- [5] F. Häger, T. Kowark, J. Krüger, C. Vetterli, F. Übernickel, and M. Uflacker, "DT@Scrum: Integrating Design Thinking with Software Development Processes," 2015, pp. 263–289. doi: 10.1007/978-3-319-06823-7_14.
- [6] C. Vetterli *et al.*, "Jumpstarting Scrum with Design Thinking," 2013.

