

PEMBANGUNAN APLIKASI E-COMMERCE PERTANIAN DUTATANI STORE BERBASIS ANDROID

Development of Android-Based Agriculture E-Commerce Application Dutatani Store

Yoel Tanujaya¹, Rosa Delima², Antonius Rachmat Chrismanto³, dan Argo Wibowo⁴

^{1,2,3} Program Studi Informatika Universitas Kristen Duta Wacana Yogyakarta, Indonesia

⁴Program Studi Sistem Informasi Universitas Kristen Duta Wacana Yogyakarta, Indonesia

Jalan Doktor Wahidin Sudirohusodo 5-25, Yogyakarta, Indonesia

Telp. (0274) 563929, Fax. (0274) 513235

E-mail : yoeltan77@gmail.com

(Makalah diterima, 11 Agustus 2020 – Disetujui, 07 Juni 2021)

ABSTRAK

E-commerce di Indonesia terus meningkat dari tahun ke tahun, baik jumlah transaksi dan pengguna maupun jumlah aplikasi *e-commerce*, masing-masing memiliki sasaran market produk yang berbeda. Namun sektor pertanian di Indonesia belum merasakan dampak *e-commerce*. Hal ini disebabkan oleh rendahnya pilihan aplikasi *e-commerce* pertanian dan antarmuka sistem belum memenuhi tingkat kepuasan. Oleh karena itu, melalui penelitian ini dibangun aplikasi *e-commerce* berbasis android yang diberi nama *Dutatani Store*. Pembangunan aplikasi *Dutatani Store* menggunakan *framework* antarmuka Flutter, dan bahasa pemrograman Dart. Pengembangan aplikasi menggunakan metode *Rapid Application Development* (RAD), dalam kurun waktu dua bulan dengan tim beranggotakan empat orang. Evaluasi sistem dilakukan melalui uji usabilitas terhadap aplikasi untuk mengetahui tingkat kebergunaan aplikasi. Hasil pengujian menunjukkan setiap responden dapat menyelesaikan keseluruhan task skenario. Tampilan *list view* memiliki nilai usabilitas yang lebih tinggi dengan tingkat kesuksesan yang lebih tinggi pula dibandingkan dengan *card view*.

Kata kunci: pertanian, *e-commerce*, aplikasi, *list view*, *card view*

ABSTRACT

E-commerce usage in Indonesia is increasing over years, both in terms of number of transaction, number of users, and as well as number of *e-commerce* platform, each of which targeting different product market. However, agriculture sector still have not been impacted by it. This is due to small number of agricultural *e-commerce* platform, and few of that platform offers unsatisfactory level of user interface. Therefore, in this research we develop a new agricultural *e-commerce* called *Dutatani Store*, using Flutter as a user interface framework, and Dart programming language. Rapid Application Development (RAD) approach is applied in this development within two months period by 4 members team. System evaluation is done through usability testing to determine usability level of the application. Result shows that every respondents were successfully finished all five task scenarios that given to them. Also, *list view* achieved higher usability level with more success level compared to *card view*.

Key words: agriculture, *e-commerce*, application, *list view*, *card view*

PENDAHULUAN

Aplikasi *e-commerce* di Indonesia semakin banyak sehingga menciptakan persaingan yang semakin ketat dalam mengajak pengguna menggunakan aplikasi *e-commerce* tertentu. Toko daring atau *e-commerce* semakin populer dalam 10 tahun terakhir. Jumlah *merchant* tumbuh 17% dalam kurun waktu 2006-2016 dengan total 26,2 juta (Badan Pusat Statistik, 2016). Jumlah transaksi juga mengalami peningkatan eksponensial sebesar 151% pada tahun 2018, dengan total nilai transaksi Rp 77,8 triliun.

Salah satu sektor yang dapat memanfaatkan teknologi *e-commerce* saat ini adalah pertanian. Dalam kurun waktu 2003-2013, Indonesia mengalami penurunan jumlah usaha jenis pertanian. Jenis usaha tani yang mendominasi di Indonesia adalah usaha rumah tangga dengan jumlah usaha 26,14 juta rumah tangga, menurun 16,32% dari hasil Sensus Pertanian 2003 yang tercatat 31,23 juta rumah tangga. Jumlah perusahaan pertanian berbadan hukum 4.165 perusahaan dan jenis usaha pertanian lainnya 5.922 unit.

Tim dari Universitas Kristen Duta Wacana menyusun *blueprint* sistem informasi pertanian yang terintegrasi, meliputi *web mapping* (Chrismanto, Delima, Wibowo, Santoso, & Kristiawan, 2019), *e-learning* pertanian, *forum*, dan *commerce* (Delima, Santoso, & Purwadi, Architecture vision for Indonesian Integrated Agriculture Information Systems using TOGAF framework, 2016). Penelitian sebelumnya telah menghasilkan sistem *e-commerce* pertanian yang dijalankan pada *web desktop* (Delima *et al.*, 2018).

Sebagai pengembangan lanjutan, penelitian ini akan membangun *e-commerce* pertanian untuk perangkat bergerak berbasis android dan iOS bernama Dutatani Store. Aplikasi ini akan mengimplementasi dua tampilan produk, yaitu *card view* dan *list view*. Kedua tampilan tersebut diadopsi oleh banyak aplikasi *e-commerce* di seluruh dunia, masing-masing memiliki karakteristik tampilan yang berbeda, baik ukuran, informasi yang ditampilkan, posisi gambar maupun teks.

Proses pembangunan aplikasi mengadopsi metode *Rapid Application Development* (RAD). Metode ini efektif mempercepat proses pengembangan sistem, dapat dilakukan dengan jumlah tim yang sedikit, dan hasil akhir dapat dirilis lebih cepat. Tim pengembang beranggotakan empat orang termasuk *product owner*, dan *product owner* dapat meminta penambahan atau pengurangan atau pergantian fitur aplikasi secara langsung kepada tim pengembang pada tahap *coding*. Metode penelitian menjelaskan proses pembangunan aplikasi dimulai dari tahap perencanaan, pembangunan purwarupa fitur dan antarmuka, serta tahap akhir yaitu pengujian usabilitas.

Detail aplikasi akhir dijelaskan pada bagian hasil dan analisis, termasuk hasil uji usabilitas.

BAHAN DAN METODE

Requirement Planning

Kegiatan ini dimulai dari perencanaan fitur dan tampilan aplikasi, kemudian dilakukan diskusi antara tim pengembang, *product owner*, dan beberapa *stakeholder* dari kelompok tani. Tim pengembangan aplikasi yang terlibat empat orang, dengan tugas masing-masing sebagai project manager, *front-end developer*, dan *back-end developer*. Metode aplikasi yang digunakan adalah *Rapid Application Development* (RAD). Tahapan dari RAD adalah *planning*, *design* (mencakup pembuatan *prototype*, penambahan fitur, dan evaluasi), dan *testing* (pada penelitian ini menggunakan *usability testing*).

Terdapat dua fitur utama dalam keseluruhan proses penggunaan aplikasi, yaitu:

- Fitur Pembeli
 - o Pencarian produk
 - o Filter produk berdasarkan status (*ready*, *pre-order*, kosong)
 - o Pencarian toko
 - o Detail produk
 - o Kategori produk
 - o *Register*
- Fitur Admin
 - o Login
 - o Lupa *password*
 - o *Logout*
 - o Pencarian produk
 - o Pengaturan produk (tambah, edit, hapus produk)
 - o Ubah profil
 - o Laporan pemesanan
 - o Tambah pesanan

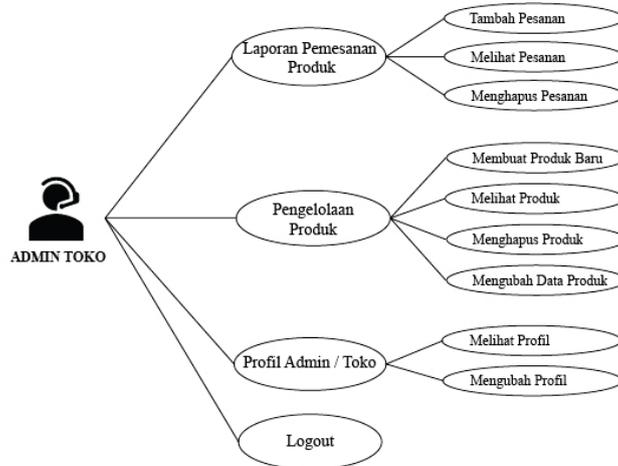
Diagram *use-case* tiap *role* beserta *sitemap* halaman aplikasi untuk setiap fitur dijelaskan pada Gambar 1 hingga Gambar 4.

Untuk basis data, Dutatani Store menggunakan MySQL yang diakses dengan REST API berbahasa pemrograman php. Terdapat 11 tabel yang digunakan, yaitu tabel produk (*master_produk_tani*), user (*master_user*), detail user (*master_detail_user*), pemesanan (*trans_pemesanan_commerce*), toko (*master_kel_tani*), satuan produk (*master_satuan_saprotan*), kategori produk (*master_kategori_alat_bahan*), provinsi, kabupaten, kecamatan, dan kelurahan_*desa*.

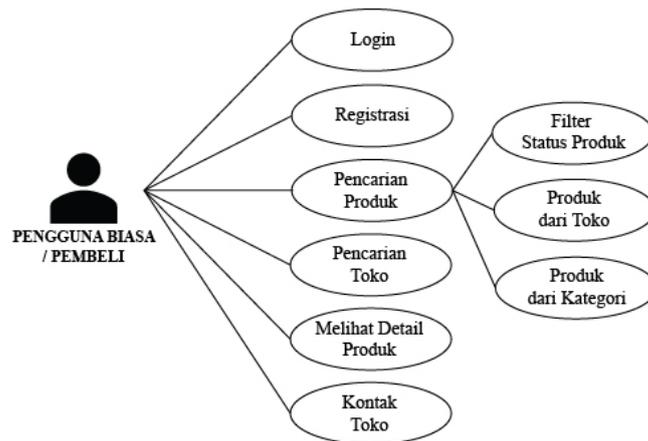
Gambar 5 menjelaskan relasi antara *master_user* dan *master_detail_user* yang merupakan relasi mandatory 1:1 (*one-to-one*), dimana setiap user harus memiliki detail informasi ketika melakukan login atau registrasi.

Setiap user bisa memiliki atau tidak memiliki akun toko, sehingga relasi antara master_detail_user dengan master_kel_tani merupakan relasi optional 0:1 (zero-to-one).

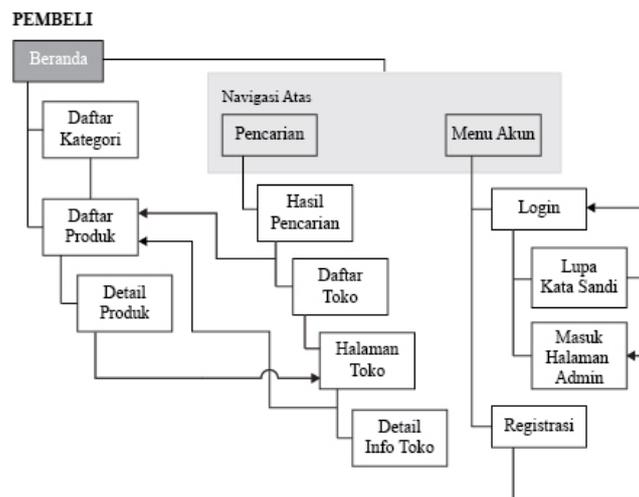
Produk yang dijual pada aplikasi Dutatani pasti memiliki toko, sehingga relasi antara master_produk_tani dengan master_kel_tani bersifat mandatory 1:N (one-to-many).



Gambar 1. Use-case diagram penjual

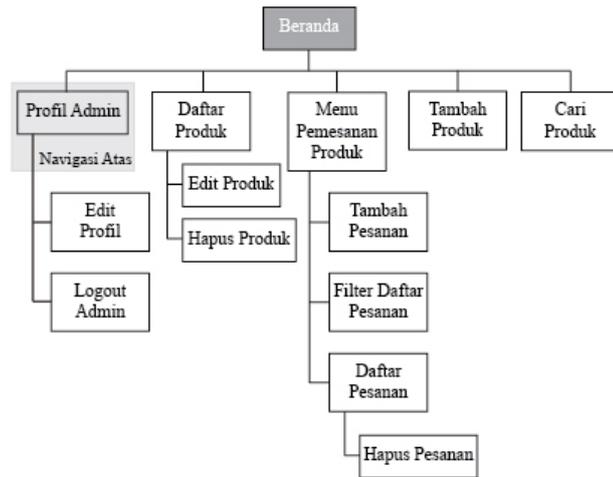


Gambar 2. Use-case diagram pembeli

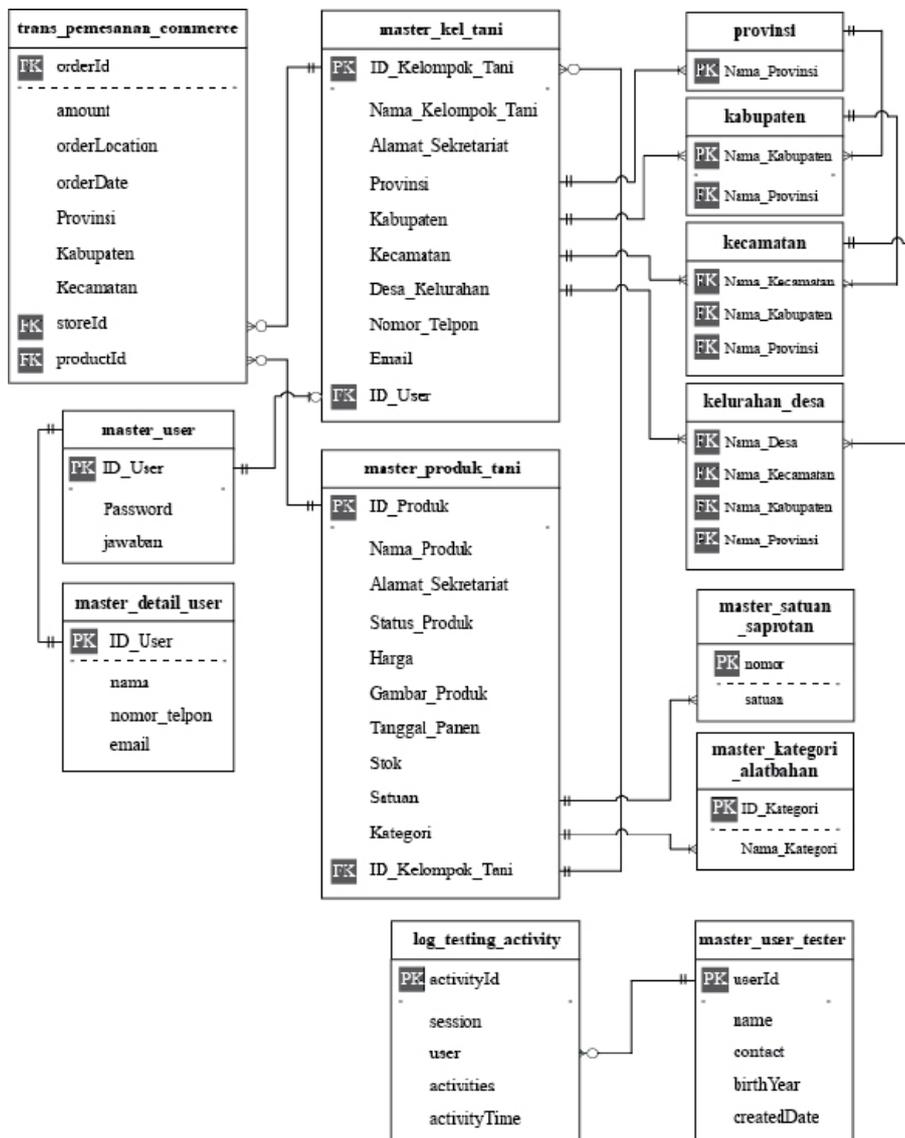


Gambar 3. Sitemap halaman pembeli

PENJUAL / ADMIN



Gambar 4. Sitemap halaman penjual



Gambar 5. Skema basis data Dutatani Store

Iterative Prototyping

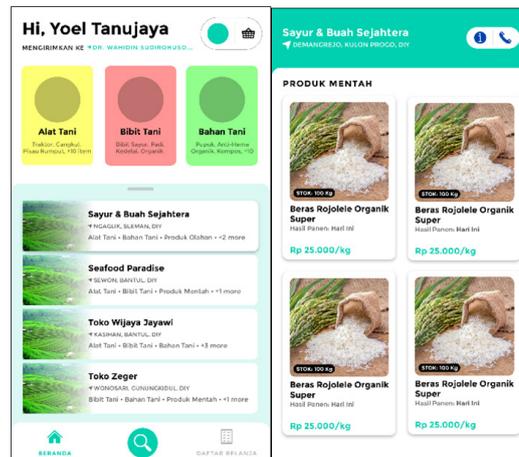
Pembangunan *prototype* fitur dan tampilan antarmuka aplikasi dilakukan melalui proses iterasi desain. Pada setiap proses iterasi, user dan tim pengembang melakukan evaluasi untuk membenahi bagian yang tidak sesuai dengan rancangan awal, atau terdapat perubahan fitur di tengah proses pengembangan aplikasi. Iterasi desain serta fitur dijelaskan pada Tabel 3. Setelah melalui evaluasi secara berkelanjutan dari user, stakeholder, dan tim pengembang aplikasi, fitur aplikasi pada iterasi terakhir adalah:

- Fitur Pembeli
 - o Pencarian produk
 - o Filter berdasarkan status produk: ready, pre-order, kosong
 - o Pencarian toko
 - o Ubah tampilan hasil pencarian produk atau toko (*card view* dan *list view*)
 - o Ubah tampilan rekomendasi produk (*card view* dan *list view*)
 - o Detail produk
 - o Kategori produk
 - o Register
- Fitur Penjual
 - o Login

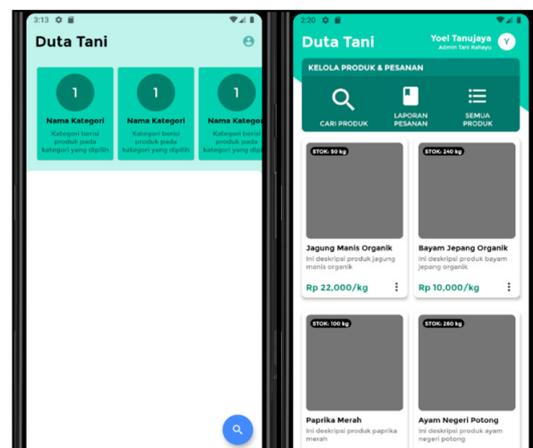
- o Lupa password
- o Logout
- o Pencarian produk
- o Ubah tampilan produk (*card view* dan *list view*)
- o Edit produk
- o Hapus produk
- o Tambah produk
- o Ubah profil
- o Laporan pemesanan (*filter* berdasarkan produk per daerah, urutkan berdasarkan waktu pesanan atau produk terlaris)
- o Tambah pesanan
- o Hapus pesanan

Pengembangan *prototype* fitur dan antarmuka dilakukan secara iteratif, sesuai dengan kebutuhan user serta saran dan masukan dari *stakeholder*, beserta anggota tim pengembang. Gambar 6 adalah iterasi awal antarmuka aplikasi berbentuk *sketch* ilustrasi, dan belum diimplementasikan dalam kode program.

Gambar 7 adalah iterasi *prototype* pertama dalam bentuk kode program yang dapat dijalankan pada perangkat. Data produk belum tersambung dengan database Dutatani, menu pencarian masih belum dapat digunakan, data produk dan login pengguna melalui database lokal, dan belum dapat menyimpan sesi login pengguna.



Gambar 6. Sket/ilustrasi antarmuka awal



Gambar 7. Iterasi prototipe pertama

Pada iterasi selanjutnya, halaman beranda dapat menampilkan produk dengan tipe tampilan *list view*, menu pencarian muncul dalam bentuk modal window, dengan halaman tersendiri untuk menampilkan hasil pencarian dengan tipe tampilan *list view*. Pada fase ini, halaman pengelolaan akun hanya memiliki fitur login pengguna yang ada pada sistem *database*. Antarmuka pada fase ini dapat dilihat pada Gambar 8.

Fase keempat, tampilan *list view* mendapat beberapa perubahan desain dan penempatan informasi produk. Menu pencarian produk berpindah ke bagian atas aplikasi, dan tambahan tombol untuk mengakses menu akun. Fase ini mengalami perubahan yang cukup signifikan dengan adanya tambahan fitur untuk registrasi pengguna baru, dan fitur pengaturan ulang kata sandi pengguna, yang memerlukan beberapa koneksi dengan tabel *database* baru dan fitur validasi tambahan untuk proses registrasi serta validasi akun melakukan perubahan kata sandi. Selain itu, fase ini memberikan akses penjual yang login untuk dapat menambahkan produk baru. Gambar 9 menampilkan desain antarmuka pada fase ini.

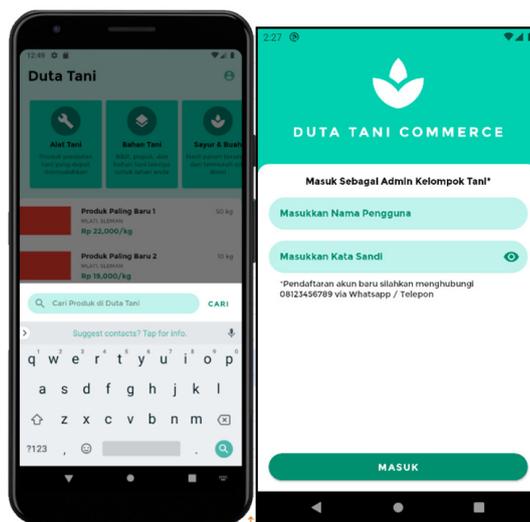
Fase selanjutnya adalah iterasi terakhir sebelum implementasi final aplikasi *Dutatani Store*. Terdapat

perubahan pada menu pencarian yang berubah bentuk menjadi search bar di bagian atas, menu akun berpindah ke bagian kiri, serta fitur untuk mengubah tampilan dari tipe *list view* menjadi *card view* dan sebaliknya. Posisi dan komponen search bar sama antara halaman pembeli dengan halaman penjual untuk menjaga konsistensi tampilan. Untuk item produk, terdapat beberapa informasi tambahan yang dihilangkan dari item untuk menyederhanakan tampilan, dan desain daftar kategori yang telah diperbaharui dengan daftar kategori produk yang sudah final, kontribusi dari masukan *product owner* serta *stakeholder*. Gambar 10 menampilkan antarmuka untuk fase ini.

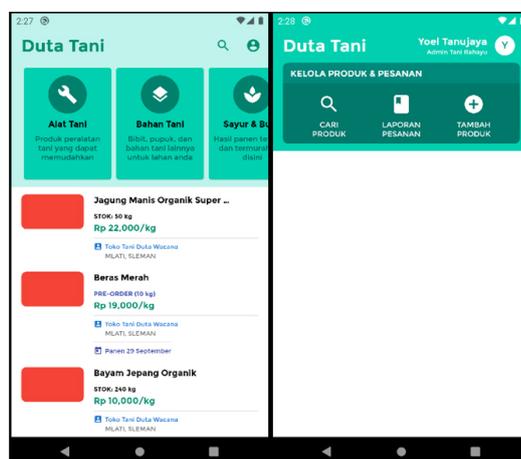
Usability Testing

Proses pengujian aplikasi *Dutatani Store* akan dilakukan dengan dua cara, yaitu *functional testing* dan *usability testing*. *Usability testing* dilakukan dengan meminta responden untuk menyelesaikan *task scenario* pengujian, dijelaskan pada Tabel 1.

Pengujian usability aplikasi dilakukan secara *unmoderated* dan *online*, menggunakan aplikasi



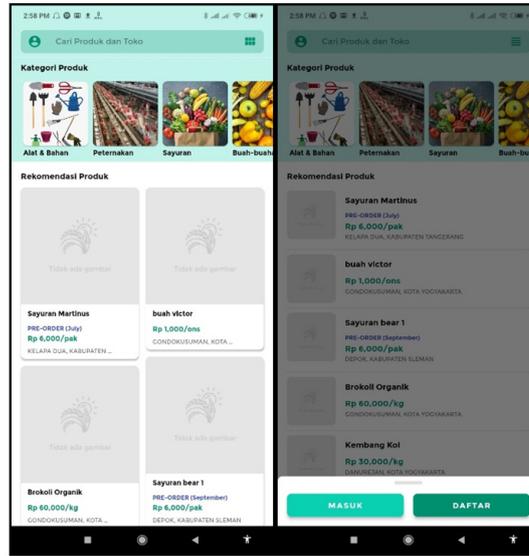
Gambar 8. Iterasi prototype kedua



Gambar 9. Iterasi prototipe ketiga

pengujian yang telah dimodifikasi untuk mengirimkan log berisi pilihan menu atau tombol atau item yang dipilih oleh responden, serta waktu responden memilih

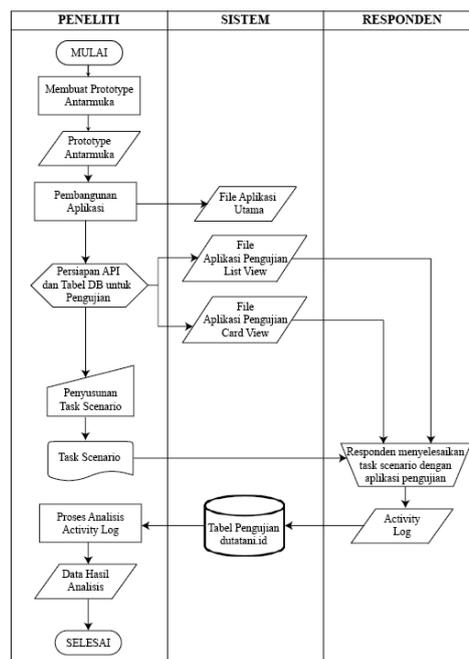
menu tersebut ke server Dutatani. Skema database untuk penyimpanan log pengujian dan proses pengujian secara online dapat dilihat pada Gambar 11 dan Gambar 12.



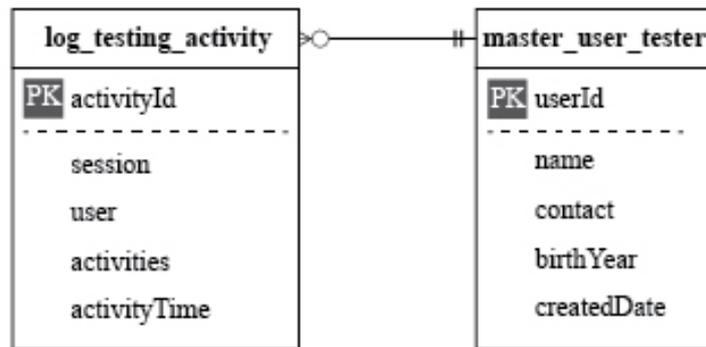
Gambar 10. Iterasi prototype keempat

Tabel 1. Daftar task scenario

No	Task scenario	Goal
1	Mencari produk telur ayam dari penjual yang berada di Kecamatan Gondokusuman	Responden memilih produk yang mengandung nama “telur ayam” milik penjual yang berada di Kecamatan Gondokusuman
2	Memilih produk beras dengan harga terendah	Responden memilih produk yang mengandung nama “beras” dengan harga terendah
3	Mencari produk yang berstatus pre-order pada bulan Oktober	Responden memilih produk yang berstatus pre-order bulan Oktober
4	Mencari produk bawang merah	Responden memilih produk yang mengandung nama “bawang merah”
5	Mencari/memilih satu produk sayuran, satu produk cabai rawit, dan satu produk organik	Responden memilih tiga produk dengan masing-masing kondisi



Gambar 11. Alur pengujian aplikasi



Gambar 12. Skema basis data pengujian

HASIL DAN PEMBAHASAN

Implementasi Sistem

Terdapat dua halaman utama pada aplikasi, yaitu halaman pembeli dan halaman penjual. Pengguna dapat menjalankan aplikasi Dutatani tanpa perlu mendaftarkan akun untuk menu yang bersifat pencarian produk atau informasi tentang toko. Sementara pengguna yang ingin atau telah mendaftarkan dirinya melalui proses registrasi, dapat menjual produk pertanian miliknya di Dutatani, dan mendapat fitur-fitur khusus untuk penjual seperti laporan pemesanan, pengaturan produk, dan profil.

Gambar 13 adalah halaman utama pembeli. Pada halaman utama pembeli, pengguna bisa melakukan pencarian produk, pencarian toko, dan mengakses menu akun Dutatani. Pencarian produk dapat dilakukan dengan tiga cara. Pertama, menggunakan menu pencarian di *textfield* (Gambar 13b). Kedua, mencari produk berdasarkan kategori. Ketiga, mencari produk pada bagian rekomendasi.

Setiap item produk yang dipilih akan menampilkan modal window berisi detail produk seperti nama, harga, satuan, toko, dan kontak toko. Pada menu pencarian produk, pengguna dapat mem-*filter* hasil pencarian berdasarkan status produk yang dijual. Untuk melihat toko, pengguna dapat menggunakan menu pencarian dan memilih tab toko, atau pengguna dapat memilih ikon ‘i’ pada *modal* detail produk (Gambar 13c) untuk masuk ke halaman toko.

Gambar 13d adalah halaman toko, pengguna dapat melihat produk yang dijual di toko tersebut, detail informasi toko seperti lokasi dan kontak toko, tombol untuk menelepon dan melakukan pencarian alamat toko dengan aplikasi Google Maps. Pada seluruh halaman, tampilan produk dapat diubah dari *card view* ke *list view* atau sebaliknya, dengan memilih ikon di kanan atas.

Gambar 15a menampilkan menu masuk/login, dimana pengguna dapat mengisi username dan kata sandi untuk masuk ke akun halaman penjualan, atau jika pengguna lupa kata sandi, dapat menekan teks ‘Lupa Kata Sandi?’

untuk mengganti kata sandi. Gambar 15b menampilkan halaman penggantian kata sandi, pengguna dapat memasukkan username akun, kemudian menjawab pertanyaan verifikasi yang telah diisi saat registrasi. Jika jawaban valid, pengguna dapat mengganti kata sandi akun dengan data yang baru. Proses *login* atau registrasi yang berhasil akan otomatis menampilkan halaman utama penjual seperti pada Gambar 16a.

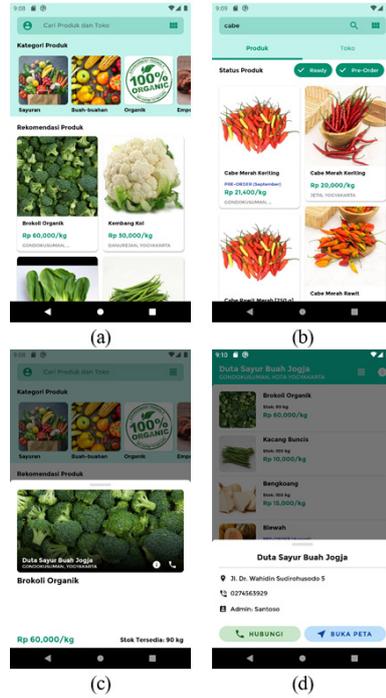
Halaman utama penjualan menampilkan daftar produk, tombol untuk menambahkan produk baru, tombol untuk masuk ke halaman pemesanan, dan tombol profil di kiri atas. Setiap item produk terdapat menu untuk edit atau hapus produk. Memilih menu + produk, pengguna akan diarahkan ke halaman Tambah produk pada Gambar 16b. Setelah data terisi lengkap dan foto berhasil diunggah, pengguna akan diarahkan kembali ke halaman utama penjual.

Gambar 17a merupakan halaman pemesanan yang dapat diakses dengan memilih tombol ‘Pesanan’ dari halaman utama penjual. Pada halaman ini pengguna dapat melihat daftar pesanan urut berdasarkan waktu pesanan. Tiap item pesanan dapat dihapus dengan menggeser item pesanan ke arah kanan. Untuk mengubah urutan pengguna, tekan tombol di bagian kiri atas dan memilih sistem urutan berdasarkan waktu, daerah pemesan, atau jumlah pesanan. Tombol (+) pada bagian kanan atas digunakan untuk menambahkan pesanan baru. Data pesanan berisi nama produk yang dipesan, jumlah, daerah pemesan berdasarkan provinsi, kabupaten, dan kecamatan (Gambar 17b).

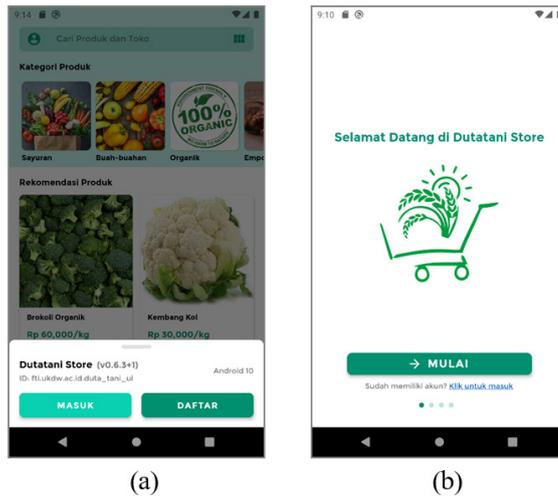
Memilih ikon di bagian kiri atas akan menampilkan halaman profil (Gambar 18a). Di halaman ini pengguna dapat mengubah profil dengan memilih ‘Edit Profil’ menuju halaman edit (Gambar 18b) atau melakukan logout akun dengan memilih menu ‘Keluar Akun’.

Penggunaan Aplikasi

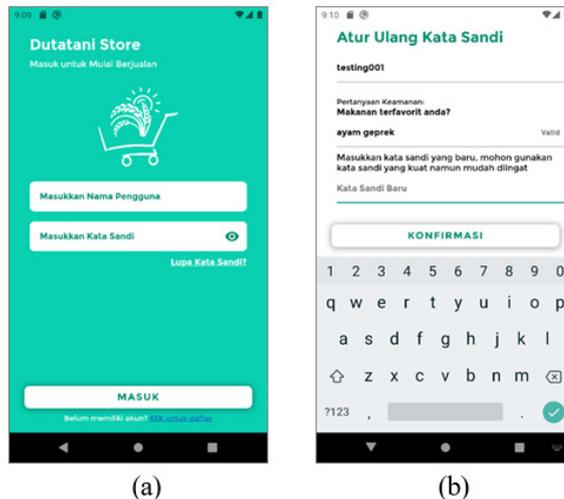
Aplikasi Dutatani Store masih berada pada tahap pengembangan awal untuk keperluan pengujian usabilitas aplikasi, sehingga penggunaan dalam tim



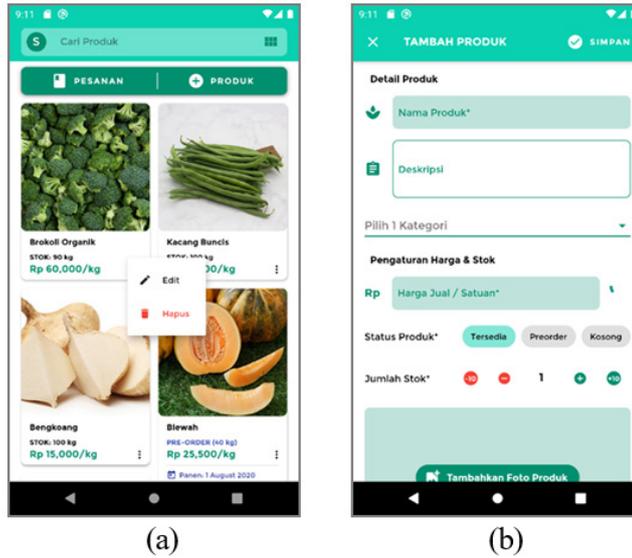
Gambar 13. Halaman utama pembeli



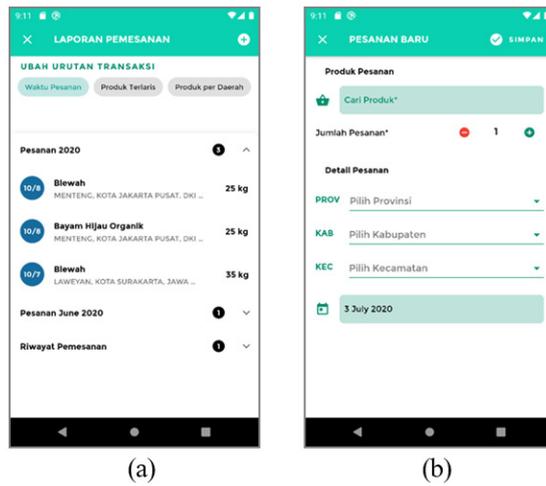
Gambar 14. Halaman akun dan registrasi



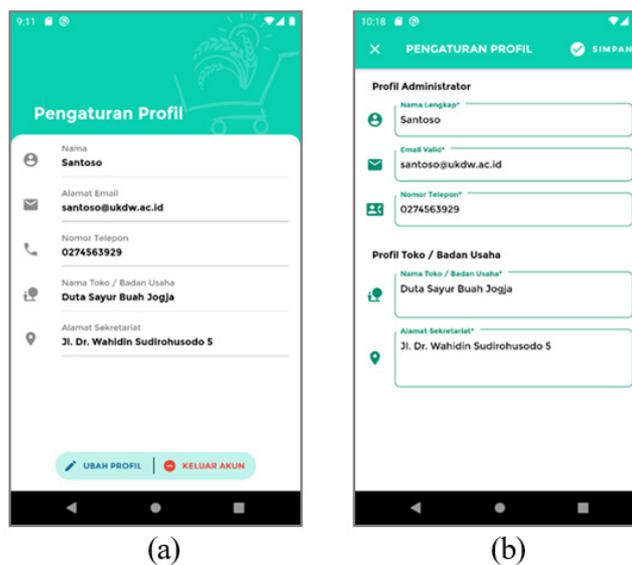
Gambar 15. Halaman login dan ubah password



Gambar 16. Halaman penjual dan tambah produk



Gambar 17. Halaman pemesanan



Gambar 18. Halaman profil

maupun ke pihak eksternal masih terbilang rendah. Ke depan, aplikasi Dutatani Store akan melakukan integrasi ke beberapa layanan eksternal, tidak terbatas pada kanal pembayaran *digital*, penggunaan akun sosial media untuk pendaftaran akun, penggunaan *One Time Password* (OTP) untuk mengakses akun Dutatani secara aman, dan lain lain.

Pengujian Online

Untuk menanggulangi dampak pandemi Covid-19 terhadap uji usabilitas, pengujian dilakukan secara *online* dan *unmoderated*, menggunakan aplikasi yang telah dimodifikasi untuk dapat merekam setiap menu yang dipilih oleh responden dalam bentuk *log* sederhana, kemudian mengirim *log* tersebut ke *database* Dutatani. Baris *code* fungsi pendaftaran responden dan pengiriman *log* terdapat pada Gambar 19 hingga Gambar 22. Halaman pendaftaran responden aplikasi dapat dilihat pada Gambar 23.

Setiap responden yang mendaftarkan diri tersimpan pada sistem sebagai 1 ID, sehingga dapat di-*filter* dengan mudah ketika melakukan analisis hasil. ID responden di *generate* secara otomatis, dan responden yang sama dapat mendaftar lebih dari satu kali. Setiap ID memiliki data nama, tahun lahir, kontak, dan waktu responden

melakukan pendaftaran. ID tersebut disimpan oleh aplikasi untuk memudahkan responden melanjutkan *task* jika keluar dari aplikasi secara tidak sengaja.

Terdapat beberapa *task scenario* yang harus diuji oleh responden, tiap *task* tersimpan sebagai *session*, sehingga peneliti dapat menentukan berapa lama suatu *task* diselesaikan oleh responden. Setelah menyelesaikan seluruh *task*, responden diminta memilih menu ‘Selesai’ pada aplikasi, sehingga aplikasi akan otomatis keluar dan tidak mengirimkan *log* lagi ke *database*. Pembagian *task* dibuat per sesi *task*, setiap sesi *task* memiliki waktu mulai dan berakhir (ketika responden menekan lanjut *task* berikutnya pada menu pengujian seperti pada Gambar 23b).

Gambar 24 adalah data *log* responden, mencakup ID, nama, kontak, tahun lahir, dan waktu responden memulai pengujian. Gambar 25 adalah contoh potongan *log* aktivitas pengujian. Penjelasan *string log* “melihat card produk Bayam Jepang Organik dari toko Tani Rahayu, card ProductItemCard” menandakan responden melihat produk bernama “Bayam Jepang Organik”, milik toko “Tani Rahayu”, aplikasi pengujian yang diuji menggunakan tipe tampilan “Card View”. “ProductItemCard” adalah *widget* yang diakses, yaitu *widget* item produk bertipe *card*.

```
showToast(String session, String name, String activity, String _class) {
//   String msg = activity+', card '+ class;
String msg = activity+', list '+ class;
Future<String> add = TestActivity().addActivities(new TestActivity(session: session,
user: name, activities: msg, activityTime: DateTime.now()));
add.then((res) {
  print(msg);
});
}
}
```

Gambar 19. Code fungsi pengiriman log

```
@JsonSerializable(explicitToJson: true)
class TestActivity {
String session;
String user;
String activities;
DateTime activityTime;

TestActivity(this.session, this.user, this.activities, this.activityTime);

factory TestActivity.fromJson(Map<String, dynamic> json) => _TestActivityFromJson(json);
Map<String, dynamic> toJson() => _TestActivityToJson(this);

Future<String> addActivities(TestActivity ta) async {
final body = jsonEncode(_TestActivityToJson(ta));
var response = await http.Client().post('http://dutatani.id/si_mcommerce/api/testing/new_activity.php',
body: body, headers: {"Content-type": "application/json"});
// var response = await http.Client().post('http://10.0.2.2/dutatani_api/api/testing/new_activity.php',
body: body, headers: {"Content-type": "application/json"});

if (response.statusCode == 200) {
return response.body;
} else {
throw Exception('Failed to create activities');
}
}
}
```

Gambar 20. Code pengelolaan log aktivitas

```
setState(() {
  _isLoading = true;
  var r = Random.secure();
  String id = randomAlphaNumeric(3, provider: CoreRandomProvider.from(r));
  Future<String> createTester = Tester().createTester(new Tester(userid: id, name: name, contact: contact,
  birthYear: year, createdAt: DateTime.now()));
  createTester.then((onValue) async {
    SharedPreferences prefs = await SharedPreferences.getInstance();
    prefs.setString('user', id);
    prefs.setString('session', 'TASK 1');
    setState(() {
      _isLoading = false;
    });
    Navigator.pushAndRemoveUntil(context, MaterialPageRoute(builder: (context) => HomeActivity(name: id,
    session: 'TASK 1',)), ModalRoute.withName('TesteLogin'));
    CustomWidget().showToast('TASK 1', id, 'start', widget.runtimeType.toString());
  }).catchError((onError) {
    setState(() {
      _isLoading = false;
    });
    _testScaffold.currentState.showSnackBar(SnackBar(content: Text('Gagal mendaftar pengujian. Silahkan coba
    lagi.')));
  });
});
```

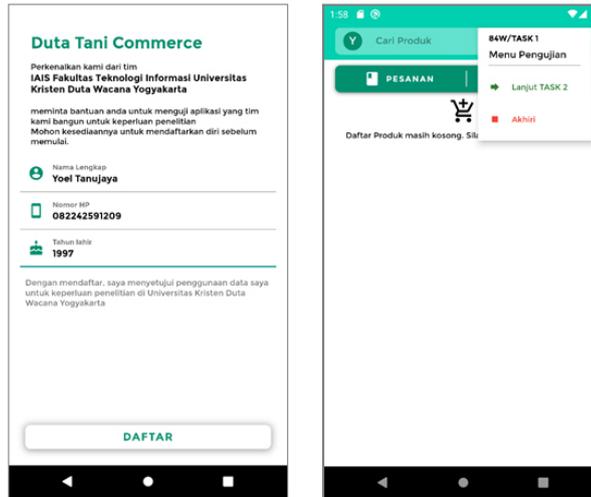
Gambar 21. Code pendaftaran responden

```

setState() {
  _isLoading = true;
  var r = Random.secure();
  String id = randomAlphaNumeric(3, provider: CoreRandomProvider.from(r));
  Future<String> createTester = Tester().createTester(new Tester(userId: id, name: name, contact: contact,
  birthYear: year, createdAt: DateTime.now()));
  createTester.then(onValue) async {
    SharedPreferences prefs = await SharedPreferences.getInstance();
    prefs.setString('user', id);
    prefs.setString('session', 'TASK 1');
    setState() {
      _isLoading = false;
    };
    Navigator.pushAndRemoveUntil(context, MaterialPageRoute(builder: (context) => HomeActivity(name: id,
    session: 'TASK 1',)), ModalRoute.withName('/testerLogin'));
    CustomWidget().showToast('TASK 1', id, 'start', widget.runtimeType.toString());
  }).catchError((onError) {
    setState() {
      _isLoading = false;
    };
    _testScaffold.currentState.showSnackBar(SnackBar(content: Text('Gagal mendaftar pengujian. Silahkan coba
    lagi.')));
  });
});

```

Gambar 22. Code pengelolaan data responden



(a)

(b)

Gambar 23. Halaman pendaftaran responden dan menu pengujian

```

{
  "id": "210",
  "name": "Yoel",
  "contact": "083822525315",
  "birthYear": "1997",
  "createdAt": "2020-05-31 19:28:29"
},

```

Gambar 24. Data responden

```

"TASK 1": [
  {
    "activities": "start, card Testinglogin",
    "activityTime": "2020-05-14 11:16:14"
  },
  {
    "activities": "melihat card produk Cabe Rawit Merah dari toko Tani Rahayu, card ProductItemCard",
    "activityTime": "2020-05-14 11:16:18"
  },
  {
    "activities": "dismiss detail produk Cabe Rawit Merah, card DetailProductModalSheet",
    "activityTime": "2020-05-14 11:16:19"
  },
  {
    "activities": "melihat card produk Bayam Jepang Organik dari toko Tani Rahayu, card ProductItemCard",
    "activityTime": "2020-05-14 11:16:20"
  },
  {
    "activities": "dismiss detail produk Bayam Jepang Organik, card DetailProductModalSheet",
    "activityTime": "2020-05-14 11:16:21"
  },
  {
    "activities": "memilih menu akun, card HomeActivity",
    "activityTime": "2020-05-14 11:16:24"
  },
]

```

Gambar 25. Data log pengujian

Hasil dan Analisis

Pengujian dilakukan secara *online* dengan meminta bantuan 14 orang responden untuk menyelesaikan *task scenario*, tujuh orang menggunakan aplikasi *card view*, dan tujuh orang lainnya menggunakan aplikasi *list view*.

Tabel 2 menjabarkan waktu setiap responden yang menggunakan aplikasi *card view* dalam menyelesaikan masing-masing *task scenario*. Tabel 3 menjabarkan waktu setiap responden yang menggunakan aplikasi *list view*. Terdapat dua data yang bernilai 0 pada waktu pengujian. Hal ini karena adanya masalah pada pengiriman *log* dari perangkat responden menuju *server* Dutatani, sehingga *log* aktivitas pada *task* tersebut tidak berhasil terkirim.

Dalam proses perhitungan *rata-rata* akan dilakukan pengecualian untuk data dengan waktu 0. Task 1, 2, dan 4 memiliki *rata-rata* waktu pada *range* 13-23 detik, sedangkan Task 3 dan 5 mencatatkan *rata-rata* waktu lebih dari 30 detik. Dalam kasus penyelesaian Task 3, beberapa responden tidak tahu bagaimana melakukan pencarian produk berdasarkan status produk dari halaman awal, sehingga responden harus masuk ke tiap kategori atau *scrolling* dari daftar rekomendasi.

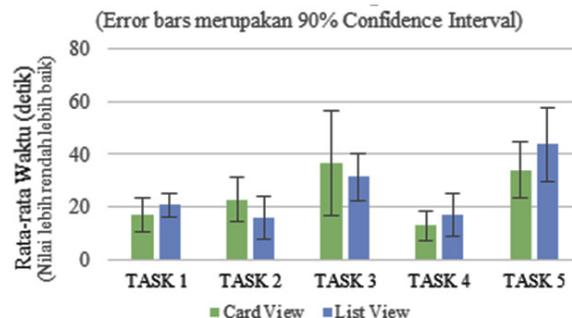
Secara keseluruhan, responden aplikasi dengan tampilan *list view* menyelesaikan lima *task* yang ditentukan dengan *rata-rata* waktu 26,6 detik per *task*, sedangkan *rata-rata* waktu responden *card view* adalah 26,51 detik per *task*. Hasil *rata-rata* waktu per *task* direpresentasikan pada Gambar 26, dan *rata-rata* waktu

Tabel 2. Time on task responden Card View

ID	Hasil time on task (dalam detik)				
	T1	T2	T3	T4	T5
6H7	24	31	30	0	68
46E	10	32	0	23	30
M50	6	10	20	11	27
705	12	6	49	5	43
K23	10	22	12	11	29
2FB	21	43	96	18	24
213	36	16	50	24	17
Rata-rata	17,0	22,9	36,7	13,1	34
Standar deviasi	10,57	13,25	31,95	9,01	16,91

Tabel 3. Time on task responden list view

ID	Hasil time on task (dalam detik)				
	T1	T2	T3	T4	T5
368	22	16	31	9	42
263	32	26	49	31	92
373	12	7	39	9	30
X84	22	14	51	41	43
785	10	8	8	11	24
276	16	43	28	7	48
0Tm	19	7	30	18	36
Rata-rata	20,88	16,13	31,38	17,13	43,75
Standar Deviasi	7,37	13,21	14,58	13,10	22,28



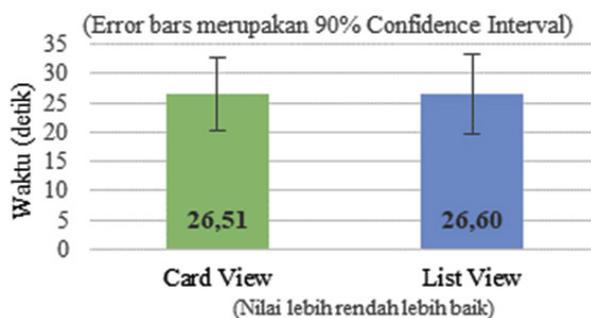
Gambar 26. Rata-rata waktu per task

seluruh *task* per tipe tampilan disajikan pada Gambar 27.

Selanjutnya adalah detail hasil akhir dari *task* untuk masing-masing responden. Tabel 4 merupakan jabaran hasil *task* untuk responden dengan aplikasi *card view*, dan Tabel 5 merupakan jabaran hasil *task* responden aplikasi *list view*. Terdapat tiga sesi *task* yang berakhir *Failed* (dua diantaranya dengan waktu 0) dan dua sesi *task* yang berakhir *Partially Success* untuk aplikasi *card view*, sementara untuk *list view* hanya terdapat dua sesi

task yang berakhir *Partially Success*. *Partially Success* merupakan *task* yang berhasil diselesaikan namun hasil akhir kurang tepat, atau kurang lengkap untuk kasus Task 5.

Rasio kesuksesan untuk *task* yang dilakukan dengan aplikasi *list view* adalah 97,1% (4,9 *task success* dari 5 *task*), sedangkan rasio kesuksesan *task card view* 88,6% (4,4 *task success* dari 5 *task*). Hasil rata-rata keberhasilan *task* disajikan pada Gambar 28.



Gambar 27. Rata-rata waktu seluruh task

Tabel 4. Hasil task responden card view

ID	Hasil kesuksesan task scenario				
	T1	T2	T3	T4	T5
6H7	S	S	S	F	S
46E	S	S	F	S	S
M50	S	S	S	S	S
705	S	S	S	S	S
K23	F	S	S	S	S
2FB	S	S	S	S	S
213	P	S	S	S	P
Success ratio (%)	79	100	86	86	93
Success	5	7	6	6	6
Partial	1	0	0	0	1
Failed	1	0	1	1	0

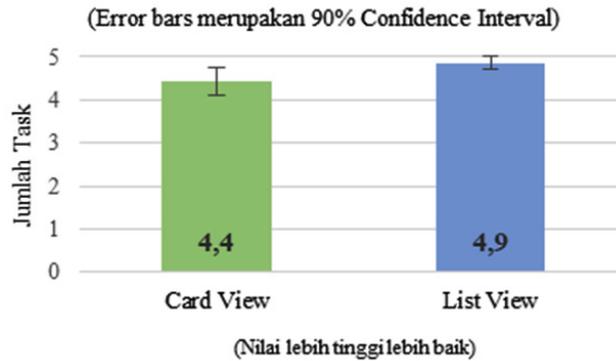
Tabel 5. Hasil task responden list view

ID	Hasil kesuksesan task scenario				
	T1	T2	T3	T4	T5
368	S	S	S	S	P
263	S	S	S	S	S
373	S	S	S	S	S
X84	S	S	S	S	S
785	S	S	S	S	S
276	S	P	S	S	S
0Tm	S	S	S	S	S
Success Ratio (%)	100	93	100	100	93
Success	7	6	7	7	6
Partial	0	1	0	0	1
Failed	0	0	0	0	0

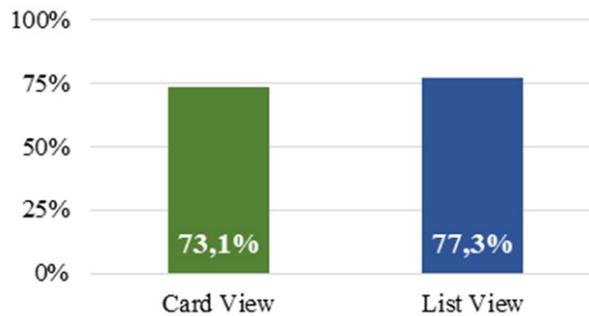
Tabel 6 menjabarkan hasil *performance metrics* keseluruhan *task* masing-masing responden untuk tampilan *card view* dan *list view*. *List view* mendapat rata-rata *combined metrics* 77,27%, lebih tinggi dibandingkan dengan *card view* dengan nilai 73,11%. Hasil *combined metrics* direpresentasikan pada Gambar 29.

Terdapat beberapa keterbatasan yang ditemui dalam proses pengujian usabilitas aplikasi, yaitu dalam hal responden pengujian. Responden pengujian masih belum sesuai dengan target responden utama untuk

aplikasi ini, yaitu petani, anggota kelompok tani, *supplier* produk, dan penjual produk. Adanya pembatasan sosial berskala besar menyusul meluasnya pandemi Covid-19 di Indonesia yang menyebabkan pembatalan rencana pengujian langsung kepada kelompok tani, sehingga pengujian dilakukan oleh kalangan umum yang memiliki pengetahuan tentang informatika. Selain responden, rendahnya jumlah sampel data produk untuk pengujian mengakibatkan signifikansi hasil uji masih rendah.



Gambar 28. Rata-rata keberhasilan task



Gambar 29. Hasil combined metrics

Tabel 6. Hasil performance metrics usability test

ID	Time	Success ratio	Time ratio	Combined
6H7	38,25	80%	23%	51%
46E	23,75	80%	66%	73%
M50	14,8	100%	92%	96%
705	23,0	100%	68%	84%
K23	16,8	80%	86%	83%
2FB	40,4	100%	17%	58%
213	28,6	80%	51%	66%
368	24,0	90%	65%	78%
263	46,0	100%	0%	50%
373	19,4	100%	79%	89%
X84	34,2	100%	35%	67%
785	12,2	100%	100%	100%
276	28,4	90%	52%	71%
0Tm	22,0	100%	71%	86%

Proses pembangunan aplikasi *Dutatani Store* dilakukan selama dua bulan, dari Maret 2020 hingga Mei 2020. Selama dua bulan terdapat banyak perubahan dari *user* terhadap fitur dan fungsi yang diaplikasikan. Penggunaan metode pengembangan ini setiap perubahan fitur dapat diimplementasikan lebih mudah dan lebih cepat, serta setiap *bug* atau kesalahan proses dapat diketahui dengan segera. Proses ini lebih dinamis dari segi implementasi fitur, dan lebih mudah di monitor dengan adanya *git* dan laporan harian dari masing-masing anggota tim. Salah satu contoh perubahan yang cukup signifikan pada aplikasi *Dutatani Store* adalah penambahan fitur registrasi.

Pada rancangan awal, *member* *Dutatani* di *input* secara manual, langsung melalui *database*, sehingga tidak ada fitur untuk pengelolaan akun (*login*, *logout*, *register*, ubah *password*). Tim pengembang dapat langsung menambahkan desain antarmuka untuk registrasi dan *login*, menambahkan API untuk mengendalikan fitur *login* dan *logout*, serta menerapkan fitur tersebut pada revisi aplikasi dalam waktu +- 2 minggu. Dengan penggunaan metode RAD, fitur tersebut bisa diimplementasikan di tengah proses *prototyping* iteratif, dan dapat dievaluasi.

Responden pengujian usability membagikan umpan balik untuk beberapa komponen aplikasi, antara lain:

- Tombol untuk mengakses menu akun sulit ditemukan
- Tidak ada tombol *back* di bagian kiri atas judul menu, sehingga harus menggunakan tombol *back* sistem operasi yang menyebabkan aplikasi keluar tiba-tiba
- Tombol edit profil sulit ditemukan.

Selain itu, terdapat kendala komunikasi antara aplikasi dengan *server* *Dutatani*, baik dalam hal pengiriman data yang besar seperti foto produk, atau pengiriman data dalam frekuensi yang tinggi seperti *log* pengujian.

KESIMPULAN

Hasil uji usability menyimpulkan aplikasi berjalan sesuai dengan requirement yang telah ditetapkan, dan tingkat kesuksesan responden dalam menyelesaikan lima *task scenario* pengujian mencapai 93%, dengan rata-rata waktu penyelesaian seluruh *task* 26,6 detik. Pengujian ini juga menunjukkan perbedaan hasil antara tampilan *card view* dan *list view*, dimana tampilan *list view* memiliki tingkat kebergunaan lebih baik dibandingkan dengan *card view*, dengan nilai gabungan 77,27% berbanding 73,11%, dan tingkat kesuksesan penyelesaian *task scenario* mencapai 98%, dibandingkan dengan 88%.

Pendekatan *Rapid Application Development* juga berdampak positif terhadap proses pengembangan aplikasi, dimana seluruh proyek dapat diselesaikan dalam kurun waktu dua bulan, dengan jumlah anggota tim yang terlibat empat orang.

UCAPAN TERIMA KASIH

Terima kasih disampaikan kepada tim *Dutatani* dan Lembaga Penelitian dan Pengabdian pada Masyarakat (LPPM) Universitas Kristen Duta Wacana yang telah mendukung proses pembangunan aplikasi dan penelitian ini. Terima kasih juga disampaikan kepada Fakultas Teknologi Informasi Universitas Kristen Duta Wacana yang telah memfasilitasi proses penelitian selama kurang lebih 5 bulan.

DAFTAR PUSTAKA

- Badan Pusat Statistik. 2016. Sensus Ekonomi 2016. Jakarta.
- Chrismanto, A. R., R. Delima, A. Wibowo, H. B. Santoso, and R. A. Kristiawan. 2019. Developing Agriculture Land Mapping using Rapid Application Development (RAD): A Case Study from Indonesia. *International Journal of Advanced Computer Science and Applications*. pp. 232-241.
- Delima, R., H. B. Santoso, and J. Purwadi. 2016. Architecture vision for Indonesian Integrated Agriculture Information Systems using TOGAF framework. 2016 International Conference on Informatics and Computing (ICIC). Lombok: IEEE.
- Delima, R., H. B. Santoso, G. H. Aditya, J. Purwadi, and A. Wibowo. 2018. Development of Sales Modules for Agricultural E-Commerce Using Dynamic System Development Method. *International Journal of New Media Technology*. pp. 95-103.
- Delima, R., H. B. Santoso, N. Andriyanto, and A. Wibowo. 2018. Development of Purchasing Module for Agriculture E-Commerce using Dynamic System Development Model. *International Journal of Advanced Computer Science and Applications*. pp. 86-96.
- Kütz, M. 2016. Introduction to E-Commerce: Combining Business and Information Technology. Bookboon.com.