Jurnal Ilmu Komputer Agri-Informatika

Tersedia secara *online* di:
http://journal.ipb.ac.id/index.

Volume 7 Nomor 2 halaman 124– 134
e ISSN: 2654-9735, p ISSN: 2089-6026

Pengembangan Aplikasi Android Patriot Pangan sebagai Sarana *e-Participation* untuk Sistem Ketahanan Pangan Nasional

Patriot Pangan Android Application Development as a Means of e-Participation for the National Food Security System

FAJAR MAULANA[1]*, DEAN APRIANA RAMADHAN[1]

Abstrak

Permasalahan ketahanan dan kerawanan pangan masih menjadi salah satu perhatian utama di dunia. Sistem ketahanan pangan Indonesia saat ini masih terkendala dengan sulitnya koordinasi antarsektor sehingga proses pengumpulan data masih berjalan lambat dan tidak rutin dilakukan. Patriot Pangan dapat menjadi solusi untuk sistem *input* data kerawanan pangan secara *real-time* dengan menerapkan konsep *e-Participation.* Penelitian ini difokuskan pada pembuatan aplikasi *mobile* dengan sistem operasi Android dari sistem Patriot Pangan yang mampu menjadi sarana *e-Participation* untuk membantu pemerintah dalam mewujudkan ketahanan pangan nasional. Penelitian ini dilakukan menggunakan Scrum sebagai metode pengembangan dengan jumlah *sprint* sebanyak lima *sprint*. Komunikasi data antara server dan aplikasi *mobile* dilakukan dengan metode REST API. Hasil penelitian yang dilakukan adalah pengembangan aplikasi *mobile* yang mampu memfasilitasi masyarakat untuk melaporkan kerawanan pangan di sekitar mereka melalui ponsel pintar pribadi mereka dengan atau tanpa koneksi internet.

Kata Kunci**:** android, *e-participation*, kerawanan, *mobile*, pangan.

Abstract

Food insecurity and vulnerability is still one of the main concerns in the world. Indonesia's current food security system has a coordination problem between its sector. Consequently, data collection is slow and not frequent. Patriot Pangan can be a solution with a system for detecting food insecurity in real-time by implementing the concept of e-Participation. This research is focused on creating an Android mobile app from Patriot Pangan system that is able to become a means of e-Participation to assist the government in realizing national food security. This research was conducted using Scrum as a development method, with a total of five sprints. Data communication between the server *and mobile app is done using REST API method. This result of this research is a mobile app that is able to facilitate the public to report food insecurities through their personal smartphones with or without an internet connection.*

Keywords: android, e-participation, food, insecurities, mobile.

PENDAHULUAN

Permasalahan mendesak yang dihadapi oleh banyak negara berkembang salah satunya adalah mengenai ketahanan pangan dan kerawanan pangan. Kegagalan sebuah negara dalam membuat program ketahanan pangan yang baik dapat berdampak pada terancamnya masa depan negara tersebut (FAO 2010). Pemerintah melalui Kementerian Pertanian telah membuat sebuah sistem ketahanan pangan nasional bernama Sistem Kewaspadaan Pangan dan Gizi (SKPG) sebagai alat deteksi dini kerawanan pangan yang diharapkan dapat membantu penyelenggaraan program ketahanan pangan di Indonesia. Namun, Lamabelawa (2006) mengatakan bahwa SKPG masih terkendala dengan sulitnya koordinasi antarsektor sehingga

[1]Departemen Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Pertanian Bogor, Jl. Meranti Wing 20 lv 5-6, Kampus IPB Dramaga, Dramaga, Bogor 16680;
*Penulis Korespondensi: Surel: of.maulana@gmail.com

proses yang dilakukan untuk mengumpulkan dan melaporkan data yang ada di lapangan berjalan dengan lambat dan tidak rutin.

Salah satu manfaat dari perkembangan internet yang semakin mudah diakses oleh masyarakat adalah *e-Participation. E-Participation* atau partisipasi digital berfungsi sebagai alat pemerintah dalam mengajak masyarakat terlibat dalam pengambilan kebijakan pemerintah (Banjac 2017). Keuntungan dari penggunaan *e-Participation* adalah pemerintah mampu mengetahui kondisi akurat dari sebuah situasi di tempat tertentu (Macintosh 2004). Hal ini dapat bermanfaat bagi sistem ketahanan pangan di Indonesia karena dengan menerapkan *e-Participation* pemerintah bisa mendapatkan data yang lebih akurat dan cepat mengenai kerawanan pangan di suatu daerah. Selain internet, perkembangan teknologi lainnya yang cukup signifikan pertumbuhan penggunaannya di masyarakat adalah ponsel pintar. Menurut data survei Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) di bulan April tahun 2018, diketahui penetrasi internet di Indonesia adalah 143.26 juta jiwa atau sekitar 54.68% dari total penduduk di Indonesia. Sebanyak 70.96% dari penduduk urban, 45.42% dari penduduk rural urban, dan 42.06% penduduk rural memiliki ponsel pintar.

Sistem Patriot Pangan (Panatagama *et* al. 2019; Asfarian *et al*. 2020; Nurhadryani *et al*. 2020) berusaha menyelesaikan permasalahan lambatnya pelaporan kerawanan pangan di Indonesia dengan menggunakan internet secara *crowdsourcing*. *Crowdsourcing* adalah solusi efektif dalam menjalankan sebuah pekerjaan yang sulit dilakukan oleh sebuah komputer dengan mengandalkan bantuan banyak manusia untuk menyelesaikan pekerjaan tersebut (Garcia-Molina *et al.* 2017). Dalam sistem Patriot Pangan, pekerjaan ini adalah mengumpulkan data kerawanan pangan yang pada penelitian ini adalah jumlah keluarga yang mengalami kerawanan pangan di berbagai daerah secara *real-time* di berbagai tempat di Indonesia. Pekerjaan tersebut dilakukan oleh relawan yang disebut Patriot Pangan yang dapat berasal dari anggota karang taruna ataupun masyarakat umum yang tertarik untuk menjadi relawan. Penggunaan ponsel pintar yang semakin terjangkau oleh masyarakat menjadikan ponsel pintar sebagai alat yang baik dalam mengumpulkan dan menyebarkan data lingkungan di sekitar mereka (Kanhere 2011). Berdasarkan hal tersebut dan data penggunaan ponsel pintar di Indonesia, diperlukan pengembangan aplikasi *mobile* yang mampu membantu masyarakat dalam melaporkan kerawanan pangan di sekitar mereka.

Cara kerja sistem Patriot Pangan adalah relawan yang sudah terverifikasi oleh *admin* Patriot Pangan akan mengunjungi rumah warga untuk melakukan survei kepada kepala keluarga atau perwakilan keluarga lainnya tentang kondisi kerawanan pangan keluarga tersebut menggunakan indikator kerawanan pangan yang telah disediakan oleh *product owner*. Hasil survei yang didapatkan kemudian akan dikumpulkan di server Patriot Pangan dan data tersebut dapat digunakan oleh pihak yang membutuhkan, seperti pemerintah atau lembaga lain untuk melihat status kerawanan pangan suatu daerah berdasarkan banyaknya keluarga yang mengalami kerawanan pangan di daerah tersebut. Selain itu, relawan yang telah mendapatkan capaian jumlah pelaporan tertentu bisa mendapatkan *reward* dengan harapan relawan tersebut semakin bersemangat untuk melakukan survei.

Penelitian ini ditujukan untuk mengembangkan *front-end* dari sistem Patriot Pangan yang berupa sebuah aplikasi *mobile* Android sebagai sarana *e-Participation.* Versi Android paling rendah untuk menjalankan aplikasi yang dikembangkan adalah versi 5.0 Lollipop. Aplikasi yang dikembangkan berfungsi sebagai alat dalam melakukan survei data kerawanan pangan dengan menggunakan indikator kerawanan pangan yang telah disediakan oleh *product owner* dalam menentukan apakah daerah tersebut mengalami kerawanan pangan atau tidak. Sistem Patriot Pangan akan mengirimkan data yang dikumpulkan oleh masyarakat ke pemerintah ataupun lembaga lain yang berkepentingan secara *real-time,* sehingga sistem ketahanan pangan di Indonesia diharapkan dapat menjadi lebih baik lagi.

METODE

Rancangan Arsitektur Pengembangan

Arsitektur sistem yang akan dikembangkan pada penelitian ini merupakan hasil diskusi antara tim pengembang dengan pihak lain yang terkait, seperti *product owner, scrum master*, dan yang paling penting adalah diskusi antara pengembang yang terlibat di dalam penelitian. Arsitektur yang dibuat harus mempertimbangkan kemampuan tim dalam pengembangan sehingga memperkecil kemungkinan untuk terjadinya kendala selama pengembangan dilakukan.

Tahapan Penelitian

Penelitian dilakukan dengan menggunakan *scrum* sebagai metode pengembangan perangkat lunak. *Scrum* dirancang untuk menyelesaikan permasalahan yang bersifat kompleks dan adaptif secara kreatif dan produktif demi memberikan produk yang bernilai paling baik untuk permasalahan tersebut (Schwaber dan Sutherland 2017). Pemilihan *scrum* sebagai metode pengembangan dikarenakan pengembangan perangkat lunak dan sistem Patriot Pangan cukup kompleks dan masing-masing dari anggota tim yang terlibat dalam pengembangan sistem tersebut memiliki bidang pengembangan yang berbeda-beda, sehingga membutuhkan kolaborasi antar anggota tim yang baik dan efektif dalam penyelesaian penelitian ini.

Scrum roles yang terlibat di dalam pengembangan sistem Patriot Pangan terdiri dari *product owner*, *scrum master*, dan tim pengembang. Seluruh *scrum roles* akan bergabung dan berkolaborasi di sebuah tim yang disebut sebagai tim *scrum*. *Product owner* adalah pihak yang menentukan fungsi apa yang akan dikembangkan selama penelitian berlangsung, *scrum master* sebagai pihak yang memandu tim *scrum* bekerja agar penelitian dapat berjalan sesuai dengan kaidah *scrum*, serta tim pengembang yang bertugas untuk mengembangkan produk.

Langkah pengembangan dengan *scrum* yang dilakukan terdiri dari banyak *sprint* yang dilaksanakan. Setiap *sprint* berlangsung selama 2 minggu dan setiap *sprint* terdiri dari *sprint planning*, pengembangan produk, *daily scrum*, *sprint review*, dan *sprint retrospective*.

- *Sprint Planning*: *Sprint Planning* merupakan sebuah tahapan dimana tim *scrum* melakukan *brainstorming* untuk merencanakan pekerjaan yang akan dilakukan selama *sprint* berjalan. Penetapan keluaran dari *sprint* yang dijalankan atau *sprint goals* dan metode yang digunakan serta cara untuk mencapai *sprint goals* tersebut didiskusikan dengan seluruh anggota dari tim *scrum* dengan dipandu oleh *scrum master*. Pada tahap ini tim *scrum* memilih dan menentukan *product backlog* apa yang akan dimasukkan ke *sprint backlog e-participation* dan yang akan dimasukkan ke *sprint backlog* dari modul *e-initiative*.
- Pengembangan produk: Tahap ini mengembangkan produk yang dilakukan berdasarkan rencana pengembangan yang telah dibuat pada saat *sprint planning*. Masing-masing tim yang terlibat di pengembangan modul *e-participation* dan *e-initiative* akan mengerjakan *increment* dari sistem Patriot Pangan di tahap ini. Kebutuhan sistem yang terdapat pada *sprint backlog* akan dikerjakan sesuai dengan metode dan cara yang telah ditentukan.
- *Daily Scrum*: *Daily scrum* merupakan sebuah tahapan yang dilakukan setiap hari selama *sprint* berlangsung. *Daily scrum* bertujuan untuk memeriksa perkembangan dari pekerjaan yang dilakukan terhadap *sprint goals*, dan juga perkembangan dari *sprint* yang dijalankan terhadap penyelesaian proyek yang sedang dikerjakan. Pada *daily scrum*, tim *scrum* Patriot Pangan akan membuat rencana pekerjaan untuk 24 jam kedepan, dan dilakukan setiap harinya di waktu yang sama melalui bantuan *tools* seperti Slack.
- *Sprint Review*: *Sprint review* merupakan tahapan yang dilakukan pada saat sebuah *sprint* berakhir. Tim *scrum* Patriot Pangan akan berkumpul dengan bertatap muka mendiskusikan hasil pekerjaan dan memeriksa pekerjaan apa saja yang telah dilakukan selama *sprint* berlangsung. Tim akan menyesuaikan *product backlog* dengan situasi pengembangan yang terjadi selama *sprint* tersebut berlangsung. Hasil dari *sprint review* merupakan sebuah

product backlog yang telah diperbarui yang akan digunakan dalam proses *sprint planning* berikutnya.

- *Sprint Retrospective*: *Sprint retrospective* adalah tahapan dimana tim *scrum* Patriot Pangan melakukan introspeksi terhadap anggota tim dalam kinerjanya selama *sprint* tersebut berlangsung. Hal yang didiskusikan selama tahapan ini terkait dengan kinerja sumber daya manusia di dalam tim, kinerja dari *tools* yang digunakan, hubungan antar anggota tim, serta hal lain yang terkait dengan bagaimana pekerjaan dilaksanakan selama *sprint* berlangsung. *Sprint retrospective* dilakukan setelah *sprint review* dilakukan dan sebelum *sprint planning* berikutnya dimulai. Keluaran dari tahapan ini adalah rencana dalam meningkatkan kinerja tim *scrum* dalam *sprint* berikutnya.

Lingkungan Pengembangan

Spesifikasi perangkat keras dan perangkat lunak yang digunakan untuk penelitian ini dapat dilihat pada Tabel 1.

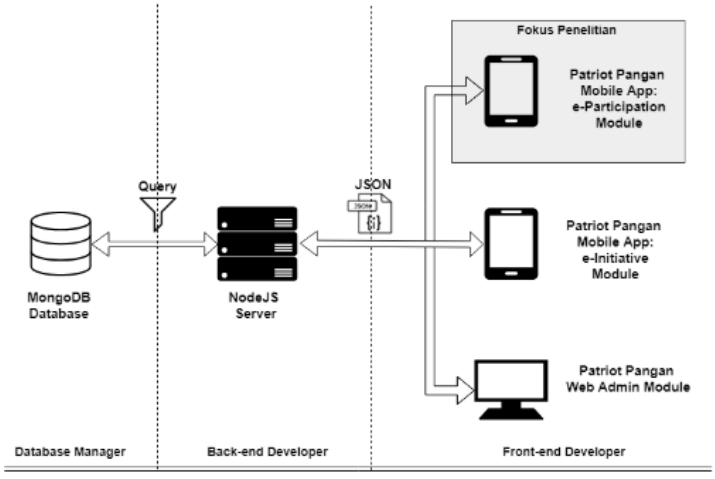
Tabel 1 Spesifikasi perangkat keras dan perangkat lunak yang digunakan

Perangkat Keras	Perangkat Lunak
CPU: Intel core i5 7200U @2.50 GHz	Android Studio 3.2
RAM : 8GB DDR4	Postman 6.6.1
GPU : Nvidia GeForce 930MX	Google Chrome 71.0.3578.98
Ponsel Pintar dengan RAM 4GB	Microsoft Windows 10 64-bit

HASIL DAN PEMBAHASAN

Arsitektur Sistem

Ilustrasi rancangan arsitektur sistem Patriot Pangan dapat dilihat pada Gambar 1. Arsitektur ini diadopsi dan disempurnakan berdasarkan konsep yang dibuat oleh Panatagama *et al*. (2019).



Gambar 1 Rancangan arsitektur pengembangan sistem Patriot Pangan.

Pengembangan sistem Patriot Pangan melibatkan lima orang tim pengembang yang terdiri dari satu orang *database manager*, satu orang *back-end developer*, dan tiga orang *front-end developer*. *Database manager* berperan sebagai pihak yang mengembangkan skema pangkalan data yang baik dan efisien serta melakukan optimasi pangkalan data. *Back-end developer* bertugas sebagai pihak yang menghubungkan *front-end* dengan pangkalan data dan *business logic* dari sistem yang akan dikembangkan. *Back-end developer* akan menyediakan *application programming interface* (API) yang akan digunakan oleh *front-end* sistem untuk berkomunikasi dengan pangkalan data dan server. *Front-end developer* memiliki peran untuk mengembangkan antarmuka yang akan digunakan oleh pengguna dalam mengakses sistem Patriot Pangan. *Front-end* dari sistem Patriot Pangan terbagi menjadi tiga modul, yaitu modul

e-Participation, dan modul *e-Initiative* yang keduanya dalam bentuk aplikasi *mobile*, serta modul *admin* dalam bentuk *website. Front-end developer* modul *e-Participation* menjadi ruang lingkup penelitian ini.

Product Backlog

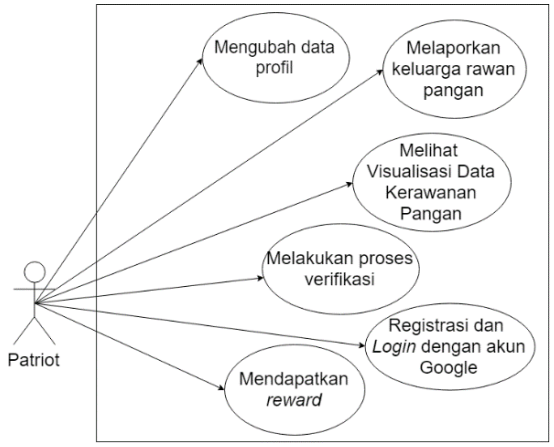
Pengembangan sistem Patriot Pangan dilakukan dengan menjadikan *product backlog* sebagai sumber utama dari dokumentasi daftar fitur yang akan diimplementasikan dalam sistem. *Product backlog* dibuat di awal *sprint* pertama berjalan berdasarkan hasil diskusi antara *product owner, scrum master*, dan *developers*. *Product backlog* akan diperbarui di akhir *sprint* pada tahap *sprint review* untuk memastikan bahwa *product backlog* yang baru telah disesuaikan dengan kebutuhan baru yang mungkin bertambah atau berubah ketika sebuah *sprint* berjalan. *Product backlog* khusus untuk pengembangan modul *e-Participation* selama penelitian dilakukan dapat dilihat pada Tabel 2.

Tabel 2 *Product backlog* aplikasi *mobile* Patriot Pangan *e-Participation*

No.	Fungsi	Prioritas	Tingkat Kesulitan	*Sprint* Ke-
1	Analisis Proses Bisnis e-*Participation*	Tinggi	Sedang	1
2	Analisis Aktor yang terlibat pada aplikasi e-*Participation*	Tinggi	Sedang	1
3	Autentikasi untuk masuk ke dalam aplikasi	Tinggi	Tinggi	1
4	Fungsi profil patriot	Tinggi	Tinggi	1
5	Fungsi ubah profil patriot	Sedang	Sedang	2
6	Fungsi verifikasi patriot	Sedang	Tinggi	2
7	Fungsi input data kerawanan pangan (Bagian 1)	Tinggi	Sedang	2
8	Fungsi input data kerawanan pangan (Bagian 2)	Tinggi	Sedang	3
9	Melihat daftar keluarga tanggungan patriot	Tinggi	Sedang	3
10	Mode *offline* untuk input data kerawanan pangan	Tinggi	Tinggi	4
11	Label tenggat waktu pelaporan	Sedang	Sedang	4
12	Fungsi tambah foto dan deskripsi laporan	Tinggi	Tinggi	5
13	Fungsi *reward* untuk patriot	Sedang	Sedang	5
14	Fungsi visualisasi data kerawanan pangan	Tinggi	Sedang	5

Sprint 1

Backlog yang dikerjakan pada *sprint* 1 adalah analisis proses bisnis modul *e-participation*, analisis aktor yang terlibat, pembuatan fungsi autentikasi pengguna dengan menggunakan akun Google, dan pembuatan halaman profil patriot. Analisis proses bisnis dan aktor yang terlibat dilakukan dengan mengadakan diskusi yang melibatkan *product owner* dan tim *development* lainnya. Hasil dari proses analisis dan diskusi tersebut dapat disederhanakan menjadi *use case diagram* yang dapat dilihat pada Gambar 2.



Gambar 2 *Use case diagram e-Participation*.

Beberapa *requirement* utama yang diberikan oleh *product owner* adalah pengguna yang akan menggunakan aplikasi harus terverifikasi dengan data kependudukan yang dapat dipertanggungjawabkan, sehingga memperkecil kemungkinan penyalahgunaan sistem. Selain itu pengguna juga diharapkan dapat mengubah informasi pribadi seperti tempat tinggal dengan

pertimbangan kemungkinan pengguna pindah domisili ke daerah lain. Hal lain yang disampaikan oleh *product owner* adalah aplikasi mampu memberikan informasi statistik mengenai pelaporan yang telah dilakukan oleh pengguna dan aplikasi menyediakan sebuah sistem *reward* untuk pengguna.

Pengguna yang masuk ke dalam aplikasi harus *login* atau registrasi menggunakan akun Google sehingga akan mempermudah proses autentikasi. Apabila pengguna baru terdaftar di sistem, pengguna akan diminta untuk memasukkan data pribadi seperti alamat, jenis kelamin, dan alamat pengguna. Data patriot kemudian akan dimasukan ke dalam kelas model yang diberi nama UserModel yang dapat dilihat di Gambar 3. *Library* Volley versi 1.1.1 digunakan untuk menangani seluruh koneksi ke server *back-end* melalui API. *Library* ini dipilih karena mudah penggunaannya serta telah teroptimisasi sehingga dapat berjalan secara *asynchronous* di *background* aplikasi. Gambar 4 adalah potongan program untuk membuat *request body* dengan bantuan *library* Volley. Halaman profil patriot dibuat agar pengguna dapat melihat data umum pengguna dan ringkasan singkat aktivitas yang telah dilakukan selama menggunakan aplikasi. Gambar 5 adalah potongan kode program untuk membuat dan mengirimkan *request* ke server *back-end* dengan bantuan kelas JsonObjectRequest pada *library* Volley.

```
public class UserModel {
    private String id;
    private String googleId;
    private String id_token;
    private String name;
    private String email;
    private String birthDate;
    private String phone;
    private String homePhone;
    private String gender;
    private String provinsi_id;
    private String kabupaten_id;
    private String kecamatan_id;
    private String desa_id;
    private String provinsiName;
    private String kabupatenName;
    private String kecamatanName;
    private String desaName;
    private String jalan;
    private String jenisTempatTinggal;
    private Boolean isVerified;
    private String photoProfileUrl;
```

Gambar 3 Variabel di kelas UserModel.

```
public void registerPatriotAccount(UserModel model,
                                   final SingleDataConnectionListener<JSONObject> listener
                                   ){
    final String TAG = "ConnectionManager";
    String url = ConnectionConstant.baseUrl + ConnectionConstant.URL_REGISTER_PATRIOT;

    JSONObject postData = new JSONObject();
    try {
        postData.put( name: "id_token", model.getId_token());
        postData.put( name: "name", model.getName());
        postData.put( name: "birthDate", model.getBirthDate());
        postData.put( name: "gender", model.getGender());
        postData.put( name: "phone", model.getPhone());
        postData.put( name: "homePhone", model.getHomePhone());
        postData.put( name: "provinsi", model.getProvinsi_id());
        postData.put( name: "kabupaten", model.getKabupaten_id());
        postData.put( name: "kecamatan", model.getKecamatan_id());
        postData.put( name: "desa", model.getDesa_id());
        postData.put( name: "jalan", model.getJalan());
        postData.put( name: "jenisTempatTinggal", model.getJenisTempatTinggal());
    } catch (JSONException e) {
```

Gambar 4 Potongan program pembuatan *request body* untuk *Volley*.

```
JsonObjectRequest request = new JsonObjectRequest(Request.Method.POST, url,
        postData, new Response.Listener<JSONObject>() {
    @Override
    public void onResponse(JSONObject response) {
            listener.onSuccess(response, successMessage: "Successfully Registered Account");
    }
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {
        Log.d(TAG, msg: "onErrorResponse: Failed to request to server");
        listener.onFailed(Constant.REQUESTCODE_REGISTERFAILED, failedMessage: "failed to connect");
    }
}){
    @Override
    public String getBodyContentType() { return "application/json; charset=UTF-8"; }
};
requestQueue.add(request);
```

Gambar 5 Potongan kode program proses *request* untuk *register* ke server *back-end* dengan bantuan *library* Volley.

Pengujian yang dilakukan pada *sprint* ini dilakukan dengan metode *black box testing.* Hasil pengujian adalah seluruh fungsi dapat dijalankan dengan tanpa kendala, namun pengujian lebih mendalam akan diserahkan ke *product owner.*

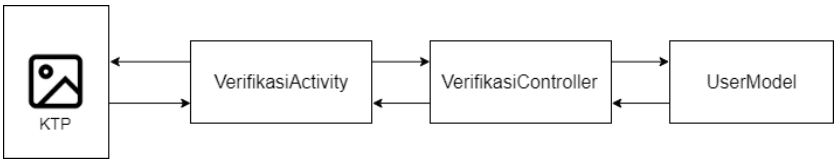
Sprint 2

Sprint backlog yang dihasilkan setelah berdiskusi dengan tim *scrum* pada saat *sprint planning* adalah pembuatan fitur *edit* profil patriot, verifikasi patriot, dan pengerjaan fitur *input*

data kerawanan pangan. Menurut *product owner*, pengubahan profil patriot dibuat agar patriot dapat mengubah lokasi patriot, dan penggantian foto profil melalui galeri atau mengambil foto baru. Fitur verifikasi patriot dibuat untuk memastikan bahwa patriot telah terverifikasi oleh sistem dan memperkecil kemungkinan pemalsuan data kerawanan pangan untuk tujuan tertentu. Verifikasi patriot dilakukan dengan membuat halaman khusus dimana patriot harus mengunggah foto KTP dan *selfie* patriot dengan memegang KTP dirinya.

Pada *sprint* ini, pola arsitektur MVC (*Model-View-Controller*) baru diterapkan ke dalam penelitian. Menurut Sokolova (2011), penerapan MVC ke dalam pengembangan aplikasi Android dapat berakibat pada meningkatnya pemeliharaan dan kinerja dari aplikasi. Penerapan MVC dilakukan dengan cara memisahkan *business logic* dengan *presentation logic*.

Kelas *activity* dari suatu halaman akan dijadikan perantara antara *View* yang berupa komponen tampilan seperti tombol, dan *Controller* yang mengolah semua *business logic* yang dilakukan di halaman tersebut seperti contohnya mengunggah foto KTP ke server. Ketika tombol unggah ditekan di *View*, kelas *activity* akan mendeteksi interaksi pengguna, kemudian akan memanggil *method* di kelas *Controller* untuk melakukan proses pengunggahan. Apabila proses pengunggahan selesai dilakukan, tautan yang menyimpan foto KTP di server akan disimpan ke dalam *Model* kemudian *Model* tersebut akan disimpan ke dalam penyimpanan lokal. Ilustrasi penerapan MVC pada fungsi verifikasi pengguna dapat dilihat di Gambar 6. Berdasarkan pengujian yang dilakukan dengan metode *black box testing*, diperlukan sedikit perbaikan untuk fitur verifikasi agar tetap dapat menampilkan foto KTP dan *selfie* apabila pengguna *logout* dan *login* kembali ke aplikasi.



Gambar 6 Ilustrasi penerapan MVC pada halaman verifikasi pengguna.

Sprint 3

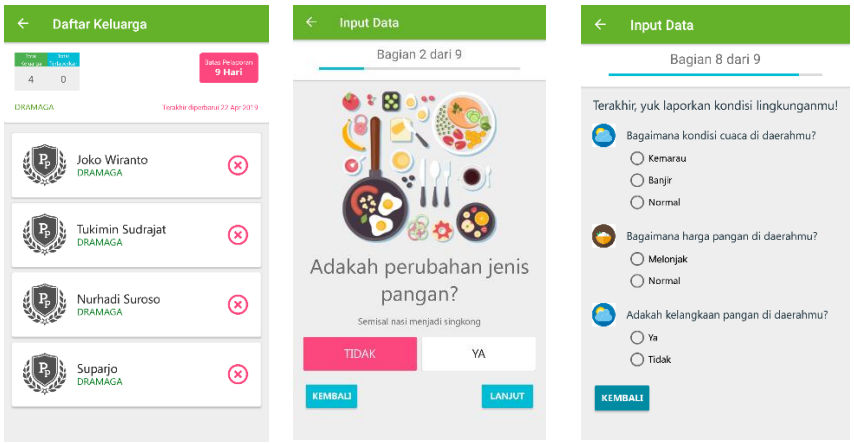
Sprint backlog yang ditentukan adalah melanjutkan pembuatan fitur *input* data kerawanan pangan, dan pembuatan halaman daftar keluarga yang ditanggung oleh patriot. *Sprint* ini fokus untuk menyelesaikan fungsi utama dari proses bisnis sistem Patriot Pangan, yaitu sistem untuk melaporkan kerawanan pangan yang terjadi di sekitar pengguna. Fitur *input* kerawanan pangan terdiri dari tujuh pertanyaan utama, dan pertanyaan tambahan mengenai kondisi cuaca, harga pangan, dan kelangkaan pangan di daerah tersebut. Semakin banyak indikator terpenuhi, semakin parah juga tingkat kerawanan pangan yang terjadi. Daftar pertanyaan utama yang menjadi indikator kerawanan pangan dapat dilihat di Tabel 3. Hasil tampilan input data kerawanan pangan dapat dilihat di Gambar 7.

Tabel 3 Pertanyaan utama yang menjadi indikator kerawanan pangan

No	Indikator kerawanan pangan	Tingkat kerawanan
1	Adanya kekhawatiran kekurangan pangan (ya/tidak)	Rawan 1/ringan
2	Adanya perubahan jenis pangan (ya/tidak)	Rawan 2
3	Adanya penurunan frekuensi dan/atau jumlah konsumsi pangan orang dewasa (ya/tidak)	Rawan 3
4	Adanya penurunan frekuensi dan/atau jumlah konsumsi pangan balita (ya/tidak)	Rawan 4
5	Mengalami tidak makan dalam satu hari (ya/tidak)	Rawan 4
6	Adanya penurunan berat badan balita dalam dua bulan terakhir karena kurang makan (ya/tidak)	Rawan 5/berat
7	Adanya penurunan berat badan orang dewasa karena kurang makan (ya/tidak)	Rawan 5/berat

Kemudian pembuatan halaman daftar keluarga tanggungan menjadi *backlog* berikutnya. Fitur ini akan menampilkan daftar keluarga yang ditugaskan kepada patriot oleh

admin sistem Patriot Pangan untuk dilaporkan keadaan pangan mereka. Untuk menampilkan data keluarga, aplikasi akan membuat *request* ke server untuk mendapatkan daftar keluarga berdasarkan *id* patriot. Server kemudian akan mengembalikan data dalam bentuk JSON yang berisi *array of object* data keluarga seperti contohnya yang dapat dilihat di Gambar 8. Kemudian setelah mendapatkan data keluarga, JSON data keluarga akan di-*parsing* sehingga dapat dimasukkan ke dalam model yang diberi nama KeluargaModel yang akan disimpan ke dalam penyimpanan lokal di aplikasi. Potongan program untuk *parsing* objek JSON dan memasukkan data keluarga ke dalam model KeluargaModel dapat dilihat di Gambar 9.



Gambar 7 Tampilan fungsi lapor kerawanan pangan dan daftar keluarga tanggungan.


```
{
    "address": {
        "provinsiId": {
            "_id": "5c6588cfa7f9211990ab06bd",
            "provinsi": "BANTEN"
        },
        "kabupatenId": {
            "_id": "5c6589af483b7f29d8a6061c",
            "kabupaten": "KABUPATEN SERANG"
        },
        "kecamatanId": {
            "_id": "5c658b421f55ab1cc450abbf",
            "kecamatan": "CIKANDE"
        },
        "desaId": {
            "_id": "5c658e0663a9a534a03773cb",
            "desa": "CIKANDE"
        },
        "jalan": "Pondok Indah 6A"
    },
    "patriotId": "5c77978d4d8b4721a955ef67",
    "isSejahtera": "false",
    "_id": "5ca76b0ef2a521398b2a3399",
    "namaKRT": "Jones Sebastian",
    "pekerjaanKRT": "Caleg",
    "phone": "0896777777",
    "__v": 0,
    "reported_at": "2019-04-05T16:18:40.531Z"
},
```


Gambar 8 Contoh data keluarga yang didapatkan dari server dalam bentuk JSON.

```
for (int i  = 0; i<keluargaArray.length();i++){
    JSONObject keluargaObject = keluargaArray.getJSONObject(i);
    JSONObject addressObject = keluargaObject.getJSONObject("address");
    JSONObject provinsiObject = addressObject.getJSONObject("provinsiId");
    JSONObject kabupatenObject = addressObject.getJSONObject("kabupatenId");
    JSONObject kecamatanObject = addressObject.getJSONObject("kecamatanId");
    JSONObject desaObject = addressObject.getJSONObject("desaId");
    String id = keluargaObject.getString( name: "_id");
    KeluargaModel model = new KeluargaModel(id);

    model.setNamaKepalaKeluarga(keluargaObject.getString( name: "namaKRT"));
    model.setPekerjaanKepalaKeluarga(keluargaObject.getString( name: "pekerjaanKRT"));
    model.setPhone(keluargaObject.getString( name: "phone"));
    model.setProvinsiName(provinsiObject.getString( name: "provinsi"));
    model.setKabupatenName(kabupatenObject.getString( name: "kabupaten"));
    model.setKecamatanName(kecamatanObject.getString( name: "kecamatan"));
    model.setDesaName(desaObject.getString( name: "desa"));
    model.setJalan(addressObject.getString( name: "jalan"));
```

Gambar 9 Potongan kode program untuk menyimpan data keluarga ke model keluarga.

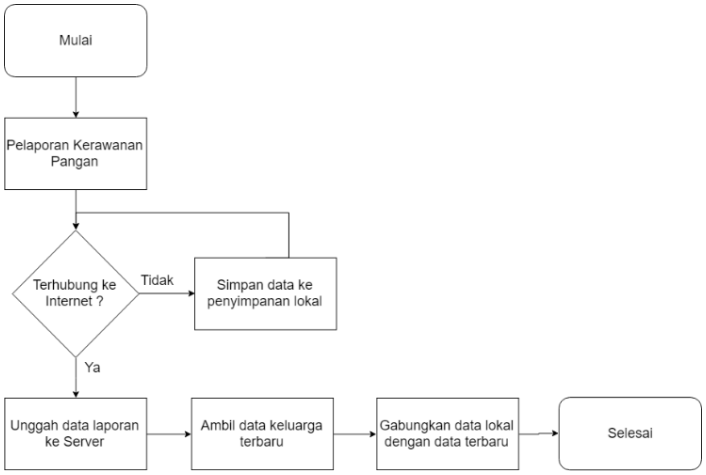
Setelah dilakukan pengujian dengan *black box testing*, diketahui bahwa penggunaan memori ponsel untuk membuka fungsi ini cukup besar. Penyebab hal ini terjadi adalah ukuran dan resolusi gambar yang cukup besar dan belum optimal untuk dimunculkan di aplikasi. Setelah melakukan beberapa uji coba dengan ukuran gambar yang berbeda, aplikasi dapat berjalan dengan performa yang baik selama ukuran gambar memiliki resolusi tidak lebih dari 400x400 piksel untuk gambar yang berukuran besar dan ukuran gambar tidak lebih dari 100 *kilobytes* untuk satu buah gambar. Kendala ini kemudian dilaporkan ke pihak pembuat aset gambar untuk mengubah ukuran dan resolusi dari gambar yang digunakan.

Sprint 4

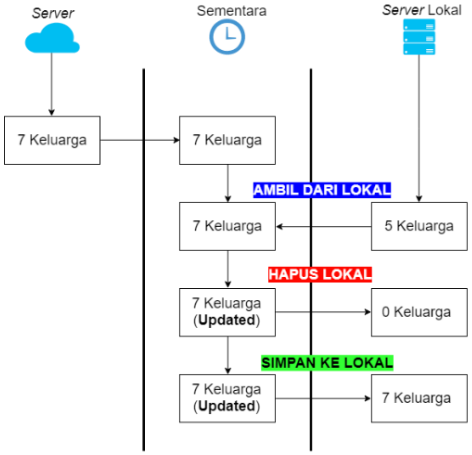
Sprint backlog yang dibuat setelah *sprint planning* adalah pada *sprint* ini penelitian akan difokuskan untuk mengembangkan mode luring untuk fitur *input* kerawanan pangan, dan pembuatan label tenggat waktu pelaporan terakhir. Hal tersebut didasari atas fakta bahwa tidak semua tempat akan terjangkau oleh jaringan internet untuk melakukan proses *input* data

kerawanan pangan, sehingga fungsi tersebut harus didukung dengan mode luring. Mode luring akan aktif setelah pengguna mengisi serluruh pertanyaan yang tersedia dan menekan tombol selesai namun server tidak dapat dijangkau karena ketidaktersediaan jaringan internet atau server yang sedang tidak berfungsi. Data laporan kemudian akan disimpan ke dalam basis data lokal yang tersedia, yaitu SQLite. Untuk membuat basisdata SQLite di aplikasi, penelitian ini memanfaatkan *library* Room versi 1.1.1 yang telah direkomendasikan oleh Google sehingga pembuatan basis data lokal dapat dilakukan dengan mudah dan optimal. Ilustrasi skema penerapan mode luring berbentuk diagram alir dapat dilihat di Gambar 10. Pada diagram alir dapat dilihat bahwa terdapat sebuah proses untuk menggabungkan data hasil pengambilan data keluarga terbaru dari server setelah pelaporan sukes dengan data lokal yang tersimpan di basisdata lokal. Proses penggabungan ini dilakukan agar data keluarga di basisdata lokal selaras dengan data yang tersedia di server.

Proses penggabungan data harus dilakukan dengan baik agar tidak ada data yang hilang pada saat proses penyelarasan dilakukan. Langkah yang dilakukan adalah, setelah mendapatkan data terbaru dari server, data tersebut akan dimasukkan ke dalam sebuah variabel baru untuk tempat penampungan sementara. Kemudian data lokal akan dibandingkan dengan data sementara tersebut. Apabila terdapat keluarga yang sama, maka keluarga itu akan diselaraskan datanya dengan mengambil data penting di keluarga lokal dan digabungkan ke data sementara. Dengan begitu data yang terdapat di data sementara merupakan data terbaru yang telah selaras dengan data lokal. Ilustrasi proses penggabungan data dapat dilihat pada Gambar 11. Hasil pengujian yang dilakukan di *sprint* ini didapatkan bahwa fungsi *input* data kerawanan pangan dapat berjalan tanpa jaringan internet sehingga fungsi ini berhasil diimplementasikan.



Gambar 10 Diagram alir penerapan skema luring untuk input data kerawanan pangan.



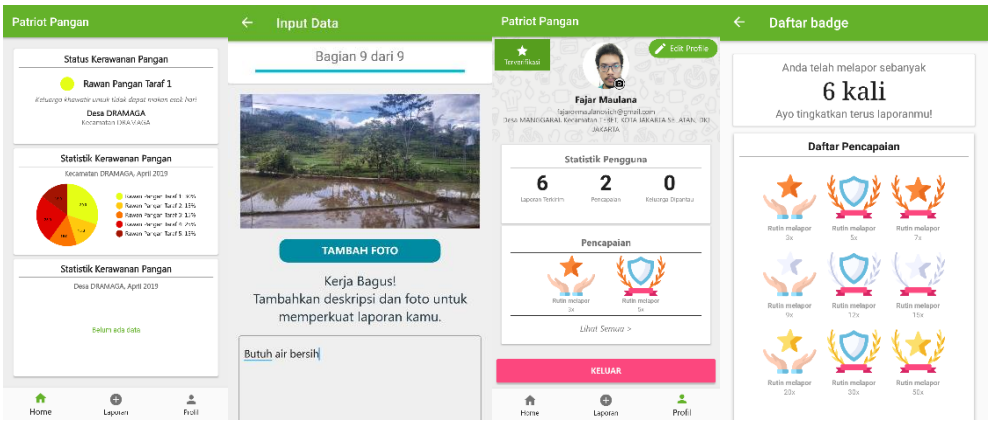
Gambar 11 Ilustrasi penggabungan data terbaru dengan data lokal.

Sprint 5

Sprint ini merupakan *sprint* terakhir dari rangkaian penelitian yang dilakukan. *Sprint backlog* yang ditentukan untuk *sprint* ini adalah pengerjaan fitur penambahan foto dan deskripsi untuk laporan kerawanan pangan, fitur *reward*, dan visualisasi data kerawanan pangan. Fitur penambahan foto dan deskripsi bertujuan untuk menambah kredibilitas setiap laporan yang dilakukan. Fitur *reward* dibuat untuk memberikan penghargaan kepada patriot atas laporan yang telah dilakukannya dengan harapan patriot semakin semangat dalam membuat laporan lainnya. Fitur *reward* tersendiri akan memberikan *badge* yang dapat dikoleksi oleh patriot ketika patriot berhasil mengunggah laporan ke server sebanyak 3 kali, 5 kali, 7 kali, 9 kali, 12 kali, 15 kali, 20 kali, 30 kali, dan 50 kali.

Badge tersebut dapat dilihat di halaman profil patriot. Dan yang terakhir adalah fungsi visualisasi data kerawanan pangan yang akan ditampilkan menjadi statistik di halaman *home* patriot. Fungsi visualisasi pada awalnya akan dibuat ke dalam bentuk peta, namun berdasarkan hasil diskusi antara pihak pengembang dan pihak *product owner*, disimpulkan bahwa

visualisasi ke dalam bentuk peta kerawanan pangan masih terlalu sulit dan kompleks untuk dikembangkan. Hasil tampilan halaman visualisasi data, tambah foto laporan, dan halaman *reward* untuk patriot dapat dilihat di Gambar 12.



Gambar 12 Tampilan halaman visualisasi data, tambah foto laporan, dan *reward* untuk patriot.

SIMPULAN

Kesimpulan yang didapatkan dari penelitian ini adalah, pengembangan aplikasi *mobile* modul *e-Participation* Patriot Pangan berhasil dilakukan selama lima *sprint* yang telah dilaksanakan. Total *product backlog* yang telah dikerjakan adalah sebanyak 14 *backlog* dengan perincian empat *backlog* di *sprint* pertama, tiga *backlog* di *sprint* kedua, dua *backlog* di *sprint* ketiga, dua *backlog* di *sprint* keempat, dan tiga *backlog* di *sprint* kelima. Fungsi utama dari aplikasi *mobile* ini yaitu *input* data kerawanan pangan berhasil dibuat dan penerapan mode luring untuk fungsi tersebut juga telah berhasil diimplementasikan dengan menerapkan pola arsitektur MVC.

Penelitian berikutnya dapat dilakukan untuk mengembangkan visualisasi data kerawanan pangan dengan bentuk peta interaktif yang memuat berbagai data kerawanan pangan berdasarkan daerah-daerah yang ada di Indonesia. Pengembangan peta kerawanan pangan dapat dilakukan dengan membuat server web khusus untuk mengolah *shapefiles* dari peta Indonesia yang dapat dimodifikasi untuk menampilkan berbagai informasi mengenai kerawanan pangan yang terjadi di Indonesia. Jika server web tersebut tersedia, aplikasi hanya perlu menampilkan *webview* dari server web tersebut sehingga penggunaan sumberdaya ponsel menjadi lebih hemat. Saran berikutnya adalah pengembangan aplikasi dapat dibuat untuk versi Android di bawah versi 5.0 Lollipop sehingga dapat menjangkau lebih banyak pengguna, dan pembuatan fungsi melihat *track record* dari laporan yang dimasukkan agar pengguna dapat melihat status laporan yang telah dibuat.

DAFTAR PUSTAKA

[APJII] Asosiasi Penyelenggara Jasa Internet Indonesia. 2018. Buletin APJII [Internet]. [diunduh 2019 Apr 23]; ed 23:1 https://apjii.or.id/downfile/file/BULETINAPJIIEDISI23April2018.pdf

Asfarian A, Putra RP, Panatagama AP, Nurhadryani Y, Ramadhan DA. 2020. E-Initiative for Food Security: Design of Mobile Crowdfunding Platform to Reduce Food Insecurity in Indonesia. Di dalam: *2020 8th International Conference on Information and Communication Technology (ICoICT)*. Jun 24. Yogyakarta, Indonesia. Hlm 1-5. IEEE.

Banjac M. 2017. E-Participation as a technology of citizenship. *Teorij Praks*. 54(1):73-91, 188. doi:URN:NBN:SI:DOC-7HHJ0C3N.

[FAO] Food and Agriculture Organization. 2010. *The State of Food Insecurity in the World: Addressing Food Insecurity in Protracted Crises*. Roma (IT): FAO.

Garcia-Molina H, Joglekar M, Marcus A, Parameswaran A, dan Verroios V. 2016. Challenges in data crowdsourcing. *IEEE Transact Knowled Dat Eng*. 28(4):901-911.

Kanhere SS. 2011. Participatory sensing: Crowdsourcing data from mobile smartphones in urban spaces. Di dalam: *IEEE 12th International Conference on Mobile Data Management;* Lulea, Swedia, 2011 Jun 6-9. Washington (US): IEEE. hlm 3-6.

Lamabelawa YRG. 2006. Analisis sistem kewaspadaan pangan dan gizi (SKPG) dalam mengatasi masalah gizi buruk di Kabupaten Lembata Propinsi Nusa Tenggara Timur [disertasi]. Semarang (ID): Universitas Diponegoro.

Macintosh A. 2004. Characterizing e-Participation in policy-making. Di dalam: *Proceedings of the 37th Annual Hawaii International Conference on System Sciences*; Hawaii, Amerika Serikat, 2004 Jan 5-8. Washington (US): IEEE.

Nurhadryani Y, Ramadhan W, Asfarian A. 2020. Development and optimization of NoSQL database in food insecurity early warning system based on local community participation. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*. 5(2):161-8.

Panatagama AP, Nurhadryani Y, Asfarian A. 2019. Analysis and Design of Patriot Pangan: Towards Electronic Participation and Initiative Platform to Help Reduce Food Insecurity in Indonesia. Di dalam *2019 IEEE R10 Humanitarian Technology Conference (R10-HTC)* 2019 Nov 12. Depok, Indonesia. Hlm. 159-164. IEEE.

Pressman RS. 2010. *Software Engineering: A Practitioner's Approach Ed ke-7*. New York (US): McGraw-Hill.

Schwaber K, Sutherland J. 2017. The scrum guide™ [Internet]. [diunduh 2018 Des 29]. https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf.

Sokolova K, Lemercier M, Garcia L. 2013. Android passive MVC: a novel architecture model for the android application development. Di dalam: *Fifth International Conferences on Pervasive Patterns and Applications;* Valencia, Spanyol, 2013 Mei 27 – Jun 1. Wilmington (US): IARIA. hlm 7-12.