

## Pengembangan *Back End* Berbasis REST API pada Sistem E-Partisipasi dan E-Inisiatif Patriot Pangan

### *Development of Back End-Based REST APIs on Patriot Pangan E-Participation and E-initiative Systems*

ADAM FIRDAUS<sup>1\*</sup>, DEAN APRIANA RAMADHAN<sup>1</sup>

#### Abstrak

Kelaparan yang terjadi di Indonesia pada tahun 2018 termasuk ke dalam kategori serius. Sistem Kerawanan Pangan dan Gizi yang dimiliki oleh Badan Kerawanan Pangan masih belum cukup cepat untuk mendeteksi kerawanan pangan yang terjadi di masyarakat. Oleh karena itu, dibuatlah sebuah sistem deteksi dini kerawanan pangan bernama Patriot Pangan. Sistem ini mengandalkan partisipasi aktif dari masyarakat untuk melaporkan kondisi keluarga rawan pangan. Pengembangan sistem ini dilakukan secara tim dengan menggunakan metode pengembangan *Scrum* dengan total *sprint* sebanyak lima kali. Setiap *sprint* dilakukan selama dua minggu. Artefak yang digunakan meliputi *product backlog*, *sprint backlog*, *use case diagram*, dan *class diagram*. Penelitian ini telah berhasil mengembangkan *back end* yang berbasis REST API dengan bahasa pemrograman javascript dan *framework* Express.js. API menerima masukan dan memberikan keluaran dalam bentuk *Javascript Object Notation*. API yang dibuat kemudian digunakan oleh aplikasi E-Partisipasi, E-Inisiatif, dan web patriot pangan. Aplikasi E-Partisipasi berfokus pada pembuatan laporan kerawanan pangan yang kemudian dikumpulkan dan dibuat agregasi kerawanan pangan setiap bulan. Aplikasi E-Inisiatif berfokus pada proses *campaign* dan pengumpulan donasi berupa uang dan barang. Sedangkan web patriot pangan digunakan oleh administrator patriot pangan untuk mengontrol seluruh sistem dan menampilkan hasil agregasi kerawanan pangan.

Kata Kunci: *back end*, kerawanan pangan, REST API, *scrum*, sistem informasi.

#### Abstract

*Hunger occurring in Indonesia in 2018 falls into the serious category. The Food and Nutrition Insecurity system possessed by the Food Insecurity Agency is still not fast enough to detect food insecurity. Therefore, an early detection system for food insecurity was made called the Patriot Pangan. This system relies on active participation from the public to report food insecure family conditions. The development of this system is carried out in teams using the Scrum development method with a total of five sprints. This research has succeeded in developing a back end based on REST API with javascript programming language and Express.js framework. The API accepts input and provides output in the form of Javascript Object Notation. The API created is used by E-Participation, E-Initiative, and Patriot Pangan web applications. E-participation application focuses on making reports on food insecurity which are then collected and made aggregate food insecurity every month. E-initiative application focuses on the campaign process and collecting donations in the form of money and goods. While the Patriot Pangan web is used by Patriot Pangan administrator to control the entire system and display the aggregation results of food insecurity.*

Keywords: *back end*, food insecurity, information systems, REST API, *scrum*.

## PENDAHULUAN

Menurut Global Hunger Index pada tahun 2018, Indonesia menempati peringkat ke-73 dari 119 negara untuk tingkat kelaparan dan masuk ke dalam kategori serius. Hal ini dapat menjadi ancaman serius bagi potensi produktivitas generasi muda Indonesia. Hal tersebut diperkuat oleh hasil penelitian yang menyatakan bahwa sebanyak 20.3 juta penduduk Indonesia mengalami kelaparan pada tahun 2014-2016 (FAO, IFAD, UNICEF, WFP, dan WHO 2017). Masalah kelaparan ini berdampak pada tingginya prevalensi balita pendek

<sup>1</sup>Departemen Ilmu Komputer, FMIPA, Institut Pertanian Bogor.

\*Penulis Korespondensi: Tel/Faks: 0251-8625584; Surel: adam\_22@apps.ipb.ac.id

(*stunting*) dan balita kurus (*wasting*) yang melewati batas yang telah ditentukan oleh World Health Organization (WHO). Prevalensi balita pendek (*stunting*) di Indonesia mencapai 29% dengan batas standar WHO yaitu 20% (Kemenkes 2017). Dampak jangka panjang dari *stunting* adalah gangguan perkembangan otak, IQ rendah, sistem imun yang lemah, serta risiko yang lebih besar terhadap penyakit serius seperti diabetes dan kanker (Martins *et al.* 2011).

Indonesia telah memiliki Badan Ketahanan Pangan (BKP) yang bertugas untuk melaksanakan pengkajian, pengembangan dan koordinasi di bidang pemantapan ketahanan pangan. BKP sendiri memiliki Sistem Kewaspadaan Pangan dan Gizi (SKPG) sebagai alat deteksi dini terhadap situasi pangan dan gizi suatu wilayah. Namun demikian, SKPG belum berjalan secara efektif karena pengumpulan dan pengolahan data secara manual membutuhkan waktu yang lama. Akibatnya, deteksi dini kelaparan pada kejadian luar biasa seperti di Asmat pada tahun 2017 tidak dapat berjalan sesuai yang diharapkan.

Pada lain pihak, kemajuan teknologi saat ini telah banyak membantu berbagai aspek kehidupan masyarakat, salah satunya adalah penyebaran informasi melalui internet. Sebagai contoh, media seperti *kitabisa.com* memanfaatkan partisipasi bersama dari masyarakat untuk mengatasi suatu masalah dengan sistem *crowdfunding*. Dalam hal ini, bukan tidak mungkin bila pemerintah melibatkan partisipasi masyarakat dalam mengatasi masalah kerawanan pangan, maka kasus luar biasa seperti yang terjadi di Asmat dapat dicegah.

Pada penelitian sebelumnya yang dilakukan oleh Panatagama (2019) telah berhasil mengembangkan sebuah aplikasi yang berperan dalam pengumpulan data kerawanan pangan bernama Patriot Pangan. Namun masih dibutuhkan beberapa pengembangan untuk menyempurnakan aplikasi tersebut. Selain itu, partisipasi masyarakat pada aplikasi Patriot Pangan hanya sebatas pelaporan data saja. Sehingga selanjutnya akan dikembangkan pula fitur dalam aplikasi yang bertujuan untuk menarik partisipasi masyarakat dalam bentuk donasi, baik itu berupa uang ataupun makanan.

Pengembangan aplikasi ini dilakukan secara tim yang terdiri dari pengembang *back end*, *front-end*, aplikasi Android, basis data analis, dan UI/UX analis. Dalam hal ini, penulis bertugas untuk mengembangkan *back end* dari aplikasi Patriot Pangan. Oleh karena itu, penelitian ini hanya difokuskan pada pengembangan *back end* pada aplikasi Patriot Pangan.

## METODE

### Ruang Lingkup Penelitian

Penelitian ini mencakup pengembangan *back end* berbasis REST API yang keluarannya berupa teks dalam format *Javascript Object Notation* (JSON). *Back end* yang dikembangkan dapat digunakan untuk aplikasi Android dan *web front-end*. Fitur utama yang dikembangkan adalah modul E-Partisipasi dan E-Inisiatif.

### Arsitektur Sistem

Sistem ini terbagi menjadi *back end*, *web front-end*, dan aplikasi Android. Penelitian ini berfokus pada pengembangan *back end* berbasis REST API menggunakan Node.JS. Ilustrasi dari arsitektur sistem Patriot Pangan yang akan dikembangkan dapat dilihat pada Gambar 1.

### Tahapan Penelitian

Penelitian ini dilakukan dengan menerapkan metode pengembangan *scrum*. *Scrum team* terbagi atas *product owner*, *scrum master*, dan tim pengembang. *Product owner* bertanggung jawab dalam perencanaan fitur dari produk. *Scrum master* bertugas untuk memastikan seluruh anggota memahami teori, praktik, aturan, dan tata nilai dalam *scrum*. Tim pengembang bertugas untuk mengembangkan produk untuk mencapai *sprint goal*.

Metode *scrum* terdiri atas empat tahapan, yaitu *sprint planning*, *daily scrum*, *sprint review*, dan *sprint retrospective* (Schwaber dan Sutherland 2017). Tahap pertama merupakan *sprint planning*, pada tahap ini *scrum team* berkolaborasi melakukan *brainstorming* untuk menentukan *scrum goal* yang ingin dicapai. *Scrum goal* adalah objektif yang akan dicapai

selama *sprint* melalui implementasi *product backlog*. Objektif tersebut disampaikan oleh *product owner*. Tahap kedua adalah *daily scrum*. *Daily scrum* dilakukan oleh tim pengembang setiap hari dengan durasi paling lama adalah 15 menit (Schwaber dan Sutherland 2017). *Daily scrum* dapat dilakukan secara langsung dengan tatap muka maupun tidak langsung melalui aplikasi *chatting*.

Setelah melakukan *daily scrum* selama 14 hari, akan dilanjutkan dengan *sprint review* dengan durasi maksimal adalah empat jam (Schwaber dan Sutherland 2017). Pada tahap ini, tim pengembang akan menyampaikan dan mendemonstrasikan *increment* yang telah dikerjakan. Kemudian tim pengembang juga akan menyampaikan masalah yang terjadi sepanjang *sprint* dan bagaimana cara menyelesaikannya. Tahap terakhir adalah *sprint retrospective*, yaitu tahap saat tim pengembang mengevaluasi *sprint* yang telah dilakukan. Bila ada kesalahan dalam *sprint* sebelumnya, maka dapat dijadikan sebagai pelajaran dan terhindar dari kesalahan yang sama pada *sprint* selanjutnya (Schwaber dan Sutherland 2017).

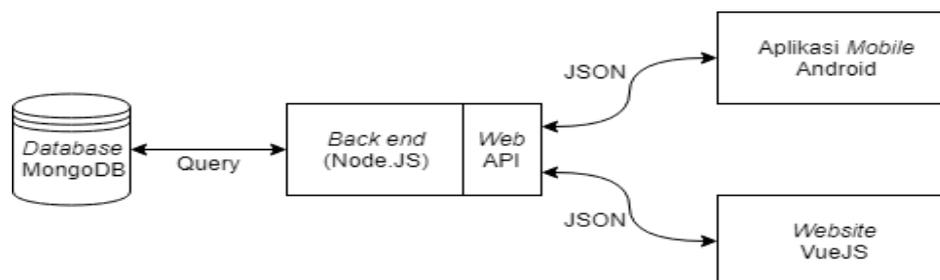
## HASIL DAN PEMBAHASAN

### Artefak

Sebelum proses *scrum* dimulai, *product owner* menjelaskan deskripsi sistem yang ingin dibuat menggunakan *use case diagram*. Pada *use case diagram* dapat digambarkan bahwa proses bisnis inti dari sistem patriot pangan terbagi menjadi dua, yaitu E-Inisiatif dan E-Partisipasi. E-Inisiatif merupakan sebuah wadah bagi masyarakat melakukan *crowdfunding* untuk membantu mengatasi suatu kasus kerawanan pangan. Sedangkan E-Partisipasi merupakan sebuah mekanisme untuk menciptakan *early warning system* dengan mengandalkan partisipasi masyarakat dalam mengawasi lingkungan sekitarnya. Adapun *use case diagram* dapat dilihat pada Gambar 2.

### *Sprint* 1

Fungsi pertama yang dikembangkan adalah autentikasi pengguna. Terdapat tiga pengguna yang dapat melakukan autentikasi, yaitu patriot, donatur, dan administrator. Untuk mempermudah proses pendaftaran, autentikasi bagi patriot dan donatur dapat dilakukan menggunakan akun Google. Sedangkan untuk administrator tetap menggunakan email dan kata sandi yang didaftarkan ke sistem.

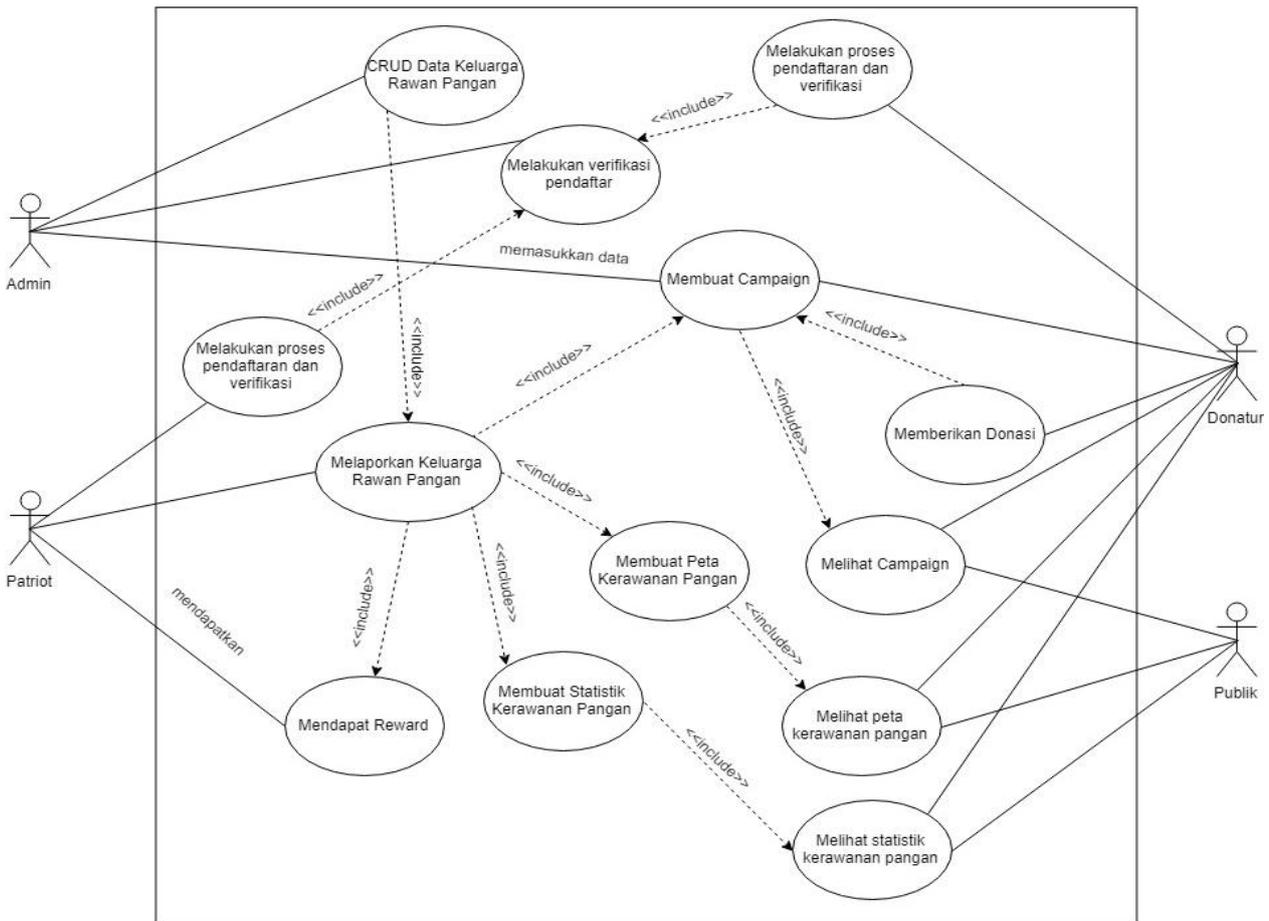


Gambar 1 Arsitektur sistem Patriot Pangan.

Proses autentikasi donatur dan patriot menggunakan metode Oauth2 dari Google. Metode ini membutuhkan *id\_token* yang didapatkan oleh aplikasi Android (Asfarian *et al.* 2020; Nurhadryani *et al.* 2020) melalui pengisian email dan kata sandi akun Google. Setelah *id\_token* diterima, server akan mengambil *payload* dan melakukan pengecekan ke basis data (Gambar 3). Akun yang telah terdaftar akan menerima token akses, sedangkan akun yang belum terdaftar harus melengkapi data yang dibutuhkan sesuai jenis pengguna. Detail jenis pengguna dapat dilihat pada Gambar 4.

Autentikasi yang digunakan untuk administrator tidak menggunakan metode Oauth2. Hal ini karena pembuatan akun administrator tidak terbuka untuk umum. Pembuatan akun administrator yang baru dilakukan oleh administrator yang telah ada sebelumnya.

Administrator dapat melakukan autentikasi menggunakan email dan kata sandi lalu menerima token akses.



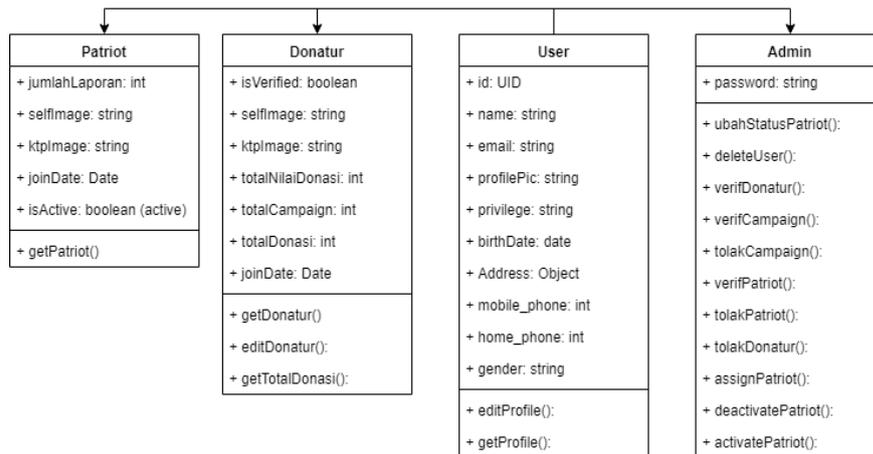
Gambar 2 Use case diagram sistem Patriot Pangan (E-Inisiatif & E-Partisipasi).

```
const { OAuth2Client } = require('google-auth-library');
module.exports.getGoogleUser = (code, GOOGLE_CLIENT_ID,
CLIENT_SECRET = '', REDIRECT_URI = '') => {
  var client = new OAuth2Client(GOOGLE_CLIENT_ID, CLIENT_SECRET, REDIRECT_URI);
  return client.verifyIdToken({
    idToken: code, audience: GOOGLE_CLIENT_ID
  })
  .then(login => {
    var payload = login.getPayload();
    var userid = payload['sub'];
    return { name: payload['name'], pic: payload['picture'],
      googleId: payload['sub'], email_verified: payload['email_verified'],
      email: payload['email'], payload };
  })
  .then(user => { return user })
  .catch(err => { console.log(err) return null })
}
```

Gambar 3 Pengambilan *payload* dari *id\_token*.

Token akses yang diberikan berbentuk JSON Web Token (JWT), yaitu standar terbuka berbasis JSON untuk membuat token akses yang mengandung sejumlah informasi. Informasi yang disimpan dalam JWT pada aplikasi ini berupa nama aplikasi, ID pengguna, masa berlaku, dan jenis kewenangan pengguna. Proses pembuatan token dapat dilihat pada Gambar 5.

Token akses digunakan sebagai pengidentifikasi bila ada *client* yang hendak menggunakan API. API hanya bisa diakses menggunakan akses token. Token hanya dapat digunakan untuk mengakses API yang kewenangannya terkandung dalam token tersebut. Sebagai contoh, token akses milik donatur tidak dapat digunakan untuk mengakses fungsi yang dimiliki administrator karena kewenangan yang dimiliki tidak tepat. Token dikirimkan melalui *request header authorization* dengan prefix 'Bearer'. Potongan kode dari proses verifikasi token dapat dilihat pada Gambar 6.



Gambar 4 Class diagram pengguna.

Fungsi lain yang dikembangkan adalah ubah data akun pengguna, unggah foto profil, dan input data keluarga miskin oleh administrator. Fungsi yang dibuat dalam bentuk API dapat diakses oleh *client* melalui URL. Metode akses yang tersedia antara lain adalah GET, POST, PUT, dan DELETE. GET digunakan untuk mengambil data, POST digunakan untuk mengirimkan data, PUT digunakan bila ada data yang akan diubah, dan DELETE digunakan untuk menghapus data. Sebagai contoh metode GET diterapkan pada API yang mengambil daftar patriot karena untuk mendapatkan data tersebut server tidak memerlukan masukan data. Contoh metode POST adalah pada API untuk menambahkan keluarga miskin. API ini membutuhkan masukan dalam bentuk JSON yang berisi alamat keluarga, nama, pekerjaan, dan nomor telepon dari kepala rumah tangga. Selanjutnya API akan memberikan keluaran yang berisi status *request* dan hasil dari data yang disimpan. Daftar API lain yang dihasilkan pada sprint ini dapat dilihat pada Tabel 1, sedangkan contoh penggunaan API dapat dilihat pada Gambar 7.

```

let signToken = (user) => {
  return JWT.sign({
    iss: 'Patriot Pangan',
    sub: user._id,
    iat: new Date().getTime(),
    exp: new Date().setDate(new Date().getDate() + 5),
    privilege: user.privilege
  }, JWT_SECRET);
}
  
```

Gambar 5 Pembuatan akses token.

```

let checkToken = (req, res, next) => {
  let token = req.headers['x-access-token'] || req.headers['authorization'];
  if (token.startsWith('Bearer ')) {
    token = token.slice(7, token.length);
  }

  if (token) {
    JWT.verify(token, config.JWT_SECRET, (err, decoded) => {
      if (err) {
        return res.json({
          success: false,
          message: 'Akses token tidak valid'
        });
      } else {
        req.decoded = decoded;
        next();
      }
    });
  } else {
    return res.json({
      success: false,
      message: 'Akses token tidak terdeteksi'
    });
  }
};
  
```

Gambar 6 Potongan kode verifikasi token akses.

**Sprint 2 dan 3**

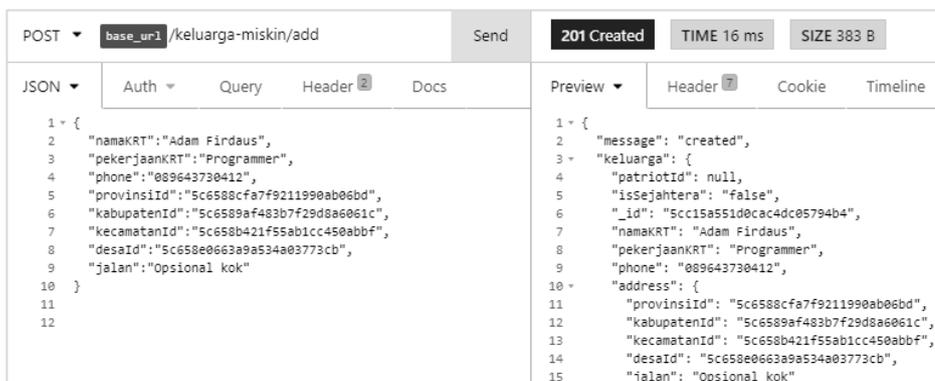
Pada *sprint* ini, fungsi pertama yang dibuat adalah verifikasi akun pengguna. Verifikasi akun pengguna dilakukan dengan mengirim foto diri dan KTP yang kemudian diterima oleh administrator dan diverifikasi manual oleh administrator. Jika verifikasi berhasil, donatur dapat melakukan pembuatan *campaign*, sedangkan patriot dapat melakukan pelaporan kerawanan pangan.

Fungsi selanjutnya yang dikembangkan adalah penugasan keluarga miskin ke patriot dan input laporan kerawanan pangan pada aplikasi E-Partisipasi. Penugasan patriot dilakukan oleh administrator berdasarkan alamat patriot. Seorang patriot dapat mengawasi sebanyak 10 keluarga di daerahnya. Patriot wajib membuat laporan kerawanan pangan setiap bulan. Laporan kerawanan pangan yang dibuat berupa menjawab 7 pertanyaan kondisi keluarga dan 3 pertanyaan kondisi lingkungan. Selain itu, patriot juga harus melampirkan deskripsi dan gambar keluarga miskin dalam laporannya. Potongan *class diagram* dapat dilihat pada Gambar 8.

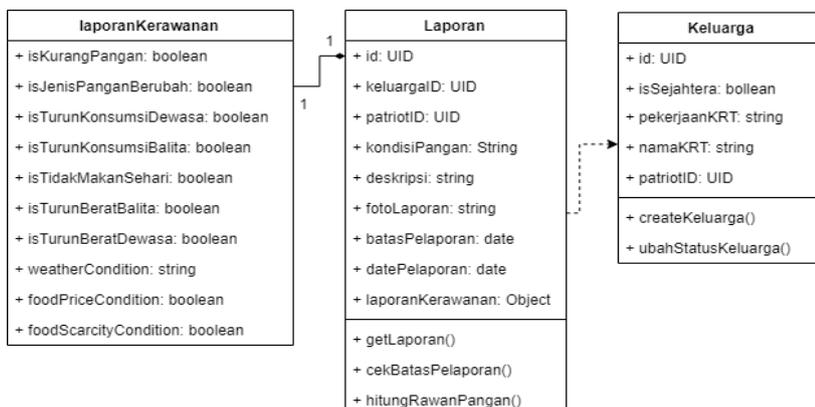
Fungsi selanjutnya adalah pembuatan dan verifikasi *campaign*. *Campaign* dibuat oleh donatur yang telah diverifikasi. Kemudian *campaign* tersebut diverifikasi oleh administrator. Jika lolos verifikasi maka *campaign* akan ditampilkan pada aplikasi E-Inisiatif dan donatur dapat melakukan donasi. Daftar atribut yang dibutuhkan dalam pembuatan *campaign* dapat dilihat pada Gambar 9.

Tabel 1 Daftar API *Sprint 1*

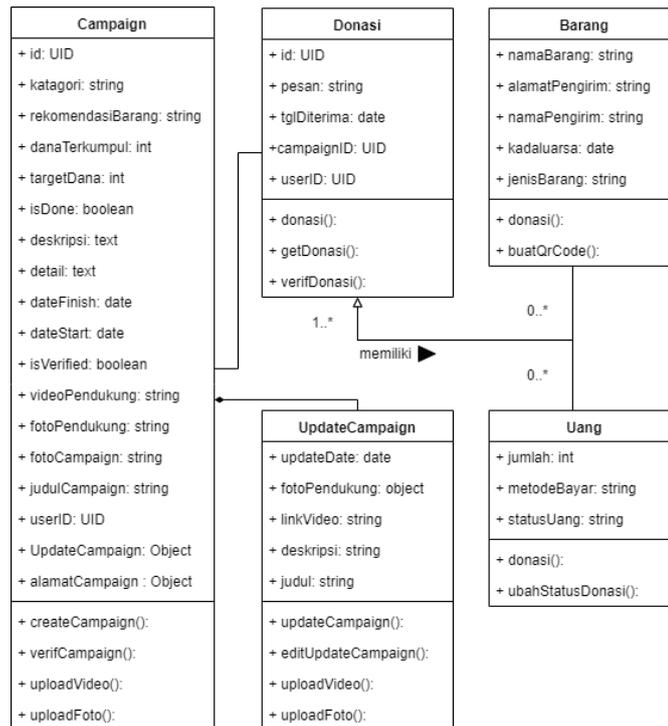
Method	URL	Keterangan
POST	/oauth/user-check/:privilege	Mengecek apakah pengguna sudah terdaftar
POST	/oauth/register/:privilege	Register donatur dan patriot
POST	/oauth/upgrade-to/:privilege	Menambahkan <i>privilege</i> pada akun
POST	/oauth/login/admin	Login administrator
POST	/keluarga-miskin/add	Menambahkan data keluarga miskin
PUT	/users/:privilege	Mengubah data pengguna
POST	/users/images	Unggah foto pengguna



Gambar 7 Contoh penggunaan API.



Gambar 8 Potongan *class diagram* untuk E-Partisipasi.



Gambar 9 Potongan Class Diagram untuk E-Inisiatif.

Donasi dapat dilakukan dalam dua bentuk, yaitu uang dan barang. Donasi uang dilakukan dengan mentransfer sejumlah uang yang ditambahkan tiga angka unik ke Bank yang telah disediakan. Sedangkan untuk donasi barang, donatur dapat menempelkan kode QR yang dicetak oleh aplikasi E-Inisiatif dan mengirimkan barang ke alamat tertentu. Ketika barang sampai, maka kode QR akan dipindai dan transaksi donasi barang akan terkonfirmasi. Contoh implementasi donasi barang dapat dilihat pada Gambar 10. Pada akhir *sprint*, seluruh fungsi berhasil dibuat dan API dapat digunakan oleh aplikasi E-Inisiatif dan E-Partisipasi. Daftar API dapat dilihat pada Tabel 2.

Tabel 2 Daftar API *Sprint 2* dan *3*

Method	URL	Keterangan
GET	/patriot/activate/:_id	Aktivasi akun patriot
GET	/donatur/verify/:_id	Verifikasi akun donatur
POST	/keluarga-miskin/assign-patriot	Penugasan keluarga miskin ke patriot
POST	/laporan/keluarga-miskin/create/:keluargaId	Pembuatan laporan rawan pangan
POST	/campaign/create	Membuat <i>campaign</i>
GET	/campaign	Menampilkan daftar <i>campaign</i>
GET	/campaign/verify/:_campaignId	Verifikasi <i>campaign</i> oleh administrator
POST	/donasi/add/transfer	Donasi uang
POST	/donasi/barang/create	Donasi barang
GET	/donasi/confirm/:donasiId	Konfirmasi donasi

### *Sprint 4* dan *5*

Sebagian API yang dihasilkan pada *sprint 4* dan *5* dapat dilihat pada Tabel 3. Fungsi pertama yang dikembangkan adalah fungsi untuk melakukan *update campaign*. *Update* yang dimaksud adalah memasukkan informasi tambahan mengenai kondisi terbaru dari suatu campaign sehingga donatur dapat melihat informasi tambahan tersebut. Informasi yang dapat disertakan pada *update campaign* dapat berupa waktu *update*, deskripsi, video, dan foto pendukung. Fungsi selanjutnya yang dikembangkan adalah pembatalan donasi secara otomatis bagi donasi uang yang telah melewati batas waktu. Fungsi ini tidak berbentuk API yang dapat diakses melalui sebuah URL, melainkan sebuah *cron job* yang otomatis berjalan di setiap 10 menit. Selain itu, fungsi ini juga akan mengirimkan notifikasi pembatalan donasi bila donatur sedang *online*. Gambar 11 merupakan potongan kode fungsi yang dijalankan oleh *cron job*.

```

donasiBarang: async (req, res, next) => {
  let { sub, privilege } = req.decoded;
  if (!privilege.includes("donatur")) {
    return res.status(400).json({message: "Unauthorized."})
  }
  let { campaignId, namaBarang, quantity, pesan } = req.value.body;
  console.log(namaBarang, quantity);
  let donasi = new Donasi({ userId: sub, campaignId, jenis: 'barang',
    pesan, barang: { isDelivered: false } });
  for (let index = 0; index < namaBarang.length; index++) {
    donasi.barang.data.push({ namaBarang: namaBarang[index],
      quantity: quantity[index]});
  }
  await donasi.save();
  return res.json({ message: "success", donasi });
}

```

Gambar 10 Implementasi Donasi Barang.

```

async overdueDonation() {
  let now = new Date();
  let donasis = await Donasi.find({$and:[{'uang.dueDate':{$lt:now}},
    {'uang.isPaid':{$eq:false}}]});
  .then((data) => {return data});
  for (let i = 0; i < donasis.length; i++) {
    let socketUser = this.socketUsers.find(x => x.userId === donasis[i].userId.toString()) || null;
    if (socketUser) {
      this.io.to(socketUser.id).emit('e-inisiatif-confirm-donation', {title: 'Donasi dibatalkan',
        message: 'Donasi anda senilai '+ donasis[i].uang.transfer.nominal
          + ' dibatalkan karena melewati batas waktu pembayaran.', isPaid: false
        });
    }
    await donasis[i].remove();
  }
}
}
cron.schedule('*/10 * * * *', async () => {
  cronJob.overdueDonation();
});

```

Gambar 11 Potongan kode fungsi overdueDonation dan cron job.

*Cron job* juga digunakan untuk menjalankan fungsi yang mengubah status *campaign* menjadi selesai bila *campaign* telah melewati batas akhir periode. Fungsi ini bertujuan agar *campaign* yang telah selesai tidak ditampilkan lagi pada daftar *campaign* yang ada pada aplikasi E-Inisiatif. Selain itu, *campaign* yang telah selesai tidak dapat menerima donasi lagi.

Fungsi notifikasi dibutuhkan untuk beragam *event*, antara lain notifikasi bila akun donatur telah diverifikasi oleh administrator, notifikasi bila donasi telah melewati batas waktu, dan notifikasi bila donasi telah diterima. Fungsi notifikasi dilakukan secara *real time* menggunakan *library* Socket.IO yang memungkinkan untuk melakukan komunikasi dua arah antara aplikasi klien dan server. Adapun contoh kode dan pesan notifikasi dapat dilihat pada Gambar 12.

```

let io = req.app.get('io');
let socketUsers = req.app.get('socketUsers');
let socketUser = socketUsers.find(x => x.userId === _id) || null;
if (socketUser) {
  io.to(socketUser.id).emit('e-inisiatif-verify-donatur', {
    title: 'Verifikasi Donatur', verificationStatus: 'verified',
    message: 'Akun anda telah diverifikasi sebagai donatur.'});
}

```

Gambar 12 Potongan kode notifikasi menggunakan Socket.IO.

Tabel 3 Daftar API *Sprint* 4 dan 5

URL	Keterangan
/campaign/update/:campaignId	Menambahkan <i>update</i> pada <i>campaign</i>
/campaign/update/uploadFoto/:campaignId/:updateCampaignId	Mengunggah foto <i>update campaign</i>
/agregasi-rawan-pangan/provinsi/current	Menampilkan agregasi kerawanan pangan terkini pada tingkat Provinsi
/agregasi-rawan-pangan/provinsi/current/id/:_id	Menampilkan agregasi kerawanan pangan terkini pada Kabupaten di suatu Provinsi
/agregasi-rawan-pangan/provinsi/history/id/:_id	Menampilkan riwayat agregasi kerawanan pangan suatu Provinsi
/agregasi-rawan-pangan/kabupaten/history/id/:_id	Menampilkan riwayat agregasi kerawanan pangan suatu Kabupaten

Fungsi terakhir yang dikembangkan adalah pembuatan agregasi laporan kerawanan pangan setiap daerah. Menurut Panatagama (2019) agregasi kerawanan pangan suatu daerah dapat diperoleh dari nilai modus tingkat kerawanan pangan yang terjadi pada daerah tersebut.

Sebagai contoh, bila pada suatu daerah terdapat 8 dari 10 keluarga yang memiliki tingkat kerawanan pangan 2, maka agregasi kerawanan pangan daerah tersebut adalah tingkat 2. Data agregasi disimpan secara bertingkat mulai dari desa, kecamatan, kabupaten, dan provinsi. Fungsi perhitungan agregasi ini dilakukan setiap bulan dan dijalankan oleh *cron job*. Pada akhir *sprint* 4 dan 5 seluruh fungsi berhasil dijalankan.

## SIMPULAN

Penelitian ini berhasil mengembangkan *back end* berbasis REST API untuk aplikasi E-Partisipasi dan E-Inisiatif patriot pangan menggunakan Node.JS dengan *framework* Express.Js dan basis data non-relasional MongoDB. Pengembangan dilakukan dengan menggunakan metode *scrum* sebanyak 5 *sprint* dan menghasilkan sejumlah 23 API yang sudah berhasil diuji. Sistem ini diharapkan dapat membantu untuk melakukan deteksi dini terhadap kasus rawan pangan dan mewadahi masyarakat untuk melakukan *crowdfunding* sehingga dapat mengatasi kasus rawan pangan dengan cepat. Namun proses penyumbangan uang melalui transfer bank masih dilakukan secara *dummy*. Diharapkan sistem ini dapat menggunakan *payment gateway* sesungguhnya sehingga masyarakat bisa melakukan donasi secara langsung. Selain itu, pada proses penerimaan donasi barang masih belum ada aplikasi klien yang menangani proses pemindaian kode QR dan mengubah status donasi barang menjadi diterima. Pada pengembangan selanjutnya diharapkan aplikasi klien yang bertugas untuk melakukan konfirmasi penerimaan barang dapat dikembangkan.

## DAFTAR PUSTAKA

- 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. In 2020 8th International Conference on Information and Communication Technology (ICoICT) 2020 Jun 24 (pp. 1-5). IEEE.
- [FAO, IFAD, UNICEF, WFP, WHO] Food and Agriculture Organization, International Fund for Agricultural Development, United Nations Children's Fund, World Food Programme, dan World Health Organization. 2017. The State of Food Security and Nutrition in the World 2017. Building resilience for peace and food security. Roma (IT): FAO.
- [GHI] Global Hunger Index. 2018. Global Hunger Index: Forced Migration and Hunger. Dublin (IE): GHI.
- [Kemenkes] Kementerian Kesehatan. 2017. Buku Saku Pemantauan Status Gizi Tahun 2017. Jakarta (ID): Kementerian Kesehatan.
- Martins VJB, Clemente APG, Florencio TMMT, Franco MDCP, Grillo LP, Martins PA, Santos CDL, Sawaya AL, Vieira MDFA. 2011. Long-Lasting Effects of Undernutrition. PMC [Internet]. [diunduh 2018 Nov 23]; 8(6):1817–1846. Tersedia pada: [www.ncbi.nlm.nih.gov/pmc/articles/PMC3137999](http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3137999).
- 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*. 2020 May 6:161-8.
- 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. In 2019 IEEE R10 Humanitarian Technology Conference (R10-HTC)(47129) 2019 Nov 12 (pp. 159-164). IEEE.
- Schwaber K, Sutherland J. 2017. The Scrum Guide™. [Internet]. [diunduh 2 Januari 2019]. Tersedia pada: <https://www.scrumguides.org/download.html>.