

Rancang Bangun dan Pemanfaatan Teknologi Barcode dalam Sistem Informasi Data Ternak di Desa Binaan LAZ Al-Azhar

Nurul Fajrin Ariyani, Navinda Meutia, Adhatus Solichah Ahmadiyah, Kelly Rossa Sungkono, Riyanarto Sarno, Abdul Munif, Shintami Chusnul Hidayati
Departemen Teknik Informatika, Institut Teknologi Sepuluh Nopember

Email:
nurulfajrin@if.its.ac.id

ABSTRAK

Para peternak umumnya meningkatkan kualitas hewan ternaknya dengan cara mengawinkan silang antara dua ras hewan ternak yang berbeda untuk didapatkan hasil peranakan yang lebih baik. Namun dalam perkawinan silang ini harus dicatat dengan jelas asal bibit dan keturunannya untuk menghindari kemungkinan peranakan yang cacat atau kurang baik. Peranakan cacat ini bisa dihasilkan dari perkawinan keturunan ternak yang terlalu dekat, misalnya antara pejantan dengan betina anak turunannya. Selama ini, para peternak kambing di desa binaan LAZ Al-Azhar mencatat data perkawinan, pemberian vaksin, serta kondisi kesehatan kambing secara manual menggunakan buku. Namun hal ini akan sangat menyusahakan apabila jumlah kambing yang ditenakkan sangat banyak, terlebih data yang harus dicatat juga rumit. Dengan adanya Sistem Informasi Data Ternak (Siternak) yang memanfaatkan penggunaan teknologi *barcode*, diharapkan para peternak dapat terbantu dalam melakukan pencatatan dan penilikan perkawinan kambing secara cepat dan mudah. Kode *barcode* akan dicetak dan dikalungkan ke leher ternak. Kalung tersebut sewaktu-waktu dapat dipindai menggunakan alat *scanner* atau kamera *handphone* untuk melihat data ternak secara detil melalui perangkat bergerak. Berdasarkan pengujian yang dilakukan terhadap peternak pada saat pelaksanaan kegiatan pengabdian, disimpulkan bahwa Siternak dapat dioperasikan dengan cukup mudah dan dapat membantu pengguna mengelola data ternaknya secara efisien. Pelaksanaan kegiatan pengabdian mulai dari pembangunan sistem hingga siap diberikan kepada pihak mitra membutuhkan waktu delapan bulan.

Kata Kunci: aplikasi bergerak, aplikasi web, *barcode*, sistem informasi data ternak.

PENDAHULUAN

Lembaga Amil Zakat (LAZ) Al-Azhar adalah satuan kerja yang dibentuk oleh Yayasan Pesantren Islam Al-Azhar yang bertujuan untuk memberdayakan masyarakat dhuafa melalui optimalisasi dana zakat, infaq, sedekah, dan dana sosial kemanusiaan lainnya yang dibenarkan oleh syariat agama dan sumber daya yang ada di masyarakat. Lembaga ini tidak berorientasi pada pengumpulan profit bagi pengurus organisasi. Lembaga ini memiliki berbagai macam program untuk mendorong keberdayaan masyarakat berbasis kearifan lokal sebagai bentuk pengabdian kepada masyarakat. Salah satu programnya yaitu desa binaan yang terletak di Desa Puncu, Kediri.

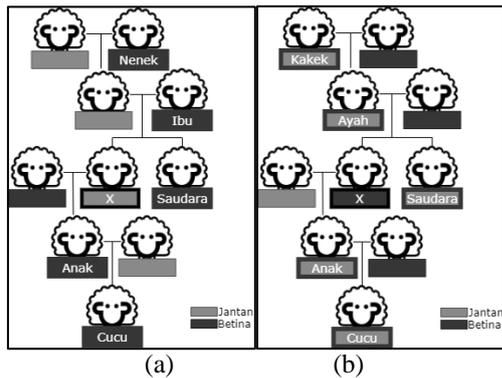
Desa binaan tersebut memiliki usaha peternakan kambing dan telah memiliki pencatatan untuk mengidentifikasi hewan ternaknya. Akan tetapi, pencatatan yang dilakukan masih secara manual tertulis di buku. Sehingga, untuk mendapatkan sebuah informasi

dari kumpulan data pada buku tersebut memakan waktu yang cukup lama dan membutuhkan banyak tenaga.

Kambing-kambing yang ditenakkan disana dapat dimanfaatkan untuk berbagai macam hal, misalnya aqiqah. Terdapat beberapa persyaratan yang harus dipenuhi agar kambing dapat disembelih untuk aqiqah (Qoyimah 2010). Syarat-syarat tersebut antara lain:

- Cukup umur, yaitu kira-kira berumur satu tahun, boleh jantan atau betina.
- Tidak cacat, yaitu bahwa hewan yang disembelih untuk akikah tersebut tidak buta total, buta sebelah, kurus kering (tulangnya tidak bersum-sum) dan pincang. Juga tidak sah berakikah dengan hewan yang telinga dan ekornya terpotong lebih dari sepertiga, ompong, dan gila.

Sering kali terjadi perkawinan kambing dengan keturunan yang sama. Hal tersebut tentunya memungkinkan kambing yang terlahir memiliki fisik dengan kemungkinan cacat yang besar. Untuk mencegah hal tersebut, setiap kambing di peternakan dapat diberikan sebuah identitas fisik yang unik agar



Gambar 1. Ilustrasi silsilah ternak.

identifikasi kambing menjadi lebih mudah. Mengacu pada syarat penyembelihan kambing di atas, untuk pemberian identitas secara fisik kepada kambing tidak boleh sampai melukai kambing. Sehingga, salah satu bentuk pemberian identitas fisik pada kambing dapat menggunakan *barcode* yang dipakaikan atau dikalungkan pada tubuh kambing, disebut dengan *necktag*.

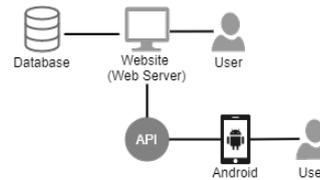
Dari permasalahan di atas, maka dibangunlah sebuah aplikasi sistem informasi untuk mencatat identitas ternak dengan mengimplementasikan teknologi *barcode*. Aplikasi yang dibangun oleh peneliti terdiri dari dua platform, yaitu web dan mobile. Aplikasi mobile juga dibuat dengan tujuan agar peternak dapat dengan mudah mengoperasikan aplikasi Siternak. Selain itu, kamera smartphone bisa dimanfaatkan untuk memindai kode *barcode* pada *necktag* bila pengguna tidak memiliki perangkat pemindai atau *barcode scanner*. Aplikasi website akan digunakan sebagai penyedia layanan API yang akan digunakan oleh aplikasi mobile, serta memiliki fungsionalitas sistem yang lengkap. Aplikasi mobile akan mengikuti fungsi utama pada aplikasi web.

TINJAUAN PUSTAKA

Kode Batang (Barcode)

Kode batang atau barcode merupakan pengkodean yang terdiri dari susunan baris dengan ketebalan dan jarak antar baris yang berbeda-beda. Barcode dapat dibaca oleh mesin karena memiliki standar pengkodean yang umum. Secara umum terdapat dua jenis *barcode* yang umum digunakan: *one-dimensional* (1D atau linear) dan *two-dimensional* (2D). *Barcode* 2D mampu menyimpan lebih banyak data dibandingkan dengan jenis pertama. Namun, keduanya digunakan oleh tipe aplikasi dan perangkat pemindai yang berbeda. Pada aplikasi Siternak, kode yang digunakan adalah *barcode* 1D. Walaupun terdapat beragam simbol dan penggunaan, tetapi semua tujuan kode batang adalah sama, yaitu mengodekan string karakter menjadi serangkaian garis batang dan spasi (Yudhanto and Kom 2011).

Tujuan penggunaan *barcode* pada aplikasi Siternak adalah untuk memudahkan pengguna menilik data salah satu hewan ternak tanpa perlu mengetikkan kode ternaknya. Pengguna cukup memindai kode batang



Gambar 2. Deskripsi umum sistem



Gambar 3. Alur online aplikasi mobile Siternak.



Gambar 4. Alur offline aplikasi mobile Siternak.

dengan menggunakan alat pemindai ataupun kamera smartphone.

JSON

JSON (Java Script Object Notation) merupakan salah satu bahasa yang ideal sebagai bahasa pertukaran-data (Crockford 2012). Hal ini dikarenakan JSON memiliki format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan oleh mesin. Pada dasarnya, JSON memiliki format teks dan tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang sangat umum.

JSON terdiri dari dua struktur utama, yaitu *object* dan *value*. *Object* bertindak sebagai struktur skema tempat menyimpan data, sedangkan *value* memuat daftar nilai terurut dalam bentuk array. Kedua struktur ini merupakan struktur data universal yang didukung oleh semua bahasa pemrograman modern. Pada aplikasi Siternak, pertukaran dan penyimpanan data sementara di aplikasi mobile menggunakan format data JSON. Dengan mempertimbangkan bahwa JSON juga memiliki format data yang sederhana dan efisien, sesuai dengan kapasitas penyimpanan smartphone yang juga sangat terbatas.

Web Service

Web Service atau layanan web adalah suatu sistem perangkat lunak yang ditujukan untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan. Sebuah aplikasi yang memiliki bentuk berbeda seperti website dan mobile dan dioperasikan pada perangkat yang berbeda membutuhkan dukungan web service untuk dapat saling bertukar data. Standar pertukaran data antar perangkat tersebut menggunakan format XML.

Web Service menyediakan layanan yang memungkinkan aplikasi dapat diakses oleh perangkat yang berbeda platform, sistem operasi, maupun bahasa compiler. Contoh arsitektur perangkat lunak yang mengimplementasikan web service seperti SOAP dan REST (Anon n.d.). Siternak mengadaptasi arsitektur REST yang menggunakan metode HTTP untuk memberikan layanan webnya (Trisianto 2013).

Tabel 1. Spesifikasi karakteristik aktor pada Siternak

Aktor	Tugas	Hak Akses
Administrator	Mengelola data Mendaftarkan peternak	Mengelola (melihat, menambah, mengubah, menghapus) data ternak, data ras, data pemilik, data penyakit, data riwayat penyakit, data kematian, data perkawinan Mendaftarkan akun peternak Mengelola data peternak dan data peternakan Melihat grafik ternak Melihat dan mengunduh laporan ternak berdasarkan rentang waktu yang diinginkan Melihat dan mengunduh berkas barcode
Ternak	Mengelola data	Mengelola data ternak, data ras, data pemilik, data penyakit, data riwayat penyakit, data kematian, data perkawinan Melihat grafik ternak Melihat dan mengunduh laporan ternak berdasarkan rentang waktu yang diinginkan Melihat dan mengunduh berkas barcode

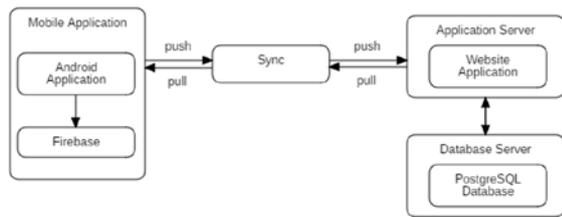
Tabel 2. Kebutuhan fungsional sistem Siternak

Kode	Kebutuhan Fungsional	Deskripsi
F01	Menangani kelola data ternak	Menampilkan data ternak, melakukan penambahan data ternak, melakukan perubahan data ternak, serta melakukan penghapusan data ternak.
F02	Menangani kelola data ras	Menampilkan data ras, melakukan penambahan data ras, melakukan perubahan data ras, serta melakukan penghapusan data ras.
F03	Menangani kelola data penyakit	Menampilkan data penyakit, melakukan penambahan data penyakit, melakukan perubahan data penyakit, serta melakukan penghapusan data penyakit.
F04	Menangani kelola data riwayat penyakit	Menampilkan data riwayat penyakit, melakukan penambahan data riwayat penyakit, melakukan perubahan data riwayat penyakit, serta melakukan penghapusan data riwayat penyakit.
F05	Menangani kelola data perkawinan	Menampilkan data perkawinan, melakukan penambahan data perkawinan, melakukan perubahan data perkawinan, serta melakukan penghapusan data perkawinan.
F06	Menangani kelola data pemilik	Menampilkan data pemilik, melakukan penambahan data pemilik, melakukan perubahan data pemilik, serta melakukan penghapusan data pemilik.
F07	Menangani kelola data kematian ternak	Menampilkan data kematian ternak, melakukan penambahan data kematian ternak, melakukan perubahan data kematian ternak, serta melakukan penghapusan data kematian ternak.
F08	Menangani kelola data peternakan	Menampilkan data peternakan, melakukan penambahan data peternakan, melakukan perubahan data peternakan, serta melakukan penghapusan data peternakan.
F09	Menangani kelola data peternak	Menampilkan data peternak, melakukan penambahan data peternak, melakukan perubahan data peternak, serta melakukan penghapusan data peternak.
F10	Melihat grafik berdasarkan kategori	Menampilkan grafik ternak berdasarkan umur, menampilkan grafik ternak berdasarkan ras, menampilkan grafik ternak berdasarkan jumlah kelahiran ternak, menampilkan grafik ternak berdasarkan jumlah kematian ternak.
F11	Melihat data keluarga ternak berdasarkan hasil pencarian <i>necktag</i>	Menampilkan data ayah dan ibu ternak, menampilkan data saudara-saudara ternak, menampilkan data anakan ternak, menampilkan data kakek dan nenek ternak, menampilkan data cucu ternak.
F12	Melihat hasil pencocokan perkawinan	Menampilkan hasil boleh atau tidak pasangan ternak dikawinkan
F13	Melihat <i>barcode</i>	Menampilkan <i>barcode</i> dari setiap <i>necktag</i>
F14	Mengunduh be	Mengunduh berkas <i>barcode</i>
F15	Melihat laporan	Menampilkan laporan ternak berdasarkan rentang waktu
F16	Mengunduh berkas laporan	Mengunduh berkas laporan berdasarkan rentang waktu

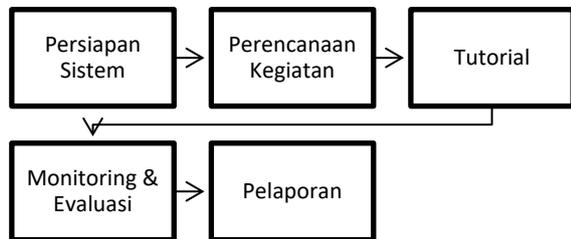
REST *Web Service* bekerja dengan REST server menyediakan jalur untuk akses *resource* atau data, sedangkan REST *client* melakukan akses *resource* dan kemudian menampilkan atau menggunakannya. *Resource* yang dihasilkan umumnya adalah berbentuk JSON dan XML. Dalam mengakses sebuah *resource*, REST menggunakan konsep URI dimana *method* yang digunakan adalah GET (Warkim and Sensuse 2017).

Sinkronisasi Offline Storage

Tidak setiap saat tersedia koneksi *mobile* internet yang memadai. Koneksi internet bisa turun-naik atau tidak ada saat di perjalanan. Sehingga mekanisme penyimpanan yang tepat perlu dipilih, baik untuk penyimpanan perangkat lokal maupun untuk penyimpanan server berbasis awan. Mesin penyimpanan yang baik akan memastikan informasi disimpan secara



Gambar 6. Arsitektur sistem Siternak.



Gambar 7. Tahapan pelaksanaan kegiatan pengabdian.



Gambar 5. Tampilan antarmuka aplikasi mobile Siternak

meyakinkan, mengurangi *bandwidth*, dan meningkatkan responsivitas.

Sinkronisasi *offline storage* akan berguna pada aplikasi mobile ketika pengguna melakukan perubahan pada data ternak. Ketika koneksi internet tidak ada, hasil perubahan data akan disimpan pada *offline storage* di perangkat pengguna terlebih dahulu sebagai antrian, dan setelah kembali tersambung dengan koneksi internet, sistem akan melakukan sinkronisasi otomatis.

Aplikasi Siternak mengimplementasikan Firebase sebagai layanan basis data di penyimpanan offline pada perangkat bergerak. Firebase merupakan salah satu dari produk Google yang fokus menyediakan *database realtime* dan *backend* sebagai layanan (*Backend as a Service*). Selain itu, Firebase merupakan suatu aplikasi yang memungkinkan pengembang membuat API untuk sinkronisasi *client* yang berbeda-beda dan disimpan pada penyimpanan awan Firebase (Wahyujati and others 2017).

ANALISIS DAN PERANCANGAN

Aplikasi Siternak dibangun dengan menerapkan metode *waterfall*. Metode ini merupakan metode pengembangan perangkat lunak yang memiliki 5 tahapan di dalamnya, yaitu: tahap analisis dan definisi kebutuhan, tahap perancangan sistem, tahap implementasi, tahap pengujian, dan tahap perawatan (Binanto 2014).

Analisis Permasalahan

Masalah utama dari pengelolaan kambing berjumlah ratusan ekor pada peternakan kambing di Desa Puncu adalah pengenalan identitas riwayat kambing untuk perkawinan. Saat ini, peternakan tersebut telah memiliki pencatatan data ternak manual dalam buku. Sehingga, untuk mencari informasi data riwayat ternak diperlukan banyak waktu dan terkadang informasi yang didapatkan tidak lengkap karena luput dari pemindaian data. Kendala yang dihadapi adalah ketika kambing yang dikawinkan ternyata merupakan kambing pada satu garis keturunan.

Hal tersebut akan menyebabkan anak kambing yang dilahirkan kemungkinan besar memiliki cacat fisik. Apabila ternak mengalami kecacatan, ternak tersebut tidak dapat dimanfaatkan untuk disembelih berdasarkan syarat penyembelihan ternak.

Tim pengabdian mengusulkan solusi untuk mengatasi masalah tersebut berupa sistem informasi data ternak berbasis web dan *mobile* untuk pencatatan yang lebih praktis, efektif dan efisien. Setiap kambing di peternakan diberikan identitas berupa *barcode* agar pemeriksaan identitas kambing lebih cepat. *Barcode* tersebut dapat dipindai melalui pemindaian *barcode* pada sistem. Pemindaian *barcode* akan menghasilkan *output* berupa data riwayat dan keluarga ternak yang dipindai. Sistem yang dibangun juga memiliki fitur untuk melihat simulasi boleh atau tidaknya pasangan ternak dikawinkan dengan memasukkan identitas *necktag* ternak jantan dan betina. Apabila pasangan ternak yang dimasukkan merupakan ternak dengan satu garis keturunan, maka sistem akan mengeluarkan pesan bahwa pasangan ternak tidak boleh untuk dikawinkan. Tetapi jika pasangan ternak bukan dari satu garis keturunan, maka sistem akan mengeluarkan pesan bahwa pasangan ternak boleh untuk dikawinkan.

Pembacaan identitas ternak menggunakan teknologi *barcode* memberikan kemudahan bagi peternak untuk mengetahui informasi dari banyak data. Sistem *mobile* juga dibangun untuk memudahkan peternak dalam melakukan pemindaian *barcode*. *Smartphone* saat ini sudah dilengkapi dengan fitur kamera dan jaringan internet, sehingga memungkinkan peternak untuk melakukan pemindaian *barcode* melalui fitur meskipun peternak tersebut tidak memiliki perangkat pemindai *barcode*.

Aturan Perkawinan Ternak

Berdasarkan permasalahan yang dihadapi oleh peternak kambing di Desa Puncu, seekor ternak tidak boleh dikawinkan dengan turunan keluarganya sendiri



Gambar 8. Tampilan antarmuka aplikasi web www.siternak.net.

untuk menghindari kecacatan keturunan. Untuk menghindari perkawinan dengan turunan yang dekat, maka setiap ternak harus memiliki identitas fisik dan data riwayat yang lengkap.

Seekor ternak tidak boleh dikawinkan dengan ternak yang berada pada satu garis keturunan, baik vertikal maupun horizontal. Akan tetapi, aturan larangan ini dalam Siternak hanya berlaku sampai 3 generasi berturut-turut. Sehingga, apabila ternak yang dikawinkan memiliki hubungan buyut dan cicit, maka ternak dapat dikawinkan. Lalu, apabila ternak berjenis kelamin jantan, maka ternak tidak boleh dikawinkan dengan ibu-nya, nenek-nya, saudara betina-nya (baik se-ayah dan se-ibu ataupun hanya se-ayah atau se-ibu saja), anak betina-nya, dan cucu betina-nya. Apabila ternak berjenis kelamin betina, maka ternak tidak boleh dikawinkan dengan ayahnya, kakek-nya, saudara jantan-nya (baik se-ayah dan se-ibu ataupun hanya se-ayah atau se-ibu saja), anak jantan-nya, dan cucu jantan-nya. Jika ternak yang dipasangkan tidak memiliki hubungan keturunan, maka ternak boleh untuk dikawinkan. Ilustrasi dapat dilihat pada Gambar 1 (a) untuk ternak jantan dan Gambar 1 (b) untuk ternak betina.

Deskripsi Umum Sistem

Sistem Siternak terdiri dari dua aplikasi, yaitu aplikasi web dan aplikasi mobile menggunakan platform Android. Deskripsi umum sistem ditunjukkan pada Gambar 2. Aplikasi web akan bertindak sebagai aplikasi sistem informasi dan juga sebagai web server penyedia API untuk aplikasi mobile. Aplikasi mobile dapat menjalankan fungsi utama seperti yang tersedia pada aplikasi web. Aplikasi web akan dibangun dengan menggunakan arsitektur MVC (Model-View-Controller) dari kerangka kerja Laravel.

Dengan memanfaatkan kemampuan *offline storage* dari *Firestore Realtime Database*, aplikasi *mobile* akan tetap dapat menjalankan fungsionalitasnya dalam menambah dan mengubah data walaupun dalam keadaan *offline* (tanpa internet yang terhubung). Setelah itu, sistem akan otomatis melakukan sinkronisasi ketika perangkat terhubung dengan internet.

Pada Gambar 3 menunjukkan alur aplikasi *mobile* ketika perangkat dalam keadaan *online*. Untuk mendapatkan atau mengolah data, representasi data akan diperoleh dari API pada *web server*. Request akan

dikirimkan dari perangkat *mobile* dan respon akan diperoleh dari *web server* yang terhubung langsung dengan basis data.

Sedangkan pada Gambar 4 menunjukkan alur aplikasi *mobile* ketika perangkat dalam keadaan *offline*. Kemampuan *Firestore* ini akan dimanfaatkan untuk menambah dan mengubah data dan disimpan di internal perangkat bergerak saja. Ketika perangkat kehilangan koneksi internet saat pengguna sedang menambah atau mengubah data, maka data akan ditampung sebagai antrian oleh *Firestore Realtime Database* dan akan ditambahkan ke basis data apabila koneksi internet telah kembali.

Spesifikasi Kebutuhan Perangkat Lunak

Spesifikasi kebutuhan perangkat lunak berisi kebutuhan perangkat lunak yang direpresentasikan dalam bentuk kebutuhan fungsional, diagram kasus penggunaan, dan diagram aktivitas.

1) Aktor

Aktor adalah pihak-pihak yang terlibat dan berinteraksi langsung dengan sistem. Aktor yang berada pada sistem ini dibahas lebih rinci pada Tabel 1.

2) Kebutuhan Fungsional

Kebutuhan fungsional merupakan layanan yang harus dimiliki oleh perangkat lunak. Kebutuhan ini mencakup bagaimana sistem harus bereaksi pada input tertentu dan bagaimana perilaku sistem pada situasi tertentu. Kebutuhan fungsional dari perangkat lunak untuk aplikasi ini dijelaskan pada Tabel 2.

3) Diagram Kasus Penggunaan

Kasus penggunaan didasarkan pada hasil analisis kebutuhan fungsional dari perangkat lunak. Setiap kasus penggunaan melihat data memiliki *extend* kasus penggunaan menambah, mengubah, dan menghapus data.

Perancangan Sistem

Aplikasi web dibangun sebagai aplikasi utama dan *web server* penyedia API untuk aplikasi *mobile* agar aplikasi *mobile* dapat terintegrasi dengan aplikasi web. Pada aplikasi *mobile*, dapat dimanfaatkan kemampuan *offline storage* dari *Firestore Realtime Database*. *Firestore* akan digunakan ketika perangkat *mobile* dalam keadaan *offline* saat pengguna sedang menambahkan data atau mengubah data saja. *Firestore* akan menampung data yang telah ditambah atau diubah tersebut ke dalam antrian. Sehingga, ketika perangkat *offline*, pengguna hanya dapat mengakses beberapa fungsionalitas saja dari aplikasi *mobile*.

Arsitektur sistem untuk kedua aplikasi ini dijelaskan pada Gambar 5. Aplikasi web akan terhubung langsung dengan basis data dan sekaligus menjadi *web server*. Aplikasi *mobile* akan melakukan pertukaran data langsung dengan *web server* ketika perangkat *mobile* dalam keadaan *online*. Ketika perangkat *mobile* dalam keadaan *offline*, aplikasi akan melakukan pertukaran data dengan *Firestore*. Sinkronisasi data pada *Firestore* dengan *web server* dilakukan ketika perangkat mendapatkan koneksi internet. Saat sinkronisasi dilakukan, data dari *web server* akan ditarik oleh *Firestore* untuk mengecek

apakah ada perubahan pada data yang telah tersimpan, kemudian semua data pada antrian Firebase akan ditarik ke *web server*.

IMPLEMENTASI

Implementasi Pembuatan dan Pembacaan Barcode

Pembuatan dan pembacaan *barcode* pada aplikasi web dan aplikasi *mobile* tidak sama. *Library* yang digunakan berbeda, sehingga implementasinya-pun berbeda. Implementasi pembacaan dan pembuatan *barcode* pada aplikasi web menggunakan *library* milon/*barcode*. Untuk dapat menggunakan *library* ini, diperlukan perintah composer require milon/*barcode* pada *project*. Aplikasi ini hanya membutuhkan jenis *barcode* satu dimensi dengan kode 128. Kode ini merupakan *barcode* yang memiliki kerapatan tinggi, sehingga dapat mengkodekan keseluruhan simbol ASCII sebanyak 128 karakter dalam luasan yang paling minim dibanding *barcode* jenis lain.

Pembacaan *barcode* dilakukan oleh alat pembaca *barcode* (*barcode scanner*). Sehingga, *field* pencarian *necktag* pada aplikasi akan langsung membaca teks, bukan lagi dalam bentuk *barcode*. Pembuatan *barcode* dilakukan menggunakan *library* yang sudah ditambahkan sebelumnya.

Implementasi pembacaan dan pembuatan *barcode* pada aplikasi *mobile* menggunakan dua *library* zxing. Proses pembacaan *barcode* dibutuhkan kedua *library* tersebut. *Library* pertama digunakan untuk membuat *activity* yang meng-*extends* fungsi *CaptureActivity*, yaitu *activity* untuk memanfaatkan fitur kamera pada *smartphone*. *Library* kedua digunakan untuk menginisiasi *activity* sebelumnya dan mengatur agar kamera dapat membaca *barcode*. Pembuatan *barcode* cukup menggunakan *library* kedua saja. Pada setiap *necktag* ternak, dibuat gambar *barcode* dengan *bitmap*.

Implementasi Sinkronisasi Offline Storage

Pembuatan *offline storage* diimplementasikan menggunakan Firebase dan Android Studio. Fitur yang mengimplementasikan *offline storage* adalah fitur menambahkan data dan mengubah data. Saat perangkat *mobile offline* dan pengguna sedang dalam keadaan menambah atau mengisi data, maka data akan disimpan ke dalam antrian Firebase ketika pengguna menekan tombol simpan (tambah/ubah). Ketika perangkat kembali *online*, maka sistem akan melakukan sinkronisasi data pada Firebase dengan data pada basis data yang dikirimkan melalui *web server*.

Penerapan penambahan data saat keadaan *offline* ada pada setiap *activity* penambahan data. Penerapan pengubahan data saat keadaan *offline* ada pada setiap *activity* pengubahan data. Sinkronisasi antara Firebase dengan basis data dari *web server* akan dilakukan ketika koneksi internet didapatkan. Sinkronisasi berjalan saat pengguna membuka aplikasi dan berada pada *main activity* atau *dashboard*. Sinkronisasi dilakukan dengan memanggil dua fungsi, yaitu fungsi *pull* atau fungsi

menarik data dari basis data ke Firebase, dan fungsi *push* atau fungsi mengirim data ke basis data dari Firebase.

Pada fungsi *pull* tidak semua data dari basis data ditarik untuk disimpan ke dalam Firebase. Data yang disimpan pada Firebase hanya data yang dibutuhkan ketika pengguna menambahkan atau mengubah data. Data tersebut digunakan pada *field* atau isian *form* data dengan jenis masukan berupa *select options*. Ketika pengguna memilih *field* tersebut, maka data *select options* akan diambil langsung dari Firebase, bukan *web server*. Data *select options* pada Firebase disimpan dalam *node* dengan nama "options".

Fungsi *push* akan mengirim semua data pada Firebase yang berada dalam *node* bernama "datas". Pada *node* ini terdapat dua *child node*, yaitu *node* *addData* dan *node* *editData*. *Node* ini akan terhapus jika data yang ada didalamnya telah berhasil dimasukkan ke dalam basis data pada *web server*. *Node* *addData* menyimpan antrian data yang ditambahkan ketika perangkat *mobile* sedang *offline*. Dan *node* *editData* menyimpan antrian data yang diubah ketika perangkat *mobile* sedang *offline*. Fungsi ini melakukan proses pengiriman data ke *web server* secara terpisah. Dengan artian, setiap *child* pada *addData* dan *editData* akan melakukan pengiriman data masing-masing, karena *child* tersebut merupakan representasi dari masing-masing tabel pada basis data.

PELAKSANAAN KEGIATAN PENGABDIAN

Sesuai dengan rencana kegiatan pengabdian masyarakat yang dijabarkan pada Gambar 6, tim pengabdian telah melaksanakan ke lima kegiatan tersebut yaitu: Persiapan Sistem, Perencanaan Kegiatan, Tutorial, Monitoring dan Evaluasi, dan Pelaporan.

Pada tahap Persiapan Sistem, tim pengabdian telah melakukan pengadaan piranti/alat yang dibutuhkan selama pelaksanaan terkait dengan pengadaan piranti *barcode scanner* dan komponen pendukung lainnya. Tim pengabdian juga melakukan finalisasi tampilan antarmuka aplikasi baik dalam bentuk *mobile application* maupun *web application*. Tampilan dari kedua antarmuka tersebut bisa dilihat pada Gambar 7 dan Gambar 8.

Untuk mendukung ketersediaan aplikasi agar dapat diakses dengan mudah oleh LAZ Al-Azhar maupun desa binaan, maka tim pengabdian melakukan pengadaan *hosting server* untuk tiga tahun. Aplikasi sistem informasi ternak yang dibangun oleh tim pengabdian disebut dengan *Siternak*. Untuk mengoperasikan aplikasi *Siternak* melalui browser, bisa diakses melalui alamat www.siternak.net. Sedangkan untuk mengoperasikan *Siternak mobile*, pengguna harus mengunduh dan menginstall aplikasi terlebih dahulu melalui alamat <http://bit.ly/siternak2020>. Dokumen user manual dan panduan langkah instalasi sistem juga telah diunggah pada alamat repositori yang sama.

Pada tahap Perencanaan Kegiatan, tim pengabdian telah melakukan koordinasi dengan pihak mitra dan melakukan penyusunan dokumen *user manual*. Dokumen ini ditujukan untuk memudahkan pengguna aplikasi baik

sebagai administrator maupun peternak agar dapat mengoperasikan aplikasi secara mandiri.

Sebagai implementasi tahap ketiga kegiatan, yaitu Tutorial, tim pengabdian telah melaksanakan kegiatan tutorial secara daring menggunakan aplikasi Zoom pada tanggal 26 Oktober 2020. Pelatihan diikuti oleh jajaran pihak mitra dari LAZ Al-Azhar Jawa Timur dan Pusat Jakarta.

Mempertimbangkan respon dari Bapak Aditya Kusuma (Kepala Perwakilan LAZ Al-Azhar Jatim) yang sangat positif terhadap aplikasi Siternak, dan minat LAZ Al-Azhar Pusat Jakarta untuk turut mengelola aplikasi tersebut, maka tim pengabdian merencanakan kegiatan lanjutan. Tindak lanjut dari pelaksanaan kegiatan pelatihan tutorial aplikasi secara daring, tim pengabdian merencanakan pelatihan lanjutan yang khusus ditargetkan untuk tim IT dari pihak LAZ Al-Azhar Pusat Jakarta. Tim IT tersebut menjadi pengelola utama server aplikasi Siternak yang telah dihosting oleh tim pengabdian.

UJI COBA DAN EVALUASI

Pengujian sistem dilakukan dengan metode *black box* untuk menguji masing-masing fungsionalitas yang sudah dirancang pada Siternak. Metode *black box* merupakan metode pengujian perangkat lunak yang memeriksa fungsionalitas dari suatu perangkat lunak tanpa memandang struktur internalnya. Beberapa pihak mitra LAZ Al-Azhar dilibatkan sebagai pengujian pengguna aplikasi.

Salah satu fungsionalitas yang penting dari Siternak adalah fungsi melihat hasil pencocokan perkawinan. Pengujian melihat hasil pencocokan perkawinan merupakan pengujian terhadap kemampuan aplikasi untuk menampilkan hasil apakah ternak boleh dikawinkan atau tidak berdasarkan pada garis keturunan dan aturan perkawinan ternak. Hasil ujicoba aplikasi Siternak telah berhasil menampilkan hasil penilikan perkawinan sesuai dengan aturan yang ditetapkan.

KESIMPULAN DAN KEBERLANJUTAN

Pada mulanya kegiatan ini hanya ditujukan untuk dimanfaatkan oleh para perternak di desa binaan LAZ Al-Azhar Jatim, namun mempertimbangkan respon pihak mitra yang sangat positif dan menghendaki agar pemanfaatan aplikasi ini bisa diperluas untuk semua desa binaan LAZ yang ada di Indonesia, maka tim pengabdian dan pihak mitra menyusun rencana kegiatan keberlanjutan dengan melibatkan tim IT dari LAZ Al-Azhar Pusat Jakarta. Adapun rencana kegiatan keberlanjutan tersebut dalam bentuk sebagai berikut:

1. Tutorial role administrator dan serah terima pengelolaan server hosting Siternak kepada tim IT LAZ Al-Azhar Pusat.
2. Tutorial instalasi dan pengoperasian aplikasi mobile Siternak dengan role peternak.

Kedua kegiatan tersebut direncanakan akan dapat diselesaikan pada Desember 2020.

Kendala dan Solusi

Kendala yang dihadapi dalam pelaksanaan kegiatan pengabdian masyarakat ini di antaranya, kegiatan harus dilaksanakan secara daring dengan mempertimbangkan situasi pandemi yang masih sedang berlangsung di Indonesia. Dikarenakan tim pengabdian tidak memungkinkan berkomunikasi dan berinteraksi langsung dengan para peternak di desa binaan Puncu Kediri, maka semua fitur pembangunan aplikasi ini didasarkan pada asumsi dan permintaan dari koordinator pembina LAZ Al-Azhar. Dalam hal ini, pihak LAZ Al-Azhar yang nantinya akan bertanggung jawab untuk mengimplementasikan dan melatih para peternak di desa binaanya untuk dapat memanfaatkan aplikasi Siternak secara massal.

UCAPAN TERIMA KASIH

Pengabdian masyarakat ini didukung oleh Departemen Teknik Informatika, Fakultas Teknologi Elektro dan Informatika Cerdas, melalui pendanaan kegiatan pengabdian masyarakat Skema Dana Departemen ITS Tahun 2020.

DAFTAR PUSTAKA

- Anon. n.d. "What Is REST - REST API Tutorial." Retrieved October 30, 2020 (<https://restfulapi.net/>).
- Binanto, I. 2014. "Analisa Metode Classic Life Cycle (Waterfall) Untuk Pengembangan Perangkat Lunak." 2014.
- Crockford, Douglas. 2012. "Json." *ECMA International*.
- Qoyimah, Qoyimah. 2010. "Analisis Hukum Islam Terhadap Mekanisme Pelaksanaan Dan Pendistribusian Aqiqah Sebagai Konsumsi Pengajian Di Yayasan Al-Jihad Surabaya." IAIN Sunan Ampel Surabaya.
- Trisianto, Muhammad Yudha. 2013. "Perancangan web service pada aplikasi inventaris yang berplatform. net." PhD Thesis, Universitas Widyatama.
- Wahyujati, Desta, and others. 2017. "Implementasi Teknologi Firebase Pada Aplikasi Pencarian Lokasi Service Kamera Berdasarkan Rating Berbasis Android." PhD Thesis, STMIK AKAKOM Yogyakarta.
- Warkim, Warkim, and Dana Indra Sensuse. 2017. "Model Integrasi Sistem Dengan Pendekatan Metode Service Oriented Architecture Dan Model View Controller Pada Pusat Penelitian Perkembangan Iptek Lembaga Ilmu Pengetahuan Indonesia." *Jurnal Teknik Informatika Dan Sistem Informasi* 3(1).
- Yudhanto, Yudha, and S. Kom. 2011. "Sejarah Teknologi Barcode." Tersedia: [Http://ilmukomputer.Org/Wpcontent/Uploads/2011/03/Sejarah-Barcode-Yudha.Pdf](http://ilmukomputer.Org/Wpcontent/Uploads/2011/03/Sejarah-Barcode-Yudha.Pdf) [20 September 2012].