

# Pembangunan Sistem Administrasi Surat Menyurat di Paroki Hati Santa Perawan Maria Tak Bercela Kumetiran

Alisha Aileen<sup>1</sup>, Martinus Maslim<sup>2</sup>, Suryanti Ch.<sup>3</sup>

Program Studi Informatika, Fakultas Teknologi Industri, Universitas Atma Jaya Yogyakarta  
Jl. Babarsari No.43, Janti, Caturtunggal, Kec. Depok, Kabupaten Sleman 55281,  
Daerah Istimewa Yogyakarta, Indonesia

Email: <sup>1</sup>alishaileen05@gmail.com, <sup>2</sup>martinus.maslim@uajy.ac.id, <sup>3</sup>ch.suryanti@uajy.ac.id

***Abstrak.** Sistem administrasi surat yang dijalankan oleh Gereja Katolik Hati Santa Perawan Maria Tak Bercela Kumetiran dinilai kurang efisien karena prosesnya memakan waktu dan tenaga ekstra. Selain itu, surat yang diterima Sekretariat Paroki kerap memiliki format yang berbeda. Ketidaksamaan data tersebut disebabkan oleh miskomunikasi antara Ketua Lingkungan dan Sekretariat Paroki. Masalah tersebut melatarbelakangi dibangunnya sistem informasi yang dapat mengakomodir administrasi tersebut serta melacak disposisinya. Sistem dibangun berbasis website dan terbagi menjadi back end dan front end. Aplikasi back end menggunakan kerangka kerja Node JS dan Express JS dengan arsitektur REST API. Sementara front end dibangun menggunakan kerangka kerja Vue JS untuk menangani interaksi dengan klien. Pembangunan menghasilkan sebuah website yang dapat menangani administrasi surat. Berdasarkan respon dari beberapa pengurus gereja, sistem yang dibangun mempermudah pengajuan surat oleh umat serta mempermudah proses verifikasi suratnya. Selain itu, sistem juga dapat menjaga konsistensi format data.*

***Kata Kunci:** gereja, sistem administrasi, pembangunan sistem*

## 1. Pendahuluan

### 1.1. Latar Belakang

Gereja Katolik Hati Santa Perawan Maria Tak Bercela Kumetiran melayani umatnya dalam melaksanakan sakramen/kegiatan gereja. Umat dapat mengajukan surat dengan datang ke Ketua Lingkungan dimana umat tersebut tinggal (Ketua Lingkungan adalah salah satu umat lingkungan yang ditunjuk oleh Sekretariat Paroki untuk mengetuai lingkungan tersebut) lalu mengisi data untuk surat. Setelah ditandatangani/diverifikasi Ketua Lingkungan, surat diserahkan kepada pengurus gereja untuk selanjutnya diproses dan ditandatangani oleh Romo Paroki. Di Gereja Katolik Hati Santa Perawan Maria Tak Bercela Kumetiran Yogyakarta, pengurus tersebut adalah Sekretariat Paroki. Sekretariat Paroki adalah divisi yang bertugas mengelola data administrasi surat. Setelah selesai diproses, surat tersebut dapat digunakan untuk kebutuhan sakramen gereja seperti baptis, krisma, pernikahan, dan lain-lain. Administrasi surat masih menggunakan cara manual dalam pengolahan data. Sehingga kerap terjadi redundansi/duplikasi data bahkan ketidaksesuaian antara data satu dengan yang lainnya. Selain itu, pengumpulan data yang dilakukan secara manual kerap memakan waktu dan tenaga yang relatif banyak sehingga dianggap kurang efisien.

Teknologi telah berkembang pesat dan dapat memunculkan perubahan cara-cara kerja baru yang lebih efektif dan efisien [1] sehingga masalah administrasi manual dapat diatasi dengan adanya pembangunan sistem informasi. Sistem mendigitalisasi proses administrasi dan menangani data-data umat dan kegiatan gereja. Sistem ini nantinya akan terdiri atas dua sistem utama yaitu aplikasi web yang berperan sebagai admin dan aplikasi web yang berperan sebagai antarmuka untuk umat-umat Gereja Katolik Hati Santa Perawan Maria Tak Bercela Kumetiran. Berdasarkan latar belakang, maka diperoleh rumusan masalah yaitu: (1) Bagaimana sistem ini dapat membantu Umat, Ketua Lingkungan, Sekretariat Paroki, dan Romo Paroki dalam mengelola administrasi surat menyurat di Paroki Kumetiran

Sistem ini hanya mengelola surat menyurat yang berkaitan dengan administrasi umat yang berhubungan dengan Paroki Hati Santa Perawan Maria Tak Bercela Kumetiran, yaitu: (1) Surat izin Pelayanan Ekaristi (2) Surat Keterangan Beasiswa (3) Surat Keterangan (4) Surat Baptis Anak (5) Surat Baptis Dewasa (6) Surat Komuni I/Krisma (7) Surat Keterangan Calon Pengantin (8) Surat Pelayanan Minyak Suci (9) Surat Keterangan Kematian (10) Surat Keterangan Pindah Domisili. Umat sendiri adalah para pemeluk suatu agama [2]. Dalam hal ini adalah agama Katolik. Umat dapat meliputi semua orang dari semua umur dan semua golongan yang memeluk agama Katolik (umat Katolik). Seorang umat hanya bisa terdaftar di satu paroki. Sementara paroki sendiri merupakan suatu wilayah tertentu dimana sekumpulan umat Katolik tinggal [3]. Jadi, sistem ini akan menangani permintaan umat Katolik yang ada di Paroki Hati Santa Perawan Maria Tak Bercela Kumetiran atas surat-surat yang sudah disebutkan. Pembangunan sistem informasi ini memiliki tujuan yaitu membantu proses administrasi surat menyurat yang ada di Paroki Hati Santa Perawan Maria Tak Bercela Kumetiran.

## 2. Tinjauan Pustaka

Pada penelitian ini, sistem informasi dibangun untuk mengatasi pendataan umat beserta sakramen yang pernah diambil umat tersebut. Sistem tersebut berbasis web dan dibuat dengan bahasa PHP dan JavaScript. Dengan bantuan sistem tersebut, pendataan dan penyimpanan data umat dapat dilakukan dengan lebih mudah [4].

Pada penelitian selanjutnya, sebuah sistem informasi juga dibangun untuk pengolahan data jemaat per rayon. Pembangunan sistem dilakukan dengan bahasa PHP menggunakan kerangka kerja CodeIgniter dan MySQL sebagai basis data. Sistem tersebut bersifat partisipatif sehingga memudahkan pengumpulan data. Pengurus gereja tidak perlu mengumpulkan data secara manual namun para jemaat dapat berpartisipasi memasukan datanya secara mandiri sehingga memudahkan pengumpulan data serta menghemat waktu dan tenaga [5].

Penelitian selanjutnya merupakan pembangunan aplikasi berbasis android. Aplikasi tersebut dapat menyimpan dan menampilkan informasi seputar gereja, acara/kegiatan gereja, dan berita-berita terbaru. Sehingga umat tidak perlu pergi ke papan pengumuman gereja untuk mengetahui informasi terbaru [6].

Penelitian berikutnya adalah pembangunan sistem yang dapat menyediakan informasi untuk umat seperti acara gereja dan ibadah. Selain itu, sistem tersebut juga menyediakan halaman tersendiri untuk admin. Admin mengunggah informasi dan berkas-berkas melalui website admin. Lalu seluruh umat gereja dapat melihat informasi tersebut di website yang berbeda sehingga memudahkan penyebaran informasi [7].

Penelitian selanjutnya merupakan pembangunan aplikasi berbasis android dengan tujuan mempermudah penyebaran informasi tanpa menggunakan kertas. Aplikasi tersebut ditujukan kepada para petugas internal gereja terkait jadwal pelayanan dan kepanitiaan acara-acara gereja. Aplikasi ini dibangun menggunakan Android Studio dan Firebase sebagai basis data [8].

## 3. Metodologi Penelitian

Metodologi yang digunakan pada penelitian ini adalah sebagai berikut: (1) Wawancara dengan pengurus gereja dan mengumpulkan data yang dibutuhkan dalam pembangunan sistem. Wawancara dapat memberi jawaban dari pertanyaan yang disiapkan bahkan dalam praktiknya, pertanyaan-pertanyaan tersebut dapat berkembang selama wawancara berlangsung [9]. (2) Analisis kebutuhan sistem dari data yang didapat saat wawancara. Hasil dari tahap ini berupa use case diagram dari kebutuhan yang sudah dipaparkan pada tahap pertama. (3) Perancangan sistem untuk mempersiapkan rancangan aplikasi berdasarkan hasil dari analisis kebutuhan sistem. Hasil dari tahap ini berupa rancangan-rancangan seperti *Entity Relationship Diagram* (ERD), *Class Diagram*, dan desain *mockup* antarmuka aplikasi. (4) Pembangunan siste

berdasarkan spesifikasi kebutuhan yang sudah ditentukan dengan menulis kode program. (5) Pengujian sistem menggunakan metode *black box testing* untuk mengetahui kehandalan sistem [10]. (6) Penulisan laporan yang merupakan tahap akhir dari pembangunan sistem informasi. Semua hasil dari aplikasi yang sudah dibangun akan dicatat dan dievaluasi.

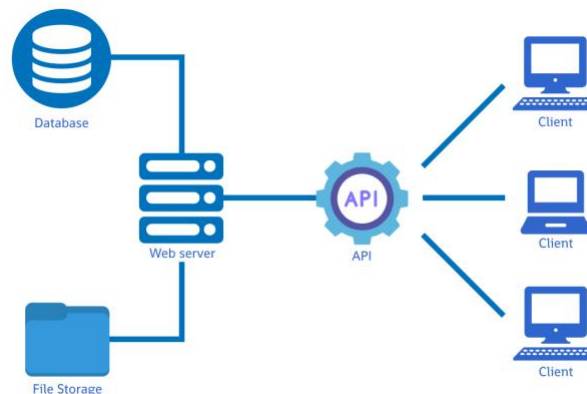
## 4. Hasil dan Diskusi

### 4.1. Analisis Sistem

Pembangunan sistem menghasilkan sebuah sistem bernama Sigrejak. Sigrejak terbagi menjadi dua yaitu *back end* dan *front end*. *Back end* Sigrejak dibangun dengan menggunakan Node JS dan kerangka kerja Express JS. MySQL digunakan sebagai DBMS (*Database Management System*). Sementara untuk *front end*, pengkodean dilakukan menggunakan kerangka kerja Vue JS yang memiliki arsitektur MVVM (Model, View, View Model) sehingga memungkinkan konsep SPA (*Single Page Application*) dan meminimalisir proses *render* halaman web pada sisi klien.

### 4.2. Perancangan Sistem

Sigrejak dirancang untuk menggunakan arsitektur REST API atau *Representational State Transfer Application Programming Interface*.



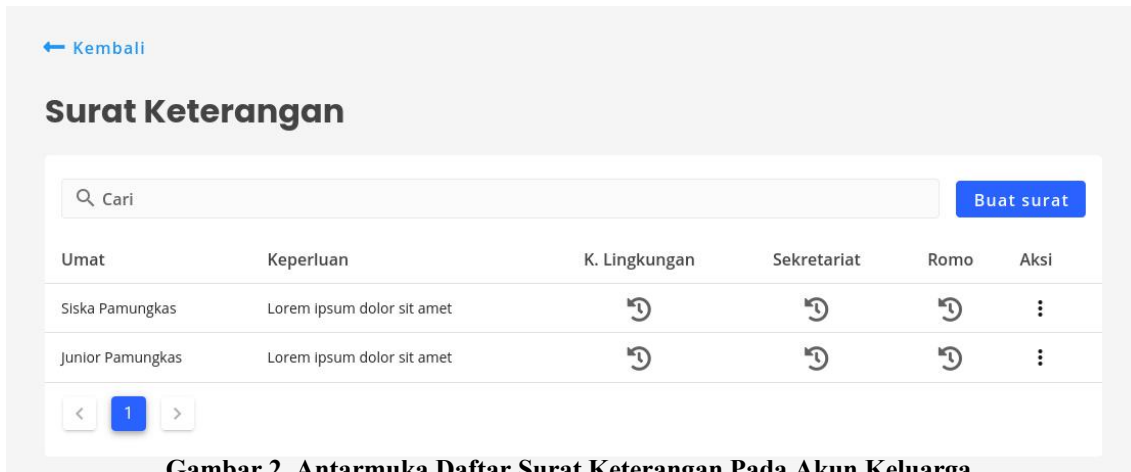
**Gambar 1. Arsitektur Sigrejak**

Gambar 1 merepresentasikan arsitektur Sigrejak. Pada Gambar 1, terlihat bahwa API menjadi jembatan komunikasi antara klien dan *web server*. Ketika klien mengirim *request* untuk suatu data, API meneruskan *request* tersebut. Selanjutnya request diproses oleh *Web Server*. *Web Server* akan mengambil data dari *Database*. Jika klien meminta berkas, *Web Server* akan mengambil dahulu nama berkas yang disimpan di *Database* lalu mencari berdasarkan nama berkas di *File Storage* dan mengirimkan berkas dan data melalui API untuk diteruskan ke klien. Sehingga dapat dikatakan bahwa API menghubungkan aplikasi *front end* dan *back end*.

### 4.3. Implementasi Sistem

#### 4.3.1 Keluarga

Kepala Keluarga memiliki akun Keluarga. Data surat dapat tampil pada akun Keluarga merupakan surat yang dibuat oleh Kepala Keluarga yang sedang login. Data disajikan dalam bentuk tabel. Tabel tersebut juga menampilkan status verifikasi surat. Surat masih bisa diubah jika Ketua Lingkungan belum memverifikasi surat. Surat hanya bisa dicetak jika semua ketiganya sudah memverifikasi surat. Berikut tampilan antarmuka pada Gambar 2.



**Gambar 2. Antarmuka Daftar Surat Keterangan Pada Akun Keluarga**

Gambar 2 menunjukkan antarmuka daftar surat. Terdapat indikator verifikasi supaya dapat memudahkan Kepala Keluarga mengetahui apakah surat tersebut sudah diverifikasi atau belum.

```

const post = async (req, res) => {
  let id = uuidv4(),
  {
    id_keluarga, id_lingkungan, ketua_lingkungan, id_umat, pendidikan, id_ortu, keperluan,
    isKetuaLingkungan,
  } = req.body,
  created_at = getTodayDate(),
  ketua_lingkungan_approval = 0,
  ketua_lingkungan_approval_stamp = null,
  kode_lingkungan = await getKodeLingkungan(id_lingkungan)

  let no_surat = await generateNomorSurat('F3', kode_lingkungan, tableName)

  if(isKetuaLingkungan === true) {
    ketua_lingkungan_approval = 1
    ketua_lingkungan_approval_stamp = getDateTime()
  } else {
    ketua_lingkungan = null
  }

  try {
    let sql = `INSERT INTO ${tableName} SET ?`
    let result = await db(sql, [ {
      id, no_surat, id_keluarga, id_lingkungan, ketua_lingkungan, id_umat, pendidikan, id_ortu,
      keperluan, ketua_lingkungan, ketua_lingkungan_approval, ketua_lingkungan_approval_stamp,
      created_at } ])

    // Catat ke log surat
    LogSuratcontroller.post(id, 0, 0)

    if(isKetuaLingkungan === true) { LogSuratcontroller.post(id, 2, 1) }

    res.status(200).send({ message: "Success adding data", result: result, })
  } catch (error) {
    console.log(error.message)
    res.status(500).send({ message: "Failed adding data", error: error.message, })
  }
}

```

**Gambar 3. Kode Untuk Membuat Surat Baru (Surat Keterangan)**

Data umat/anggota keluarga yang disimpan sebenarnya hanya id umat sebagai Foreign Key. Data pokok umat tidak disimpan ke tabel surat. Jadi, Data diambil melalui relasi tabel surat dan tabel umat. Id lingkungan yang dicatat di tabel surat adalah id lingkungan tempat umat yang dipilih tinggal. Nomor surat merupakan hasil generate sistem. Nomor surat memiliki struktur yang kurang lebih sama yaitu *015.kode surat.kode lingkungan/nomor surat/bulan/tahun*. Perbedaan ada pada kode surat dan kode lingkungan. Nomor surat juga di-*reset* saat berganti tahun. Berikut adalah potongan kode generate surat:

```

const generateNomorSurat = async (kodeSurat, idLingkungan, tableName) => {
  // Get waktu saat ini untuk menentukan bulan dan tahun
  let ts = Date.now(),
      date_ob = new Date(ts),
      month = date_ob.getMonth() + 1,
      year = date_ob.getFullYear(),
      lastNumber = await getJumlahSuratBasedOnTahun(tableName, year)

  // Pengubahan bulan menjadi 2 digit jika tidak 2 digit
  idLingkungan = idLingkungan < 10 ? `0${idLingkungan}` : idLingkungan
  month = month < 10 ? `0${month}` : month

  // 015 adalah kode Paroki Kumetiran
  return `015.${kodeSurat}.${idLingkungan}/${lastNumber + 1}/${month}/${year}`
}

const getJumlahSuratBasedOnTahun = async (tableName, year) => {
  const sql = `SELECT COUNT(id) AS jumlah
              FROM ${tableName}
              WHERE YEAR(created_at) = ${year}`

  let result = await db(sql)

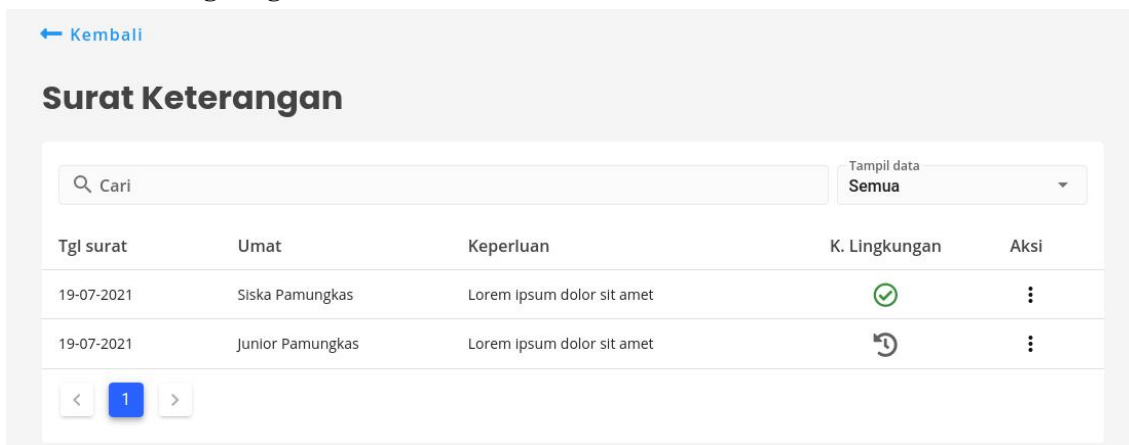
  return result[0].jumlah
}

```

**Gambar 4. Kode Untuk Generate Nomor Surat (Surat Keterangan)**

Pada Gambar 4 terdapat dua fungsi yang menjadi basis pembuatan nomor surat. Fungsi pertama adalah `generateNomorSurat()` dan `getJumlahSuratBasedOnTahun()`. Fungsi `generateNomorSurat()` akan menerima parameter `kodeSurat` misal F1, F2, F3, dan seterusnya. Kemudian fungsi akan menentukan bulan dan tahun pembuatan surat menggunakan fungsi bawaan Javascript. Kemudian memanggil fungsi `getJumlahSuratBasedOnTahun()` dan memberi argumen nama tabel dan tahun pembuatan surat. Fungsi tersebut akan menghitung jumlah semua surat sesuai nama tabel dan tahun. Sehingga didapat nomor surat terakhir pada tabel tersebut. Jumlah surat terbaru didapat dari nomor surat terakhir ditambah satu. Semua data di-concat dan dikembalikan ke fungsi `post()` pada *controller* surat.

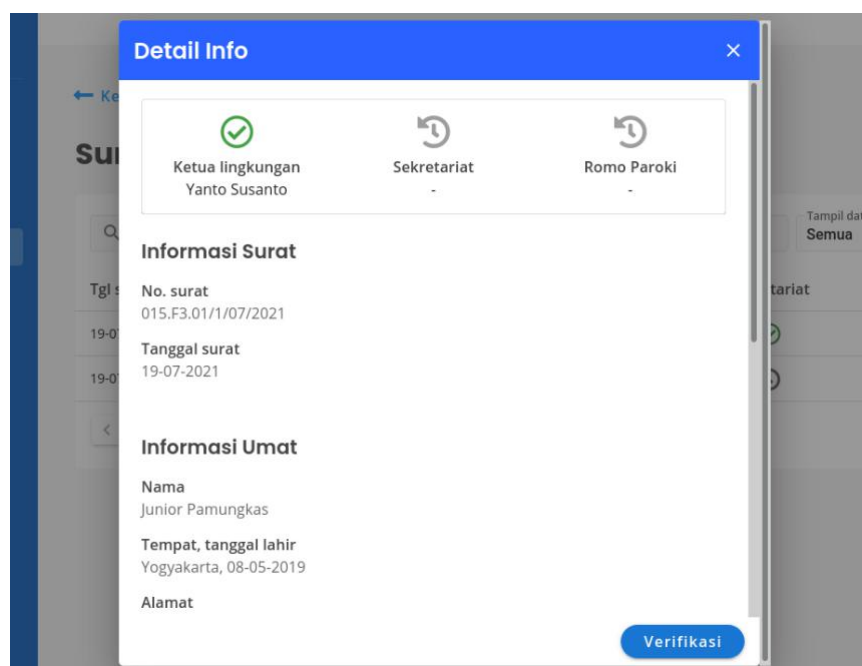
### 4.3.2. Ketua Lingkungan



**Gambar 5. Antarmuka Manajemen Verifikasi Surat Ketua Lingkungan**

Gambar 5 merupakan antarmuka halaman manajemen verifikasi surat yaitu Surat Keterangan pada akun Ketua Lingkungan. Halaman ini menampilkan semua surat yang memiliki id lingkungan yang sama dengan id lingkungan dari lingkungan yang diketuai oleh Ketua Lingkungan terkait (surat dari umat yang tinggal di lingkungan yang diketuai Ketua Lingkungan tersebut). Terdapat fitur pencarian dan filter. Filter dibedakan menjadi tiga yaitu semua, belum diverifikasi, dan sudah diverifikasi.

### 4.3.3. Pengurus (Sekretariat dan Romo Paroki)



**Gambar 6. Antarmuka Manajemen Verifikasi Surat Keterangan Pada Akun Sekretariat**

Gambar 6 merupakan antarmuka komponen detail informasi surat, yaitu Surat Keterangan, pada akun Sekretariat. Komponen ini menyediakan informasi surat yang dibuat Kepala Keluarga termasuk status verifikasinya. Verifikasi dilakukan dengan menekan tombol verifikasi. Antarmuka pada akun Romo Paroki maupun Romo sebenarnya mirip. Perbedaannya ada pada level verifikasi saja.

**Gambar 7. Antarmuka Atur Jadwal Baptis Anak**

Surat Baptis Anak maupun Surat Baptis Dewasa dapat terverifikasi Sekretariat setelah Sekretariat menentukan jadwal baptis dan Romo Pembaptis. Sekretariat akan dialihkan ke halaman atur jadwal baptis sehingga antarmuka pada Gambar 5 muncul. Bagian kiri merupakan kalender yang menyediakan informasi jadwal baptis yang sudah ada dan info jadwal baptis saat jadwal diklik. Sementara bagian kanan merupakan form pilih jadwal. Sekretariat dapat menyimpan data jadwal dengan menekan tombol “Simpan”. Jadwal akan tersimpan, surat akan terverifikasi, dan data tanggal baptis yang ada pada tabel *Detail\_Umat* akan berubah.

#### 4.4. Pengujian Fungsionalitas Perangkat Lunak

Untuk menguji kehandalan perangkat lunak, dilakukan pengujian dengan metode *black box testing*. Terdapat 71 *test case* yang menguji kesesuaian ekspektasi *output* dan *output* sebenarnya. Hasil pengujian menunjukkan bahwa sistem mengeluarkan *output* sesuai ekspektasi.

#### 4.5. Hasil Pengujian Terhadap Pengguna

Pengujian terhadap pengguna dilakukan kepada dua orang Sekretariat Paroki dan empat orang dari divisi Litbang Paroki Hati Santa Perawan Maria Tak Bercela Kumetiran dengan metode wawancara. Berikut adalah hasil wawancara pada saat pengujian:

**Tabel 1. Pengukuran *Availability* (Han)**

<i>Fitur</i>	<i>Komentar &amp; saran</i>
Kelola Anggota keluarga	Antarmuka mudah dipahami, penambahan maupun ubah data mudah dilakukan
Kelola Surat secara keseluruhan	Antarmuka mudah dipahami, pengajuan surat mudah dilakukan, koreksi dari Ketua lingkungan mudah dibaca
Manajemen verifikasi secara keseluruhan	Antarmuka mudah dipahami, disposisi surat jelas, proses verifikasi mudah, akan lebih baik jika disposisi surat tidak berhenti hanya sampai lingkup pengurus tetapi dapat dilanjutkan kepada pihak diluar pengurus (selain Ketua Lingkungan, Sekretariat, Romo Paroki, dan Romo)

## 5. Kesimpulan dan Saran

### 5.1. Kesimpulan

Berdasarkan hasil analisis, pembahasan, dan pengujian terhadap aplikasi Sigrejak, dapat disimpulkan bahwa sistem yang dibangun memudahkan penanganan surat-menyurat. Sistem dapat mengakomodir pengajuan dan disposisi surat dari Ketua Lingkungan, Sekretariat, hingga Romo Paroki. Umat tidak perlu datang ke Ketua Lingkungan untuk membuat surat sehingga menghemat waktu dan tenaga. Ketua Lingkungan dapat memeriksa surat dari umat dan memberi umpan balik. Sekretariat dan Romo Paroki dapat memverifikasi surat kapan saja dan dimana saja. Sistem juga dapat mengakomodir penjadwalan dan penyebaran informasi jadwal baptis. Sistem dapat mencegah terjadinya redundansi data serta menjaga kesamaan format data.

### 5.2. Saran

Saran yang dapat diberikan penulis untuk penelitian selanjutnya adalah sebagai berikut:

(1) Penambahan jenis Role pengurus dibuat agar bisa ditambahkan sehingga role pengurus pada sistem tidak hanya Ketua Lingkungan, Sekretariat Paroki, Romo, dan Romo Paroki. Supaya disposisi dapat berlanjut ke role baru tersebut sesuai kebutuhan pada masa mendatang. (2) Pembuatan antarmuka khusus untuk menampilkan daftar surat dan detail surat bagi akun pengurus dengan role baru seperti dirujuk pada saran nomor 1.

## Referensi

- [1] H. Nugroho, "Pengaruh It Dan Strategi Sistem Informasi Pada Pola Kerja Organisasi," *Konf. Nas. Sist. Inf.*, no. December, p. 2012, 2016.
- [2] "Arti kata umat - Kamus Besar Bahasa Indonesia (KBBI) Online."
- [3] <https://kbbi.web.id/umat> (accessed Jan. 14, 2021).
- [4] "Paroki Cengkareng, Trinitas | Apa itu Paroki?" <https://trinitas.or.id/gereja-dan-paroki/121-apa-itu-paroki.html> (accessed Jan. 14, 2021).
- [5] H.M. Jogiyanto, *Analisa dan Desain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktik Aplikasi Bisnis*. Yogyakarta, Indonesia: Andi, 2005.
- [6] A.Vriska, Suprihadi, Skripsi, Prog. Studi. T. Inf., Fak. Tek. Infor. Univ. Kristen Satya Wacana, "Perancangan Sistem Informasi Data Jemaat Berbasis Partisipatif Jemaat Menggunakan Framework Codeigniter (Studi Kasus: Gereja Kristen Injili di Tanah Papua Jemaat GKI Betlehem Wamena)," Salatiga, Indonesia, Apr., 2015.
- [7] R. D. Axel, X. Najoan, B. A. Sugiarto, J. T. Elektro-ft, and M. Manado, "Rancang Bangun Aplikasi Berbasis Android Untuk Informasi Kegiatan Dan Pelayanan Gereja," *J. Tek. Elektro dan Komput.*, vol. 6, no. 1, pp. 1–6, 2017.
- [8] B. P. Nugroho and S. Jayanti, "Implementasi Sistem Informasi Berbasis Web (Studi Kasus Gereja GKE Sion Palangkaraya)," *J. SAINTEKOM*, vol. 7, no. 2, p. 138, 2017, doi: 10.33020/saintekom.v7i2.44.
- [9] Sugiarto, "Metode pengumpulan data primer," di *Teknik Sampling*. Jakarta, Indonesia: Gramedia Pustaka Utama, 2001.
- [10] K. Mohd. Ehmer and K. Farneena, "A Comparative Study of White Box , Black Box and Grey Box Testing Techniques," *Int. J. Adv. Comput. Sci. Appl.*, vol. 3, no. 6, pp. 12–15, 2012.