

## **BAB 2**

### **TINJAUAN PUSTAKA**

#### **1.1 Pemilihan Umum (Pemilu) Tradisional**

Indonesia merupakan salah satu negara yang menjunjung tinggi demokrasi, dengan menyelenggarakan pemilihan presiden dan wakil presiden, pemilihan legislatif bahkan pemilihan kepala desa dilakukan secara *voting*. *Voting* mengacu pada cara untuk menentukan pilihan dengan mencoblos atau pun mencontreng pilihannya. Kegiatan ini berlangsung pada waktu-waktu tertentu. Di Indonesia, pemilihan umum dilakukan sesuai dengan Undang-Undang Nomor 7 Tahun 2017 tentang pemilihan umum. Dinyatakan dengan jelas bahwa pemilihan dilakukan dengan prinsip keterusterangan, universalitas, kebebasan, kerahasiaan, kejujuran dan keadilan. Dan dalam menyelenggarakan pemilu, penyelenggara pemilu harus melaksanakan Pemilu berdasarkan pada asas sebagaimana dimaksud, dan penyelenggaraannya harus memenuhi prinsip: a. Mandiri; b. Jujur; c. Adil; d. Berkepastian hukum; e. Tertib; f. Terbuka; g. Proporsional; h. Profesional; i. Akuntabel; j. Efektif; dan k. Efisien (UU, 2017).

Direktur Tata Negara Ditjen Administrasi Hukum Umum (AHU), Kartiko Nurintias (2019), mengatakan bahwa pemilu tradisional memiliki kelebihan dan kekurangan. Keunggulan sistem ini adalah menghasilkan kepala daerah yang memiliki kedekatan kepada rakyat, mendorong majunya calon kepala daerah yang kredibel, partisipasi rakyat lebih terlibat, serta bisa diterapkannya prinsip *one man one vote*. Sedangkan kelemahan dari penyelenggaraan PEMILU tradisional ini yaitu bisa terjadi bias dimana orang yang memiliki dana besar punya peluang besar terpilih, memerlukan dana cukup besar dalam pelaksanaannya dan memunculkan biaya politik tinggi. Selain itu, adanya praktek politik uang yang dominan dan sudah menghantui setiap proses pilkada serta berpotensi menimbulkan konflik horizontal di tengah masyarakat.

## 1.2 E-Voting

Pada penelitian ini menggunakan sistem pemilihan umum berbasis elektronik (e-Voting). Menurut (Risnanto, 2017), *Electronic voting* adalah suatu metode pemungutan suara dan perhitungan suara dalam suatu pemilihan dengan menggunakan perangkat elektronik. Tujuan dari *electronic voting* adalah menyelenggarakan pemungutan suara dengan biaya hemat dan perhitungan suara yang cepat dengan menggunakan sistem yang aman dan mudah untuk di lakukan audit, e-Voting juga mudah dalam pelaksanaan pemilihan (Hardjaloka & Simarmata, 2011), pemilih tidak harus datang ke tempat pemilihan untuk memilih (Prananda, Anra, & Pratiwi, 2017).

E-Voting mempunyai tantangan tersendiri seperti kepercayaan yang masih rendah terhadap teknologi dan kekhawatiran ada pihak lain yang melakukan manipulasi data (Hutagalung, 2012). Prinsip utama dari E-Voting harus sama dengan pemungutan suara reguler dan sesuai dengan undang-undang dan prinsip-prinsip pemilu. E-Voting harus menawarkan tingkat keamanan dan kepercayaan yang sama dengan pemilihan tradisional. E-Voting harus seragam dan rahasia, hanya orang yang memenuhi syarat yang diizinkan untuk memilih. Setiap pemilih harus dapat memberikan hanya satu suara. Seorang pemilih harus tidak dapat dibuktikan mendukung siapa. Pengumpulan suara harus aman, andal, dan akuntabel. Berbagai masalah dalam penggunaan sistem voting pilkada konvensional, dapat diatasi dengan menerapkan Teknologi Informasi dan komunikasi (TIK) yaitu elektronik voting.

## 2.3 Autentikasi

Dalam menjamin keamanan sistem, maka diperlukan sistem autentikasi yang memadai. Autentikasi adalah jaminan bahwa entitas yang berkomunikasi merupakan entitas yang diklaimnya (Syahputri, 2019). Menurut (Waisantoro, 2014) Melakukan autentikasi terhadap sebuah objek adalah melakukan konfirmasi terhadap kebenarannya. Pada saat yang sama, mengautentikasi seseorang biasanya untuk memverifikasi identitasnya. Pada sistem komputer, otentikasi biasanya dilakukan pada saat login atau permintaan akses.

Adapun Metode autentikasi (Syahputri, 2019) adalah sebagai berikut:

a. *Something you know*

Metode ini mengandalkan kerahasiaan informasi, seperti kata sandi dan PIN. Metode ini berasumsi bahwa tidak ada seorangpun yang mengetahui rahasia itu kecuali orang tersebut.

b. *Something you have*

Metode ini bergantung pada barang-barang unik seperti kartu magnetik / kartu pintar, token perangkat keras, token USB, dan lain sebagainya. Metode ini berasumsi bahwa tidak ada seorangpun yang memiliki barang tersebut kecuali orang tersebut.

c. *Something you are*

Metode ini menghandalkan keunikan bagian-bagian tubuh yang tidak mungkin ada pada orang lain seperti sidik jari, suara atau sidik retina. Cara ini berasumsi bahwa bagian tubuh seperti sidik jari dan sidik retina, tidak mungkin sama dengan orang lain.

d. *Something you do*

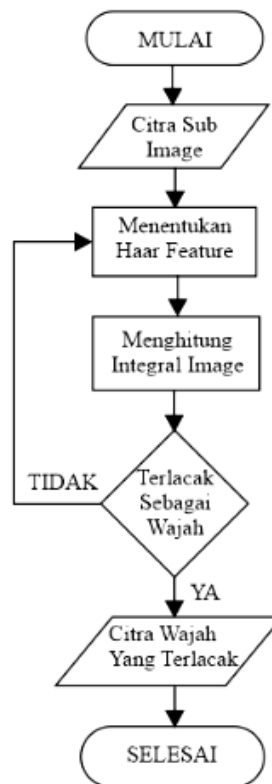
Melibatkan bahwa setiap pengguna melakukan sesuatu secara berbeda. Contoh: Penggunaan analisis suara (*voice recognition*), pengenalan wajah (*face recognition*), dan analisis tulisan tangan.

## 2.4 Face Recognition

Penelitian ini menggunakan sistem pemilihan umum berbasis pengenalan wajah. Pengenalan wajah merupakan salah satu teknologi yang terangkum dalam sistem biometrik. Teknologi ini telah banyak digunakan dalam proses verifikasi dan pengenalan identitas, selain itu juga dapat mengidentifikasi retina mata, sidik jari dan iris mata. Dalam lingkup aplikasi itu sendiri, pengenalan citra wajah ditangkap oleh perangkat lain (yaitu kamera) untuk menangkap citra wajah seseorang, dan kemudian membandingkannya dengan wajah sebelumnya (pelatihan data) yang disimpan dalam database sistem (Angeline dkk., 2019).

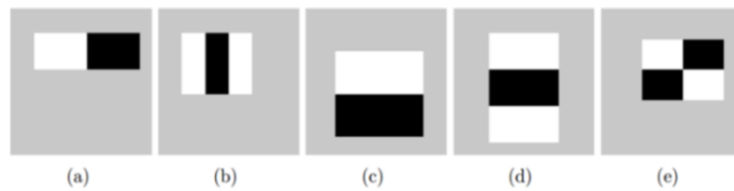
Pada proses pendeteksi wajah digunakan algoritma *haar cascade* (Sayeed & Daniel, 2018). Umumnya, fungsi seperti *Haar-Like feature* digunakan untuk

mendeteksi objek dalam gambar digital. Istilah *Haar-Like feature* dinyatakan sebagai fungsi matematis dalam bentuk kotak (*Haar Wavelet*), dan prinsipnya sama dengan fungsi *Fourier*. Berikut adalah alur proses metode Haar Cascade Classifier:



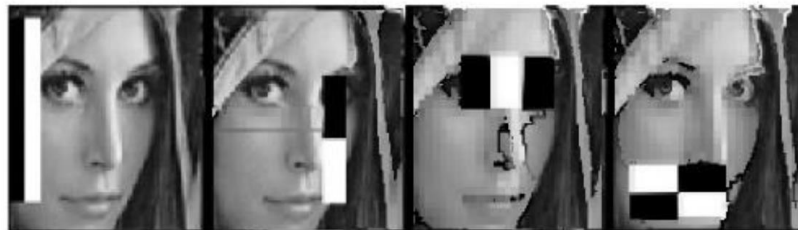
**Gambar 2.1** Alur proses metode *Haar Cascade Classifier*.

Pada awalnya pengolahan citra dilakukan hanya dengan melihat nilai RGB dari setiap piksel, namun metode ini terbukti tidak efektif. Ini kemudian dikembangkan oleh Viola dan Jones untuk membentuk fitur mirip *Haar-Like feature*. Kemudian, Viola dan Jones mengembangkan dan membentuk fungsi Haar-like feature yaitu memproses gambar dalam kotak-kotak, di mana ada beberapa piksel dalam satu kotak. Selanjutnya kotak tersebut diproses untuk menghasilkan perbedaan nilai yang menunjukkan area terang dan gelap. Nilai-nilai tersebut nantinya akan digunakan sebagai dasar untuk pengolahan citra. Metode penghitungan nilai fungsi ini adalah dengan mengurangkan nilai piksel area putih dari piksel area hitam.



**Gambar 2.2** Fitur *Haar*.

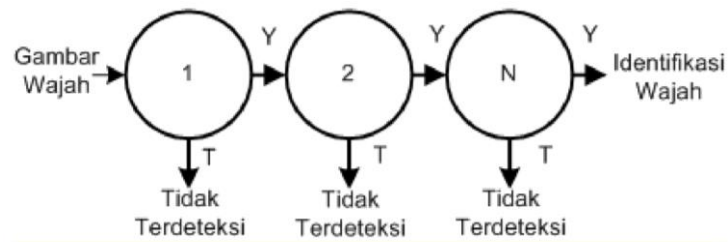
Untuk mendapatkan perbedaan kondisi tingkat kecerahan, seluruh citra harus dalam bentuk nilai rata-rata yang dinormalisasi dari perubahan sebelumnya. Nilai varians dari semua gambar ini lebih rendah dari gambar lainnya memiliki sedikit informasi maka akan dibuang dari penilaian.



**Gambar 2.3** Contoh Pencarian Dengan *HaarLike Feature*.

Media berupa citra integral digunakan algoritma Haar dalam memudahkan proses penghitungan nilai fitur. Citra integral merupakan citra yang nilai setiap pikselnya adalah penjumlahan dari nilai piksel kiri atas dan nilai piksel kanan bawah. Misalnya, piksel (a, b) memiliki nilai akumulasi dari semua piksel (x, y). Dimana  $x \leq a$  dan  $y \leq b$ .

Saat menggunakan metode *haar cascade*, beberapa jenis citra dapat diolah, salah satunya *grayscale*. *Cascade Classifier* adalah langkah penghitungan berulang kali nilai *Haar Feature* untuk mendapatkan hasil yang lebih akurat. Gambar berikut menampilkan alur kerja dari *Cascade Classifier*. Pada klasifikasi tahap pertama, setiap sub-citra akan diklasifikasikan dengan fitur, jika hasilnya tidak memenuhi standar maka hasilnya akan ditolak. Pada klasifikasi tahap 2, setiap sub-citra akan direklasifikasi. Jika ambang batas yang diperlukan diperoleh, tahap filter berikutnya (tahap klasifikasi 3) dimasukkan. Hingga sub-citra yang lolos akan berkurang hingga mendekati citra yang sesungguhnya.



**Gambar 2.4** Metode Deteksi *Cascade Classifier*.

Pengenalan wajah adalah proses lanjutan dari deteksi wajah. Pada deteksi wajah yang mendeteksi sebagian dari wajah seseorang, wajah tersebut dapat diperoleh dari sebuah gambar atau video. Dengan menggunakan hasil pelatihan dari *haar cascade*. Hasil dari proses ini kemudian digabungkan dengan proses *Image Matching* menggunakan algoritma *Local Binary Pattern Histogram (LBPH)*. Dengan metode ini, foto yang dipelajari akan dicocokkan dengan hasil deteksi aliran kamera. Dalam stream ini, nilai histogram yang diekstrak dari citra kemudian digunakan untuk mencocokkan pola persamaan histogram dengan beberapa citra dalam database.

Ciri utama pengenalan wajah menggunakan metode ini adalah komposisi pola tekstur mikro, yang merupakan operator non-parametrik yang digunakan untuk mendeskripsikan tata letak lokal citra. LBPH diartikan sebagai perbandingan nilai biner dari piksel di tengah gambar dengan nilai sekitar 8 piksel. Misalnya, pada gambar 3x3, nilai biner di tengah gambar dibandingkan dengan nilai di sekitarnya. Dengan mengurangi nilai piksel di tengah gambar dan nilai piksel di sekitarnya, jika hasilnya lebih besar dari atau sama dengan 0, atur nilainya menjadi 1; jika hasilnya kurang dari 0, setel nilainya menjadi 0. Setelah itu, susun 8 nilai biner searah jarum jam dan sebaliknya, dan ubah 8 digit biner menjadi nilai desimal untuk mengganti nilai piksel di tengah gambar. Rumus untuk mencari ruang biner dan nilai LBPH adalah sebagai berikut.

Threshold LBPH = nilai sekeliling - nilai pixel tengah

Setelah menyusun biner searah jarum jam, jika salah satu kotak ambang biner adalah 1, masukkan nilai biner sesuai levelnya, tetapi jika 0, hasilnya juga sama dengan 0. Terakhir, tambahkan nilai LBP. Untuk mencocokkan wajah

pemilik, digunakan persamaan untuk mendapatkan perkiraan nilai histogram yang akan digunakan sebagai nilai prediksi untuk mengidentifikasi pemilik wajah. Ini adalah persamaan yang digunakan untuk mencari perkiraan nilai histogram.

$$D = \sqrt{\sum_{i=1}^n (hist\ 1_i - hist\ 2_i)^2}$$

Nilai D dipergunakan sebagai pembanding antara wajah yang terdapat di database dan wajah yang dideteksi dengan kamera.

Ada 2 jenis perbandingan wajah, yakni perbandingan 1: 1 dan perbandingan 1: N. Untuk mencoba mengenali hanya satu orang dari sekumpulan orang (seperti di aplikasi buka kunci wajah), maka yang yang diperlukan adalah perbandingan 1: 1. Jika mengidentifikasi sekelompok orang, maka yang yang diperlukan adalah pengenalan wajah 1:N. Untuk melakukan Perbandingan 1:1 maka dibutuhkan bantuan *Face Match API* agar dua wajah dapat dibandingkan untuk menentukan apakah merupakan milik orang yang sama atau bukan. *Face Match API* juga biasa disebut sebagai *Face Authentication API* karena berfungsi untuk mengautentikasi pengguna.

*Face Match API* adalah ekstensi dari *FaceVector API*. API merespons kembali dengan jarak kecocokan antara dua gambar yang dikirim ke server. Bergantung pada jarak *Euclidean* ini, API merespons jika gambar cocok atau tidak. Jika *Euclidean Distance* dari citra kurang dari ambang 0,5, dapat disimpulkan bahwa wajah adalah sesuai dan secara alami diatas 0,5 dapat dianggap tidak sesuai.

## 2.5 Aplikasi Mobile

Dalam mengakses informasi melalui perangkat seluler, biasanya akan membuka aplikasi untuk mencari informasi yang dibutuhkan. Aktivitas tersebut secara tidak langsung menggunakan aplikasi mobile di perangkat seluler. Menurut (Pressman dan Bruce, 2014) Aplikasi mobile adalah sistem operasi yang dibuat secara khusus untuk platform seluler (seperti, iOS, Android, atau Windows Mobile). Dalam banyak kasus, antarmuka pengguna aplikasi seluler memiliki mekanisme interaksi unik yang disediakan oleh platform seluler,

interoperabilitas dengan sumber daya berbasis web (akses ke berbagai informasi terkait aplikasi) dan untuk pengumpulan, analisis, dan fungsi pemrosesan lokal yang diformat. Informasi yang paling cocok untuk platform seluler. Selain itu, aplikasi seluler memberikan kemampuan penyimpanan yang persisten dalam platform.

## 2.6 Sistem Operasi Android

### 2.6.1. Pengertian Android

Hampir setiap orang membutuhkan perangkat android untuk berkomunikasi, mencari informasi dan berita terbaru, melakukan pekerjaan dan berbagai aktivitas lainnya. Menurut (Nazruddin Safaat H. 2015) Android merupakan sistem operasi seluler berlandaskan *Linux*, yang meliputi sistem operasi, *middleware*, dan aplikasi. Android memberi pengembang platform terbuka untuk membuat aplikasi mereka. Android adalah generasi baru platform seluler yang dapat menyediakan pengembangan yang diinginkan oleh pengembang. Adapun ikon dari sistem operasi Android ini sering disebut juga dengan sebutan robot hijau, berikut adalah gambar dari robot hijau tersebut :



**Gambar 2.5** Ikon dari sistem operasi Android.

### 2.6.2. Kelebihan Sistem Operasi Android

Adapun beberapa kelebihan dari sistem operasi Android adalah sebagai berikut (Safaat, 2011):

- a. *Complete Platform* Sistem operasi Android adalah sistem operasi yang menyediakan banyak alat berguna untuk membangun aplikasi, dan pengembang dapat mengembangkannya lebih lanjut.



- b. *Open Source Platform* Android yang bersifat open source menjadikan sistem operasi ini mudah dikembangkan oleh para developer karena bersifat terbuka.
- c. *Free Platform Developer* yaitu dapat dengan bebas memperluas, membagikan, mempromosikan tanpa menuntaskan anggaran lisensi.

## 2.7 Metode Pengembangan Perangkat Lunak

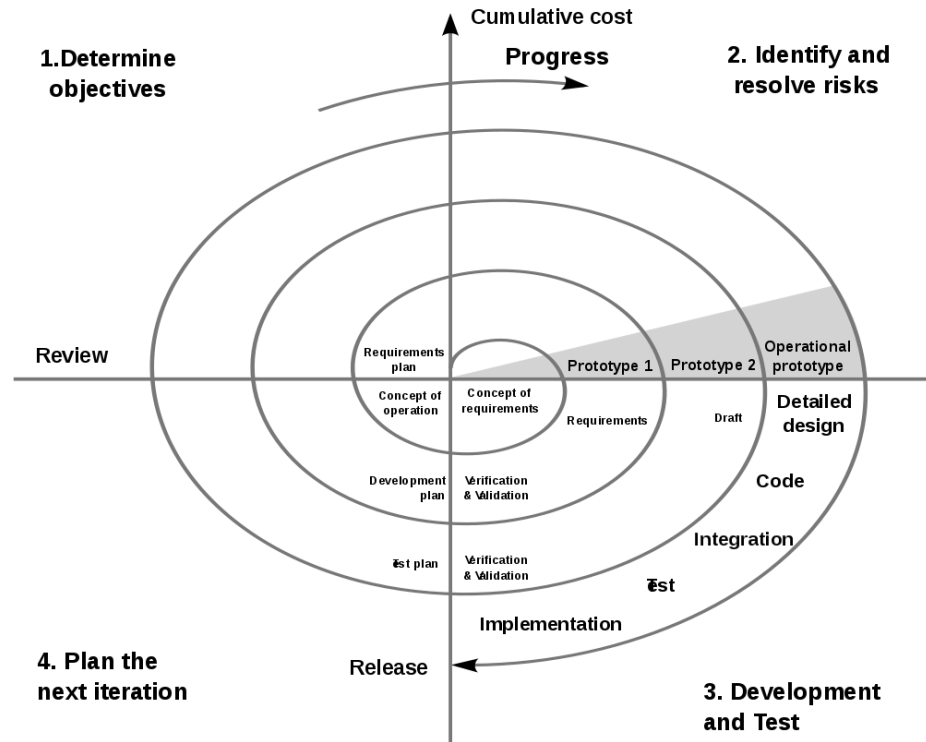
Metode pengembangan sistem informasi adalah metode yang digunakan dalam mengembangkan sistem berdasarkan dengan proses-proses yang berjalan secara runtut hingga didapatkan sistem yang sesuai.

### 2.7.1 Metode Spiral

Pada penelitian ini, metode pengembangan sistem informasi yang digunakan adalah metode spiral. Model spiral atau metode spiral adalah bentuk perkembangan yang menggunakan metode *iteratif natural model prototype* dan mengintegrasikannya dengan segi sistematis eskalasi model *waterfall*. Tahapan *prototype* adalah model dimana perangkat lunak menjadi *prototype* (model tidak lengkap), dan contoh diperlihatkan kepada pengguna atau pelanggan untuk mendapatkan umpan balik. Jika *prototype*-nya sudah sesuai dengan keinginan user atau customer, maka proses dilanjutkan dengan membuat produk sesungguhnya dengan menambah dan memperbaiki kekurangan dari *prototype* tadi.

Model tersebut juga memadukan desain top-down dengan desain bottom-up, dimana desain top-down pertama kali mendefinisikan sistem global kemudian dilanjutkan dengan detail sistem, sedangkan desain bottom-up menerapkan konsep sebaliknya. *Desain top-down* biasanya diterapkan pada model *waterfall* dengan urutan, sedangkan *desain bottom-up* biasanya diterapkan untuk mendapatkan umpan balik dari dua kombinasi ini (yaitu desain dan prototipe dan kombinasi atas-bawah dan bawah) Model *prototype*. Hal ini juga berlaku untuk model *waterfall* dan *prototype*, maka model spiral ini bisa dikatakan model proses yang menggabungkan dua

model. Oleh karena itu, model ini biasanya digunakan untuk pengembangan perangkat lunak yang besar dan kompleks.



**Gambar 2.6** Metode Spiral.

Metode Spiral Dari gambar tersebut, proses dimulai dari inti bergerak searah dengan jarum jam mengelilingi spiral. Lintasan pertama putaran menghasilkan perkembangan spesifikasi produk. Putaran berikutnya digunakan untuk mengembangkan *prototype* dan secara bertahap mengembangkan versi perangkat lunak yang lebih kompleks. Setiap jalur melalui area perencanaan akan menyesuaikan dengan rencana proyek. Sesuaikan biaya dan jadwal berdasarkan umpan balik yang dikumpulkan dari evaluasi pelanggan.

Pada model spiral, kita dapat melakukan dua jenis perubahan, yaitu sebagai berikut:

#### 1. Perubahan besar

Ketika pelanggan meminta perubahan besar dalam persyaratan untuk modul tertentu, maka kami hanya mengubah modul itu dan melakukan

pengujian untuk integrasi dan unit. Dan untuk ini, kami selalu memilih satu siklus baru karena dapat mempengaruhi modul yang ada. Perubahan besar dapat berupa fungsionalitas perangkat lunak.

## 2. Perubahan kecil

Setiap kali klien meminta perubahan kecil dalam aplikasi tertentu, maka tim perangkat lunak membuat perubahan kecil bersama dengan modul baru yang harus dikembangkan secara bersamaan dalam satu siklus. Dan kami tidak pernah menggunakan siklus atau iterasi baru karena variasi kecil tidak memengaruhi fungsionalitas yang ada, dan juga membutuhkan sumber daya dan waktu ekstra. Perubahan kecil dapat berupa UI (perubahan *frontend*).

### 2.7.2 Keuntungan dan Kelemahan Metode Spiral

#### a. Keuntungan Metode Spiral

1. Perubahan fleksibel diperbolehkan dalam model spiral.
2. Perkembangan tersebut dapat didistribusikan menjadi bagian-bagian yang lebih kecil.
3. Pelanggan juga dapat menggunakan aplikasi pada tahap awal.
4. Kejelasan lebih lanjut untuk pengembang dan teknisi pengujian.
5. Ini akan memberikan penggunaan prototipe secara luas.

#### b. Keuntungan Metode Spiral

1. Ini tidak cocok untuk produk kecil dan berisiko rendah karena bisa mahal untuk proyek yang lebih kecil.
2. Ini adalah model tradisional, dan dengan demikian pengembang hanya melakukan pekerjaan pengujian juga.
3. Tidak ada persyaratan proses peninjauan dan tidak ada kiriman paralel yang diperbolehkan dalam model spiral.
4. Dalam model spiral, manajemen agak sulit; itulah mengapa ini merupakan proses yang kompleks.
5. Jumlah maksimum fase menengah membutuhkan dokumen yang tidak perlu.

## 2.8 Tools yang dipergunakan

Dalam penelitian ini, ada beberapa *tools* yang digunakan. Adapun *tools* tersebut adalah sebagai berikut:

### 2.8.1 Android Studio

Menurut Falker (2013), Android Studio adalah IDE Google, yang diperkenalkan pada acara Google I / O pada Mei 2013, dan merupakan IDE. Alternatif untuk Eclipse IDE. Di situs resminya disebutkan bahwa Android Studio adalah IDE resmi untuk mengembangkan aplikasi android berdasarkan kecerdasan IDEA. Pada proses pembuatan aplikasi, penulis menggunakan software Android Studio 2.3.3, Java Development Kit (JDK) versi 7, Standard Development Kit (SDK) versi 14. Adapun struktur Android Studio, adalah sebagai berikut:

#### a. Java

Java adalah bahasa konkuren, berdasarkan kelas, berorientasi objek, dibuat sederhana, dan banyak programmer dapat menggunakannya.

#### b. Resources

Struktur Resources adalah struktur dalam android studio yang berfungsi sebagai tempat desain sebuah aplikasi yang akan dibuat. Sumber daya folder dapat ditemukan di direktori app / res, yang berisi:

- Folder Drawable berfungsi sebagai tempat meletakkan gambar yang kita gunakan.
- Folder tata letak berfungsi sebagai tempat xml untuk desain yang digunakan.
- Menu folder dirancang sebagai penahan xml untuk bilah operasi.
- Digunakan untuk menempatkan folder nilai konfigurasi statis, misalnya Tali, warna, ukuran dan gaya.c. Android Manifest.xml

Manifes Android adalah xml yang digunakan untuk mengatur segala sesuatu yang terjadi dalam aplikasi. Semua aktivitas harus dilaporkan ke struktur utama androidmanifest.xml. Judul, ikon dan versinya juga dilaporkan di sini.

#### d. Intent

Intent adalah pesan yang digunakan untuk mengaktifkan tiga komponen dasar aplikasi Android, yaitu aktivitas, layanan, dan penerima siaran. Aktivasi komponen ini dapat dilakukan dalam aplikasi yang sama atau berbeda, seperti menjalankan "aktivitas", "memulai layanan", atau mengirim pesan ke penerima siaran. Ketika ada komunikasi antar komponen, Intent akan menyimpan paket informasi yang digunakan dalam proses tersebut.

### 2.8.2 Codeigniter

CodeIgniter merupakan jaringan aplikasi web open source yang dikembangkan untuk menciptakan aplikasi php dinamis. CodeIgniter adalah framework PHP dengan model MVC (model, view, controller), yang dipakai untuk membuat website dinamis menggunakan PHP, yang dapat memudahkan developer dalam membuat aplikasi web. Selain ringan dan cepat, CodeIgniter juga mempunyai dokumentasi dan contoh penerapan kode yang super lengkap. Dokumen lengkap ini merupakan alasan fundamental mengapa banyak orang memilih CodeIgniter sebagai kerangka kerja pilihan.

### 2.8.3 Visual Studio Code

Visual Studio Code adalah editor kode sumber yang sangat ringan namun kuat yang dapat dijalankan dari desktop. Ini memiliki dukungan built-in untuk JavaScript, scripting, dan Node.js, dan memiliki beberapa ekstensi yang tersedia untuk bahasa lain, termasuk C ++, C #, Python, dan PHP. Ini didasarkan pada Github's Electron, yang merupakan versi lintas platform dari Atom, komponen pengeditan kode berdasarkan JavaScript dan HTML5. Editor adalah lingkungan pengembangan terintegrasi (IDE) berfitur lengkap yang dirancang untuk pengembang yang menggunakan teknologi cloud terbuka Microsoft. Visual Studio Code menggunakan alat NET open source untuk memberikan dukungan untuk kode ASP.NET C # dan membangun alat untuk pengembang Omnisharp NET dan kompilator Roslyn. Antarmuka didasarkan pada gaya penjelajah umum, sehingga

mudah digunakan. Panel kiri menunjukkan semua file dan folder yang Anda miliki aksesnya, dan panel editor kanan menampilkan konten file yang terbuka. Dalam hal ini, editor berfungsi dengan baik dan enak dipandang. Ia juga memiliki fungsi yang baik, dengan intellisense dan pelengkapan otomatis, dan dapat digunakan untuk JSON, CSS, HTML, {less} dan Node.js.

Visual Studio Code dirancang untuk digunakan dengan alat yang ada, dan Microsoft menyediakan dokumentasi untuk membantu pengembang bersama, serta bantuan untuk digunakan dengan ASP.NET 5, Node.js dan Microsoft Script, dan alat yang dapat digunakan untuk membantu. Bangun dan kelola aplikasi Node.js. Visual Studio Code sebenarnya ditujukan untuk pengembang JavaScript, yang membutuhkan alat pengembangan lengkap untuk skrip sisi server, dan mungkin ingin mengambil risiko meninggalkan Node.js untuk menggunakan kerangka kerja berbasis .NET. Visual Studio Code merupakan editor kode rute platform yang ringan tetapi kredibel, siapa pun dapat memakainya untuk membuat aplikasi web.

#### **2.8.4 Xampp**

XAMPP merupakan perangkat lunak gratis yang membantu banyak sistem operasi dan mewujudkan himpunan dari beberapa program. Fungsinya sebagai server mandiri (localhost), yang mencakup program Apache HTTP Server, database MySQL, dan penerjemah bahasa yang ditulis dalam bahasa pemrograman PHP dan Perl. Nama XAMPP adalah akronim dari X (4 sistem operasi yakni Apache, MySQL, PHP dan Perl). Program ini tersedia di bawah GNU (Lisensi Publik Umum) dan gratis, Ini adalah server web yang mudah digunakan yang dapat memberikan tampilan web dinamis.

#### **2.8.5 MySql**

MySQL dapat digunakan untuk membuat dan mengelola database beserta isinya. MySQL dapat dimanfaatkan untuk menambahkan, merubah dan menghapus data yang terdapat di dalam database. MySQL adalah sistem manajemen database yang bersifat relasional. Artinya data

yang dikelola dalam database akan ditempatkan pada beberapa tabel terpisah, sehingga pengolahan data akan lebih cepat.

#### **2.8.6 Facex.io**

FaceX adalah sistem yang menyediakan platform bagi perusahaan untuk mengimplementasikan Pengenalan Wajah ke dalam aplikasi mereka dengan mudah. Fleksibilitasnya memungkinkan pengembang untuk mengintegrasikan API dan SDK Pengenalan Wajah Akurasi Tinggi hanya dengan beberapa baris kode. Adapun fitur yang disediakan adalah sebagai berikut:

1. API's (Permintaan HTTP POST untuk Deteksi Wajah, Pengenalan & Deteksi Landmark dari Gambar)
2. Mobile SDKs (Integrasikan berbagai fungsi pengenalan wajah ke aplikasi seluler Anda dengan mudah)
3. Docker Images (Solusi Pengenalan Wajah Lokal melalui Instans Docker)
4. Custom Developments (Perancangan solusi pengenalan Wajah terintegrasi ujung ke ujung yang disesuaikan dengan kasus penggunaan)

Adapun teknologi pengenalan wajah yang disediakan, antara lain:

- Deteksi wajah (Mendeteksi wajah dari foto / video)
- Landmark Wajah (Bergantung pada kebutuhan, Apabila perlu mencari landmark)
- Pengenalan Wajah (Mengidentifikasi orang menggunakan database yang dibuat)
- Deteksi Spoof (Memastikan Keamanan)
- Pelacakan Wajah (Memantau dan melacak gerakan wajah setelah deteksi).

#### **2.8.7 Nox Emulator**

Nox App Player adalah emulator Android yang memungkinkan Anda menggunakan aplikasi sistem operasi Android dengan nyaman. Emulator Android dapat mensimulasikan berbagai ponsel Android, tablet, Wear OS, dan perangkat Android TV di komputer. Nox App Player hadir dengan konfigurasi yang telah ditentukan sebelumnya untuk jenis perangkat umum,

yang dapat mentransfer data lebih cepat daripada perangkat yang terhubung melalui USB. Emulator android ini berjalan di PC, mirip dengan sistem Android.

Setiap instance emulator Android menggunakan perangkat virtual Android (AVD) untuk mengonfigurasi ukuran, faktor bentuk, versi Android, dan berbagai fitur perangkat kerasnya. Untuk menguji aplikasi secara efektif, perlu membuat AVD untuk memodelkan setiap jenis perangkat yang dirancang untuk aplikasi tersebut. Setiap AVD bertindak sebagai perangkat independen dan memiliki penyimpanan khusus untuk menyimpan data pengguna, dan sebagainya.

Emulator Android menyediakan hampir semua fungsi perangkat Android yang sebenarnya. Simulator dapat mensimulasikan panggilan masuk dan pesan teks, menentukan lokasi perangkat, mensimulasikan kecepatan jaringan yang berbeda, mensimulasikan rotasi dan sensor perangkat keras lainnya, mengakses Google Play Store, dan lain-lain.

Beberapa kelebihan Nox player yang belum tentu dimiliki emulator lain adalah sebagai berikut:

1. Nox Emulator lebih cepat dan tidak mudah freeze
2. Dapat diatur menggunakan mode kinerja tinggi
3. Pemutar aplikasi ini dirancang sesuai dengan kernel Android 4.4.2 dan kompatibel dengan sistem X86 (32-bit) dan AMD
4. Respon Speed Knox sangat cepat dan stabil
5. Dapatkan pengalaman bermain game terbaik
6. Manajer file, Facebook Lite, dan aplikasi kamera telah diinstal
7. Dengan fungsi anti shaking (goyang).

### **2.8.8 Postman**

Postman adalah aplikasi yang digunakan sebagai klien REST untuk menguji REST API. Pengembang API sering menggunakan Postman sebagai alat untuk menguji API yang mereka buat. Postman adalah alat yang digunakan dalam proses pengembangan API. Saat ini terdapat banyak fungsi yang berguna untuk proses pengembangan API, termasuk:



- Collection

Sekumpulan permintaan API yang dapat disimpan atau diatur ke dalam folder. Anda dapat dengan mudah mengelompokkan permintaan berdasarkan proyek yang sedang diproses.

- Environment

Konfigurasi yang digunakan untuk menyimpan atribut dan atribut yang dapat digunakan atau dimanipulasi selama permintaan API.

- Response

Pengembang dapat membuat API Mockup sebelum benar-benar menerapkannya ke dalam proyek.

- Mock Server

Dengan menggunakan fungsi ini, Mockup API yang dibuat menggunakan fungsi "Sample Response" dapat diakses dari Internet, sama seperti jika Mockup API telah diimplementasikan dan disebarkan ke server.

- Script Test

Verifikasi fungsionalitas respons, termasuk tes menulis sesuai kebutuhan.

- Automated Test (Runner)





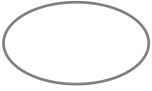
Skip pengujian untuk menjalankan permintaan dalam koleksi secara otomatis.

## 2.9 Unified Modeling Language (UML)

*Unified Modelling Language (UML)* adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk *visualisasi*, merancang dan mendokumentasikan sistem piranti lunak, *UML* menawarkan sebuah standar untuk merancang model sebuah sistem. (Malu. 2015) Tujuan pemakaian *UML* adalah untuk memodelkan suatu sistem yang menggunakan konsep berorientasi objek dan membuat bahasa pemodelan yang dapat digunakan oleh manusia maupun mesin.

## 2.10 Use Case Diagram


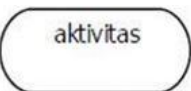



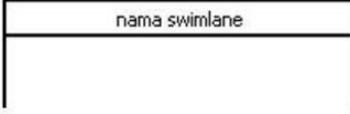
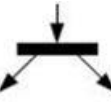
*Use Case Diagram* mendeskripsikan deskripsi level tertinggi dari fungsi spesifik dalam sistem dalam bentuk komponen, event, atau class. Kasus penggunaan ditentukan sebagai urutan operasi terkait (skenario) yang diselesaikan secara otomatis dan dijalankan secara manual untuk menyelesaikan satu tugas bisnis. Diagram use case bersifat statis, menunjukkan sekumpulan use case dan aktor (jenis kelas khusus). Diagram ini sangat penting untuk mengatur dan memodelkan perilaku sistem yang diinginkan dan diharapkan dari pengguna (Afrida dan Wibowo. 2018). Deskripsi simbol lihat Tabel 2.1

No	Gambar	Nama	Keterangan
1		Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case.
2		Include	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (independent).
3		Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.
4		System	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
5		Use Case	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.


**Tabel 2.1** *Use Case Diagram.*

### 2.11 Activity Diagram

Diagram aktivitas menggambarkan rangkaian proses dari aktivitas yang terbentuk dalam operasi, sehingga dapat juga digunakan untuk aktivitas lain, seperti *use case* atau interaksi. Diagram aktivitas direpresentasikan dengan simbol, setiap simbol memiliki arti dan tujuannya. (Utama.2018) Untuk deskripsi simbol grafik aktif, lihat Tabel 2.2.

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir 	Status akhir dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
Fork 	Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel.

**Tabel 2.2** Activity Diagram.

<b>Join</b> 	Digunakan untuk menunjukkan kegiatan yang digabungkan.
--	--

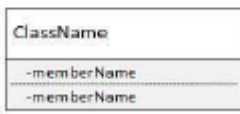


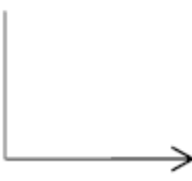
**Tabel 2.3** *Activity Diagram (Lanjutan).*

## 2.12 Class Diagram


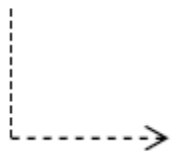

Diagram kelas menggambarkan struktur sistem dari perspektif pendefinisian kelas-kelas yang akan digunakan untuk membangun sistem. Kelas memiliki apa yang disebut properti, metode, atau operasi. (Rosa dan Saladin. 2019) Berikut penjelasan tentang sifat dan cara:

1. Atribut adalah variabel yang dimiliki oleh kelas.
2. Operasi atau metode adalah fungsi yang dimiliki kelas.

Tabel 2.3 mencantumkan deskripsi simbol:

No.	Simbol	Deskripsi
1	<b>Kelas</b> 	Kelas pada struktur sistem.
2	<b>Antarmuka/interface</b> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3	<b>Asosiasi/association</b> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4	<b>Asosiasi berarah/ directed association</b> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .


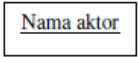

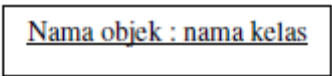

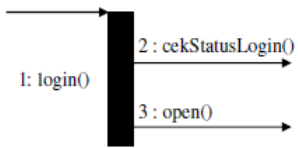
**Tabel 2.3** *Class Diagram.*

5	<b>Generalisasi</b> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umumkhusus).
6	<b>Kebergantungan/dependensi</b> 	Relasi antar kelas dengan makna kebergantungan antar kelas
7	<b>Agrgasi/aggregation</b> 	Relasi antar kelas dengan makna semua-bagian ( <i>whole-part</i> )

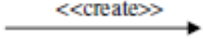
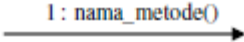
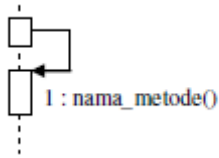
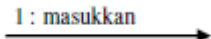
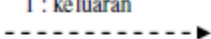
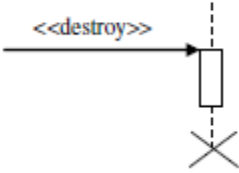
**Tabel 2.3** *Class Diagram*(Lanjutan).

### 2.13 *Sequence Diagram*

Diagram urutan menggambarkan perilaku objek dalam *use case* dengan menggambarkan siklus hidup objek dan pesan yang dikirim dan diterima antar objek. Oleh karena itu, untuk mendeskripsikan diagram sekuens, perlu diketahui objek yang terlibat dalam *use case* dan metode yang dimiliki kelas yang dipakai ke dalam objek. Diagram urutan perlu dibuat untuk melihat skenario dalam *use case*. Jumlah *sequence diagram* yang harus digambar minimal sebanyak definisi *use case* dengan prosesnya sendiri-sendiri, atau yang penting semua *use case* sudah ditentukan, dan interaksi jalur pesan dimasukkan ke dalam *sequence diagram* Sehingga semakin banyak *use case* yang ditentukan maka semakin banyak pula *sequence diagram* yang harus dibuat. (Rosa dan Saladin. 2019) Simbol-simbol pada *sequence diagram* dapat dilihat pada Tabel 2.4

No	Simbol	Deskripsi
1	<p>Aktor</p>  <p>Atau</p>  <p>Tanpa waktu aktif</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan dalam menggunakan kata benda diawal frase nama aktor.</p>
2	<p>Garis hidup/<i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek</p>
3	<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan</p>
4	<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang berhubungan dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p>  <p>Maka cekStatusLogin() dan open() dilakukan didalam metode login(). Aktor tidak memiliki waktu aktif</p>

**Tabel 2.4** *Sequence Diagram.*

5	<p>Pesan tipe <i>create</i></p> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6	<p>Pesan tipe <i>call</i></p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,</p>  <p>Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.</p>
7	<p>Pesan tipe <i>send</i></p> 	Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8	<p>Pesan tipe <i>return</i></p> 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
9	<p>Pesan tipe <i>destroy</i></p> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada <i>create</i> maka ada <i>destroy</i>

Tabel 2.5 Sequence Diagram (Lanjutan).

## 2.14 Pengujian *Black Box*

*Black box testing* adalah berurusan dengan perangkat lunak dengan kinerja internal yang tidak diketahui. Dengan cara ini, penguji dapat memperlakukan perangkat lunak sebagai "kotak hitam", yang tidak penting dalam kontennya, tetapi harus melalui proses pengujian internal dan eksternal (Nuryanto dan Wintoro. 2017). Uji kotak hitam, yaitu sejenis perangkat lunak uji yang dibuat menurut spesifikasi fungsional, tanpa desain uji dan kode program. Tujuan dari pengujian adalah untuk mengetahui apakah fungsionalitas, masukan dan keluaran dari perangkat lunak memenuhi spesifikasi yang dibutuhkan. Untuk melakukan pengujian kotak hitam, kasus pengujian diperlukan untuk mencoba menggunakan semua fungsi perangkat lunak, terlepas dari apakah itu memenuhi spesifikasi yang diperlukan. Kasus uji yang dibuat dalam pengujian kotak hitam memerlukan kasus penggunaan nyata dan kasus penggunaan yang salah. Berikut adalah 10 tipe pengujian dari metode

Berikut adalah beberapa tipe pengujian dari metode Black Box menurut (Julian 2015) yang digunakan pada penelitian ini:

### 1. *Equivalence Partitioning* :

Membagi inputan menjadi kelas data yang dapat digunakan untuk menggenerasi kasus uji.

### 2. *Comparison Testing* :

Uji setiap versi dengan data yang sama untuk memastikan semua versi menghasilkan keluaran yang sama.

## 2.15 Penelitian Terkait

Pada tahun-2013, Shelve Nidya Neyman, Muhammad Fikri Isnaini dan Sri Nurdianti menghasilkan penelitian yang berjudul Penerapan Sistem E-Voting pada Pemilihan Kepala Daerah di Indonesia (*The Application of E-Voting Systems in the Local Elections in Indonesia*). Penelitian ini berhasil menerapkan sistem E-Voting yang terdiri dari system pendaftaran, validasi dan pengaktifan pemilih, login/masuk ke sistem, sistem pemilihan, dan sistem perhitungan suara. Penerapan tersebut dalam bentuk simulasi komputer yang dibangun menggunakan bahasa



pemrograman PHP dan dijalankan melalui web browser. Fitur Keamanan sistem yang diterapkan pada sistem E-Voting ini meliputi:

1. Penerapan Captcha
2. Penggunaan Session
3. Penerapan Algoritme Kunci Publik, Kunci Simetri, dan Fungsi Hash
4. Pengecekan Karakter pada Saat Login

Namun pada penelitian ini, sistem yang dibuat hanya dapat dijalankan melalui web browser dan masih menerapkan sistem autentikasi yang memungkinkan pemilih melakukan manipulasi dalam pelaksanaannya karena tidak memiliki sistem autentikasi yang memadai seperti *face recognition*.

Sedangkan Raditya Prananda, Hengky Anra, dan Helen Sasty Pratiwi pada tahun 2017 menghasilkan penelitian yang berjudul Rancang Bangun Aplikasi E-Voting Berbasis Android (Studi Kasus : Pemilihan Ketua Organisasi di Lingkungan Fakultas Teknik Universitas Tanjungpura). Penelitian ini menghasilkan aplikasi E-Voting berbasis android untuk memfasilitasi partisipasi dan pelaksanaan kegiatan voting. Pengguna dapat berpartisipasi dalam kegiatan voting hanya dengan menggunakan smartphone Android yang terkoneksi dengan internet. Berdasarkan hasil implementasi, aplikasi ini sudah berjalan dengan baik pada perangkat mobile Android, mulai dari Android versi 4.3 (Jelly Bean) hingga Android versi 6.0.1 (Marshmallow). Namun aplikasi ini masih belum menggunakan sistem autentikasi sehingga berpotensi untuk dimanipulasi oleh pemilih.

Selain itu pada tahun 2018 Rizki Amelia, Saiful Bahri, Wisnu Sanjaya menghasilkan penelitian yang berjudul PERANCANGAN APLIKASI E-VOTE BERBASIS MOBILE ANDROID PADA PEMILIHAN KETUA RT NGESTIHARJO RT 02/15 SISWODIPURAN BOYOLALI. Penelitian ini menghasilkan aplikasi RT-Vote yang dapat digunakan untuk membantu proses pemilihan Ketua RT di Kampung Ngestiharjo, RT 02/15, Siswodipuran, Boyolali. Aplikasi RT-Vote masih mengalami kendala dalam menghubungkan database dengan aplikasi RT-Vote, sehingga sistem RT-Vote masih perlu ditingkatkan.

Selain itu, aplikasi tersebut masih menggunakan kode verifikasi barcode dengan sistem pengetikan, dan sistem pemindaian tidak dapat digunakan dengan kamera.

Sedangkan pada tahun 2018, Oka Nurul Huda, Fajri Profesio, dan Putra Jaroji menghasilkan penelitian yang berjudul **APLIKASI E-VOTING BERBASIS MOBILE MENGGUNAKAN OPTICAL CHARACTER RECOGNATION**. Penelitian ini menghasilkan aplikasi e-voting menggunakan teknik *Optical Character Recognition* (OCR). Selama proses pemungutan suara untuk masuk ke aplikasi, gunakan OCR untuk mengidentifikasi nomor identifikasi siswa yang terdaftar di KTM. Nomor ID akan dicocokkan dengan database siswa untuk menentukan apakah mereka memiliki hak suara dan kemudian memberikan hak pilihnya. Aplikasi pemungutan suara elektronik dibangun di atas platform seluler dan memiliki informasi yang cepat. Namun, aplikasi ini masih dapat dimanipulasi karena tidak dapat menjamin keaslian pemilih selama proses pelaksanaannya.

Pada tahun-2019, Rosyidah Siregar, Yessi Fitri Annisah Lubis, dan Nenna Irsa Syahputri menghasilkan penelitian yang berjudul **Aplikasi Absensi Berbasis Android Menggunakan Differensi Fitur Biometrika Wajah**. Penelitian ini menggunakan fitur biometrika wajah untuk memvalidasi dan mengautentikasi identitas pemilik wajah menggunakan nilai differensi fitur sederhana sehingga meminimalisir kompleksitas proses autentikasi selain tetap menjaga akurasi dari hasil autentikasi. Dengan nilai ambang selisih sebesar 30–50% sistem yang dibangun dapat mendeteksi pemilik wajah dengan akurat dengan pencahayaan dan posisi yang berbeda. kemiripan yang cukup tinggi. Tetapi pada aplikasi ini, wajah dengan pemilik yang sama nilai selisih dianggap masih terlalu besar yaitu selisih tertinggi adalah 48.03 sehingga memungkinkan kesalahan deteksi pada individu lainnya dengan wajah yang mirip.