

Penerapan Algoritma *Divide and Conquer* pada Perancangan Sistem Identitas Penduduk Berbasis *Fingerprint*

David Sinandar¹, Pono Budi Mardjoko¹ dan Harlianto Tanudjaja²

ABSTRACT: Identity of citizens and residents in Indonesia is currently using cards for the physical form of identity document, i.e. identification card. A physical form is vulnerable to loss and damage, creating problems for people and government. One of the solutions to these problems is by using fingerprint-based identification system, using a divide and conquer algorithm. The main program is set to be ready to detect fingerprints. Once a fingerprint is scanned using a fingerprint scanner, the pattern read by the fingerprint sensor is changed to text form that can be stored in the database. The text pattern used by the fingerprint scanner then can be compared to fingerprint text patterns stored in the database. If a match is not found, then the program will ask the person to fill his/her full identity and later when the same fingerprint pattern is found, then all identity data is displayed. Identities already registered can be changed if there is a change of identity, and can be molded into a physical form, if necessary. This system is designed using Visual Basic 6.0 software as the main program and using MySQL Workbench 6.0 as the database builder. The system testing involved a total of 240 samples of fingerprints, and from the test results obtained the search time is lower when compared with the sequential method.

KEYWORDS: identity, identification card, fingerprint, divide and conquer

ABSTRAK: Identitas penduduk di Indonesia saat ini menggunakan bentuk fisik kartu yaitu Kartu Tanda Penduduk (KTP). Bentuk fisik ini rentan terhadap kehilangan dan kerusakan yang dapat menimbulkan masalah bagi masyarakat dan pemerintah. Salah satu solusi masalah ini adalah dengan menggunakan sistem identifikasi berbasis sidik jari, menggunakan algoritma divide and conquer. Program utama diatur untuk selalu siap mendeteksi sidik jari. Setelah sidik jari dipindai menggunakan sensor sidik jari, sensor mengubah pola sidik jari tersebut kedalam bentuk teks sehingga dapat disimpan pada basis data. Pola teks tersebut digunakan oleh sensor untuk dibandingkan dengan pola teks sidik jari yang tersimpan dalam database. Apabila tidak ditemukan pola sidik jari yang sama, maka program akan meminta orang tersebut untuk mengisi identitas lengkap dan apabila ditemukan pola sidik jari yang sama, maka seluruh data identitas akan ditampilkan. Identitas yang sudah didaftarkan dapat diubah apabila ada perubahan identitas, dan dapat dicetak ke dalam bentuk fisik apabila diperlukan. Perancangan sistem ini menggunakan perangkat lunak Visual Basic 6.0 sebagai program utama dan menggunakan MySQL Workbench 6.0 sebagai pembangun basis data. Pengujian sistem ini melibatkan total 240 sampel sidik jari, dan dari hasil pengujian didapatkan waktu pencarian yang lebih rendah bila dibandingkan dengan metode sekuensial.

KATA KUNCI: identitas, kartu tanda penduduk (KTP), sidik jari, divide and conquer

PENDAHULUAN

Identitas adalah keadaan, sifat atau ciri-ciri khusus seseorang atau benda[1]. Berdasarkan definisi tersebut, maka identitas setiap individu pasti memiliki ciri khusus yang tidak dimiliki individu lainnya. Adanya identitas akan memberikan pemahaman bahwa setiap individu memiliki perbedaan dan ciri khas masing-masing. Identitas merupakan sebuah sarana yang digunakan oleh negara untuk mendata jumlah penduduk yang ada di negara tersebut. Indonesia menggunakan Kartu Tanda Penduduk (KTP) sebagai kartu identitas diri. Bentuk fisik KTP yang berupa kartu sejauh ini dinilai masih memiliki kekurangan. Salah satu kekurangannya adalah rentan terhadap kehilangan, dapat rusak, dan dapat hancur jika terkena musibah (banjir, kebakaran, dan sebagainya). Keamanan ini dinilai masih kurang, dan bila ditinjau lebih jauh, bentuk identitas ini dapat digantikan oleh teknologi yang memiliki tingkat keamanan lebih seperti *password*, retina mata, sidik jari, atau DNA manusia. Salah satu contohnya adalah sidik jari. Tidak ada orang yang memiliki kesamaan fisik-baik rupa wajah maupun bentuk tubuh-sama persis 100%. Tidak ada dua sidik jari yang sama di muka bumi ini[2]. Pada perancangan ini akan menggunakan media sidik jari sebagai pengganti identitas diri.

Berdasarkan beberapa peristiwa yang terjadi, seperti kebakaran yang telah disebutkan, dapat disimpulkan bahwa bentuk fisik dokumen identitas penduduk Indonesia saat ini masih memiliki resiko yang dapat membahayakan. Hilangnya identitas diri akan menimbulkan masalah bagi setiap penduduk Indonesia, dan juga pemerintah. Diperlukan suatu cara untuk mengurangi resiko akan hilangnya dokumen identitas setiap warga dan diperlukan suatu langkah yang dapat mengatasi permasalahan waktu dalam pencarian data. Oleh karena itu, diharapkan perancangan sistem ini dapat mewakili identitas diri dari setiap individu, memiliki tingkat keamanan identitas yang lebih tinggi, serta dapat mempersingkat waktu kerja dalam pencarian identitas diri yang diinginkan. Tujuan perancangan sistem ini adalah membangun suatu sistem identitas kependudukan yang menggunakan sidik jari sebagai media identitas utama dan memanfaatkan algoritma *divide and conquer*.

KAJIAN PUSTAKA

Sistem yang dirancang berfungsi untuk mengidentifikasi identitas dari penduduk menggunakan sidik jari sebagai media pengenalan, yang pada proses penyimpanan dan pencarian datanya menggunakan algoritma *divide and conquer*. Sistem ini menggunakan dua buah komputer yang digunakan sebagai tempat penyimpanan data (*database*) yang terhubung dengan menggunakan jaringan lokal nirkabel. Terdapat program utama pada salah satu komputer *database*. Input data dilakukan dari program utama dan data tersebut kemudian diolah. Proses

¹ Jurusan Teknik Elektro Fakultas Teknik Universitas Tarumanagara Jakarta

² Jurusan Teknik Elektro Fakultas Teknik Universitas Atmajaya Jakarta

pengolahan ini berkaitan dengan lokasi penyimpanan pada *database*. Setelah diolah, data kemudian disimpan pada salah satu *database*.

Pada sistem yang dirancang, *user* harus menempelkan sidik jari pada pemindai sidik jari sesuai dengan ketentuan. Setelah sidik jari terdeteksi oleh alat pemindai sidik jari, data dari sidik jari tersebut dikirimkan ke *database* dan dicocokkan. Apabila terdapat kesamaan sidik jari dengan salah satu data yang tersimpan, maka data ditampilkan. Namun, apabila data tidak ditemukan, maka ditampilkan pesan bahwa sidik jari belum terdaftar sehingga *user* wajib untuk mendaftarkan data diri. Jika data telah tersedia, maka setelah data ditampilkan, *user* dapat memilih untuk mencetaknya atau mengganti isi data bila diperlukan.

Pada proses pendaftaran data diri, *user* terlebih dahulu memasukkan *input* berupa sidik jari dari jempol kanan menggunakan alat *fingerscan* sebanyak empat kali. Setelah muncul pemberitahuan bahwa pendaftaran sidik jari berhasil, maka *user* diminta untuk memasukkan seluruh identitas diri sesuai dengan tampilan yang telah tersedia. Data yang perlu didaftarkan adalah Nomor Induk Kependudukan (NIK), nama, tempat/tanggal lahir, jenis kelamin, golongan darah, alamat (RT/RW, Kel/Desa, Kecamatan), agama, status perkawinan, pekerjaan, kewarganegaraan dan foto. Pengambilan foto dapat dilakukan dengan dua cara, yaitu dengan menggunakan kamera (*web-cam*) yang telah tersedia pada komputer, atau foto dapat diambil melalui media penyimpanan data eksternal. Setelah selesai memasukkan data, maka seluruh data diri tersimpan di dalam *database*.

Perancangan ini akan menggunakan *database* yang dibangun secara terpisah. Pemisahan penyimpanan data dilakukan berdasarkan nomor induk kependudukan ganjil/genap, sehingga komputer pertama memiliki *database* kependudukan yang memiliki nomor induk kependudukan ganjil (*database* ganjil), dan komputer lainnya memiliki *database* yang berisikan data penduduk dengan nomor induk kependudukannya genap (*database* genap).

Pada proses pencarian data, *input* data berupa sidik jari diolah, dan program utama mengirimkan data tersebut untuk dicocokkan dengan data yang terdapat pada kedua *database*. Apabila ditemukan data yang cocok pada salah satu *database*, maka ditampilkannya seluruh data yang tersimpan. Apabila tidak ditemukan data yang cocok pada kedua *database*, maka akan ditampilkan pesan bahwa sidik jari belum terdaftar, kemudian *user* diwajibkan untuk mendaftarkan diri. Algoritma *divide dan conquer* digunakan pada proses pemisahan ketika program melakukan penyimpanan data dan ketika dilakukan pencarian data secara iteratif.

Perancangan sistem ini dilengkapi dengan mesin cetak (*printer*) untuk mencetak dokumen identitas bila diperlukan, setelah sebelumnya *user* mendaftarkan diri terlebih dahulu kemudian mengakses data diri dengan sidik jari yang telah didaftarkan.

Perancangan sistem ini menggunakan jaringan lokal nirkabel, sehingga *user* dapat mengakses data dirinya melalui komputer lain selama pada komputer tersebut terdapat program utama dan terhubung dalam satu jaringan dengan *database*. Komputer ini dinamakan komputer *client*. Komputer *client* membutuhkan sebuah pendeteksi sidik jari untuk memberikan *input* kepada program utama. Komputer *client* dapat menggunakan fungsi pengambil gambar dan pencetak data, apabila fitur *web-cam* dan mesin pencetak tersedia.

Penggunaan jaringan lokal nirkabel, dimaksudkan agar ketika dalam penerapannya, sistem ini tidak memerlukan kabel dalam menghubungkan komputer satu dengan komputer lainnya dan memungkinkan penggunaan dalam satu wilayah secara *online*, dan mampu menekan biaya untuk pemasangan kabel jaringan.

Diagram Blok

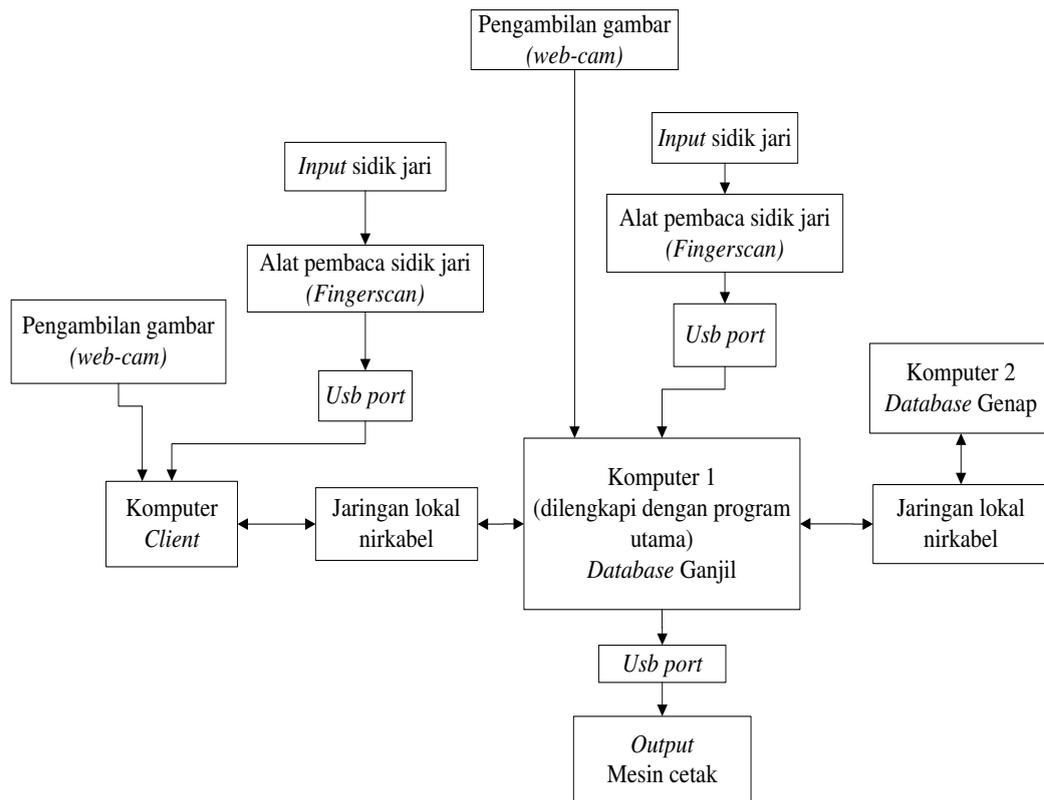
Perancangan sistem ini berfungsi untuk menjadikan sidik jari sebagai identitas diri seseorang. Oleh karena itu, diperlukan data berupa sidik jari dari orang tersebut. Sidik jari dimasukkan dengan menggunakan alat pemindai sidik jari yang terhubung dengan *USB port* pada komputer. Sedangkan data identitas diri dimasukkan dengan menggunakan komputer. Data yang akan disimpan kemudian diolah pada program utama, kemudian data tersebut akan didistribusikan pada salah satu *database* ganjil atau *database* genap sesuai dengan pengaturan yang dilakukan oleh program utama. Pada proses pengambilan gambar, menggunakan fitur *web-cam* yang tersedia pada komputer. Identitas diri dapat dicetak dengan menggunakan *printer*. Diagram blok dari sistem yang dirancang dapat dilihat pada Gambar 1.

Algoritma Divide and Conquer

Algoritma *divide and conquer*, memiliki 3 alur bagian yaitu alur *divide*, alur *conquer*, dan alur *combine*. Alur *divide*, membagi-bagi masalah menjadi beberapa sub-masalah yang memiliki kemiripan dengan masalah semula namun berukuran lebih kecil (idealnya berukuran hampir sama). Alur *conquer*, memecahkan (menyelesaikan) masing-masing sub-masalah (secara rekursif). Alur *combine*, menggabungkan solusi masing-masing sub-masalah sehingga membentuk solusi masalah semula[3].

Pembagian dilakukan pada objek masukan (*input*) yang berupa tabel (*larik/array*), matriks, eksponen, dan objek lain terantung pada masalah yang ada. Pada tiap sub-masalah harus memiliki karakteristik permasalahan yang serupa dengan karakteristik masalah awal, sehingga metode *divide and conquer* dapat berjalan dengan baik pada persoalan yang diselesaikan dengan pengulangan dengan memanggil dirinya sendiri (rekursif). Algoritma

ini juga dapat diterapkan pada proses pengulangan biasa (iteratif), karena pada prinsipnya proses iteratif hampir sama dengan proses rekursif.



■ Gambar 1. Diagram Blok Sistem yang Dirancang

Pemindai Sidik Jari (*Fingerprint Scanner*)

Alat pemindai sidik jari (*fingerprint scanner*) merupakan sebuah alat yang mampu membaca pola sidik jari. Diketahui bahwa pola sidik jari dari setiap individu berbeda. Terdapat suatu cabang ilmu yang mempelajari mengenai pola sidik jari, yang disebut *Dermatoglyphics*. *Dermatoglyphics* berasal dari bahasa Yunani *Derma* berarti kulit dan *Glyph* yang berarti ukiran. Ilmu ini mendasarkan pada teori epidermal atau garis-garis pada permukaan kulit. *Dermatoglyphics* mempunyai dasar ilmu pengetahuan yang kuat karena didukung oleh penelitian, ilmu ini meyakini bahwa sidik jari adalah “cetak biru” seseorang[4]. Para ahli tertarik dengan sidik jari karena;

1. Sidik jari tiap orang berbeda. Tidak ada orang yang memiliki sidik jari yang sama dan sidik jari tidak bisa dipalsukan.
2. Sidik jari bersifat permanen. Pola sidik jari tidak akan berubah dari sejak lahir hingga meninggal dunia.
3. Sidik jari mudah diklasifikasikan dan diukur. Mudah dilihat dengan mata telanjang dan bisa diintegrasikan dengan teknologi dan disimpan dalam *database*[4].

Pembacaan pola sidik jari dilakukan dengan menggunakan alat pemindai sidik jari. Menurut *Henry Classification System*, pola sidik jari dapat dibedakan menjadi 3 jenis, yaitu *loop pattern*, *whorl pattern*, dan *arch pattern*. Tampilan pola sidik jari dapat dilihat pada Gambar 2 [5].



■ Gambar 2. Bentuk Pola Sidik Jari [5]

Berdasarkan pola-pola inilah jenis sidik jari dapat dibedakan secara umum. Namun, ada alat pemindai sidik jari pola ini tidaklah cukup untuk membedakan setiap sidik jari yang ada. Terdapat satu metode yang digunakan oleh alat pemindai sidik jari saat melakukan pembacaan sidik jari. Alat pemindai sidik jari akan membentuk suatu pola unik yang disebut dengan *minutiae*. Pola inilah yang akan digunakan oleh alat pemindai sidik jari saat melakukan pencocokan sidik jari.

Teknik Pembacaan Sidik Jari

Terdapat 4 metode yang dapat digunakan untuk melakukan pembacaan sidik jari, diantaranya;

1. Metode Optis

Metode optis merupakan metode yang menggunakan cahaya untuk merekam pola sidik jari. Alat perekam sidik jari yang digunakan adalah kamera digital. Terdapat tempat untuk meletakkan sidik jari yang disebut *scan area*. Tepat di bawah *scan area*, terdapat pemancar cahaya yang menerangi permukaan *scan area*. Hasil pantulan cahaya tersebut ditangkap oleh alat penerima yang selanjutnya menyimpan gambar sidik jari tersebut ke dalam memori.

Kelemahan metode optis adalah hasil *scanning* sangat tergantung dari kualitas sidik jari. Jika kualitas sidik jari tidak sempurna, rusak atau luka, maka kualitas hasil pembacaan tidak maksimal. Kelemahan lain pada metode ini adalah dapat diretas dengan menggunakan jari palsu. Keunggulan dari metode ini adalah mudah dilakukan dan tidak membutuhkan biaya yang mahal.

2. Metode Ultrasonik

Merupakan metode yang menggunakan suara yang memiliki frekuensi sangat tinggi untuk menembus lapisan epidermal kulit. Suara tersebut dibuat dengan menggunakan *transducer piezoelectric*. Setelah suara tersebut akan memantulkan energi yang akan ditanggapi dengan alat sejenis. Pola pantulan yang dihasilkan digunakan untuk menyusun citra sidik jari yang dibaca. Keunggulan dari metode ini adalah, proses pembacaan sidik jari tidak dipengaruhi oleh tangan atau permukaan *scanner* yang kotor. Metode serupa juga digunakan pada dunia kedokteran.

3. Metode Kapasitansi

Metode kapasitansi, merupakan metode yang menggunakan kapasitansi dalam pembentukan citra sidik jari. *Scan area* berfungsi sebagai lempeng kapasitor, dan kulit ujung jari berfungsi sebagai lempeng kapasitor lainnya. Adanya *ridge* (gundukan) dan *valley* (lembah) pada sidik jari, maka kapasitas dari kapasitor masing-masing orang akan berbeda. Kelemahan pada metode kapasitansi adalah adanya listrik statis pada jari setelah dilakukan pengukuran. Tangan harus ditanahkan (*grounding*) untuk menghilangkan listrik statis ini.

4. Metode Thermal

Pada metode thermal, digunakan perbedaan suhu antar *ridge* dengan *valley* pada sidik jari untuk mengetahui pola sidik jari. Metode yang dilakukan adalah dengan menggosokkan ujung jari pada *scan area*. Bila ujung jari hanya diletakkan saja, dalam waktu singkat, suhunya akan sama karena adanya proses keseimbangan.

Modul Program Database dan Jaringan

Pada perancangan sistem ini akan dibangun dua buah *database* secara terpisah yang digunakan sebagai tempat penyimpanan data. *Database* pada sistem ini dibangun terpisah sebagai penerapan dari algoritma *divide and conquer*. Pemisahan *database* ini merupakan perwujudan dari algoritma *divide and conquer* yaitu langkah *divide*, dimana pada prinsip kerja dari langkah ini yaitu adanya pembagian masalah menjadi beberapa sub-masalah yang memiliki karakteristik yang mirip dengan masalah utama namun berukuran lebih kecil.

Masalah utama yang muncul adalah dengan menggunakan satu buah *database*, waktu pencarian data lebih memakan waktu. Pemisahan *database* dilakukan untuk memperkecil masalah utama tersebut, dimana sub-masalah (salah satu *database*) memiliki karakteristik yang sama dengan masalah utama (seluruh *database*) namun berukuran lebih kecil karena telah dibagi menjadi dua. Hasil pemisahan *database* ini dapat mempersingkat waktu pencarian data, karena bila data ditemukan pada *database* yang pertama, maka program tidak akan menelusuri *database* yang kedua.

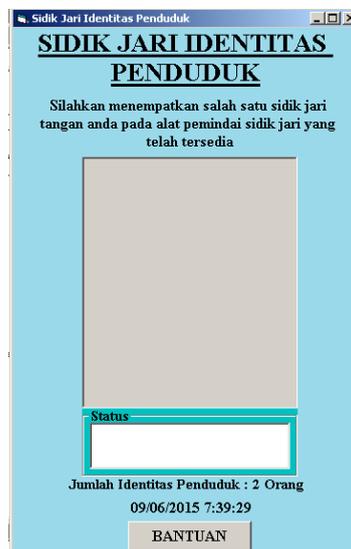
Pemisahan data dilakukan oleh program utama dengan mengidentifikasi nilai ganjil/genap dari nomor induk kependudukan yang telah disimpan oleh *user* ketika selesai mendaftarkan dirinya. Kemudian data disimpan pada *database* ganjil atau *database* genap sesuai dengan hasil dari pengolahan yang dilakukan oleh program utama. Kedua buah *database* ini akan dihubungkan dengan jaringan lokal nirkabel. Penghubungan jaringan lokal nirkabel menggunakan pengaturan pada alamat *Internet Protocol* (IP). Program utama membaca pengaturan alamat IP pada *text file* config.ini yang tersimpan pada direktori yang sama sesuai program utama. Hal ini dilakukan agar operator dapat menyesuaikan alamat IP dari jaringan yang digunakan tanpa harus mengakses ke dalam baris program utama. Pada *filetext* config.ini terdapat dua perintah yang masing-masing mengatur *server* ganjil dan *server* genap. Pengaturan yang tersimpan pada *file* config.ini antara lain, jenis *database* yang digunakan, alamat IP, *user id* dan *password* pada *database*, dan nama *database* yang digunakan. Program utama tidak boleh dijalankan apabila salah satu *database* mengalami gangguan, tidak aktif atau tidak dapat diakses. Tabel 1 berisi nama dan tipe data pada *database* yang digunakan.

Modul Program Sidik Jari dan Identitas

Program sidik jari dan identitas disusun dengan menggunakan perangkat lunak *Visual Basic 6.0*. Program utama diatur untuk selalu berada dalam kondisi siap untuk mencocokkan sidik jari, yang berada pada jendela utama. Tampilan jendela utama program dapat dilihat pada Gambar 3.

■ **Tabel 1.** Perancangan Tabel *Database*

Nama data	Tipe data
Id (<i>Primary Key</i>)	Bigint (20)
Nomor Induk Kependudukan	Bigint (40)
Nama	Varchar (255)
Tempat lahir	Varchar (255)
Tanggal lahir	Tanggal
Jenis kelamin	Varchar (64)
Golongan darah	Varchar (2)
Status perkawinan	Varchar (64)
Agama	Varchar (16)
Pekerjaan	Varchar (255)
Alamat	Varchar (1024)
RT	Varchar (8)
RW	Varchar (8)
Kelurahan/desa	Varchar (255)
Kecamatan	Varchar (255)
Kewarganegaraan	Varchar (4)
Foto	Longblob
Gambar sidik jari	Longblob
<i>Template</i> sidik jari	Varchar (4096)
Waktu	Timestamp



■ **Gambar 3.** Tampilan Jendela Utama Program

Pencocokan Sidik Jari

Apabila terdapat sidik jari yang terdeteksi oleh alat pemindai sidik jari, maka sidik jari tersebut akan dikonversi ke dalam bentuk *template* dengan format *text*. Kemudian seluruh *template* di dalam kedua *database* di-load oleh Flexcode SDK dan dicocokkan dengan *template* sidik jari yang terdeteksi.

Terdapat dua buah kondisi setelah sidik jari di-*scan*, yaitu sidik jari ditemukan dan sidik jari tidak ditemukan. Apabila ditemukan sidik jari yang cocok, maka Flexcode SDK akan mengembalikan nilai pada program utama bahwa ditemukan sidik jari yang cocok dengan data yang tersimpan. Nilai yang dikembalikan pada program utama adalah nilai dari nomor induk kependudukan. Nomor induk kependudukan ini kemudian digunakan oleh program utama sebagai acuan untuk mencari identitas pada kedua *database*. Pencarian pada kedua *database* merupakan perwujudan dari metode *conquer*, yaitu menyelesaikan seluruh sub-masalah. Proses

pencocokan dilakukan secara iterasi (pengulangan biasa tanpa memanggil dirinya sendiri). Apabila terdapat NIK yang cocok pada *database* pertama, maka pencarian akan dihentikan dan data hasil pencocokan akan dikembalikan untuk digabungkan pada program utama (proses *combine*). Apabila tidak ditemukan NIK yang sama pada *database* pertama, maka akan dilakukan pencocokan pada *database* kedua. Setelah identitas ditemukan, identitas tersebut digabungkan dengan program utama untuk ditampilkan (proses *combine*).

Setelah data ditemukan, program utama menampilkan pesan bahwa data ditemukan, kemudian memindahkan jendela utama menjadi jendela identitas diri serta menampilkan isi data diri. Data diri yang ditampilkan merupakan hasil dari penjemputan data dari hasil pencarian secara iterasi pada *database*. Tampilan jendela utama program sidik jari yang terdeteksi dapat dilihat pada Gambar 4.



■ Gambar 4. Tampilan Sidik Jari Ditemukan

Apabila ketika proses pencocokan sidik jari tidak ditemukan, Flexcode SDK akan mengembalikan nilai bahwa tidak ada sidik jari yang cocok, dan program utama akan menampilkan pesan bahwa sidik jari tersebut tidak terdaftar dan pemberitahuan bahwa pembacaan sidik jari dimulai. Program akan memindahkan layar utama menjadi layar pendaftaran identitas. Tampilan kondisi sidik jari tidak ditemukan dapat dilihat pada Gambar 5.



■ Gambar 5. Tampilan Sidik Jari Tidak Ditemukan

Pendaftaran Sidik Jari dan Identitas

Pada proses pendaftaran sidik jari dan identitas, setelah seluruh identitas diisi lengkap, maka akan dilakukan pendaftaran sidik jari. Pada proses pendaftaran sidik jari, *user* hanya dapat mendaftarkan satu buah sidik jari saja, dan dibutuhkan sampel dari sidik jari yang ingin didaftarkan sebanyak empat kali. Sidik jari tersebut akan dikonversi oleh Flexcode SDK kedalam bentuk *long text*.

Pada proses pengambilan sampel sidik jari sebanyak empat kali, sidik jari yang digunakan haruslah sidik jari yang sama. Setelah selesai memberikan sampel, program menampilkan pesan bahwa registrasi sidik jari sukses. Pendaftaran sidik jari dan identitas diakhiri dengan menekan tombol "simpan data", dan dinyatakan

sukses apabila muncul pesan bahwa simpan berhasil. Ini menandakan bahwa data telah tersimpan di dalam *database*. Sebelum disimpan di dalam *database*, program akan membandingkan nilai dari NIK, apabila nilai NIK ganjil, maka akan dilakukan penyimpanan data ke dalam *database* ganjil, dan apabila nilai NIK genap maka penyimpanan akan dilakukan pada *database* genap. Gambar 6. merupakan tampilan dari jendela identitas di mana pada jendela tersebut *user* dapat mengisi data dirinya secara lengkap.



■ Gambar 6. Tampilan Jendela Identitas

Modul Program Penghitung Waktu (*Timer*)

Modul program penghitung waktu digunakan untuk mengetahui berapa lama waktu yang diperlukan sistem dalam mendeteksi sidik jari, dan berapa lama waktu yang dibutuhkan sistem untuk menemukan identitas yang cocok. Modul program *timer* ini bekerja apabila *template* sidik jari yang di-*scan* terdeteksi atau data telah terdaftar di dalam *database*.

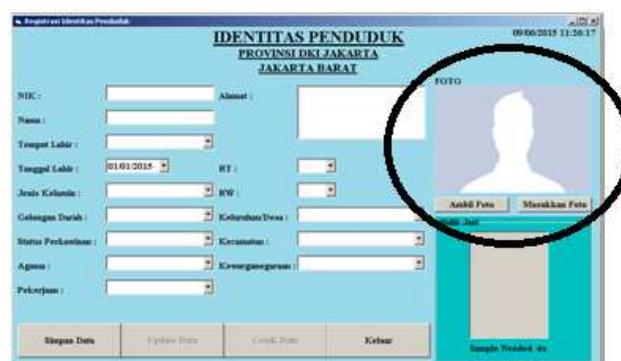
Pada modul program *timer* terdapat 3 buah waktu yang akan ditampilkan, yaitu waktu *load* adalah waktu yang dibutuhkan oleh alat Flexcode SDK untuk mendeteksi pola sidik jari, membuat *template* sidik jari, dan menampilkan gambar sidik jari. *Search time* yaitu waktu yang diperlukan program utama untuk mencocokkan NIK pada sidik jari yang dipindai dengan NIK yang terdaftar pada *database*. *Total time* yaitu waktu total yang diperlukan sistem untuk mengambil pola sidik jari sampai menemukan data yang diinginkan. Tampilan modul program timer dapat dilihat pada Gambar 7.



■ Gambar 7. Tampilan Modul Program *Timer*

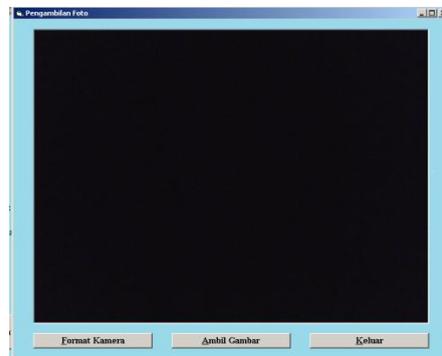
Modul Program Pengambil Gambar

Program pengambil gambar bekerja ketika *user* ingin mengambil atau memasukkan foto. Terdapat 2 tombol pilihan pada halaman registrasi identitas, yaitu tombol pilihan “ambil foto” dan “masukkan foto”. Pilihan tombol “ambil foto” digunakan untuk mengambil foto menggunakan *web-cam*, sedangkan pilihan tombol “masukkan foto” digunakan untuk memasukkan foto yang telah disimpan setelah diambil melalui *web-cam*, atau dapat memasukkan foto melalui media penyimpanan eksternal lainnya. Tampilan modul program pengambil gambar dapat dilihat pada Gambar 8.



■ Gambar 8. Tampilan Modul Program Pengambil Gambar

Apabila tombol “ambil foto” ditekan, maka muncul jendela baru yang berisikan tampilan yang ditangkap oleh *web-cam*. Ketika halaman ini ditampilkan, maka secara otomatis *web-cam* akan bekerja. Terdapat 3 buah tombol, yaitu tombol “format kamera”, tombol “ambil gambar” dan tombol “keluar”. Tombol “format kamera” digunakan untuk mengatur besar gambar yang ditangkap kamera, jenis pilihan warna yang diinginkan, dan menampilkan besar ukuran *file* gambar yang dihasilkan. Tombol “ambil gambar” digunakan untuk mengambil gambar sesuai dengan yang ditampilkan pada *web-cam*. Ketika *user* menekan tombol “ambil gambar”, maka tampilan yang ada pada layar akan langsung disimpan dan memunculkan jendela penyimpanan. Tombol “keluar” digunakan untuk kembali pada halaman registrasi. Tampilan jendela *web-cam* dapat dilihat pada Gambar 9.



■ Gambar 9. Tampilan Jendela Web-Cam

Modul Program Pencetak Data

Identitas yang telah tersimpan dapat dicetak dengan menggunakan pilihan cetak data. Identitas dapat dicetak apabila *user* telah menyimpan identitas diri terlebih dahulu. Setelah tombol pilihan “cetak data” dipilih, maka akan muncul jendela cetak lembaran identitas penduduk yang menampilkan data yang siap dicetak. Mesin cetak (*printer*) harus terhubung dengan sistem untuk dapat melakukan cetak data. Tampilan jendela cetak lembaran identitas penduduk dapat dilihat pada Gambar 10.



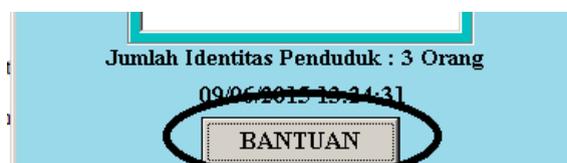
■ Gambar 10. Tampilan Jendela Cetak Lembaran Identitas Penduduk

Modul Program Pengubah Identitas

Apabila *user* ingin merubah data, maka *user* harus mengakses data dirinya terlebih dahulu dengan menempelkan sidik jarinya pada alat pemindai sidik jari. Setelah muncul pemberitahuan bahwa sidik jari dikenal dan data ditemukan, program kemudian menampilkan identitas *user*. Kemudian *user* dapat mengubah data yang ingin diganti. Seluruh jenis data dapat diubah, kecuali Nomor Induk Kependudukan dan pola sidik jari. Setelah selesai diubah, data harus disimpan dengan cara memilih tombol “Update data” yang tersedia.

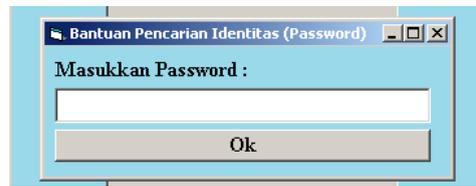
Modul Program Bantuan

Modul program bantuan berfungsi apabila dibutuhkan data identitas dari seseorang, namun orang tersebut tidak dapat hadir untuk mengakses data dirinya, atau apabila orang tersebut meninggal dunia. Modul program dapat diakses dengan menekan tombol “Bantuan” pada jendela utama. Tampilan tombol “Bantuan” pada jendela utama dapat dilihat pada Gambar 11.



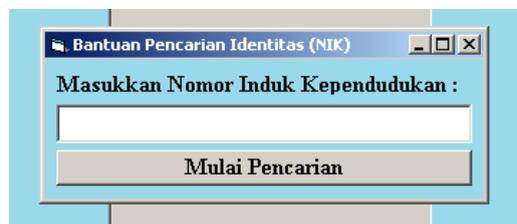
■ Gambar 11. Tampilan Tombol “Bantuan” Pada Jendela Utama Sistem

Modul program bantuan ini hanya dapat digunakan oleh operator. Oleh karena itu, modul program ini dilengkapi dengan fungsi pengamanan *password* yang hanya diketahui oleh operator tersebut. Setelah tombol “bantuan” dipilih, maka akan muncul jendela bantuan pencarian identitas (*password*). Tampilan jendela bantuan pencarian identitas (*password*) dapat dilihat pada Gambar 12.



■ Gambar 12. Tampilan Jendela Bantuan Pencarian Identitas (*Password*)

Setelah *password* dimasukkan, maka akan muncul jendela info yang memberitahukan bahwa *password* benar atau salah. Program bantuan bisa dilanjutkan apabila *password* yang dimasukan operator benar. Kemudian muncul jendela bantuan pencarian identitas (NIK). Operator wajib meminta nomor induk kependudukan dari data orang yang ingin diakses. Tampilan jendela nomor induk kependudukan dapat dilihat pada Gambar 13.



■ Gambar 13. Tampilan Jendela Bantuan Pencarian Identitas (NIK)

Setelah nomor induk dimasukkan, maka tombol “mulai pencarian” dapat ditekan untuk mencari apakah terdapat nomor induk kependudukan tersebut di dalam *database*. Apabila nomor induk dari orang tersebut cocok, maka identitas dapat diakses dan dapat dicetak. Namun, apabila nomor induk kependudukan tidak ditemukan, maka tidak ada data yang ditampilkan.

HASIL PENGUJIAN DAN ANALISIS

Pengujian Operasi Koneksi Database

Pengujian operasi *database* dilakukan dengan menghubungkan kedua *database* pada satu jaringan yang sama, dengan mendaftarkan alamat IP pada *file config.ini* yang terdapat pada direktori program utama. Apabila alamat IP yang didaftarkan pada *file* tersebut salah, atau koneksi *database* mengalami gangguan, maka ketika program utama dijalankan muncul pesan yang menandakan koneksi *database* belum terhubung. Gangguan jaringan penyebab *error* koneksi dapat disebabkan oleh beberapa faktor, antara lain;

1. Penulisan alamat IP, nama *database*, ID, dan *password database* pada *filetext config.ini* salah.
2. Koneksi jaringan putus atau tidak stabil.
3. *Database* belum diaktifkan.

Pengujian Operasi Program Sidik Jari dan Identitas

Pengujian dilakukan dengan menjalankan program utama yang telah terhubung dengan *database*. Kemudian dilakukan pemindaian sidik jari pada alat pemindai sidik jari yang telah disediakan. Hasil pengujian menunjukkan bahwa apabila sidik jari dikenal, maka menampilkan pesan bahwa data ditemukan dan menampilkan waktu lama proses pencarian, nomor induk kependudukan, dan memindahkan jendela identitas yang menampilkan seluruh identitas diri. Apabila sidik jari tidak dikenal, hasil pengujian menunjukkan bahwa muncul pesan bahwa sidik jari tidak dikenal atau belum terdaftar, yang kemudian dilanjutkan dengan proses pendaftaran sidik jari dan identitas diri.

Pengujian berikutnya dilakukan dengan pendaftaran identitas. *User* diwajibkan untuk mengisi seluruh identitas yang diminta. Apabila ada bagian yang belum terisi, maka muncul pesan bahwa masih terdapat bagian yang belum terisi. Pada bagian nomor induk kependudukan, *user* wajib mengisi bagian tersebut dengan data numerik. Apabila terdapat huruf pada bagian tersebut, maka muncul pesan bahwa nomor induk kependudukan wajib diisi oleh angka, dan *user* tidak dapat melanjutkan untuk mengisi bagian berikutnya.

Setelah seluruh data diisi, kemudian dilanjutkan dengan pengujian penyimpanan data. Pengujian dilakukan dengan menekan tombol “simpan data”. Setelah tombol “simpan data” ditekan, muncul pesan bahwa simpan berhasil. Pengujian berikutnya dilakukan dengan melihat isi dari *database* apakah terdapat penambahan data sesuai dengan data yang disimpan atau tidak dengan cara menekan tombol “refresh data” pada *database*.

Pengujian Operasi Program Pengambil Gambar

Terdapat 2 buah pilihan pada pengujian program pengambil gambar, yaitu pengujian tombol “ambil foto” dan pengujian tombol “masukkan foto”. Pengujian tombol “ambil foto” dilakukan dengan menekan tombol tersebut. Hasil pengujian menunjukkan bahwa setelah tombol “ambil foto” ditekan menampilkan jendela pengambilan foto yang mengaktifkan *web-cam*, dan memiliki 3 tombol, yaitu tombol “format kamera”, tombol “ambil gambar”, dan tombol “keluar”.

Pengujian berikutnya dilakukan dengan menekan setiap tombol yang tersedia pada jendela pengambilan foto. Apabila tombol “format kamera” ditekan, maka muncul jendela video format menampilkan pilihan besar foto, jenis warna dan ukuran besar *file* yang akan disimpan. Pada jendela video format, pada bagian *Resolution*, terdapat beberapa pilihan resolusi yang tersedia antara lain 160 x 120 *pixel*, 176 x 144 *pixel*, 320 x 240 *pixel*, 352 x 288 *pixel*, 640 x 480 *pixel*, dan pada bagian *Pixel Depth (bits) and Compression* terdapat 3 buah pilihan yaitu RGB 24, I424, dan YUY2. Kombinasi dari resolusi dan jenis *pixel depth* yang dipilih menghasilkan ukuran *file* yang bervariasi. Ukuran *file* yang dihasilkan dapat dilihat pada Tabel 2.

■ **Tabel 2.** Ukuran *File* Foto yang Dihasilkan Sesuai Resolusi Gambar dan Kedalaman Pixel (*bytes*)

Resolusi (<i>pixel</i>) Jenis <i>Pixel</i> <i>Depth</i>	160 x 120	176 x 144	320 x 240	352 x 288	640 x 480
RGB 24	57600	76032	230400	304128	921600
I420	28800	38016	115200	152064	460800
YUY2	38400	50688	153600	202752	460800

Pengujian berikutnya dilakukan dengan menekan tombol “ambil gambar” pada jendela pengambilan foto. Hasil pengujian menunjukkan bahwa setelah tombol “ambil gambar” ditekan muncul tampilan jendela yang digunakan untuk menyimpan gambar yang diambil. Terdapat dua pilihan, yaitu *Save* untuk menyimpan gambar dan *Cancel* untuk membatalkan penyimpanan gambar. Penyimpanan gambar dapat dilakukan dengan mencari direktori dan menuliskan nama *file* sesuai keinginan dan menekan tombol “*Save*”. Terdapat dua tipe format gambar yang dapat digunakan, yaitu format JPEG dan bitmap.

Pengujian berikutnya dilakukan dengan menekan tombol “keluar”. Hasil pengujian menampilkan ketika tombol “keluar” ditekan, program menutup fungsi *web-cam*, dan kembali menampilkan jendela registrasi identitas penduduk.

Pengujian berikutnya adalah pengujian tombol “masukkan foto”. Tombol ini digunakan untuk memasukkan foto yang telah diambil atau memasukkan foto menggunakan media penyimpanan eksternal lainnya. Setelah tombol “masukkan foto” ditekan, program akan menampilkan jendela *Open*, yang digunakan untuk memilih foto yang akan digunakan. Terdapat dua pilihan pada jendela *Open*, yaitu tombol “*Open*” untuk membuka foto yang sudah dipilih, dan tombol “*Cancel*” untuk membatalkan pemilihan foto. Setelah tombol “*Cancel*” ditekan, jendela *Open* menutup dan menampilkan pesan bahwa tidak ada foto yang dipilih.

Pengujian Operasi Program Pengubah Data

Pengujian operasi pengubah data dilakukan dengan mengakses data sidik jari terlebih dahulu. Kemudian dilakukan penggantian pada keseluruhan data, kecuali pada data sidik jari dan nomor induk kependudukan tidak dapat diubah. Kemudian data disimpan dengan menekan tombol “*Update data*”. Hasil pengujian menunjukkan muncul pesan bahwa *Update data* berhasil. Kemudian pengujian dilakukan dengan menutup jendela identitas penduduk dan kemudian mengakses identitas yang tadi telah diubah untuk melihat apakah perubahan yang dilakukan tersimpan pada *database*.

Pengujian Operasi Program Pencetak Data

Pengujian operasi program pencetak data dilakukan dengan mengakses data terlebih dahulu, dan menekan tombol “cetak data”. Hasil pengujian menunjukkan setelah tombol “cetak data” dipilih, muncul tampilan jendela baru yaitu jendela cetak lembar identitas penduduk. Pada jendela cetak lembar identitas penduduk ini terdapat pilihan untuk mencetak data, setelah menghubungkan mesin cetak (*printer*) terlebih dahulu.

Pengujian Operasi Program Bantuan

Pengujian operasi program bantuan dilakukan dengan menekan tombol “bantuan” yang terdapat pada jendela utama program. Tombol ini digunakan untuk memunculkan identitas seseorang apabila orang tersebut mengalami kecelakaan pada sidik jari yang telah didaftarkan atau apabila orang tersebut meninggal dunia. Setelah tombol “bantuan” ditekan, hasil pengujian menunjukkan muncul tampilan jendela baru yang meminta *user* untuk memasukkan *password*. Apabila *password* yang dimasukkan benar, maka muncul tampilan jendela baru yang meminta *user* untuk memasukkan nomor induk dari identitas orang yang ingin diakses, dan apabila

password tidak benar maka muncul pesan bahwa *password* salah dan tidak dapat mengakses identitas yang dikehendaki kemudian jendela bantuan tertutup dan kembali ke jendela utama program.

Pengujian dilakukan dengan memasukkan nomor induk kependudukan sebanyak enam kali dengan nomor induk yang masing-masing memiliki perbedaan. Tiga buah nomor induk pertama yang akan diuji terdaftar, dan tiga buah nomor induk yang akan diuji belum terdaftar. Hasil pengujian menunjukkan, apabila nomor induk yang dimasukkan terdaftar, maka muncul pesan bahwa data ditemukan. Kemudian ditampilkan waktu pencarian, dan jendela identitas beserta seluruh isinya. Hasil pengujian operasi program bantuan dapat dilihat dalam Tabel 3.

Pengujian Pengaruh Jumlah Data Terhadap Waktu Pencarian

Pengujian dilakukan dengan mengukur waktu total pencarian terhadap banyaknya jumlah data yang tersimpan pada kedua *database*. Waktu yang digunakan untuk melakukan pengukuran pada pengujian ini adalah waktu cari/*search time* pada modul *timer*.

■ **Tabel 3.** Hasil Pengujian Berdasarkan Nomor Induk Kependudukan Pada Modul Program Bantuan

Nomor induk kependudukan	Hasil
15710421109400010	Data ditemukan, identitas ditampilkan
31730250105400035	Data ditemukan, identitas ditampilkan
31730211103900033	Data ditemukan, identitas ditampilkan
3603120907940004	Data tidak ditemukan, program bantuan tertutup
317302461292000510	Data tidak ditemukan, program bantuan tertutup
3171056907950003	Data tidak ditemukan, program bantuan tertutup

Pengambilan waktu ukur menggunakan 6 buah sidik jari berbeda, dimana 3 sidik jari pertama terdapat pada *database* ganjil, dan 3 sidik jari berikutnya terdapat pada *database* genap. Setiap sidik jari diambil sampel sebanyak 5 kali. Waktu yang didapat dari 5 kali pengambilan sampel ini kemudian dirata-rata. Hasil pengukuran waktu pencarian sidik jari pada *database* genap dapat dilihat pada Tabel 4, Tabel 5, dan Tabel 6 dan hasil pengukuran waktu pencarian sidik jari pada *database* ganjil dapat dilihat pada Tabel 7, Tabel 8, Tabel 9. Dari hasil pengukuran dapat dilihat bahwa apabila data yang tersimpan semakin banyak maka waktu pencarian data yang dibutuhkan akan semakin bertambah.

■ **Tabel 4.** Pengaruh Jumlah Data Terhadap Waktu Pencarian (Sidik Jari Pertama pada *Database* Genap)

Sidik jari yang tersimpan di dalam kedua <i>database</i> (buah)	Waktu pencarian ke- (detik)					Waktu rata-rata pencarian (detik)
	1	2	3	4	5	
40	0,37	0,39	0,28	0,39	0,39	0,36
80	0,42	0,36	0,25	0,45	0,38	0,37
120	0,44	0,35	0,42	0,46	0,34	0,41
160	0,51	0,43	0,51	0,46	0,74	0,53
200	1,16	0,97	0,51	0,35	0,33	0,66
240	1,37	1,27	0,76	1,81	1,13	1,27

■ **Tabel 5.** Pengaruh Jumlah Data Terhadap Waktu Pencarian (Sidik Jari Kedua pada *Database* Genap)

Sidik jari yang tersimpan di dalam kedua <i>database</i> (buah)	Waktu pencarian ke- (detik)					Waktu rata-rata pencarian (detik)
	1	2	3	4	5	
40	0,13	0,33	0,34	0,32	0,4	0,3
80	0,42	0,3	0,35	0,34	0,29	0,34
120	0,3	0,46	0,35	0,29	0,38	0,36
160	0,6	0,49	0,35	0,35	0,48	0,45
200	0,32	1,09	0,38	0,43	0,39	0,52
240	1,14	0,77	0,69	0,85	0,71	0,83

■ **Tabel 6.** Pengaruh Jumlah Data Terhadap Waktu Pencarian (Sidik Jari Ketiga pada *Database* Genap)

Sidik jari yang tersimpan di dalam kedua <i>database</i> (buah)	Waktu pencarian ke- (detik)					Waktu rata-rata pencarian (detik)
	1	2	3	4	5	
40	0,38	0,34	0,28	0,35	0,22	0,31
80	0,32	0,19	0,41	0,37	0,37	0,33
120	0,42	0,34	0,34	0,3	0,31	0,34
160	0,37	0,33	0,42	0,42	0,48	0,4
200	0,53	0,4	0,41	0,59	0,63	0,51
240	1,27	0,99	0,77	0,69	0,78	0,9

■ **Tabel 7.** Pengaruh Jumlah Data Terhadap Waktu Pencarian (Sidik Jari Pertama pada *Database Ganjil*)

Sidik jari yang tersimpan di dalam kedua <i>database</i> (buah)	Waktu pencarian ke- (detik)					Waktu rata-rata pencarian (detik)
	1	2	3	4	5	
40	0,37	0,28	0,27	0,36	0,16	0,29
80	0,3	0,13	0,44	0,31	0,25	0,29
120	0,315	0,56	0,29	0,32	0,31	0,36
160	0,47	0,6	0,42	0,34	0,33	0,43
200	0,52	0,9	0,56	0,46	0,4	0,58
240	1,34	0,43	0,5	0,67	0,94	0,78

■ **Tabel 8.** Pengaruh Jumlah Data Terhadap Waktu Pencarian (Sidik Jari Kedua pada *Database Ganjil*)

Sidik jari yang tersimpan di dalam kedua <i>database</i> (buah)	Waktu pencarian ke- (detik)					Waktu rata-rata pencarian (detik)
	1	2	3	4	5	
40	0,36	0,32	0,11	0,3	0,31	0,28
80	0,33	0,39	0,37	0,32	0,29	0,34
120	0,34	0,33	0,35	0,45	0,36	0,37
160	0,79	0,37	0,34	0,42	0,41	0,47
200	0,91	0,51	0,44	0,59	0,35	0,56
240	0,65	0,63	1,05	0,311	1,14	0,76

■ **Tabel 9.** Pengaruh Jumlah Data Terhadap Waktu Pencarian (Sidik Jari Ketiga pada *Database Ganjil*)

Sidik jari yang tersimpan di dalam kedua <i>database</i> (buah)	Waktu pencarian ke- (detik)					Waktu rata-rata pencarian (detik)
	1	2	3	4	5	
40	0,33	0,22	0,32	0,32	0,3	0,3
80	0,12	0,39	0,39	0,31	0,35	0,31
120	0,31	0,25	0,37	0,36	0,33	0,32
160	0,39	0,34	0,38	0,43	0,3	0,37
200	0,45	0,57	0,38	0,62	0,56	0,52
240	0,71	1,40	0,61	1,35	0,7	0,95

Pengujian Pengaruh Teknik Pencarian Data Terhadap Waktu Pencarian

Pengujian dilakukan untuk membandingkan apakah terdapat perbedaan waktu pencarian data ketika pencarian dilakukan dengan menggunakan algoritma *divide and conquer* dan ketika pencarian dilakukan dengan menggunakan metode sekuensial. Pada pengujian yang dilakukan dengan menggunakan dua algoritma *divide and conquer*, didapatkan waktu pencarian rata-rata dari 3 sampel sidik jari yang berbeda. Pengukuran serupa dilakukan pada *database* yang menggunakan metode sekuensial dan pada *database* tersimpan jumlah data keseluruhan yang sama dengan pengujian sebelumnya. Hasil pengukuran dapat dilihat pada Tabel 10, Tabel 11, dan Tabel 12.

■ **Tabel 10.** Pengaruh Jumlah Data Terhadap Waktu Pencarian (Sidik Jari Pertama pada *Database Gabungan*)

Sidik jari yang tersimpan di dalam satu <i>database</i> (buah)	Waktu pencarian ke- (detik)					Waktu rata-rata pencarian (detik)
	1	2	3	4	5	
40	0,44	0,48	0,47	0,57	0,49	0,49
80	0,55	0,61	0,36	0,41	0,63	0,51
120	0,66	0,56	0,47	0,62	0,51	0,56
160	0,49	0,63	0,54	0,42	0,49	0,51
200	1,04	0,6	0,76	0,77	0,68	0,77
240	1,2	3,3	1,9	1,24	0,81	1,69

Hasil pengukuran pada program utama yang mengakses satu *database* kemudian dibandingkan dengan hasil pengukuran pada program utama yang menggunakan dua *database*. Hasil pengukuran dapat dilihat pada Tabel 13. Berdasarkan hasil pengukuran dapat dilihat bahwa adanya perbedaan waktu pencarian dimana dengan menggunakan algoritma *divide and conquer* memiliki waktu pencarian yang lebih cepat dari metode sekuensial.

Penggunaan algoritma *divide and conquer* memiliki waktu pencarian data yang lebih cepat. Namun, ditemukan beberapa faktor eksternal yang dapat mempengaruhi waktu pencarian data ketika pengujian dari sistem ini dilakukan. Faktor-faktor tersebut antara lain;

1. Koneksi jaringan yang tidak stabil.
2. Prosesor pada komputer *database* sedang mengerjakan hal lain, sehingga pengerjaan menjadi tidak optimal.
3. Komputer dan sensor sidik jari mengalami *overheating* sehingga memperlambat kerja prosesor dan mempengaruhi waktu pencarian.

4. Lokasi komputer pengakses dan komputer *database* berjauhan dengan *router* sehingga sinyal melemah dan menghambat laju aliran data.

■ **Tabel 11.** Pengaruh Jumlah Data Terhadap Waktu Pencarian (Sidik Jari Kedua pada *Database* Gabungan)

Sidik jari yang tersimpan di dalam satu <i>database</i> (buah)	Waktu pencarian ke- (detik)					Waktu rata-rata pencarian (detik)
	1	2	3	4	5	
40	0,47	0,36	0,52	0,44	0,57	0,47
80	0,45	0,52	0,44	0,56	0,49	0,49
120	0,51	0,57	0,6	0,49	0,63	0,56
160	0,66	0,54	0,43	0,49	0,51	0,53
200	0,74	0,63	0,54	0,81	0,52	0,65
240	1,75	1,83	1,68	1,02	0,92	1,44

■ **Tabel 12.** Pengaruh Jumlah Data Terhadap Waktu Pencarian (Sidik Jari Ketiga pada *Database* Gabungan)

Sidik jari yang tersimpan di dalam satu <i>database</i> (buah)	Waktu pencarian ke- (detik)					Waktu rata-rata pencarian (detik)
	1	2	3	4	5	
40	0,47	0,56	0,42	0,43	0,5	0,48
80	0,51	0,49	0,41	0,58	0,42	0,48
120	0,64	0,52	0,55	0,48	0,44	0,53
160	0,66	0,61	0,42	0,55	0,47	0,54
200	0,71	0,67	0,62	0,81	1,5	0,86
240	1,78	1,63	1,33	1,01	1,22	1,39

■ **Tabel 13.** Hasil Perbandingan Waktu Pencarian

Sidik jari yang tersimpan di dalam <i>database</i> (buah)	Waktu rata-rata metode pencarian komputasi paralel (detik)						Waktu rata-rata metode pencarian sekuensial (detik)		
	Data yang disimpan pada <i>database</i> genap			Data yang disimpan pada <i>database</i> ganjil			orang 1	orang 2	orang 3
	orang 1	orang 2	orang 3	orang 1	orang 2	orang 3			
40	0,36	0,3	0,31	0,29	0,28	0,3	0,49	0,47	0,48
80	0,37	0,34	0,33	0,29	0,34	0,31	0,51	0,49	0,48
120	0,41	0,36	0,34	0,36	0,37	0,32	0,56	0,56	0,53
160	0,53	0,45	0,4	0,43	0,47	0,37	0,51	0,53	0,54
200	0,66	0,52	0,51	0,58	0,56	0,52	0,77	0,65	0,86
240	1,27	0,83	0,9	0,78	0,76	0,95	1,69	1,44	1,39

KESIMPULAN

Kesimpulan yang dapat ditarik dari hasil pengujian dan analisis dari perancangan sistem kartu tanda penduduk berbasis *fingerprint* adalah, seluruh fitur yang terdapat pada program *Visual Basic* yang meliputi operasi *database*, sidik jari dan identitas, pengambil gambar, pencetak data, dan operasi bantuan dapat berjalan sesuai yang direncanakan. Berdasarkan hasil pengujian didapatkan bahwa waktu pencarian pada program yang menggunakan algoritma *divide and conquer* lebih singkat bila dibandingkan dengan waktu pencarian yang dilakukan secara sekuensial, namun terdapat beberapa faktor eksternal yang perlu diperhatikan. Sistem yang dibangun dapat menggunakan sidik jari sebagai media identitas utama dan sistem dapat bekerja dengan memanfaatkan algoritma *divide and conquer*.

DAFTAR PUSTAKA

- [1] S. Wojowasito, *Kamus Bahasa Indonesia*. Malang: C.V. Pengarang, 1999. pp. 125.
- [2] I. H. Misbach, *Dahsyatnya Sidik Jari*. Jakarta: Visimedia, 2010. Hal 11.
- [3] A. Pinandito, ST, M.MT, *Design and Analysis of Algorithm, Divide and Conquer Algorithm*, PTIIK Universitas Brawijaya.
- [4] BrainEvo. <http://www.sidikjariindonesia.com/>. 17 Juni 2015, Rabu pukul 10.25 WIB
- [5] S. C. Dass, *Classification of Fingerprints.pdf*, 31 Mei 2015, Minggu pukul 09.15 WIB