

Implementasi Sistem Monitoring Perangkat Klien pada Yamaha Smart Gateway SGX808

Helmi Nizar¹, Sabriansyah Rizqika Akbar, S.T, M.Eng.², Adharul Muttaqin, S.T, M.T.³
Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya, Malang
Jl. Veteran No 8, Malang 65145, Indonesia
e-mail: helminizar94@gmail.com¹, sabrian@ub.ac.id², adharul@ub.ac.id³

Abstrak - *Internet of Things (IoT)* secara umum merujuk kepada sebuah skenario agar kemampuan konektivitas dan komputing yang dimiliki sebuah jaringan komputer dapat menjangkau objek-objek, sensor-sensor, dan benda-benda yang digunakan sehari-hari yang tidak dikategorikan sebagai komputer, dan memungkinkan perangkat-perangkat ini untuk menghasilkan, menggunakan, serta saling bertukar data dengan sesedikit mungkin campur tangan dari manusia. Salah satu penerapan dari *IoT* yaitu pada bidang ekonomi, misalnya pada sebuah *convenience store* yang didalamnya terdapat berbagai macam perangkat, misalnya komputer, kamera, mesin kasir, *scanner*, *printer*, *smartphone*, dan lain sebagainya. Pada umumnya, setiap perangkat yang ada tersebut memiliki sistem mereka masing-masing dan tidak terhubung satu sama lain. Sehingga akan merepotkan apabila harus mengecek tiap perangkat untuk mengetahui apakah perangkat tersebut bekerja atau tidak. Untuk itulah perlu dibuat sebuah sistem monitoring seluruh perangkat yang ada pada tempat tersebut untuk memudahkan dalam pengawasan serta perawatannya. Sistem monitoring perangkat sendiri dapat dilakukan melalui dua jenis aplikasi, yaitu aplikasi *native* (terpasang pada sebuah perangkat) dan aplikasi berbasis *web*. Untuk aplikasi berbasis *web* memiliki kelebihan tersendiri, yaitu menghemat memori karena tidak ada aplikasi yang harus dipasang. Selain itu aplikasi berbasis *web* juga bersifat *open platform*, sehingga dapat dibuka oleh semua perangkat yang memiliki *web browser* dan terhubung ke *internet*. Maka dari itu dalam penelitian ini difokuskan ke aplikasi monitoring berbasis *web*.

Kata kunci: *IoT*, monitoring, *server*, aplikasi, *web*

1. PENDAHULUAN

Internet of Things (IoT) secara umum merujuk kepada sebuah skenario agar kemampuan konektivitas dan komputing yang dimiliki sebuah jaringan komputer dapat menjangkau objek-objek, sensor-sensor, dan benda-benda yang digunakan sehari-hari serta tidak dikategorikan sebagai komputer, dan memungkinkan perangkat-perangkat ini untuk menghasilkan, menggunakan, serta saling bertukar data dengan sesedikit mungkin campur tangan dari manusia.

Penerapan *IoT* sendiri saat ini sudah banyak, namun dalam riset sebagian besar masih terfokus ada pada bidang medis dan lingkungan. Selain itu, sebenarnya masih banyak sektor riset yang bisa dilakukan pemanfaatan *IoT*, salah satunya sektor ekonomi. Karena semakin berkembangnya ekonomi maka tuntutan kebutuhannya pun semakin banyak, termasuk kebutuhan teknologi penunjang termasuk *IoT*. Salah satu objek dalam bidang ekonomi yang dapat dilakukan penelitian *IoT* adalah pada *convenience store*.

Dalam sebuah *convenience store* yang di dalamnya terdapat berbagai macam perangkat, misalnya komputer, kamera, mesin kasir, *scanner*, *printer*, *smartphone*, dan lain sebagainya. Pada umumnya, setiap perangkat yang ada tersebut memiliki sistem mereka masing-masing dan tidak terhubung satu sama lain. Sehingga akan merepotkan apabila harus mengecek tiap perangkat untuk mengetahui apakah perangkat tersebut bekerja atau tidak. Untuk itulah perlu dibuat sebuah sistem monitoring seluruh perangkat yang ada pada tempat tersebut untuk memudahkan dalam pengawasan serta perawatannya.

Sistem monitoring perangkat sendiri dapat dilakukan melalui dua jenis aplikasi, yaitu aplikasi *native* (terpasang pada sebuah perangkat) dan aplikasi berbasis *web*. Untuk aplikasi berbasis *web* memiliki kelebihan tersendiri, yaitu menghemat memori karena tidak ada aplikasi yang harus dipasang. Selain itu aplikasi berbasis *web* juga bersifat *open platform*, sehingga dapat dibuka oleh semua perangkat yang memiliki *web browser* dan terhubung ke *internet*. Maka dari itu dalam penelitian ini difokuskan ke aplikasi monitoring berbasis *web*.

Mengetahui kebutuhan tersebut, *Yamaha Corporation* yang salah satu divisinya, *Yamaha Sound & Networking Division*, bergerak di bidang jaringan komputer berusaha memenuhi kebutuhan tersebut. *Yamaha Sound & Networking* mempunyai sebuah produk dengan nama *Yamaha Smart Gateway SGX808*, yang dapat digunakan sebagai solusi untuk permasalahan seperti yang sudah dijelaskan diatas. Hal ini karena selain bertugas sebagai *router*, perangkat ini juga dapat bertugas sebagai *server* yang dapat menyimpan data ke *database* maupun media penyimpanan lain yang terhubung ke perangkat ini, sehingga sangat cocok digunakan dalam kasus ini.

Untuk itulah, dengan bekerja sama dengan *Yamaha Sound & Networking Division*, dilakukan penelitian untuk membuat sebuah aplikasi monitoring yang akan dipasang ke dalam perangkat *Yamaha Smart Gateway SGX808*, sehingga akan bisa digunakan untuk menyelesaikan permasalahan seperti yang sudah dijelaskan diatas.

2. DASAR TEORI

2.1. Internet Protocol (IP)

Internet Protokol (IP) merupakan sebuah fungsi yang paling umum digunakan untuk saling bertukar data di internet, dengan bentuk blok data yang disebut “datagram”, dari sebuah host ke host yang dituju, dan memungkinkan untuk berkomunikasi pada jaringan yang berbeda. IP menjadi sangat penting karena apapun yang terhubung ke internet harus mendapatkan sebuah alamat IP. Saat ini, dengan semakin berkembangnya internet hampir semua perangkat dapat diberikan alamat IP dan terhubung ke internet.

2.2. Internet Control Message Protocol (ICMP)

Internet control message protokol (ICMP) merupakan sebuah metode yang digunakan oleh host-host dan router-router untuk saling berkomunikasi satu sama lain pada layer network dan bertukar informasi. ICMP biasanya digunakan untuk mengetahui kondisi sebuah koneksi yang dimiliki oleh sebuah host atau router ke host atau router yang lain.

2.3. Web Server

Web server bertugas untuk melayani fungsi permintaan yang menggunakan protokol HTTP. Dimana dalam penerapannya, HTTP merupakan komponen yang sangat diperlukan yang berada di sisi server pada arsitektur aplikasi berbasis web. Langkah pengolahan dasar yang dilakukan pada web server diawali ketika menerima permintaan dari protokol HTTP. Kemudian ditentukan apakah file yang diminta tergolong sumber daya file statis atau dinamis. Beberapa contoh file statis seperti video, gambar, atau file web HTML. Apabila file tergolong statis, maka server akan mencari dan mengambil file untuk kemudian membalas permintaan klien dengan mengirimkan balasan file yang diminta. Sedangkan apabila file yang diminta tergolong file dinamis, yaitu file yang berisi berbagai parameter dari permintaan, seperti waktu, atau informasi tambahan lainnya, maka akan ditangani oleh container pada server. Beberapa contoh dari HTTP web server adalah Apache, Apache Tomcat, Microsoft IIS, Nginx, dan Google GWS (Qiao, et al., 2014).

2.4. Database

Database atau basis data adalah koleksi dari data-data yang terorganisasi dengan cara sedemikian rupa sehingga mudah disimpan, dimanipulasi, diperbaharui, dicari, dan diolah dengan perhitungan-perhitungan tertentu, serta dihapus (Kusrini, 2007). Salah satu dari aplikasi database server yaitu MySQL.

2.5. Yamaha Smart Gateway SGX808

Yamaha Smart Gateway SGX808 adalah sebuah wireless LAN router yang berbasiskan sistem operasi Linux dan mendukung IEEE802.11b/g/n. Pada perangkat dilengkapi dengan lingkungan yang mendukung untuk menjalankan perangkat lunak dari pihak ketiga untuk dapat dipasang serta dijalankan pada produk ini.



Gambar 1 Yamaha Smart Gateway SGX808

Gambar 1 menunjukkan perangkat Yamaha Smart Gateway SGX808. Produk ini ditujukan untuk Small and Medium Business (SMB), contohnya seperti sebuah convenience store. Untuk memenuhi kebutuhan dari SMB maka produk ini memiliki beberapa kelebihan, misalnya platform jaringan khusus untuk SMB serta lingkungan sistem LAMP.

Pada Yamaha Smart Gateway SGX808 terpasang Apache, MySQL, dan PHP pada sistem operasi Linux dan memungkinkan perangkat ini untuk menjalankan aplikasi berbasis web. Aplikasi pihak ketiga dapat dipasang pada perangkat ini untuk menambah fitur yang tidak tersedia pada perangkat ini.

Pada Yamaha Smart Gateway SGX808 terdapat platform jaringan khusus untuk SMB. Pada perangkat ini dapat dipasang aplikasi pihak ketiga yang akan dapat membantu kinerja dari sebuah SMB. Aplikasi akan berjalan pada sistem operasi Linux dengan ukuran kecil sehingga akan bekerja secara efisien. Aplikasi dapat juga diakses oleh pengguna melalui Wifi, VPN, maupun dari Ethernet secara langsung.

Berikut adalah spesifikasi umum dari Yamaha Smart Gateway SGX808.

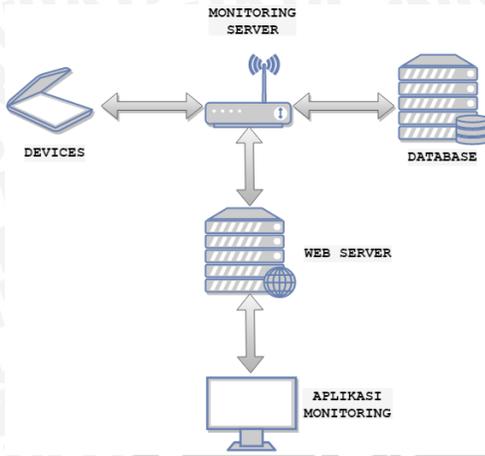
Tabel 1 Spesifikasi Yamaha Smart Gateway SGX808

| | |
|--------------|--|
| Flash ROM | 256 MB |
| DRAM | 256 MB |
| WAN port | 1 (10BASE-T/100BASE-TX MDI/MDIX-auto) |
| LAN port | 4 (10BASE-T/100BASE-TX MDI/MDIX-auto) |
| Wireless LAN | 2.4 GHz Standards: IEEE802.11b/g/n (Maximum clients 31) |
| USB port | 1 (USB storage, USB modem) |
| SD Card slot | 1 (Maximum 32GB) |
| Manajemen | Web Configuration, syslog, YMS-NMA |
| VPN | IPsec (2 destinations) |
| Aplikasi | Linux, Apache, MySQL, PHP |

3. PERANCANGAN DAN IMPLEMENTASI

3.1. Perancangan Sistem

3.1.1. Gambaran umum sistem



Gambar 2 Gambaran umum sistem

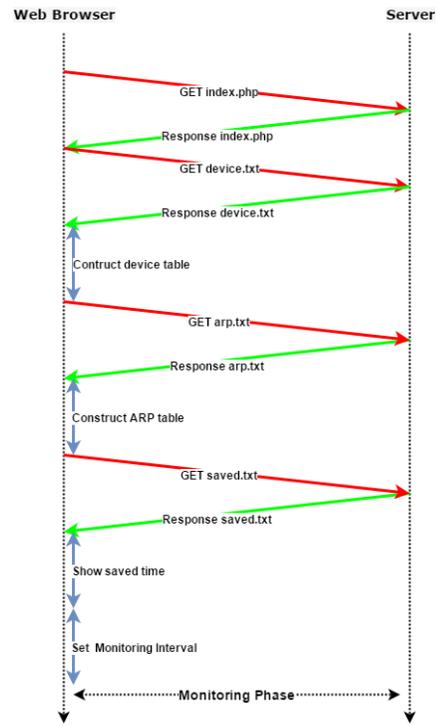
Gambar 2 merupakan gambaran umum dari sistem yang dibuat. Sistem akan terbentuk atas 5 bagian utama, berikut adalah penjelasan dari tiap bagian beserta hubungannya dengan bagian lain.

- *Devices*, merupakan perangkat-perangkat berbasis *IP* yang akan dimonitoring oleh *server*.
- Perangkat-perangkat tadi akan berkomunikasi dengan *server* menggunakan protokol *ICMP* untuk mengetahui status perangkat melalui koneksi dengan atau tanpa kabel.
- *Monitoring Server (Server)* yang digunakan untuk memonitoring perangkat dan menampilkan ke *web* adalah *Yamaha Smart Gateway SGX808*.
- *Server* dapat menyimpan data dari perangkat-perangkat yang dimonitoring ke *database*.
- *Server* mengakses *database* menggunakan *SQL Query* yang disertakan ke *PHP*.
- *Web server* akan menyimpan data yang dimiliki ke *server* berupa *plain text*.
- *Web server* akan berkomunikasi dengan *server* menggunakan *PHP*.
- *Web browser* bertugas menampilkan data dari perangkat yang dimonitoring *server* beserta statusnya. Selain itu *web browser* juga merupakan tempat bagi pengguna untuk melakukan masukkan data ke *server*.
- Pada *web browser* akan menggunakan *javascript* untuk fungsi monitoring, serta menggunakan *PHP* dan *Ajax* untuk berkomunikasi dengan *web server*.

3.1.2. Alur kerja sistem

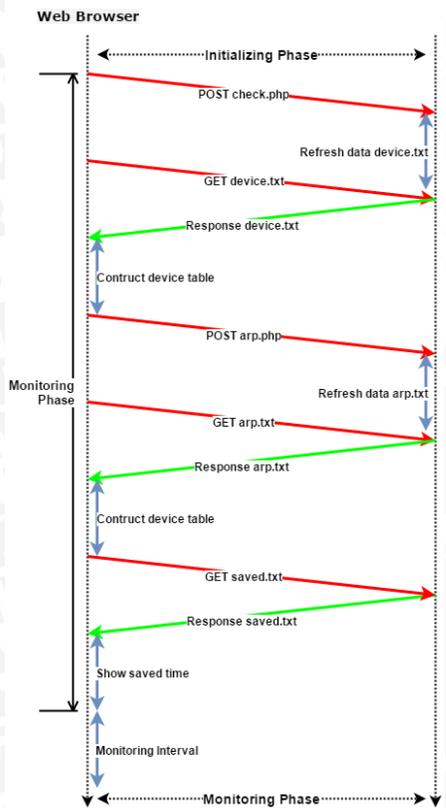
Pada alur kerja sistem ini akan dijelaskan alur kerja dari sistem monitoring perangkat pada saat tidak ada interaksi yang dilakukan oleh pengguna, serta komunikasi antara *web browser* dengan *server*. Proses dalam *web browser* akan menggunakan *javascript* dan *Ajax*, sedangkan komunikasi antara *web browser* dengan *server* menggunakan *PHP*. Alur kerja sistem ini dibagi

menjadi dua tahap, yaitu tahap inialisasi serta tahap monitoring.



Gambar 3 Alur kerja sistem pada tahap inialisasi

Pada gambar 3 merupakan alur kerja sistem saat tahap inialisasi. Tahap ini berlangsung ketika *web browser* pertama kali melakukan akses ke *server*. Dimulai dengan permintaan halaman *web* ke *server*, lalu melakukan permintaan *HTTP GET* untuk file *device.txt* yang berisi daftar perangkat yang terdaftar pada *server* dan selanjutnya membuat tampilan untuk tabel daftar perangkat pada *web browser* menggunakan *HTML*. Selanjutnya akan dilakukan hal yang sama untuk *arp.txt* dan tampilan tabel *ARP*, serta *saved.txt* dan tampilan untuk waktu penyimpanan daftar perangkat ke *database*. Pada akhir tahap inialisasi akan dilakukan *set interval monitoring* yang secara *default* bernilai 5 detik. Setelah tahap inialisasi selesai maka masuk ke tahap berikutnya yaitu tahap monitoring.

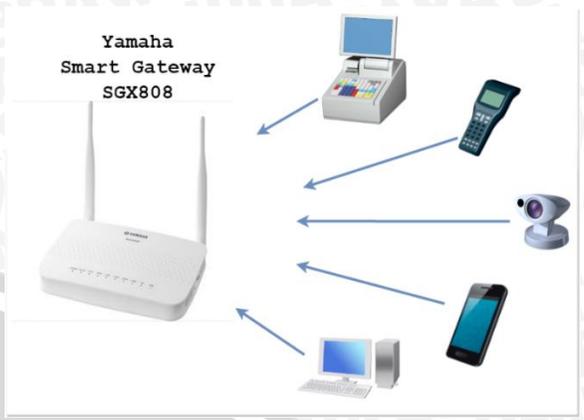


Gambar 4 Alur kerja sistem pada tahap monitoring

Pada gambar 4 merupakan alur kerja sistem pada tahap monitoring. Setelah masuk ke tahap monitoring maka *web browser* akan mengirimkan permintaan *POST check.php* ke *server*, lalu *server* akan menggunakan *check.php* untuk me-*refresh* data daftar perangkat yang dimiliki beserta status perangkat yang terdaftar. Lalu *web browser* lalu melakukan permintaan *HTTP GET* untuk file *device.txt* yang baru *server* dan selanjutnya me-*refresh* tampilan daftar perangkat pada *web browser*. Selanjutnya melakukan hal yang sama untuk tabel *ARP* dengan mengirimkan permintaan *POST arp.php*. Sedangkan untuk waktu penyimpanan *database* hanya dilakukan permintaan *GET saved.txt* karena nilai ini tidak berubah setiap saat, namun hanya saat ada *event* tertentu saja. Setelah semua proses selesai maka akan menunggu sampai *interval monitoring* terlewati lagi dan mengulangi tahapan monitoring dari awal lagi dan begitu seterusnya.

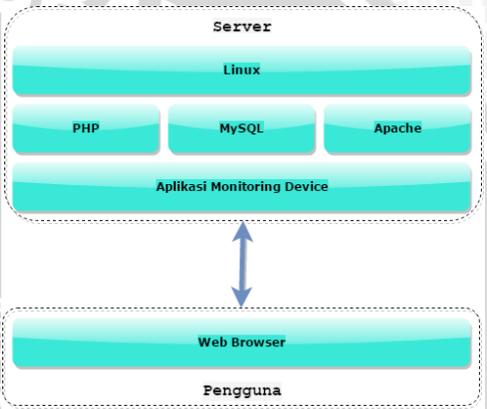
3.1.3. Perancangan Perangkat Keras

Pada gambar 5 merupakan ilustrasi perancangan perangkat keras yang digunakan dalam sistem ini. *Yamaha Smart Gateway SGX808* akan bertindak sebagai *server* monitoring. Sedangkan perangkat yang akan dimonitoring adalah semua perangkat berbasis yang terhubung ke *server* dan terdaftar ke daftar perangkat. Pada gambar diatas merupakan ilustrasi contoh perangkat yang terhubung ke *server*, karena permasalahan dari *convenience store* maka ilustrasi yang dilakukan merupakan perangkat-perangkat yang kemungkinan besar ada dalam sebuah *convenience store*.



Gambar 5 Perancangan perangkat keras

3.1.4. Perancangan Perangkat Lunak

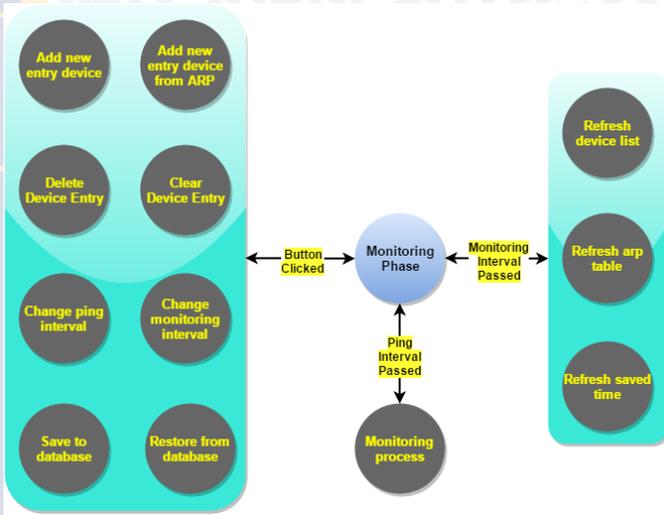


Gambar 6 Perancangan perangkat lunak

Gambar 6 merupakan ilustrasi perangkat lunak yang digunakan dalam sistem ini. Pengguna akan menggunakan *web browser* untuk mengakses aplikasi monitoring perangkat yang ada pada *server*. Sedangkan pada sisi *server*, aplikasi monitoring perangkat tersebut terhubung dengan aplikasi *PHP*, *MySQL*, serta *Apache* yang sudah terinstal pada *server*. Aplikasi-aplikasi tersebut berada pada sistem operasi *Linux* yang ada pada *server*.

3.1.5. Perancangan Aplikasi Berbasis Web

Gambar 7 merupakan gambaran umum aplikasi monitoring perangkat. Aplikasi ini secara garis besar dibagi menjadi tiga bagian, yang pertama yaitu pada gambar sebelah kanan ketika *interval monitoring* sudah terlewati. Untuk penjelasan dari bagian ini sudah dijelaskan sebelumnya pada alur kerja sistem.



Gambar 7 Gambaran umum aplikasi monitoring perangkat

Sedangkan bagian yang kedua yaitu pada gambar bagian bawah saat *interval ping* sudah terlewati. Proses monitoring yang dimaksudkan adalah penggunaan fungsi yang dimiliki oleh *Yamaha Smart Gateway SGX808* yaitu fungsi *Network Backup* yang telah dimodifikasi untuk penelitian ini. Fungsi ini bekerja dengan melakukan pengecekan koneksi dari perangkat yang diinginkan menggunakan *ping* lalu menyimpan hasilnya pada file *log*.

Lalu bagian dari aplikasi yang ketiga adalah saat ada interaksi dari pengguna aplikasi monitoring perangkat ini. Ada 8 jenis interaksi yang dapat dilakukan oleh pengguna.

- Tambah daftar perangkat baru
- Tambah daftar perangkat baru dari tabel *ARP*
- Hapus daftar perangkat
- Hapus semua daftar perangkat
- Mengubah *interval ping*
- Mengubah *interval monitoring*
- Simpan data daftar perangkat ke *database*
- Muat data daftar perangkat dari *database*

3.2. Implementasi Sistem

Implementasi sistem dapat dilakukan apabila proses perancangan dari sistem telah dipenuhi, karena implementasi sistem ini mengacu pada perancangan sistem yang telah dibuat sebelumnya. Pada bagian implementasi sistem memaparkan spesifikasi perangkat lunak dan perangkat keras yang digunakan pada sistem.

3.2.1. Implementasi Perangkat Keras

Dalam implementasi perangkat keras ini berdasarkan pada perancangan perangkat keras yang sudah dilakukan sebelumnya. *Server* yang digunakan adalah *Yamaha Smart Gateway SGX808*. Sedangkan perangkat yang akan terhubung ke *server* ada 2, yaitu laptop yang terhubung menggunakan *ethernet* serta *smartphone* yang terhubung melalui *wifi*.

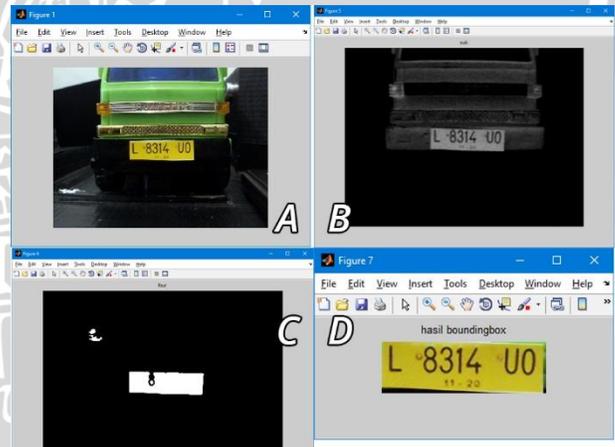


Gambar 8 Gambar implementasi perangkat keras

Gambar 8 merupakan implementasi perangkat keras yang digunakan dalam penelitian ini. Perangkat sebelah kiri adalah *server Yamaha Smart Gateway SGX808*, sedangkan perangkat yang tengah adalah *smartphone* yang akan dimonitoring serta pada sebelah kanan adalah laptop yang bertindak selain sebagai perangkat yang dimonitoring juga bertindak sebagai media yang digunakan untuk mengakses aplikasi monitoring perangkat.

3.2.2. Implementasi Pencarian Kandidat Plat Nomor

Proses pencarian kandidat plat nomor dilakukan untuk memisahkan area kandidat plat nomor dengan objek lainnya pada suatu citra. Hal ini dilakukan agar dalam pengenalan masing-masing karakter plat nomor bisa lebih fokus pada area tertentu.

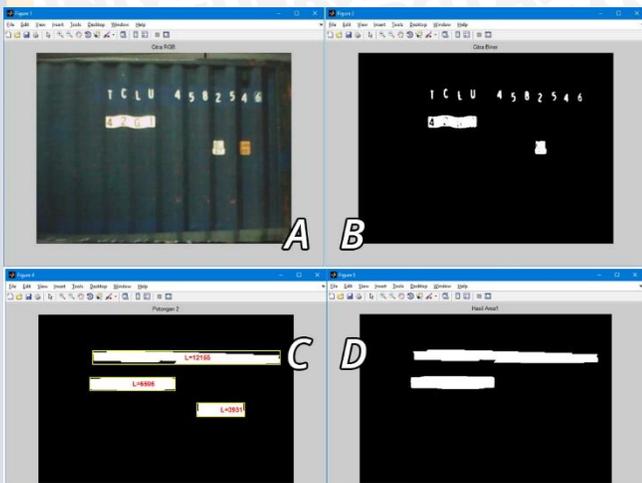


Gambar 9 Proses Pencarian Kandidat Plat Nomor

Keterangan:

- A) Citra asli / RGB
- B) Hasil Subtraksi pada Pendeteksian Warna Kuning
- C) Hasil *Thresholding* dengan ambang batas 90
- D) Hasil *cropping* objek berwarna kuning

3.2.3. Implementasi Pencarian Kandidat Nomor Kontainer

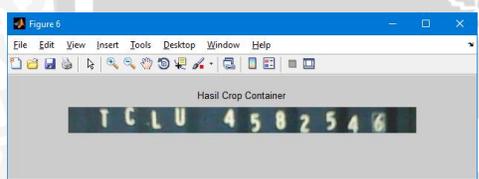


Gambar 10 Proses Pencarian Kandidat Nomor Kontainer

Keterangan:

- A) Citra asli / RGB
- B) Hasil *Thresholding* dengan ambang batas 190
- C) Hasil Segmentasi Objek Kandidat Nomor Kontainer
- D) Hasil Seleksi objek putih yang memiliki luas area antara 6000 hingga 13000 piksel.

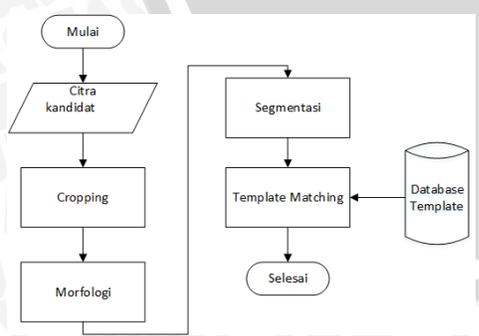
Setelah dilakukan proses seleksi objek, selanjutnya adalah memilih objek terbesar dari ConvexArea.



Gambar 11 Hasil Cropping Citra Nomor Kontainer

3.2.4. Implementasi Pengenalan Karakter

Setelah didapatkan citra kandidat plat dan nomor kontainer, langkah berikutnya adalah melakukan proses pengenalan masing-masing karakter dari plat dan nomor kontainer.



Gambar 12 Diagram Alir Proses pengenalan karakter

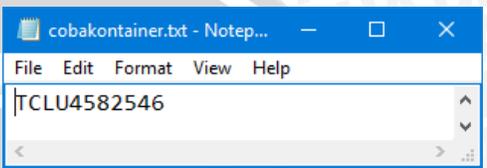
Pada gambar 3.8 awalnya citra kandidat akan dilakukan *cropping*, Lalu setelah itu dilakukan proses morfologi untuk memperbaiki bentuk objek agar lebih mudah dikenali. Langkah

selanjutnya adalah melakukan segmentasi tiap karakter untuk memisahkan karakter satu dengan lainnya. Kemudian tiap karakter terpisah akan masing-masing dilakukan proses *template matching* dengan seluruh citra template untuk mencari citra *template* yang memiliki kesamaan pola dengan citra hasil segmentasi.



Gambar 13 Citra Segmentasi

Pada gambar 3.9 merupakan citra kandidat yang dilakukan proses segmentasi karakter.



Gambar 14 Hasil Pengenalan Karakter

Pada gambar 3.10 merupakan hasil pengenalan karakter nomor kontainer yang tersimpan pada *file notepad*.

3.2.5. Implementasi Sinkronisasi Program Matlab dengan Delphi

Agar program pada aplikasi delphi dapat membaca program identifikasi citra yang telah dibuat dimatlab, terlebih dahulu harus melakukan kompilasi program matlab menjadi file yang bersifat Executable. Lalu file tersebut yang akan dieksekusi oleh aplikasi dekstop berbasis delphi.

| Name | Date modified | Type | Size |
|----------------------|------------------|---------------|----------|
| mccExcludedFiles.log | 14/03/2016 13:33 | Text Document | 864 KB |
| plat11.exe | 14/03/2016 13:33 | Application | 7.502 KB |
| readme.txt | 14/03/2016 13:33 | Text Document | 2 KB |

Gambar 15 Hasil Kompilasi Program Matlab

4. PENGUJIAN

4.1. Pengujian Akurasi Identifikasi Plat Nomor

Untuk mengukur persentase akurasi hasil pengolahan citra plat nomor pada setiap citra maupun keseluruhan menggunakan formula sebagai berikut:

$$Akurasi = \frac{\text{Jumlah Karakter yang dikenali dengan benar}}{\text{Jumlah total karakter}} \times 100\%$$

Tabel Error! No text of specified style in document..2 Hasil pengujian akurasi identifikasi plat nomor

| No | Citra Plat | Hasil Teks | Persentase |
|----|---|------------|------------|
| 1 |  | L8314UO | 100% |
| 2 |  | L8526UY | 100% |
| 3 |  | L9301UE | 100% |
| 4 |  | L9930UY | 100% |
| 5 |  | B9612UEK | 100% |
| 6 |  | S9371U6 | 87,5% |
| 7 |  | L8238UV | 100% |
| 8 |  | L9902UP | 100% |
| 9 |  | L8829UT | 100% |
| 10 |  | *P*G8** | 33,33% |
| 11 | Rata-rata | | 90,14% |

4.2. Pengujian Akurasi Identifikasi Nomor Kontainer

Tabel Error! No text of specified style in document..3 Hasil pengujian akurasi identifikasi citra nomor kontainer

| No | Citra Plat | Hasil Teks | Persentase |
|----|--|-------------|------------|
| 1 |  | CSLU1155484 | 100% |
| 2 |  | E1SU370?790 | 72,7% |
| 3 |  | E1TU134748? | 81,8% |
| 4 |  | EMCU3319663 | 100% |
| 5 |  | TCLU4582546 | 100% |
| 6 |  | TCLU4582551 | 100% |
| 7 |  | TCLU4582607 | 100% |
| 8 |  | TCLU4582612 | 100% |
| 9 |  | TCLU4582660 | 100% |

Berdasarkan tabel 4.1, didapatkan rata-rata persentase total tiap plat menghasilkan persentase 90,14% dari karakter yang diidentifikasi dengan benar dengan kondisi cahaya yang baik.

| | | | |
|----|---|-------------|-----|
| 10 |  | ?ML8276?076 | 36% |
| 11 | Rata-rata | | 89% |

Berdasarkan tabel 4.2, didapatkan rata-rata persentase total tiap kontainer menghasilkan persentase 89% dari karakter yang diidentifikasi dengan benar dengan kondisi cahaya yang baik.

4.3. Hasil Pengujian Waktu Total Sistem

Pada pengujian waktu total sistem, dilakukan pengukuran waktu total operasional setiap truk pembawa kontainer pada gerbang keluar depot. Pada purwarupa, diambil 6 kali pengujian yang hasilnya bersifat cocok, dari total 10 kali pengujian.

Tabel Error! No text of specified style in document..4 Tabel pengujian waktu pada purwarupa

| NO | Nomor Plat | Nomor kontainer | Identifikasi Citra | | | (D) | Total Waktu |
|-----------|------------|-----------------|--------------------|-------|-------|-------|-------------|
| | | | (A) | (B) | (C) | | |
| 1 | L8314UO | CSLU1155484 | 00:28 | 00:26 | 00:42 | 00:34 | 01:46 |
| 2 | L9930UY | EMCU3319663 | 00:36 | 00:32 | 00:42 | 00:53 | 02:05 |
| 3 | B9612UEK | TCLU4582546 | 00:29 | 00:28 | 00:42 | 00:35 | 01:47 |
| 4 | L8238UV | TCLU4582607 | 00:29 | 00:28 | 00:42 | 00:35 | 01:47 |
| 5 | L9902UP | TCLU4582612 | 00:32 | 00:31 | 00:42 | 00:28 | 01:40 |
| 6 | L8829UT | TCLU4582660 | 00:31 | 00:29 | 00:42 | 00:30 | 01:42 |
| Rata-rata | | | 00:31 | 00:29 | 00:42 | 00:36 | 01:48 |

Keterangan:

- (A) : Waktu Identifikasi Citra Plat Nomor
- (B) : Waktu Identifikasi Citra Nomor Kontainer
- (C) : Waktu total identifikasi Citra
- (D) : Waktu Pencetakan Report
- (Total Waktu) : (C)+(D)+30 detik (estimasi truk keluar depot)

Waktu identifikasi citra plat dan nomor kontainer berjalan secara paralel (bersamaan), sehingga disediakan ruang waktu selama 42 detik untuk menampung keduanya.

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil perancangan, implementasi, pengujian dan analisis yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut:

- 1) Proses identifikasi 10 sampel citra plat nomor menggunakan metode *template matching* dengan perbandingan bit 0 pada kondisi cahaya yang baik menghasilkan akurasi 90,14% karakter dikenali dengan benar. Sedangkan pada proses identifikasi 10 sampel citra nomor kontainer menggunakan metode *template matching* dengan perbandingan bit 0 menghasilkan akurasi 89% karakter dikenali dengan benar.
- 2) Berdasarkan pencocokan hasil identifikasi citra plat dan nomor kontainer dengan data plat dan nomor kontainer pada

database MySQL, dari 10 kali pengujian menghasilkan 6 kali pengujian yang menghasilkan kedua data (plat dan nomor kontainer) cocok.

- 3) Pada setiap pemrosesan truk-kontainer, rata-rata menghasilkan total waktu proses 1 menit 48 detik.

5.2. Saran

Dari kesimpulan penelitian ini, terdapat beberapa saran yang bisa menjadi acuan dalam pengembangan penelitian selanjutnya. Pada implementasi sistem khususnya pada proses identifikasi plat nomor, data plat nomor yang mengalami kesalahan pengenalan karakter disebabkan sistem tidak dapat membedakan dengan baik antara citra karakter dengan latar belakang atau warna karakter dengan latar belakang tidak kontras. Maka dari itu disarankan pada purwarupa sebaiknya menggunakan citra plat nomor yang memiliki warna karakter yang kontras terhadap latar belakang. Namun ini sulit diwujudkan pada kondisi riil dilapangan karena kondisi plat nomor truk yang berbeda-beda. Pada Implementasi sistem proses identifikasi nomor kontainer, data nomor kontainer yang mengalami kesalahan pengenalan karakter disebabkan terjadi kemiripan pola karakter yang diidentifikasi dengan karakter lain pada database. Hal ini dikarenakan terjadi kemiringan posisi kontainer yang berpengaruh pada posisi citra nomor kontainer yang ditangkap. Agar pengenalan karakter nomor kontainer mendapatkan hasil baik, maka sebaiknya membuat rekayasa lingkungan kerja yang membuat kontainer tetap lurus dan meminimalisir kemiringan posisi, dengan cara memberi pembatas tepi jalan pada jalan yang dilewati truk pembawa kontainer yang lebarnya mendekati lebar truk pembawa kontainer agar truk pembawa kontainer tetap lurus dan jika terjadi kemiringanpun tidak terlalu signifikan. Selain itu permasalahan lain yang muncul adalah ada karakter yang tidak dikenali dengan baik dikarenakan visualisasi karakter tidak jelas. Hal itu disebabkan kondisi cahaya yang gelap mengingat pada sistem ini masih menggunakan metode *thresholding* dengan nilai ambang yang statis, hanya berlaku pada kondisi cahaya tertentu. Maka dari itu, dalam pengembangan lebih lanjut tentang sistem ini, sebaiknya menggunakan metode *thresholding* yang memiliki nilai ambang batas yang bersifat dinamis, agar sistem dapat digunakan pada kondisi pencahayaan seperti apapun (fleksibel).

DAFTAR PUSTAKA

- [1] Arini, dkk, 2015. Pendeteksian posisi plat nomor mobil menggunakan metode morfologi dengan operasi dilasi, filling holes, dan opening. [e-journal]. Tersedia melalui: Jurnal Teknik Informatika Universitas Islam Negeri Syarif Hidayatullah Jakarta <<http://journal.uinjkt.ac.id/index.php/ti/article/download/1941/1511>> [diakses 2 Februari 2016]
- [2] Dephub, 2008. Peraturan Menteri Perhubungan : Penyelenggaraan dan Pengusahaan Depo Peti Kemas. Tersedia melalui: <http://jdih.dephub.go.id/assets/uudocs/permen/2008/km_no._47_tahun_2008.pdf> [Diakses 22 Maret 2016]
- [3] Hartanto S., Sugiharto A., Endah S. N., 2014. *Optical Character Recognition Menggunakan Algoritma Template Matching Correlation*, [e-journal] Tersedia di: <<http://ejournal.undip.ac.id/index.php/jmasif/article/view/8435/7163>> [Diakses 30 Februari 2016]
- [4] Hermawan, I., 2015. *Rancang Bangun Sistem Pendeteksi dan Pengenalan Angka Pada Meteran Air PDAM Menggunakan Metode Template Matching Correlation*, [e-journal] Tersedia di: <<http://journal.nurulfikri.ac.id/index.php/JTT/article/viewFile/39/28>> [Diakses 30 Februari 2016]
- [5] Kurnianto, Danny, 2015. *Belajar Image Processing – Fungsi Negatif Citra Untuk Transformasi Citra Grayscale Pada Domain Spasial* [online] Tersedia di:< <http://dannykurnianto.dosen.st3telkom.ac.id/belajar-image-processing-fungsi-negatif-citra-untuk-transformasi-citra-grayscale-pada-domain-spasial/>> [Diakses 26 Maret 2016]
- [6] Mujiono, dkk, 2012. *Implementasi pengenalan kode peti kemas secara otomatis menggunakan fitur eigen*. [e-journal]. Tersedia melalui: Digital Library Institut Teknologi Sepuluh Nopember <<http://digilib.its.ac.id/implementasi-pengenalan-kode-peti-kemas-secara-otomatis-menggunakan-fitur-eigen-19839.html>> [diakses 22 Maret 2016]
- [7] Permadi, Yuda, 2015. *Aplikasi pengolahan citra untuk identifikasi kematangan mentimun berdasarkan tekstur kulit buah menggunakan metode ekstraksi ciri statistik*. [e-journal]. Tersedia melalui: Jurnal Informatika Universitas Ahmad Dahlan <<http://journal.uad.ac.id/index.php/JIFO/article/download/2044/1308>> [diakses 22 Maret 2016]
- [8] POLRI, 2012. *Registrasi dan Identifikasi Kendaraan Bermotor*. Tersedia melalui: <<https://www.polri.go.id/pustaka/pdf/PERATURAN%20KAPOLRI%20NOMOR%205%20TAHUN%202012%20TENTANG%20REGISTRASI%20DAN%20IDENTIFIKASI%20KENDARAAN%20BERMOTOR.pdf>> [Diakses 22 Maret 2016].
- [9] Sianipar, 2013. *Pemrograman Matlab dalam Contoh dan Penerapan*, Bandung : Penerbit Informatika Bandung

