

# Pengembangan Sistem GUI Terintegrasi Basis Data SQL sebagai Implementasi Sistem *Monitoring* pada *Wire Selection Welding Machine*

Muhammad Fahar Rizki  
Prodi Teknologi Rekayasa Elektronika, Jurusan Teknik Elektro  
Politeknik Negeri Batam  
Batam, Indonesia  
fahar664@gmail.com

**Abstrak**—Penelitian ini membahas mengenai penerapan sistem GUI yang terintegrasi dengan basis data sebagai sistem monitoring pada *Wire Selection Welding Machine*. Penelitian ini didasarkan atas adanya permasalahan yang terjadi pada salah satu mitra perusahaan di Batam, yaitu adanya kesalahan prosedur dalam proses pengambilan kabel yang dapat menyebabkan *product reject* dan berakhir kerugian, baik waktu maupun material. Berdasarkan permasalahan tersebut, maka telah dirancang sebuah *user interface* menggunakan bahasa pemrograman C# dalam bentuk *windows form* yang dimuat dalam empat halaman yang ditujukan untuk mengatur dan memantau jalannya proses pengambilan kabel dan telah berfungsi dengan baik serta dirancang pula basis data yang dihubungkan terhadap *user interface* dengan berbasis *SQL Server* yang didalamnya memuat empat tabel. Meskipun demikian, basis data yang dirancang masih perlu dilakukan perbaikan pada data tertentu, seperti data tanggal dan waktu. Dengan adanya sistem tambahan ini, diharapkan pengguna (*user* atau *operator*) dapat melakukan proses pengambilan kabel secara optimal.

**Kata Kunci**—Basis Data, GUI, Monitoring, SQL Server

## I. PENDAHULUAN

Proses seleksi kabel tentu memerlukan ketelitian agar kabel yang dipilih sudah benar. Pada suatu mitra terkait, ditemukan adanya *error* selama proses nya seperti adanya kabel yang tidak terambil atau adanya satu jenis kabel yang terambil tetapi lebih dari satu. Hal ini dapat terjadi karena beberapa faktor, seperti adanya tekanan waktu yang dihadapi *operator* dan kurangnya indikasi pemantauan selama pengerjaan. Jika ini terus berlanjut, maka tidak sedikit kabel yang telah diseleksi mengalami cacat yang memicu produk menjadi *reject* dan berakhir kerugian waktu dan material.

Berdasarkan permasalahan tersebut, maka dapat dibuat sebuah sistem tambahan dimana dapat memonitori proses dalam seleksi kabel. Ada beberapa tawaran solusi atas permasalahan tersebut. Pada komponen *detector* dapat menggunakan sensor TCS3200 dimana ini adalah sensor yang dapat mendeteksi sebuah objek melalui pendekatan warna[1], [2], [3], [4]. Selain itu, pada komponen *detector* dapat juga menggunakan sensor ultrasonik jenis HC-SR04 dimana sensor ini merupakan sensor dengan akurasi tinggi dan pembacaan data yang lebih stabil [5], [6], [7], [8]. Terakhir, dapat menggunakan *sensor proximity* yang termasuk dalam kategori sensor jarak, namun dalam rentang deteksi yang lebih pendek[9].

Dari beberapa solusi yang ada, maka dapat diputuskan untuk menggunakan komponen *detector* berupa proximity sensor. Hal itu dikarenakan objek yang dideteksi hanya

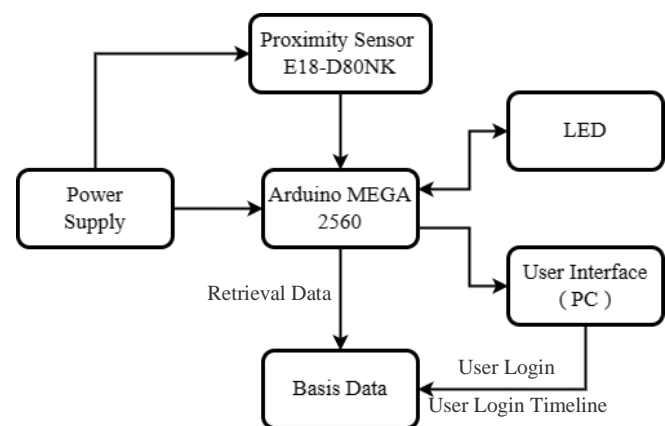
berjarak maksimal 5 cm. Atas dasar itulah, maka diambil sebuah penelitian dengan topik “Pengembangan Sistem GUI Terintegrasi Basis Data SQL sebagai Implementasi Sistem *Monitoring* pada *Wire Selection Welding Machine*”. Meskipun demikian, penelitian ini akan lebih difokuskan pada sistem GUI yang diintegrasikan ke basis data. Hal ini dikarenakan proses seleksi kabel yang diterapkan perlu adanya monitoring dan diharapkan dapat terhubung dengan basis data sehingga dapat diketahui riwayat histori dari proses seleksi kabel.

## II. METODOLOGI PENELITIAN

Penelitian ini akan melalui setidaknya empat tahapan utama, yaitu tahap perancangan sistem, tahap perancangan *user interface*, tahap perancangan basis data, dan tahap pengujian.

### A. Perancangan Sistem

Sistem yang dirancang menggunakan *proximity sensor* tipe E18-D80NK sebagai *detector* dan LED sebagai indikator yang nantinya akan ditampilkan pada *user interface* melalui PC. Tidak hanya itu, sistem yang dirancang akan terhubung ke basis data untuk melihat *track record* dari operator tersebut. Lebih jelasnya dapat dilihat pada Gambar 1.

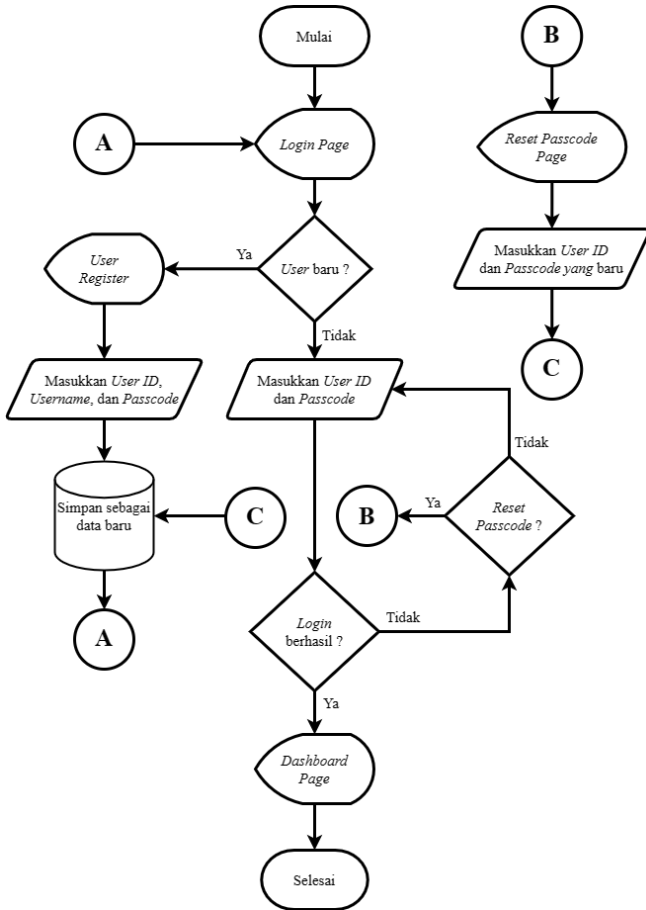


Gambar 1 Diagram Blok Sistem

Berdasarkan diagram di atas, basis data menerima data dari PC berupa status *operator login* atau *logout* yang akan disimpan pada tabel *User\_LoginTimeline*, sementara dari Arduino MEGA berupa serangkaian proses seleksi kabel yang akan disimpan dan diperbarui pada tabel *Retrieval\_Data*. Lebih jelasnya dapat dilihat pada Gambar 3.

### B. Perancangan User Interface

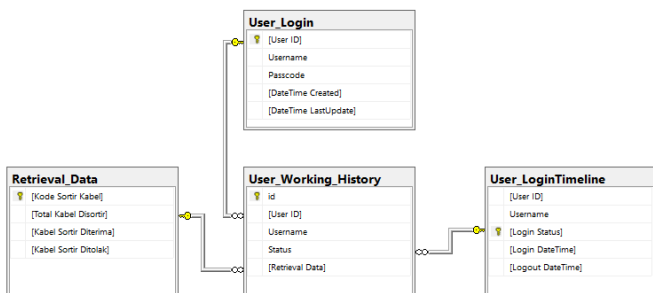
Pada tahapan ini *user interface* akan dirancang dengan menggunakan pemrograman bahasa C# melalui *software Microsoft Visual Studio 2022*. Dalam hal ini, *user interface* akan ditampilkan dalam bentuk *windows form* dan dimuat dalam 4 halaman dimana halaman tersebut adalah halaman *User Login, Dashboard, Register, dan Reset Passcode*.



Gambar 2 Diagram Alir User Interface

### C. Perancangan Basis Data

Basis data dirancang dengan menggunakan *software Microsoft SQL Server Management Studio 2019* dimana basis data tersebut memuat 3 tabel yang terdiri dari tabel *User Login, User Timeline, dan User Work History*. Pada tabel *User Login*, data yang akan disimpan berupa *User ID, Username, dan Passcode*; Pada tabel *User Timeline* berupa *User ID, Username, Login DateTime, Logout DateTime*; Pada tabel *User Work History* berupa *User ID, Username, Retrieve Accepted, Retrieve Rejected*.



Gambar 2 Skema Relasi Tabel pada Basis Data

### D. Pengujian

Pengujian dilakukan melalui skenario berdasarkan sudut pandang dari *operator*, mulai dari *operator* harus *login* terlebih dahulu sebelum mengoperasikan sistem sampai pada proses ketika *operator* telah mengambil semua jenis kabel sesuai dengan urutan kabel yang telah dipilih.

#### 1) Pengujian Waktu Pengoperasian User Interface

Pengujian ini ditujukan untuk mengetahui durasi *operator* dalam menentukan jenis kabel yang akan diambil. Pengujian ini perlu dilakukan agar proses pengambilan kabel masih tetap berjalan dengan optimal meskipun diterapkannya prosedur tambahan berupa *input* pada *user interface* sehingga tidak akan menambah waktu pengerjaan secara keseluruhan.

Pengujian ini dilakukan dengan menentukan target durasi *input* pada *user interface* selama 3 detik dikarenakan setelah proses pengambilan kabel masih ada proses tambahan berupa *welding* (penyatuan) keseluruhan kabel yang telah diambil. Durasi waktu dimulai ketika *operator* telah masuk ke halaman *dashboard* dan telah menetapkan *port* arduino yang sesuai.

#### 2) Pengujian Konektivitas Interface dengan Basis Data

Pengujian ini ditujukan untuk menentukan apakah *interface* terhubung dengan basis data dengan baik atau tidak. Selain itu ditujukan pula untuk menentukan *delay* pada basis data terhadap pembaruan data yang ditampilkan pada *interface* ketika pengoperasian telah berhasil.

Pengujian ini dilakukan pada dua halaman, yaitu halaman *reset passcode*, dan *register*. Pengujian ini dilakukan dengan membandingkan waktu secara *real-time* yang tertera pada *user interface* melalui *messagebox* dan waktu yang tertera pada basis data.

#### 3) Pengujian Operasional User Interface dan Basis Data

Pengujian ini ditujukan untuk menentukan tingkat keberhasilan baik pada *user interface* maupun basis data ketika beroperasi.

Pengujian operasional pada *user interface* dilakukan pada halaman *Dashboard* dengan beberapa jenis percobaan yang dicantumkan pada tabel 1. Sementara pengujian operasional pada basis data dilakukan pada 3 tabel, yakni tabel *User\_Login, Retrieval\_Data, dan User\_LoginTimeline* dengan beberapa jenis percobaan yang tercantum pada tabel 2.

Tabel 1 Pengujian Indikator Keberhasilan User Interface

Jenis Percobaan	Indikasi Keberhasilan	Status Keberhasilan
Aktivasi button "ON"	Indikator berwarna hijau sesuai dengan <i>input</i> data	
Aktivasi button "OFF"	Semua indikator berwarna kuning	

Indikator berwarna merah	Indikator berwarna merah seteah sebelumnya berwarna hijau	
Aktivasi Timer	Ditampilkan timer pada <i>Timer Left</i>	
<i>Remaining</i>	Perubahan nilai remaining pada semua indikator berwarna merah	
Aktivasi button "Logout"	User diarahkan ke halaman <i>login</i>	
<i>Connected and Disconnected Port</i>	Pesan yang ditampilkan pada <i>Connection Status</i> adalah <b>Connected</b> dan semua button yang ada pada <i>Input Box</i> dapat dioperasikan	
	Pesan yang ditampilkan pada <i>Connection Status</i> adalah <b>Disonnected</b> dan semua button yang ada pada <i>Input Box</i> tidak dapat dioperasikan	

Tabel 2 Pengujian Indikator Keberhasilan pada Basis Data

Tabel Basis Data	Jenis Percobaan	Indikasi Keberhasilan	Status Keberhasilan
<i>User Login</i>	Pembaruan data	Perubahan data pada kolom <i>DateTime LastUpdate</i> dan <i>Passcode</i>	
	Penambahan data baru	Ditambahkann ya data baru pada tabel <i>User_Login</i>	
<i>Retrieval Data</i>	Pembaruan data	Perubahan data pada kolom <b>Total Kabel Disortir, Kabel Sortir Diterima, dan Kabel Sortir Ditolak</b>	
<i>User LoginTimeline</i>	Pembaruan data status "Login/Logout"	Perubahan data pada kolom <i>Login Status</i>	
	Pembaruan waktu ketika status <i>Login/Logout</i>	Perubahan data pada kolom <i>Login DateTime</i> ketika <i>Login</i> dan <i>Logout DateTime</i> ketika <i>Logout</i>	

Untuk menentukan tingkat keberhasilan (%) dapat dirumuskan dengan persamaan berikut [10], [11].

$$\text{Tingkat Keberhasilan (\%)} = \frac{\text{Total Percobaan Berhasil}}{\text{Total Jenis Percobaan}} \times 100\% \quad \dots (1)$$

#### 4) Pengujian Performa Basis Data

Pengujian ini dilakukan untuk mengetahui beban kinerja dari tabel yang terdapat pada basis data yang digunakan ketika data dikirim dari *user interface*. Beban kinerja pada basis data akan mempengaruhi penggunaan memori yang digunakan dan akan berdampak pada waktu respon terhadap *user* serta penggunaan sumber daya yang berlebihan.

Pengujian ini dilakukan dengan mengoperasikan *user interface*, mulai dari proses *login*, *reset passcode*, *user*

*register*, hingga mengoperasikan sistem melalui halaman *dashboard* sebelum akhirnya *user* keluar dari halaman *dashboard* dan *logout* kemudian kembali ke halaman *login*. Pengujian beban kerja ini dibedakan atas pengujian beban kueri dan pengujian beban tabel pada basis data. Hasil pengujian beban kueri akan dicantumkan seperti pada tabel 3.

Tabel 3 Pengujian Beban Kerja Kueri pada *SQL Query Stress*

Percobaan	Total Eksekusi	Total Thread	Waktu per Eksekusi ( milidetik )
1			
2			
3			
4			
5			
6			
7			
8			

### III. HASIL DAN PEMBAHASAN

Penelitian ini memberikan hasil *output* berupa tampilan *user interface*, serta desain basis data dengan beberapa pengujian yang dilakukan dengan rincian sebagai berikut.

#### A. User Interface

*User Interface* dimuat dalam 4 halaman dimana halaman pertama kali ketika *operator* menjalankan sistem adalah halaman *User Login* seperti pada Gambar 4.

##### 1) Halaman User Login



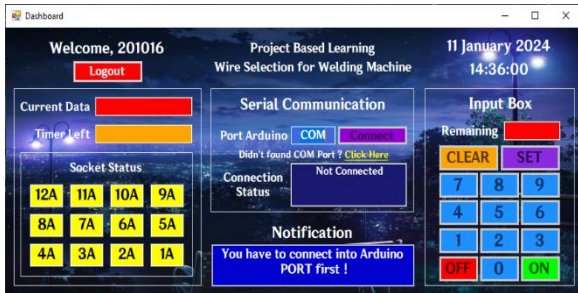
Gambar 3 Halaman *User Login*

Untuk pengoperasian halaman *User Login* dapat ditunjukkan dalam beberapa hal berikut..

- User* dapat memasukkan *User ID* dan *Passcode* sebelum mengoperasikan sistem. *User ID* disini berupa 6 digit angka dan *Passcode* dapat berupa kombinasi huruf dan angka dengan total maksimal 12 karakter.

- b) Membuat identitas sebagai *user* baru dengan menekan label **Register Here** berwarna kuning dan akan diarahkan ke halaman **User Register** seperti pada Gambar 6.
- c) Mengubah *passcode* dari *user* dengan menekan label **Reset Passcode** berwarna kuning dan akan diarahkan ke halaman **Reset Passcode** seperti pada Gambar 7.

2) Halaman Dashboard



Gambar 4 Halaman Dashboard

Dalam pengoperasian halaman *dashboard*, ada beberapa hal yang harus diperhatikan oleh *user* dengan rincian berikut.

- a) *User* harus menghubungkan *port* mikrokontroler terlebih dahulu sebelum beroperasi.
- b) Sebelum mengoperasikan sistem, *user* harus memastikan jenis data dan jumlah operasi yang dilakukan telah ditentukan.
- c) Tombol **CLEAR** digunakan untuk menghapus jenis data dan jumlah operasi.
- d) Tombol **SET** digunakan untuk memasukkan *input* jumlah operasi yang dilakukan. Jika terjadi kesalahan *input*, *user* hanya perlu menekan tombol **SET** kembali.
- e) Sebelum *logout* dari halaman *dashboard*, *user* harus memastikan bahwa sistem dalam kondisi **OFF** dan *port* mikrokontroler dalam kondisi *disconnected*.

3) Halaman Register



Gambar 5 Halaman Register

4) Halaman Reset Passcode



Gambar 6 Halaman Reset Passcode

B. Basis Data

Basis data dimuat dalam 4 tabel, dimana 3 tabel diantaranya merupakan tabel utama yang meliputi tabel *User\_Login*, *User\_LoginTimeline*, dan *Retrieval\_Data*, serta 1 tabel tambahan berupa tabel *User\_Working\_History*.

1) Tabel *User\_Login*

User ID	Username	Passcode	DateTime Created	DateTime LastUpdate
200006	Muhammad Fahar Rizki	carrier213	2024-01-14 16:56:59.257	2024-01-15 11:23:16.000
201006	Mhd Rizki	scrool447	2024-01-14 16:42:21.243	2024-01-15 10:59:50.723
201016	Fahar Rizki	carrier2330	2024-01-14 16:27:13.127	2024-02-16 18:16:09.473
210026	Muhammad Rizki	scharch889	2024-01-14 16:48:21.610	2024-01-15 11:01:57.523
211006	M Rizki	scharch886	2024-01-14 16:35:57.793	2024-01-15 11:17:33.980
211016	M Fahar	scarch88	2024-01-14 16:32:59.610	2024-01-15 11:04:54.203
211026	Muhammad Fahar	scharch880	2024-01-14 16:45:14.697	2024-01-15 11:18:42.947
211106	Mhd Fahar	scharch887	2024-01-14 16:40:14.603	2024-01-15 11:14:31.290
212016	Muh Fahar	scrool445	2024-01-14 16:53:16.933	2024-01-15 11:15:49.710
212206	Muh Rizki	scharch988	2024-01-14 16:55:28.903	2024-01-15 11:13:02.993

Gambar 7 Tabel *User\_Login* pada Basis Data

Pada tabel ini memuat beberapa data dengan rincian sebagai berikut.

- a) [*User ID*], merupakan kombinasi 6 digit angka sebagai identitas dari pengguna (*user* atau *operator*).
- b) *Username*, merupakan identitas nama dari pengguna yang dimuat dalam maksimal 20 karakter.

- c) *Passcode*, merupakan kombinasi dari total maksimal 12 karakter sebagai kode sandi identitas dari pengguna.
- d) [*DateTime Created*], merupakan data waktu yang menunjukkan kapan pertama kali data pengguna tersebut terdaftar.
- e) [*DateTime LastUpdate*], merupakan data waktu yang menunjukkan kapan terakhir kali pengguna melakukan perubahan data, seperti perubahan *passcode*.

### 2) Tabel Retrieval Data

Pada tabel ini memuat beberapa data dengan rincian sebagai berikut.

- a) [Kode Sortir Kabel], merupakan data 4 digit yang menunjukkan rentang dari kabel yang akan disortir.
- b) [Total Kabel Disortir], merupakan data yang menunjukkan total dari proses sortir yang dilakukan.
- c) [Kabel Sortir Diterima], merupakan data yang menunjukkan total proses sortir yang dilakukan berjalan dengan baik.
- d) [Kabel Sortir Ditolak], merupakan data yang menunjukkan total kesalahan yang terjadi selama proses sortir kabel.

Kode Sortir Kabel	Total Kabel Disortir	Kabel Sortir Diterima	Kabel Sortir Ditolak
0304	NULL	NULL	NULL
0305	9	9	1
0306	NULL	NULL	NULL
0307	NULL	NULL	NULL
0308	NULL	NULL	NULL
0309	NULL	NULL	NULL
0310	NULL	NULL	NULL
0311	NULL	NULL	NULL
0312	NULL	NULL	NULL

Gambar 8 Tabel *Retrieval\_Data* pada Basis Data

### 3) Tabel User\_LoginTimeline

Pada tabel ini memuat beberapa data dengan rincian sebagai berikut.

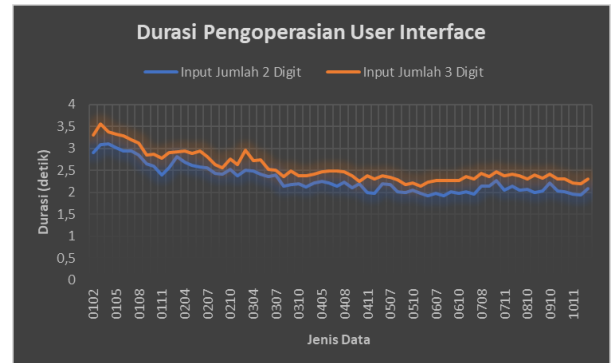
- a) [*User ID*], merupakan kombinasi 6 digit angka sebagai identitas dari pengguna.
- b) *Username*, merupakan identitas nama dari pengguna yang terdiri dari maksimal 20 karakter.
- c) [*Login Status*], merupakan data yang terdiri dari dua kondisi (*LOGIN* dan *LOGOUT*).
- d) [*Login DateTime*], merupakan data waktu yang menunjukkan kapan terakhir kali pengguna *login*.
- e) [*Logout DateTime*], merupakan data waktu yang menunjukkan kapan terakhir kali pengguna *logout*.

User ID	Username	Login Status	Login DateTime	Logout DateTime
200106	Muhammad Fahar Rizki	LOGIN	2023-12-21 17:04:11.787	NULL
201006	Mhd Rizki	LOGOUT	2023-12-25 20:21:06.813	2023-12-25 20:24:09.157

Gambar 9 Tabel *User\_LoginTimeline* pada Basis Data

## C. Pengujian

### 1) Pengujian Waktu Pengoperasian User Interface



Gambar 10 Grafik Durasi Pengoperasian *User Interface*

Pada Gambar 10 terlihat bahwa durasi pengoperasian *user interface* cenderung berada pada rentang waktu 2 – 2,5 detik. Waktu tersebut didapat karena posisi *layout* pada *button* digit angka dan *button* operasi yang lebih didominasi pada posisi atas sehingga proses *input* yang melibatkan digit angka 7, 8, dan 9 memiliki durasi pengoperasian yang lebih singkat.

### 2) Pengujian Konektivitas Interface dengan Basis Data

Tabel 4 Perbandingan Waktu Penambahan Data Baru antara *User Interface* dengan Basis Data

Percobaan	Waktu ( <i>real-time</i> )		Delay ( detik )
	Interface	Basis Data	
1	16 : 27 : 12 ,530	16 : 27 : 13 ,127	0,597
2	16 : 32 : 59 ,595	16 : 32 : 59 ,610	0,015
3	16 : 35 : 57 ,774	16 : 35 : 57 ,793	0,019
4	16 : 40 : 16 ,576	16 : 40 : 16 ,603	0,027
5	16 : 42 : 21 ,238	16 : 42 : 21 ,243	0,005
6	16 : 45 : 14 ,679	16 : 45 : 14 ,697	0,018
7	16 : 48 : 21 ,580	16 : 48 : 21 ,610	0,03
8	16 : 53 : 16 ,926	16 : 53 : 16 ,933	0,007
9	16 : 55 : 28 ,888	16 : 55 : 28 ,903	0,015
10	16 : 56 : 29 ,239	16 : 56 : 29 ,257	0,018
<b>Rata - Rata</b>			<b>0,0751</b>

Pada tabel 4 terlihat bahwa pada percobaan 1 terdapat perbedaan waktu antara *user interface* pada basis data yang relatif signifikan dibandingkan percobaan lainnya. Hal ini bisa terjadi dikarenakan pada percobaan 1, basis data tidak memiliki data apapun di dalamnya sehingga proses penambahan data baru memerlukan waktu yang lebih lama.

Tabel 5 Perbandingan Waktu Pembaruan Data antara *User Interface* dengan Basis Data

Percobaan	Waktu ( <i>real-time</i> )		Delay ( detik )
	Interface	Basis Data	
1	10 : 53 : 38 ,400	10 : 53 : 48 ,420	0,02
2	10 : 57 : 37 ,973	10 : 57 : 37 ,973	0
3	10 : 59 : 50 ,723	10 : 59 : 50 ,723	0

4	11 : 01 : 57 ,523	11 : 01 : 57 ,523	0
5	11 : 11 : 18 ,679	11 : 11 : 18 ,680	0,001
6	11 : 13 : 02 ,991	11 : 13 : 02 ,993	0,002
7	11 : 14 : 31 ,290	11 : 14 : 31 ,290	0
8	11 : 15 : 49 ,708	11 : 15 : 49 ,710	0,002
9	11 : 17 : 33 ,978	11 : 17 : 33 ,980	0,002
10	11 : 19 : 56 ,129	11 : 19 : 56 ,177	0,048
<b>Rata - Rata</b>			<b>0,0075</b>

Berdasarkan pada tabel 3 dan tabel 4 terlihat bahwa rata-rata jeda waktu (*delay*) pada saat melakukan pembaruan data hanya memerlukan 10% dari jeda waktu pada saat melakukan penambahan data. Hal ini dapat terjadi karena proses pembaruan data tidak memerlukan proses inisiasi tipe data yang akan disimpan pada basis data, berbeda halnya dengan proses penambahan data baru dimana inisiasi tipe data perlu dilakukan.

### 3) Pengujian Operasional User Interface dan Basis Data

Tabel 6 Hasil Pengujian Tingkat Keberhasilan pada *User Interface*

Jenis Percobaan	Indikasi Keberhasilan	Status Keberhasilan
Aktivasi <i>button</i> "ON"	Indikator pada <i>Socket Status</i> berwarna hijau sesuai dengan <i>input</i> data yang ditampilkan pada "Current Data"	Berhasil
Aktivasi <i>button</i> "OFF"	Semua indikator pada <i>Socket Status</i> mulai dari 1A sampai 12A berwarna kuning	Berhasil
Indikator berwarna merah	Indikator pada <i>Socket Status</i> yang berwarna hijau berubah menjadi warna merah	Berhasil
Aktivasi <i>Timer</i>	Ditampilkan <i>timer</i> pada <i>Timer Left</i>	Berhasil
<i>Remaining</i>	Perubahan nilai <i>remaining</i> pada semua indikator berwarna merah	Berhasil
Aktivasi <i>button</i> "Logout"	<i>User</i> diarahkan ke halaman <i>login</i>	Berhasil
<i>Connected and Disconnected Port</i>	Pesan yang ditampilkan pada <i>Connection Status</i> adalah <b>Connected</b> dan semua <i>button</i> yang ada pada <i>Input Box</i> dapat dioperasikan	Berhasil
	Pesan yang ditampilkan pada <i>Connection Status</i> adalah <b>Disconnected</b> dan semua <i>button</i> yang ada pada <i>Input Box</i> tidak dapat dioperasikan	Berhasil

Berdasarkan pada tabel 5 terlihat bahwa keseluruhan percobaan dinyatakan telah berhasil sehingga dengan menerapkan persamaan ( 1 ), maka didapat :

$$\text{Tingkat Keberhasilan (\%)} = \frac{\text{Total Percobaan Berhasil}}{\text{Total Jenis Percobaan}} \times 100\%$$

$$\text{Tingkat Keberhasilan (\%)} = \frac{8}{8} \times 100\%$$

$$\text{Tingkat Keberhasilan (\%)} = 100\%$$

Dengan persentase tersebut maka dapat disimpulkan bahwa *user interface* yang telah dirancang dapat berfungsi dengan baik dan sesuai harapan.

Tabel 7 Hasil Pengujian Keberhasilan pada Basis Data

Tabel Basis Data	Jenis Percobaan	Indikasi Keberhasilan	Status Keberhasilan
<i>User Login</i>	Pembaruan data	Perubahan data pada kolom <i>DateTime</i> , <i>LastUpdate</i> dan <i>Passcode</i>	Berhasil
	Penambahan data baru	Ditambahkannya data baru pada tabel <i>User_Login</i>	Berhasil
<i>Retrieval Data</i>	Pembaruan data	Perubahan data pada kolom <b>Total Kabel Disortir</b> , <b>Kabel Sortir Diterima</b> , dan <b>Kabel Sortir Ditolak</b>	Berhasil
<i>User LoginTimeline</i>	Pembaruan data status "Login/Logout"	Perubahan data pada kolom <i>Login Status</i>	Berhasil
	Pembaruan waktu ketika status <i>Login/Logout</i>	Perubahan data pada kolom <i>Login DateTime</i> ketika <i>Login</i> dan <i>Logout DateTime</i> ketika <i>Logout</i>	Gagal

Pada tabel 6 terjadi kegagalan pada saat pembaruan waktu ketika dalam status *login* maupun *logout*. Hal ini terjadi karena pada saat data waktu diperbarui pada salah satu kolom (kolom *Login DateTime* dan *Logout DateTime*), kolom yang lainnya juga ikut diperbarui dengan data waktu yang sama. Oleh karena itu, perlu adanya pembeda antara kolom *Login DateTime* dan *Logout DateTime* sehingga pada status *Login* maka data waktu hanya diperbarui pada kolom *Login DateTime*, begitu juga dengan status *Logout*.

Dengan menerapkan persamaan ( 1 ), maka didapat :

$$\text{Tingkat Keberhasilan (\%)} = \frac{\text{Total Percobaan Berhasil}}{\text{Total Jenis Percobaan}} \times 100\%$$

$$\text{Tingkat Keberhasilan (\%)} = \frac{4}{5} \times 100\%$$

$$\text{Tingkat Keberhasilan (\%)} = 80\%$$

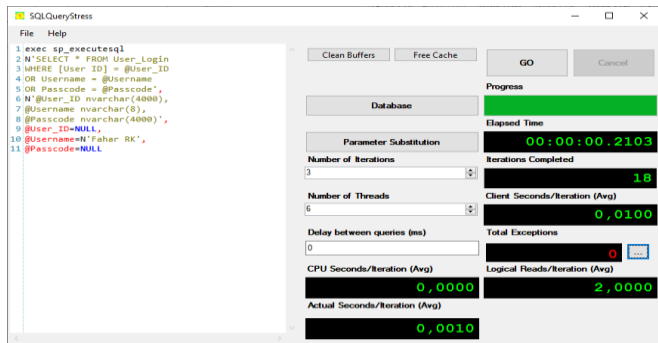
Dengan persentase yang mencapai 80 %, maka dapat disimpulkan bahwa basis data yang dirancang dapat berfungsi dengan baik namun perlu adanya perhatian tertentu terhadap kumpulan data yang telah tersimpan pada basis data, terutama data mengenai tanggal dan waktu.

#### 4) Pengujian Performa Basis Data

Keseluruhan dari kueri yang telah dieksekusi selama *user interface* beroperasi akan ditampilkan pada *SQL Server Profiler* dengan hasil seperti pada Gambar 11. Kemudian masing-masing kueri akan dilakukan pengujian beban kerja pada kueri melalui *SQL Query Stress* dengan menentukan total kueri tersebut dieksekusi dan menentukan asumsi total terjadinya tabrakan data dalam satu waktu (*thread*) seperti pada gambar 12. Hasil dari pengujian ini ditampilkan pada tabel 8. Selain itu, pengujian beban kerja juga dilakukan pada tabel basis data melalui *SQL Server Management Studio* dengan hasil pengujian seperti pada gambar 13.



Gambar 11 Tampilan Daftar Kueri yang Dieksekusi pada *SQL Server Profiler*



Gambar 12 Tampilan Uji Pengaturan Beban Kueri pada *SQL Query Stress*

#### 1. Pengujian Beban Kerja Kueri pada Basis Data

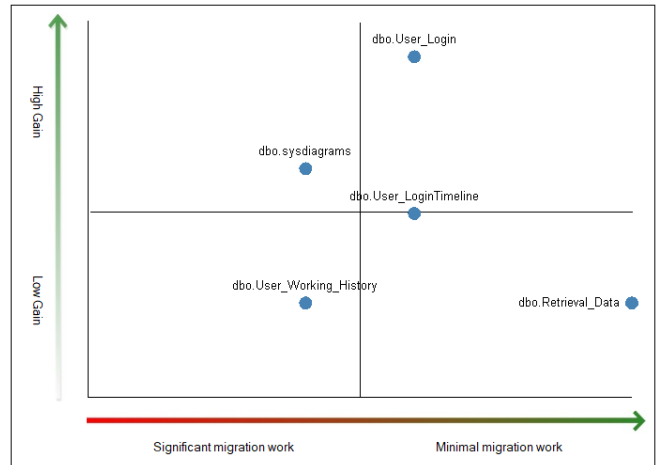
Tabel 8 Hasil Pengujian Beban Kerja Kueri pada *SQL Query Stress*

Percobaan	Total Eksekusi	Total Thread	Waktu per eksekusi (milidetik)
1	3	4	0
2	3	5	0
3	3	6	10
4	3	7	17
5	6	1	0
6	6	2	10
7	6	3	15
8	6	4	20

Pada tabel 8, total *thread* yang dimaksud menunjukkan jumlah data yang dikirim secara bersamaan (tabrakan data) dalam satu kali proses eksekusi. Ketika waktu per eksekusi menunjukkan nilai diatas 0, maka total *thread* tersebut menjadi batasan maksimum kueri dapat bekerja dengan normal. Dari hasil pengujian ini menunjukkan bahwa ketika suatu kueri dieksekusi lebih banyak, maka batasan jumlah tabrakan data

yang terjadi juga menjadi lebih sedikit agar basis data dapat bekerja dengan normal.

#### 2. Pengujian Beban Kerja Tabel pada Basis Data



Gambar 13 Grafik Performa Tabel pada Basis Data

Pada gambar 13, grafik pada axis-X (*migration work*) menunjukkan besarnya usaha atau proses optimasi yang diperlukan, dimana semakin ke kanan maka proses optimasi memori yang diperlukan semakin sedikit. Sementara pada axis-Y (*gain*) menunjukkan besarnya hasil optimasi penggunaan memori yang diperoleh ketika melakukan optimasi. Pada gambar 13 juga terlihat bahwa tabel *User\_Login* memiliki potensi yang tinggi untuk dilakukannya optimasi memori ini karena penggunaan kueri yang lebih banyak dialokasikan pada tabel ini, sehingga beban kerja yang diterima pada tabel ini menjadi lebih berat.

#### IV. KESIMPULAN

Berdasarkan hasil temuan dari beberapa pengujian, mulai dari pengujian terhadap *user interface* sampai pengujian terhadap basis data, sistem ini dinilai berhasil dalam hal *monitoring* proses pengambilan kabel melalui penggunaan *user interface*. Hal ini didasari oleh sistem *user interface* yang telah sinkron terhadap sistem *hardware* dan dapat terhubung dengan basis data. Di sisi lain, penerapan basis data pada sistem ini juga dinilai dapat membantu dalam mencatat seluruh data yang terkalit dengan proses yang terjadi selama pengambilan kabel. Namun, pada data tertentu seperti data tanggal dan waktu masih ditemukan adanya *error* berupa pembaruan data tanggal dan waktu pada dua kolom tabel yang berbeda dan terjadi dalam waktu yang bersamaan. Oleh karena itu, pada tabel yang terdapat pada basis data dan melibatkan data berupa tanggal dan waktu, diperlukan adanya perhatian khusus terhadap data tersebut.

#### REFERENSI

[1] Sandra, R. Damayanti, Y. Hendrawan, B. Susilo, and S. Oktavia, "Prediction of tomatoes maturity using TCS3200 color sensor," in *IOP Conference Series: Earth and Environmental Science*, Institute of Physics Publishing, May 2020. doi: 10.1088/1755-1315/475/1/012011.

- [2] S. Khairunnisa, R. Amelza, N. Azizah Lubis, and M. Darma Putri, "Kajian Spectrometer Menggunakan Sensor Cahaya TCS3200."
- [3] A. Wibowo, I. Parlina, A. Wanto Teknik Informatika, S. Tunas Bangsa Pematangsiantar, and R. Artikel, "RANCANG BANGUN MESIN SORTIR BUAH KELAPA SAWIT BERDASARKAN TINGKAT KEMATANGAN MENGGUNAKAN SENSOR WARNA TCS3200 BERBASIS ARDUINO UNO INFO ARTIKEL ABSTRAK," vol. 1, no. 2, pp. 9–15, 2022, doi: 10.55123.
- [4] K. Sugiarto, K. Sulthana Hilmi, A. Pradana, and I. Fadhillah, "Identifikasi Zona Warna Dengan Sensor TCS3200 Pada Robot KRSTI," *Jurnal Sistim Informasi dan Teknologi*, pp. 32–37, Dec. 2023, doi: 10.60083/jsisfotek.v5i4.320.
- [5] F.- Puspasari, I.- Fahrurrozi, T. P. Satya, G.- Setyawan, M. R. Al Fauzan, and E. M. D. Admoko, "Sensor Ultrasonik HCSR04 Berbasis Arduino Due Untuk Sistem Monitoring Ketinggian," *Jurnal Fisika dan Aplikasinya*, vol. 15, no. 2, p. 36, Jun. 2019, doi: 10.12962/j24604682.v15i2.4393.
- [6] M. Mutava Gabriel and K. Paul Kuria, "Arduino Uno, Ultrasonic Sensor HC-SR04 Motion Detector with Display of Distance in the LCD." [Online]. Available: [www.ijert.org](http://www.ijert.org)
- [7] I. H. Santoso and A. I. Irawan, "Analisis Perbandingan Kinerja Sensor Jarak HC-SR04 dan GP2Y0A21YK Dengan Menggunakan Thingspeak dan Wireshark," *Jurnal Rekayasa Elektrika*, vol. 18, no. 1, Apr. 2022, doi: 10.17529/jre.v18i1.23359.
- [8] D. Somadani and A. H. Ginanjar, "PROTOTIPE PENERANGAN JALAN UMUM (PJU) PINTAR BERBASIS ARDUINO MENGGUNAKAN SOLAR PANEL, SENSOR HC-SR04 DAN SENSOR LDR," 2018.
- [9] P. Kejĳik, C. Kluser, R. Bischofberger, and R. S. Popovic, "A low-cost inductive proximity sensor for industrial applications," *Sens Actuators A Phys*, vol. 110, no. 1, pp. 93–97, 2004, doi: <https://doi.org/10.1016/j.sna.2003.07.007>.
- [10] A. Supargo, S. Negeri, and P. Trenggalek, "Penerapan Metode Pembelajaran Direct Instructions dalam Upaya Peningkatan Prestasi belajar Penjaskes yang Memuat Renang Pada Siswa Kelas X TKJ 1 Semester 1 SMK Negeri 1 Pogalan Trenggalek Tahun," *Jurnal Terapan Pendidikan Dasar dan Menengah* /, vol. 66, no. 1, 2021, doi: 10.28926/jtpdm.v1i1.1.
- [11] Z. Abidin Jurusan Biologi Fakultas Sains Dan, "PERSENTASE TINGKAT KEBERHASILAN PENETASAN TELUR PENYU SISIK (*Eretmochelys imbricata* L.) PADA SARANG ALAMI DI PULAU PRAMUKA DAN PULAU KOTOK KECIL KEPULAUAN SERIBU," 2013.