

Konsep Sistem Terdistribusi pada pelayanan terpadu kalurahan menggunakan TCP

Joko Triyono^{1*}, Elroy Abram Anugraha Sitepu², Adi Kurniawan³

¹Rekayasa Sistem Komputer/Fakultas
Sains dan Teknologi Informasi
Universitas Akprind Indonesia

^{1*}jack@akprind.ac.id

²Rekayasa Sistem Komputer/Fakultas
Sains dan Teknologi Informasi
Universitas Akprind Indonesia

²royroy49822@gmail.com

³Rekayasa Sistem Komputer/Fakultas
Sains dan Teknologi Informasi
Universitas Akprind Indonesia

³adiku004@gmail.com

Abstrak— Komunikasi data dalam dunia nyata selamanya harus selalu menggunakan jalur internet sepenuhnya ada beberapa bagian sistem data yang lebih baik tetap berapa di jalur lokal dengan alasan keamanan. Dalam pelayanan Sistem Informasi Kependudukan Kalurahan walaupun sudah menggunakan aplikasi akan tetapi sifatnya personal, tidak bisa menjangkau pihak masyarakat secara langsung. Beberapa penelitian tentang sistem terdistribusi telah memberikan hasil bahwa pengembangan model sistem terdistribusi akan memberikan dampak positif untuk meningkatkan nilai dari informasi. Dimana sistem tidak harus semuanya online, akan tetapi bisa dengan semi online atau bahkan bersifat lokal dan bisa diakses oleh pihak lain menggunakan algoritma tertentu. Transmission Control Protokol menjadi salah satu protokol yang cukup handal untuk mengatasi komunikasi data, melalui port. Dengan menempatkan sebuah Server TCP, maka Client TCP dimanapun berada selama bisa mengakses Server tersebut akan bisa melakukan komunikasi data. Dalam penelitian ini akan diuji diuji apakah aplikasi tersebar bisa saling berkomunikasi menggunakan TCP.

Kata kunci — Client, Information, Server, TCP .

Abstract— Real-world data communication must always use the internet entirely; however, some parts of the data system are better left on local networks for security reasons. In the Village Population Information System service, although applications are used, they are personal in nature and cannot reach the public directly. Several studies on distributed systems have shown that developing a distributed system model will have a positive impact on increasing the value of information. The system does not have to be entirely online; it can be semi-online or even local, accessible by other parties using specific algorithms. Transmission Control Protocol (TCP) is a reliable protocol for handling data communication through ports. By placing a TCP server, TCP clients, wherever they are, as long as they can access the server, can communicate. This study will test whether distributed applications can communicate with each other using TCP.

Keyword — Client, Information, Server, TCP .

I. PENDAHULUAN

Komunikasi data dalam dunia nyata selamanya harus selalu menggunakan jalur internet sepenuhnya, ada beberapa bagian sistem atau data yang lebih baik tetap berada di jalur local dengan alasan keamanan atau tidak tersedianya sarana dan prasarana online yang memadai. Termasuk dalam pelayanan Sistem Informasi Kependudukan (SISDUK) Kalurahan, walaupun sistem informasi atau aplikasi sudah tersedia akan tetapi masyarakat atau pengguna hanya bisa mengakses melalui perantara operator SISDUK, saat masyarakat membutuhkan dokumen atau informasi terkait kependudukan, maka masyarakat harus melalui beberapa tahapan seperti dari RT, RW, Dukuh baru ke bagian Administrasi kalurahan. Dalam beberapa kasus penerapan teknologi beberapa kalurahan telah menyediakan layanan melalui Medsos seperti

WA untuk menjembatani permasalahan tersebut diatas dan masih banyak kendala yang muncul karena inkonsistensi data.

Beberapa pustaka yang menguatkan perlunya penelitian ini adalah konsep komunikasi data antar sistem informasi menggunakan transmisi kontrol protokol yang bisa digunakan untuk komunikasi data dalam jaringan komputer dan memastikan data terkirim [1], Juga tentang sistem berbasis MySQL untuk sinkronisasi data cabang dan pusat [2], Pendekatan lain menggunakan REST API dan format JSON untuk pertukaran data antar-cabang terbukti fleksibel dalam menangani data semi-terstruktur [3], rancang bangun sistem terdistribusi pada toko roti valentine tanjung jabung barat untuk peningkatan efektifitas sistem [4], Juga tentang tentang Optimalisasi Manajemen Terdistribusi pada Sistem Operasi Cloud Computing bahwa pendekatan algoritma prediksi dalam lingkungan

cloud dapat mengurangi waktu respon secara signifikan juga memungkinkan skalabilitas yang lebih baik dan cepat [5], juga tentang implementasi sistem terdistribusi pada replikasi dan web service [6], Web Service untuk transaksi data pada aplikasi sistem informasi kelembagaan petani dengan metode REST dengan output berupa JSON untuk komunikasi dan transaksi data pada aplikasi [7], tentang model aplikasi terdistribusi menggunakan FTP [8], Model Kontrol Transaksi RDBMS Menggunakan Trigger dan Waktu Server [9] yang membahas tentang pemanfaatan trigger dan waktu server, tentang Studi perkembangan dan implementasi sistem basis data terdistribusi: literature review menghasilkan bahwa konsep sistem basisdata terdistribusi saat ini masih digunakan untuk mendukung tata kelola sistem informasi yang mengashilkan informasi yang sesuai dengan tujuan pengguna [10]. Ditambahkan dengan beberapa teori yang mendukung sistem terdistribusi dan database terdistribusi [11], [12], [13], [14], [15], [16], [17].

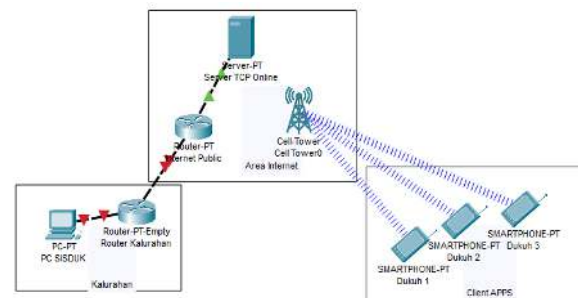
Bagian PENDAHULUAN membahas latar belakang masalah, tinjauan pustaka secara ringkas, maksud dan tujuan riset dilakukan.

Untuk menjawab permasalahan tersebut, dibutuhkan sebuah sistem informasi terdistribusi yang memungkinkan pengguna mengakses melalui aplikasi kecil di smartphone dari area publik ke SISDUK yang ada di mesin kalurahan walaupun tidak ditempatkan di area publik (online) dengan tetap menjaga keamanan, konsistensi dan integritas data. Dengan penambahan sebuah aplikasi pemandu yang biasa disebut "server" pada area publik (online). Dengan menggunakan teknik client server chatting melalui jalur port yang telah disepakati, akan sangat memungkinkan komunikasi data bisa terjalin. Melalui studi kasus pelayanan SISDUK terdistribusi diharapkan penelitian ini bisa melahirkan konsep sistem informasi terdistribusi dan menjadi model aplikatif untuk meningkatkan layanan serta nilai sebuah informasi atau data yang telah ada dan tidak perlu di letakkan pada area publik/online. Sehingga diharapkan akan banyak aplikasi-aplikasi yang telah digunakan dan bersifat localhost bisa dieksploitasi datanya

untuk kebutuhan sistem atau informasi yang lain yang terpisah secara geografis dan jaringan.

II. METODOLOGI PENELITIAN

Metode penelitian ini dilakukan dalam skala laboratorium, dimana penelitian ini dilakukan di laboratorium komputer jaringan di bawah fakultas Sains dan Teknologi Informasi Universitas Akprind Indonesia. Data dan aplikasi serta infrastruktur jaringan yang digunakan sebagai simulasi adalah (1) sebuah sistem informasi kependudukan berbasis web (simulasi) sebagai penyedia dan pengolah data SISDUK, (2) Aplikasi Server Chat berbasis TCP yang di tempatkan pada sebuah server sebagai pemandu komunikasi antar klient, (3) Aplikasi Desktop Client SISDUK, yang ditempatkan bersama dengan sistem informasi SISDUK, dimana client ini akan menerima stream dari luar untuk diproses ke sistem informasi SISDUK dan memberikan jawaban ke pihak peminta informasi, (4) Client Mobile (APP), yang akan ditanam di smartphone pengguna dalam hal ini Dukuh, dari aplikasi ini client akan berkomunikasi dengan Aplikasi Desktop Client SISDUK untuk mendapatkan informasi yang dibutuhkan. Seperti melaporkan kematian penduduk untuk di proses akta kematian, pembuatan surat pengantar, dan beberapa fasilitas pelayanan kependudukan yang bisa di eksplor dari sistem informasi SISDUK yang telah ada. Gambar 1 menunjukkan desain arsitektur sistem secara umum.



Gambar 1. Design Arsitektur Sistem
Area Internet, adalah area publik tempat hosting untuk menempatkan Aplikasi Server, yang bisa diakses melalui jalur internet secara umum.
Kalurahan, adalah lokasi dimana Aplikasi SISDUK yang telah ada dan Client Desktop SISDUK (aktif dengan account ClientSISDUK)

di tempatkan, lokasi ini akan selalu terhubung ke internet melalui infrastruktur yang telah ada.

Client APPS, adalah device smartphone atau mobile device yang dibawa oleh Dukuh, dimana didalamnya di install aplikasi Client APPS SISDUK yang aktif dengan account masing-masing dukuh.

Tahap-tahap yang akan dilakukan adalah

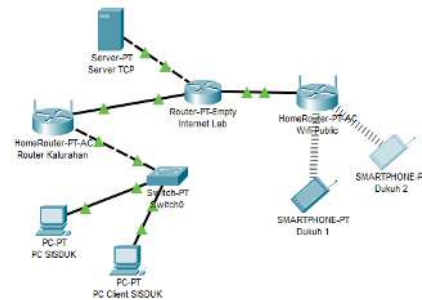
1. Instalasi Arsitektur jaringan simulasi skala laboratorium.
2. Pembuatan user baru pada database aplikasi SISDUK dengan account privileges hanya SELECT dan Pembuatan VIEW untuk masing-masing fasilitas yang akan diperbolehkan untuk di akses oleh pihak luar, dan diberikan akses kepada user baru.
3. Pemasangan Aplikasi Server TCP pada server internet.
4. Pemasangan Aplikasi Client Desktop pada komputer di kalurahan
5. Instalasi Client APPS SISDUK pada device/smartphone dukuh
6. Melakukan ujicoba koneksi ke server dari masing-masing Client.
7. Melakukan ujicoba transaksi dari Client APPS SISDUK, dan Client Desktop SISDUK melihat notifikasi.
8. Algoritma komunikasi data

III. HASIL DAN PEMBAHASAN

Hasil penelitian setelah dilakukan ujicoba skala laboratorium adalah semua model sistem terdistribusi dengan sebuah server publik dan beberapa client lokal hanya bisa berhubungan dengan server publik dan tidak bisa dengan sesama client secara jaringan.

A. Instalasi Arsitektur Jaringan Skala Laboratorium

Dengan menggunakan software simulator bentuk instalasi jaringan seperti terlihat pada gambar 2 dan pemetaan IP seperti terlihat pada tabel 1.

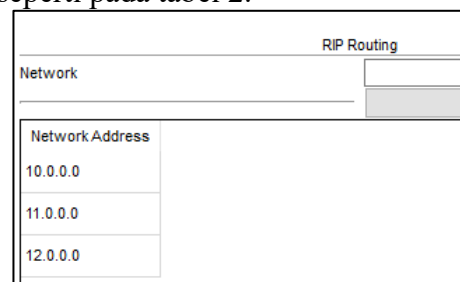


Gambar 2. Instalasi Jaringan

Tabel 1 Pemetaan IP Address

Device	Interface	IP Address
Router Internet Lab	Ethernet7/0	11.11.11.2/8
	Ethernet8/0	12.12.12.1/8
	Ethernet9/0	10.10.10.1/8
Router Kalurahan	Internet	10.10.10.2/8
	Lan	192.168.1.1/24
Wifi Public	Internet	12.12.12.2/8
	Lan	192.168.0.1/28
Server Public	FastEthernet/0	11.11.11.1/8
PC SISDUK	FastEthernet/0	192.168.1.2/24
PC Client SISDUK	FastEthernet/0	192.168.1.3/24
Smartphone Dukuh 1	Wireless0	192.168.0.4/28
Smartphone Dukuh 2	Wireless0	192.168.0.5/28

Setelah dilakukan setting routing menggunakan dinamic routing RIP pada router Internet lab seperti terlihat pada gambar 3 akan dibuktikan bahwa jaringan sudah sesuai dengan keadaan sesungguhnya dengan pengujian ping dengan hasil seperti pada tabel 2.



Gambar 3. Setting Routing RIP

Dari tabel 2 terlihat bahwa Server public bisa di ping oleh semua client, sedangkan antar client diluar jaringan tidak bisa di ping,

Tabel 2. Pengujian Ping

Dari	Tujuan	Hasil
PC SISDUK	Server Public	berhasi 1
	PC Client SISDUK	berhasi 1
	Smartphone Dukuh 1	gagal
	Smartphone Dukuh 2	Gagal
PC Client SISDUK	Server Public	berhasi 1

	PC SISDUK	berhasi
		1
	Smartphone Dukuh 1	gagal
	Smartphone Dukuh 2	Gagal
Smartphone Dukuh 1	Server Public	Berhasi
		1
	PC Client SISDUK	gagal
	PC SISDUK	gagal
Smartphone Dukuh 2	Server Public	Berhasi
		1
	PC SISDUK	Gagal
	PC Client SISDUK	Gagal



Gambar 4. Aplikasi Server TCP

B. Pembuatan user baru pada database aplikasi SISDUK dengan account privileges hanya SELECT

Dengan alasan keamanan data pada SISDUK yang telah ada, maka untuk mengakses database perlu dibuatkan user baru dengan privileges hanya bisa melihat (SELECT) pada database tersebut.

```
CREATE USER 'sisduk'@'%' IDENTIFIED VIA mysql native password USING '***'; GRANT USAGE ON *.* TO 'sisduk'@'%' REQUIRE NONE WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0 MAX_USER_CONNECTIONS 0;
```

```
REVOKE ALL PRIVILEGES ON `db\_masyarakat\_android`. * FROM 'sisduk'@'%'; GRANT SELECT, SHOW VIEW, EXECUTE ON `db\_masyarakat\_android`. * TO 'sisduk'@'%';
```

Dengan privileges tersebut maka user 'sisduk' hanya akan bisa mengakses (melihat) database 'SISDUK'.

C. Pemasangan Aplikasi Server TCP pada server internet.

Server diletakkan pada ip 11.11.11.1 yang bisa diakses dari semua device, aplikasi Server TCP di letakkan di folder /home/jarkomdb/servertcp dengan nama file Server.class. Untuk selanjutnya aplikasi tersebut dijalankan pada mode terminal dengan perintah `java Server`. Gambar 4 menunjukkan proses menjalankan aplikasi Server TCP.

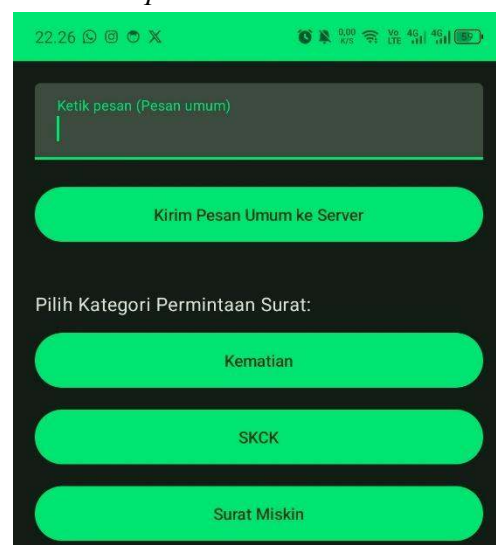
D. Pemasangan Aplikasi Client SISDUK pada komputer di kalurahan



Gambar 5. Aplikasi Client SISDUK

Aplikasi Client SISDUK diletakkan pada mesin dengan IP 192.168.1.2, dijalankan dalam mode terminal, setelah running aplikasi tersebut akan selalu merespon secara otomatis setiap request yang masuk dan memberikan jawaban sesuai dengan request, aplikasi ini setelah di running tidak memerlukan campur tangan operator sampai komputer dimatikan, gambar 5 menunjukkan tampilan aplikasi Client SISDUK.

E. Instalasi Client APPS SISDUK pada device/smartphone dukuh



Gambar 6. Tampilan Aplikasi APPS SISDUK

Instalasi Client APPS SISDUK di smartphone bisa dilakukan dengan melakukan setup/install apps tersebut di masing-masing smartphone dukuh, model yang dilakukan pada penelitian ini dengan export aplikasi dari android studio ke smartphone yang dituju. Gambar 6 menunjukkan tampilan aplikasi Client APPS SISDUK.

Dalam penelitian ini ujicoba dilakukan pada 3 fasilitas layanan yaitu layanan permintaan surat Kematian untuk diterbitkan Akta Kematian oleh Kalurahan, kedua permintaan surat pengantar SKCK dan ketiga Surat Keterangan Miskin.

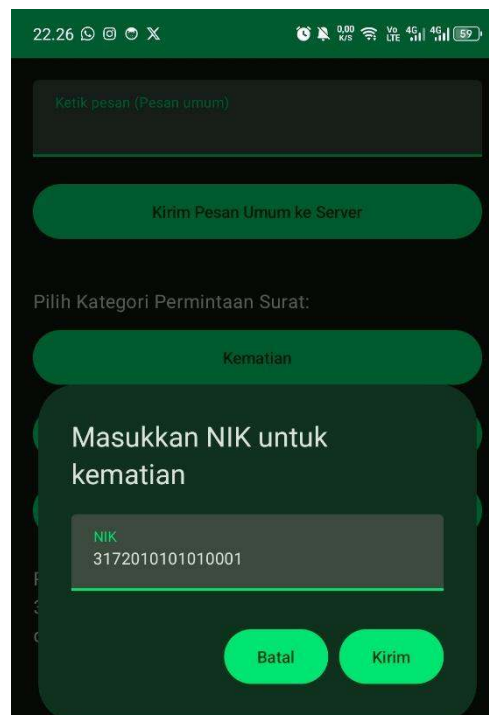
F. Melakukan ujicoba koneksi ke server dari masing-masing Client.

Pada saat client dijalankan maka server akan menampilkan client yang terhubung secara otomatis, pertama kali yang harus dijalankan adalah ServerTCP, selanjutnya Client bisa dijalankan. Pada saat Client SISDUK dijalankan seperti diperlihatkan di gambar 5, maka Server akan menampilkan informasi seperti pada gambar 4 dengan notifikasi “*sisduk terhubung*”. Begitu juga saat Client APPS SISDUK dijalankan seperti gambar 6, maka Server juga akan menampilkan informasi seperti pada gambar 4 dengan notifikasi “*dukuh 1 terhubung*”.

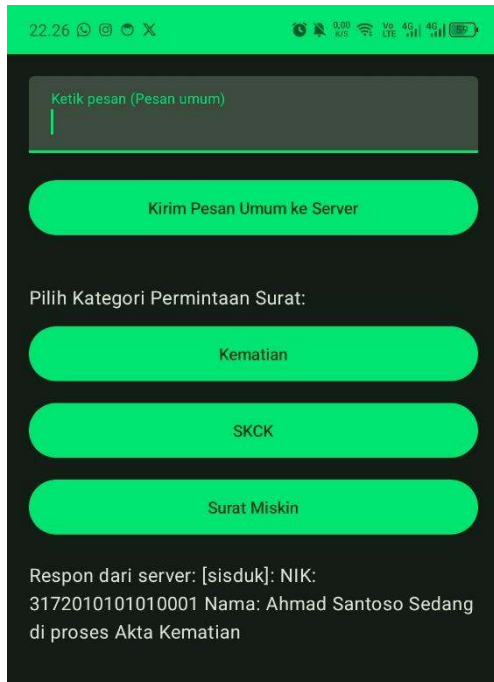
G. Melakukan ujicoba transaksi dari Client APPS SISDUK, dan Client Desktop SISDUK melihat notifikasi.

Untuk melakukan ujicoba bagian ini kita mulai dari Aplikasi APPS SISDUK seperti pada gambar 6.

Uji Coba Permintaan Akte Kematian



Gambar 7. Tampilan Memasukkan NIK
Diawali dengan menekan tombol “*Kematian*” maka akan ditampilkan halaman berikutnya untuk memasukkan NIK orang yang meninggal tersebut. Gambar 7 menunjukkan tampilan halaman memasukkan NIK, lalu selanjutnya menekan tombol “*Kirim*”, maka permintaan tersebut akan dikirimkan ke Client SISDUK melalui ServerTCP dengan format data @:*sisduk:kematian:NIK*, maka server tcp akan meneruskan ke client SISDUK, saat Client SISDUK menerima permintaan tersebut, pesat/data akan diuraikan dicarikan data kependudukan tersebut berdasarkan NIK Gambar 5 Menunjukkan saat ada permintaan dari dukuh., selanjutnya Client SISDUK akan mengirimkan kembali data hasil pencarian kepada Client APPS SISDUK, baik itu ketemu atau tidak ditemukan hasil akan terlihat seperti pada gambar 8.



Gambar 8 Hasil Respon Server

Pada gambar 8 terlihat hasil respon dari server bahwa data NIK yang dimasukkan sudah diberikan jawaban dengan memberikan identitas nama dari NIK yang dilaporkan. Begitu juga selanjutnya untuk Permintaan Pengantar SKCK dan Permintaan Surat Keterangan Miskin akan dilakukan dengan cara yang sama.

H. Algoritma Komunikasi Data

Dalam penelitian ini sesuai dengan skema dari design arsitektur sistem pada gambar 1

- Server TCP akan running dengan membuka port layanan 123456

```
private static final int PORT = 12345;
```

- Client SISDUK akan secara otomatis membuat account dengan nama “sisduk”, dengan mengirimkan output stream ke server saat pertama kali aplikasi running seperti di perlihatkan pada gambar 9.

```
import java.io.*;
import java.net.*;

public class Client {
    public static void main(String[] args) throws IOExce
        Socket socket = new Socket("11.11.11.1", 12345);

    BufferedReader in = new BufferedReader(new
        InputStreamReader(socket.getInputStream()));
    PrintWriter out = new PrintWriter(socket.getOutp
        ()); true);
    out.println("sisduk");
    BufferedReader console = new BufferedReader(new
        InputStreamReader(System.in));
```

Gambar 9 Script 1

Pada baris 6 adalah saat client membuat socket untuk menghubungi ServerTCP pada IP 11.11.11.1 dan port 12345 selanjutnya pada baris 10 client akan mengirimkan output stream pertama kali ke server dengan data “sisduk”, yang artinya akan dibaca oleh server sebagai login account sisduk ke sistem. Gambar 10 menunjukkan posisi script server saat menerima input stream dari client yang pertama kali.

```
try {
    in = new BufferedReader(new InputStreame
    socket.getInputStream());
    out = new PrintWriter(socket.getOutputStre
    true);
    // Dapatkan username
    username = in.readLine();
    synchronized (clients) {
        clients.put(username, out);
    }
    System.out.println(username + " terhubung.
    String message;
    while ((message = in.readLine()) != null)
        if (message.startsWith("#:")) {
```

Gambar 10 Script Server 2

Pada baris 34 server membaca input stream lalu melakukan sinkronisasi data client tersebut ke daftar client, jika sukses maka akan ditampilkan notifikasi client/username tersebut terhubung pada baris 38.

- Client APPS SISDUK saat dijalankan pertama kali akan diminta memasukkan nama dukuh kedalam sistem gambar script 11 menunjukkan posisi memasukkan nama dukuh.

```
BufferedReader console = new BufferedRea
    InputStreamReader(System.in));
    String namadukuh;
    System.print("Masukkan Nama Dukuh :");
    namadukuh = console.readLine();
    out.println(namadukuh);
    // Listener thread untuk menerima pesan
    new Thread() -> {
```

Gamba 11. Script Client Dukuh

Pada baris 12 sampai 15 menunjukkan proses untuk meminta memasukkan nama dukuh dan akan dikirim ke server sebagai output stream yang pertama kali serta akan direspon seperti pada gambar 10.

- Permintaan surat keterangan dari Client APPS SISDUK.

Konsep dari permintaan ini mengikuti aturan dengan private message antara user dukuh dengan user sisduk dengan format dasar @:sisduk:key:NIP, dimana @ sebagai kode unik awal, sisduk sebagai user yang dituju dan NIP sebagai data NIP yang akan dicari.

Pada gambar script 12 memperlihatkan script saat akan memilih layanan.

```

5 // New category buttons
6 Text(text = "Pilih Kategori Permintaan Sur
7 Spacer(modifier = Modifier.height(8.dp))
8
9 Button(
10     onClick = {
11         currentCategory = "kematian"
12         showNikDialog = true
13     },
14     modifier = Modifier.fillMaxWidth()
15 ) {
16     Text("Kematian")
17 }
18

```

Gambar 12. Script Layanan

Button Kematian saat di klik akan memasukkan data “kematian” kedalam variable *currentCategory*, begitu juga untuk layanan yang lain, data ini akan dijadikan sebagai “key” selanjutnya akan ditampilkan form input NIK dengan script seperti pada gambar 13.

```

19 if {showNikDialog} {
20     AlertDialog(
21         onDismissRequest = { showNikDialog = false },
22         title = { Text("Masukkan NIK untuk
23             $currentCategory") },
24         text = {
25             Column {
26                 TextField(
27                     value = nikInput,
28                     onValueChange = { nikInput = it },
29                     label = { Text("NIK") },
30                     modifier = Modifier.fillMaxWidth()
31                 )
32             }
33         }
34     )
35 }
36

```

Gambar 13. Script NIK

Data NIK yang dimasukkan akan disimpan pada variabel *nikInput*, untuk selanjutnya variable tersebut di gabungkan untuk dikirimkan ke server, terlihat pada gambar 14.

```

33 confirmButton = {
34     Button(
35         onClick = {
36             showNikDialog = false
37             // Construct the message for the se
38
39             val messageToSend =
40                 "$tujuan$currentCategory:$nikInput"
41             CoroutineScope(Dispatchers.IO).launch {
42                 val result =
43                     sendMessageToServer(messageToSe
44                     withContext(Dispatchers.Main) {
45                         response = result
46                     }
47                 )
48             }
49         }
50     )
51 }
52

```

Gambar 14 Script confirmButton

Pada baris 139 menunjukkan penggabungan variabel tersebut dan dimasukkan ke variabel *messageToSend* selanjutnya pada baris 141 data tersebut dikirimkan sebagai *outputStream*.

Sehingga jika digabungkan variabel-variabel tersebut akan menjadi sebuah message private. Dimana variable *tujuan* berisi `var tujuan by remember { mutableStateOf("@:sisduk:") }` Maka jika NIK yang dikirim adalah 34040607979548, *messageToSend* akan berisi `@:sisduk:kematian: 34040607979548`

Saat Server TCP menerima message tersebut akan diproses seperti terlihat pada gambar 15.

```

String message;
while ((message = in.readLine()) != null) {
    if (message.startsWith("@:")) {
        // Format: $targetUser:$pesan
        String[] parts = message.split(":", 4);
        if (parts.length == 4) {
            String target = parts[1]; String pa = parts[2];
            PrintWriter targetOut = clients.get(target);
            if (targetOut != null) {
                targetOut.println(message);
            } else {
                out.println("User tidak ditemukan." + target);
            }
        }
    }
}

```

Gambar 15. Server Message

Baris 40 input stream akan masukkan ke variable *message* selanjutnya akan dicek apakah *message* diawali dengan key “@:”, jika yaa maka *message* tersebut akan di split berdasarkan pemisah simbol : (titik dua), maka akan didapatkan tujuan pesan pada variable *parts[1]*, data tersebut akan digunakan sebagai penunjuk client yang dihubungi melalui *PrintWriter targetout* pada baris 46 lalu baris 48 akan mengirimkan output stream *targetOut(message)*. Saat Client SISDUK menerima pesan tersebut akan diolah seperti terlihat pada gambar 15.

```

//if (serverMsg.startsWith("@:")) {
// Format: $targetUser:$pesan
String[] parts = serverMsg.split(":", 4);
if (parts[1].equals("kematian")) {
    //cari di database
String jdbcUrl = "jdbc:mysql://localhost:3306/sisduk";
String username = "sisduk";
String password = "1234567890";
try {
    Connection connection = DriverManager.getConnection(jdbcUrl, username,
        password);
    System.out.println("koneksi berhasil!");
    Integer ketemu = 0;
    String query = "SELECT * FROM odmsisduk where nik="+parts[2];
    PreparedStatement statement = connection.prepareStatement(query);
    ResultSet resultSet = statement.executeQuery();
    while (resultSet.next()) {
        out.println("NIK: " + resultSet.getString("nik")+
            "nama:" + resultSet.getString("nama"));
        System.out.println("ketemu");
        ketemu++;
    }
} catch (SQLException e) {
    out.println("Masuk NIK "+parts[2]+" Tidak ditemukan");
}
}

```

Gambar 15 Script Client Sisduk

Pada baris 28 *serverMsg* akan di check dulu apakah diawali dengan simbol @:, jika yaa maka *serverMsg* akan di split, lalu *key* akan diarahkan sesuai dengan isinya seperti pada baris 31 untuk key “kematian”, selanjutnya data akan dicari pada database SISDUK dengan filter NIP, hasil pencarian akan dikirimkan ke client yang meminta seperti pada baris 44 jika data ketemu dan baris 50 jika data tidak ditemukan, begitu juga untuk data-data yang lain. Setelah data dikirimkan maka Client SISDUK akan menerima informasi atau notifikasi seperti pada gambar 8.

Setelah semua pengujian dilakukan diperoleh hasil bahwa semua fungsionalitas dari sistem ini bisa berjalan dengan lancar sesuai dengan rancangan yang direncanakan.

IV. KESIMPULAN

Berdasarkan hasil penelitian dapat disimpulkan bahwa konsep terdistribusi bisa digunakan untuk menghubungkan sistem

informasi yang tidak berada dalam satu jaringan (saling asing) dengan bantuan sebuah server TCP yang diletakkan pada area publik, Keamanan Data akan lebih terjamin karena client tidak bisa mengakses data tersebut selain itu sistem informasi juga tidak diletakkan di area publik. Dengan konsep ini terbukti bahwa informasi dapat disebar/atau diakses oleh pihak luar sehingga nilai manfaat data dan informasi akan lebih meningkat. Beberapa saran yang belum diuji dalam penelitian ini adalah seberapa besar kemampuan server TCP dalam melayani Client APPS.

UCAPAN TERIMA KASIH

Ucapan terimakasih dan apresiasi yang sebesar-besarnya kepada pihak Universitas Akprind Indonesia yang telah memberikan fasilitas laboratorium jaringan untuk mengeksplor penelitian ini.

REFERENSI

- [1] J. Triyono, H. and R. Tefa, "Konsep Komunikasi Data Antar Sistem Informasi Menggunakan Transmisi Kontrol Protokol (TCP)," *Konsep Komunikasi Data Antar Sistem Informasi Menggunakan Transmisi Kontrol Protokol (TCP)*, vol. 9(1), pp. 34-41, 2024.
- [2] A. N. and D. P. , "Perancangan sistem informasi terdistribusi pada jaringan cabang ritel berbasis MySQL dan JSON," *Jurnal Teknologi dan Sistem Informasi*, pp. 23-30, 2021.
- [3] R. A. S. and M. L. , "Integrasi data terdistribusi menggunakan REST API dan format JSON pada sistem multi-cabang," *Jurnal Ilmu Komputer*, pp. 23-30, 2020.
- [4] A. R. H. R. N. T. M. and A. S. , "Rancang Bangun Sistem Terdistribusi pada Toko Roti Valentine Tanjung Jabung Barat," *Polygon : Jurnal Ilmu Komputer Dan Ilmu Pengetahuan Alam*, p. 157-165., 2024.
- [5] S. F. Dita, T. D. Anggraeni and E. Rilvani, "Optimalisasi Manajemen Terdistribusi pada Sistem Operasi Cloud Computing," *Merkurius : Jurnal Riset Sistem Informasi dan Teknik Informatika*, vol. 3(1), pp. 194-197, 2025.
- [6] J. Triyono, P. Nadira and C. A. Subhkan, "IMPLEMENTASI SISTEM TERDISTRIBUSI MENGGUNAKAN REPLIKASI DATABASE DAN WEB SERVICE," in *Prosiding Seminar Nasional Multidisiplin Ilmu*, Yogyakarta, 2021.
- [7] R. Pratama, "Web Service untuk transaksi data pada aplikasi sistem informasi kelembagaan petani dengan metode REST," *Jurnal Informatika Universitas Pamulang*, vol. 7(2), pp. 271-279, 2022.
- [8] J. Triyono, E. Fatkhayah, H. I. Ramadhan and N. I. Yatim Fadlan, "Model Aplikasi Terdistribusi Dengan Menerapkan Sinkronisasi Data Searah Melalui FTP Server Dengan Format CSV Studi Kasus Warung Snack KWT Kemuning," in *Prosiding Seminar Nasional Teknologi Informasi dan Bisnis*, Surakarta, 2022.
- [9] J. Triyono, P. Haryani and A. Padmanaba, "Model Kontrol Transaksi RDBMS Menggunakan Trigger dan Waktu Server," *Jurnal Teknologi*, vol. 12, no. 1, pp. 80-86, 2019.
- [10] H. Murti, E. Supriyanto, R. Redjeki, E. Lestariningsih and e. ardhianto, "Studi perkembangan dan implementasi sistem basis data terdistribusi: literature review," *JIP (Jurnal Informatika Polinema)*, vol. 10(2), pp. 249-256, 2024.
- [11] D. Comer, *Internetworking with TCP/IP Volume one*, Prentice Hall, 2000.
- [12] M. J. D. Kenneth L. Calvert, *TCP/IP Sockets in Java: Practical Guide for Programmers*, Morgan Kaufmann Publishers, 2002.
- [13] A. D. Khamis and S. Subair, "Security Framework for Distributed Database System," *Journal of Data Analysis and Information Processing*, vol. 7, pp. 1-13, 2019.
- [14] A. & W. D. Tanenbaum, *Computer Networks*, Prentice Hall, 2010.
- [15] W. Steven, *TCP/IP Illustrated, Volume 1: The Protocols*, Addison-Wesley, 1994.
- [16] E. R. Harold, *Java Network Programming Fourth Edition*, -: O'Reilly Media; Fourth Edition (November 12, 2013), 2013.
- [17] A. Hamzah, E. Susanti and S. , *MODUL PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK LANJUT*, Yogyakarta: AKPRIND PRESS, 2018.