

## LAMPIRAN

Surat Permohonan Izin Penelitian



### FAKULTAS SAINS DAN TEKNOLOGI UNIVERSITAS LABUHANBATU

PROGRAM STUDI :

AGROTEKNOLOGI - TEKNOLOGI INFORMASI - SISTEM INFORMASI - MANAJEMEN INFORMATIKA

Jl. SM. Raja No. 126-A KM. 3,5 Aek Tapa - Rantauprapat - Sumatera Utara - Pos 21415

Telp./Fax. (0624) 21901

Nomor : 05/TI/FST-ULB/2026  
Hal : Permohonan Izin Penelitian

Kepada Yth.  
**PT. KOLONI KARYA TEKNOLOGI**  
di -  
Tempat

Sehubungan dengan rencana Penelitian untuk Skripsi/Tugas Akhir Mahasiswa Program Studi S-1 Teknologi Informasi Fakultas Sains dan Teknologi tersebut dibawah ini :

Nama : RIZKY RAHMAN RAMBE  
NPM : 2208100083  
Program Studi : T-1 Teknologi Informasi  
Judul Tugas Akhir : RANCANG BANGUN WEBSITE PENGIRIM PESAN  
SEBAGAI MEDIA PROMOSI MELALUI INTEGRASI  
WHATSAPP API

Lokasi Penelitian : PT. KOLONI KARYA TEKNOLOGI  
Jl. Tengku Amir Hamzah, Sioldengan, Kec. Rantau Sel., Kab.  
Labuhanbatu, Sumatera Utara 21419

Untuk keperluan tersebut diatas, agar kiranya dapat memberi izin pelaksanaan penelitian di wilayah Bapak/Ibu. Dalam proses pelaksanaannya segala sesuatu yang berkaitan dengan penelitian tersebut akan diselesaikan oleh mahasiswa yang bersangkutan.

Demikian hal ini kami sampaikan atas perhatian dan bantuannya diucapkan terima kasih.

Rantauprapat, 22 Januari 2026  
Fakultas Sains dan Teknologi  
K. Prodi Teknologi Informasi

**Rahmadani Pane, S.Kom, M.Kom**  
NIDN. 0110058601

## Surat Balasan Izin Penelitian



Office: Koloni Square, Jl. Tengku Amir Hamzah, Sioldengan, Kec. Rantau Selatan, Labuhanbatu.  
Rantau Prapat, Labuhanbatu, 21412

Nomor : Rantau Prapat, 22 Januari 2026  
Lampiran : -  
Hal : Balasan Surat Izin Penelitian

Kepada Yth. DEKAN FAKULTAS SAINS DAN TEKNOLOGI  
Program Studi T-1 Teknologi Informasi  
Di Rantau Prapat

Assalamualaikum Wr.Wb  
Dengan Hormat,

Sesuai Surat dari Kaprodi T-1 Teknologi Informasi Fakultas Sains dan Teknologi Universitas  
Labuhanbatu Nomor 05/TI/FST-ULB/1/2026 Tanggal 22 Januari 2026 tentang permohonan izin  
penelitian mahasiswa yang bernama :

Nama : RIZKY RAHMAN RAMBE  
NPM : 2208100083  
Program Studi : T-1 Teknologi Informasi  
Judul Skripsi : Rancang Bangun Website Pengirim Pesan Sebagai Media  
Promosi Melalui Integrasi WhatsApp API

Dengan Surat ini kami dari PT. KOLONI KARYA TEKNOLOGI memberikan izin penelitian  
bersedia memberikan data data yang diperlukan oleh Mahasiswa tersebut.

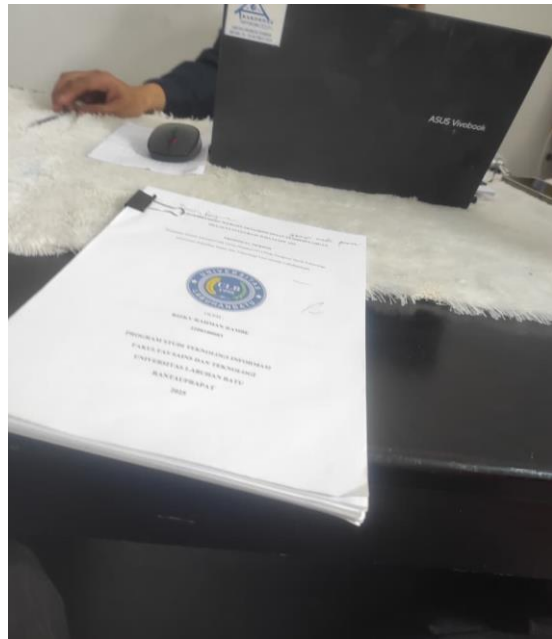
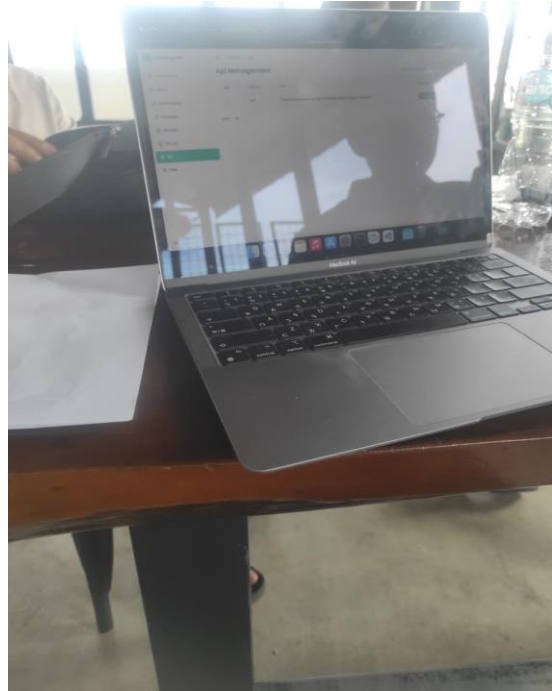
Demikian hal ini kami sampaikan, terima kasih

Pemilik PT. Koloni Karya Teknologi



Doni Pmayungan, S. T., M.T

## Dokumentasi Penelitian



## Code Program WhatsApp API

```
1 import makeWASocket, { useMultiFileAuthState, fetchLatestBaileysVersion, makeCacheableSignalKeyStore, DisconnectReason, WASocket } from "@whiskeysockets/baileys";
2 import { Boom } from "@hapi/boom";
3 import P from "pino";
4 import qrcode from "qrcode-terminal";
5 import express, { Request, Response } from "express";
6 import { handleCommand } from "../src/config/handler";
7
8 let sock: WASocket | null = null;
9
10 async function startSock(): Promise<void> {
11   const { state, saveCreds } = await useMultiFileAuthState("./auth_session");
12   const { version } = await fetchLatestBaileysVersion();
13
14   const socket = makeWASocket({
15     version,
16     logger: P({ level: "silent" }),
17     auth: {
18       creds: state.creds,
19       keys: makeCacheableSignalKeyStore(state.keys, P({ level: "silent" })),
20     },
21   });
22
23   socket.ev.on("connection.update", (update) => {
24     const { qr, connection, lastDisconnect } = update;
25
26     if (qr) qrcode.generate(qr, { small: true });
27
28     if (connection === "close") {
29       const shouldReconnect =
30         (lastDisconnect?.error as Boom)?.output?.statusCode !== DisconnectReason.loggedOut;
31
32       console.log("Disconnected:", lastDisconnect?.error);
33       if (shouldReconnect) startSock();
34     }
35
36     if (connection === "open") console.log(200);
37   });
38
39   socket.ev.on("creds.update", saveCreds);
40
41   socket.ev.on("messages.upsert", async ({ messages }) => {
42     const msg = messages[0];
43     if (!msg.message) return;
44     await handleCommand(msg, socket);
45   });
46
47   sock = socket;
48 }
49
50 const app = express();
51 app.use(express.json());
52
53 const SECRET_KEY = "Rahman1231";
54
55 interface MediaPayload {
56   url: string;
57   mimetype: string;
58   filename: string;
59 }
60
61 interface SendRequest {
62   phone: string;
63   message: string;
64   token: string;
65   media?: MediaPayload;
66 }
67
68 app.post("/send", async (req: Request<{}, {}, SendRequest>, res: Response) => {
69   const { phone, message, token, media } = req.body;
70
71   if (!token || token !== SECRET_KEY) {
72     return res.status(403).json({ success: false, message: "Token tidak valid" });
73   }
74
75   if (!sock) {
76     return res.status(400).json({ success: false, message: "Bot belum terhubung ke WhatsApp" });
77   }
78
79   if (!phone) {
80     return res.status(400).json({ success: false, message: "Nomor wajib diisi" });
81   }
82
83   if (!message && !media?.url) {
84     return res.status(400).json({ success: false, message: "Pesan atau media wajib diisi" });
85   }
86
87   try {
88     const jid = `${phone.replace(/\\/D/g, "")}@s.whatsapp.net`;
89
90     if (media?.url) {
91       await sock.sendMessage(jid, { text: message || "" });
92       return res.json({ success: true, message: "Pesan teks terkirim ke ${phone}" });
93     }
94
95     const mime = (media.mimetype || "").toLowerCase();
96     const caption = message || "";
97
98     if (mime.startsWith("image/")) {
99       await sock.sendMessage(jid, {
100         image: { url: media.url },
101         caption,
102       });
103       return res.json({ success: true, message: "Gambar terkirim ke ${phone}" });
104     }
105
106     if (mime.startsWith("video/")) {
107       await sock.sendMessage(jid, {
108         video: { url: media.url },
109         caption,
110       });
111       return res.json({ success: true, message: "Video terkirim ke ${phone}" });
112     }
113
114     if (mime.startsWith("audio/")) {
115       await sock.sendMessage(jid, {
116         audio: { url: media.url },
117         mimetype: media.mimetype || "audio/mpeg",
118         ptt: false,
119       });
120       return res.json({ success: true, message: "Audio terkirim ke ${phone}" });
121     }
122
123     await sock.sendMessage(jid, {
124       document: { url: media.url },
125       mimetype: media.mimetype || "application/octet-stream",
126       fileName: media.filename || "file",
127       caption: caption || undefined,
128     });
129     return res.json({ success: true, message: "Dokumen terkirim ke ${phone}" });
130   } catch (err) {
131     console.error("Gagal mengirim pesan:", err);
132     return res.status(500).json({ success: false, message: "Gagal mengirim pesan." });
133   }
134 }
135
136 const PORT = 5000;
137 app.listen(PORT, async () => {
138   await startSock();
139 });
140
141 setInterval(() => {}, 1000);
```