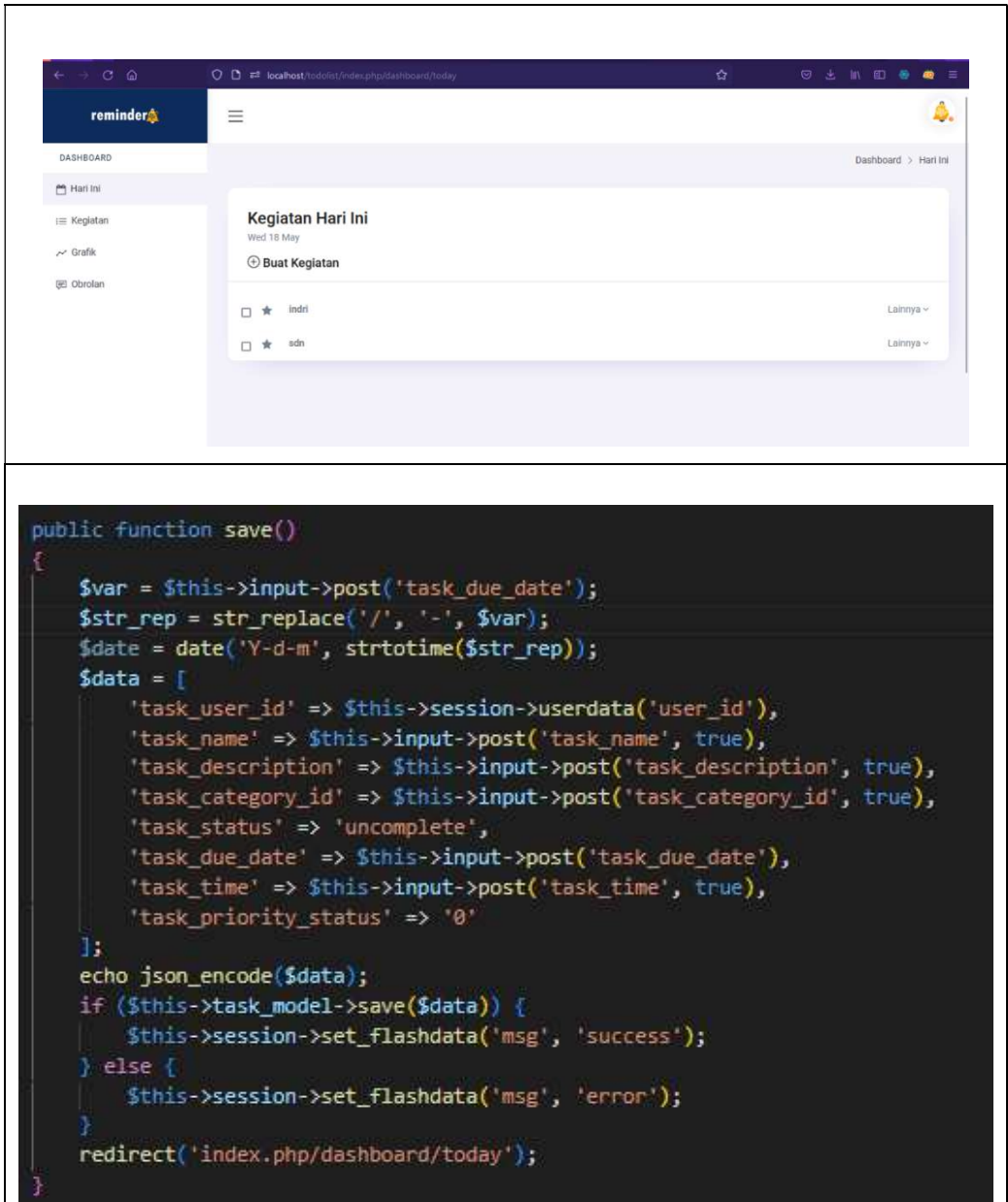


LAMPIRAN

Implementasi Proses Aplikasi

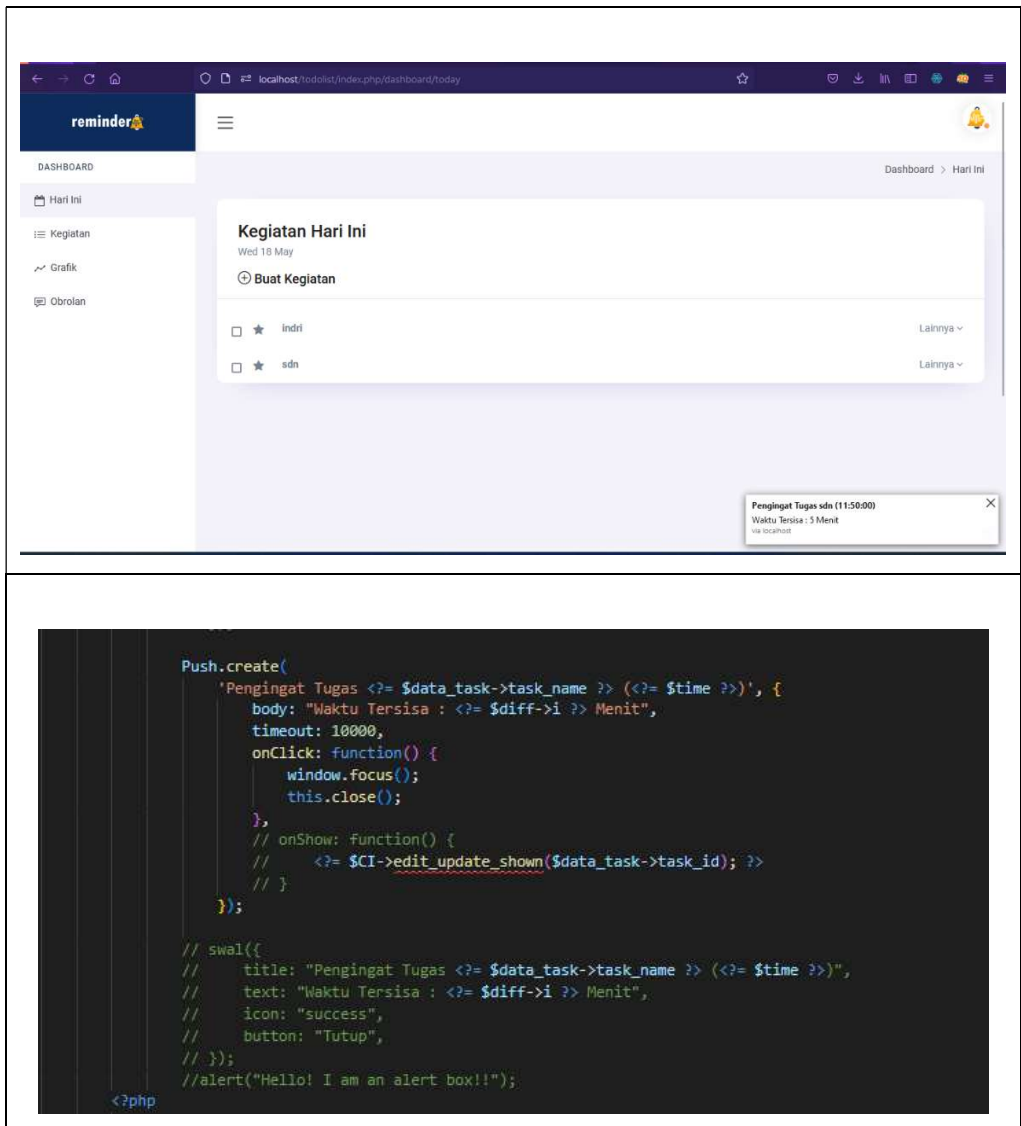
➤ *Reminder Kegiatan*

a. Implementasi Input *Reminder Kegiatan*



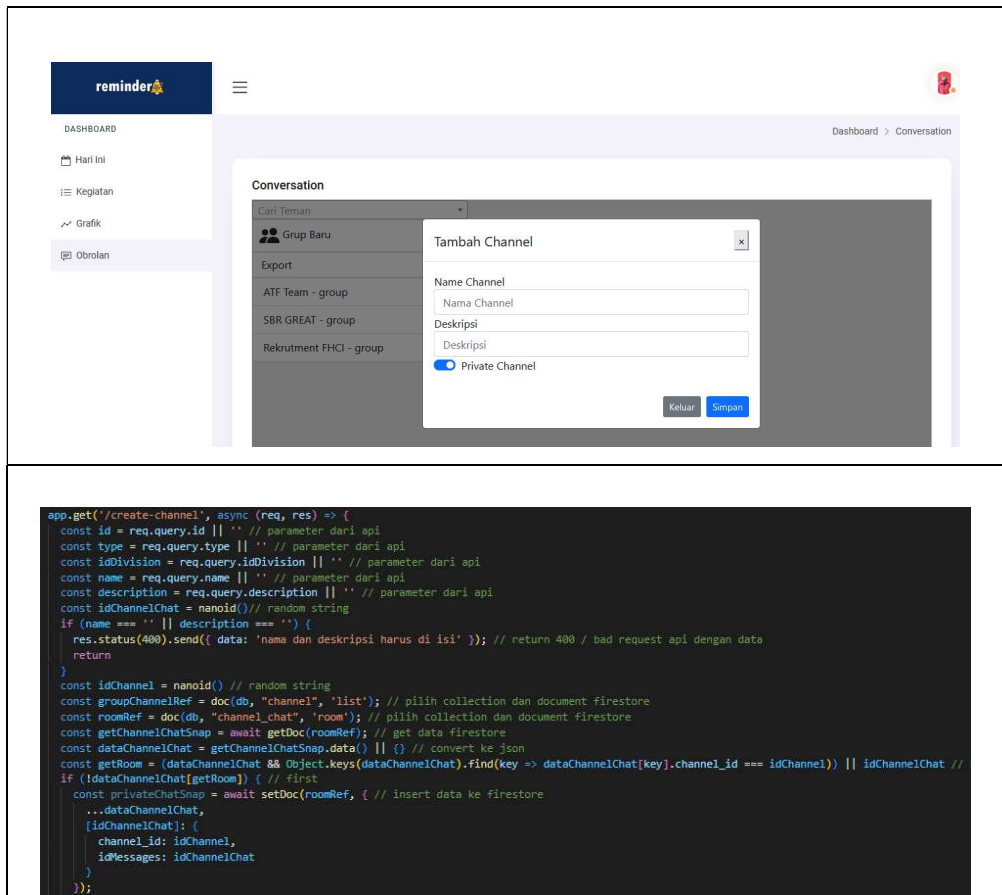
Gambar 4. 26 Implementasi Input *Reminder Kegiatan*

b. Implementasi Sistem Kirim Notifikasi *Reminder*



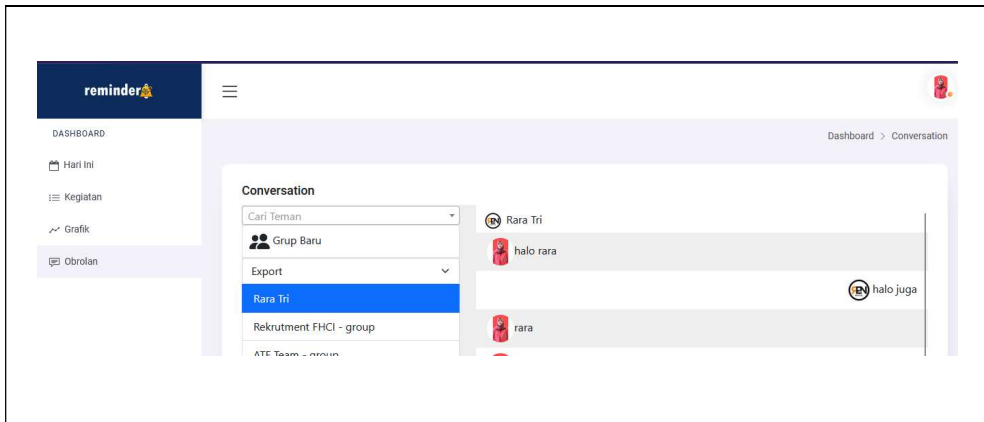
Gambar 4. 27 Implementasi Sistem Kirim Notifikasi *Reminder*

➤ **Implementasi Komunikasi Asinkron**
c. Implementasi Pembuatan Channel atau Grup



Gambar 4. 28 Implementasi Pembuatan Channel atau Grup

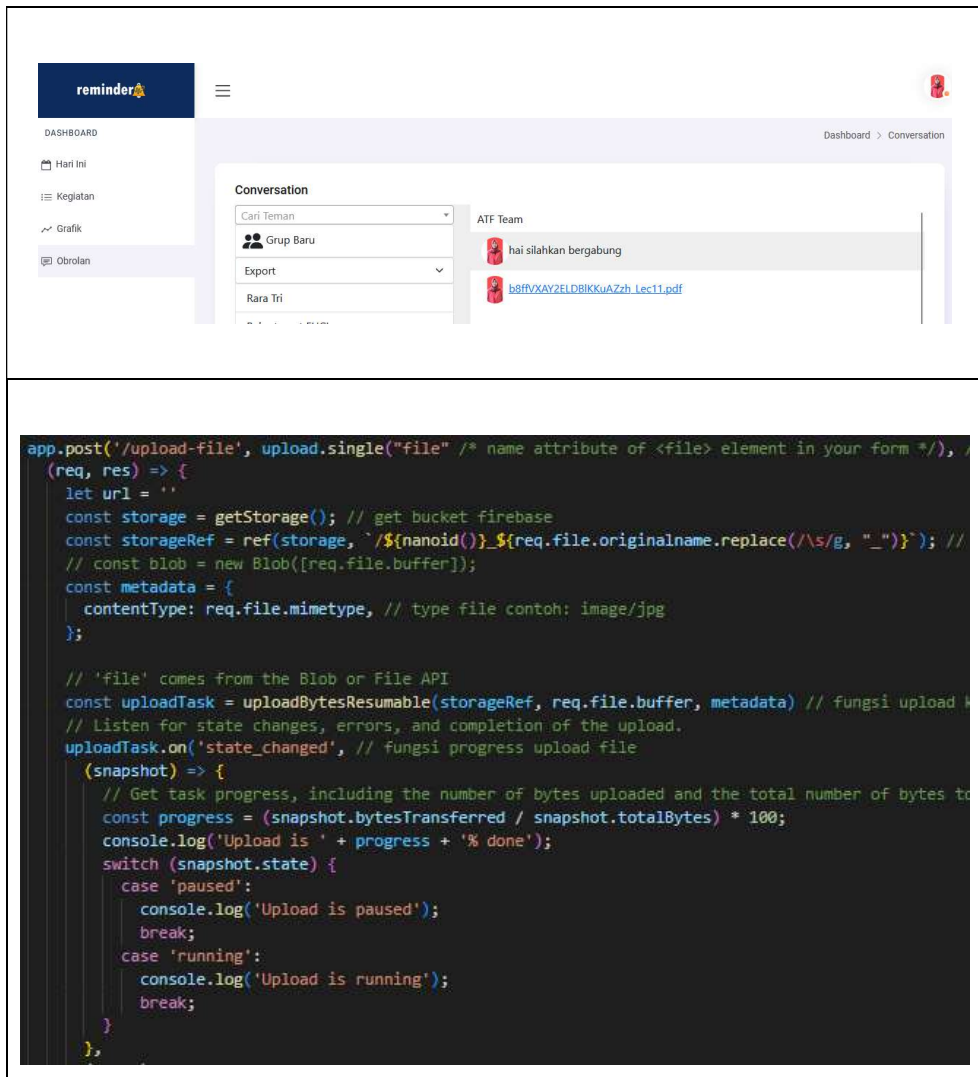
d. Implementasi Kirim & Terima Chatt



```
.then(response => response.json())
.then(async res => {
  // console.log(res)
  const { data } = res
  const idSelected = $('#tab-message').find('input#idSelected')[0].value
  const getEl = $('#tab-message').find('input#idSelected')[0]
  const getType = $(getEl).attr('dataType')
  let allData = {}
  let result = {}
  let listDetailUserRes = {}
  if (res && res.data) {
    const msg = res.data
    if (getType === 'channel') {
      const res = await pushMessageChannel(idSelected, '<%= id %>', msg) //
      const { data, listDetailUser } = res
      allData = res
      result = data
      listDetailUserRes = listDetailUser
    }
    else {
      const res = await pushMessage(idSelected, '<%= id %>', msg) // bikin p
      const { data } = res
      allData = res
      result = data
    }
  }
  socket.emit('chat message', getType !== 'channel' ? { // eksekusi socket
    message: msg,
    room: result.room,
    idRoom: Object.keys(result.room)[0],
    to: idSelected,
    from: '<%= id %>',
    for: 'receiver',
    isNewMessage: result.is_new_message,
    type: '',
    allData
  }
})
```

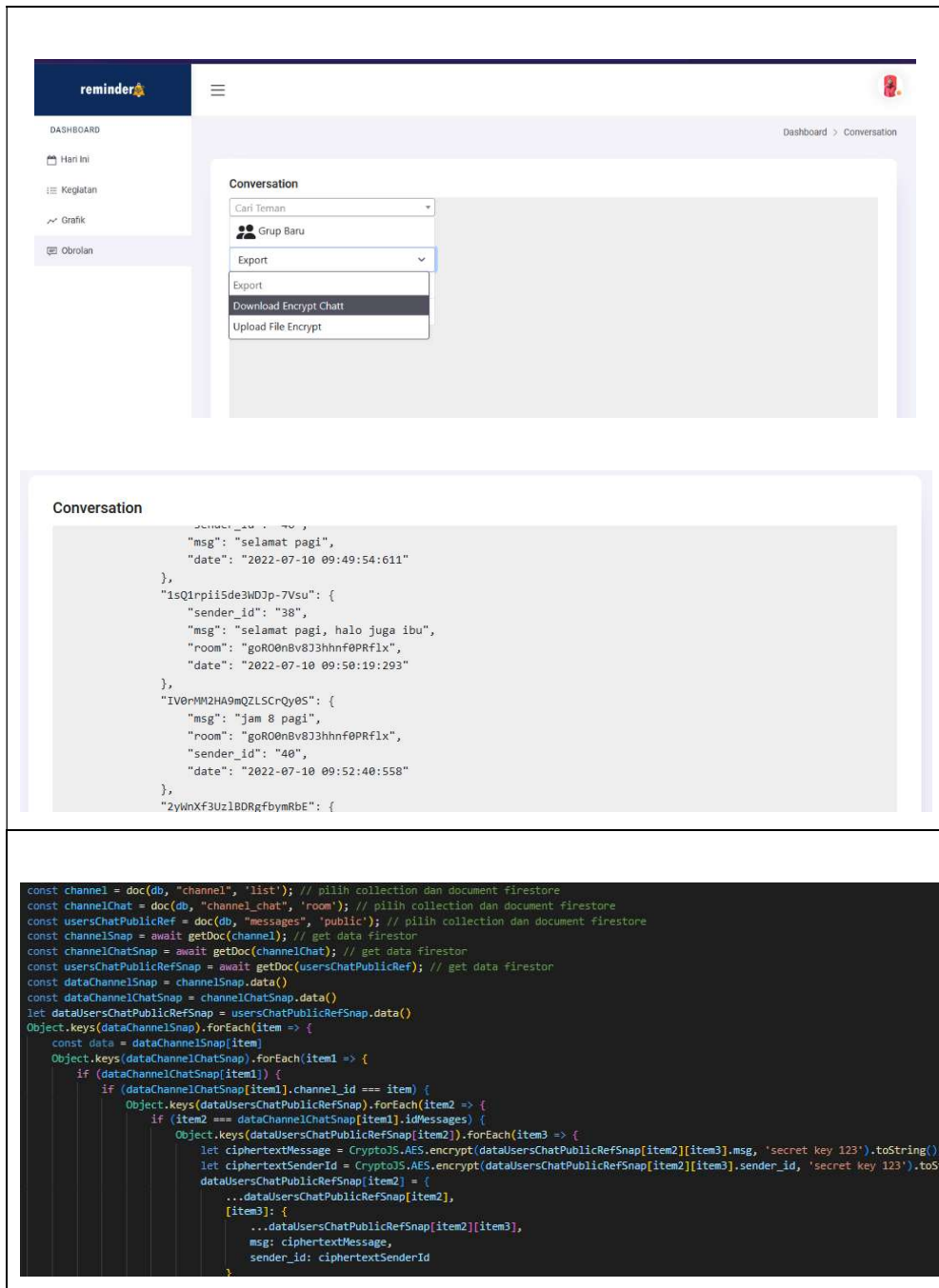
Gambar 4. 29 Implementasi Kirim & Terima Chatt

e. Implementasi Kirm File



Gambar 4. 30 Implementasi Kirm File

f. Implementasi Enkripsi & Dekripsi atau Keamanan Chatting



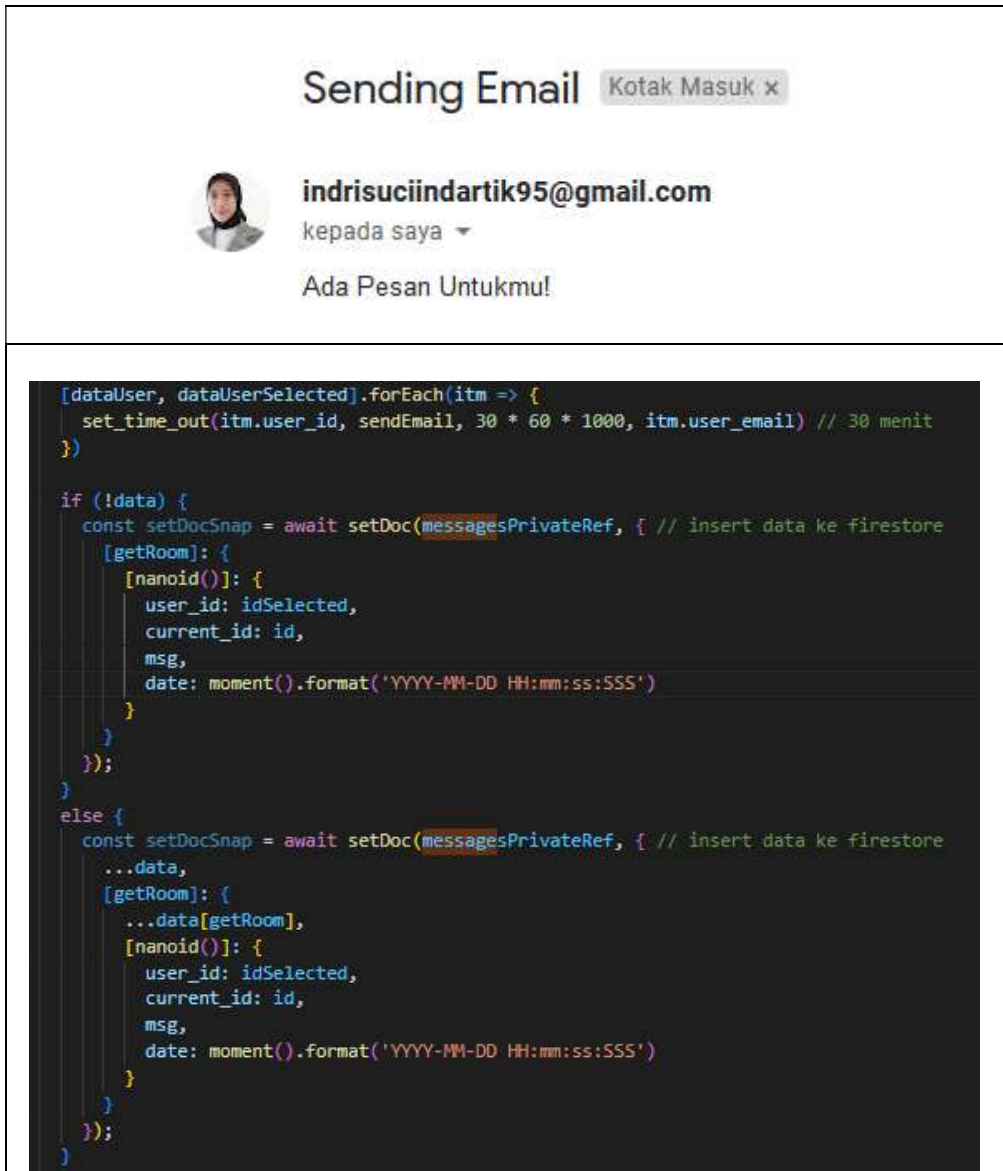
The image displays a web application interface for chat encryption. The top part shows a sidebar with navigation options like 'Dashboard', 'Hari Ini', 'Kegiatan', 'Grafik', and 'Obrolan'. The main content area is titled 'Conversation' and features a search bar, a 'Grup Baru' button, and a dropdown menu with options: 'Export', 'Download Encrypt Chats', and 'Upload File Encrypt'. Below this, a sample chat message is shown with its metadata.

```
Conversation
{
  "sender_id": "38",
  "msg": "selamat pagi",
  "date": "2022-07-10 09:49:54:611"
},
{
  "1sQ1rpii5de3MD3p-7Vsu": {
    "sender_id": "38",
    "msg": "selamat pagi, halo juga ibu",
    "room": "goRO0nBv8J3hhnf0PRflx",
    "date": "2022-07-10 09:50:19:293"
  },
  "IV0rMM2HA9mQZL5CrQy05": {
    "msg": "jam 8 pagi",
    "room": "goRO0nBv8J3hhnf0PRflx",
    "sender_id": "40",
    "date": "2022-07-10 09:52:40:558"
  },
  "2yWnXF3Uz1BDRGfbyMrbE": {
```

```
const channel = doc(db, "channel", 'list'); // pilih collection dan document firestore
const channelChat = doc(db, "channel_chat", 'room'); // pilih collection dan document firestore
const usersChatPublicRef = doc(db, "messages", 'public'); // pilih collection dan document firestore
const channelSnap = await getDoc(channel); // get data firestore
const channelChatSnap = await getDoc(channelChat); // get data firestore
const usersChatPublicRefSnap = await getDoc(usersChatPublicRef); // get data firestore
const dataChannelSnap = channelSnap.data()
const dataChannelChatSnap = channelChatSnap.data()
let dataUsersChatPublicRefSnap = usersChatPublicRefSnap.data()
Object.keys(dataChannelSnap).forEach(item => {
  const data = dataChannelSnap[item]
  Object.keys(dataChannelChatSnap).forEach(item1 => {
    if (dataChannelChatSnap[item1].channel_id === item) {
      if (dataChannelChatSnap[item1].channel_id === item) {
        Object.keys(dataUsersChatPublicRefSnap).forEach(item2 => {
          if (item2 === dataChannelChatSnap[item1].idMessages) {
            Object.keys(dataUsersChatPublicRefSnap[item2]).forEach(item3 => {
              let ciphertextMessage = CryptoJS.AES.encrypt(dataUsersChatPublicRefSnap[item2][item3].msg, 'secret key 123').toString();
              let ciphertextSenderId = CryptoJS.AES.encrypt(dataUsersChatPublicRefSnap[item2][item3].sender_id, 'secret key 123').toString();
              dataUsersChatPublicRefSnap[item2] = {
                ...dataUsersChatPublicRefSnap[item2],
                [item3]: {
                  ...dataUsersChatPublicRefSnap[item2][item3],
                  msg: ciphertextMessage,
                  sender_id: ciphertextSenderId
                }
              }
            }
          }
        }
      }
    }
  }
}
```

Gambar 4. 31 Implementasi Enkripsi atau Keamanan Chatting

g. Implementasi Sistem Kirim Notifikasi Email



Gambar 4. 32 Implementasi Sistem Kirim Notifikasi Email