

# 15,000 emails. 8 months. All delivered to you from the only one roothouse.

reinhardt1010.id – 31 January 2023

From

<https://reinhardt1010.id/blog/2023/01/31/15000-emails-8-months-all-delivered-to-you-from-the-only-one-roothouse>.

Scan the QR Code to view the article on your device or web browser.



---

Content may subject to copyright. Visit the original website to view copyright and licensing information about this content. QR Code is a registered trademark of DENSO WAVE, Inc. in Japan and other countries. Generated on 2024-09-17 23:22:37.

---

It was June 2021 and we were trying to send emails for those joining **HISHOT 2021**, an online community seminar held by the **Computer Science Student Association of BINUS University** (HINTI BINUS).

If we decide to use Gmail, well, one day Google could ban us if we send more and more automated emails in the long-term. Good thing that the web hosting service behind himti.or.id also offers a SMTP email service, which we can use instead to send to them on behalf of **noreply@himti.or.id**. The web hosting provider rate-limits the inbuilt SMTP service to the maximum of **200 emails per hour**. But just to be safe, we're just going to use a half of them.

That means we have to schedule the emails to be sent at a constant rate. Sure, we can use a MySQL database to store the pending emails, but Reinhart doesn't want to run and cron PHP scripts inside the remote server. At the end of the day, he decided to write a Python script and run them on his own PC.

But one question remains. Is there a better way than just letting that big laptop to run scripts 24/7? Do we have a more energy-efficient way to do this?

Apparently before Reinhart got his first Raspberry Pi 3 as a birthday gift and later bought an Orange Pi Zero, and they're both unused. So we decided to try the smaller house, and oh boy, we made it **root**.



Hey, you may already read about [Shift's log about sending emails](#) before. But sure, everyone's working to get the job done.

Here we fetch all draft messages to the roothouse before asking her to deliver one-by-one. And spoiler alert, **the roothouse is getting bigger!**

## Version 0: A greedy Python script.

We first iterated on creating a simple Python script which does these things:

1. Get a list of draft emails, straight from a remote MySQL database
2. For each emails, send them via SMTP

Simple, right? The original script really worked well. However, another problem came when the script needs to fetch all of the pending emails. Or in other words, if there are 1,000 email messages waiting to be sent, **the script horribly fetches ALL of them before being able to be sent.**

Well, that's too much and we should fetch them little by little. Not to mention there could be some messages which we need to prioritize for and to let Shift recover again when the script crashes.

## Version 1: A slightly efficient script.

Then we decided to send emails by **priority levels** by upgrading the database and our SQL query. And as expected, this version worked even better especially when queuing low-priority emails (e.g. attendance receipts and e-certificates). We also tried to fetch **at most 50 pending emails** per batch, before sending and re-checking them, so the overall delivery time can be significantly improved.

Additionally, we started to **accept and parse email messages written in Markdown** so yeah, we can finally make your next emails from HIMTI less boring. [HIMTI's Unified Registration](#) system also used Markdown for creating event announcements, and we're very happy to integrate that into our emails.

This version worked well until we had to think about delivering **e-certificates** for **HISHOT 2021**. No, not by sending a OneDrive or Google Drive link to the list of e-certificates, but attaching them **directly** into the message. So here comes Version 2.

## Version 2: Because e-certificates broke our script.

Sending e-certificates are much slower than regular, plaintext/HTML/Markdown emails for one reason: they're huge.

There are two main strategies for this, whether the script is instructed to fetch the file locally and add them to the email list, or by including the entire attached message into the database. We ended up choosing the latter one because that means we can also send anything beyond images and e-certificates.

So as usual, we upgraded the database and script, but then **the hosting provider blocked our websites for storing too much data inside the database**. That's quite a bummer, though, that we have to regularly clear sent emails from the database to be able to schedule new emails for more people.

## Version 3: Concurrency.

As this great email delivery solution for HISHOT was adopted to more HIMTI events, including TECHNO 2021, COMPUTERUN 2.0, and HIMTI ELECTION 2021, we realized that we have to upgrade our infrastructure to be able to send more emails, for more events and people!

We also realized that not every message have to be sent over the himti.or.id SMTP server. For example, internal message for event committees when a new participant have registered and paid for the event.

That's why we decided to use **Shiftine's private email address** to send these committee messages. And more than that, we start to use **GitHub Actions** to help us check emails more regularly, being able to **sleep()** when there are no more emails to send. We can also prioritize and strategize the

delivery of different emails from separate events, like sending HIMTI ELECTION receipts during the day and COMPUTERUN 2.0 reminders during the night.

## **Some personal reflections.**

Yeah, that was great! I was able to share my "can't hibernate, i'm overpowered" spirit to help people in the real world. And as I now help to maintain [BINUS Today](#)'s list of articles as part of my chore, seeing those sent emails feels like a nostalgia for me.

Oh, right, you can check out our final code at <https://github.com/alterine0101/emailer.py/>.