BAB 1

PENDAHULUAN

1.1 Latar Belakang

Pengumuman di jurusan Teknik Informatika Unpar pada umumnya dilakukan oleh email. Pengumuman lewat email ini praktis karena tidak perlu menunggu sampai email sampai ke tujuan dan dijamin sampai ke tujuan. Selain itu, konten yang disampaikan melalui email fleksibel. Konten tidak harus hanya tulisan tapi dapat menambahkan lampiran, mengubah gaya tulisan dan lain-lain.

Namun, email kurang terorganisir dengan baik. Email yang masuk sering tercampur dengan email lain sehingga mahasiswa kesulitan mencari email yang penting. Dampaknya, pengumuman-pengumuman penting sering tidak terbaca secara tidak sengaja.

Pada skripsi ini, akan dibuat solusi masalah tadi dengan membuat suatu sistem. Sistem ini akan menangkap email-email pengumuman yang masuk ke sebuah email khusus untuk menangkap pengumuman. Pertama email yang masuk ke email khusus akan diperiksa pengirimnya. Apabila pengirim adalah email yang terdaftar sebagai email yang berhak melakukan pengumuman maka email tersebut adalah email pengumuman. Setelah itu email tersebut akan dibuatkan permanent link dan disisipkan pada basis data. Lalu, mahasiswa akan menerima permanent link tersebut melalui notifikasi dari akun Line@. Line@ adalah layanan dari Line Corporation yang memudahkan pemilik bisnis atau organisasi menyampaikan pesan kepada pengikutnya di aplikasi pengirim pesan Line.

Sistem ini akan dibangun sebagai fitur tambahan pada BlueTape, sebuah website milik jurusan teknik Informatika Unpar. Pembangunan fitur ini membutuhkan modifikasi BlueTape sehingga dapat dijalankan di Heroku dan menggunakan basis data PostgreSQL. Heroku adalah layanan yang memungkinkan pengembang membangun, menjalankan, dan mengoperasikan aplikasi di dalam internet. Selain itu, sistem ini membutuhkan beberapa fitur dari layanan surel GMail dan layanan pengirim pesan instan Line@.

1.2 Rumusan Masalah

- Bagaimana cara mengubah BlueTape agar dapat mendukung fitur kolektor pengumuman dengan bantuan Heroku dan PostgreSQL ?
- Bagaimana cara mengimplementasikan kolektor pengumuman pada BlueTape ?

1.3 Tujuan

- Melakukan perawatan pada BlueTape agar dapat mendukung fitur kolektor pengumuman dengan bantuan Heroku dan PostgreSQL
- Mengimplementasikan fitur kolektor pengumuman pada BlueTape

1.4 Batasan Masalah

Pada skripsi ini masalah dibatasi dengan batasan-batasan sebagai berikut :

- Sistem ini dibuat hanya untuk jurusan teknik informatika Unpar, mahasiswa dari jurusan lain tidak dapat menggunakan sistem ini

1.5 Metodologi

Metode penelitian pada skripsi ini sebagai berikut :

1. Melakukan studi literatur tentang GMail, Line, Heroku dan PostgreSQL
2. Memodifikasi BlueTape sehingga dapat menangkap email yang masuk ke email khusus
3. Memodifikasi BlueTape sehingga dapat melakukan push notification ke akun Line@
4. Memodifikasi BlueTape sehingga dapat berjalan di Heroku menggunakan PostgreSQL
5. Melakukan pengujian
6. Menulis dokumen skripsi

1.6 Sistematika Pembahasan

1. Bab 1 : Pendahuluan Bab ini membahas gambaran umum dari skripsi.
2. Bab 2 : Dasar Teori Bab ini membahas dasar teori yang mendukung pembuatan skripsi ini.
3. Bab 3 : Analisis Bab ini membahas analisis yang dilakukan terhadap masalah yang diusung dalam skripsi ini.
4. Bab 4 : Perancangan Bab ini membahas perancangan sistem yang dibangun pada skripsi ini.
5. Bab 5 : Implementasi dan Pengujian Bab ini membahas hasil implementasi yang dilakukan beserta pengujian sistem.
6. Bab 6 : Kesimpulan dan saran Bab ini berisi kesimpulan yang didapat dari penelitian dan saran oleh penulis kepada pembaca yang hendak melanjutkan penelitian ini.

BAB 2

LANDASAN TEORI

Pada bab ini dijelaskan dasar-dasar teori mengenai *BlueTape*, *Heroku*, *PostgreSQL*, *GMail*, dan *Line@*.

2.1 *BlueTape*

BlueTape adalah perangkat lunak yang berfungsi untuk membantu urusan-urusan *paper-based* di FTIS UNPAR menjadi *paperless*. Perangkat lunak ini berbasis web dengan memanfaatkan *CodeIgniter* dan *ZURB Foundation*. Saat ini aplikasi ini memiliki dua layanan, yaitu Transkrip *Request / Manage* dan Perubahan Kuliah *Request / Manage*. Layanan Transkrip *Request / Manage* memberikan layanan untuk melakukan permohonan serta pencetakan transkrip mahasiswa. Sedangkan layanan Perubahan Kuliah *Request / Manage* memberikan layanan untuk permohonan dan pencetakan perubahan jadwal kuliah oleh dosen. [1]

2.2 *Heroku*

Heroku adalah *platform cloud* yang memungkinkan *developer* untuk membangun, menjalankan, dan mengoperasikan aplikasinya pada *cloud*. Bahasa yang didukung oleh *Heroku* adalah *Ruby*, *Node.js*, *Java*, *Python*, *Clojure*, *Scala*, *Go* dan *PHP*. [2]

2.2.1 Membangun Perangkat Lunak berbasis *PHP* di *Heroku*

Sebelum *developer* dapat membangun perangkat lunak berbasis *PHP* mereka di *Heroku*, *developer* harus membuat akun *Heroku* terlebih dahulu. Selain itu, *developer* perlu memasang *PHP* dan *Composer* di komputer yang digunakan untuk membangun perangkat lunak tersebut.[3]

Mempersiapkan *Heroku*

Sebelum *developer* dapat memulai pembangunan perangkat lunak, *developer* harus sudah memasang dan mempersiapkan *Git*. Setelah itu, *developer* perlu memasang *Heroku Command Line Interface* (CLI). *Heroku* CLI digunakan untuk mengelola perangkat lunak, *add-ons*, melihat *log* perangkat lunak, dan menjalankan aplikasi secara lokal.[3]

Setelah *Heroku* terpasang, *developer* dapat menggunakan perintah *Heroku* di *command shell*. Perintah pertama yang harus dilakukan adalah *"heroku login"*. Perintah ini berfungsi untuk otentikasi agar dapat masuk ke akun *Heroku*. Otentikasi dibutuhkan agar *Heroku* dan *Git* dapat beroperasi. [3]

[1] https://github.com/ftisunpar/BlueTape
[2] https://www.heroku.com/what
[3] https://devcenter.heroku.com/articles/getting-started-with-php

Menyiapkan Perangkat Lunak

Hal pertama yang harus dilakukan oleh *developer* adalah *clone repository* ke komputer lokal. Caranya dengan mengetikkan perintah *"git clone"* diikuti dengan alamat *git heroku*-nya di *command shell* atau *terminal*. Di dalam *repository* lokal sudah ada perangkat lunak sederhana dan juga *file composer.json*. *Heroku* menggunakan *Composer* untuk mengelola *dependency* dalam proyek *PHP*, dan *file composer.json* menjadi rujukan untuk Heroku bahwa perangkat lunak tersebut ditulis dalam bahasa *PHP*. [3]

Deploy Perangkat Lunak

Bagian ini menjelaskan cara *deploy* perangkat lunak ke *Heroku*. Pertama, *developer* perlu membuat perangkat lunak pada *Heroku*. Caranya dengan mengetikkan perintah *"heroku create"* pada *command shell* atau *terminal*. Ketika perangkat lunak dibuat, sebuah *git remote* yang otomatis dinamai *heroku* juga terbentuk. *Heroku* akan menamai perangkat lunak dengan nama acak yang nantinya bisa diganti. Untuk memulai deploy *developer* perlu mengetikkan perintah *"git push"* diikuti dengan nama *git remote* lalu nama *branch repository*. Contohnya : *"git push heroku master"* [3]

Procfile

Procfile adalah *file text* yang berfungsi untuk mendeklarasikan secara ekplisit perintah yang harus diekseskusi untuk menjalankan perangkat lunak di Heroku. Contoh isi Procfile : *"web: vendor/bin/heroku-php-apache2 web/"*. Procfile dapat berisi *single process type*, *web*, dan perintah yang diperlukan untuk menjalankan perangkat lunak tersebut. [3]

Basis Data

Heroku menyediakan tiga layanan data : Heroku Postgres, Heroku Redis, dan Apache Kafka on Heroku. Dalam skripsi ini, layanan data yang dipakai adalah Heroku Postgres yang berbasis PostgreSQL.[4]

2.3 PostgreSQL

PostgreSQL adalah sistem basis data object-relational open source. Tipe data yang didukung oleh PostgreSQL adalah :[5]

- Primitives: Integer, Numeric, String, Boolean
- Structured: Date/Time, Array, Range, UUID
- Document: JSON/JSONB, XML, Key-value (Hstore)
- Geometry: Point, Line, Circle, Polygon
- Customizations: Composite, Custom Types

2.4 GMail

GMail adalah layanan surat elektronik milik Google. Salah satu fitur GMail untuk developer adalah GMail API.

[4]https://devcenter.heroku.com/categories/data-management
[5]https://www.postgresql.org/about/

2.4.1 Cloud Pub/Sub

Langkah-langkah yang diperlukan :[6]

- Mempersiapkan GCP Console project.
- Membuat project baru atau memilih project yang sudah ada
- Mengaktifkan Cloud Pub/Sub API untuk project tersebut.
- Memasang dan menginisialisasi Cloud SDK
- Membuat Topic
- Menambahkan subcription

[7]

2.5 Line@

Line@ adalah layanan oleh LINE yang didesain khusus untuk bisnis.

•

[6]https://cloud.google.com/pubsub/docs/quickstart-console
[7]https://cloud.google.com/pubsub/docs/quickstart-console

LAMPIRAN A

KODE PROGRAM

Listing A.1: MyCode.c

```c

// This does not make algorithmic sense,
// but it shows off significant programming characters.

#include<stdio.h>

void myFunction( int input, float* output ) {
    switch ( array[i] ) {
        case 1: // This is silly code
            if ( a >= 0 || b <= 3 && c != x )
                *output += 0.005 + 20050;
            char = 'g';
            b = 2^n + ~right_size - leftSize * MAX_SIZE;
            c = (--aaa + &daa) / (bbb++ - ccc % 2 );
            strcpy(a,"hello $@?");
    }
    count = ~mask | 0x00FF00AA;
}

// Fonts for Displaying Program Code in LATEX
// Adrian P. Robson, nepsweb.co.uk
// 8 October 2012
// http://nepsweb.co.uk/docs/progfonts.pdf
```

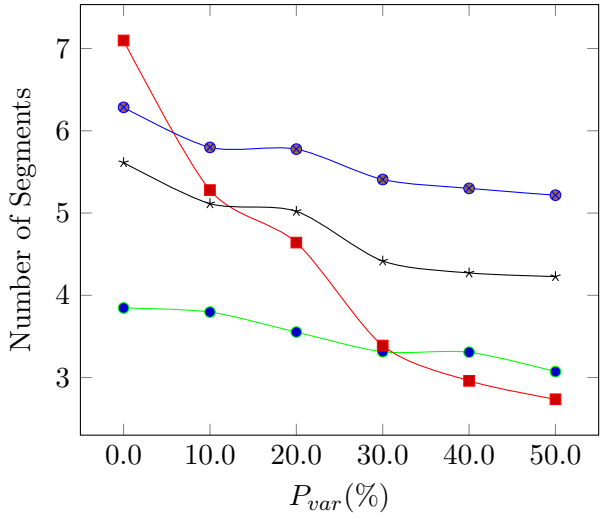
Listing A.2: MyCode.java

```java
import java.util.ArrayList;
import java.util.Collections;
import java.util.HashSet;

//class for set of vertices close to furthest edge
public class MyFurSet {
    protected int id;                                   //id of the set
    protected MyEdge FurthestEdge;                      //the furthest edge
    protected HashSet<MyVertex> set;                    //set of vertices close to furthest edge
    protected ArrayList<ArrayList<Integer>> ordered;    //list of all vertices in the set for each trajectory
    protected ArrayList<Integer> closeID;               //store the ID of all vertices
    protected ArrayList<Double> closeDist;              //store the distance of all vertices
    protected int totaltrj;                             //total trajectories in the set

    /*
     * Constructor
     * @param id : id of the set
     * @param totaltrj : total number of trajectories in the set
     * @param FurthestEdge : the furthest edge
     */
    public MyFurSet(int id,int totaltrj,MyEdge FurthestEdge) {
        this.id = id;
        this.totaltrj = totaltrj;
        this.FurthestEdge = FurthestEdge;
        set = new HashSet<MyVertex>();
        ordered = new ArrayList<ArrayList<Integer>>();
        for (int i=0;i<totaltrj;i++) ordered.add(new ArrayList<Integer>());
        closeID = new ArrayList<Integer>(totaltrj);
        closeDist = new ArrayList<Double>(totaltrj);
        for (int i = 0;i <totaltrj;i++) {
            closeID.add(-1);
            closeDist.add(Double.MAX_VALUE);
        }
    }

}
```

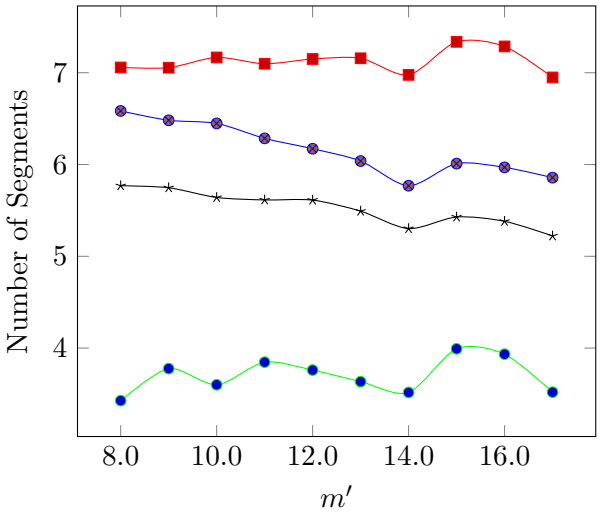
LAMPIRAN B

HASIL EKSPERIMEN

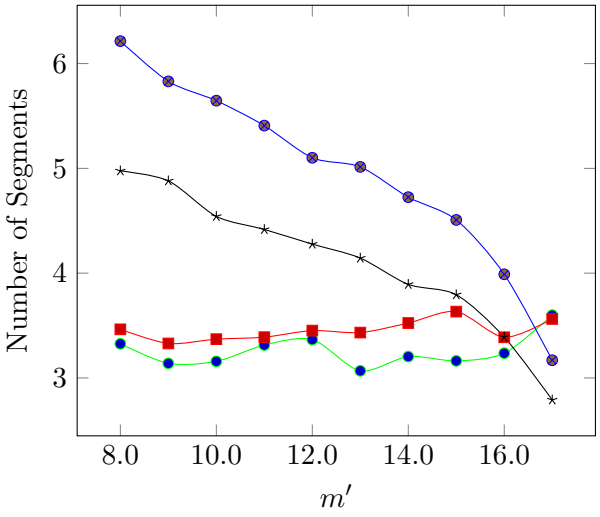
Hasil eksperimen berikut dibuat dengan menggunakan TIKZPICTURE (bukan hasil excel yg diubah ke file bitmap). Sangat berguna jika ingin menampilkan tabel (yang kuantitasnya sangat banyak) yang datanya dihasilkan dari program komputer.



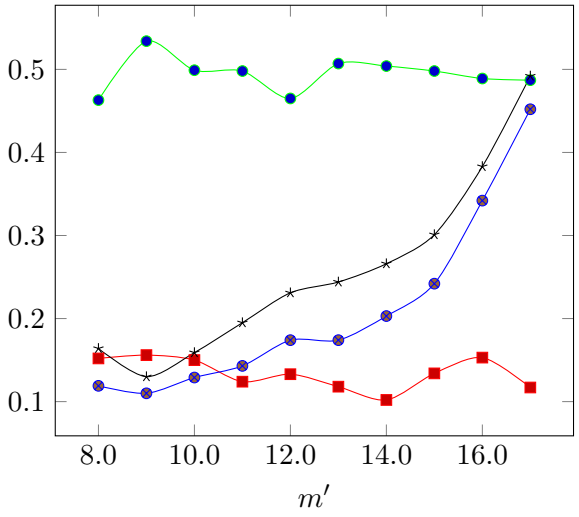
Gambar B.1: Hasil 1



Gambar B.2: Hasil 2



Gambar B.3: Hasil 3



Gambar B.4: Hasil 4