

## PENERAPAN ALGORITMA PRIM UNTUK MENENTUKAN LINTASAN TERPENDEK JARINGAN KABEL INTERNET DI UNIVERSITAS ANDALAS

ILMA PUTERI, MAHDHIVAN SYAFWAN\*, AHMAD IQBAL BAQI

*Jurusan Matematika,  
Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Andalas,  
Kampus UNAND Limau Manis Padang, Indonesia.  
email : mahdhivan@sci.unand.ac.id*

Diterima 4 Agustus 2021    Direvisi 15 September 2021    Dipublikasikan 21 Oktober 2021

**Abstrak.** Penelitian ini dilakukan untuk menentukan total panjang lintasan terpendek jaringan kabel internet yang ada pada gedung-gedung di lingkungan Universitas Andalas. Metode yang digunakan merupakan implementasi dari pohon pembangun minimum (*minimum spanning tree*) dengan mengembangkan algoritma Prim yang programnya dijalankan pada MATLAB. Data yang digunakan adalah gedung-gedung (sebagai titik pada graf), jalan antar gedung (sebagai sisi pada graf) dan jarak antar gedung (sebagai bobot pada graf). Berdasarkan hasil komputasi, diperoleh total panjang lintasan terpendek jaringan kabel internet di Universitas Andalas sebesar 3.606 meter. Hasil ini lebih efisien 1.472 meter atau 28,98% lebih optimal dibandingkan data observasi jaringan yang sudah ada, yaitu 5.078 meter.

**Kata Kunci:** pohon pembangun minimum, algoritma Prim, lintasan terpendek, Universitas Andalas

### 1. Pendahuluan

Teori graf adalah salah satu cabang ilmu matematika yang telah digunakan sejak lama. Salah satu kajian dari graf adalah pohon (*tree*). Pohon sudah digunakan sejak tahun 1857, ketika matematikawan Inggris Arthur Cayley menggunakannya untuk menghitung jumlah senyawa kimia [6]. Pengaplikasian lainnya yang menarik adalah pohon pembangun minimum (*minimum spanning tree*) yang digunakan untuk menyelesaikan permasalahan optimasi diskrit. Salah satu permasalahan yang dapat diselesaikan dengan pohon pembangun minimum ini adalah mencari lintasan terpendek [2]. Persoalan mencari lintasan terpendek di dalam graf menggunakan graf berbobot (*weighted graph*).

Salah satu aplikasi penting dari pohon pembangun minimum adalah desain jaringan kabel internet. Desain ini menghubungkan sekelompok gedung melalui

\*penulis korespondensi

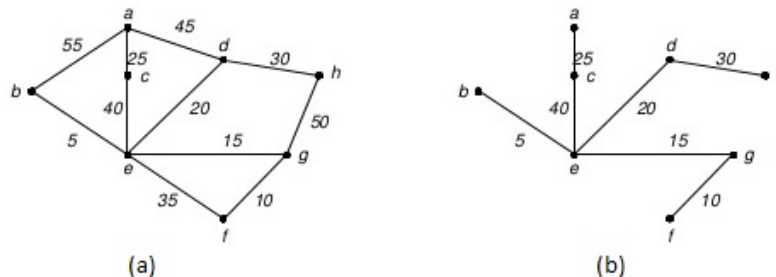
jaringan kabel *fiber optics* dengan jarak yang bervariasi. Pengaplikasian ini dapat digunakan untuk menentukan lintasan yang paling pendek tanpa siklus sehingga semua gedung terhubung oleh kabel internet dengan total jarak minimum. Hubungan kabel-kabel *fiber optics* pada jaringan internet di suatu lokasi dapat disajikan dengan menggunakan sebuah graf. Graf yang digunakan menyatakan gedung-gedung sebagai titik dan lintasan kabel sebagai sisi.

Pada tahun 1930 seorang ahli matematikawan Ceko bernama Voljtech Jarnik menemukan algoritma yang kemudian dipopulerkan kembali pada tahun 1957 oleh Robert Prim. Algoritma ini kemudian dikenal dengan nama algoritma Prim [7]. Algoritma ini tergolong dalam algoritma *greedy*, yaitu membangun suatu pohon pembangun minimum dengan memeriksa sisi demi sisi untuk memilih sisi dengan bobot kecil dan membuang bobot besar. Algoritma ini digunakan untuk mendapatkan solusi yang optimal dari suatu permasalahan [5]. Pada penelitian ini akan dirancang program algoritma Prim pada MATLAB yang dimodifikasi dari [4]. Kemudian program tersebut akan diaplikasikan secara khusus untuk menentukan lintasan terpendek jaringan kabel internet di Unand. Hasil yang diperoleh kemudian dibandingkan dengan data observasi jaringan yang sudah ada.

## 2. Landasan Teori

### 2.1. Pohon Pembangun Minimum (*Minimum Spanning Tree*)

Jika  $G$  adalah graf terhubung berbobot, maka bobot pohon pembangun  $T$  dari  $G$  didefinisikan sebagai jumlah bobot semua sisi di  $T$ . Pohon pembangun yang berbeda mempunyai bobot yang berbeda pula. Di antara semua pohon pembangun di  $G$ , pohon pembangun dengan jumlah bobot terkecil dinamakan **pohon pembangun minimum** (*minimum spanning tree*) [3]. Sebagai contoh, graf pada Gambar 1 (b) adalah pohon pembangun minimum dari graf pada Gambar 1.(a).



Gambar 1. (a)Graf berbobot  $G$  (b)pohon pembangun minimum dari  $G$

### 2.2. Syarat Keoptimalan Cut

Misalkan himpunan titik pada graf  $G$  adalah  $V(G)$ . Kemudian himpunan titik  $V(G)$  dibagi menjadi dua bagian, yaitu himpunan titik  $S$  dan himpunan titik  $\bar{S}$  dimana

$\bar{S} = V(G) - S$ . Setiap **cut** menyatakan suatu himpunan sisi yang terdiri dari sisi-sisi yang memiliki titik ujung di  $S$  dan  $\bar{S}$ . Himpunan sisi *cut* dinyatakan dengan notasi  $[S, \bar{S}]$  [1].

Syarat keoptimalan *cut* menyatakan bahwa setiap sisi pada pohon pembangun minimum memiliki bobot terkecil di antara *cut* yang lainnya. Syarat keoptimalan *cut* juga menyatakan bahwa sisi yang berbeda yang memiliki bobot terkecil yang sama pada *cut* dapat ditambahkan ke beberapa pohon pembangun minimum, yang akan dijelaskan dalam bentuk yang lebih kuat.

**Catatan 2.1.** [1] Misalkan  $F$  adalah sub himpunan sisi-sisi bobot terkecil pohon pembangun dan misalkan  $S$  adalah himpunan titik-titik di  $F$ . Misalkan bahwa  $(i, j)$  adalah sisi dengan bobot terkecil di *cut*  $[S, \bar{S}]$ . Maka pohon pembangun minimum berisi semua sisi di  $F$  termasuk sisi  $(i, j)$ .

Syarat keoptimalan *cut* memberikan karakteristik untuk pohon pembangun minimum yang merujuk pada hubungan antara satu titik pada pohon dan titik-titik diluar pohon (sisi) yang berada pada *cut* yang dihasilkan dari menghapus sisi di pohon.

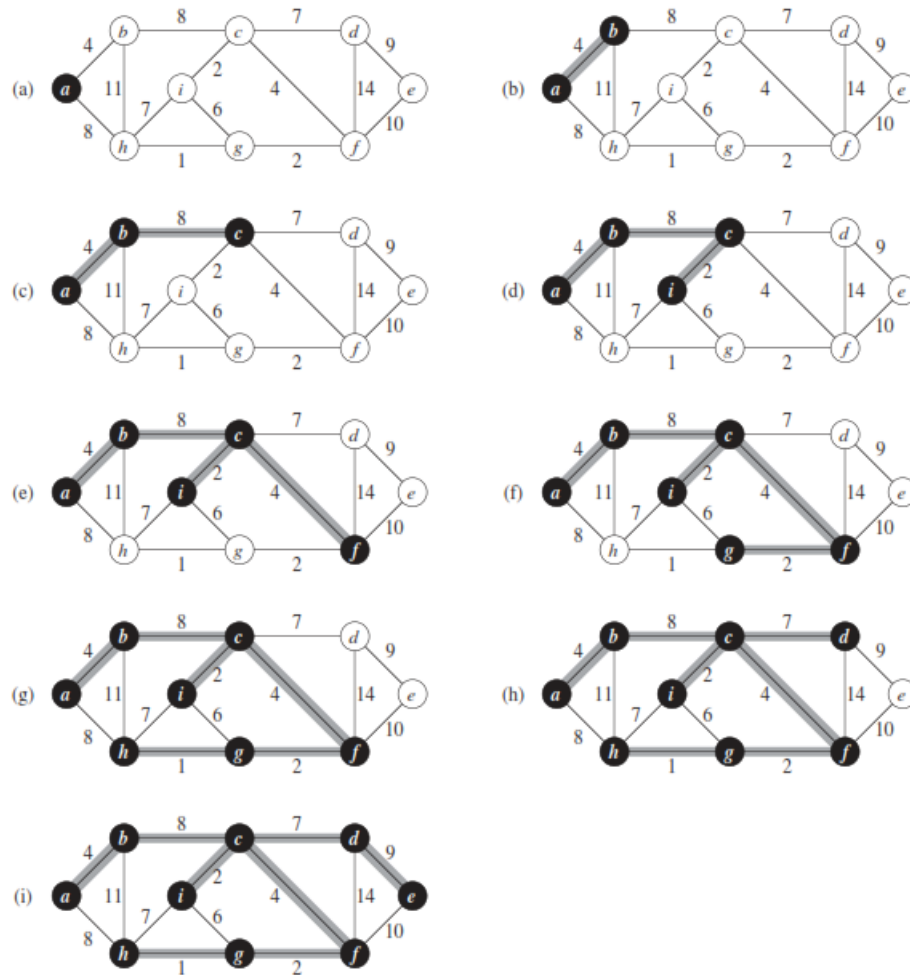
### 2.3. Algoritma Prim

Algoritma Prim membangun pohon pembangun minimum dari sebuah titik, kemudian menambahkan sisi satu per satu. Langkah ini mempertahankan pohon pembangun pada himpunan bagian titik  $S$  (titik pada pohon) dan menambahkan tetangga yang terdekat ke  $S$ . Algoritma dijalankan dengan mengidentifikasi sisi  $(i, j)$  dengan bobot minimum di himpunan sisi *cut*  $[S, \bar{S}]$ . Hal ini menambahkan sisi  $(i, j)$  ke pohon, titik  $j$  ke  $S$ , dan mengulangi langkah dasar ini sampai  $S = V(G)$  [1].

Misalkan  $G = (V(G), E(G))$  adalah graf awal dengan  $n$  titik dan  $T = (V(T), E(T))$  adalah pohon pembangun minimum yang akan dibentuk (mula-mula kosong) dengan  $V(T) = S$ . Secara formal langkah-langkah algoritma Prim adalah sebagai berikut [1]:

- (1)(a) inisialisasi : pilih sebarang titik  $v \in V(G)$ ,
  - (b) masukkan titik  $v$  ke  $S$ ,
  - (c)  $nS = 1$ ,
- (2)  $\bar{S} = V(G) - S$ ,
- (3) selama  $n(S) < n$ 
  - (a) pilih sisi  $(v, w) \in [S, \bar{S}]$  dengan bobot minimum,
  - (b) masukkan sisi  $(v, w)$  ke  $E(T)$ ,
  - (c) masukkan titik  $w$  ke  $S$ ,
  - (d)  $nS = nS + 1$ ,
  - (e) hapus titik  $w$  dari  $\bar{S}$ .

Sebagai contoh, misalkan akan ditentukan pohon pembangun minimum dari graf pada Gambar 2.a. Misalkan diawal dipilih titik  $a$ . Kemudian dipilih sisi yang terkait dengan titik  $a$  dengan bobot terkecil. Selanjutnya proses setiap langkah berjalan seperti pada Gambar 2.

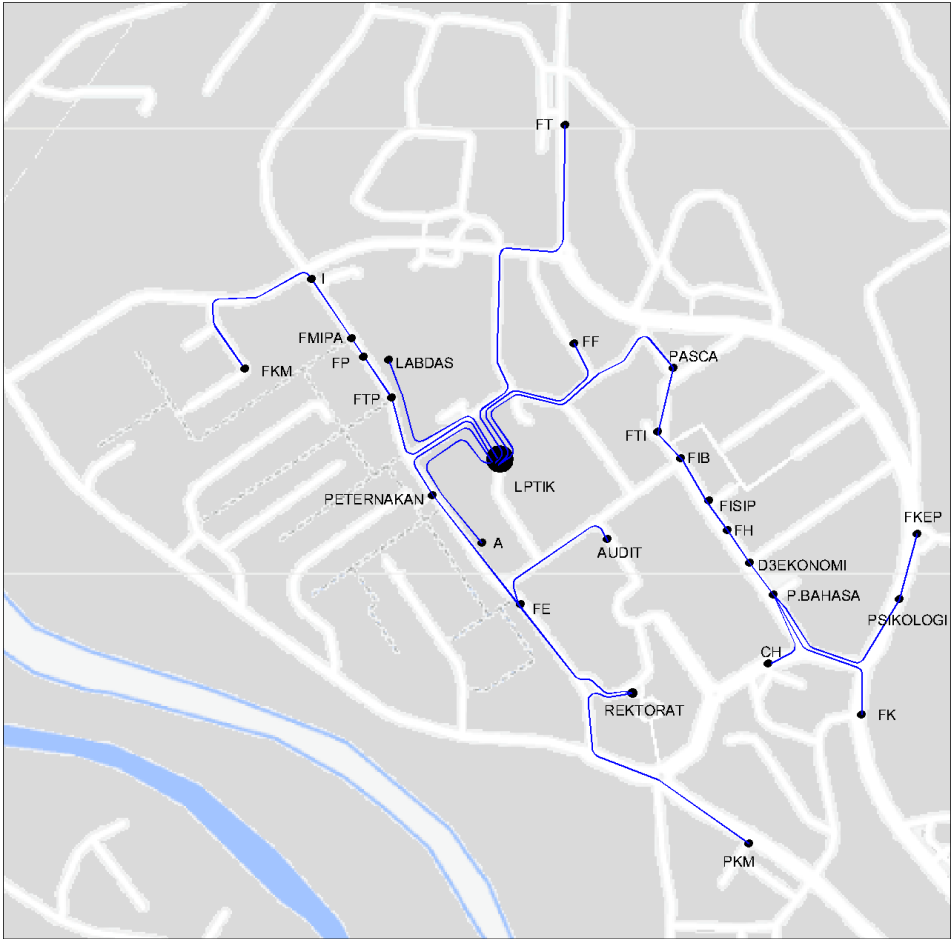


Gambar 2. Pembentukan pohon pembangun minimum dengan algoritma Prim

### 3. Pembahasan

#### 3.1. Data dan Pembentukan Graf

Berdasarkan data jaringan kabel internet di Unand yang didapatkan dari LPTIK (Lembaga Pengembangan Teknologi Informasi dan Komunikasi) [lihat Gambar 3], maka diperoleh data gedung-gedung seperti pada Tabel 1 dan data jarak antar gedung seperti pada Tabel 2. Dalam hal ini jarak diukur dengan aplikasi *google maps* (dalam satuan meter) dan hanya diukur sampai jalan yang dapat dilewati kendaraan beroda dua.



Gambar 3. Jaringan kabel internet di Unand tahun 2021

Tabel 1. Data Gedung Unand Terkait Jaringan Kabel Internet Tahun 2021

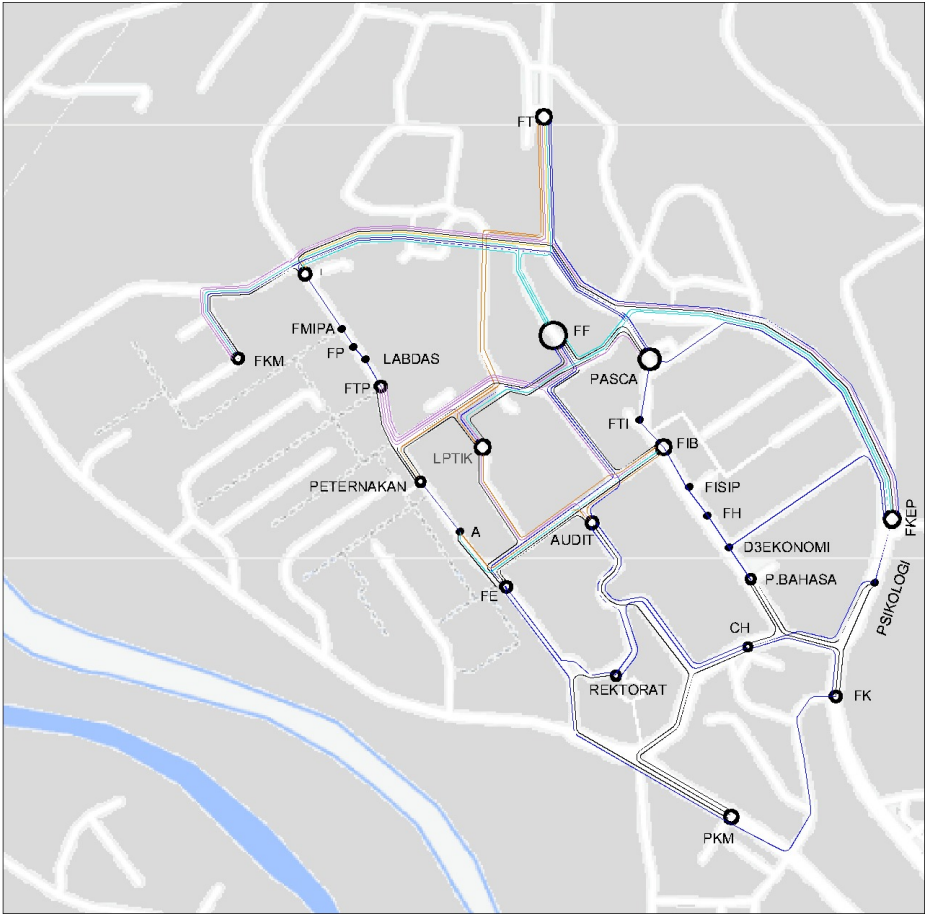
| Titik    | Nama Gedung                   | Titik    | Nama Gedung                      |
|----------|-------------------------------|----------|----------------------------------|
| $v_1$    | LPTIK                         | $v_{14}$ | Fakultas Farmasi                 |
| $v_2$    | Fakultas Kesehatan Masyarakat | $v_{15}$ | Fakultas Teknologi Informasi     |
| $v_3$    | Gedung I                      | $v_{16}$ | Program Pascasarjana             |
| $v_4$    | FMIPA                         | $v_{17}$ | Fakultas Ilmu dan Budaya         |
| $v_5$    | Fakultas Pertanian            | $v_{18}$ | Fakultas Ilmu Sosial dan Politik |
| $v_6$    | Labor Dasar                   | $v_{19}$ | Fakultas Hukum                   |
| $v_7$    | Fakultas Teknologi Pertanian  | $v_{20}$ | Program D3 Ekonomi               |
| $v_8$    | Fakultas Peternakan           | $v_{21}$ | UPT Pusat Bahasa                 |
| $v_9$    | Gedung A                      | $v_{22}$ | Convention Hall                  |
| $v_{10}$ | Fakultas Ekonomi              | $v_{23}$ | Fakultas Kedokteran              |

| Titik    | Nama Gedung     | Titik    | Nama Gedung              |
|----------|-----------------|----------|--------------------------|
| $v_{11}$ | Auditorium      | $v_{24}$ | Program Studi Psikologi  |
| $v_{12}$ | Rektorat        | $v_{25}$ | Fakultas Keperawatan     |
| $v_{13}$ | Fakultas Teknik | $v_{26}$ | Pusat Kegiatan Mahasiswa |

Tabel 2. Data Jaringan Kabel Internet di Unand Tahun 2021

| No.   | Titik Awal | Titik Akhir | Sisi               | Bobot (meter) |
|-------|------------|-------------|--------------------|---------------|
| 1     | $v_1$      | $v_6$       | $(v_1, v_6)$       | 258           |
| 2     |            | $v_7$       | $(v_1, v_7)$       | 223           |
| 3     |            | $v_8$       | $(v_1, v_8)$       | 200           |
| 4     |            | $v_9$       | $(v_1, v_9)$       | 276           |
| 5     |            | $v_{13}$    | $(v_1, v_{13})$    | 613           |
| 6     |            | $v_{14}$    | $(v_1, v_{14})$    | 313           |
| 7     |            | $v_{16}$    | $(v_1, v_{16})$    | 463           |
| 8     | $v_2$      | $v_3$       | $(v_2, v_3)$       | 281           |
| 9     | $v_3$      | $v_4$       | $(v_3, v_4)$       | 110           |
| 10    | $v_4$      | $v_5$       | $(v_4, v_5)$       | 23            |
| 11    | $v_5$      | $v_7$       | $(v_5, v_7)$       | 52            |
| 12    | $v_8$      | $v_{10}$    | $(v_8, v_{10})$    | 184           |
| 13    | $v_{10}$   | $v_{11}$    | $(v_{10}, v_{11})$ | 241           |
| 14    |            | $v_{12}$    | $(v_{10}, v_{12})$ | 237           |
| 15    | $v_{12}$   | $v_{26}$    | $(v_{12}, v_{26})$ | 454           |
| 16    | $v_{15}$   | $v_{16}$    | $(v_{15}, v_{16})$ | 86            |
| 17    |            | $v_{17}$    | $(v_{15}, v_{17})$ | 40            |
| 18    | $v_{17}$   | $v_{18}$    | $(v_{17}, v_{18})$ | 71            |
| 19    | $v_{18}$   | $v_{19}$    | $(v_{18}, v_{19})$ | 25            |
| 20    | $v_{19}$   | $v_{20}$    | $(v_{19}, v_{20})$ | 71            |
| 21    | $v_{20}$   | $v_{21}$    | $(v_{20}, v_{21})$ | 57            |
| 22    | $v_{21}$   | $v_{22}$    | $(v_{21}, v_{22})$ | 164           |
| 23    |            | $v_{23}$    | $(v_{21}, v_{23})$ | 258           |
| 24    |            | $v_{24}$    | $(v_{21}, v_{24})$ | 208           |
| 25    | $v_{24}$   | $v_{25}$    | $(v_{24}, v_{25})$ | 170           |
| Total |            |             |                    | 5078          |

Dari Gambar 3 kemudian dibangun semua kemungkinan lintasan jaringan kabel internet di Unand. Diperoleh gambar semua kemungkinan lintasan jaringan kabel internet di Unand seperti Gambar 4 dan datanya seperti Tabel 3. Data ini adalah data yang akan digunakan pada program algoritma Prim.



Gambar 4. Semua kemungkinan jaringan kabel internet di Unand

Tabel 3. Data Awal

| No. | Titik Awal | Titik Akhir | Sisi            | Bobot (meter) |
|-----|------------|-------------|-----------------|---------------|
| 1   | $v_1$      | $v_7$       | $(v_1, v_7)$    | 223           |
| 2   |            | $v_8$       | $(v_1, v_8)$    | 200           |
| 3   |            | $v_{10}$    | $(v_1, v_{10})$ | 275           |
| 4   |            | $v_{11}$    | $(v_1, v_{11})$ | 336           |
| 5   |            | $v_{13}$    | $(v_1, v_{13})$ | 613           |
| 6   |            | $v_{14}$    | $(v_1, v_{14})$ | 313           |
| 7   |            | $v_{16}$    | $(v_1, v_{16})$ | 463           |
| 8   |            | $v_{17}$    | $(v_1, v_{17})$ | 415           |
| 9   |            | $v_{25}$    | $(v_1, v_{25})$ | 904           |
| 10  | $v_2$      | $v_3$       | $(v_2, v_3)$    | 281           |
| 11  |            | $v_{13}$    | $(v_2, v_{13})$ | 809           |
| 12  |            | $v_{14}$    | $(v_2, v_{14})$ | 717           |
| 13  |            | $v_{16}$    | $(v_2, v_{16})$ | 879           |

| No. | Titik Awal | Titik Akhir | Sisi               | Bobot (meter) |
|-----|------------|-------------|--------------------|---------------|
| 14  |            | $v_{25}$    | $(v_2, v_{25})$    | 1298          |
| 15  | $v_3$      | $v_4$       | $(v_3, v_4)$       | 110           |
| 16  |            | $v_{13}$    | $(v_3, v_{13})$    | 543           |
| 17  |            | $v_{14}$    | $(v_3, v_{14})$    | 498           |
| 18  |            | $v_{16}$    | $(v_3, v_{16})$    | 660           |
| 19  |            | $v_{25}$    | $(v_3, v_{25})$    | 1079          |
| 20  | $v_4$      | $v_5$       | $(v_4, v_5)$       | 23            |
| 21  | $v_5$      | $v_6$       | $(v_5, v_6)$       | 17            |
| 22  | $v_6$      | $v_7$       | $(v_6, v_7)$       | 35            |
| 23  | $v_7$      | $v_8$       | $(v_7, v_8)$       | 147           |
| 24  |            | $v_{14}$    | $(v_7, v_{14})$    | 436           |
| 25  |            | $v_{16}$    | $(v_7, v_{16})$    | 586           |
| 26  | $v_8$      | $v_{17}$    | $(v_7, v_{17})$    | 564           |
| 27  |            | $v_9$       | $(v_8, v_9)$       | 76            |
| 28  |            | $v_{13}$    | $(v_8, v_{13})$    | 713           |
| 29  |            | $v_{14}$    | $(v_8, v_{14})$    | 413           |
| 30  | $v_9$      | $v_{10}$    | $(v_9, v_{10})$    | 108           |
| 31  |            | $v_{11}$    | $(v_9, v_{11})$    | 297           |
| 32  |            | $v_{17}$    | $(v_9, v_{17})$    | 376           |
| 33  | $v_{10}$   | $v_{11}$    | $(v_{10}, v_{11})$ | 241           |
| 34  |            | $v_{12}$    | $(v_{10}, v_{12})$ | 237           |
| 35  |            | $v_{14}$    | $(v_{10}, v_{14})$ | 512           |
| 36  |            | $v_{17}$    | $(v_{10}, v_{17})$ | 320           |
| 37  |            | $v_{26}$    | $(v_{10}, v_{26})$ | 497           |
| 38  | $v_{11}$   | $v_{12}$    | $(v_{11}, v_{12})$ | 222           |
| 39  |            | $v_{14}$    | $(v_{11}, v_{14})$ | 383           |
| 40  |            | $v_{17}$    | $(v_{11}, v_{17})$ | 191           |
| 41  |            | $v_{22}$    | $(v_{11}, v_{22})$ | 356           |
| 42  |            | $v_{26}$    | $(v_{11}, v_{26})$ | 570           |
| 43  | $v_{12}$   | $v_{22}$    | $(v_{12}, v_{22})$ | 318           |
| 44  |            | $v_{26}$    | $(v_{12}, v_{26})$ | 454           |
| 45  | $v_{13}$   | $v_{14}$    | $(v_{13}, v_{14})$ | 372           |
| 46  |            | $v_{16}$    | $(v_{13}, v_{16})$ | 472           |
| 47  |            | $v_{25}$    | $(v_{13}, v_{25})$ | 891           |
| 48  | $v_{14}$   | $v_{16}$    | $(v_{14}, v_{16})$ | 252           |
| 49  |            | $v_{17}$    | $(v_{14}, v_{17})$ | 338           |
| 50  |            | $v_{25}$    | $(v_{14}, v_{25})$ | 693           |
| 51  | $v_{15}$   | $v_{16}$    | $(v_{15}, v_{16})$ | 86            |
| 52  |            | $v_{17}$    | $(v_{15}, v_{17})$ | 40            |
| 53  | $v_{16}$   | $v_{25}$    | $(v_{16}, v_{25})$ | 532           |
| 54  | $v_{17}$   | $v_{18}$    | $(v_{17}, v_{18})$ | 71            |
| 55  | $v_{18}$   | $v_{19}$    | $(v_{18}, v_{19})$ | 25            |
| 56  | $v_{19}$   | $v_{20}$    | $(v_{19}, v_{20})$ | 71            |
| 57  | $v_{20}$   | $v_{21}$    | $(v_{20}, v_{21})$ | 57            |
| 58  |            | $v_{25}$    | $(v_{20}, v_{25})$ | 345           |
| 59  | $v_{21}$   | $v_{22}$    | $(v_{21}, v_{22})$ | 164           |
| 60  |            | $v_{23}$    | $(v_{21}, v_{23})$ | 258           |
| 61  |            | $v_{24}$    | $(v_{21}, v_{24})$ | 208           |
| 62  | $v_{22}$   | $v_{23}$    | $(v_{22}, v_{23})$ | 222           |
| 63  |            | $v_{24}$    | $(v_{22}, v_{24})$ | 178           |
| 64  |            | $v_{26}$    | $(v_{22}, v_{26})$ | 370           |
| 65  | $v_{23}$   | $v_{24}$    | $(v_{23}, v_{24})$ | 106           |

| No.   | Titik Awal | Titik Akhir | Sisi               | Bobot (meter) |
|-------|------------|-------------|--------------------|---------------|
| 66    |            | $v_{26}$    | $(v_{23}, v_{26})$ | 367           |
| 67    | $v_{24}$   | $v_{25}$    | $(v_{24}, v_{25})$ | 170           |
| Total |            |             |                    | 24996         |

### 3.2. Program Algoritma Prim

Pada bagian ini akan dibahas program komputasi menggunakan algoritma Prim untuk menentukan bobot sisi terkecil antara titik, dengan mengacu dari [4]. Program berbentuk fungsi dengan input matriks ketetanggaan graf awal yang didefinisikan sebagai  $G$ , output matriks pohon pembangun minimum dinamakan  $T$  dan output jumlah bobot pohon pembangun minimum dinamakan *totweight*. Untuk input matriks dapat didefinisikan pada *command window* atau dengan *impor data* dari file *microsoft excel*. Di bawah ini adalah langkah-langkah yang digunakan pada program *prim*.

- (1) Pengecekan kesimetrian matriks dengan perintah

```
[n,m] = size(G);
if n ~= m
    disp(' Error: Number of rows and columns in the matrix must
    be the same '))
    return
if sum(diag(G))~= 0 | G~=G'
    disp(' Error: Adjacency matrix must be symmetric ')
    return
end
```

- (2) Inisialisasi

- (a) Pilih titik awal yaitu titik  $1 \in V(G)$  dan masukkan ke matriks  $S$  sehingga jumlah titik di graf  $T$  adalah 1. Berikut perintah yang digunakan:

```
S = [1];
nS = 1;
```

- (b) Karena sisi di graf  $T$  masih kosong maka jumlah sisi adalah 0. Berikut perintah yang digunakan:

```
nmst = 0;
```

- (3) Definisikan titik-titik yang belum berada di  $T$ . Misalkan  $\bar{S}$  adalah himpunan titik yang belum berada di  $T$  dengan perintah yang digunakan :

```
Sbar = [2:n]';
nSbar = n-1;
```

- (4) Kemudian lakukan perulangan selama jumlah titik di  $T$  lebih kecil dari jumlah titik di  $G$ .

- (a) Inisialisasi bobot terkecil dengan perintah :

```
minweight =  $\infty$  ;
```

- (b) Lakukan perulangan untuk memilih sisi dengan bobot terkecil dengan pengecekan bobot sisi satu persatu dari *cut* titik yang berada di pohon dan yang belum berada di pohon dengan pemanggilan indeks matriks dengan perintah :

```
for i = 1 : nS
    for j = 1 : nSbar
        ii = S(i);
        jj = Sbar(j);
        if G(ii,jj)<minweight && G(ii,jj)>0
            minweight = G(ii,jj);
            iisave = ii; jjsave = jj;
            jsave = j;
        end
    end
end
```

- (c) Tambahkan nilai sisi graf  $T$  dengan perintah :

```
nmst = nmst + 1;
```

- (d) Simpan titik ujung dari sisi terpilih kedalam matriks dengan perintah :

```
mst(nmst,1) = iisave;
mst(nmst,2) = jjsave;
```

- (e) Simpan bobot terpilih kedalam matriks *weight* dengan perintah :

```
weight(nmst,1) = minweight;
```

- (f) Tambahkan nilai jumlah titik graf  $T$  dengan perintah :

```
nS = nS + 1;
S = [S; jjsave];
```

- (g) Hapus titik akhir terpilih *jjsave* dari  $\bar{S}$  dan kurangkan nilai jumlah titiknya dengan perintah :

```
for j = jsave+1:nSbar
    Sbar(j-1) = Sbar(j);
end
nSbar = nSbar - 1;
```

- (5) Setelah dilakukan *looping*, hitung nilai jumlah bobot sisi-sisi dari pohon pembangun minimum dengan perintah :

```
totweight = sum(weight);
```

- (6) Karena akan ditampilkan graf dari pohon pembangun minimum, semua titik dan sisinya dideskripsikan dalam matriks persegi dengan perintah :

```
T = sparse(mst(:,1)',mst(:,2)',full(weight));
[a,b] = size(T);
c = b-a;
X = [full(T);zeros(c,n)];
```

(7) Untuk menampilkan graf dari pohon pembangun minimum digunakan perintah:

```
view(biograph(X,[ ],'ShowArrows','off','ShowWeights','on','Scale',
0.8));
```

### 3.3. Hasil Program Algoritma Prim untuk Menentukan Panjang Minimum Jaringan Kabel Internet di Unand

Dengan menggunakan program algoritma Prim, diperoleh *output* pohon pembangun minimum dari jaringan kabel internet di Unand, seperti pada Gambar 5.

---

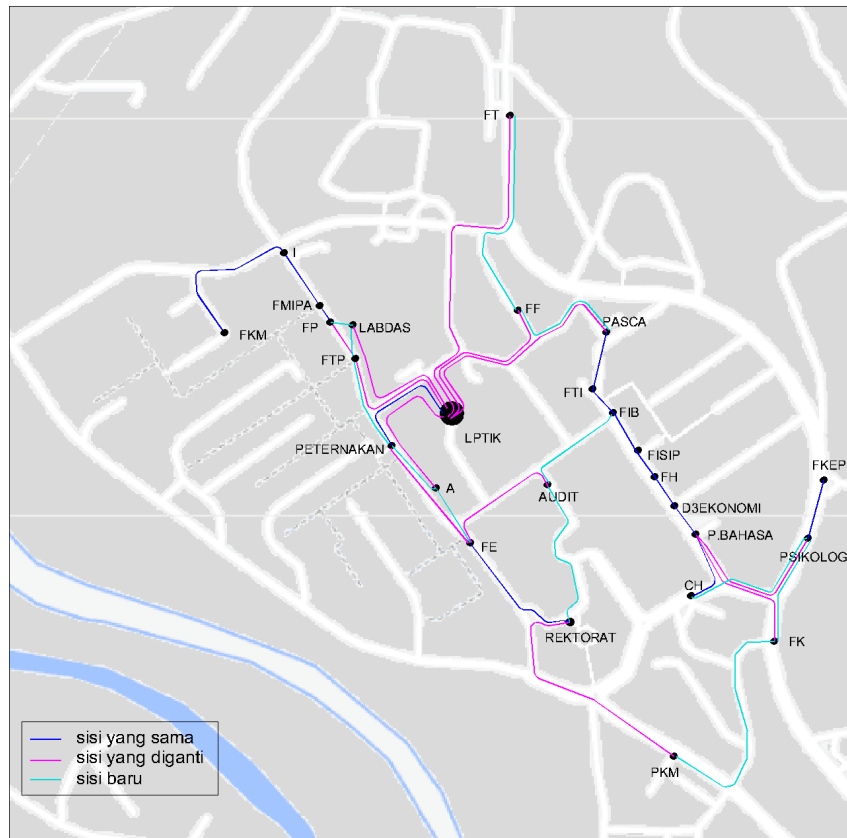
|                                         |  |
|-----------------------------------------|--|
| Total weights of minimum spanning tree: |  |
| 3606                                    |  |

---

|                                                    |     |
|----------------------------------------------------|-----|
| Edges in minimum spanning tree and their weights : |     |
| (3,2)                                              | 281 |
| (4,3)                                              | 110 |
| (5,4)                                              | 23  |
| (6,5)                                              | 17  |
| (7,6)                                              | 35  |
| (8,7)                                              | 147 |
| (1,8)                                              | 200 |
| (8,9)                                              | 76  |
| (9,10)                                             | 108 |
| (12,11)                                            | 222 |
| (10,12)                                            | 237 |
| (14,13)                                            | 372 |
| (16,14)                                            | 252 |
| (17,15)                                            | 40  |
| (15,16)                                            | 86  |
| (11,17)                                            | 191 |
| (17,18)                                            | 71  |
| (18,19)                                            | 25  |
| (19,20)                                            | 71  |
| (20,21)                                            | 57  |
| (21,22)                                            | 164 |
| (24,23)                                            | 106 |
| (22,24)                                            | 178 |
| (24,25)                                            | 170 |
| (23,26)                                            | 367 |

Gambar 5. *Output* program algoritma Prim

Dari hasil program algoritma Prim terdapat beberapa perbedaan lintasan jaringan kabel yang ada saat ini dengan hasil komputasi algoritma Prim. Gambar dan tabel dari sisi-sisi yang berbeda ditunjukkan pada Gambar 6 dan Tabel 4. Dari hasil tersebut, didapatkan 12 sisi yang berbeda. Jika sisi-sisi pada graf awal dihapus dan digantikan dengan sisi hasil algoritma Prim, maka total lintasan jaringan kabel internet akan lebih pendek.



Gambar 6. Perbandingan Jaringan Kabel Internet di Unand antara yang ada saat ini dan yang dihasilkan dari Algoritma Prim

Tabel 4. Perbandingan Sisi Graf Tahun 2021 dan Graf Hasil Algoritma Prim

| Sisi yang Diganti                           | Sisi Baru                         |
|---------------------------------------------|-----------------------------------|
| (Labor Dasar, LPTIK)                        | (Fakultas Pertanian, Labor Dasar) |
| (Fakultas Peternakan, FTP)                  | (Labor Dasar, FTP)                |
| (FTP, LPTIK)                                | (FTP, Fakultas Peternakan)        |
| (LPTIK, gedung A)                           | (Fakultas Peternakan, gedung A)   |
| (Fakultas Peternakan, FE)                   | (gedung A, FE)                    |
| (FE, Auditorium)                            | (Auditorium, Rektorat)            |
| (Rektorat, PKM)                             | (PKM, FK)                         |
|                                             | (Auditorium, FIB)                 |
| (LPTIK, FT)                                 | (FT, FF)                          |
| (LPTIK, FF)                                 | (FF, Program Pascasarjana)        |
| (LPTIK, Program Pascasarjana)               |                                   |
| (UPT Pusat Bahasa, FK)                      | (CH, Program Studi Psikologi)     |
| (UPT Pusat Bahasa, Program Studi Psikologi) | (Program Studi Psikologi, FK)     |

Dari proses algoritma Prim di atas diperoleh panjang total lintasan kabel adalah 3.606 meter, sedangkan panjang total lintasan kabel yang ada saat ini berdasarkan data observasi adalah 5.078 meter. Hal ini berarti dengan menggunakan algoritma Prim dapat menghemat lintasan kabel sepanjang 1.472 meter atau 28,98% lebih optimal.

#### 4. Kesimpulan

Pada penelitian ini telah dihitung total panjang lintasan terpendek jaringan kabel internet di Universitas Andalas dengan menggunakan algoritma Prim yang dikembangkan dari [4]. Berdasarkan hasil komputasi, diperoleh total panjang lintasan terpendek jaringan kabel internet di Universitas Andalas sebesar 3.606 meter. Hasil ini lebih efisien 1.472 meter atau 28,98% lebih optimal dibandingkan data observasi jaringan yang sudah ada, yaitu 5.078 meter. Untuk penelitian selanjutnya, disarankan agar melakukan perbandingan efisiensi komputasi dengan menggunakan algoritma yang lainnya, seperti algoritma Kruskal dan algoritma Solin.

#### 5. Ucapan Terima kasih

Penulis mengucapkan terimakasih kepada Bapak Prof. Dr. Muhafzan, Ibu Dr. Lyra Yulianti dan Bapak Efendi, M.Si yang telah memberikan masukan dan saran sehingga paper ini dapat diselesaikan dengan baik.

#### Daftar Pustaka

- [1] Ahuja, R. K., Magnanti, T. L. dan Orlin, J. B., 1993, *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, Upper Saddle River, New Jersey
- [2] Akhirina, T. Y. dan Afrizal, T., Pendekatan Matriks Ketetangaan Berbobot untuk Solusi Minimum Spanning Tree (MST), *Satuan Tulisan Riset dan Inovasi Teknologi*, 4(3):pp.280-287.
- [3] Chartrand, G. dan Zhang, P., 2006, *Introduction to Graph Theory*, The McGraw-Hill, New York
- [4] Kundur, V., 2005, Prims Algorithm, <https://www.mathworks.com/matlabcentral/fileexchange/8569-prims-algorithm>, diakses pada 18 Mei 2021
- [5] Nugraha, D. W., 2012, Penerapan Kompleksitas waktu Algoritma Prim untuk Menghitung Kemampuan Komputer Melaksanakan Perintah, *Jurnal Ilmiah Foristek*, 2(2):pp.195-207
- [6] Rinaldi, M., 2010, *Matematika Diskrit*, Edisi Ketiga, Informatika, Bandung
- [7] Rosen, K. H., 2019, *Discrete Mathematics and Its Applications*, Edisi Kedelapan, McGraw-Hill Education, New York