PENGEMBANGAN *HYBRID MODIFIED PENALTY* DAN ALGORITMA PRIM UNTUK MENYELESAIKAN MASALAH MPDCMST (*MULTI PERIOD DEGREE CONSTRAINED MINIMUM SPANNING TREE*)

(Skripsi)

Oleh

RAHMAT WIKA KENCANA

JURUSAN ILMU KOMPUTER

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS LAMPUNG

2017

ABSTRAK

THE HYBRID OF MODIFIED PENALTY AND MODIFIED PRIM'S ALGORITHM TO SOLVE THE MULTI PERIODS INSTALLATION PROBLEM

Oleh

RAHMAT WIKA KENCANA

The Multi Period Degree Constrained Minimum Spanning Tree (MPDCMST) is a problem f or determining the installation of a network. The backbone of the problem is the MST (Minimum Spanning Tree) problem. If on MST we add degree restriction on every vertex, then it becomes the Degree Constrained Minimum Spanning Tree (DCMST) Problem. In addition to that resctriction, if we add another constraint, which is period, then it becomes The Multi Period Degree Constrained Minimum Spanning Tree (MPDCMST) problem. In this research will use hybrid methods between Modified Prim and Modified Penalty to solved MPDCMST. Modified Penalty is used to tackle the edge which its insertion cause degree violation, and Modified Prim is used because during the installation processes the existing network is still connected. We are not using Kruskal's algorithm and its modification because there is no guarantee that the existing network still connected. In this research we developed three algorithms named as WAK1, WAK2, WAK3. The results shows that the best algorithm is WAK3.

Keyword : *multi periods, degree constrained, installation of a network, Modified Penalty, Modified Prim*

ABSTRAK

PENGEMBANGAN *HYBRID MODIFIED PENALTY* DAN ALGORITMA PRIM UNTUK MENYELESAIKAN MASALAH MPDCMST (*MULTI PERIOD DEGREE CONSTRAINED MINIMUM SPANNING TREE*)

Oleh

RAHMAT WIKA KENCANA

Multi Period Degree Constrained Minimum Spanning Tree (MPDCMST) adalah masalah penentuan instalasi suatu jaringan dengan adanya kendala tahap dan inter koneksi suatu jaringan. *Backbone* dari masalah ini adalah masalah MST (*Minimum Spanning Tree*). Jika MST diberi tambahan kendala *degree* maka masalah menjadi DCMST (*Degree Constrained Minimum Spanning Tree*), dan jika DCMST ditambah lagi dengan kendala *period* (tahap) maka masalah tersebut menjadi masalah MPDCMST. Algoritma Prim merupakan salah satu algoritma yang terkenal untuk menyelesaikan masalah MST, akan tetapi jika algoritma tersebut digunakan untuk menyelesaikan DCMST ataupun MPDCMST, maka dalam penerapannya akan dilakukan modifikasi terhadap Algoritma Prim tersebut. Pada penelitian ini akan digunakan *hybrid* antara *Modified* Prim dan *Modified Penalty* untuk menyelesaikan MPDCMST. *Modified Penalty* digunakan untuk memodifikasi *edge* yang telah ada dan akan masuk dalam jaringan yang keberadaan *edge* tersebut akan melanggar kendala *degree*. Dari penelitian ini dengan melakukan *hybrid Modified Penalty* dan *Modified* Prim dikembangkan 3 Algoritma yang diberi nama WAK1, WAK2, WAK3. Dari hasil pengujian yang dilakukan menggunakan kasus uji seperti pada penelitian sebelumnya, hasil solusi terbaik dihasilkan oleh Algoritma WAK3.

Kata Kunci : *multi periods, degree constrained,* instalasi jaringan, *Modified Penalty, Modified Prim*

PENGEMBANGAN *HYBRID MODIFIED PENALTY* DAN ALGORITMA PRIM UNTUK MENYELESAIKAN MASALAH MPDCMST (*MULTI PERIOD DEGREE CONSTRAINED MINIMUM SPANNING TREE*)

Oleh

RAHMAT WIKA KENCANA

Skripsi

Sebagai Salah Satu Syarat Untuk Memperoleh Gelar
SARJANA KOMPUTER

pada

Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam

JURUSAN ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
2017

Judul Skripsi	:	**PENGEMBANGAN *HYBRID MODIFIED PENALTY* DAN ALGORITMA PRIM UNTUK MENYELESAIKAN MASALAH MPDCMST (*MULTI PERIOD DEGREE CONSTRAINED MINIMUM SPANNING TREE*)**
Nama Mahasiswa	:	**Rahmat Wika Kencana**
Nomor Pokok Mahasiswa	:	1317051051
Jurusan	:	Ilmu Komputer
Fakultas	:	Matematika dan Ilmu Pengetahuan Alam

MENYETUJUI

1. Komisi Pembimbing

Dra. Wamiliana, M.A., Ph.D.	**Astria Hijriani, S.Kom., M.Kom.**
NIP. 19631108 198902 2 001	NIP. 19810308 200812 2 002

2. Mengetahui

Ketua Jurusan Ilmu Komputer

Dr. Ir. Kurnia Muludi, M.S.Sc.
NIP. 19640616 198902 1 001

MENGESAHKAN

1. Tim Penguji

Ketua : **Dra. Wamiliana, M.A., Ph.D.** .....

Sekertaris : **Astria Hijriani, S.Kom., M.Kom.** .....

Penguji
Bukan Pembimbing : **Ir. Machudor Yusman, M.Kom.** .....

2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Prof. Warsito, S.Si., D.E.A., Ph.D.
NIP. 19710212 199512 1 001

Tanggal Lulus Ujian Skripsi : **25 Januari 2017**

PERNYATAAN

Saya yang bertanda tangan di bawah ini, menyatakan bahwa skripsi saya yang berjudul "Pengembangan *Hybrid Modified Penalty* dan Algoritma Prim Untuk Menyelesaikan Masalah MPDCMST (*Multi Period Degree Constrained Minimum Spanning Tree*)" merupakan karya saya sendiri dan bukan hasil karya orang lain. Semua tulisan yang tertuang di skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti bahwa skripsi saya merupakan hasil penjiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang telah saya terima.

Bandar Lampung, 25 Januari 2017

Rahmat Wika Kencana
NPM. 1317051051

RIWAYAT HIDUP

Penulis dilahirkan pada tanggal 6 Desember 1995 di Pulung Kencana, Way Abung. Penulis merupakan anak pertama dari dua bersaudara dengan ayah bernama Widodo dan ibu bernama Markamah. Penulis menyelesaikan pendidikan formal pertama kali di SD Negeri 1 Bumi Dipasena Abadi tahun 2007, kemudian melanjutkan pendidikan menengah pertama di SMP Negeri 1 Punggur dan selesai pada tahun 2010. Pendidikan menengah atas di SMA Negeri 1 Kotagajah diselesaikan penulis pada tahun 2013.

Pada tahun 2013, penulis terdaftar sebagai mahasiswa Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung melalui jalur SBMPTN. Pada bulan Februari tahun 2016, penulis melakukan Kerja Praktik di PT. AirMedia Persada, Yogyakarta. Pada bulan Juli tahun 2016 penulis melaksanakan Kuliah Kerja Nyata di Taman Sari Kecamatan Selagai Lingga Kabupaten Lampung Tengah. Selama menjadi mahasiswa, penulis menjadi anggota Bidang MediInfo Himpunan Mahasiswa Jurusan Ilmu Komputer pada tahun periode 2014-2015 dan Asisten Laboratorium dan Asisten Dosen Jurusan Ilmu Komputer pada tahun periode 2014-2017.

PERSEMBAHAN

Puji dan syukur saya ucapkan kepada Allah SWT atas

segala nikmat dan karunia-Nya sehingga skripsi ini dapat diselesaikan.

Kupersembahkan karya kecilku ini untuk:

Ibuku, yang telah melahirkanku

Ibuku, yang telah merawat dan membesarkanku

Ibuku, yang telah mendidikku

Ayahku tercinta, yang telah membesarkanku dengan seluruh kasih dan sayangnya, memberikan pengetahuannya, dan selalu mendukung serta mendoakan untuk keberhasilanku.

Adik serta keluarga besarku yang selalu kusayangi

dan, Almamater yang kubanggakan

UNIVERSITAS LAMPUNG

MOTTO

“Study while others are sleeping; work while others are loafing, prepare while others are playing, and dream while others are wishing.”
(William Arthus Ward)

“Success is not the key to happiness. Happiness is the key to success. If you love what you are doing, you will be successful.
Example is not the main thing in influencing others. It is the only thing.
There are two means of refuge from the miseries of life: music and cats. Until he extends his circle of compassion to include all living things, man will not himself find peace.
The only ones among you who will be really happy are those who will have sought and found how to serve.
The purpose of human life is to serve, and to show compassion and the will to help others.
Happiness is nothing more than good health and a bad memory.
Do something wonderful, people may imitate it.
Constant kindness can accomplish much. As the sun makes ice melt, kindness causes misunderstanding, mistrust, and hostility to evaporate.
Sometimes our light goes out but is blown into flame by another human being. Each of us owes deepest thanks to those who have rekindled this light. “
(Albert Schweitzer)

SANWACANA

Puji dan syukur penulis ucapkan kehadirat Allah SWT atas segala rahmat, hidayah dan kesehatan yang diberikan kepada penulis sehingga dapat menyelesaikan penulisan skripsi ini.

Skripsi ini disusun sebagai syarat untuk memperoleh gelar Sarjana Komputer di Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Universitas Lampung. Judul dari skripsi ini adalah "Pengembangan *Hybrid Modified Penalty* Dan Algoritma Prim Untuk Menyelesaikan Masalah MPDCMST (*Multi Period Degree Constrained Minimum Spanning Tree*)".

Dalam penyusunan skripsi ini, penulis banyak menghadapi kesulitan. Namun, berkat bantuan dan dorongan dari berbagai pihak, akhirnya penulis dapat menyelesaikan skripsi ini. Untuk itu pada kesempatan ini, penulis mengucapkan terimakasih kepada:

1. Kedua orang tua, Bapak Widodo dan Ibu Markamah, Adikku tersayang Mutiara Wika Azyaza, serta keluarga besar yang selalu memberikan doa, motivasi dan kasih sayang yang tak terhingga.
2. Ibu Dra. Wamiliana, Ph.D. sebagai pembimbing I, yang telah membimbing penulis dan memberikan ide, kritik serta saran sehingga penulisan skripsi ini dapat diselesaikan.

3. Ibu Astria Hijriani, S.Kom., M.Kom. sebagai pembimbing II, yang telah memberikan saran, bantuan dan membimbing penulis dalam pembuatan skripsi ini.
4. Bapak Ir. Machudor Yusman, M.Kom. sebagai pembahas, yang telah memberikan masukan-masukan yang bermanfaat dalam perbaikan skripsi ini.
5. Bapak dan Ibu Dosen serta seluruh staf Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung yang telah memberikan ilmu dan pengalaman dalam hidup untuk menjadi lebih baik.
6. Seluruh mahasiswa angkatan 2013 yang berjuang bersama dalam suka maupun duka, saling membantu, saling bertanya dalam menyelesaikan karya-karyanya selama di Universitas Lampung. Kebersamaan yang telah dilalui menjadi pengalaman berharga bagi penulis.

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan, akan tetapi sedikit harapan semoga skripsi ini bermanfaat bagi perkembangan ilmu pengetahuan terutama bagi teman-teman Ilmu Komputer.

Bandar Lampung, 25 Januari 2017

Rahmat Wika Kencana

DAFTAR ISI

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR TABEL

Halaman

BAB I

PENDAHULUAN

1.1.Latar Belakang

Jaringan adalah suatu bentuk relasi atau hubungan antara dua objek atau lebih. Saat ini jaringan memegang peranan penting dalam aktifitas keseharian saat ini. Sehingga tidak dapat dipungkiri, bahwa aktifitas manusia saat ini sangat erat kaitannya dengan jaringan. Sebagai contoh adalah jaringan listrik, telekomunikasi, jalan, air bersih, dan komputer. Akan tetapi, segala kegiatan tersebut dapat terganggu apabila terjadi kerusakan pada jaringan tersebut, misalnya pada jaringan PDAM. Apabila terjadi kebocoran pada suatu titik, maka dapat menghambat penyaluran air ke tempat lain yang menyambung pada titik tersebut. Begitu pula sama halnya dengan instalasi *router/hub* ataupun *hotspot* yang ada dalam sebuah jaringan. Apabila titik atau node sebagai pusat *router/hub* ataupun hotspot, mengalami kerusakan atau *down*, maka semua perangkat yang menghubung dengan titik tersebut tidak dapat mengakses jaringan.

Permasalahan yang sering terjadi dalam sebuah jaringan adalah tahap dimana dilakukannya instalasi jaringan tersebut. Apabila dana untuk dilakukannya instalasi jaringan memenuhi atau sesuai dengan target, maka instalasi jaringan dapat dilakukan dalam satu tahap. Namun, pada kenyataannya, sering kali

suatu instansi mengalami permasalahan instalasi jaringan. Permasalahan umum yang sering dialami instansi ini adalah berupa keterbatasan dana yang ada. Tetapi walaupun dengan keterbatasan dana ini instalator tetap menginginkan agar semua titik-titik yang ada dapat terhubung semua dengan jaringan. Permasalahan ini dapat dipresentasikan dalam bentuk graf dengan *vertex* (titik) mewakili lokasi/rumah/komputer dan *edges* (garis) mewakili pipa/kabel/jalan.

Diberikan suatu graf G(V,E) dengan tiap garisnya diberi bobot positif, maka *Minimum Spanning Tree* (MST) adalah suatu *spanning tree* T dari G yang bobotnya paling kecil (*minimum*). *Spanning Tree* T adalah *tree* yang memuat semua titik dari G, dan *tree* adalah graf yang terhubung dan tidak memuat sirkuit. Untuk menyelesaikan MST terdapat dua algoritma yang terkenal yaitu Algoritma Kruskal's (1956) dan Prim (1957).

DCMST (*Degree Constrained Minimum Spanning Tree*) adalah MST yang diberikan kendala *degree* (derajat) pada tiap titiknya.

Dalam penelitian ini maksimal degree (*di*) yang digunakan adalah $di \leq 3$ (dimana nilai i = 1, 2, ..., n). Alasan dibatasi dengan 3 *degree* adalah karena jika nilai $di > 4$ maka hal ini disebut juga sebagai permasalahan MST. Selain itu alasan diberikan $di \leq 3$ bukan $di \leq 2$ karena ketika kondisi ini diberikan, jaringan yang terbentuk akan tereduksi menjadi TSP (*Traveling Salesman Problem*). Penelitian Garey dan Johnson (1979) menunjukkan bahwa permasalahan DCMST adalah masalah yang masuk dalam kelas (*NP*-

Complete) dengan mereduksinya ke TSP. TSP adalah permasalahan dimana seorang salesman yang ingin mengunjungi sebuah kota dari titik pertama dapat kembali ke titik tempat dia bermula. Wamiliana dkk (2004), memperkenalkan metode *Tabu Search* dan juga *modified penalty* untuk menyelesaikan masalah DCMST. Hasil yang didapat menggunakan metode ini adalah *Tabu Search Heuristic* dapat menghasilkan solusi yang baik serta dengan *modified penalty,* dibuktikan bahwa *heuristic* yang dikembangkan lebih layak dari metode lain.

Instalasi jaringan multi tahap atau yang biasa disebut dengan MPDCMST (*Multi Period Degree Constrained Minimum Spanning Tree**)*** adalah DCMST yang diberi kendala tahap. Kawarta (2002) menginvestigasi permasalahan ini dan menyelesaikan dengan menggunakan metode *hybrid* relaksasi Lagrange dan *heuristic branch exchange* dan dilakukan implementasi hingga 100 *vertex* tetapi masih terdapat solusi *infeasible* (tidak layak). Junaidi dkk (2008), menggunakan 3 tahap, $di \leq 3$, serta memodifikasi algoritma greedy pada proses sorting edges. Junaidi dkk (2008), menguji metodenya dengan menggunakan data yang sama dengan Wamiliana dkk (2005) dan data yang diambil dari *Traveling Salesman Problem Library* (TSPLIB). TSPLIB adalah *library* (perpustakaan) TSP. Dari hasil kesimpulan penelitian, menurut Junaidi dkk (2008), masih ada metode yang lebih layak lagi untuk diinvestigasi. Hybrid antara greedy dan teknik DFS juga telah dikembangkan oleh Wamiliana dkk. (2005, 2010, 2013a, 2013b, 2015a, 2015b). Dari penelitian ini didapat hasil bahwa penggunaan teknik ini dapat meningkatkan

kelayakan hasil dari permasalahan yang diberikan, namun dalam pemecahan masalahnya masih terdapat *forest* pada jaringan karena metode yang digunakan menggunakan modifikasi dari algoritma Kruskal.

Pada penelitian ini difokuskan untuk menyelesaikan masalah MPDCMST dengan menggunakan algoritma *modified* Prim dan *penalty* serta pembatasan $di \leq 3$ agar dapat mengatasi permasalahan *forest* dan TSP. Karena penggunaan algoritma Kruskal masih memiliki beberapa kendala masalah seperti *forest* (Wamiliana, 2013b), maka digunakan algoritma Prim agar dapat menangani permasalahan ini. *Modified penalty* digunakan untuk mengatasi masalah kendala *degree* yang dihadapi oleh garis dengan bobot kecil tapi jika diinstal melanggar kendala *degree*.

1.2.Rumusan Masalah

Fokus masalah yang dihadapi pada penelitian ini adalah bagaimana menyelesaikan masalah MPDCMST (*Multi Period Degree Constrained Minimum Spanning Tree)* yang memiliki sejumlah *vertex* dan setiap *vertex* hanya memiliki *degree* maksimal 3 menggunakan algoritma Prim dan *modified penalty.*

1.3.Batasan Masalah

Batasan masalah pada penelitian ini adalah:

1. Algoritma yang digunakan adalah algoritma Prim dan *modified penalty*.
2. Data yang digunakan adalah data penelitian sebelumnya oleh Wamiliana dkk. (2005).

1.4.Tujuan

Tujuan dari dilakukan penelitian ini adalah :

1. Membuat algoritma untuk menyelesaikan masalah *Multi Period Degree Constrained Minimum Spanning Tree* pada instalasi jaringan menggunakan algoritma *modified* Prim dan *modified penalty.*
2. Membuat *source code* berdasarkan tujuan pertama.
3. Mengimplementasi *source code* menggunakan data yang ada di literatur.

1.5.Manfaat

Manfaat yang didapat dari penelitian ini adalah :

1. Memberikan solusi untuk mencari pemecahan masalah MPDCMST dalam instalasi jaringan dengan menggunakan *modified penalty* dan algoritma Prim.
2. Dengan *source code* yang dibuat menjadi program, pemecahan masalah instalasi jaringan semakin dipermudah karena user hanya perlu menginputkan data.

BAB II

TINJAUAN PUSTAKA

Pada bab ini diberikan definisi dan istilah yang digunakan pada penelitian ini.

2.1 Konsep Dasar Graf

Berikut ini adalah konsep dasar graf :

Graf

Graf G didefinisikan sebagai pasangan himpunan (V, E), ditulis dengan notasi G = (V, E). Dalam hal ini, V merupakan himpunan tidak kosong dari titik-titik (*vertex*) digambarkan dalam titik-titik, dan E adalah himpunan sisi-sisi (*edge*) digambarkan dalam garis-garis yang menghubungkan sepasang titik (Munir, 2009).

***Degree* (Derajat)**

Misalkan v adalah titik suatu graf G, derajat titik v (simbol d(v)) adalah jumlah garis yang berhubungan dengan titik v dan garis suatu *loop* dihitung dua kali (Wibisono, 2004).

Path

Path dengan panjang n dari v ke w adalah *walk* dari v ke w yang semua garisnya berbeda. *Path* dari v ke w dituliskan sebagai $v = v_0\, e_1\, v_1\, e_2\, v_2 \ldots e_n\, v_n = w$ dengan $e_i \neq e_j$ untuk $i \neq j$ (Wibisono, 2004).

Sirkuit

Sirkuit dengan panjang n adalah *path* yang dimulai dan diakhiri pada titik yang sama. Sirkuit adalah *path* yang berbentuk $v_0\, e_1\, v_1\, e_2\, v_2 \ldots v_{n-1}\, e_n\, v_n$ dengan $v_0 = v_n$ (Wibosono, 2004).

Graf Terhubung

Graf G disebut graf terhubung jika untuk setiap pasang titik u dan v di dalam himpunan V terdapat *path* dari u ke v. Jika tidak, maka graf G disebut graf tak terhubung. Keterhubungan dua buah titik adalah penting di dalam graf. Jika dua buah titik terhubung maka pasti titik yang pertama dapat dicapai dari titik yang kedua. Pada graf berarah, graf G dikatakan terhubung jika graf tak-berarahnya terhubung (graf tak-berarah dari G diperoleh dengan menghilangkan arahnya). Keterhubungan dua buah titik pada graf berarah dibedakan menjadi terhubung kuat dan terhubung lemah. Dua titik, u dan v pada graf berarah G disebut terhubung kuat jika terdapat lintasan garis dari u ke v dan juga sebaliknya garis berarah dari v ke u. Jika u dan v tidak terhubung kuat tetapi tetap terhubung pada graf tak-berarahnya, maka u dan v dikatakan terhubung lemah (Munir, 2009).

Graf Berbobot

Graf berbobot adalah graf yang setiap garisnya diberi suatu nilai. Bobot pada tiap garis dapat berbeda-beda bergantung pada masalah yang dimodelkan dengan graf. Bobot dapat menyatakan jarak antara dua tiang listrik, kapasitas, biaya perjalanan antara dua kota, waktu tempuh pesan (message) dari sebuah titik komunikasi ke titik komunikasi lain, ongkos dan produksi (Munir, 2009).

Pohon (*Tree)*

Pohon (*Tree*) didefinisikan sebagai graph terhubung yang tidak mengandung sirkuit, sedangkan hutan (*forest*) adalah graph yang tidak mengandung sirkuit. Jadi pohon adalah hutan yang terhubung.

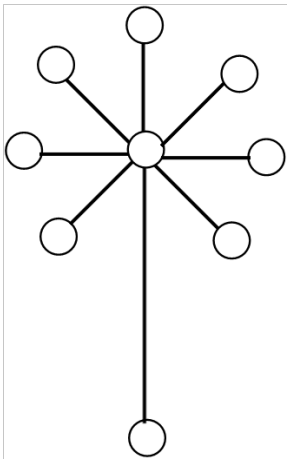
Terdapat 2 syarat dalam suatu *tree* :

1) Suatu graf G disebut terhubung apabila untuk setiap dua titik dari graf G selalu terdapat jalur yang menghubungkan kedua titik tersebut.

2) Sirkuit atau *cycle* adalah suatu lintasan tertutup dengan derajat setiap titik dua.

(Astuti, 2006).

Contoh dari *Tree* digambarkan pada Gambar 2.1.



Gambar 2.1 Contoh *Tree*

Suatu Graf G dengan n buah titik adalah pohon jika :

1) G terhubung dan tak mengandung sirkuit, atau

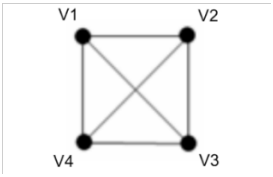
2) G tidak mengandung sirkuit dan mempunyai n - 1 buah garis, atau

3) G mempunyai n - 1 buah garis dan terhubung

Pohon Rentangan (*Spanning Tree*)

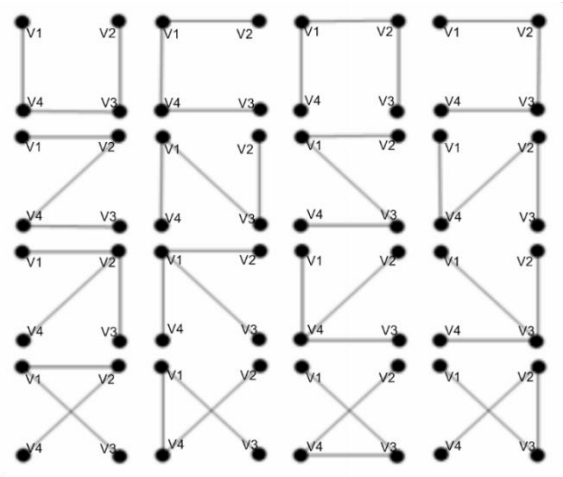
Suatu pohon rentangan atau *spanning tree* adalah suatu subgraf dari graf G yang mengandung semua simpul dari G dan merupakan suatu pohon (Astuti, 2006).

Diberikan graf G sebagai berikut.



Gambar 2.2 Graf G

Skema *Spanning Tree* dari graf G pada Gambar 2.2 adalah sebagai berikut :



Gambar 2.3 Semua *spanning tree* dari graf G

2.2 Pohon Rentang Minimal (*Minimum Spanning Tree*)

Apabila G suatu graf berbobot, maka pohon rentangan minimal dari graf adalah pohon rentangan dengan jumlah bobot terkecil. Untuk mendapatkan pohon rentangan minimal dapat digunakan algoritma berikut :

2.2.1. Algoritma KRUSKAL

Pada algoritma Kruskal, garis-garis graf diurut terlebih dahulu berdasarkan bobotnya yang menaik (dari kecil ke besar). Garis-garis yang dimasukkan ke dalam himpunan T adalah graf G sedemikian sehingga T adalah pohon. Pada keadaan awal, garis-garis yang sudah diurut berdasarkan bobot membentuk hutan (*forest*), masing-masing pohon di hutan hanya berupa satu titik. Hutan tersebut dinamakan pohon rentang (*spanning forest*), atau hutan berentang. Garis dari graf G ditambahkan ke T jika ia tidak membentuk siklus di T.

Langkah-langkah algortima Kruskal :

Langkah 0 : Garis-garis graf sudah diurut menaik berdasrkan bobotnya – dari bobot kecil ke bobot besar.

Langkah 1 : T masih kosong.

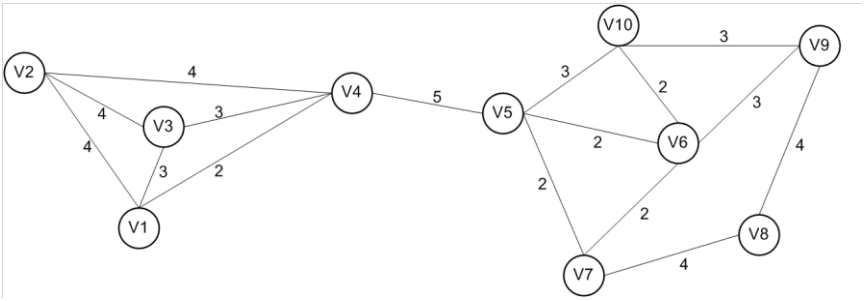
Langkah 2 : pilih garis (u, v) dengan bobot minimum yang tidak membentuk sirkuit di T. Tambahkan (u, v) ke dalam T.

Langkah 3 : ulangi langkah 2 sebanyak n-1.

(Munir, 2009).

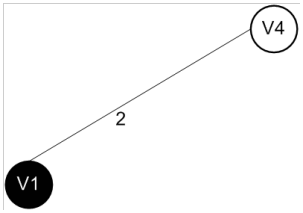
Contoh :

Diberikan graf seperti pada Gambar 2.4

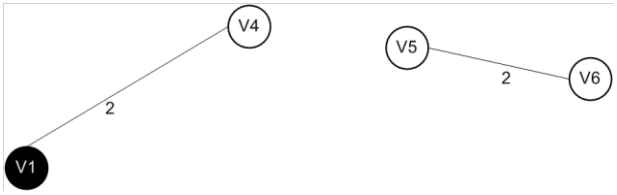


Gambar 2.4 Contoh graf untuk penentuan MST

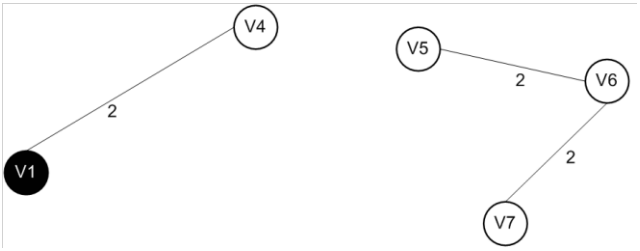
Tahap-tahap penentuan MST graf pada Gambar 2.4 menggunakan algoritma Kruskal dalam setiap tahapan dari (a)-(i) sehingga menjadi Gambar 2.5 :



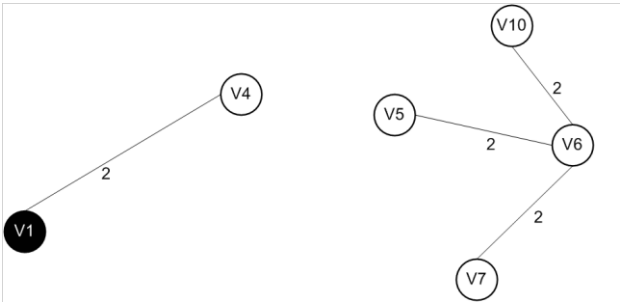
(a) Tahap 1



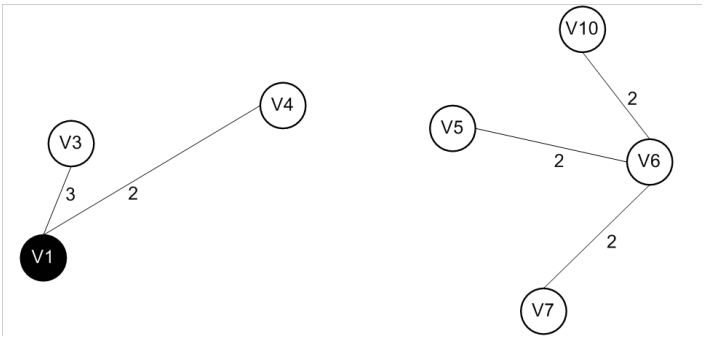
(b) Tahap 2



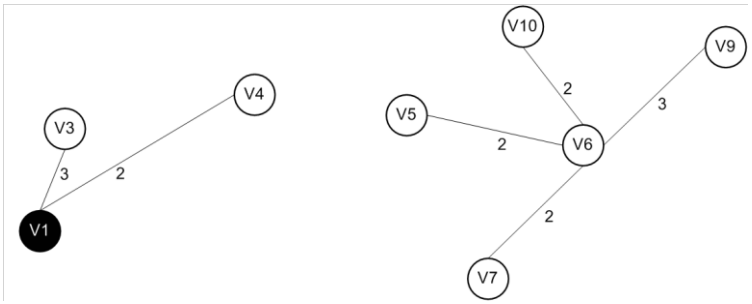
(c) Tahap 3



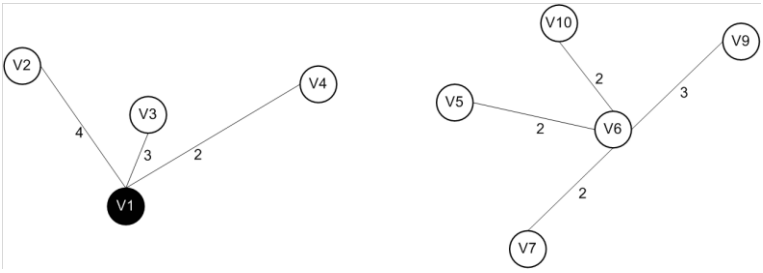
(d) Tahap 4



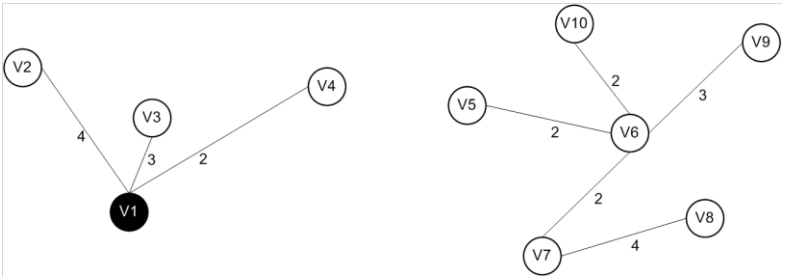
(e) Tahap 5



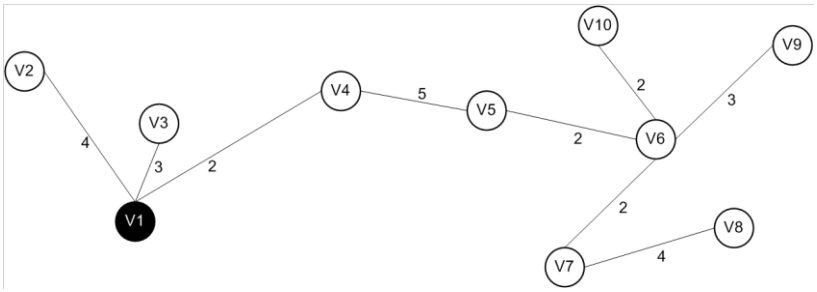
(f) Tahap 6



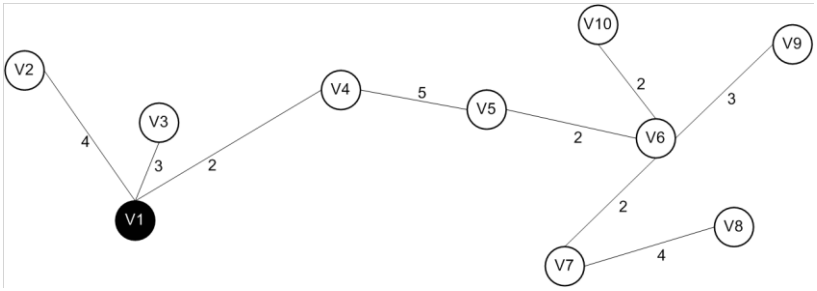
(g) Tahap 7



(h) Tahap 8



(i) Tahap 9



Gambar 2.5 MST dari graf G pada Gambar 2.4 menggunakan algoritma Kruskal

2.2.2. Algoritma PRIM

Untuk mencari pola rentang minimum T dari graf G dengan algoritma Prim mula-mula dipilih satu titik sembarang (misal v1). Kemudian ditambahkan satu garis berhubungan dengan v1 dengan bobot yang paling minimum (misal e1) dan ujung lainnya ke T sehingga T terdiri dari garis e1 dan 2 buah titik-titik ujung garis e1 (salah satunya adalah v1). Pada setiap langkah selanjutnya dipilih sebuah garis dalam E(G) yang bukan anggota E(T) dengan sifat :

a. Garis tersebut berhubungan dengan salah satu titik €V(T).
b. Garis tersebut mempunyai bobot yang paling kecil.
c. Langkah tersebut diulang-ulang hingga memperoleh (n-1) garis dalam E(T) (n adalah jumlah titik dalam G).

Secara formal algoritma Prim adalah sebagai berikut :

0. Inisialisasi : mula-mula T adalah graf kosong.
1. Ambil sembarang v € V(G). Masukkan V kedalam V(T).
2. V(G) = V(G) – {v}.
3. Untuk i = 1,2, ... , n-1 lakukan :
    a. Pilihlah garis e € E(G) dan e € E(T) dengan syarat :

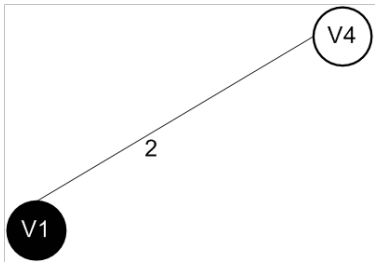
- E terhubung dengan satu titik dalam T dan tidak membentuk sirkuit
- E mempunyai bobot terkecil dibandingkan dengan semua garis yang terhubung dengan titik-titik dalam T. Misalkan w adalah titik ujung e yang tidak berada dalam T.

b. Tambah e kedalam E(T) dan w ke V(T).

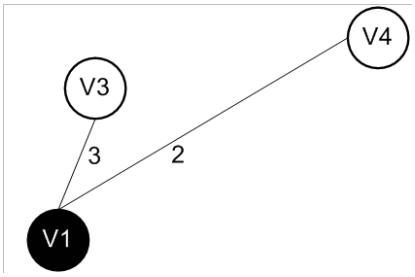
(Wibisono, 2004).

Contoh :

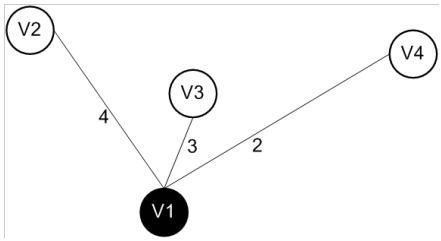
Diberikan graf seperti pada Gambar 2.4 sebelumnya. Tahap-tahap penentuan MST pada graf di Gambar 2.4 menggunakan algoritma Prim dalam setiap tahapan dari (a)-(i) sehingga menjadi Gambar 2.7 :



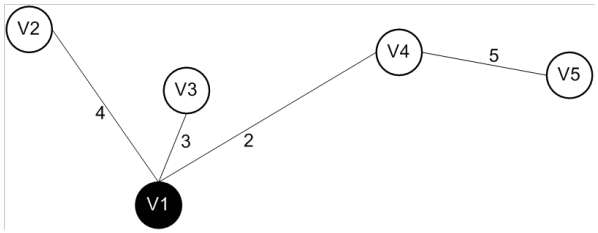
(a) Tahap 1



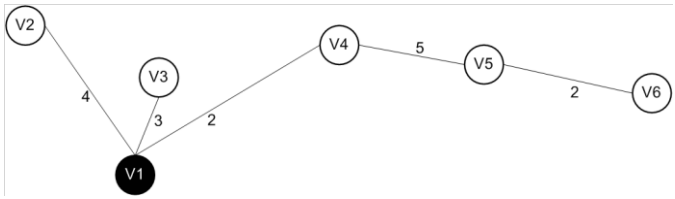
(b) Tahap 2



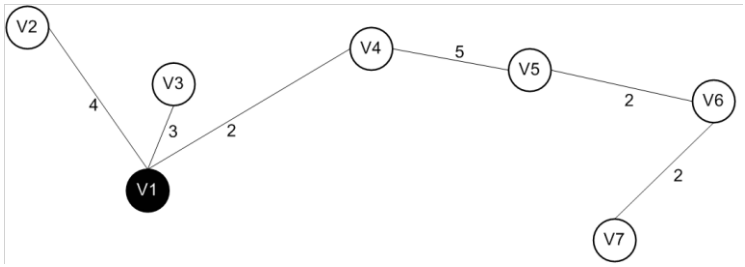
(c) Tahap 3



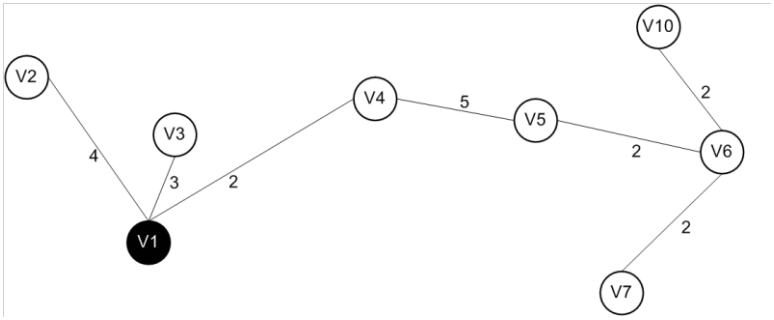
(d) Tahap 4



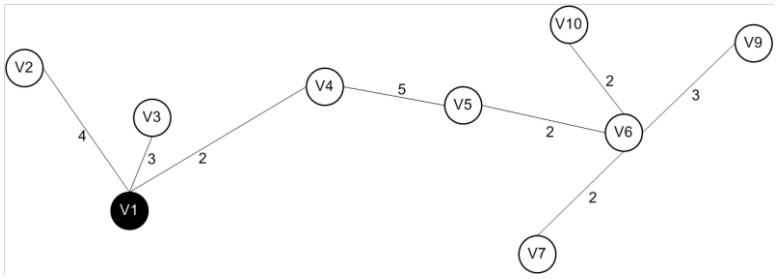
(e) Tahap 5



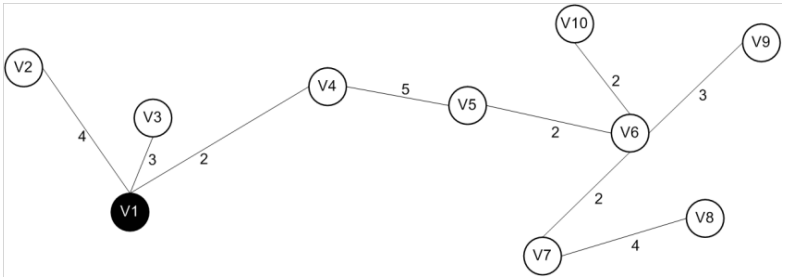
(f) Tahap 6



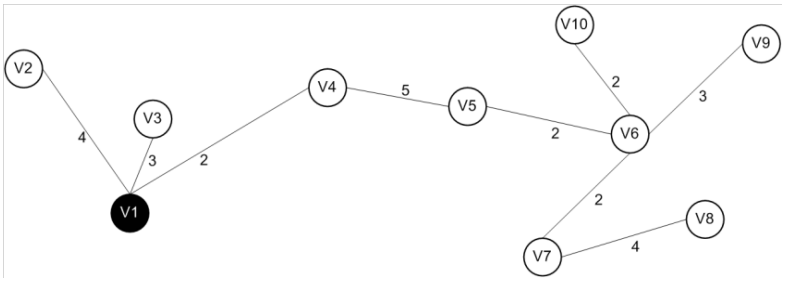
(g) Tahap 7



(h) Tahap 8



(i) Tahap 9



Gambar 2.6 MST dari graf G pada Gambar 2.4 menggunakan algoritma Prim

2.3 MST dan Beberapa Turunannya

DCMST (*Degree Constrained Minimum Spanning Tree*)

DCMST adalah MST yang diberikan kendala *degree* pada setiap titiknya. Permasalahan DCMST adalah untuk menentukan *minimum spanning tree* T dari G dimana *degree* dari *vertex* dibatasi dengan persamaan, $1 \leq bi \leq n$ (Caccetta dan Wamiliana, 2001).

MPDCMST (*Multi Period Degree Constrained Minimum Spanning Tree*)

Multi Period Degree Constrained Minimum Spanning Tree (MPDCMST) adalah masalah lanjutan dari *Degree Constrained Minimum Spanning Tree* (DCMST). Perbedaaan di antara kedua masalah ini adalah periode saat dilakukannya instalasi jaringan. Biasanya dalam implementasi dikehidupan sehari-hari, instalasi jaringan harus dilakukan dalam beberapa periode karena mengalamai kendala kekurangan dana saat instalasi. Pada DCMST, semua *vertex* (dapat dipresentasikan sebagai komputer/kota) dapat diinstal dalam satu tahap. Masalah ini diaplikasikan dalam permasalahan kehidupan sehari-hari seperti instalasi, suplai air, suplai gas, listrik, kabel dan telepon .

2.4 *Java Language*

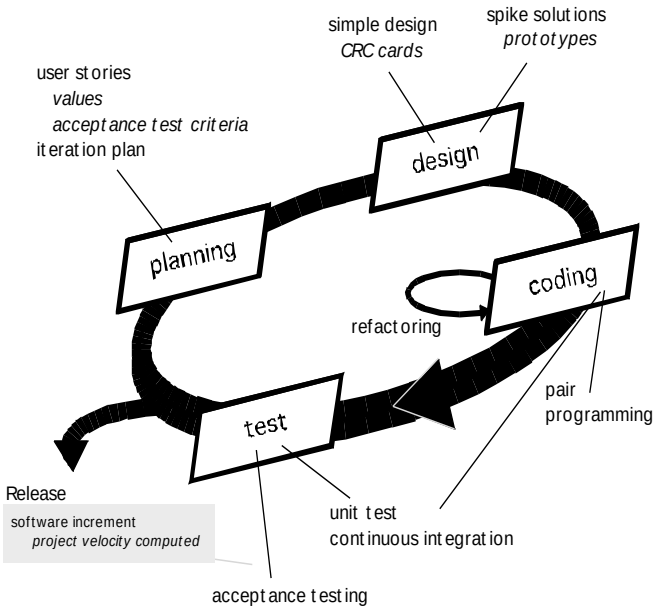
Java diciptakan oleh James Gosling, Patrick Naughton, Chris Warth, Ed Frank, dan Mike Sheridan di *Sun Microsystem* pada tahun 1991. Bahasa ini awalnya disebut *Oak* namun kemudian diganti *Java* pada tahun 1995. Dorongan asli dibentuknya *Java* tidak berasal internet, sebaliknya, motivasi utama diciptakannya bahasa ini adalah kebutuhan untuk bahasa platform-independen yang dapat digunakan untuk membuat perangkat lunak untuk dimasukkan dalam berbagai perangkat elektronik konsumen (Oracle, 2016).

2.5 *Extreme Programming*

Extreme Programming, atau XP, adalah metode ringan pengembangan perangkat lunak berdasarkan nilai-nilai kesederhanaan, komunikasi, umpan balik, dan keberanian. XP dikembangkan oleh Kent Beck pada tahun 1996

yang menulis buku, *Extreme Programming Explained*. XP dirancang untuk digunakan dengan tim kecil yang membutuhkan untuk mengembangkan perangkat lunak dengan cepat dengan lingkungan yang mudah cepat berubah. XP bekerja dengan membawa keseluruhan tim bersama-sama di hadapan praktek sederhana, dengan umpan balik yang cukup untuk memungkinkan tim dapat melihat di mana mereka berada dan untuk menyesuaikan praktek sesuai dengan situasi unik mereka (Lamia, 2002).

Gambar 2.6 adalah alur penelitian *Extreme Programming*.



Gambar 2.7 Alur Metode *Xtreme Programming* (Pressman, 2009)

BAB III

METODE PENELITIAN

3.1 Tempat dan Waktu Penelitian

Penelitian dilakukan di Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung. Waktu penelitian dilakukan pada semester ganjil tahun ajaran 2016-2017.

3.2 Bahan dan Alat

Bahan penelitian yang dibutuhkan untuk mendukung penelitian ini adalah data masalah dari berbagai literatur mengenai algoritma Prim dan *modified penalty* beserta pengaplikasiannya dalam kehidupan sehari-hari.

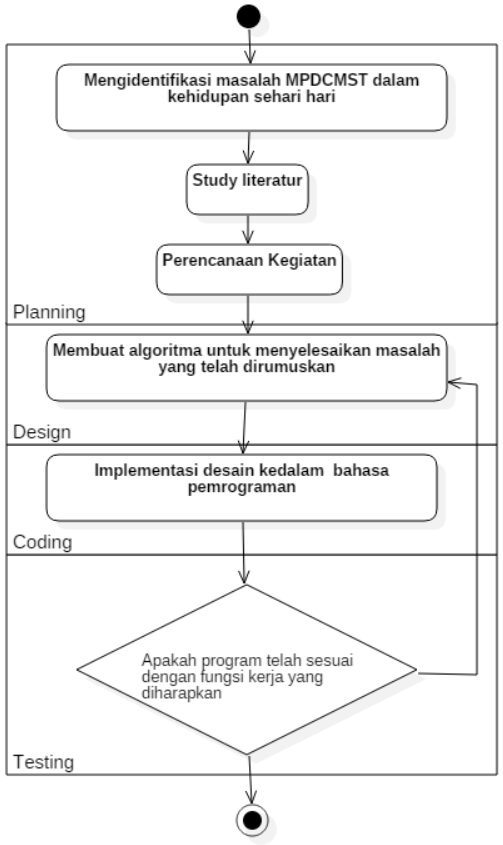
Dalam pengerjaan penelitian ini, alat yang digunakan untuk mendukung penelitian itu sendiri adalah sebagai berikut :

1. Perangkat keras (*Hardware*) Laptop, dengan spesifikasi :
    a. *Processor* Core I3-4005U
    b. RAM 6GB
    c. *Hardisk* 500GB
    d. Mouse

2. Perangkat lunak (*Software*), yang meliputi :
    a. Sistem Operasi : Windows 7 Ultimate
    b. Bahasa pemograman : *Java*
    c. Aplikasi tool pemograman : netbeans 8.0
    d. Program *utilitas* : sublimetext, notepad++

3.3 Tahapan Penelitian

Pelaksanaan penelitian ini dilakukan dengan menggunakan metode *Extreme Programming*, dimana dalam pelaksanaannya terdapat 4 tahap utama. Penjelasan alur metode *Extreme Programming* diberikan seperti pada Gambar 3.1.



Gambar 3.1 Alur Metode Penelitian

Berikut ini adalah penjelasan dari alur metode penelitian yang digunakan :

1. *Planning*

Pada tahap ini dilakukan identifikasi masalah MPDCMST (*Multi Period Degree Constrained Minimum Spanning Tree*) dalam kehidupan sehari-hari. Identifikasi masalah pada penelitian ini adalah mengenai penyelesaian masalah instalasi jaringan multi tahap menggunakan hybrid algoritma Prim dan *modified penalty*. Penyelesaian masalah ini bertujuan untuk mendapatkan jarak atau biaya minimum pada setiap tahap instalasi dengan melalui 3 tahap dan setiap *vertex* dibatasi dengan degree ≤ 3. Selanjutnya studi literatur dilakukan, yakni mengumpulkan informasi mengenai algoritma Prim dan modified penalty serta penerapannya dalam kehidupan sehari-hari. Pada penelitian ini beberapa literatur merujuk terhadap paper-paper internasional yang sebelumnya pernah melakukan penelitian yang berkaitan tentang masalah ini.

2. *Design*

Tahap kedua adalah proses pembuatan algoritma program untuk penyelesaian masalah yang telah dirumuskan. Algoritma yang akan dibuat adalah hasil pengembangan dari algoritma Prim. Pada tahap ini juga dilakukan perancangan *input*, *output* dan *flowchart* dari program yang akan dibuat. Berikut ini diberikan *pseudo code* algoritma *modified* Prim dan *modified penalty* beserta penjelasannya untuk menyelesaikan masalah MPDCMST.

```
inisialisasi
  V={1}, T=0, n=jumlah_vertex, k=1, kMax=3
begin
  while k < kMax
  do
    if |HVTk| > MaxVTk
      stop
    else
      Tk = 0
      while Tk < (|HVTk|-1)
      do
        cari edge terpendek dari vertex yang terhubung dalam V
        simpan edge terpilih kedalam temp (T)
        if vertex yang akan dihubungkan tidak termasuk dalam HVTk
          lanjutkan ke edge berikutnya
        else
          if penambahan e menyebabkan sirkuit
            lanjutkan ke edge berikutnya
          else
            if penambahan e melanggar degree restriction
              lanjutkan ke edge berikutnya
            else
              simpan vertex sumber dari e kedalam V
              simpan e kedalam T
              Tk++
            endif
          endif
        endif
      end
```

```
      while Tk < (MaxVTk-1)
      do
        cari edge terpendek dari vertex yang terhubung dalam V
        simpan edge terpilih kedalam temp (T)
        if penambahan e menyebabkan sirkuit
          lanjutkan ke edge berikutnya
        else
          if penambahan e melanggar degree restriction
            lanjutkan ke edge berikutnya
          else
            simpan vertex sumber dari e kedalam V
            simpan e kedalam T
            Tk++
          endif
        endif
      end
      k++;
    endif
  endwhile
end
```

Catatan :

- HVTk = Himpunan *vertex-vertex* yang harus diinstal pada tahap ke-k atau sebelumnya.
- maxVTk = Maksimal *vertex* yang dapat diinstal pada tahap ke-k.
- kMax = Jumlah maksimal periode instalasi.
- V = Himpunan *vertex* yang telah terhubung.

- T = Himpunan *edge* dalam *graf*.
- Tk = Jumlah *edge* terinstal pada tahapan ke-k.
- k = Periode instalasi sekarang.

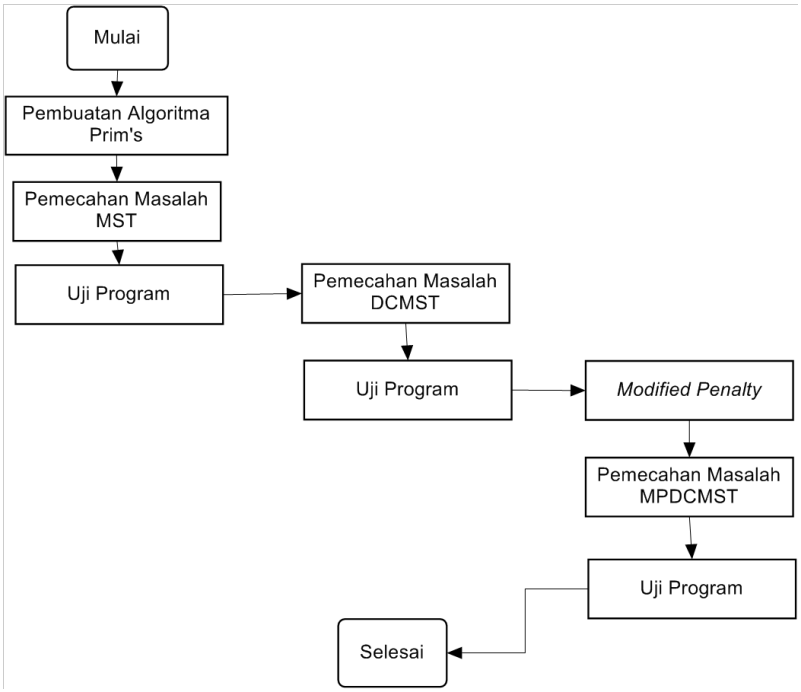
Penjelasan *pseudo code* algoritma *modified* Prim :

1. Inisialisasi variabel V sebagai himpunan *vertex* dengan nilai awal berisi *vertex* 1. Inisialisasi variabel T sebagai himpunan kosong yang digunakan untuk menyimpan *edge*. Inisialisasi variabel n sebagai jumlah *vertex* yang diinstal. Inisialisasi variabel k dengan 1 dan kMax dengan 3.
2. Jika jumlah *vertex* yang harus diinstal pada tahap sekarang (|HVTk|) lebih besar dari jumlah vertex yang dapat diinstal pada tahap sekarang, maka muncul pesan *error* dan hentikan program.
3. Cari dan simpan *edge* terpendek dari *vertex* dalam V ke variabel e.
4. Jika *vertex* yang terhubung tidak termasuk dalam *vertex* prioritas maka ke *edge* selanjutnya.
5. Jika penambahan e menyebabkan sirkuit, maka ke *edge* selanjutnya.
6. Jika penambahan e melanggar *degree restriction*, lakukan uji nilai e dengan e yang telah terinstal, jika nilai e.baru < e.terinstal lakukan *update* nilai e pada V dan T, jika e.baru > e.terinstal maka ke *edge* selanjutnya.
7. Simpan *vertex* sumber dari e kedalam V.
8. Simpan e kedalam T

9. Jika jumlah *edge* dalam T (|T|) kurang dari jumlah *vertex* prioritas pada periode ini (|HVTk|) dikurang satu, maka kembali ke tahap 3.
10. Cari dan simpan edge terpendek dari *vertex* dalam V ke variabel e
11. Jika penambahan e menyebabkan sirkuit, maka lanjutkan ke *edge* selanjutnya.
12. Jika penambahan e melanggar *degree restriction*, lakukan uji nilai e dengan e yang telah terinstal, jika nilai e.baru < e.terinstal *update* nilai e pada V dan T, jika e.baru > e.terinstal maka lanjutkan ke *edge* selanjutnya.
13. Simpan *vertex* sumber dari e kedalam V.
14. Simpan e kedalam T
15. Jika jumlah edge dalam T (|T|) kurang dari jumlah *vertex* maksimal yang dapat diinstal pada periode ini (|MaxVTk|) dikurang satu, maka (kembali) ke tahap 10.
16. Ubah nilai k menjadi nilai (k+1).
17. Jika jumlah periode sekarang (k) kurang dari jumlah periode maksimal (kMax), maka kembali ke tahap 2.
18. Selesai

3. *Coding*

Pada tahap ini terdapat alur *flowchart* yang menunjukkan beberapa tahap. *Flowchart* pada tahap koding digambarkan seperti pada Gambar 3.2.

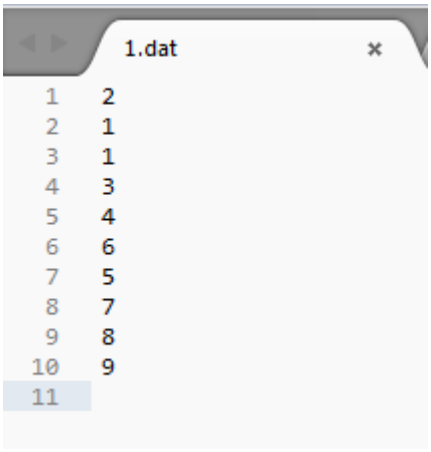


Gambar 3.2 *Flowchart* tahap koding.

Pada tahap koding, dibagi menjadi 3 tahap. Pada pengaplikasian metodenya, algoritma Prim digunakan pada 2 masalah yaitu masalah MST dan DCMST dan *modified penalty* berlaku pada pemecahan masalah MPDCMST. Penjelasan masing-masing tahap adalah sebagai berikut.

a. *Modified* Prim

   Metode ini digunakan untuk menyelesaikan masalah MST dan DCMST. Pada data Wamiliana dkk. (2015a) terdapat 10 folder dimulai dari 10 sampai 100. Pada masing-masing folder terdapat hingga 30 *problem* yang dapat diujicoba. Sebagai sampel untuk memperjelas contoh dari penelitian ini, digunakan satu file dengan jumlah *vertex* = 5, dengan isi file seperti pada Gambar 3.3.



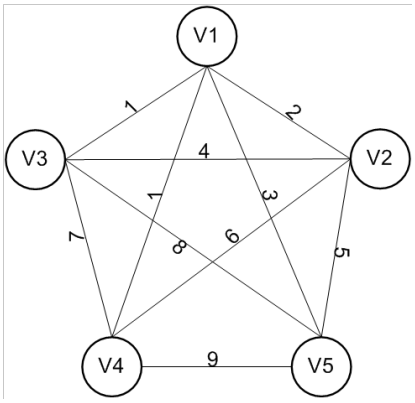
Gambar 3.3 Isi Data Folder 5 File 1.dat

Jika file 1.dat diubah menjadi sebuah nilai *edges* dengan titik-titik koordinat maka didapat hasil seperti Tabel 3.1 berikut.

Tabel 3.1 Tabel Koordinat Edges

***Vertex* Awal**	***Vertex* Tujuan**	**Nilai *Edges***
1	2	2
1	3	1
1	4	1
1	5	3
2	3	4
2	4	6
2	5	5
3	4	7
3	5	8
4	5	9

Agar lebih mudah dipahami, Tabel 3.1 direpresentasikan ke dalam bentuk graf pada Gambar 3.4.



Gambar 3.4 Visualisasi Tabel 3.1

V adalah simbol *vertex* atau titik-titik yang ada. Pada Gambar 3.4 terdapat 5 *vertex* yang disimbolkan dengan V_0-V_4. Garis yang saling menghubungkan dari satu titik ke titik lainnya pada gambar disebut dengan *branch*. Angka-angka yang terdapat pada *branch* adalah jarak atau biaya dari satu titik ke titik lainnya. Gambar 3.5 disebut sebagai graf berbobot, karena setiap titik ke titik lainnya terdapat *branch* yang saling menghubungkan.

a. MST (*Minimum Spanning Tree*)

   Pada permasalahan MST program yang dibuat akan dapat menginputkan jumlah *vertex* yang akan digunakan, dan masalah keberapa yang akan diselesaikan. Tahap-tahap yang dilakukan untuk menyelesaikan MST dengan algoritma Prim ini adalah sebagai berikut.

1. Tentukan *vertex* sumber dengan menginputkan pada program *vertex* keberapa yang akan dijadikan sumber.
2. Lakukan *sorting vertex* terdekat dengan *vertex* sumber atau *vertex* yang telah terinstal.
3. Lakukan pengujian apakah terjadi sirkuit atau tidak jika *vertex* terpendek yang terpilih diinstal.
4. Instal *vertex* terdekat yang sudah dipilih.
5. Lakukan terus tahap 2-4 hingga semua *vertex* terinstal.

Output manual dari file 1.dat dengan jumlah *vertex* 5 dengan sumber *vertex* = V_1 :

E(1,3) = 1

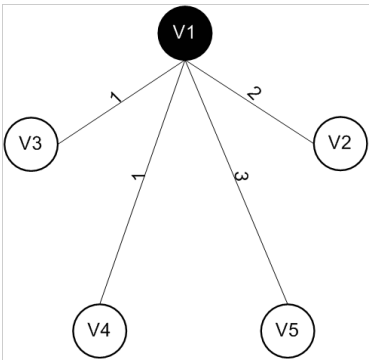
E(1,4) = 1

E(1,2) = 2

E(1,5) = 3

Total *cost* = 7

Output manual divisualisasikan seperti pada Gambar 3.5.



Gambar 3.5 Graf *output solving* MST dengan *root vertex* V_1

Output manual dari file 1.dat dengan jumlah *vertex* 5 dengan sumber *vertex* = V_3 :

E(3,1) = 1

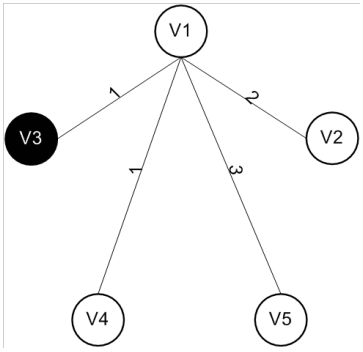
E(1,4) = 1

E(1,2) = 2

E(1,5) = 3

Total *cost* = 7

Output manual divisualisasikan seperti pada Gambar 3.6.



Gambar 3.6 Graf *output solving* MST dengan *root vertex* V_3

Vertex dengan warna hitam adalah *vertex* sumber. Walaupun sumber *vertex* diubah-ubah, hasil dari total *cost* selalu sama karena tidak adanya pembatasan *degree*, berbeda dengan permasalah DCMST selanjutnya.

b. DCMST (*Degree Constrained Minimum Spanning Tree*)

Permasalahan DCMST adalah masalah MST yang deberi kendala batasan *degree*. Tahap-tahap yang dilakukan untuk menyelesaikan DCMST dengan algoritma Prim ini adalah sebagai berikut.

1. V*ertex* sumber pada permasalah DCMST selalu diletakkan pada *vertex* pertama yaitu V_1.
2. Lakukan *sorting vertex* terdekat dengan *vertex* sumber atau *vertex* yang telah terinstal.
3. Lakukan pengujian apakah terjadi sirkuit atau tidak jika *vertex* terpendek yang terpilih diinstal.
4. Lakukan pengujian apakah *degree vertex* sumber yang terhubung dengan *vertex* terpilih ≤ 3 atau ≤ 4 tidak. Jika iya lakukan langkah 5, jika *degree* telah memenuhi batas, maka ulangi mulai langkah 2.
5. Instal *vertex* terdekat yang sudah dipilih.
6. Jika terdapat *vertex* yang belum terinstal lakukan terus tahap 2-4.
7. Selesai.

Output manual dari file 1.dat dengan jumlah *vertex* 5, sumber *vertex* = V_1 dan batasan *degree* ≤ 3 :

E(1,3) = 1

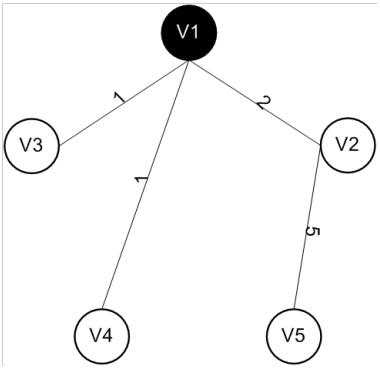
E(1,4) = 1

E(1,2) = 2

E(2,5) = 5

Total Cost = 9

Output manual divisualisasikan seperti pada Gambar 3.7.

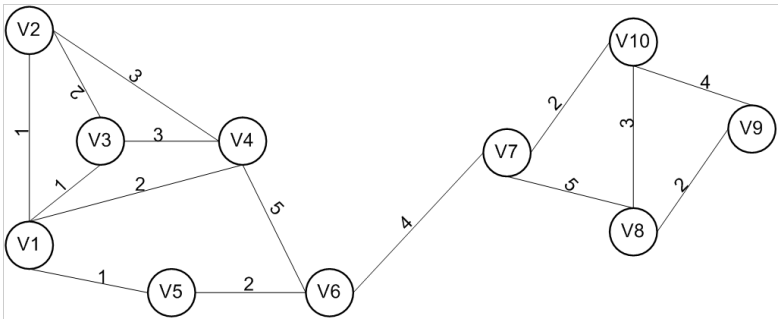


Gambar 3.7 Graf *output solving* DCMST.

b. *Modified Penalty*

Metode *modified penalty* dilakukan untuk menyelesaikan masalah MPDCMST. MPDCMST sendiri adalah permasalahan MST yang diberi kendala batasan *degree* dan instalasi yang dilakukan secara bertahap. Tahap yang dilakukan sama halnya dengan yang telah dijelaskan pada *pseudo code* di tahap *design*. Tahap-tahap ini diterapkan ke program untuk memecahkan masalah MPDCMST dengan data Wamilliana dkk. (2015a).

Diberikan sebuah graf dengan jumlah *vertex* 10 seperti pada Gambar 3.8.



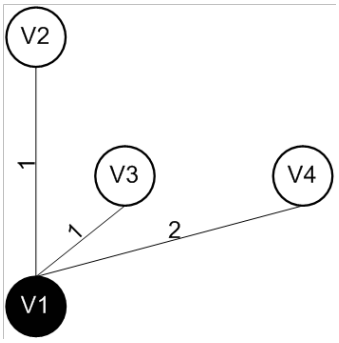
Gambar 3.8 Graf dengan 10 *vertex*

Pada permasalahan MPDCMST ini diberi kendala *degree* ≤ 3. Selanjutnya terdapat 3 tahap instalasi jaringan pada graf. V_1 akan bergungsi sebagai *vertex* sumber. |HVTk| adalah himpunan titik-titik yang harus diinstal pada tahap ke k atau sebelumnya. |HVTk| diinisiasi sebagai berikut :

c. Tahap 1 : *vertex* V_2 dan V_4.
d. Tahap 2 : *vertex* V_5.
e. Tahap 3 : *vertex* V_{10}.

1) Tahap 1

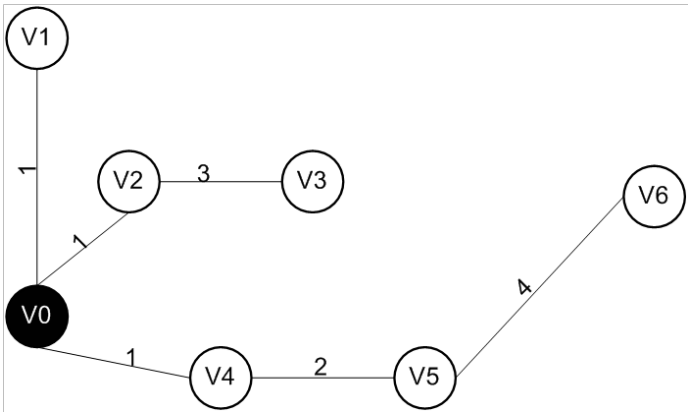
Pada tahap 1, *vertex* yang wajib diinstal adalah V_2 dan V_4. Sesuai dengan metode pemecahan masalah yaitu algoritma Prim, maka cari jarak terdekat dengan *vertex* sumber atau *vertex* yang telah terinstal dengan kendala *degree* ≤ 3, dan tidak diperbolehkannya terjadi sirkuit ataupun *forest*. Sehingga didapat graf seperti pada Gambar 3.9.



Gambar 3.9 Graf hasil tahap 1

2) Tahap 2

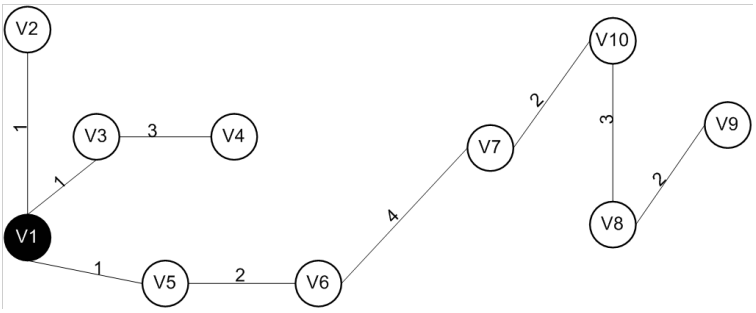
Selanjutnya pada tahap 2, *vertex* yang wajib diinstal adalah V_5. Dapat dilihat dari graf gambar 3.8, bahwa V_5 lebih dekat dengan V_1 dibandingkan dengan V_4, karena batas *degree* V_1 telah memenuhi syarat maka V_5 tidak dapat dikoneksikan dengan V_1 secara langsung, sehingga pada tahap 2 ini, *modified penalty* (pembobotan ulang nilai *edges*) dilakukan. Graf hasil tahap 2 setelah dilakukan *modified penalty* divisualisasikan seperti pada Gambar 3.10.



Gambar 3.10 Graf hasil tahap 2

3) Tahap 3

Pada tahap 3 *vertex* yang diinstal adalah V_{10}. Tahapan atau langkah-langkah yang dilakukan sama halnya dengan tahap 1, dan jika didapati *vertex* yang lebih dekat dengan *vertex* sumber atau *vertex* yang telah terinstal, tetapi syarat *degree* telah terpenuhi maka lakukan pembobotan ulang nilai *edges*. Graf hasil tahap 3 dimana semua *vertex* telah terinstal diberikan pada Gambar 3.1.



Gambar 3.11 Graf hasil tahap 3

3.4 Pengujian

Tahap terakhir yaitu pengujian. Pada tahap ini program yang sudah dibuat diujicoba sepenuhnya dengan menggunakan berbagai data yang ada. Dalam penelitian ini data yang digunakan adalah data dari penelitian Wamiliana dkk. (2015a) sebelumnya, dan terdapat sekitar 300 data. Pada pengujian sistem terdapat rencana pengujian atau skenario pengujian yaitu:

1. Pengujian program dengan kendala MST dengan menggunakan data 10 sampai 100 *vertex.*
2. Pengujian program dengan kendala DCMST dengan menggunakan data 10 sampai 100 *vertex.*

3. Pengujian program dengan kendala MPDCMST dengan menggunakan data 10 sampai 100 *vertex.*
4. Pengujian program dengan kendala MPDCMST dimana penginstalan |HVTk| dilakukan terlebih dahulu dan dilanjutkan dengan *vertex* lain dengan batas |MaxVTk| dengan menggunakan data 10 sampai 100 *vertex.*
5. Pengujian program dengan kendala MPDCMST dimana penginstalan |HVTk| dilakukan setelah |MaxVTk|-|HVTk| *vertex* lain diinstal terlebih dahulu dengan menggunakan data 10 sampai 100 *vertex.*

BAB IV

SIMPULAN DAN SARAN

5.1 Simpulan

Berdasarkan hasil penelitian yang telah dilakukan dapat diambil kesimpulan sebagai berikut :

1. Masalah *Multi Period Degree Constrained Minimum Spanning Tree* pada instalasi jaringan dapat diselesaikan dengan menggunakan algoritma *modified* Prim dan *modified penalty*.
2. Dengan penerapan algoritma ke *source code* untuk menyelesaikan masalah *Multi Period Degree Constrained Minimum Spanning Tree*, instalasi jaringan multi tahap dapat diselesaikan dengan cepat dan hasil yang lebih baik.
3. Setelah dilakukan pengujian algoritma yang telah dibuat (WAK1, WAK2, WAK3) dengan data Wamiliana dkk. (2005) didapat hasil bahwa algoritma WAK1, WAK2, WAK3 lebih baik dari pada hasil algoritma WADR1, WADR2 oleh Wamiliana dkk. (2015a).

5.2 Saran

Dari hasil penelitian yang dilakukan terdapat beberapa saran sebagai berikut :

1. Algoritma WAK masih dapat dikembangkan dengan menggabungkan algoritma Kruskal kedalamnya, dengan harapan hasil yang didapat akan lebih baik.
2. Visualisasi dapat ditampilkan dengan menambah angka *edge* sehingga lebih mudah dipahami.

DAFTAR PUSTAKA

Astuti, Yuni Dwi. 2006. *Logika dan Algoritma BAB 3*. Jakarta : Universitas Gunadarma.

Caccetta, L., dan Wamiliana, 2001. Heuristics Algorithms for Defree Constrained Minimum Spanning Tree Problems, in Proceeding of the International Congress on Modelling and Simulation (MODSIM 2001), *Canberra*, Editors: F.Ghassemi *et al.*, pp. 2161-2166,2001.

Cacceta, L., dan Wamiliana, 2004. A First level Tabu Search Heuristic for Degree Constrained Minimum Spanning Tree problem. *Journal of Quantitive Methods*, Vol.1 No.1, pp. 20-32.

Garey, M.R., D.S. Jhonson, *Computer and Intractibility, A Guide to the Theory of NP-Completeness*. Freemann, San Frasisco, 1979.

Junaidi A., Wamiliana, Dwi Sakethi, and Edy Tri Baskoro, 2008. 'Computational Aspects for multiperiod degree constrained minimum spanning tree problem, *Jurnal Sains MIPA*, Special Edition, January 2008, pp. 1-5.

Kawarta R., 2002. 'A Multiperiod degree constrained minimum spanning tree problem', *European Journal of Operational Reasearch*, Vol. 143, pp. 53-63.

Kruskal, J.B., On the Shortest Spanning Tree of a Graph and the Travelling Salesman Problem. *Proc.Amer.Math.Soc.* 7, 48-50, 1956.

Lamia dkk. 2002. *Extreme Programming*. *Software Enginering CSS 423 B – MWF 11-12.*

Liu, C.L. 1995. *Dasar-dasar Matematika Diskrit.* Jakarta : Penertbit PT Gramedia Pustaka Utama.

Munir, Rinaldi. 2009. *Matematika Diskrit*. Edisi Ketiga. Bandung:Informatika.

Oracle. 2016. *Java Fundamentals BAB 1. Java*. New York

Pressman, Roges S. 2010. *'Software Enginering'.* New York. McGraw-Hill Company.

Prim, R.C., Shortest Connection Networks and Some Generalizations. *Bell System Technical Journal*, 36, 1389-1401, 1957.

Wamiliana, 2002. The Modified Penalty Methods for The Degree Constrained Minimum Spanning Tree Problem, *Jurnal Sains dan Teknologi*, **Vol,** pp. 1-2

Wamiliana and L. Caccetta, 2004. "Tabu Search Based Heuristics for the Degree Constrained Minimum Spanning Tree Problem", Proceeding of South East Asia Mathematical Society Conference, pp. 133-140

Wamiliana, 2004. 'Solving the Degree Constrained Minimum Spanning Tree Using Tabu and Penalty Method', *Jurnal Teknik Industri,* **Vol.6** No.1, pp. 1-9

Wamiliana, Dwi Sakethi, Akmal J., and Edy Tri Baskoro, 2005. 'The design of greedy algorithm for solving the multiperiod degree constrained minimum spanning tree problem', *Jurnal Sains dan Teknologi,* **Vol.11** No.2, pp. 93-96

Wamiliana, Dwi Sakethi, and Restu Yuniarti, 2010. 'Computational Aspects of WARD1 and WARD2 Algorithm for The Multi Period Degree Constrained Minimum Spanning Tree Problem, Prosiding Seminar Nasional MIPA dan Aplikasinya (SNMAP), 2010, hal. 208-214

Wamiliana, dan Louis Caccetta, 2013a. The Modified CW1 Algorithm For The Degree Restricted Minimum Spanning Tree Problem. *International Journal of Engineering and Technology Development*, Vol.1, No.2, *August* 2013.

Wamiliana, Amanto, and Mustofa Usman. 2013b. "Comparative Analysis for The Multi Period Degree Constrained Minimum Spanning Tree Problem", in *Proc. International Conference on Engineering and Technology Development* (ICETD), 2013, pp. 39-43.

Wamiliana, Faiz A.M. Elfaki, Mustofa Usman, and M. Azram, 2015a. Some Greedy Based Algorithms for Multi Period Degree Constrained Minimum Spanning Tree Problem. *ARPN Journal of Engineering and Applied Science,* **Vol.10** Issue 21, pp.10147-10152

Wamiliana, Mustofa Usman, Dwi Sakethi, Restu Yuniarti and Ahmad Cucus, 2015b. The Hybrid of Depth First Search Technique and Kruskal's Algorithm for Solving The Multi Period Degree Constrained Minimum Spanning Tree Problem. *IEEE Explore, 2016.* http://ieeexplore.ieee.org/xpl/articleDetails.jps?arnumber=7516333&newsearch=true&queryText=wamiliana

Wibisono, Samuel. 2004. *Matematika Diskrit.* Yogyakarta : Penerbit Graha Ilmu.

LAMPIRAN

Jumlah Vertek = 10

HVTk = {1,2,3}

No. Problem	MST	DCMST	WAK1	WAK2	WAK3
1.dat	900	900	900	1207	900
2.dat	923	1076	1076	1521	1076
3.dat	746	746	929	1030	929
4.dat	858	858	963	968	963
5.dat	1248	1259	1465	1621	1465
6.dat	1347	1401	1401	1535	1401
7.dat	1912	1912	1912	2253	1912
8.dat	466	466	466	466	466
9.dat	1478	1511	1511	1884	1511
10.dat	1519	1704	1920	2181	1920
11.dat	1184	1372	1372	1540	1372
12.dat	1703	1703	1703	1871	1703
13.dat	838	838	1238	1213	1238
14.dat	1126	1126	1579	2026	1579
15.dat	747	747	1228	2377	1228
16.dat	726	726	867	990	867
17.dat	777	777	777	922	777
18.dat	1011	1085	1085	1330	1085
19.dat	822	832	832	927	832
20.dat	1306	1306	1306	1527	1306
21.dat	1677	1677	1677	1841	1677
22.dat	2088	2246	2246	3142	2246
23.dat	793	869	869	1359	869
24.dat	1234	1369	1369	1369	1369
25.dat	1237	1237	1418	2200	1418
26.dat	1262	1355	1602	1364	1602
27.dat	910	910	910	929	910
28.dat	1240	1240	1364	1495	1364
29.dat	775	1008	1182	1290	1182
30.dat	1030	1108	1164	1430	1164
RATA-RATA	1129.433	1178.8	1277.7	1526.933	1277.7

Jumlah Vertek = 20

HVTk = {1,2,3}

No. Problem	MST	DCMST	WAK1	WAK2	WAK3
1.dat	982	1103	1103	1225	1103
2.dat	1192	1198	1250	1473	1250
3.dat	851	874	874	1114	874
4.dat	1244	1274	1393	1477	1393
5.dat	1331	1390	1388	1732	1388
6.dat	1012	1211	1211	1359	1211
7.dat	1389	1411	1986	2458	1986
8.dat	1552	1772	1772	1946	1772
9.dat	1044	1044	1351	1311	1351
10.dat	903	903	1106	1258	1106
11.dat	1682	1698	2044	2461	2044
12.dat	1020	1267	1121	1347	1121
13.dat	923	1088	1321	1795	1321
14.dat	1440	1453	1737	1816	1737
15.dat	1162	1162	1162	1185	1162
16.dat	1230	1297	1297	1510	1297
17.dat	1544	1618	1618	1833	1618
18.dat	1634	1838	1905	2481	1905
19.dat	987	1140	1146	1351	1146
20.dat	949	1058	1179	1425	1179
21.dat	1298	1560	1560	1801	1560
22.dat	1088	1243	1688	1980	1688
23.dat	1087	1133	1437	2018	1437
24.dat	1421	1451	1571	2294	1571
25.dat	1259	1381	1381	1392	1381
26.dat	1226	1289	1416	2263	1416
27.dat	1124	1247	1247	1427	1247
28.dat	1265	1419	1419	1519	1419
29.dat	811	1061	1061	1225	1061
30.dat	1233	1387	1387	1929	1387
RATA-RATA	1196.1	1299	1404.367	1680.167	1404.367

Jumlah Vertek = 30

HVTk = {1,2,3,4,5,6}

No. Problem	MST	DCMST	WAK1	WAK2	WAK3
1.dat	1158	1348	1379	1541	1379
2.dat	1115	1268	1390	1790	1390
3.dat	1080	1126	1135	1570	1135
4.dat	854	888	1201	1403	1147
5.dat	1322	1388	1388	1891	1388
6.dat	1247	1707	1753	2051	1753
7.dat	1031	1077	1202	1711	1202
8.dat	1395	1591	1591	2515	1591
9.dat	1375	1508	1566	1737	1566
10.dat	799	936	1260	1329	1260
11.dat	1780	1814	1814	2072	1814
12.dat	1246	1385	1553	1921	1553
13.dat	1411	1653	1896	2324	1896
14.dat	1105	1241	1510	1856	1510
15.dat	1270	1289	1351	1677	1351
16.dat	1367	1539	1594	1745	1594
17.dat	1209	1339	1737	1820	1737
18.dat	878	884	926	1277	926
19.dat	1357	1400	1553	1911	1553
20.dat	760	826	963	1719	963
21.dat	804	849	1016	1232	1016
22.dat	793	1363	1358	2294	1441
23.dat	1370	1777	1777	2659	1777
24.dat	1182	1224	1340	1534	1340
25.dat	1410	1532	2065	2570	2065
26.dat	1108	1240	1434	1820	1434
27.dat	1085	1190	1219	1435	1219
28.dat	1147	1197	1278	1647	1278
29.dat	1364	1364	1589	1615	1589
30.dat	1301	1643	1804	1911	1804
RATA-RATA	1177.433	1319.533	1454.733	1819.233	1455.7

Jumlah Vertek = 40

HVTk = {1,2,3,4,5,6,7,8,9}

No. Problem	MST	DCMST	WAK1	WAK2	WAK3
1.dat	897	964	1163	1914	1163
2.dat	1192	1234	1302	1862	1357
3.dat	1028	1191	1257	1753	1257
4.dat	1009	1035	1405	1575	1262
5.dat	996	1138	1230	1566	1221
6.dat	1094	1334	1322	1714	1322
7.dat	1223	1363	1402	1814	1402
8.dat	1057	1185	1285	1787	1285
9.dat	986	1048	1242	1403	1242
10.dat	983	1067	1208	1427	1115
11.dat	1275	1362	1567	1882	1567
12.dat	1025	1115	1138	1878	1138
13.dat	1541	1784	1907	2174	1971
14.dat	1267	1429	1444	1822	1444
15.dat	946	1037	1410	1770	1410
16.dat	1096	1297	1594	1642	1486
17.dat	1163	1182	1261	1792	1261
18.dat	1115	1401	1377	1855	1377
19.dat	1029	1219	1266	1500	1266
20.dat	1044	1132	1383	2072	1383
21.dat	1011	1187	1238	1626	1238
22.dat	1296	1307	1479	1808	1479
23.dat	1185	1259	1455	1767	1480
24.dat	1353	1383	1611	1695	1638
25.dat	1255	1804	1747	1879	1747
26.dat	1103	1268	1268	1332	1268
27.dat	1507	1531	1647	2200	1647
28.dat	1172	1287	1385	1922	1385
29.dat	1619	1927	1927	2394	1927
30.dat	1070	1119	1269	1438	1269
RATA-RATA	1151.233	1286.3	1406.3	1775.433	1400.233

Jumlah Vertek = 50

HVTk = {1,2,3,4,5,6,7,8,9,10,11,12}

No. Problem	MST	DCMST	WAK1	WAK2	WAK3
1.dat	1067	1188	1343	1680	1343
2.dat	1253	1430	1674	2119	1674
3.dat	1102	1205	1347	1546	1367
4.dat	1082	1243	1354	1697	1297
5.dat	1586	1669	1798	2122	1823
6.dat	1322	1549	1637	1894	1637
7.dat	1314	1460	1791	1717	1675
8.dat	1317	1468	1592	1871	1592
9.dat	1455	1513	1670	1952	1680
10.dat	1042	1129	1456	1918	1376
11.dat	1363	1606	1575	2055	1575
12.dat	1243	1419	1515	2101	1515
13.dat	832	924	977	1368	967
14.dat	1143	1360	1385	1591	1385
15.dat	1034	1157	1441	1626	1444
16.dat	1459	1574	1692	2226	1692
17.dat	1348	1519	1900	2115	1842
18.dat	1160	1434	1607	1876	1552
19.dat	1234	1330	1508	1818	1508
20.dat	1247	1490	1561	1972	1561
21.dat	1028	1156	1269	1832	1269
22.dat	1292	1557	1693	1969	1646
23.dat	1110	1153	1459	1791	1459
24.dat	1103	1122	1263	1349	1263
25.dat	1325	1432	1600	2104	1600
26.dat	1339	1390	1750	1847	1692
27.dat	1119	1147	1214	1684	1214
28.dat	1481	1587	2138	2340	1703
29.dat	800	891	1089	1174	1043
30.dat	1503	1592	1869	2226	1678
RATA-RATA	1223.433	1356.467	1538.9	1852.667	1502.4

Jumlah Vertek = 60

HVTk = {1,2,3,4,5,6,7,8,9,10,11,12,13,14}

No. Problem	MST	DCMST	WAK1	WAK2	WAK3
1.dat	1129	1214	1437	2007	1459
2.dat	1099	1323	1570	2428	1570
3.dat	1150	1319	1567	1808	1524
4.dat	1201	1451	1622	2174	1530
5.dat	1231	1313	1413	2077	1413
6.dat	1227	1425	1475	1925	1475
7.dat	950	1075	1158	1689	1158
8.dat	1421	1459	1623	1812	1623
9.dat	1293	1456	1708	2372	1791
10.dat	1086	1207	1318	1783	1318
11.dat	1027	1168	1321	1845	1282
12.dat	1546	1729	2008	2350	1966
13.dat	1160	1270	1613	2558	1581
14.dat	1393	1766	2026	2609	2026
15.dat	1237	1403	1777	2439	1568
16.dat	1209	1313	1433	1708	1433
17.dat	1096	1206	1425	1799	1462
18.dat	1379	1628	1702	2480	1699
19.dat	1182	1390	1528	2353	1528
20.dat	1324	1398	1600	2123	1524
21.dat	1210	1244	1467	1885	1453
22.dat	1305	1406	1690	2332	1690
23.dat	1023	1179	1463	1508	1463
24.dat	1224	1356	1751	2191	1751
25.dat	810	904	1263	1380	1263
26.dat	960	1166	1164	1368	1164
27.dat	1001	1121	1468	2109	1468
28.dat	1319	1463	1597	2257	1573
29.dat	979	1101	1358	1655	1245
30.dat	1096	1169	1393	1922	1449
RATA-RATA	1175.567	1320.733	1531.267	2031.533	1514.967

Jumlah Vertek = 70

HVTk = {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18}

No. Problem	MST	DCMST	WAK1	WAK2	WAK3
1.dat	1066	1378	1455	1567	1455
2.dat	1470	1747	1900	2236	1861
3.dat	993	1240	1307	1628	1307
4.dat	1289	1380	1548	1919	1548
5.dat	1535	1652	1808	2429	1808
6.dat	1411	1676	1721	2216	1721
7.dat	1306	1455	1744	1880	1744
8.dat	1234	1496	1557	1937	1559
9.dat	1263	1466	1527	2003	1527
10.dat	1247	1384	1490	2114	1497
11.dat	1402	1577	1830	2401	1830
12.dat	1312	1341	1743	2103	1832
13.dat	1216	1410	1500	1845	1500
14.dat	1167	1386	1533	1721	1519
15.dat	1397	1445	1534	1925	1534
16.dat	1098	1324	1527	1968	1544
17.dat	1321	1639	1906	2525	1831
18.dat	1056	1157	1521	1718	1521
19.dat	1154	1307	1497	1773	1447
20.dat	977	1151	1159	1526	1159
21.dat	1195	1278	1458	1940	1463
22.dat	1257	1316	1541	2749	1498
23.dat	1196	1313	1664	1965	1606
24.dat	1232	1362	1509	2130	1509
25.dat	1449	1521	1684	2500	1674
26.dat	1158	1479	1779	2012	1567
27.dat	1175	1275	1519	1874	1464
28.dat	1262	1396	1503	1898	1503
29.dat	953	984	1077	1389	1097
30.dat	1472	1766	1932	2329	1932
RATA-RATA	1242.1	1410.033	1582.433	2007.333	1568.567

Jumlah Vertek = 80

HVTk = {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21}

No. Problem	MST	DCMST	WAK1	WAK2	WAK3
1.dat	1199	1493	1563	1711	1563
2.dat	1283	1489	1631	2375	1631
3.dat	1368	1511	1758	2550	1758
4.dat	1083	1266	1448	1793	1430
5.dat	1480	1670	1716	2110	1716
6.dat	1187	1362	1406	1902	1406
7.dat	1479	1707	1922	2306	1922
8.dat	972	1160	1241	1445	1241
9.dat	1032	1240	1486	1816	1486
10.dat	1441	1589	1888	2354	1802
11.dat	1338	1457	1585	2174	1568
12.dat	1135	1189	1341	1686	1341
13.dat	1252	1386	1525	1922	1400
14.dat	1161	1257	1417	1719	1464
15.dat	1363	1833	1904	2434	1833
16.dat	1029	1258	1391	2013	1387
17.dat	1368	1492	1678	2088	1678
18.dat	1382	1606	1891	2333	1855
19.dat	1337	1464	1621	2044	1568
20.dat	1277	1380	1600	1909	1600
21.dat	1337	1487	1643	2258	1652
22.dat	1050	1189	1552	2078	1470
23.dat	1131	1381	1543	2112	1512
24.dat	1374	1498	1642	1951	1640
25.dat	1159	1398	1948	2418	1840
26.dat	1156	1269	1389	2172	1417
27.dat	1225	1380	1876	1987	1746
28.dat	1260	1301	1396	1799	1396
29.dat	1076	1241	1460	1672	1460
30.dat	1171	1354	1467	1634	1467
RATA-RATA	1236.833	1410.233	1597.6	2025.5	1574.967

Jumlah Vertek = 90

HVTk = {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18, 19,20,21}

No. Problem	MST	DCMST	WAK1	WAK2	WAK3
1.dat	1402	1558	1572	2335	1653
2.dat	1376	1490	1883	1952	1804
3.dat	1393	1546	1666	1845	1593
4.dat	1329	1479	1610	2130	1620
5.dat	1338	1444	1590	2008	1528
6.dat	1243	1357	1730	1868	1565
7.dat	1212	1340	1570	1681	1557
8.dat	1245	1445	1426	2461	1429
9.dat	1402	1590	1755	2070	1751
10.dat	1224	1343	1597	1729	1597
11.dat	1397	1585	1816	2126	1816
12.dat	1032	1254	1663	2004	1663
13.dat	1080	1164	1310	1938	1310
14.dat	1186	1296	1356	1832	1356
15.dat	1102	1268	1552	1720	1553
16.dat	1128	1303	1259	1596	1259
17.dat	1579	1737	1826	2209	1825
18.dat	1119	1261	1327	1850	1327
19.dat	1442	1598	1689	2140	1777
20.dat	1241	1428	1623	2394	1623
21.dat	1147	1399	1574	1906	1542
22.dat	1290	1502	1667	1912	1711
23.dat	982	1155	1236	1583	1166
24.dat	1331	1587	1722	1972	1730
25.dat	939	1031	1078	1213	1078
26.dat	1023	1177	1404	1774	1392
27.dat	1356	1469	1651	2137	1595
28.dat	1278	1435	1605	2036	1549
29.dat	1353	1422	1816	2162	1791
30.dat	1271	1485	1630	2271	1622
RATA-RATA	1248	1404.933	1573.433	1961.8	1559.4

Jumlah Vertek = 100

HVTk = {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18, 19,20,21,22,23,24}

No. Problem	MST	DCMST	WAK1	WAK2	WAK3
1.dat	1168	1354	1624	1949	1573
2.dat	1264	1450	1623	2057	1615
3.dat	1260	1361	1442	1811	1442
4.dat	1064	1239	1324	1648	1335
5.dat	1251	1403	1749	2185	1560
6.dat	1160	1371	1536	1964	1561
7.dat	1298	1371	1401	2398	1433
8.dat	1215	1356	1589	2425	1589
9.dat	1203	1339	1811	2273	1607
10.dat	1382	1547	1733	2292	1732
11.dat	1154	1251	1407	1507	1407
12.dat	1117	1378	1738	2064	1577
13.dat	1139	1238	1415	2014	1354
14.dat	1474	1538	1810	2433	1717
15.dat	1322	1473	1641	2134	1641
16.dat	1138	1252	1317	1725	1317
17.dat	1233	1317	1600	1903	1536
18.dat	1325	1397	1638	2141	1590
19.dat	1225	1422	1414	1963	1414
20.dat	1163	1267	1770	2192	1574
21.dat	1325	1439	1636	2014	1606
22.dat	1151	1243	1277	1661	1277
23.dat	1442	1568	1703	2156	1703
24.dat	1138	1193	1409	2015	1502
25.dat	1172	1300	1383	1602	1383
26.dat	1121	1292	1355	1910	1355
27.dat	1214	1376	1563	1977	1473
28.dat	1218	1330	1501	1892	1497
29.dat	1431	1623	1859	2147	1859
30.dat	1256	1436	1587	2080	1528
RATA-RATA	1234.1	1370.8	1561.833	2017.733	1525.233