

Algoritma Greedy untuk Optimalisasi Ruangan dalam Penyusunan Jadwal Perkuliahan

Shinta Oktaviana, Abdurrahman Naufal
Jurusan Teknik Informatika dan Komputer
Politeknik Negeri Jakarta
Indonesia

shinta.oktaviana@tik.pnj.ac.id, abdurrahman.naufal.tik13@mhs.w.tik.pnj.ac.id

Diterima: 7 Mei 2017. Disetujui 16 Mei 2017. Dipublikasikan Mei 2017

Abstrak - Algoritma greedy merupakan algoritma yang bersifat heuristik dan urutan logisnya disusun berdasarkan langkah-langkah penyelesaian masalah yang disusun secara sistematis. Dalam penelitian ini algoritma greedy digunakan untuk mengoptimalkan penggunaan ruangan yang ada di Jurusan Teknik Informatika dan Komputer Politeknik Negeri Jakarta (JTik). Tujuan dilakukannya penelitian ini adalah untuk mengetahui apakah algoritma greedy dapat mengoptimalkan pembuatan Sistem Penjadwalan Mata Kuliah di JTik dalam mengetahui jumlah ruangan yang kosong setelah digunakannya algoritma, dan mengetahui apakah metode menggunakan algoritma greedy lebih baik dari pada sistem manual. Data yang digunakan adalah jadwal mata kuliah tahun ajaran 2016/2017 ganjil genap, dan 2015/2016 ganjil. Hasil akhir pembuatan sistem berupa suatu jadwal mata kuliah yang dapat dilihat ruangan kosong yang ada dan durasi kosongnya ruangan tersebut. Pengujian dilakukan dengan cara membandingkan banyaknya ruangan kosong hasil dari sistem yang menggunakan algoritma greedy dengan sistem manual. Berdasarkan data jadwal mata kuliah yang diinput ke dalam sistem ditunjukkan bahwa algoritma greedy memiliki ruangan kosong lebih banyak, hal itu menunjukkan bahwa algoritma greedy mampu mengoptimalkan ruangan pada penjadwalan mata kuliah JTik secara baik.

Kata Kunci: Algoritma Greedy, penjadwalan, optimalisasi ruangan.

I. PENDAHULUAN

Menciptakan suatu jadwal kuliah yang optimal cukup sulit dikarenakan banyak variabel yang saling terkait sehingga membutuhkan suatu penanganan yang signifikan. Penjadwalan kuliah merupakan pengaturan penempatan waktu, kelas (kelompok mahasiswa), dosen, ruangan, jumlah mata kuliah yang dibuka semester tersebut, dan memperhatikan sejumlah aturan yang dimiliki oleh organisasi tersebut. Sistem Penjadwalan dikatakan baik apabila dapat memberikan solusi terhadap

faktor-faktor tertentu, seperti dosen tidak boleh mengajar banyak kelas dalam satu waktu, ruangan tidak boleh diisi oleh banyak mata kuliah dalam satu waktu, dan kelas tidak boleh mengikuti banyak mata kuliah dalam satu waktu.

Dalam penjadwalan kuliah di Jurusan Teknik Informatika dan Komputer Politeknik Negeri Jakarta (JTik) yang ada saat ini banyak dosen yang tidak mendapatkan ruangan untuk jam pengganti. Penyusunan jadwal bisa memakan waktu berminggu-minggu dengan minimal pengerjaannya sampai dengan 7 minggu. Dibutuhkan ketelitian yang tinggi karena harus memperhatikan banyak variabel agar jadwal tidak bentrok. Mahasiswa tidak memiliki ruangan kosong untuk belajar di luar jam kuliah.

Cara manual yang dilakukan saat ini juga memungkinkan terjadinya kesalahan dengan intensitas yang cukup banyak. Ketersediaan ruangan yang sedikit, penjadwalan yang berubah-ubah di awal semester merupakan isu-isu yang perlu diperhatikan dalam penjadwalan mata kuliah di JTik. Dalam penelitian ini akan diuji seberapa optimal pemanfaatan ruangan menggunakan algoritma greedy dalam penerapannya ke jadwal kuliah di JTik. Sehingga diharapkan dapat menjadi referensi dalam meminimalisir isu-isu penjadwalan di JTik.

II. TINJAUAN PUSTAKA

A. Algoritma Greedy

“Algoritma adalah urutan logis langkah-langkah penyelesaian masalah yang disusun secara sistematis” [1]. Awalnya kata algoritma merupakan istilah yang merujuk kepada aturan-aturan aritmetis untuk menyelesaikan persoalan dengan menggunakan bilangan numerik Arab, namun pada abad ke-18 istilah ini telah berkembang sehingga makna algoritma menjadi lebih luas lagi menjadi suatu urutan langkah atau prosedur yang jelas dan diperlukan untuk menyelesaikan suatu

permasalahan. Kata algoritma berasal dari latinisasi nama seorang ahli matematika dari Uzbekistan Al Khawārizmi (hidup sekitar abad ke-9), sebagaimana tercantum pada terjemahan karyanya lam bahasa latin dari abad ke-12 "Algorithmi de numero Indorum".

Algoritma Greedy merupakan metode yang paling populer dalam memecahkan persoalan optimasi [2]. Hanya ada dua macam persoalan optimasi, yaitu maksimasi dan minimasi. Pada penelitian ini, algoritma greedy yang digunakan menerapkan pendekatan maksimasi. Algoritma Greedy adalah algoritma yang memecahkan masalah langkah per langkah. Pada setiap langkah terdapat banyak pilihan yang perlu dieksplorasi. Persoalan optimalisasi dalam konteks algoritma greedy disusun oleh elemen-elemen sebagai berikut:

1. Himpunan Kandidat, C.

Himpunan ini berisi elemen-elemen pembentuk solusi. Pada setiap langkah, satu buah kandidat diambil dari himpunannya.

2. Himpunan Solusi, S.

Himpunan ini berisi kandidat-kandidat yang terpilih sebagai solusi persoalan. Dengan kata lain, himpunan solusi adalah himpunan bagian dari himpunan kandidat.

3. Fungsi Seleksi (*Selection Function*)

Fungsi ini dinyatakan dengan predikat seleksi. Merupakan fungsi yang pada setiap langkah memilih kandidat yang paling memungkinkan mencapai solusi optimal. Kandidat yang sudah dipilih pada suatu langkah tidak pernah dipertimbangkan lagi pada langkah selanjutnya.

4. Fungsi Kelayakan (*Feasible*)

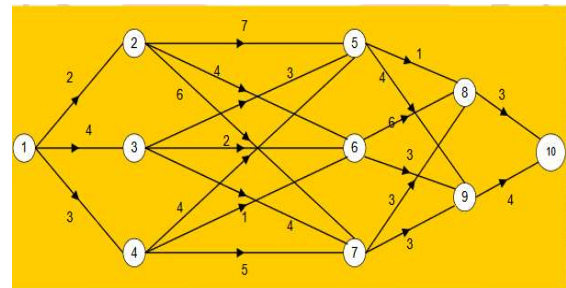
Fungsi ini dinyatakan dengan predikat layak. Fungsi kelayakan ini merupakan fungsi yang memeriksa apakah suatu kandidat yang telah dipilih dapat memberikan solusi yang layak, yakni kandidat tersebut bersama-sama dengan himpunan solusi yang sudah terbentuk tidak melanggar batasan/aturan (*constraints*) yang ada. Kandidat yang layak dimasukkan ke dalam himpunan solusi, sedangkan yang tidak layak dibuang dan tidak pernah dipertimbangkan lagi.

5. Fungsi Obyektif

Fungsi obyektif ini merupakan sebuah fungsi yang memaksimalkan atau meminimumkan nilai solusi. Dengan kata lain, algoritma greedy melibatkan pencarian sebuah himpunan bagian, S, dari himpunan kandidat, C yang dalam hal ini, S harus memenuhi beberapa kriteria yang ditentukan, yaitu menyatakan suatu solusi dan S dioptimasi oleh fungsi obyektif. Ada kalanya solusi optimum global yang diperoleh dari algoritma greedy yang diharapkan sebagai solusi optimum dari persoalan, belum tentu merupakan solusi

optimum (terbaik), tetapi solusi sub-optimum atau pseudo-optimum. Hal ini dikarenakan algoritma greedy tidak beroperasi secara menyeluruh terhadap semua alternatif solusi yang ada dan terdapat beberapa fungsi seleksi yang berbeda, yaitu jika fungsi seleksi tidak identik dengan fungsi obyektif karena fungsi seleksi biasanya didasarkan pada fungsi obyektif. Sehingga harus dipilih fungsi yang tepat jika menginginkan algoritma menghasilkan solusi optimal atau nilai yang optimum. Jadi, pada sebagian masalah algoritma greedy tidak selalu berhasil memberikan solusi yang benar-benar optimum, tetapi algoritma greedy pasti memberikan solusi yang mendekati (*approximation*) nilai optimum.

Menurut Angga & Munir [1] jika jawaban terbaik mutlak tidak diperlukan, maka algoritma greedy sering berguna untuk menghasilkan solusi cukup baik (*approximation*), daripada menggunakan algoritma yang lebih rumit untuk menghasilkan solusi yang terbaik. Bila algoritma greedy optimum, maka keoptimalannya itu dapat dibuktikan secara matematis.



Gambar 1. Contoh Skema Algoritma Greedy [1]

B. Penjadwalan

Baker [3] menyatakan bahwa pengertian penjadwalan adalah kegiatan pengalokasian sumber-sumber atau mesin-mesin yang ada untuk menjalankan sekumpulan tugas dalam jangka waktu tertentu.

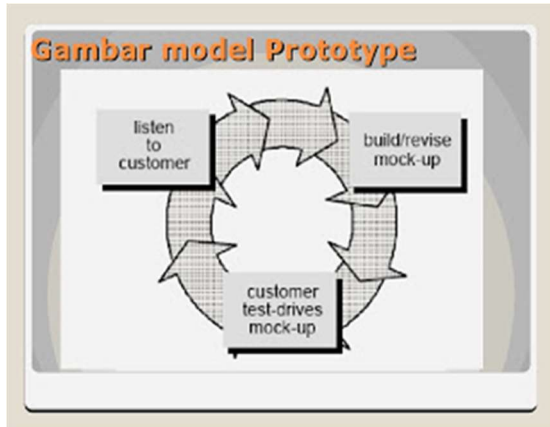
C. JQuery

JQuery adalah library atau kumpulan kode JavaScript siap pakai [4]. Keunggulan menggunakan JQuery dibandingkan dengan JavaScript standar, yaitu menyederhanakan kode JavaScript dengan cara memanggil fungsi-fungsi yang disediakan oleh JQuery.

III. METODE PENGEMBANGAN SISTEM

Pada penelitian ini, metode pengembangan sistem yang digunakan adalah metode *prototyping*. Hal ini dikarenakan penggunaan *prototyping* dalam pembuatan sebuah sistem menjadikan pengembangan lebih dekat dengan *user* [5]. Metode

ini memiliki beberapa tahapan diantaranya adalah seperti pada Gambar 2.



Gambar 2. Metode Prototyping [6]

A. Requirements and definition

Pada tahap ini, peneliti melakukan beberapa hal untuk mengetahui kebutuhan dari sistem yang dibangun. Adapun tahapannya adalah :

- Wawancara dilakukan untuk mengumpulkan semua informasi sebagai bahan untuk melakukan penelitian kepada ketua Program Studi Teknik Informatika
- Menganalisa permasalahan apa yang akan dicari penyelesaiannya.
- Mencari data-data, mempelajari jurnal-jurnal, artikel, atau buku dan sumber informasi dari internet yang digunakan sebagai bahan referensi sebelum memecahkan permasalahan yang dihadapi.
- Menganalisa alat dan bahan yang dibutuhkan dalam penyelesaian masalah.

B. System and software design

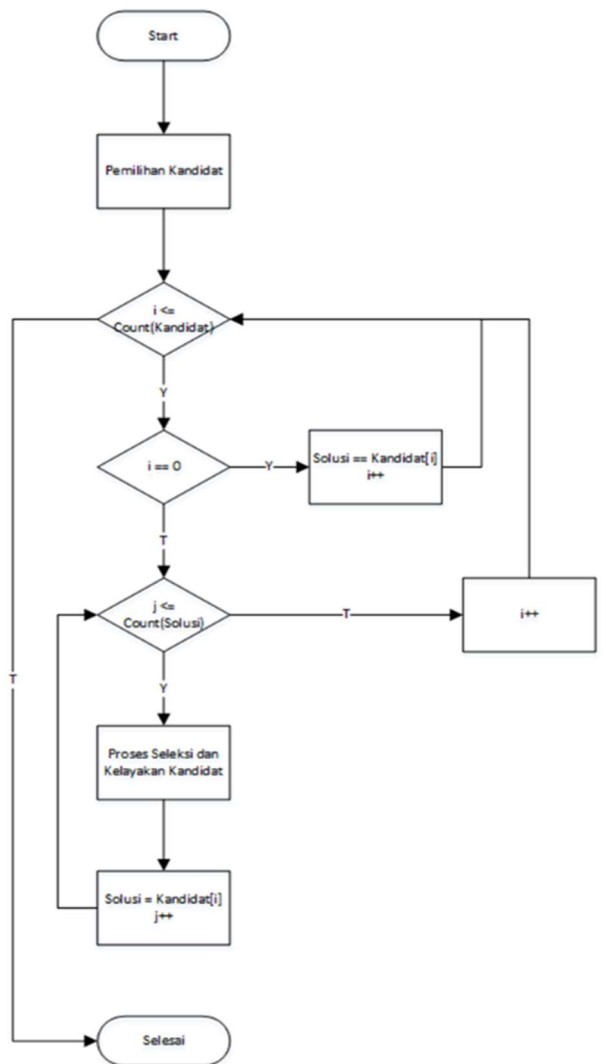
Pada tahap ini, dibuat perancangan dari sistem:

- Cara kerja sistem secara keseluruhan didesain menggunakan notasi diagram flow chart dan data flow diagram. Diagram sistem dapat dilihat pada Gambar 3 dan Gambar 4.
- Anggota himpunan kandidat merupakan suatu array yang terdiri dari data dosen, mata kuliah, ruangan, jumlah jam, dan kelas.
- Fungsi seleksi yang mengecek apakah pasangan data yang didapat dari himpunan kandidat dapat dinyatakan sebagai bagian dari solusi. Pada fungsi ini dimasukkan sejumlah aturan yang dimiliki organisasi JTik dalam menyusun jadwal, seperti aturan pasangan data yang dinyatakan bentrok, aturan jumlah jam maksimal dosen mengajar, dll.
- Fungsi Objektif yang digunakan pada penelitian ini bertujuan menentukan mana solusi paling baik, dari sekumpulan solusi yang ada.

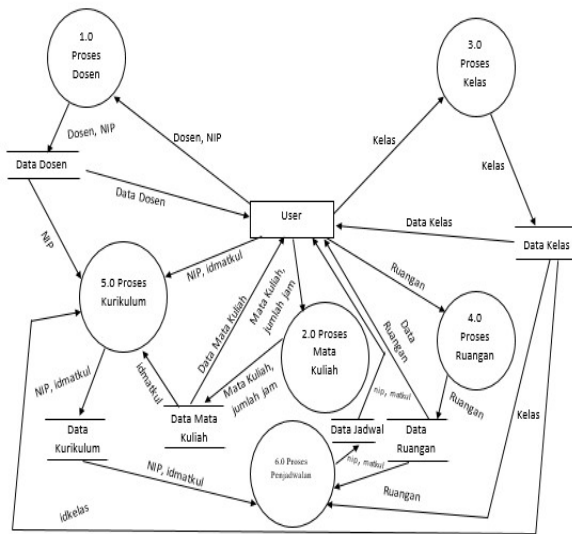
C. Implementation and Unit Testing

Pada tahap ini, semua yang telah dirancang akan diimplementasikan sesuai hasil rancangan. Berikut adalah tahapannya:

- Melakukan implementasi dan pembangunan sistem sebagai penyelesaian masalah, dengan membuat sistem penjadwalan dengan algoritma greedy
- Uji coba dengan melakukan perbandingan jumlah ruangan kosong pada sistem manual dengan sistem yang menggunakan algoritma greedy.



Gambar 3. Flowchart Penjadwalan



Gambar 4. Data Flow Diagram Level 1

D. System Testing

Pada tahap ini, metode pengujian memanfaatkan data riil dari JTİK dengan data pada Tabel 1. Pengujian dilakukan dengan menggunakan browser Google Chrome dan Microsoft Excel atau Adobe Reader. Pengujian dilakukan dengan melakukan pengamatan di kedua alat ini. Metode pengujian ini menggunakan sebuah tabel yang memiliki 2 kolom utama, kolom yang pertama merupakan metode lama yang sudah diterapkan, sementara kolom kedua merupakan metode yang menggunakan algoritma greedy. Pengujian ini menghitung berapa waktu kosong yang dimiliki oleh setiap ruangan dari jadwal perkuliahan yang terbentuk oleh sistem, dimana waktu kosong tersebut dihitung perhari dalam sepekan.

TABEL 1. DATA RIIL

No	Data	Jumlah
1	Ruangan	27
2	Mata Kuliah	214
3	Kelas	52
4	Dosen	47
5	Kurikulum	484

Langkah pertama pengujian adalah memastikan data yang diterapkan pada metode lama yang digunakan oleh JTİK adalah berdasarkan per ruangan. Hal ini dilakukan untuk mempermudah dalam proses perbandingan. Langkah kedua adalah mulai meng-input data dosen, mata kuliah, kelas, ruangan, dan kurikulum

berdasarkan jadwal manual JTİK. Langkah ketiga adalah mulai membandingkan setiap ruangan setiap harinya antara metode lama yang digunakan oleh JTİK dengan sistem yang telah dibuat dengan metode algoritma greedy pada modul penjadwalan. Langkah keempat adalah menyimpulkan setiap ruangan dengan perhitungan persentase ruangan kosong dengan rumus (1). Langkah kelima adalah menyimpulkan keseluruhan dari setiap kesimpulan per-ruangan dan dengan rumus (2).

$$\frac{(\text{total jam per hari})}{14} \times 100\% \tag{1}$$

$$\frac{(\text{presentase ruangan kosong tiap ruangan})}{\text{banyaknya ruangan}} \times 100\% \tag{2}$$

IV. HASIL DAN PEMBAHASAN

Data hasil pengujian dilihat dari banyaknya ruang kosong per harinya pada jadwal real dan jadwal hasil algoritma greedy. Karena hal ini dapat menjadi tolak ukur penilaian seberapa solutif kedua jadwal dalam pengoptimalisasian ruangan. Semakin banyak ruangan kosong pada hari tersebut, semakin solutif jadwal tersebut.

Berdasarkan tabel 2 ditunjukkan bahwa sistem menggunakan algoritma greedy memiliki ruang kosong lebih banyak dibandingkan penjadwalan menggunakan teknik manual yang terlihat hampir semua ruangan terisi setiap harinya. Hal ini berarti bahwa penjadwalan menggunakan algoritma greedy telah berhasil untuk mengoptimasi penggunaan ruangan.

Dari data hasil pengujian pada tabel 2 dapat dilihat bahwa jadwal yang dibuat menggunakan metode algoritma greedy lebih solutif dalam hal pengoptimalisasian ruangan ketimbang jadwal real, dapat dilihat dari data keseluruhan. Jumlah ruangan kosong pada algoritma greedy jauh lebih banyak, ini terjadi karena penumpukan jadwal pada awal-awal ruangan, sehingga membuat ruangan lain tidak kebagian jadwal dan menghasilkan ruangan kosong yang dapat dimanfaatkan. Hal itu terjadi karena algoritma greedy akan melihat kandidat terbaik pada saat itu dan setiap penyusunan kandidat akan mempengaruhi ruangan mana yang akan terisi lebih awal.

TABEL 2. DATA HASIL PENGUJIAN KESELURUHAN

Tahun Ajaran	Semester	Hari	Ruang Kosong dengan Metode Manual		Ruang Kosong dengan Metode Greedy	
			Persentase	Jumlah	Persentase	Jumlah
2016/2017	Genap	Senin	14,81%	4	74,07%	20
		Selasa	18,52%	5	74,07%	20
		Rabu	14,81%	4	74,07%	20
		Kamis	11,11%	3	77,78%	21
		Jum'at	18,52%	5	81,48%	22
		Sabtu	77,78%	21	77,78%	21
2016/2017	Ganjil	Senin	22,22%	6	81,48%	22
		Selasa	22,22%	6	81,48%	22
		Rabu	25,93%	7	81,48%	22
		Kamis	37,04%	10	81,48%	22
		Jum'at	59,26%	16	81,48%	22
		Sabtu	96,30%	26	85,19%	23
2015/2016	Ganjil	Senin	22,22%	6	74,07%	20
		Selasa	22,22%	6	77,78%	21
		Rabu	22,22%	6	77,78%	21
		Kamis	25,93%	7	77,78%	21
		Jum'at	29,63%	8	77,78%	21
		Sabtu	92,59%	25	81,48%	22

Berbeda dengan jadwal riil dengan metode manual yang digunakan oleh JTİK, data cenderung rata per harinya. Ruang kosong hanya terlihat di hari Sabtu sehingga akan membuat masalah-masalah seperti dosen tidak kebagian ruang di jam pengganti atau mahasiswa tidak memiliki ruang untuk belajar diluar jam kuliah. Jika dilihat ruang kosong di jadwal riil bisa mencapai lebih dari 50%, itu karena dalam 1 hari total jam aktif adalah 14 jam tetapi pada kenyataannya jadwal riil aktif hanya 6-12 jam per harinya dan menimbulkan ruang kosong di sisa jam ditambah ruang lain yang benar-benar tidak terpakai di hari itu

V. KESIMPULAN DAN SARAN

Berdasarkan pelaksanaan penelitian yang telah dilakukan dapat disimpulkan bahwa penerapan algoritma greedy terhadap sistem penjadwalan mata kuliah pada JTİK mampu mengatasi optimasi ruang dilihat dari banyaknya ruang yang kosong. Hal tersebut dapat dilihat dari hasil tahap pengujian yang menunjukkan bahwa pada algoritma greedy setiap ruang yang pertama kali dilalui oleh proses algoritma akan dimasukkan oleh kurikulum yang tersedia saat itu sampai batas maksimum total jam per ruang, yaitu 14 jam. Hal tersebut mengakibatkan banyak

ruang lain kosong. Ini berarti menunjukkan bahwa JTİK tidak perlu menambahkan ruang lagi karena ruang yang tersedia saat ini sudah lebih dari cukup apabila JTİK menerapkan algoritma greedy pada sistem penjadwalannya. Lebih jauh dari itu, ruang-ruang yang kosong bisa dimanfaatkan untuk keperluan lain seperti tempat belajar ataupun lab. Selain itu dengan algoritma greedy pembuatan jadwal akan lebih cepat sekitar 1 menit, banyaknya hari libur akibat teroptimisasinya jadwal pada hari-hari awal di setiap kelas dan meminimalisir kesalahan-kesalahan seperti bentrok jadwal kuliah.

Hal yang perlu diperhatikan lainnya adalah selain memberikan hasil positif, namun algoritma ini dalam penerapannya ke penjadwalan mata kuliah bersifat tidak humanis karena beberapa kejanggalan akan ditemukan dalam pengujian, yaitu:

1. Adanya dosen yang mengajar dari pagi sampai malam.
2. Adanya dosen wanita mengajar pada malam hari, dikatakan tidak humanis karena kebanyakan dari dosen wanita tidak menginginkan hal ini.
3. Tidak adanya jeda waktu untuk istirahat.
4. Mahasiswa dituntut untuk mengikuti pelajaran dari pagi sampai malam.

DAFTAR PUSTAKA

- [1] Angga, C., & Munir, R. 2012. Pengembangan Algoritma Greedy untuk Optimalisasi Penataan Peti Kemas Pada Kapal Pengangkut. *Jurnal Sarjana Institut Teknologi Bandung bidang Teknik Elektro dan Informatika*.
- [2] Juniar, Ahmad. 2015. Penerapan Algoritma Greedy pada Penjadwalan Produksi Single-Stage dengan Parallel Machine di Industri Konveksi. Vol. 16, No. 2. Jakarta.
- [3] Baker, K. 1974. *Introduction To Sequencing and Scheduling*. New York: Jhon Willey and Sons, Inc.
- [4] Sigit W, A. 2011. Pemrograman Web Aplikatif dengan Java. PT. Elex Media Komputindo. Jakarta.
- [5] Kendall, K., & Kendall, J. 2003. *Analisis dan Perancangan Sistem, Alih Bahasa oleh Thamir Abdul Hafedh Al-Hamdany*. Vol 5. Jakarta.
- [6] Roger, S. P. 2012. *Rekayasa Perangkat Lunak (Pendekatan. Praktisi) Edisi 7* : Buku 1. Yogyakarta: ANDI.
- [7] Sari, R.. 2017. Apa Saja Permasalahan yang Terjadi pada Penjadwalan di Jurusan TIK. *Wawancara. Politeknik Negeri Jakarta. [Diwawancara tanggal 25Februari2017]*