

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Electron adalah salah satu kerangka kerja terbaru yang memungkinkan pengembang membuat aplikasi *desktop* asli dengan teknologi web populer : JavaScript, HTML5, dan CSS. Dengan *Electron*, pengembang web dapat menggunakan keterampilan yang mereka miliki untuk membangun aplikasi yang memiliki banyak kemampuan seperti aplikasi *desktop* asli. *Elektron* telah menjadi sangat populer sejak dirilis dan digunakan oleh perusahaan, seperti : *Microsoft*, *Facebook*, *Slack*, dan *Docker*. Aplikasi ini dapat dikemas untuk dapat berjalan langsung di *macOS*, *Windows*, dan *Linux*. Bisa juga didistribusikan melalui *Mac App Store* atau *Microsoft Store*.

Vis.js adalah sebuah *library* visualisasi berbasis *browser* yang bersifat dinamis. *Library* ini dirancang agar mudah digunakan untuk menangani data dinamis dalam jumlah yang besar dan memungkinkan untuk memanipulasi serta berinteraksi dengan data tersebut. *Library* ini terdiri dari komponen - komponen, seperti *DataSet*, *Timeline*, *Network (tree)*, *Graph2d*, dan *Graph3d*. *DataSet* berfungsi untuk mengelola data yang tidak terstruktur. Terdapat fitur *add*, *update*, dan *remove* data di dalamnya. *Timeline* berfungsi untuk menampilkan data dalam bentuk *timeline* yang dapat disesuaikan dengan *item* dan rentangnya. *Network (tree)* berfungsi untuk menampilkan data dalam bentuk jaringan yang dinamis, dapat diatur secara otomatis, dan dapat disesuaikan. *Graph2d* berfungsi untuk menampilkan data dalam bentuk grafik dan diagram batang pada *timeline* yang interaktif sesuai yang diinginkan. *Graph3d* berfungsi untuk menampilkan data dalam bentuk grafik 3d dengan animasi yang interaktif.

GitHub adalah sebuah aplikasi berbasis *website* dengan *Version Control System* (VCS) yang menyediakan layanan untuk menyimpan *repository* dengan gratis. VCS adalah sebuah infrastruktur yang dapat mendukung pengembangan *software* secara kolaboratif. Setiap anggota yang berada di dalam sebuah tim pengembangan *software* dapat menulis kode programnya masing - masing kemudian digabungkan ke server yang sudah memiliki VCS yang digunakan. *Repository* merupakan tempat yang dapat digunakan untuk menyimpan berbagai file berupa *source code*. Aplikasi ini termasuk sangat populer dan banyak digunakan termasuk oleh perusahaan - perusahaan besar, seperti : *Facebook*, *Google*, dan *Twitter*.

Saat mahasiswa akan melakukan *FRS*, seringkali mereka kesulitan untuk melihat kurikulum tahun ajaran yang berlaku. Karena setiap kurikulum memiliki aturan yang berbeda dalam pengambilan matakuliah ataupun matakuliah yang disediakan, maka mahasiswa kadang bingung untuk memilih matakuliah apa yang akan diambil di semester berikutnya.

Maka dari itu, pada skripsi ini akan dibuat sebuah aplikasi visualisasi kurikulum 2018 berbasis *Electron* dengan menggunakan *library Vis.js*. Dengan aplikasi ini diharapkan mahasiswa dapat lebih mudah untuk melihat kurikulum yang ada sehingga mempermudah mereka untuk memilih matakuliah apa yang akan diambil di semester berikutnya, kemudian dengan penggunaan *framework cross platform Electron* ini diharapkan semua mahasiswa pengguna *macOs*, *Windows*, dan *Linux* dapat mengaksesnya dengan mudah.

1.2 Rumusan Masalah

Rumusan masalah yang akan dibahas pada skripsi ini adalah :

1. Bagaimana cara memvisualisasikan kurikulum 2018 dalam bentuk *tree*?
2. Bagaimana cara memvisualisasikan kurikulum 2018 dalam bentuk *timeline*?
3. Bagaimana cara membaca kurikulum 2018 FTIS UNPAR dari *github*?

1.3 Tujuan

Tujuan yang akan dicapai dari penulisan skripsi ini adalah :

1. Memahami cara memvisualisasikan kurikulum 2018 dalam bentuk *tree*.
2. Memahami cara memvisualisasikan kurikulum 2018 dalam bentuk *timeline*.
3. Memahami cara membaca data kurikulum 2018 FTIS UNPAR dari *github*.

1.4 Batasan Masalah

Batasan-batasan masalah yang ditetapkan adalah sebagai berikut :

1. Perangkat lunak ini hanya memvisualisasikan kurikulum milik Universitas Katolik Parahyangan tahun 2018.
2. Perangkat lunak ini hanya memvisualisasikan dalam bentuk *tree* dan *timeline*.
3. Perangkat lunak ini hanya akan menggunakan bahasa *HTML*, *PHP* dan *JavaScript*

1.5 Metodologi

Bagian - bagian pengerjaan skripsi ini adalah :

1. Melakukan studi tentang *framework Electron* dan *Library Vis.js*.
2. Mempelajari cara membuat aplikasi berbasis *Electron*.
3. Mempelajari cara memvisualisasikan data dalam bentuk *tree* dan *timeline* dengan *Vis.js*.
4. Mempelajari data kurikulum 2018 di *github* beserta cara pengambilan datanya.
5. Merancang aplikasi berbasis *Electron*.
6. Merancang visualisasi kurikulum 2018 dalam bentuk *tree*.
7. Merancang visualisasi kurikulum 2018 dalam bentuk *timeline*.
8. Mendesain antarmuka aplikasi.
9. Melakukan pengujian dan eksperimen.
10. Membuat dokumen skripsi.

1.6 Sistematika Pembahasan

Skripsi ini terdiri dari enam bab, yaitu pendahuluan, landasan teori, analisis, perancangan, implementasi, dan kesimpulan dan saran.

Bab I membahas latar belakang dibuatnya skripsi, rumusan masalah yang terdapat pada skripsi, tujuan skripsi ini dibuat, batasan masalah agar skripsi yang dibuat tidak terlalu luas, dan metodologi yang berisi langkah - langkah pengerjaan skripsi agar berjalan sistematis.

Bab II berisi teori - teori yang berfungsi sebagai referensi dalam pembuatan skripsi dan membantu dalam menyelesaikan masalah pada skripsi.

Bab III berisi analisis terhadap perangkat lunak yang telah dibuat.

Bab IV berisi perancangan perangkat lunak menggunakan aplikasi *Electron* dan *library Vis.js*.

Bab V berisi implementasi perangkat lunak yang berlandaskan teori - teori yang telah dipelajari.

Bab VI berisi kesimpulan skripsi yang telah dibuat dan juga saran yang ditujukan untuk skripsi berikutnya.

BAB 2

LANDASAN TEORI

2.1 Kurikulum 2018

Kurikulum didefinisikan sebagai seperangkat rencana dan pengaturan mengenai capaian pembelajaran, an lulusan, bahan kajian, proses, dan penilaian yang digunakan sebagai pedoman penyelenggaraan program studi menjadi sarana utama untuk mencapai tujuan tersebut.

Penyusunan Kurikulum 2018 berpegang pada prinsip bahwa kurikulum yang baik adalah kurikulum yang tidak hanya kokoh, secara teoretis konseptual dapat dipertanggungjawabkan, namun juga secara praktis dapat dilaksanakan. Selain itu kurikulum juga harus cukup fleksibel agar dapat mengakomodasi perubahan-perubahan, namun tanpa kehilangan ciri atau kekhasan dari program studi.

Dalam penyusunan Kurikulum 2018 Program Studi Teknik Informatika secara khusus juga memperhatikan Kerangka Kualifikasi Nasional Indonesia (KKNI) yang tertuang dalam Peraturan Presiden no 8 tahun 2012. KKNI merupakan pernyataan kualitas SDM Indonesia, di mana tolok ukur kualifikasinya ditetapkan berdasarkan capaian pembelajaran (learning outcomes) yang dimilikinya. Penyusunan kurikulum mengikuti tahapan perancangan kurikulum yang disarankan oleh Kemenristekdikti yang diberikan pada 2.1. Tahapan penyusunan kurikulum 2018 meliputi kegiatan sebagai berikut:

1. Melakukan evaluasi diri
2. Merumuskan profil lulusan dengan pelacakan lulusan
3. Menentukan capaian pembelajaran
4. Menentukan bahan kajian
5. Menyusun matriks pembelajaran dan bahan kajian
6. Membentuk mata kuliah
7. Menyusun struktur kurikulum dan menentukan metode pembelajaran



Gambar 2.1: Tahapan Penyusunan Kurikulum

2.1.1 Kodifikasi

Kodifikasi tiap mata kuliah dibuat berdasarkan Peraturan Rektor UNPAR No. III/PRT/2017-03/46 tentang Standar Penyusunan Kurikulum Program Studi di Lingkungan UNPAR. Kode ini terdiri atas 11 digit, dengan rincian berikut:

- 3 digit - kode khas Program Studi: AIF
- 2 digit - tahun diberlakukannya kurikulum (2 digit terakhir): 18
- 1 digit - urutan tahun pengajaran
- 1 digit - nomor urut KBI pengampu mata kuliah
- 2 digit - nomor urut mata kuliah per semester, dengan angka pada digit terakhir sebagai penentu semester; ganjil atau genap
- 2 digit - jumlah sks mata kuliah

Informasi lengkap terkait kodifikasi ini diberikan di Tabel 2.2.

Penyelenggara	Universitas	Prodi
Kode khas prodi	MKU	AIF
Tahun berlaku kurikulum	18	18
Urutan tahun pengajaran	0	1: tahun pertama 2: tahun kedua 3: tahun ketiga 4: tahun keempat
Nomor urut KBI pengampu	**	0: Prodi 1: Teori Komputasi 2: Sistem Terdistribusi 3: Sistem Informasi
Nomor urut mata kuliah	**	Urutan mata kuliah per semester, dengan angka pada digit terakhir sebagai penentu semester; ganjil atau genap
Jumlah sks	**	Jumlah sks

**Kode mata kuliah MKU ditentukan oleh universitas

Gambar 2.2: Kodifikasi mata kuliah

2.1.2 Bobot Pemrograman

Berdasarkan hasil evaluasi Kurikulum 2013, salah satu masalah yang ditemukan adalah bahwa mahasiswa masih sulit menguasai materi kuliah di jalur pemrograman, yang merupakan kuliah inti dari Prodi Teknik Informatika UNPAR. Selain karena memang logika pemrograman tidak mudah untuk dipahami, kurangnya pengalaman mahasiswa dalam membangun program komputer juga menjadi penyebab munculnya permasalahan ini.

Selain memperbaiki struktur kuliah jalur pemrograman, dan perbaikan materi perkuliahan, cara lain yang digunakan untuk mendukung kemampuan pemrograman mahasiswa adalah dengan menempatkan bobot pemrograman di kuliah-kuliah yang cocok. Bobot pemrograman ini menentukan di kuliah mana saja mahasiswa harus membangun program komputer, dan seberapa besar skala program komputer yang dibuat. Bagian pembangunan program komputer misalnya dapat diletakkan pada saat praktikum, atau dijadikan bagian dari tugas kuliah.

Besar bobot pemrograman dalam kurikulum ini adalah 0.25, 0.5, 0.75, dan 1. Penjelasan terkait masing - masing bobot ini diberikan pada Tabel 2.3.

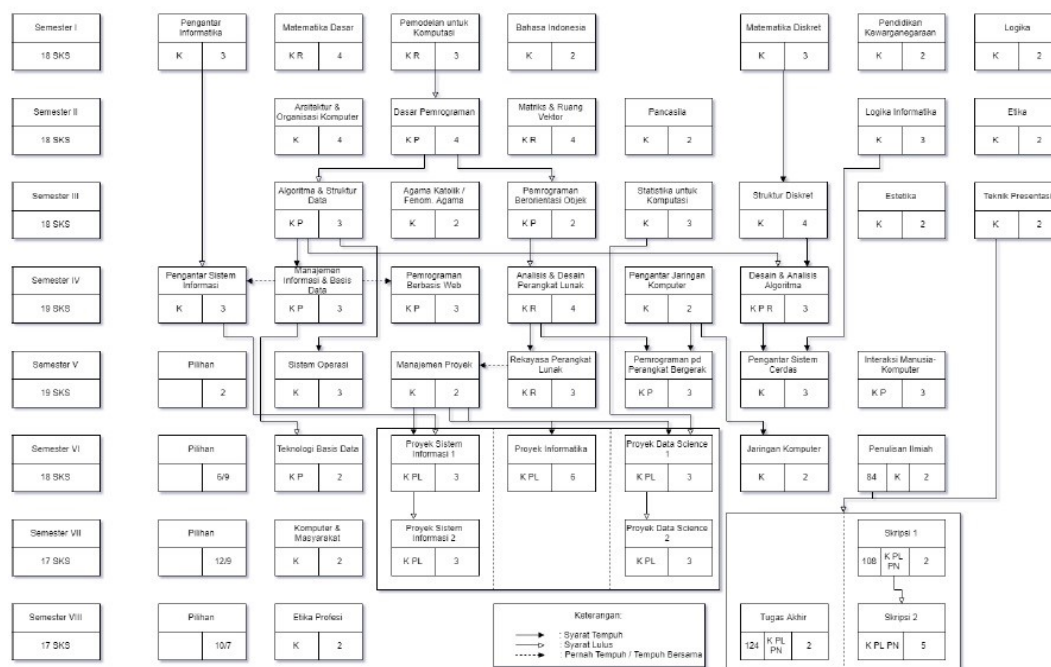
Bobot	Deskripsi
0.25	<ul style="list-style-type: none"> Minimal 1 tugas berbentuk pembangunan program komputer Kuliah tidak berpraktikum
0.5	<ul style="list-style-type: none"> Minimal setengah dari tugas yang diberikan berbentuk pembangunan program komputer Kuliah tidak berpraktikum atau yang berfokus pada analisis
0.75	<ul style="list-style-type: none"> Di luar tugas praktikum, ada tugas kuliah berupa pembangunan program komputer Kuliah berpraktikum atau merupakan kuliah skripsi
1	Kuliah berpraktikum dengan capaian pembelajaran adalah keahlian pemrograman atau merupakan kuliah proyek

Gambar 2.3: Rincian Bobot Pemrograman

2.1.3 Struktur Kurikulum

Berdasarkan hasil evaluasi Kurikulum 2013, struktur kurikulum dibangun dengan mendistribusikan mata kuliah dalam semester - semester. Struktur kurikulum ini terdiri atas 14 sks mata kuliah umum universitas, 100 sks mata kuliah wajib dan pilihan wajib prodi, dan 30 sks kuliah pilihan. Kuliah pilihan mulai diberikan di Semester 4, sedangkan kuliah pilihan wajib diberikan di Semester 6 dan 7. Kuliah pilihan wajib ini terdiri atas 3 mata kuliah, yaitu Proyek Informatika, Proyek Sistem Informasi, dan Proyek Data Science. Pohon Kurikulum 2018 dapat dilihat pada gambar 2.4.

POHON KURIKULUM 2018 TEKNIK INFORMATIKA UNPAR



Gambar 2.4: Pohon kurikulum

Penyusunan struktur kurikulum ini dilakukan dengan memperhatikan hal-hal berikut:

- Beban kredit persemester dibatasi maksimum 19 sks.
- Capaian pembelajaran yang ingin dicapai pada satu semester harus dapat mendukung capaian pembelajaran yang ingin dicapai di semester berikutnya.

- Rangkaian mata kuliah, di mana peletakan mata kuliah dasar dan prasyarat harus tepat sehingga dapat mendukung proses pembelajaran dan pemahaman mata kuliah di tahap selanjutnya. Rangkaian mata kuliah ini diberikan pada Tabel 2.5.

KBI/Pengampu	Mata Kuliah	Semester	Jumlah sks
Universitas	Bahasa Indonesia	1	14
	Pendidikan Kewarganegaraan	1	
	Logika	1	
	Etika	2	
	Estetika	2	
	Agama Katolik/Fenomenologi Agama	3	
	Pancasila	3	
Prodi	Teknik Presentasi	3	16
	Penulisan Ilmiah	6	
	Komputer dan Masyarakat	7	
	Etika Profesi	8	
	Skripsi 1	7	
	Skripsi 2	8	
Teori Komputasi	Tugas Akhir	8	53
	Pemodelan untuk Komputasi	1	
	Matematika Dasar	1	
	Pengantar Informatika	1	
	Matematika Diskret	1	
	Dasar Pemrograman	2	
	Logika Informatika	2	
	Matriks dan Ruang Vektor	2	
	Algoritma dan Struktur Data	3	
	Struktur Diskret	3	
	Pemrograman Berorientasi Objek	3	
	Statistika untuk Komputasi	3	
	Analisis dan Desain Perangkat Lunak	4	
	Desain dan Analisis Algoritma	4	
	Pengantar Sistem Cerdas	5	
	Interaksi Manusia Komputer	5	
Sistem Terdistribusi	Proyek Informatika	6	17
	Arsitektur dan Organisasi Komputer	2	
	Pemrograman Berbasis Web	4	
	Pengantar Jaringan Komputer	4	
	Sistem Operasi	5	
	Pemrograman pada Perangkat Bergerak	5	
Sistem Informasi	Jaringan Komputer	6	32
	Manajemen Informasi dan Basis Data	4	
	Pengantar Sistem Informasi	4	
	Rekayasa Perangkat Lunak	5	
	Manajemen Proyek	5	
	Teknologi Basis Data	6	
	Proyek Sistem Informasi 1	6	
	Proyek Data Science 1	6	
	Proyek Sistem Informasi 2	7	
	Proyek Data Science 2	7	

Gambar 2.5: Mata kuliah wajib sesuai bidang keilmuan

Secara umum, terdapat 4 jenis mata kuliah pada Kurikulum 2018, yaitu mata kuliah wajib, pilihan, pilihan wajib, dan sertifikasi. Keempat jenis mata kuliah ini dijelaskan pada bagian-bagian berikutnya. Selain itu, pada kurikulum 2018, diperkenalkan program, di mana masing-masing program terdiri atas banyak mata kuliah pilihan. Dengan cara ini, saat lulus, mahasiswa memiliki titik berat keahlian atau spesialisasi di bidang ilmu tertentu.

Pada Tabel 2.7 Semester 7, dapat dilihat bahwa jumlah mata kuliah wajib berkisar antara 2-3 buah dan kuliah pilihan 9-12 buah. Hal ini disebabkan adanya mata kuliah pilihan wajib jalur proyek yang dapat diambil sejak Semester 6. Jika mahasiswa memilih jalur proyek informatika, maka di Semester 7 mata kuliah wajib yang harus diambil adalah 2 buah dengan 12 sks kuliah pilihan. Di kasus ini, mahasiswa dapat mengambil 4 sks kuliah pilihan di Semester 6. Sementara itu, mahasiswa memilih jalur proyek sistem informasi, di Semester 7 mata kuliah wajib yang harus diambil adalah 3 buah dengan 9 sks kuliah pilihan. Di kasus ini, mahasiswa dapat mengambil 7 sks kuliah pilihan di Semester 6.

No	Kode	Mata Kuliah	Bobot Pemrograman	SKS
Semester 1				
1	AIF181101-03	Pemodelan untuk Komputasi	0.25	3
2	AIF181103-04	Matematika Dasar		4
3	AIF181105-02	Pengantar Informatika		2
4	AIF181107-03	Matematika Diskret		3
5	MKU180130-02	Bahasa Indonesia		2
6	MKU180110-02	Pendidikan Kewarganegaraan		2
7	MKU180120-02	Logika		2
Wajib: 18 sks, Pilihan: -				
Semester 2				
1	AIF181100-04	Dasar Pemrograman	1	4
2	AIF181202-04	Arsitektur dan Organisasi Komputer		4
3	AIF181104-03	Logika Informatika	0.25	3
4	AIF181106-03	Matriks dan Ruang Vektor	0.25	3
5	MKU180240-02	Etika		2
6	MKU180250-02	Pancasila		2
Wajib: 18 sks, Pilihan: -				
Semester 3				
1	AIF182101-03	Algoritma dan Struktur Data	0.75	3
2	AIF182103-04	Struktur Diskret	0.25	4
3	AIF182105-02	Pemrograman Berorientasi Objek	1	2
4	AIF182007-02	Teknik Presentasi		2
5	AIF182109-03	Statistika untuk Komputasi	0.25	3
6	MKU180370-02 / MKU180380-02	Agama Katolik/Fenomenologi Agama		2
7	MKU180360-02	Estetika		2
Wajib: 18 sks, Pilihan: -				
Semester 4				
1	AIF182100-04	Analisis dan Desain Perangkat Lunak	0.5	4
2	AIF182302-04	Manajemen Informasi dan Basis Data	0.75	4
3	AIF182204-03	Pemrograman Berbasis Web	1	3
4	AIF182106-03	Desain dan Analisis Algoritma	0.5	3
5	AIF182308-03	Pengantar Sistem Informasi	0.25	3
6	AIF182210-02	Pengantar Jaringan Komputer		2
Wajib: 19 sks, Pilihan: -				
Semester 5				
1	AIF183201-03	Sistem Operasi	0.25	3
2	AIF183303-03	Rekayasa Perangkat Lunak		3
3	AIF183305-02	Manajemen Proyek		2
4	AIF183107-03	Pengantar Sistem Cerdas	0.25	3

Gambar 2.6: Struktur Kurikulum 2018 (a)

No	Kode	Mata Kuliah	Bobot Pemrograman	SKS
5	AIF183209-03	Pemrograman pada Perangkat Bergerak	1	3
6	AIF183111-03	Interaksi Manusia Komputer	0.5	3
7	-	Pilihan		2
Wajib: 17 sks, Pilihan: 2 sks				
Semester 6				
1	AIF183300-02	Teknologi Basis Data	0.75	2
2	AIF183002-02	Penulisan Ilmiah		2
3	AIF183204-02	Jaringan Komputer	0.25	2
4	AIF183106-06	Proyek Informatika	1	6
	AIF183308-03	Proyek Sistem Informasi 1	1	3
	AIF183310-03	Proyek Data Science 1	1	3
5	-	Pilihan		6
	-	Pilihan		9
Wajib: 12/9 sks, Pilihan: 6/9 sks				
Semester 7				
1	AIF184001-03	Skripsi 1		3
2	AIF184303-03	Proyek Sistem Informasi 2	1	3
	AIF184305-03	Proyek Data Science 2	1	3
3	AIF184005-02	Komputer dan Masyarakat		2
4	-	Pilihan		12
	-	Pilihan		9
Wajib: 5/8 sks, Pilihan: 12/9 sks				
Semester 8				
1	AIF184000-02	Etika Profesi		2
2	AIF184002-05	Skripsi 2	0.75	5
	AIF184004-08	Tugas Akhir	0.75	8
3	-	Pilihan		10/7
Wajib: 7/10 sks, Pilihan: 10/7 sks				

Gambar 2.7: Struktur Kurikulum 2018 (b)

2.2 Prasyarat Mata Kuliah

Di Prodi Teknik Informatika terdapat 2 jenis prasyarat, yaitu prasyarat lulus dan prasyarat tempuh. Prasyarat lulus artinya seorang mahasiswa harus lulus mata kuliah prasyarat (nilai minimum D), baru dapat mengambil suatu mata kuliah, sedangkan prasyarat tempuh artinya seorang mahasiswa harus pernah menempuh mata kuliah prasyarat, sebelum dapat mengambil suatu mata kuliah. Rincian prasyarat mata kuliah wajib diberikan pada Tabel 2.8, sedangkan rincian prasyarat mata kuliah pilihan diberikan pada Tabel .

No	Kode	Mata Kuliah	Mata Kuliah Prasyarat Tempuh	Mata Kuliah Prasyarat Lulus
Semester 1				
1	AIF181101-03	Pemodelan untuk Komputasi		
2	AIF181103-04	Matematika Dasar		
3	AIF181105-02	Pengantar Informatika		
4	AIF181107-03	Matematika Diskret		
5	MKU180130-02	Bahasa Indonesia		
6	MKU180110-02	Pendidikan Kewarganegaraan		
7	MKU180120-02	Logika		
Semester 2				
1	AIF181100-04	Dasar Pemrograman		Mulai angkatan 2018: AIF181101-03
2	AIF181202-04	Arsitektur dan Organisasi Komputer		
3	AIF181104-03	Logika Informatika		
4	AIF181106-03	Matriks dan Ruang Vektor		
5	MKU180240-02	Etika		
6	MKU180360-02	Estetika		
Semester 3				
1	AIF182101-03	Algoritma dan Struktur Data		AIF181100-04
2	AIF182103-04	Struktur Diskret	AIF181107-03	
3	AIF182105-02	Pemrograman Berorientasi Objek		AIF181100-04
4	AIF182007-02	Teknik Presentasi		
5	AIF182109-03	Statistika untuk Komputasi		

Gambar 2.8: Daftar mata kuliah wajib beserta prasyaratnya (a)

No	Kode	Mata Kuliah	Mata Kuliah Prasyarat	
			Tempuh	Lulus
6	MKU180370-02 / MKU180380-02	Agama Katolik/Fenomenologi Agama		
7	MKU180250-02	Pancasila		
Semester 4				
1	AIF182100-04	Analisis dan Desain Perangkat Lunak		AIF182105-02
2	AIF182302-04	Manajemen Informasi dan Basis Data	AIF182101-03	
3	AIF182204-03	Pemrograman Berbasis Web	AIF182302-04 (bersamaan atau sudah tempuh)	
4	AIF182106-03	Desain dan Analisis Algoritma	AIF182103-04	AIF182101-03
5	AIF182308-03	Pengantar Sistem Informasi	AIF182302-04 (bersamaan atau sudah tempuh)	AIF181105-02
6	AIF182210-02	Pengantar Jaringan Komputer		
Semester 5				
1	AIF183201-03	Sistem Operasi	AIF182101-03	
2	AIF183303-03	Rekayasa Perangkat Lunak	AIF182100-04	
3	AIF183305-02	Manajemen Proyek	AIF183303-03 (bersamaan atau sudah tempuh)	
4	AIF183107-03	Pengantar Sistem Cerdas	AIF182106-03 AIF181104-03	
5	AIF183209-03	Pemrograman pada Perangkat Bergerak	AIF182210-02 AIF182100-04	
6	AIF183111-03	Interaksi Manusia Komputer		
Semester 6				
1	AIF183300-02	Teknologi Basis Data		AIF182302-04
2	AIF183002-02	Penulisan Ilmiah		
3	AIF183204-02	Jaringan Komputer	AIF182210-02	
4	AIF183106-06	Proyek Informatika	AIF183305-02	
	AIF183308-03	Proyek Sistem Informasi 1	AIF183305-02	AIF182308-03
	AIF183310-03	Proyek Data Science 1	AIF183305-02	AIF182109-03
Semester 7				
1	AIF184001-03	Skripsi 1		AIF183002-02 Sudah lulus 108 sks Mulai angkatan 2017: AIF183002-02 AIF182007-02

Gambar 2.9: Daftar mata kuliah wajib beserta prasyaratnya (b)

No	Kode	Mata Kuliah	Mata Kuliah Prasyarat	
			Tempuh	Lulus
				Sudah lulus 108 sks
2	AIF184303-03	Proyek Sistem Informasi 2		AIF183308-03
	AIF184305-03	Proyek Data Science 2		AIF183310-03
3	AIF184005-02	Komputer dan Masyarakat		
Semester 8				
1	AIF184000-02	Etika Profesi		
2	AIF184002-05	Skripsi 2		AIF184001-03 Jika diambil bersamaan dengan AIF184001-03 Prasyarat: lulus AIF183002-02 AIF182007-02 dan lulus 124 sks
3	AIF184004-08	Tugas Akhir		AIF183002-02 Sudah lulus 124 sks Mulai angkatan 2017: AIF183002-02 AIF182007-02 Sudah lulus 124 sks

Gambar 2.10: Daftar mata kuliah wajib beserta prasyaratnya (c)

No	Kode	Mata Kuliah	Prasyarat	
			Tempuh	Lulus
1	AIF182111-03	Pemrograman Kompetitif 1		AIF182101-03 (minimum C)
2	AIF182001-03	Penelitian 1		AIF181100-04
3	AIF182301-03	Pengantar Data Science		
4	AIF182112-03	Pemrograman Kompetitif 2		AIF182111-03 (minimum B)
5	AIF182102-03	Statistika dengan R		AIF182109-03
6	AIF182002-03	Penelitian 2		AIF181100-04
7	AIF183013-02	Kerja Praktek 1		
8	AIF183015-03	Pendidikan Pengabdian kepada Masyarakat		

Gambar 2.11: Daftar mata kuliah pilihan beserta prasyaratnya (a)

No	Kode	Mata Kuliah	Prasyarat	
			Tempuh	Lulus
9	AIF183117-02	Grafika Komputer	AIF181103-04	AIF182105-02
10	AIF183119-02	Keamanan Informasi		AIF181107-03
11	AIF183121-03	Pemrograman Kompetitif 3		AIF182112-03 (minimum B)
12	AIF183123-02	Topik Khusus Informatika 1		
13	AIF183113-03	Statistika Multivariat dengan R	AIF182102-03	
14	AIF183003-03	Penelitian 3		AIF181100-04
15	AIF183225-03	Sertifikasi Administrasi Jaringan Komputer 1		
16	AIF183229-02	Topik Khusus Sistem Terdistribusi 1		
17	AIF183331-03	Sistem e-Commerce		AIF182308-03
18	AIF183329-03	Basis Data dan Pemrograman SQL untuk Big Data		AIF182302-04 AIF182210-02
19	AIF183335-03	Pengantar Penambangan Data dengan Python	AIF182302-04 (atau tempuh bersama) AIF182109-03 (atau tempuh bersama)	
20	AIF183337-02	Topik Khusus Sistem Informasi 1		
21	AIF183339-02	Sertifikasi Perancangan dan Pemrograman Basis Data dengan Oracle	AIF182302-04	
22	AIF183341-03	Pola Komputasi Big Data	AIF182101-03 AIF182210-02	
23	AIF183143-03	Pemodelan Formal		AIF181104-03
24	AIF183147-02	Sertifikasi Dasar-dasar Java	AIF182105-02	
25	AIF183155-03	Metode Numerik		AIF181103-04 AIF181100-04
26	AIF183203-02	Internet of Things		AIF182210-02
27	AIF183010-03	Kerja Praktek 2		
28	AIF183112-02	Pengujian Perangkat Lunak		AIF183303-03
29	AIF183114-03	Algoritma Kriptografi	AIF183119-02	
30	AIF183116-02	Komputasi Paralel		AIF182101-03

Gambar 2.12: Daftar mata kuliah pilihan beserta prasyaratnya (b)

No	Kode	Mata Kuliah	Prasyarat	
			Tempuh	Lulus
31	AIF183120-03	Pemrograman Permainan Komputer		AIF182101-03 (minimum B)
32	AIF183122-03	Pemodelan Simulasi	AIF182101-03	
33	AIF183128-03	Topik Khusus Informatika 2		
34	AIF183232-03	Pemrograman Berbasis Web Lanjut		AIF182204-03 AIF182302-04
35	AIF183236-03	Sertifikasi Administrasi Jaringan Komputer 2		AIF183225-03
36	AIF183238-03	Topik Khusus Sistem Terdistribusi 2		
37	AIF183240-03	Sertifikasi Cyber Ops		AIF182210-02
38	AIF183342-03	Kewirausahaan Berbasis Teknologi		Sudah lulus 90 sks
39	AIF183346-03	Topik Khusus Sistem Informasi 2		
40	AIF183348-03	Sistem Kecerdasan Bisnis	AIF182302-04	
41	AIF183350-02	IBM Professional Data Science Certificate 1		AIF183335-03
42	AIF184007-04	Kerja Praktek 3		
43	AIF184109-03	Pembelajaran Mesin		AIF183107-03
44	AIF184115-02	Pencarian dan Temu Kembali Informasi		AIF181103-04
45	AIF184119-02	Perancangan Permainan Komputer		AIF182100-04 (minimum B) AIF183120-03 (minumum B)
46	AIF184123-03	Teknologi Mesin Pencari	AIF181106-03	
47	AIF184125-03	Pengolahan Bahasa Alami		AIF183107-03
48	AIF184129-03	Sertifikasi Administrasi Jaringan Komputer 3		AIF183236-03
49	AIF184235-03	Layanan Berbasis Web		AIF182204-03 AIF182302-04 AIF183204-02
50	AIF184339-03	Pengendalian dan Audit Teknologi Informasi	AIF182308-03	
51	AIF184341-03	Penambangan Data		AIF182101-03

Gambar 2.13: Daftar mata kuliah pilihan beserta prasyaratnya (c)

No	Kode	Mata Kuliah	Prasyarat	
			Tempuh	Lulus
52	AIF184337-03	IBM Professional Data Science Certificate 2		AIF183350-02
53	AIF184351-03	Analisis Big Data	AIF183335-03 AIF183341-03 (atau tempuh bersama)	
54	AIF184247-03	Jaringan Komputer Lanjut		AIF183204-02
55	AIF184006-05	Kerja Praktek 4		
56	AIF184106-02	Analisis Data Permainan Komputer		AIF184119-02 (minimum B)
57	AIF184116-02	Sistem Multi Agen	AIF183201-03 AIF183107-03	
58	AIF184222-03	Sertifikasi Administrasi Jaringan Komputer 4		AIF184129-03
59	AIF184334-03	Sistem Informasi Skala Besar		AIF182308-03
60	AIF184338-03	Manajemen Proses Bisnis	AIF182105-02 AIF182204-03	
61	AIF184332-03	Teknologi Big Data dan Cloud Computing	AIF184351-03	
62	AIF184352-03	IBM Professional Data Science Certificate 3		AIF184337-03
63	AIF184330-02	Big Data dan Machine Learning dengan Google Cloud Platform	AIF184351-03 (atau tempuh bersama)	
64	AIF184328-03	Data Science pada Domain Spesifik		AIF183113-03 atau AIF183335-03 atau AIF184351-03

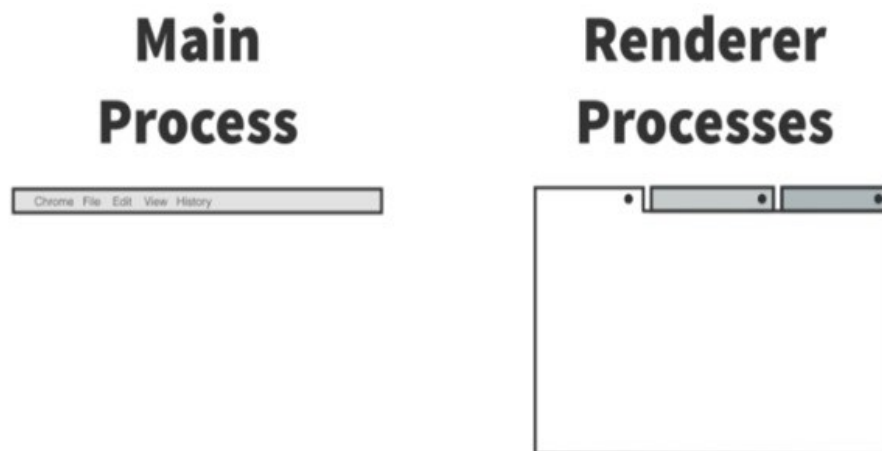
Gambar 2.14: Daftar mata kuliah pilihan beserta prasyaratnya (d)

2.3 Electron

Electron adalah sebuah kerangka kerja untuk mengembangkan aplikasi *desktop* asli lintas platform menggunakan teknologi web seperti *JavaScript*, *HTML*, dan *CSS*. Dengan menggunakan *electron* pengembang web yang terampil dapat mengembangkan aplikasi lintas platform semacam itu tanpa perlu keahlian khusus sistem operasi yang lain. Keunggulan dari kerangka kerja *electron* adalah bersifat *Open Source*, dipelihara di *Github* oleh komunitas dan bersifat *Cross Platform*, bisa dijalankan di *Mac*, *Windows* dan *Linux*. Aplikasi-aplikasi yang dibuat menggunakan *electron* seperti *Atom Editor*, *Visual Studio*, *Wordpress.com (Desktop Version)*, *Avocode* dan sebagainya.

Electron menggabungkan *browser web* yang lengkap: *chromium*, versi *open source* *Google Chrome*. Ini dibundel bersama dengan instruksi spesifik platform untuk memastikan bahwa segala sesuatu berperilaku tepat seperti yang diharapkan pengembang di semua sistem. Itu sebabnya versi *desktop slack* membutuhkan lebih dari 200MB ruang *hard drive* dengan sebagian besar *chrome* ada di dalamnya. Maka dari itu kelemahan dari kerangka kerja *electron* adalah boros *memory* dan memakan waktu yang sedikit lebih lama untuk menjalankannya, karena setiap aplikasi *electron* mengunduh bundel sebagian besar *chromium*, dan setiap aplikasi yang dijalankan mengeksekusi potongan kode yang baik. Tidak ada pembagian sumber daya di sini seperti yang ada dengan aplikasi asli, yang berarti aplikasi *electron* akan mengambil lebih banyak ruang dan memori *hard drive* daripada aplikasi yang dikembangkan dengan platform kita secara khusus.

Aplikasi yang dibangun dengan *electron* terdiri dari dua jenis proses, yaitu proses utama dan beberapa proses *renderer* dengan mengikuti pola *master-slave*. Proses utama aplikasi adalah *master* dan beberapa proses *renderer* adalah budak yang berkomunikasi menggunakan *Inter-Process Communication (IPC)*. Gambar 2.15 menunjukkan bahwa ini serupa dengan *browser Chromium* yang memiliki satu jendela utama dan banyak tab (proses *renderer*).



Gambar 2.15: Sebuah proses utama dengan beberapa proses *renderer*. Proses ini mirip dengan jendela *browser chromium* dengan banyak tab

Secara khusus, *electron* akan melihat masukan awal di manifest *package.json* yang dimasukkan ke dalam proyek untuk menentukan titik masuk aplikasi yang dijalankan sebagai proses utama.

Proses utama bertanggung jawab untuk merespons peristiwa siklus hidup aplikasi seperti *starting up*, *quitting*, *preparing to quit*, *going to the background*, *coming to the foreground* dan sebagainya. Proses utama juga bertanggung jawab untuk berkomunikasi dengan API sistem operasi asli. Misalnya untuk menampilkan dialog untuk membuka atau menyimpan *file*. Proses utama juga dapat membuat dan menghancurkan proses *renderer* menggunakan modul *BrowserWindow electron*.

Proses *renderer* dapat memuat halaman web untuk menampilkan *GUI*. Setiap proses memanfaatkan arsitektur multi-process *chromium* dan berjalan di threadnya sendiri. Tidak seperti laman web biasa, ada akses ke semua Node APIs dalam proses *renderer*, yang memungkinkan developers

- 1 memanfaatkan modul asli dan tingkat interaksi sistem yang lebih rendah.
- 2 Untuk cara membuat aplikasi dengan *electron* adalah sebagai berikut :
- 3 1. Install terlebih dahulu Node.js dan npm (pakai versi terakhir LTS yang tersedia).
- 4 2. Cek versinya dengan mengetik `node -v` dan `npm -v` pada command prompt.
- 5 3. Buat folder dan inisialisasi paket npm dengan mengetik `mkdir namaFolder` lalu ketik `cd`
- 6 `namaFolder` setelah itu ketik `npm init` pada *command prompt*. Maka isi `package.json` akan
- 7 seperti pada gambar 2.16.

```
{
  "name": "my-electron-app",
  "version": "1.0.0",
  "description": "Hello World!",
  "main": "main.js",
  "author": "Jane Doe",
  "license": "MIT"
}
```

Gambar 2.16: Package.json

- 8 4. Install aplikasi *electron* ke dalam folder yang telah dibuat dengan cara mengetik `npm install`
- 9 `--save-dev electron` pada command prompt.
- 10 5. Pada *script file* `package.json` tambahkan *command start* seperti pada gambar 2.17.

```
{
  "scripts": {
    "start": "electron ."
  }
}
```

Gambar 2.17: Start command

- 11 6. Jalankan aplikasi *electron* dengan mengetik `npm start` pada command prompt.

2.4 Vis.Js

2.4.1 Timeline

2.4.2 Network

2.5 Github

Github adalah layanan *hos web* bersama untuk proyek pengembangan perangkat lunak yang menggunakan sistem kendali versi Git dan layanan *hosting* internet secara gratis. Github berfungsi sebagai manajemen proyek dengan sistem *versioning code* yang digunakan untuk developer dalam bekerja bersama-sama dengan rekan, merencanakan proyek, dan melacak pekerjaan. Selain berfungsi untuk menyimpan repositori dan alat kolaborasi proyek, Github juga bisa sebagai sarana dalam mendapatkan pekerjaan. Ada banyak *programmer* yang mengajak *programmer* sesama pengguna Github untuk bekerjasama dalam mengerjakan suatu proyek. Hal ini bisa terjadi karena di Github memiliki halaman profil yang berisi data pribadi seperti foto, email, jumlah repositori yang dimiliki dan jumlah *follower* (semakin banyak akan menjadi pertimbangan bagi programmer lain untuk merekrut). Di Github dapat diatur status proyek menggunakan fitur *public* atau *private*. Jika *public* artinya project dapat dilihat, diubah, dan diunduh oleh siapa saja. Jika *private* artinya project hanya bisa dilihat, diubah dan diunduh oleh pemilik dan tim project yang *diinvite* pada project, tetapi fitur *private* dibatasi hanya untuk tiga kontributor, jika ingin lebih dari tiga maka diharuskan untuk *upgrade* ke Github versi berbayar (GitHub Pro). Untuk Cara menggunakan Github adalah seperti berikut :

1. Login

Sebelum mulai untuk menggunakan Github pastikan bahwa kita sudah login ke website Github terlebih dahulu, jika belum memiliki akun sebaiknya membuat akun terlebih dahulu. Saat membuat akun, kita akan diminta untuk mengisi beberapa data pribadi seperti nama dan alamat email, kemudian akan diminta untuk melakukan verifikasi secara langsung.

2. Membuat repositori

Setelah masuk ke akun Github, kita bisa langsung mulai untuk membuat repositori. Caranya dengan mengklik tombol New yang terdapat pada menu repositori, selanjutnya kita akan diarahkan untuk menuju halaman pembuatan repositori. Membuat repositori baru tentu membutuhkan beberapa informasi awal seperti nama repositori, deskripsi yang berfungsi untuk memberi penjelasan terkait repositori yang kita buat, jenis repositori yang terdiri dari *public* atau *private*. Pengaturan yang bersifat publik tentu bisa membuat orang lain mengakses repositori yang kita buat, sebaliknya jika tidak ingin ada orang lain yang melakukan akses terhadap repositori milik kita maka gunakan pengaturan *private*. Jika ketiga data tersebut sudah terpenuhi, kita bisa melanjutkan dengan mengklik *create repository*.

3. Membuat *local* folder

Membuat folder lokal yang terdapat pada komputer tentu berfungsi untuk melakukan penyimpanan *update file* dari repositori yang dibuat ketika terjadi perubahan tertentu, saat folder tersebut berhasil dibuat kita bisa membuka folder tersebut kemudian klik kanan untuk memilih menu Git Bash Here. Sehingga folder tersebut akan berubah menjadi repositori.

4. Memasukkan file ke repositori

Untuk dapat menambahkan dan memasukkan file ke repositori yang sudah dibuat, kita bisa membuat file di dalam folder tersebut kemudian dilanjutkan dengan membuka bagian Git Bash sesuai perintah.

5. Membuat commit

Fitur commit berfungsi untuk menambahkan *update file* dan komentar, juga membantu rekan tim yang ada untuk mengkonfirmasi *update file* di proyek yang sedang dikerjakan.

6. *Remote repository*

Kita dapat mengupload file yang sudah dibuat di *local disk* dengan menggunakan *remote*

repository.

7. Melakukan push ke github

Kita dapat melakukan *push* yang berfungsi untuk mengunggah hasil akhir dari pekerjaan kita, setelah itu kita bisa melakukan pengecekan pada repositori untuk memastikan bahwa file-file yang ditambahkan sudah masuk dan sesuai dengan yang diinginkan.

8. Melakukan pull dari github

Saat kita ingin melanjutkan pekerjaan kita, sebaiknya kita melakukan *pull* terlebih dahulu yang berfungsi agar file pekerjaan yang terdapat di *local folder* sudah diperbaharui dengan hasil pekerjaan tim kita.

Github menggunakan sistem pengontrol versi yang disebut dengan Git. Git adalah salah satu sistem pengontrol versi (*Version Control System*) pada proyek perangkat lunak yang diciptakan oleh Linus Torvalds pada tahun 2005. Software ini bersifat open source dan bisa di-download untuk *Linux*, *Windows*, *Solaris* dan *Mac*. Terdapat beberapa perintah dasar Git yang sering digunakan dalam pengerjaan proyek, seperti berikut :

1. git config

berfungsi untuk mengatur konfigurasi tertentu sesuai keinginan kita, seperti : e-mail, username, format file, dan sebagainya.

Contoh perintah untuk mengatur email : `git config --global user.email yourEmail`

2. git init

berfungsi untuk membuat *repository* baru.

Caranya : `git init`

3. git add

berfungsi untuk menambahkan file ke index.

Contoh perintah untuk menambahkan file bernama temp.txt yang ada di direktori lokal ke index : `git add temp.txt`

4. git clone

berfungsi untuk membuat salinan repositori lokal.

Contoh perintah untuk membuat salinan kerja dari server : `git clone username@host:/path/to/repository`

5. git commit

berfungsi untuk melakukan *commit* perubahan ke head, perintah ini dilakukan sebelum melakukan perintah *push*.

Contoh perintah untuk melakukan *commit* : `git commit -m "Keterangan commit"`

6. git push

berfungsi untuk mengirimkan perubahan ke *remote* repositori.

Contoh perintah untuk melakukan *push* : `git push`

7. git pull

berfungsi untuk menggabungkan semua perubahan yang ada di *remote* repositori ke direktori lokal.

Contoh perintah untuk melakukan *pull* : `git pull`

8. git branch

berfungsi untuk untuk me-list, membuat atau menghapus *branch*.

Contoh perintah untuk menampilkan semua branch yang ada di repositori : `git branch`

9. git checkout

berfungsi untuk membuat branch atau untuk berpindah diantaranya.

Contoh perintah untuk membuat branch baru : `git checkout -b <branch-name>`

Contoh perintah untuk berpindah dari branch satu ke lainnya : `git checkout <branch-name>`

10. git merge

berfungsi untuk menggabungkan sebuah branch ke branch aktif.

Contoh perintah untuk melakukan *merge* : `git merge`

11. git fetch

berfungsi untuk menampilkan semua objek dari *remote repository* yang tidak berada di

1 direktori kerja lokal.

2 Contoh perintah untuk melakukan fetch : `git fetch origin`

3 12. `git revert`

4 berfungsi untuk membatalkan *commit* yang ada.

5 Contoh perintah untuk melakukan *revert* : `git revert "commit number"`

6 *Version control system* adalah sistem yang mencatat semua perubahan yang dilakukan pada file
7 sehingga semua riwayatnya akan terekam dan bisa dilihat kembali nanti. Saat developer membuat
8 proyek baru, mereka selalu pembaharuan terhadap kodenya. Bahkan, setelah proyeknya online,
9 developer tetap harus memperbarui versinya, memperbaiki bug, menambahkan fitur baru, dan lain
10 sebagainya. *Version control system* membantu developer melacak perubahan yang mereka lakukan
11 terhadap basis kode. Tak hanya itu, sistem ini juga mencatat siapa saja yang membuat perubahan
12 serta memulihkan kode yang telah dihapus atau dimodifikasi. Dengan menggunakan sistem tersebut,
13 maka tidak ada kode yang saling tertimpa.

DAFTAR REFERENSI

LAMPIRAN A

KODE PROGRAM

Kode A.1: MyCode.c

```
1 // This does not make algorithmic sense,
2 // but it shows off significant programming characters.
3
4 #include<stdio.h>
5
6 void myFunction( int input, float* output ) {
7     switch ( array[i] ) {
8         case 1: // This is silly code
9             if ( a >= 0 || b <= 3 && c != x )
10                 *output += 0.005 + 20050;
11             char = 'g';
12             b = 2^n + ~right_size - leftSize * MAX_SIZE;
13             c = (--aaa + &daa) / (bbb++ - ccc % 2 );
14             strcpy(a,"hello_$@?");
15         }
16         count = ~mask | 0x00FF00AA;
17     }
18 }
19
20 // Fonts for Displaying Program Code in LATEX
21 // Adrian P. Robson, nepsweb.co.uk
22 // 8 October 2012
23 // http://nepsweb.co.uk/docs/progfonts.pdf
```

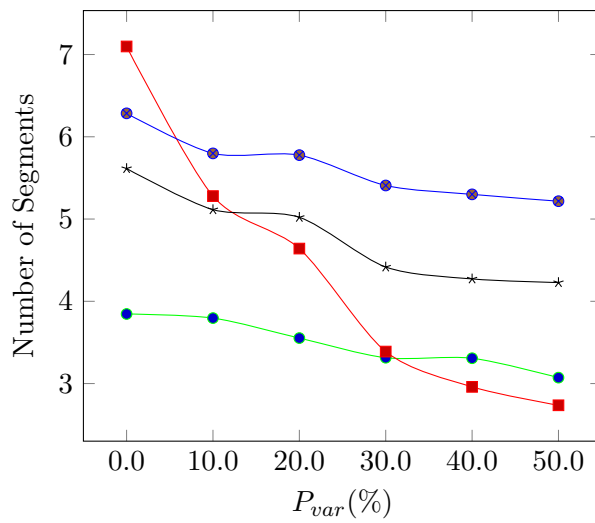
Kode A.2: MyCode.java

```
1 import java.util.ArrayList;
2 import java.util.Collections;
3 import java.util.HashSet;
4
5 //class for set of vertices close to furthest edge
6 public class MyFurSet {
7     protected int id; //id of the set
8     protected MyEdge FurthestEdge; //the furthest edge
9     protected HashSet<MyVertex> set; //set of vertices close to furthest edge
10    protected ArrayList<ArrayList<Integer>> ordered; //list of all vertices in the set for each trajectory
11    protected ArrayList<Integer> closeID; //store the ID of all vertices
12    protected ArrayList<Double> closeDist; //store the distance of all vertices
13    protected int totaltrj; //total trajectories in the set
14
15    /*
16     * Constructor
17     * @param id : id of the set
18     * @param totaltrj : total number of trajectories in the set
19     * @param FurthestEdge : the furthest edge
20     */
21    public MyFurSet(int id,int totaltrj,MyEdge FurthestEdge) {
22        this.id = id;
23        this.totaltrj = totaltrj;
24        this.FurthestEdge = FurthestEdge;
25        set = new HashSet<MyVertex>();
26        ordered = new ArrayList<ArrayList<Integer>>();
27        for (int i=0;i<totaltrj;i++) ordered.add(new ArrayList<Integer>());
28        closeID = new ArrayList<Integer>(totaltrj);
29        closeDist = new ArrayList<Double>(totaltrj);
30        for (int i = 0;i <totaltrj;i++) {
31            closeID.add(-1);
32            closeDist.add(Double.MAX_VALUE);
33        }
34    }
35 }
36 }
```


LAMPIRAN B

HASIL EKSPERIMEN

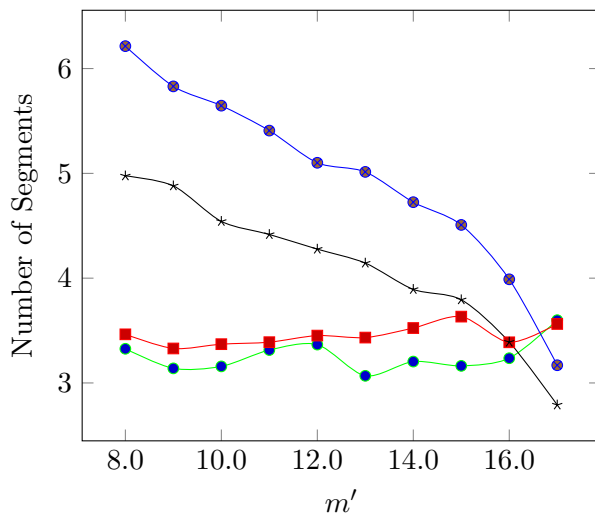
Hasil eksperimen berikut dibuat dengan menggunakan TIKZPICTURE (bukan hasil excel yg diubah ke file bitmap). Sangat berguna jika ingin menampilkan tabel (yang kuantitasnya sangat banyak) yang datanya dihasilkan dari program komputer.



Gambar B.1: Hasil 1



Gambar B.2: Hasil 2



Gambar B.3: Hasil 3



Gambar B.4: Hasil 4