

«SKRIPSI/TUGAS AKHIR»

«JUDUL BAHASA INDONESIA»



«Nama Lengkap»

NPM: «10 digit NPM UNPAR»

PROGRAM STUDI «MATEMATIKA/FISIKA/TEKNIK INFORMATIKA»
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
«tahun»

«FINAL PROJECT/UNDERGRADUATE THESIS»

«JUDUL BAHASA INGGRIS»



«Nama Lengkap»

NPM: «10 digit NPM UNPAR»

DEPARTMENT OF «MATHEMATICS/PHYSICS/INFORMATICS»
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
«tahun»

LEMBAR PENGESAHAN

«JUDUL BAHASA INDONESIA»

«Nama Lengkap»

NPM: «10 digit NPM UNPAR»

Bandung, «tanggal» «bulan» «tahun»

Menyetujui,

Pembimbing Utama

Pembimbing Pendamping

«pembimbing utama/1»

«pembimbing pendamping/2»

Ketua Tim Penguji

Anggota Tim Penguji

«penguji 1»

«penguji 2»

Mengetahui,

Ketua Program Studi

«Ketua Program Studi»

PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa «skripsi/tugas akhir» dengan judul:

«JUDUL BAHASA INDONESIA»

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal «tanggal» «bulan» «tahun»

Meterai Rp. 6000

«Nama Lengkap»
NPM: «10 digit NPM UNPAR»

ABSTRAK

«Tuliskan abstrak anda di sini, dalam bahasa Indonesia»

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Kata-kata kunci: «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Indonesia»

ABSTRACT

«Tuliskan abstrak anda di sini, dalam bahasa Inggris»

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Keywords: «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Inggris»

«kepada siapa anda mempersembahkan skripsi ini...?»

KATA PENGANTAR

«Tuliskan kata pengantar dari anda di sini ...»

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Bandung, «bulan» «tahun»

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metodologi	2
1.6 Sistematika Pembahasan	3
2 LANDASAN TEORI	5
2.1 <i>Outlook.com Calendar API</i>	5
2.2 <i>Slack API</i>	12
2.3 <i>Node.js</i>	12
2.4 Cron	12
2.4.1 Tabel	12
2.4.2 Kutipan	12
2.4.3 Gambar	13
A KODE PROGRAM	17
B HASIL EKSPERIMEN	19

DAFTAR GAMBAR

2.1	Gambar <i>Serpentes</i> dalam format png	14
2.2	Ular kecil	14
2.3	<i>Serpentes</i> betina	15
B.1	Hasil 1	19
B.2	Hasil 2	19
B.3	Hasil 3	19
B.4	Hasil 4	19

DAFTAR TABEL

2.1	Tabel contoh	12
2.2	Tabel bewarna(1)	12
2.3	Tabel bewarna(2)	12

BAB 1

PENDAHULUAN

1.1 Latar Belakang

*Outlook.com*¹ adalah sebuah kumpulan aplikasi berbasis *web* seperti *webmail*, *contacts*, *tasks*, dan *calendar* dari *Microsoft*. Fitur *calendar* sendiri pertama dirilis pada 14 Januari 2008 dengan nama *Windows Live Calendar*. Fitur *calendar* yang dimiliki oleh *Outlook.com Calendar* sendiri memiliki tampilan yang mirip dengan aplikasi kalender *desktop* pada umumnya. Seperti layaknya kalender digital pada umumnya, aplikasi *Outlook.com Calendar* juga bisa menambahkan, menyimpan, dan memodifikasi *event-event* yang dimasukkan oleh pengguna dan bisa dibuka dimana saja karena bersifat *online*.

*Slack*² adalah alat dan layanan kolaborasi tim berbasis *cloud*. *Slack* merupakan singkatan dari “*Searchable Log of All Conversation and Knowledge*”. Cara melakukan kolaborasi di aplikasi *Slack* sendiri adalah dengan komunitas, grup, atau tim bergabung ke dalam URL yang spesifik. *Room chat* yang terdapat di dalam aplikasi *Slack* biasa disebut dengan *Channel*. Ada 2 jenis *channel* di dalam aplikasi *Slack* yaitu *Public Channel* dan *Private Channel*. Pada *Public Channel*, seluruh anggota dari tim atau komunitas bisa masuk dan bergabung untuk berkomunikasi di *channel* tersebut. Tetapi pada *Private Channel*, hanya anggota yang diizinkan, ditambahkan, dan diundang oleh admin atau pembuat *channel* sajalah yang bisa ikut serta dalam berkomunikasi di dalam *channel* tersebut. *Slack* juga terintegrasi dengan banyak layanan pihak ketiga seperti contohnya adalah *Google Drive*, *Github*, *Trello*, *Dropbox*, dan masih banyak lagi layanan pihak ketiga yang bisa diintegrasikan dengan *Slack* itu sendiri.

Pada *Slack* terdapat status pengguna yang bisa diganti oleh pengguna tersebut untuk menggambarkan keadaan pengguna saat ini. Sebagai *default*, status bisa menggambarkan jika pengguna sedang “*In a meeting*” atau sedang “*Out Sick*”, dan banyak status *default* yang disediakan oleh *Slack*. Serta status pun bisa diisi oleh pengguna secara sendiri sesuai dengan apa yang ingin dituliskan oleh penggunanya. Disinilah yang menjadi latar belakang dirancangnya perangkat lunak ini yaitu terkadang pengguna lupa untuk mengganti status menjadi “*In a meeting*” saat pengguna memiliki jadwal untuk melakukan *meeting*, sehingga status di pengguna masih terlihat tersedia oleh user lain yang membuat tidak mengetahui sang pengguna sedang dalam keadaan *meeting* yang tidak dapat diganggu. Di saat seperti ini, kemungkinan untuk *meeting* terganggu oleh adanya *chat* yang masuk lewat *Slack* pun cukup tinggi.

Pada skripsi ini akan dibuat perangkat lunak yang akan membaca jadwal dari pengguna yang dicantumkan di aplikasi *Outlook.com Calendar*, lalu akan diintegrasikan kepada aplikasi *Slack* dengan mengubah dan mengganti status sesuai dengan jadwal yang telah didapatkan dari data di *Outlook.com Calendar* dari pengguna.

Perangkat lunak ini akan dibuat menggunakan *Node.js*³ dan akan memiliki 2 fungsi utama yaitu yang pertama adalah membaca dan mencatat jadwal dari *Outlook.com Calendar* yang membutuhkan adanya *Outlook.com Calendar API*. Lalu perangkat lunak ini juga memiliki fungsi kedua yaitu

¹<https://outlook.live.com/>

²<https://slack.com/>

³<https://nodejs.org>

mengubah status ke aplikasi *Slack* dengan menggunakan *Slack API*. Nantinya kedua fungsi dari perangkat lunak ini akan dijalankan secara berkala.

1.2 Rumusan Masalah

Pada perangkat lunak ini, terdapat rumusan masalah sebagai berikut:

1. Bagaimana cara mendapatkan data *event* dari *Outlook.com Calendar*?
2. Bagaimana mengubah status pada aplikasi *Slack* menggunakan *Slack API*?
3. Bagaimana cara membuat program agar dapat mengubah status pada aplikasi *Slack* di jadwal yang telah didapat dari aplikasi *Outlook.com Calendar*?

1.3 Tujuan

Adapun pada perangkat lunak ini memiliki tujuan sebagai berikut:

1. Mengetahui cara mendapatkan data *event* dari *Outlook.com Calendar*.
2. Mengetahui cara mengubah status pada aplikasi *Slack* menggunakan *Slack API*.
3. Membuat program agar dapat mengubah status pada aplikasi *Slack* di jadwal yang telah didapat dari aplikasi *Outlook.com Calendar*.

1.4 Batasan Masalah

Perancangan perangkat lunak ini dibuat berdasarkan batasan-batasan sebagai berikut:

1. Program ini dijalankan secara berkala sehingga tidak dapat menjalankan *update* status secara *real-time*.

1.5 Metodologi

Berikut adalah metodologi yang akan digunakan dalam penelitian ini:

1. Melakukan studi literatur tentang *Outlook.com Calendar*, *Slack*, dan juga *Node.js*.
2. Menggunakan aplikasi *Slack* di lingkungan tempat penulis melakukan magang.
3. Menganalisis aplikasi-aplikasi sejenis.
4. Melakukan analisis cara melakukan *synchronize* dengan aplikasi *Outlook.com Calendar* secara berkala.
5. Merancang bagian dari perangkat lunak yang akan mengambil data-data *event* dari *Outlook.com Calendar* dan yang bertugas untuk mengubah status pada *Slack* saat waktu sesuai dengan jadwal yang sudah tercatat dari *Outlook.com Calendar*.
6. Mengimplementasi bagian pengambilan data dari *Outlook.com Calendar* dan juga bagian mengatur status pada *Slack* sesuai jadwal yang telah diambil kepada perangkat lunak Integrasi *Outlook.com Calendar* dengan *Slack* serta melakukan pengujian terhadap fitur yang telah diimplementasikan.

1.6 Sistematika Pembahasan

Setiap bab dalam penelitian ini akan memiliki sistematika pembahasan yang dijelaskan ke dalam poin-poin sebagai berikut:

1. Bab 1: Pendahuluan, yaitu menjelaskan gambaran umum dari penelitian ini yang berisi tentang latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi, dan sistematika pembahasan.
2. Bab 2: Dasar Teori, yaitu menjelaskan dan membahas teori-teori yang dibutuhkan dan mendukung berjalannya penelitian ini. Meliputi tentang *Outlook.com Calendar API*, *Slack API*, *Node.js*, dan juga *Crontab*.
3. Bab 3: Analisis, yaitu membahas mengenai analisis masalah. Berisi tentang analisis aplikasi sejenis, analisis cara pengambilan data dari *Outlook.com Calendar*, dan proses pengubahan status menggunakan program pada *Slack*.
4. Bab 4: Perancangan, yaitu membahas mengenai perancangan aplikasi untuk melakukan sinkronisasi antara *Outlook.com Calendar* dengan *Slack*.
5. Bab 5: Implementasi dan Pengujian, yaitu membahas mengenai implementasi dari aplikasi yang telah dirancang dan juga pengujian aplikasi tersebut.
6. Bab 6: Kesimpulan dan Saran, yaitu berisi tentang kesimpulan dari penelitian ini dan juga saran yang dapat diberikan untuk penelitian selanjutnya.

BAB 2

LANDASAN TEORI

2.1 *Outlook.com Calendar API*

Untuk mengakses dan mendapatkan data yang terdapat di *Outlook.com Calendar*, dibutuhkan *Microsoft Graph API*. *Microsoft Graph* itu sendiri adalah gerbang untuk mendapatkan data-data yang terdapat di *Microsoft 365* dan *Outlook.com Calendar* termasuk di dalam layanan *Office 365*. *Microsoft Graph API* adalah sebuah *webservice* yang memiliki satu buah *endpoint* untuk memperoleh data yang ada pada *Azure Active Directory*, layanan *Office 365* (*SharePoint*, *OneDrive*, *Outlook/Exchange*, *Microsoft Teams*, *OneNote*, *Planner*, dan *Excel*), layanan *Enterprise Mobility and Security* (*Identity Manager*, *Intune*, *Advanced Threat Analytics*, dan *Advanced Threat Protection*), layanan *Windows 10* (*activities* dan *devices*), dan *Education* yaitu menuju *endpoint* ke <https://graph.microsoft.com>. *Microsoft Graph* dapat dipergunakan untuk membangun suatu aplikasi dengan memanfaatkan konteks unik dari masing-masing penggunaanya. Aplikasi yang dapat dibangun dengan bantuan dari *Microsoft Graph* bisa jadi memiliki fungsi seperti:

- Melihat pertemuan/ *meeting* pengguna yang berikutnya sudah terjadwalkan serta membantu pengguna untuk mempersiapkan dengan memberikan informasi dari profil peserta.
- Melihat kepada kalender pengguna sehingga bisa merekomendasikan waktu yang tepat untuk pertemuan/ *meeting* selanjutnya.
- Mengubah file *Excel* yang berada di *OneDrive* pengguna secara *real-time* dan bisa melalui ponsel pengguna.
- Memantau perubahan yang ada di kalender pengguna sehingga bisa mengirimkan peringatan jika pengguna menghabiskan terlalu banyak waktu untuk melakukan rapat, serta bisa juga merekomendasikan *event-event* yang bisa didelegasikan atau dilewatkan berdasarkan dari seberapa relevannya *event* itu kepada pengguna.
- Membantu pengguna untuk bisa memilah informasi yang ditujukan untuk pekerjaan dengan informasi yang bersifat pribadi. Misalkan dengan mengelompokkan gambar-gambar pribadi yang masuk ke *OneDrive* pribadi serta memasukkan gambar-gambar pekerjaan ke *OneDrive for Business* pengguna.

Permintaan yang populer diminta pada *Microsoft Graph API* antara lain:

Operation	URL
GET my profile	https://graph.microsoft.com/v1.0/me
GET my files	https://graph.microsoft.com/v1.0/me/drive/root/children
GET my photo	https://graph.microsoft.com/v1.0/me/photo/\$value
GET my mail	https://graph.microsoft.com/v1.0/me/messages
GET my high importance email	https://graph.microsoft.com/v1.0/me/messages?\$filter=importance%20eq%20'high'
GET my calendar events	https://graph.microsoft.com/v1.0/me/events
GET my manager	https://graph.microsoft.com/v1.0/me/manager
GET last user to modify file foo.txt	https://graph.microsoft.com/v1.0/me/drive/root/children/foo.txt/lastModifiedByUser
GET Office365 groups I'm member of	https://graph.microsoft.com/v1.0/me/memberOf/\$/microsoft.graph.group?\$filter=groupTypes/any(a:a%20eq%20'unified')
GET users in my organiza- tion	https://graph.microsoft.com/v1.0/users
GET groups in my organi- zation	https://graph.microsoft.com/v1.0/groups
GET people rela- ted to me	https://graph.microsoft.com/v1.0/me/people
GET items tren- ding around me	https://graph.microsoft.com/beta/me/insights/trending
GET my notes	https://graph.microsoft.com/v1.0/me/onenote/notebooks

Dari *Microsoft Graph* ini, barulah dapat mengakses *Outlook.com Calendar API*. Cara untuk menggunakan membangun aplikasi menggunakan *API* dari *Microsoft* yang tersedia adalah:

Mendaftarkan aplikasi yang akan dibuat ke Azure AD.

Untuk Mendaftarkan aplikasi yang akan dibuat, ada beberapa langkah yaitu:

1. Masuk ke *Microsoft App Registration Portal* dengan menggunakan akun *Microsoft* atau akun sekolah atau akun kantor.

2. Pilih dan klik tombol bertuliskan “*Add an app*”.
3. Masukkan nama untuk aplikasi yang akan dibuat dan klik tombol “*Create application*”.
4. Salin “*application ID*”. *Application ID* ini nantinya dipergunakan untuk mengkonfigurasi aplikasinya.
5. Dibawah bagian “*Platforms*”, pilih tombol “*Add Platforms*” dan pilihlah *platform* yang sesuai untuk aplikasi.

Untuk *Native* atau *Mobile apps*:

- (a) Pilih “*Native Application*”.
- (b) Salin bagian “*Built-in redirect URI*”. Hal ini diperlukan untuk menkonfigurasi aplikasinya. *Redirect URI* ini disediakan identik untuk aplikasinya yang gunanya untuk memastikan bahwa pesan yang dikirim ke *URI* hanya dikirim ke aplikasi itu.

Untuk *web apps*

- (a) Pilih “*Web*”.
 - (b) *Check* kotak dengan tulisan “*Allow Implicit Flow*” jika ingin mengaktifkan *OpenID Connect Hybrid* dan *implicit flow*. *Implicit Flow* memungkinkan aplikasi untuk menerima *sign-in info* dan juga *access token*, sedangkan nilai *default* dari bagian ini adalah *hybrid flow* dimana *flow* ini memungkinkan aplikasi untuk menerima *sign-in info* yaitu *token id*, dan juga *artifacts* atau dalam kasus ini adalah kode otorisasi yang digunakan aplikasi untuk mendapatkan *access token*.
 - (c) Tentukan *redirect URI* yang adalah bagian dari aplikasi yang dihubungi oleh *endpoint* dari *Azure AD 2.0* saat memproses permintaan otentikasi.
 - (d) Dibawah “*Application Secrets*”, pilihlah “*Generate New Password*” dan salinlah *app secret* yang terdapat di *New Password generated dialog box*. Perlu diketahui bahwa salinlah *app secret* sebelum *New password generated dialog* ditutup karena setelah ditutup, *app secret* tidak dapat diambil lagi.
6. Pilihlah “*Save*”.

Tabel berikut menunjukkan *properties* yang perlu dikonfigurasi dan disalin untuk berbagai jenis aplikasi. Nilai *assigned*. berarti harus menggunakan nilai yang diberikan oleh *Azure AD*.

Mendapatkan otorisasi.

Langkah pertama untuk mendapatkan otorisasi adalah dengan cara meminta kepada *Azure AD 2.0 /authorize endpoint*. *Azure ID* akan mengecek pengguna yang masuk dan memastikan persetujuan pengguna untuk izin permintaan aplikasi tersebut. Berikut contoh *request* untuk mendapatkan *authorize code*:

```
https://login.microsoftonline.com/{tenant}/oauth2/v2.0/authorize?
client_id=6731de76-14a6-49ae-97bc-6eba6914391e
&response_type=code
&redirect_uri=http%3A%2F%2Flocalhost%2Fmyapp%2F
&response_mode=query
&scope=offline_access%20user.read%20mail.read
&state=12345
```

Parameter		Deskripsi
<i>tenant</i>	<i>required</i>	Nilai yang dipakai untuk mengontrol siapa yang bisa mengakses masuk ke aplikasi. Nilainya bisa diisi dengan akun <i>Microsoft</i> .
<i>client_id</i>	<i>required</i>	Diisi dengan <i>Application ID</i> yang diberikan oleh portal registrasi (<i>apps.dev.microsoft.com</i>).
<i>response_type</i>	<i>required</i>	Diisi dengan nilai “ <i>code</i> ” untuk menjalankan kode otorisasi.
<i>redirect_uri</i>	<i>recommended</i>	Diisi dengan redirect uri yang nantinya akan menerima respon dari otentikasi yang akan diterima oleh aplikasi. Nilai disini harus sama dengan yang sudah diisi di portal registrasi kecuali <i>URL</i> yang disandikan. Untuk <i>native</i> dan <i>mobile apps</i> , ada nilai default untuk <i>redirect uri</i> yaitu menuju ke <i>https://login.microsoftonline.com/common/oauth2/nativeclient</i> .
<i>scope</i>	<i>required</i>	Diisi dengan hak akses pada <i>Microsoft Graph</i> yang akan diberikan pada pengguna aplikasi.
<i>response_mode</i>	<i>recommended</i>	Nilai yang akan menentukan metode kembalian <i>token</i> yang akan dihasilkan kembali ke aplikasi. Terdapat 2 nilai yaitu “ <i>query</i> ” dan “ <i>form_post</i> ”.
<i>state</i>	<i>recommended</i>	Nilai yang dipasang untuk mencegah serangan <i>cross-site request</i> . Parameter ini juga bisa digunakan untuk menyandikan informasi tentang keadaan pengguna di aplikasi sebelum melakukan permintaan otentikasi.

Setelah melakukan request untuk mendapatkan kode otentikasi, maka akan ditunjukkan ke halaman yang seperti ini.

Bentuk dari respon otentikasi seperti:

```
GET https://localhost/myapp/?
code=M0ab92efe-b6fd-f08-87dc-2c6500a7f84d
&state=12345
```

Parameter	Deskripsi
<i>code</i>	Kode otorisasi yang diminta aplikasi. Dengan kode otorisasi ini, aplikasi bisa meminta <i>access token</i> untuk mendapatkan data. Kode otorisasi memiliki umur yang singkat yaitu akan <i>expired</i> setelah 10 menit.
<i>state</i>	Nilai yang akan bernilai sama persis seperti saat melakukan <i>request</i> jika saat melakukan <i>request</i> memasukkan parameter “ <i>state</i> ” juga.

Setelah mendapat *authorize code*, maka otorisasi dilanjutkan dengan meminta kepada *Azure AD 2.0 /token endpoint* untuk mendapatkan *access token* untuk aplikasi tersebut. Aplikasi yang ingin melakukan *request* untuk mendapatkan *access token* akan memakai kode otorisasi dan juga mengirimkan *request* dengan cara “*post*”. Berikut contoh format untuk melakukan *request* kepada *endpoint /token*.

```
POST /common/oauth2/v2.0/token HTTP/1.1
Host: https://login.microsoftonline.com
Content-Type: application/x-www-form-urlencoded

client_id=6731de76-14a6-49ae-97bc-6eba6914391e
&scope=user.read%20mail.read
&code=OAAABAAAAiL9Kn2Z27UubvWFPbm0gLWQJVzCTE9UkP3pSx1aXxUjq3n8b2
JRLk4OxVXr...
&redirect_uri=http%3A%2F%2Flocalhost%2Fmyapp%2F
&grant_type=authorization_code
&client_secret=JqQX2PNo9bpM0uEihUPzryh // NOTE: Only required for web apps
```

Parameter		Deskripsi
<i>tenant</i>	<i>required</i>	Nilai yang dipakai untuk mengontrol siapa yang bisa mengakses masuk ke aplikasi. Nilainya bisa diisi dengan akun <i>Microsoft</i> .
<i>client_id</i>	<i>required</i>	Diisi dengan <i>Application ID</i> yang diberikan oleh portal registrasi (<i>apps.dev.microsoft.com</i>).
<i>grant_type</i>	<i>required</i>	Diisi dengan nilai " <i>authorization_code</i> " untuk menjalankan kode otorisasi.
<i>scope</i>	<i>required</i>	Diisi dengan hak akses pada <i>Microsoft Graph</i> yang akan diberikan pada pengguna aplikasi.
<i>code</i>	<i>required</i>	Kode otorisasi yang didapat saat melakukan request pertama.
<i>redirect_uri</i>	<i>required</i>	<i>redirect_uri</i> yang sama yang digunakan untuk mendapatkan kode otorisasi.
<i>client_secret</i>	<i>required for web apps</i>	<i>App secret</i> yang didapat saat membuat dan mendaftarkan aplikasi di portal registrasi. <i>Client_secret</i> ini hanya berlaku untuk <i>web apps</i> dan juga <i>web APIs</i> karena di <i>native app</i> tidak memungkinkan untuk menyimpan <i>client_secret</i> di gawai masing-masing.

Contoh respon jika request berhasil diproses adalah seperti berikut:

```
{
  "token_type": "Bearer",
  "scope": "user.read%20Fmail.read",
  "expires_in": 3600,
  "access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6I6k5HVEZ2ZEStZnl0aEV1Q...",
  "refresh_token": "AwABAAAAvPM1KaPlrEqdFSBzjqfTGAMxZGUTdM0t4B4..."
}
```

Setelah mendapatkan *access token*, barulah akses untuk mendapatkan data-data dari *Outlook.com Calendar* bisa diakses dengan menggunakan *Outlook.com Calendar API*. Pentingnya langkah sebelum ini dikarenakan untuk melakukan *request* kepada *Outlook.com Calendar API* diperlukan parameter *access token* saat hendak melakukan *request*. Adapun list dari *request* yang akan dipakai dari *Outlook.com Calendar API* adalah:

Request	endpoint
List events	GET /me/events GET /users/{id userPrincipalName}/events GET /me/calendar/events GET /users/{id userPrincipalName}/calendar/events GET /me/calendars/{id}/events GET /users/{id userPrincipalName}/calendars/{id}/events GET /me/calendargroup/calendars/{id}/events GET /users/{id userPrincipalName}/calendargroup/calendars/{id}/events GET /me/calendargroups/{id}/calendars/{id}/events GET /users/{id userPrincipalName}/calendargroups/{id}/calendars/{id}/events

Request di atas akan meminta event yang ada di akun pengguna. Field {id} di atas diisi oleh id dari pengguna. Sedangkan field {userPrincipalName} bisa diisi dengan nama pengguna.

Listing 2.1: Contoh respon dari request List events diatas

HTTP/1.1 200 OK

Content-type: application/json

Preference-Applied: outlook.timezone="Pacific Standard Time"

Content-length: 1932

```
{
  "@odata.context": "https://graph.microsoft.com/v1.0/$metadata#users('cd209b0b-3f83-4c35-82d2-d88a61820480')/events(subject,body,bodyPreview,value)",
  "value": [
    {
      "@odata.etag": "W/\"ZlnW4RIAV06KYYwlrFNZvQAAKGWwbw==\"",
      "id": "AAMkAGIAAAoZDOFAAA=",
      "subject": "Orientation ",
      "bodyPreview": "Dana, this is the time you selected for our orientation. Please bring the notes I sent you.",
      "body": {
        "contentType": "html",
        "content": "<html><head></head><body><p>Dana, this is the time you selected for our orientation. Please bring the notes I sent you.</p></body></html>"
      },
      "start": {
        "dateTime": "2017-04-21T10:00:00.0000000",
        "timeZone": "Pacific Standard Time"
      },
      "end": {
        "dateTime": "2017-04-21T12:00:00.0000000",
        "timeZone": "Pacific Standard Time"
      },
      "location": {
        "displayName": "Assembly Hall",
        "locationType": "default"
      }
    }
  ]
}
```



```

        "uniqueId": "Assembly Hall",
        "uniqueIdType": "private"
    },
    "locations": [
        {
            "displayName": "Assembly Hall",
            "locationType": "default",
            "uniqueIdType": "unknown"
        }
    ],
    "attendees": [
        {
            "type": "required",
            "status": {
                "response": "none",
                "time": "0001-01-01T00:00:00Z"
            },
            "emailAddress": {
                "name": "Samantha Booth",
                "address": "samanthab@a830edad905084922E17020313.onmicrosoft.com"
            }
        },
        {
            "type": "required",
            "status": {
                "response": "none",
                "time": "0001-01-01T00:00:00Z"
            },
            "emailAddress": {
                "name": "Dana Swope",
                "address": "danas@a830edad905084922E17020313.onmicrosoft.com"
            }
        }
    ],
    "organizer": {
        "emailAddress": {
            "name": "Samantha Booth",
            "address": "samanthab@a830edad905084922E17020313.onmicrosoft.com"
        }
    }
}
]
}

```

2.2 *Slack API*

2.3 *Node.js*

2.4 Cron

2.4.1 Tabel

Berikut adalah contoh pembuatan tabel. Penempatan tabel dan gambar secara umum diatur secara otomatis oleh \LaTeX , perhatikan contoh di file bab2.tex untuk melihat bagaimana cara memaksa tabel ditempatkan sesuai keinginan kita.

Perhatikan bawa berbeda dengan penempatan judul gambar gambar, keterangan tabel harus diletakkan di atas tabel!! Lihat Tabel 2.1 berikut ini:

Tabel 2.1: Tabel contoh

	v_{start}	\mathcal{S}_1	v_{end}
τ_1	1	12	20
τ_2	1		20
τ_3	1	9	20
τ_4	1		20

Tabel 2.2 dan Tabel 2.3 berikut ini adalah tabel dengan sel yang berwarna dan ada dua tabel yang bersebelahan.

Tabel 2.2: Tabel bewarna(1)

	v_{start}	\mathcal{S}_2	\mathcal{S}_1	v_{end}
τ_1	1	5	12	20
τ_2	1	8		20
τ_3	1	2/8/17	9	20
τ_4	1			20

Tabel 2.3: Tabel bewarna(2)

	v_{start}	\mathcal{S}_1	\mathcal{S}_2	v_{end}
τ_1	1	12	5	20
τ_2	1		8	20
τ_3	1	9	2/8/17	20
τ_4	1			20

2.4.2 Kutipan

Berikut contoh kutipan dari berbagai sumber, untuk keterangan lebih lengkap, silahkan membaca file referensi.bib yang disediakan juga di template ini. Contoh kutipan:

- Buku: [?]
- Bab dalam buku: [?]
- Artikel dari Jurnal: [?]
- Artikel dari prosiding seminar/konferensi: [?]
- Skripsi/Thesis/Disertasi: [?] [?] [?]
- Technical/Scientific Report: [?]
- RFC (Request For Comments): [?]
- Technical Documentation/Technical Manual: [?] [?] [?]
- Paten: [?]

- Tidak dipublikasikan: [?] [?]
- Laman web: [?]
- Lain-lain: [?]

2.4.3 Gambar

Pada hampir semua editor, penempatan gambar di dalam dokumen \LaTeX tidak dapat dilakukan melalui proses *drag and drop*. Perhatikan contoh pada file bab2.tex untuk melihat bagaimana cara menempatkan gambar. Beberapa hal yang harus diperhatikan pada saat menempatkan gambar:

- Setiap gambar **harus** diacu di dalam teks (gunakan *field LABEL*)
- *Field CAPTION* digunakan untuk teks pengantar pada gambar. Terdapat dua bagian yaitu yang ada di antara tanda [dan] dan yang ada di antara tanda { dan }. Yang pertama akan muncul di Daftar Gambar, sedangkan yang kedua akan muncul di teks pengantar gambar. Untuk skripsi ini, samakan isi keduanya.
- Jenis file yang dapat digunakan sebagai gambar cukup banyak, tetapi yang paling populer adalah tipe PNG (lihat Gambar 2.1), tipe JPG (Gambar 2.2) dan tipe PDF (Gambar 2.3)
- Besarnya gambar dapat diatur dengan *field SCALE*.
- Penempatan gambar diatur menggunakan *placement specifier* (di antara tanda [dan] setelah deklarasi gambar. Yang umum digunakan adalah **H** untuk menempatkan gambar **sesuai** penempatannya di file .tex atau **h** yang berarti "kira-kira" di sini. Jika tidak menggunakan *placement specifier*, \LaTeX akan menempatkan gambar secara otomatis untuk menghindari bagian kosong pada dokumen anda. Walaupun cara ini sangat mudah, hindarkan terjadinya penempatan dua gambar secara berurutan.
 - Gambar 2.1 ditempatkan di bagian atas halaman, walaupun penempatannya dilakukan setelah penulisan 3 paragraf setelah penjelasan ini.
 - Gambar 2.2 dengan skala 0.5 ditempatkan di antara dua buah paragraf. Perhatikan penulisannya di dalam file bab2.tex!
 - Gambar 2.3 ditempatkan menggunakan *specifier h*.

Curabitur tellus magna, porttitor a, commodo a, commodo in, tortor. Donec interdum. Praesent scelerisque. Maecenas posuere sodales odio. Vivamus metus lacus, varius quis, imperdiet quis, rhoncus a, turpis. Etiam ligula arcu, elementum a, venenatis quis, sollicitudin sed, metus. Donec nunc pede, tincidunt in, venenatis vitae, faucibus vel, nibh. Pellentesque wisi. Nullam malesuada. Morbi ut tellus ut pede tincidunt porta. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam congue neque id dolor.

Donec et nisl at wisi luctus bibendum. Nam interdum tellus ac libero. Sed sem justo, laoreet vitae, fringilla at, adipiscing ut, nibh. Maecenas non sem quis tortor eleifend fermentum. Etiam id tortor ac mauris porta vulputate. Integer porta neque vitae massa. Maecenas tempus libero a libero posuere dictum. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aenean quis mauris sed elit commodo placerat. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Vivamus rhoncus tincidunt libero. Etiam elementum pretium justo. Vivamus est. Morbi a tellus eget pede tristique commodo. Nulla nisl. Vestibulum sed nisl eu sapien cursus rutrum.

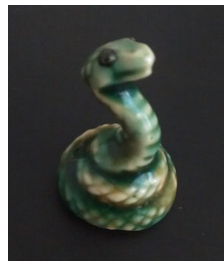
Nulla non mauris vitae wisi posuere convallis. Sed eu nulla nec eros scelerisque pharetra. Nullam varius. Etiam dignissim elementum metus. Vestibulum faucibus, metus sit amet mattis rhoncus, sapien dui laoreet odio, nec ultricies nibh augue a enim. Fusce in ligula. Quisque at magna et nulla commodo consequat. Proin accumsan imperdiet sem. Nunc porta. Donec feugiat mi at justo.



Gambar 2.1: Gambar *Serpentes* dalam format png

Phasellus facilisis ipsum quis ante. In ac elit eget ipsum pharetra faucibus. Maecenas viverra nulla in massa.

Nulla ac nisl. Nullam urna nulla, ullamcorper in, interdum sit amet, gravida ut, risus. Aenean ac enim. In luctus. Phasellus eu quam vitae turpis viverra pellentesque. Duis feugiat felis ut enim. Phasellus pharetra, sem id porttitor sodales, magna nunc aliquet nibh, nec blandit nisl mauris at pede. Suspendisse risus risus, lobortis eget, semper at, imperdiet sit amet, quam. Quisque scelerisque dapibus nibh. Nam enim. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc ut metus. Ut metus justo, auctor at, ultrices eu, sagittis ut, purus. Aliquam aliquam.



Gambar 2.2: Ular kecil

Etiam pede massa, dapibus vitae, rhoncus in, placerat posuere, odio. Vestibulum luctus commodo lacus. Morbi lacus dui, tempor sed, euismod eget, condimentum at, tortor. Phasellus aliquet odio ac lacus tempor faucibus. Praesent sed sem. Praesent iaculis. Cras rhoncus tellus sed justo ullamcorper sagittis. Donec quis orci. Sed ut tortor quis tellus euismod tincidunt. Suspendisse congue nisl eu elit. Aliquam tortor diam, tempus id, tristique eget, sodales vel, nulla. Praesent tellus mi, condimentum sed, viverra at, consectetur quis, lectus. In auctor vehicula orci. Sed pede sapien, euismod in, suscipit in, pharetra placerat, metus. Vivamus commodo dui non odio. Donec et felis.

Etiam suscipit aliquam arcu. Aliquam sit amet est ac purus bibendum congue. Sed in eros. Morbi non orci. Pellentesque mattis lacinia elit. Fusce molestie velit in ligula. Nullam et orci vitae nibh vulputate auctor. Aliquam eget purus. Nulla auctor wisi sed ipsum. Morbi porttitor tellus ac enim. Fusce ornare. Proin ipsum enim, tincidunt in, ornare venenatis, molestie a, augue. Donec vel pede in lacus sagittis porta. Sed hendrerit ipsum quis nisl. Suspendisse quis massa ac nibh pretium cursus. Sed sodales. Nam eu neque quis pede dignissim ornare. Maecenas eu purus ac urna tincidunt congue.



Gambar 2.3: *Serpentes jantan*

LAMPIRAN A

KODE PROGRAM

Listing A.1: MyCode.c

```

1 // This does not make algorithmic sense,
2 // but it shows off significant programming characters.
3
4
5 #include<stdio.h>
6
7 void myFunction( int input, float* output ) {
8     switch ( array[i] ) {
9         case 1: // This is silly code
10             if ( a >= 0 || b <= 3 && c != x )
11                 *output += 0.005 + 20050;
12             char = 'g';
13             b = 2^n + ~right_size - leftSize * MAX_SIZE;
14             c = (--aaa + &daa) / (bbb++ - ccc % 2 );
15             strcpy(a,"hello_$@?");
16         }
17         count = ~mask | 0x00FF00AA;
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

```

Listing A.2: MyCode.java

```

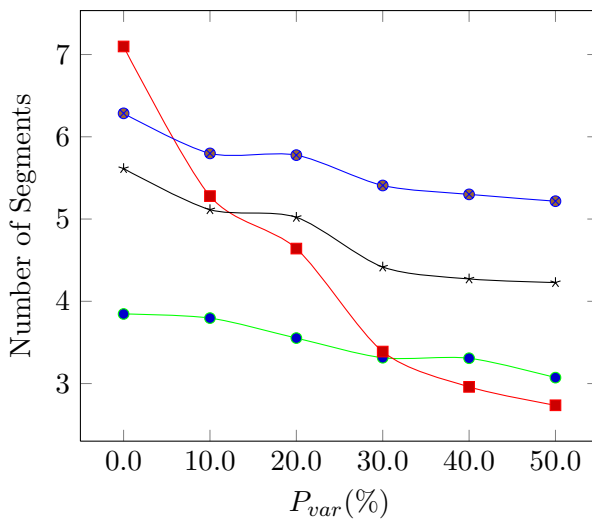
1 import java.util.ArrayList;
2 import java.util.Collections;
3 import java.util.HashSet;
4
5 //class for set of vertices close to furthest edge
6 public class MyFurSet {
7     protected int id; //id of the set
8     protected MyEdge FurthestEdge; //the furthest edge
9     protected HashSet<MyVertex> 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 }

```

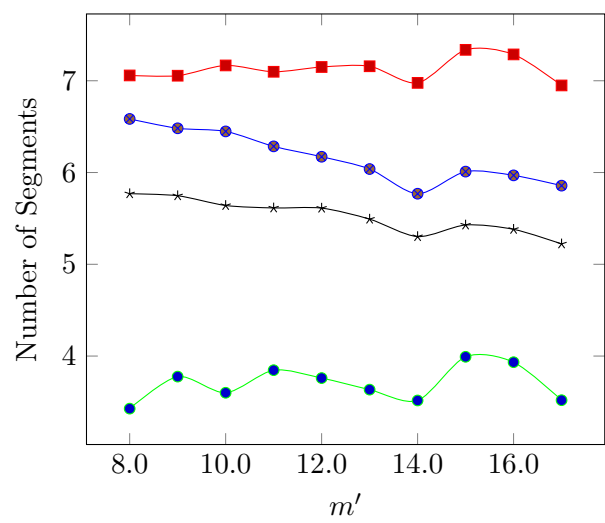

LAMPIRAN B

HASIL EKSPERIMEN

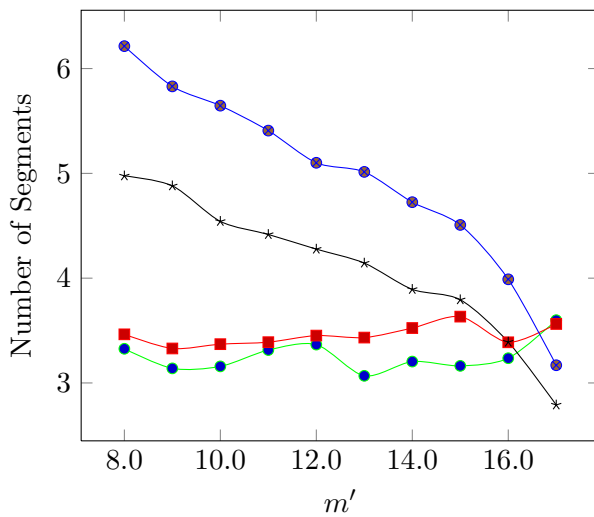
Hasil eksperimen berikut dibuat dengan menggunakan TIKZPICTURE (bukan hasil excel yg diubah ke file bitmap). Sangat berguna jika ingin menampilkan tabel (yang kuantitasnya sangat banyak) yang datanya dihasilkan dari program komputer.



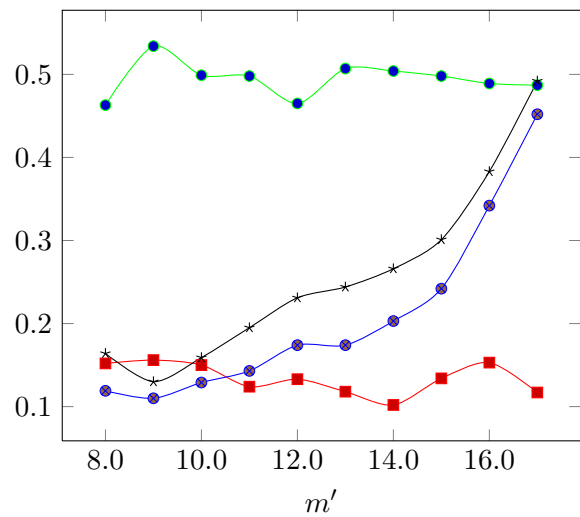
Gambar B.1: Hasil 1



Gambar B.2: Hasil 2



Gambar B.3: Hasil 3



Gambar B.4: Hasil 4