BAB 1

PENDAHULUAN

1.1 Latar Belakang

Zaman modern adalah zaman saat profesi sudah dan sedang berkembang sehingga ada banyak sekali jumlah bidangnya serta perkembangannya. Mayoritas orang menekuni bidang-bidang profesi yang tak terhitung banyaknya untuk mengembangkan setiap bidang profesi. Hal ini menyebabkan kesulitan pengaturan waktu untuk berolahraga, yang adalah salah satu kebutuhan manusia untuk menjaga kesehatan. Salah satu aktivitas olahraga yang paling mudah dan tidak memerlukan gerakan yang sulit adalah berlari, namun kegiatan ini memerlukan lahan yang cukup besar agar dapat dilakukan dengan leluasa. Selain kebutuhan lahan, aktivitas berlari lebih menyenangkan jika dilakukan di luar rumah. Agar dapat dilakukan di dalam rumah, berlari dapat dilakukan di rumah adalah *treadmill*, akan tetapi masalah lingkungan yang monoton dan membosankan di dalam rumah membuat orang enggan untuk melakukan aktivitas berlari. Bila suasana dunia luar dapat dibawa ke dalam rumah, aktivitas ini dapat dilakukan di dalam rumah, tetapi suasana yang dirasakan adalah seperti di luar rumah.

Pada skripsi ini, akan dibuat sebuah perangkat lunak yang dapat menampilkan simulasi aktivitas berlari pada lingkungan yang diinginkan saat berlari di *treadmill*. Dengan menggunakan perangkat lunak tersebut, orang yang berlari dapat menikmati pemandangan yang dipilih saat berlari di dalam rumah sehingga merasa seperti berlari di lingkungan yang dipilih tersebut.

Teknologi yang dapat dimanfaatkan untuk membuat aplikasi VR untuk berlari adalah Google Cardboard dan sensor perangkat bergerak, dan untuk *Application Programming Interface* (API) yang digunakan adalah *Google Streetview API* dan Google *Directions API*.

1.2 Rumusan Masalah

Rumusan masalah yang ada pada skripsi ini adalah:

- Bagaimana memanfaatkan Google VR SDK for Android untuk menampilkan gambar dengan perangkat VR?
- Bagaimana menampilkan hasil dari Google StreetView API dalam bentuk VR?
- Bagaimana mengintegrasikan Google Directions API, gambar Google StreetView dan Google VR dalam perangkat lunak virtual jogging?

1.3 Tujuan

Pada skripsi ini, hal-hal yang coba untuk dicapai adalah :

- Menggunakan Google VR SDK for Android untuk menampilkan gambar dengan *Google Cardboard*.
- Menampilkan hasil gambar dari Google StreetView API pada *Google Cardboard*.

- Mengintegrasikan Google Directions API, gambar dari Google StreetView dan Google VR (Cardboard) dalam perangkat lunak *virtual jogging*.

1.4 Batasan Masalah

Karena ada banyak tempat atau pemandangan di dunia ini, hanya beberapa lokasi saja yang dapat dipilih dari yang telah disediakan.

1.5 Metodologi Penelitian

Metodologi penelitian yang akan digunakan adalah sebagai berikut:

- Melakukan studi literatur dari situs-situs web tentang Google VR SDK, StreetView API, Directions, sensor tentang langkah, baik melalui media tulisan maupun video.
- Menampilkan pemandangan StreetView pada Google Cardboard.
- Mengintegrasikan *Google Directions API* dengan pemandangan *StreetView* yang telah ditampilkan pada Google Cardboard.
- Menganalisis sensor langkah dan menyinkornisasikannya dengan perubahan pemandangan StreetView.

Untuk membuat skripsi ini, peneliti . Setelah mempelajari semua komponen dari aplikasi yang akan dibuat, peneliti akan melakukan implementasi.

1.6 Sistematika Pembahasan

Dokumen dibagi ke dalam beberapa bab dengan sistematika pembahasan sebagai berikut:

1. Bab 1: Pendahuluan, yang menjelaskan gambaran umum penelitian. Mengandung latar belakang, rumusan masalah, tujuan„ batasan masalah, metodologi penelitian, serta sistematika pembahasan.
2. Bab 2: Dasar Teori, berisi landasan dari teori-teori yang berhubungan serta mendukung penelitian. Mengandung Google VR, Google *StreetView API*, Google *Directions API*, dan sensor.
3. Bab 3: Analisis, menjelaskan mengenai proses analisis masalah untuk menemukan solusi. Berisi tentang ...

BAB 2

LANDASAN TEORI

Pada bab ini akan dijelaskan mengenai Google VR, Google StreetView API, Google Directions API, dan *motion sensor*.

2.1 Google VR

Google VR adalah teknologi yang diciptakan perusahaan Google untuk membuat pemandangan dunia maya yang terlihat nyata dengan menempatkan *smartphone* pada alat yang bernama *viewer*. *Viewer* adalah alat untuk melihat dunia VR pada aplikasi VR di *smartphone*. Dua *viewer* yang diciptakan Google adalah *Google Daydream* dan *Cardboard*. Pada penelitian ini, yang digunakan *Google Cardboard*. Google menyediakan sebuah alat bantu bagi pengembang perangkat lunak untuk memudahkan pengembangan aplikasi VR yang disebut Google VR SDK.

2.1.1 Google VR SDK

Google VR SDK adalah alat bantu yang berisi kode program dari aplikasi *virtual reality* (VR) terbuka (*open source*) yang disediakan Google pada repository Github yang tersedia untuk Android (Java), Android NDK, Unity, dan iOS. Secara umum, SDK ini dibuat agar pengembang perangkat lunak dapat mempelajari serta memanfaatkan teknologi VR yang disediakan Google. Ada beberapa aplikasi yang tersedia pada SDK tersebut seperti aplikasi demo bernama hellovr dan pemutar video dalam VR, tetapi penulis akan memanfaatkan bagian aplikasi demo hellovr untuk Android Java pada Google VR SDK. Untuk menggunakan SDK ini, dibutuhkan perangkat lunk Android Studio 2.3.3 dan lebih tinggi, dengan Android SDK versi 7.1.1 (API Level 25) atau lebih tinggi. Bagian dari Google VR SDK yang akan digunakan pada penelitian ini adalah aplikasi hellovr yang akan dijelaskan pada Bagian 2.1.2.

2.1.2 Aplikasi hellovr

Bagian hellovr pada Google VR SDK adalah sebuah aplikasi demo permainan *treasure hunt*, yaitu sejenis permainan mencari bentuk yang mengapung di dunia VR dengan melihat tepat pada bentuk tersebut dan menyalakan pemicu pada Google Cardboard. Setelah kondisi untuk menangkap bentuk yang ada, bentuk tersebut akan menghilang, lalu bentuk yang lain akan muncul di tempat lain.

Komponen Aplikasi hellovr

Dunia VR pada aplikasi ini dibuat dari file *Wavefront Object* (OBJ) dengan tekstur file *Portable Network Graphics* (PNG) yang telah dengan sangat tepat dipetakan pada .obj yang ada sehingga dunia VR terlihat sangat nyata. Bentuk-bentuk yang akan dicari pengguna dibuat dari tiga file OBJ yang merepresentasikan tiga macam bentuk yang ada. Masing-masing file OBJ memiliki dua tekstur yang telah dipetakan pada masing-masing file OBJ dalam file PNG. Satu tekstur (berwarna biru) digunakan ketika pengguna sedang tidak melihat bentuk, sedangkan satu tekstur yang lain (berwarna merah muda) digunakan ketika pengguna sedang menatap bentuk yang ada di dunia VR.

Rancangan kelas Aplikasi hellovr

Aplikasi hellovr memiliki empat kelas pada programnya, di antaranya:

- Texture

  Kelas Texture adalah kelas yang memuat tekstur yang akan digunakan.

- TexturedMesh

  Kelas TexturedMesh adalah sebuah bentuk tiga dimensi yang sudah diberi tekstur sehingga terlihat indah dan berwarna.

- Util

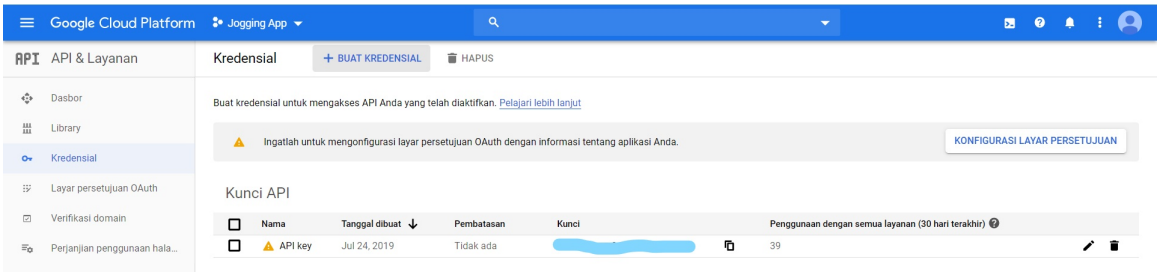
  Kelas Util adalah kelas yang digunakan untuk menghitung vektor dan sudut yang dibentuk antara mata pengguna dan bentuk yang akan dicari, serta mengatur pengaturan yang tepat untuk OpenGL, yang adalah *renderer* yang digunakan untuk menggambar bentuk dan ruangan.

- HelloVrActivity

  Kelas HelloVrActivity adalah kelas yang adalah kelas *activity* Google VR. Berikut adalah diagram kelas untuk memperjelas hubungan antara semua kelas aplikasi hellovr. Kelas ini akan menggunakan tiga kelas lainnya untuk mendapat ruangan dan bentuk yang akan digambar, serta keadaan (*state*) dari permainan, seperti sedang menatap pada bentuk atau tidak dan bagian ruangan yang sedang dilihat.

2.2 Google *StreetView API*

Google StreetView API adalah API yang disediakan Google untuk mendapatkan pemandangan sesuai masukan pengguna melalui *HTTP request*. Ada dua jenis *StreetView API* yang disediakan Google, yaitu *static* dan *dynamic*. *StreetView API* yang statis akan menampilkan pemandangan yang tetap tanpa pergerakan pada pemandangannya, sedangkan yang dinamis menampilkan pemandangan yang berubah-ubah seperti *video*. *StreetView API* yang digunakan pada penelitian ini adalah *Static StreetView API*.



Gambar 2.1: Tampilan *UI Google Cloud* saat mengakses *API Key* (*API Key disamarkan*)

2.2.1 *API Key*

Agar dapat menggunakan *API* ini, ada *API key* yang harus diperoleh pada Google Cloud Platform Console dengan memasukkan nomor kartu kredit. Gambar 2.1 menunjukkan tampilan *Google Cloud* setelah mendapatkan API key. API *key* yang diberikan terdiri atas dua puluh dan delapan belas karakter alfanumerik (bisa huruf kapital dan huruf kecil) yang dihubungkan dengan tanda "-". API *key* yang telah diperoleh akan digunakan sebagai salah satu parameter masukan agar Google API dapat diakses.

2.2.2 Penggunaan *StreetView API*

Secara umum, API diakses menggunakan URL Web sebagai berikut:

https://maps.googleapis.com/maps/api/streetview?parameters

"Parameters" pada URL Web adalah atribut-atribut dengan parameter yang diterima StreetView. Sintaks parameter tersebut adalah:

$$X = Y$$

X adalah atribut dari StreetView, sedangkan Y adalah nilainya, dan nilai tersebut harus sesuai dengan tipe dan rentang nilai masing-masing atribut. Untuk atribut kedua dan seterusnya yang akan dimasukkan dalam parameter (jika ada), dapat diteruskan dengan tanda "&", lalu diikuti dengan pola seperti rumus di atas. Saat mengakses *StreetView API*, ada dua kemungkinan hasil yang diperoleh, yaitu berhasil dan gagal. Pemanggilan *API* yang berhasil akan menghasilkan gambar pemandangan dari lokasi sesuai masukan pengguna, sementara pemanggilan yang gagal menghasilkan sebuah gambar dengan penjelasan bahwa gambar tidak tersedia.

Gambar 2.2: Pemanggilan *StreetView API* yang berhasil

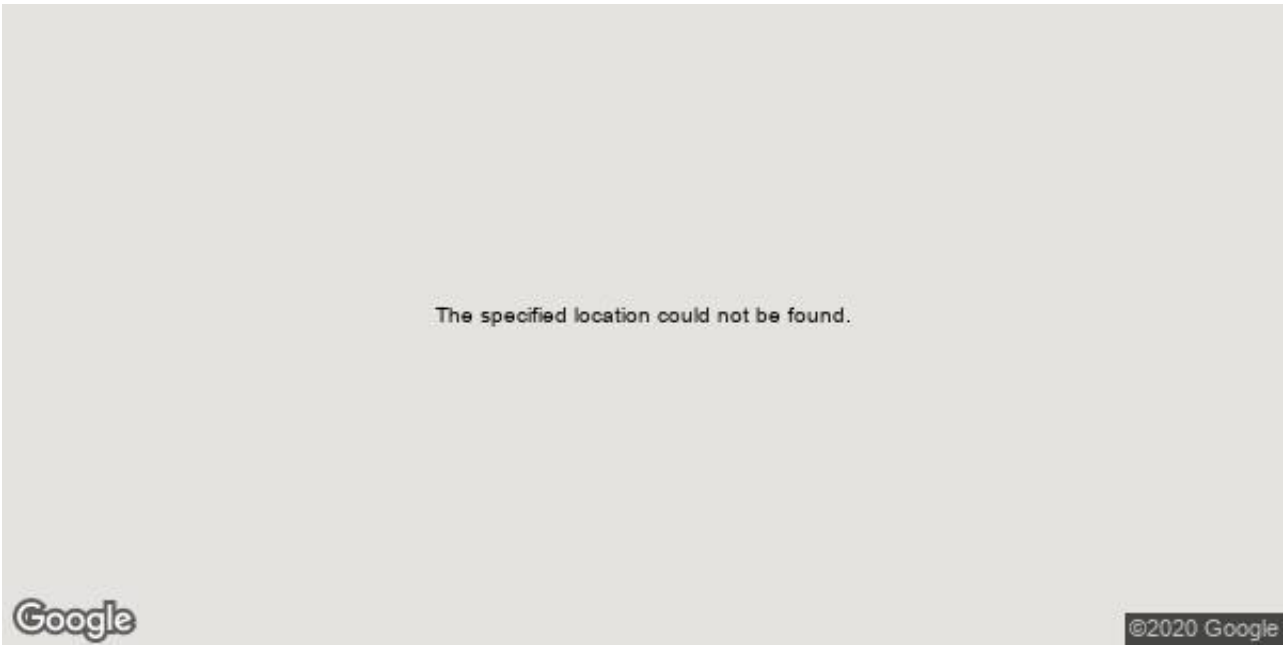
Gambar 2.2 memperlihatkan pemanggilan *StreetView API* yang berhasil, sedangkan Gambar 2.3 memperlihatkan pemanggilan yang gagal.

2.2.3 Atribut Parameter *StreetView API*

Untuk menampilkan pemandangan yang sesuai keinginan pengguna, beberapa parameter masukan harus ditentukan. Ada dua jenis parameter masukan, di antaranya parameter wajib dan parameter opsional. Pengaksesan atau pemanggilan *StreetView API* yang berhasil akan mengembalikan sebuah gambar pemandangan dari lokasi sesuai parameter masukan.

Parameter Wajib

Parameter wajib adalah parameter yang harus dimasukkan oleh pengguna dan jika tidak dimasukkan akan mengakibatkan pemanggilan yang gagal. Beberapa contoh *parameter* wajib *StreetView API* adalah *location* atau pano untuk menentukan lokasi pemandangan yang ingin ditampilkan dan *size* untuk menentukan ukuran gambar pemandangan.



Gambar 2.3: Pemanggilan *StreetView API* yang gagal

Parameter Opsional

Selain parameter wajib, ada parameter opsional, yaitu parameter yang tidak perlu diisi agar pengaksesan *API* berhasil dan biasanya atribut parameter tersebut sudah memiliki nilai bawaan (*default*). Ada beberapa parameter opsional yang dapat digunakan sebagai parameter untuk mengubah pengaturan dari pemandangan yang diambil:

- *signature*
- *heading*
- *fov* (*field of view*)
- *pitch*
- *radius*
- *source*.

Tabel 2.1 menjelaskan semua parameter opsional dari *StreetView API*:

2.3 Google *Directions API*

Google *Directions API* adalah layanan berbasis *HTTP*/*HTTPS* dari Google yang membantu mencari dan menghitung arah dari satu tempat ke tempat yang lain (sumber). Ada beberapa *mode* dari arah yang dapat dicari seperti *driving*, *transit*, *walking*, dan *cycling*. Pengaksesan *Directions API* sangat mirip dengan *StreetView API*, yaitu membutuhkan *API Key*, seperti yang dijelaskan pada Bagian 2.2.1, sebagai salah satu atribut wajib, juga memiliki atribut wajib dan opsional yang dapat diatur lewat parameter.

2.3.1 Penggunaan *Directions API*

Sintaks untuk mengaksesnya pun mirip dengan *StreetView API*, hanya saja ada perbedaan pada bagian "streetview" pada URL Web pada Bagian 2.2.2 diganti dengan "directions".

Tabel 2.1: Atribut-Atribut Opsional *StreetView API*

Nama Atribut	Tipe	Penjelasan	Rentang Nilai *Valid* (yang berdampak)
signature	String (alfabetik)	Atribut untuk memastikan bahwa *request* dikirim dengan *API key* sesuai jenis *signature* yang diatur pemilik *API key.*	-
heading	integer	menyatakan arah pandangan secara horisontal, nilai atribut menyatakan sudut yang dibentuk dari arah utara dengan arah pandang yang diinginkan (sudut yang dibentuk dari arah berlawanan jarum jam)	$0 \leq x \leq 360$
fov (*field of view*)	integer	menyatakan seberapa perbesaran pemandangan (nilai dalam satuan derajat)	$10 \leq x \leq 120$ (yang berdampak)
pitch	integer	menyatakan pandangan pengguna secara vertikal, satuan nilai dalam derajat.	$-90 \leq x \leq 90$
radius	integer	menyatakan jarak dalam meter,	$x > 0$
source	String (alfabetik)		"default" atau "outdoor"

2.4 *Motion Sensor*

Motion sensor adalah sensor pada *smartphone* yang mendeteksi pergerakan gawai *smartphone* (sumber). Pergerakan yang dapat dideteksi termasuk saat gawai dimiringkan, digoyangkan, diayunkan, atau diputar (sumber). Beberapa contoh *motion sensor* pada *smartphone* adalah:

- *accelerometer*

  Sensor yang mendeteksi gerakan *smartphone* terhadap sumbu x, y, dan z, termasuk gaya gravitasi terhadap masing-masing sumbu.

- *gravity sensor* Sensor yang mendeteksi gaya gravitasi terhadap sumbu x, y, dan z
- *gyroscope*
- *linear acceleration sensor*
- *rotation vector sensor*
- *significant motion sensor*
- *step counter*
- *step detector*

LAMPIRAN A

KODE PROGRAM

Listing A.1: Hasil pemanggilan *Directions API* yang gagal

```
{
   "geocoded_waypoints" : [
      {
         "geocoder_status" : "OK",
         "place_id" : "ChIJbYmcEu7maC4RRijB2oKhHLA",
         "types" : [ "establishment", "point_of_interest", "university" ]
      },
      {
         "geocoder_status" : "ZERO_RESULTS"
      }
   ],
   "routes" : [],
   "status" : "NOT_FOUND"
}
```

Listing A.2: Hasil pemanggilan *Directions API* yang berhasil

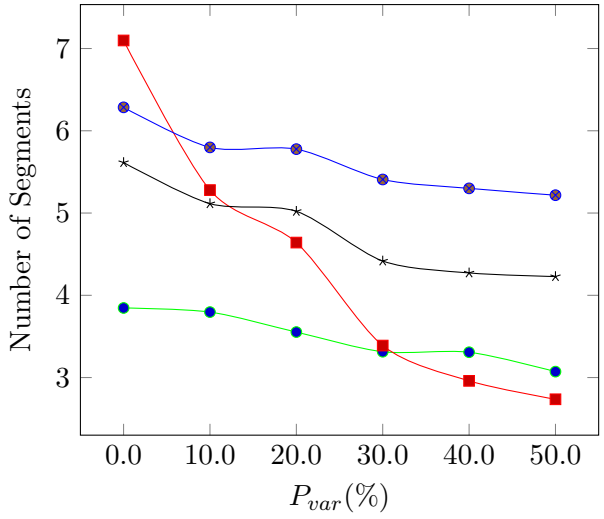
```
{
   "geocoded_waypoints" : [
      {
         "geocoder_status" : "OK",
         "place_id" : "ChIJbYmcEu7maC4RRijB2oKhHLA",
         "types" : [ "establishment", "point_of_interest", "university" ]
      },
      {
         "geocoder_status" : "OK",
         "place_id" : "ChIJU8k7DlHmaC4RQ2mUo1ERm1k",
         "types" : [ "establishment", "hospital", "point_of_interest" ]
      }
   ],
   "routes" : [
      {
         "bounds" : {
            "northeast" : {
               "lat" : -6.8746719,
               "lng" : 107.6137497
            },
            "southwest" : {
               "lat" : -6.893777099999999,
               "lng" : 107.6034922
            }
         },
         "copyrights" : "Map data  2020 ",
         "legs" : [
            {
               "distance" : {
                  "text" : "3.0 km",
                  "value" : 3042
               },
               "duration" : {
                  "text" : "10 mins",
                  "value" : 614
               },
               "end_address" : "Jl. Ir. H. Juanda No.100, Lebakgede, Kecamatan Coblong, Kota Bandung, Jawa Barat 40132, Indonesia"
                  ,
               "end_location" : {
                  "lat" : -6.893777099999999,
                  "lng" : 107.613021
               },
               "start_address" : "Jl. Ciumbuleuit No.94, Hegarmanah, Kec. Cidadap, Kota Bandung, Jawa Barat 40141, Indonesia",
               "start_location" : {
                  "lat" : -6.8746719,
                  "lng" : 107.6046127
               },
               "steps" : [
                  {
                     "distance" : {
                        "text" : "1.0 km",
                        "value" : 1008
                     },
                     "duration" : {
                        "text" : "4 mins",
                        "value" : 251
                     },
```

```
                  "end_location" : {
                     "lat" : -6.8833328,
                     "lng" : 107.6049108
                  },
                  "html_instructions" : "Head␣\u003cb\u003esouth\u003c/b\u003e␣on␣\u003cb\u003eJl.␣Ciumbuleuit\u003c/b\u003e␣
                       toward␣\u003cb\u003eJl.␣Bukit␣Hegar\u003c/b\u003e",
                  "polyline" : {
                     "points" : "tu}h@yowoSdAP|AL‘Cb@dAHHBvC‘@l@JpAXhBVdANRBdAPN@VD@?B?H?F?F?D?BAD?B?DAHCD?f@KlBe@PGB?BAb@IxA]
                          v@QbAWTGPGTIFE^MRIb@Q@An@WDANGBALGVM"
                  },
                  "start_location" : {
                     "lat" : -6.8746719,
                     "lng" : 107.6046127
                  },
                  "travel_mode" : "DRIVING"
               },
               {
                  "distance" : {
                     "text" : "1.1␣km",
                     "value" : 1067
                  },
                  "duration" : {
                     "text" : "3␣mins",
                     "value" : 191
                  },
                  "end_location" : {
                     "lat" : -6.8852083,
                     "lng" : 107.6136492
                  },
                  "html_instructions" : "Turn␣\u003cb\u003eleft\u003c/b\u003e␣onto␣\u003cb\u003eJl.␣Siliwangi\u003c/b\u003e",
                  "maneuver" : "turn-left",
                  "polyline" : {
                     "points" : "xk␣i@uqwoSCEAECKAM?K?E?G@GBIBCBEBGFGHGBADE@?BALI@Al@␣@|@{
                          @HILODEj@w@FMFIBKBO@OGaACm@KsAFc@BMFKLQRULSFODK@IAM?EACACGUQo@Qk@GUEKI[E[Eq@Eq@Ck@?
                          WDa@HUXm@Xa@Ry@Hu@Fi@@QBONkB@KLkAB]?w@"
                  },
                  "start_location" : {
                     "lat" : -6.8833328,
                     "lng" : 107.6049108
                  },
                  "travel_mode" : "DRIVING"
               },
               {
                  "distance" : {
                     "text" : "1.0␣km",
                     "value" : 967
                  },
                  "duration" : {
                     "text" : "3␣mins",
                     "value" : 172
                  },
                  "end_location" : {
                     "lat" : -6.893777099999999,
                     "lng" : 107.613021
                  },
                  "html_instructions" : "Turn␣\u003cb\u003eright\u003c/b\u003e␣after␣McDonald’s␣Simpang␣Dago␣(on␣the␣left)\
                       u003cdiv␣style=\"font-size:0.9em\"\u003ePass␣by␣Bank␣BCA␣DAGO␣(on␣the␣left␣in␣700 m)\u003c/div\
                       u003e\u003cdiv␣style=\"font-size:0.9em\"\u003eDestination␣will␣be␣on␣the␣left\u003c/div\u003e",
                  "maneuver" : "turn-right",
                  "polyline" : {
                     "points" : "pw␣i@ihyoS@SzCLlAHz@Bd@@fB@tBDfABR@L?^BZ@b@Bj@DP@dBHH@bBJB?‘@BbCLh@@J@bDL^BJ?H?X?\\BJ?N?H@F?
                          H@F?p@BB@~@D@?"
                  },
                  "start_location" : {
                     "lat" : -6.8852083,
                     "lng" : 107.6136492
                  },
                  "travel_mode" : "DRIVING"
               }
            ],
            "traffic_speed_entry" : [],
            "via_waypoint" : []
         }
      ],
      "overview_polyline" : {
         "points" : "tu}h@yowoSdAP|AL‘Cb@dAH‘Dd@~Bd@hG|@l@Fr@CNCtCq@|@SpCo@xA␣@nAe@pCiAVMCEEQAg@L[XYJGNKl@␣@|@{@VYp@}@NWF[
              EqAOaCJq@T]‘@i@L[Aa@m@uBMa@Ow@OoCDy@b@cAXa@Ry@P␣BVyCPiB@kAhFV‘BD|EFzADlBHzLp@fIXtBH‘AD"
      },
      "summary" : "Jl.␣Ciumbuleuit,␣Jl.␣Siliwangi␣and␣Jl.␣Dago/Jl.␣Ir.␣H.␣Juanda",
      "warnings" : [],
      "waypoint_order" : []
   }
],
"status" : "OK"
}
```

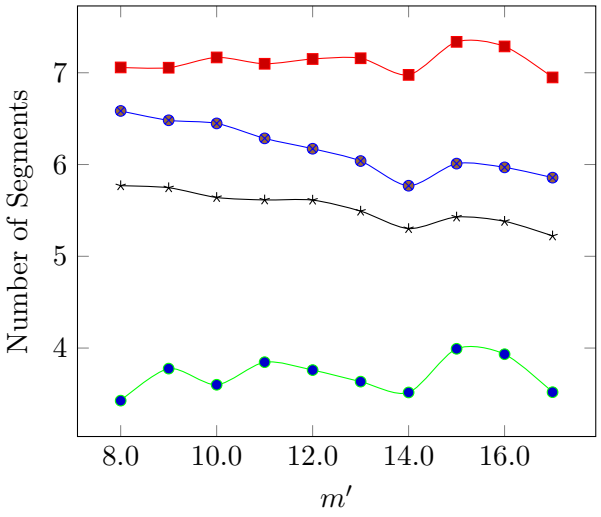
LAMPIRAN B

HASIL EKSPERIMEN

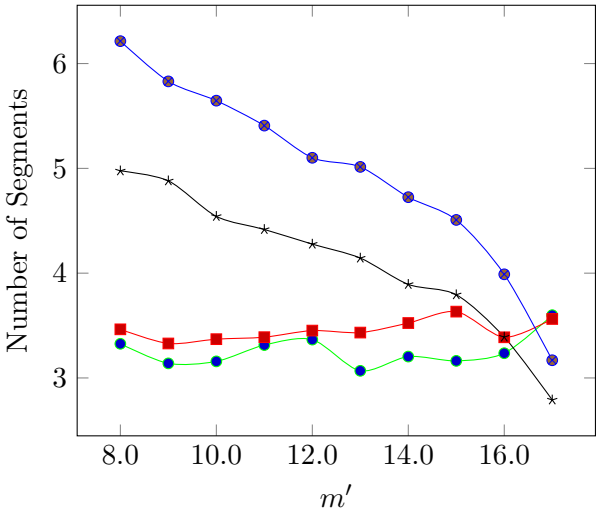
Hasil eksperimen berikut dibuat dengan menggunakan TIKZPICTURE (bukan hasil excel yg diubah ke file bitmap). Sangat berguna jika ingin menampilkan tabel (yang kuantitasnya sangat banyak) yang datanya dihasilkan dari program komputer.



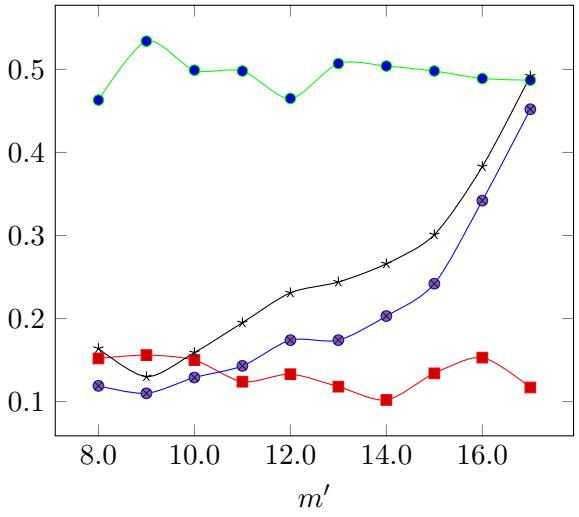
Gambar B.1: Hasil 1



Gambar B.2: Hasil 2



Gambar B.3: Hasil 3



Gambar B.4: Hasil 4