

BAB II

LANDASAN TEORI

2.1. Tinjauan Jurnal

Jarak terpendek merupakan bagian dari teori graf. Jika diberikan sebuah graf berbobot, masalah jarak terpendek adalah bagaimana mencari sebuah jalur pada graf yang meminimalkan jumlah bobot sisi pembentuk jalur tersebut. Masalah dari mencari jarak terpendek antara dua persimpangan dari peta jalan (simpul graf yang berhubungan ke persimpangan dan ujung yang berhubungan ke segmen jalan, yang tiap-tiap nya diberi bobot oleh panjang dari segmen jalan) dapat dimodelkan dari kasus spesial dari masalah jarak terpendek dalam graf.

Menurut Mutakhirah (2012:18), secara umum, pencarian jalur terpendek dibagi menjadi dua metode, yaitu metode konvensional dan metode heuristik. Metode konvensional cenderung mudah dipahami dari pada metode heuristik. Tetapi bila dibandingkan, hasil yang diperoleh dari metode heuristik lebih variatif dan waktu yang diperlukan lebih singkat.

Menurut Edsger Dijkstra (2011:27) algoritma Dijkstra merupakan salah satu varian bentuk algoritma populer dalam pemecahan persoalan terkait masalah optimasi pencarian lintasan terpendek, sebuah lintasan yang mempunyai panjang minimum dari verteks a ke z dalam graph berbobot, bobot tersebut adalah bilangan positif jadi tidak dapat dilalui oleh node negatif. Namun jika terjadi demikian, maka penyelesaian yang diberikan adalah infinity (Tak Hingga). Pada algoritma Dijkstra, node digunakan karena algoritma Dijkstra menggunakan graph berarah untuk penentuan rute lintasan terpendek.

Jadi dari algoritma di atas dapat disimpulkan bahwa algoritma dijkstra dapatn memecahkan masalah persoalan terkait masalah optimasi pencarian lintasan terpendek, dan algoritma dijkstra diguakan karena dijkstra menggunakan graph berarah untuk penentuan rute lintasan terpendek.

2.2. Konsep Dasar Program

Program adalah ekspresi pernyataan atau kombinasi yang disusun dan dirangkai menjadi satu kesatuan prosedur yang berupa urutan langkah untuk menyelesaikan masalah yang diimplementasikan dengan menggunakan bahasa pemrograman, sehingga dapat dieksekusi oleh program.

Menurut Yulikuspartono (2014:29), program merupakan sederetan instruksi atau *statement* dalam bahasa yang dimengerti oleh komputer yang bersangkutan, serta kata pemrograman menurut Sugiyono (2012:21) adalah suatu rangkaian instruksi-instruksi dalam bahasa komputer yang disusun secara logis dan sistematis.

Menurut setiawan (2015:08), program merupakan rancangan mengenai asas serta usaha (dalam ketatanegaraan, perekonomian, dsb) yang akan dijalankan. Sedangkan program computer yaitu urutan perintah yang diberikan pada computer untuk membuat fungsi atau tugas tertentu.

Pemrograman merupakan proses menimplementasikan urutan langkah untuk menyelesaikan suatu masalah dengan urutan langkah-langkah untuk menyelesaikan suatu masalah dengan menggunakan bahasa pemrograman. Orang yang membuat suatu program disebut dengan *programmer*.

Dalam bahasa pemrograman dikenal ada 3 level bahasa pemrograman.

1. *Low level language*

Bahasa yang berhubungan langsung dengan komponen suatu alat.

2. *Midle level language*

Bahasa pemrograman yang menggunakan aturan-aturan gramatika dalam penulisan pernyataan sehingga mudah di pahami dan memiliki intruksi-intruksi tertentu yang dapat di eksekusi langsung oleh computer.

3. *High level language*

Bahasa pemrograman yang langsung dapat dipahami oleh manusia.

2.3. Metode Algoritma

Kata algoritma berasal dari latinisasi nama seorang ahli matematika dari Uzbekistan Al Khawārizmi (hidup sekitar abad ke-9), sebagaimana tercantum pada terjemahan karyanya dalam bahasa latin dari abad ke-12 “Algorithmi de numero Indorum”. Pada awalnya kata algorisma adalah istilah yang merujuk kepada aturan-aturan aritmetis untuk menyelesaikan persoalan dengan menggunakan bilangan numerik arab (sebenarnya dari India, seperti tertulis pada judul di atas). Pada abad ke-18, istilah ini berkembang menjadi algoritma, yang mencakup semua prosedur atau urutan langkah yang jelas dan diperlukan untuk menyelesaikan suatu permasalahan. Masalah timbul pada saat akan menuangkan bagaimana proses yang harus dilalui dalam suatu/sebuah sistem (program) bagi komputer sehingga pada saat eksekusinya, komputer dapat bekerja seperti yang diharapkan. Programmer komputer akan lebih nyaman menuangkan prosedur komputasinya atau urutan langkah proses dengan terlebih dahulu membuat gambaran (diagram alur) diatas kertas

Sumber : <http://mautahujuga.co.id/2015/03/sejarah-singkat-algoritma.html>

Algoritma adalah langkah-langkah yang disusun secara tertulis dan berurutan untuk menyelesaikan suatu masalah. Sedangkan Algoritma Pemrograman adalah langkah-langkah yang ditulis secara berurutan untuk menyelesaikan masalah pemrograman komputer. Dalam pemrograman yang sederhana, algoritma merupakan langkah pertama yang harus ditulis sebelum menuliskan program. Masalah yang dapat diselesaikan dengan pemrograman komputer adalah masalah-masalah yang berhubungan dengan perhitungan matematik.

Menurut Thomas H. Cormen (2011:5), Algoritma adalah prosedur komputasi yang mengambil beberapa nilai atau kumpulan nilai sebagai input kemudian

di proses sebagai output sehingga algoritma merupakan urutan langkah komputasi yang mengubah input menjadi output.

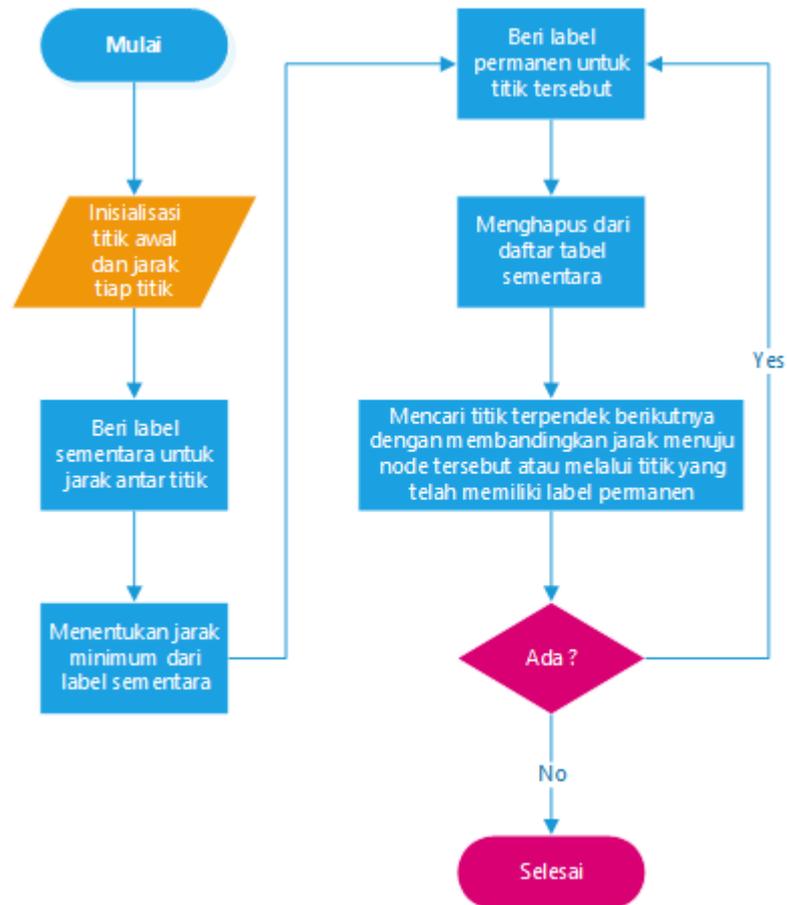
Menurut Rinaldi Munir (2011:10), Para ahli bahasa menemukan kata algorism berasal dari nama cendikiawan muslim yang terkenal yaitu Abu Ja'far Muhammad Ibnu Musa Al-Khuwarijmi (Al-Khuwarijmi dibaca oleh orang Barat menjadi algorism) dalam bukunya yang berjudul Kitab Aljabar Wal-muqabala, yang artinya “Buku Pemugaran dan Pengurangan” (The book of restoration and reduction). Dari judul buku itu kita memperoleh kata “aljabar” (algebra). Perubahan dari kata algorism menjadi algorithm muncul karena kata algorism sering dikelirukan dengan arithmetic sehingga akhiran *-sm* berubah menjadi *-thm*.

2.3.1 Algoritma Dijkstra

Algoritma Dijkstra, dinamai menurut penemunya, Edsger Dijkstra, merupakan salah satu varian dari algoritma *greedy*, yaitu salah satu bentuk algoritma populer dalam pemecahan persoalan yang terkait dengan masalah optimasi. Sifatnya sederhana dan lempang (*straight-forward*). Sesuai dengan artinya yang secara harfiah berarti tamak atau rakus (namun tidak dalam konteks negatif),

Saat ini sudah banyak algoritma yang bisa digunakan untuk menemukan pencarian rute terpendek, dan tidak bisa di pungkiri Dijkstra masih menjadi salah satu yang populer dari sekian banyak algoritma tersebut. Dijkstra merupakan salah satu varian bentuk algoritma populer dalam pemecahan persoalan terkait masalah optimasi pencarian lintasan terpendek sebuah lintasan yang mempunyai panjang minimum dari verteks *a* ke *z* dalam graph berbobot, bobot tersebut adalah

bilangan positif jadi tidak dapat dilalui oleh node negatif. Namun jika terjadi demikian, maka penyelesaian yang diberikan adalah infiniti (Tak Hingga). Pada algoritma Dijkstra, node digunakan karena algoritma Dijkstra menggunakan graph berarah untuk penentuan rute linstasan terpendek. Berikut Pseudo Code dan Flowchart Algoritma Dijkstra:



Gambar 2.1 Dijkstra Flowchart

Passcode Algoritma Dijkstra

```

function Dijkstra(Graph, source):
  for each vertex v in Graph:
    dist[v] := infinity ;
    previous[v] := undefined ;
  end for

  dist[source] := 0 ;
  Q := the set of all nodes in Graph ;

  while Q is not empty:
    u := vertex in Q with smallest distance in dist[] ;
    remove u from Q ;
    if dist[u] = infinity:
      break ;
    end if

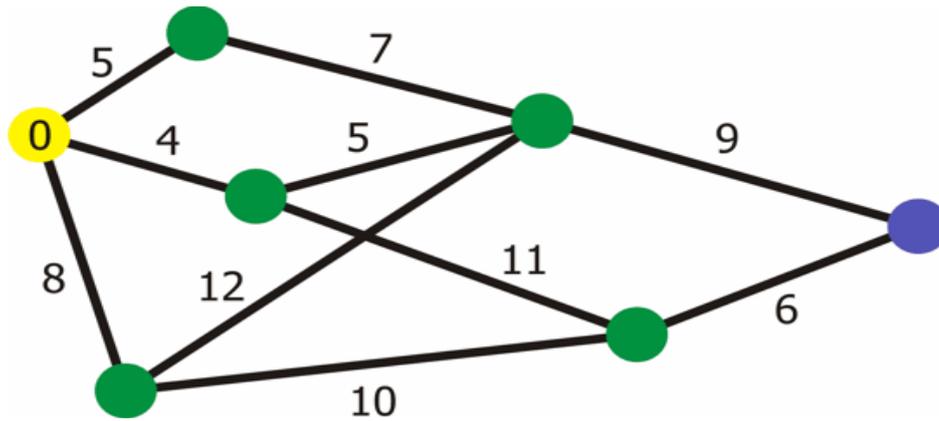
    for each neighbor v of u:
      alt := dist[u] + dist_between(u, v) ;
      if alt < dist[v]:
        dist[v] := alt ;
        previous[v] := u ;
        decrease-key v in Q;
      end if
    end for
  end while

  return dist;

```

1. Implementasi Dijkstra

Algoritma ini bertujuan untuk menemukan jalur terpendek berdasarkan bobot terkecil dari satu titik ke titik lainnya. Misalnya titik menggambarkan gedung dan garis menggambarkan jalan, maka algoritma Dijkstra melakukan kalkulasi terhadap semua kemungkinan bobot terkecil dari setiap titik.



Gambar 2.2 Implementasi Dijkstra

Pertama-tama tentukan titik mana yang akan menjadikan node awal, lalu beri bobot jarak pada node pertama ke node terdekat satu persatu, Dijkstra akan melakukan pengembangan pencarian dari satu titik ke titik lain dan ke titik selanjutnya tahap demi tahap inilah urutan logika dari algoritma Dijkstra :

1. Beri nilai bobot (jarak) untuk setiap titik ke titik lainnya, lalu set nilai 0 pada node awal dan nilai tak hingga terhadap node lain (belum terisi)
2. Set semua node “Belum Terjamah” dan set node awal sebagai “Node keberangkatan”
3. Dari no keberangkatan, pertimbangkan node tetangga yang belum terjamah dan hitung jaraknya dari titik keberangkatan. Sebagai contoh, jika titik keberangkatan A ke B memiliki bobot jarak 6 dan dari B ke node C berjarak 2, maka jarak ke C melewati B menjadi $6+2=8$. Jika jarak ini lebih kecil dari jarak sebelumnya (yang telah terekam sebelumnya) hapus data lama, simpan ulang data jarak dengan jarak yang baru.
4. Saat kita selesai mempertimbangkan setiap jarak terhadap node tetangga, tandai node yang telah terjamah sebagai “Node terjamah”. Node terjamah

tidak akan pernah di cek kembali, jarak yang disimpan adalah jarak terakhir dan yang paling minimal bobotnya.

5. Set “Node belum terjamah” dengan jarak terkecil (dari node keberangkatan) sebagai “Node Keberangkatan” selanjutnya dan lanjutkan dengan kembali ke step 3

2. Langkah-langkah menentukan lintasan terpendek pada algoritma

Dijkstra yaitu:

1. Pada awalnya pilih titik dengan bobot yang terendah dari titik yang belum terpilih, diinisialisasikan dengan „0” dan yang sudah terpilih diinisialisasikan dengan „1”.
2. Bentuk tabel yang terdiri dari titik, status, bobot dan predecessor. Lengkapi kolom bobot yang diperoleh dari jarak titik sumber ke semua titik yang langsung terhubung dengan titik sumber tersebut.
3. Jika titik sumber ditemukan maka tetapkan sebagai titik terpilih.
4. Tetapkan titik terpilih dengan label permanen dan perbarui titik yang langsung terhubung.
5. Tentukan titik sementara yang terhubung pada titik yang sudah terpilih sebelumnya dan merupakan bobot terkecil dilihat dari table dan tentukan sebagai titik terpilih berikutnya.
6. Apakah titik yang terpilih merupakan titik tujuan? Jika ya, maka kumpulan titik terpilih atau predecessor merupakan rangkaian yang menunjukkan lintasan terpendek.

Sumber : <http://kreasisaya07.co.id/2013/05/shortest-path-lintasan-terpendek.html>

3. Tujuan Algoritma Dijkstra

- a. Tujuan Algoritma Dijkstra yaitu untuk menemukan jalur terpendek berdasarkan bobot terkecil dari satu titik ke titik lainnya.
- b. Kelemahan algoritma ini adalah semakin banyak titik akan semakin memakan waktu proses
- c. Jumlah titik menentukan tingkat efektifitas dari algoritma Dijkstra

4. Urutan Logika Algoritma Dijkstra

- a. Beri nilai bobot (jarak) untuk setiap titik ke titik lainnya, lalu set nilai 0 pada node awal dan nilai tak hingga terhadap node lain (yang belum terisi).
- b. Set semua node "Belum terjamah" dan set node awal sebagai "Node keberangkatan".
- c. Dari node keberangkatan, pertimbangkan node tetangga yang belum terjamah dan hitung jaraknya dari titik keberangkatan.
- d. Setelah selesai mempertimbangkan setiap jarak terhadap node tetangga, tandai node yang telah terjamah sebagai "Node terjamah". Node terjamah tidak akan pernah di cek kembali, jarak yang disimpan adalah jarak terakhir dan yang paling minimal bobotnya.
- e. Set "Node belum terjamah" dengan jarak terkecil (dari node keberangkatan) sebagai "Node Keberangkatan" selanjutnya dan lanjutkan dengan kembali ke step 3.

Contoh :

Titik menggambarkan lokasi dan garis menggambarkan jalan, maka algoritma Dijkstra melakukan kalkulasi terhadap semua kemungkinan bobot terkecil dari setiap titik.

Pertama tentukan titik yang akan menjadi nomor kode (node) awal, lalu beri bobot jarak pada node pertama ke node terdekat satu per satu, lalu akan dilakukan pengembangan pencarian dari satu titik ke titik selanjutnya (tahap demi tahap).

1. Node awal 1, Node tujuan 5. Setiap edge yang terhubung antar node telah diberi nilai.
2. Dijkstra melakukan kalkulasi terhadap node tetangga yang terhubung langsung dengan node keberangkatan (node 1), dan hasil yang didapat adalah node 2 karena bobot nilai node 2 paling kecil dibandingkan nilai pada node lain, nilai $= 7 (0+7)$.
3. Node 2 diset menjadi node keberangkatan dan ditandai sebagai node yang telah terdatangi. Dijkstra melakukan kalkulasi kembali terhadap node-node tetangga yang terhubung langsung dengan node yang telah terdatangi. Dan kalkulasi dijkstra menunjukkan bahwa node 3 yang menjadi node keberangkatan selanjutnya karena bobotnya yang paling kecil dari hasil kalkulasi terakhir, nilai $9 (0+9)$.
4. Perhitungan berlanjut dengan node 3 ditandai menjadi node yang telah terjamah. Dari semua node tetangga belum terjamah yang terhubung langsung dengan node terjamah, node selanjutnya yang ditandai menjadi node terjamah adalah node 6 karena nilai bobot yang terkecil, nilai $11 (9+2)$.

5. Node 6 menjadi node terjamah, dijkstra melakukan kalkulasi kembali, dan menemukan bahwa node 5 (node tujuan) telah tercapai lewat node 6. Jalur terpendeknya adalah 1-3-6-5, dan nilai bobot yang didapat adalah 20 (11+9). Bila node tujuan telah tercapai maka kalkulasi dijkstra dinyatakan selesai.

2.4. Pengujian *Software/Aplikasi*

Pengujian white box, kadang-kadang disebut pengujian *glass box*, adalah metode *desaintest case* yang menggunakan struktur control desain *procedural* untuk memperoleh *test case*. Dengan menggunakan metode pengujian *white box*, perekayasa system dapat melakukan *test case* yang :

- a. Memberikan jaminan bahwa semua jalur independen pada suatu modal telah digunakan, paling tidak satu kali.
- b. Menggunakan semua keputusan logis pada sisi *true* dan *false*
- c. Mengeksekusi semua loop pada batasan mereka dan baas operasional mereka.
- d. Menggunakan struktus data internal untuk menjamin validitasnya.

Pada titik ini, dapat diajukan pernyataan uang beralsan, yaitu “Mengapa menghabiskan waktu dan energi untuk menguji logika jika kita dapat dengan lebih baik memperluas kerja yang dapat memastikan bahwa persyaratan program telah dipenuhi ?” Bila dinyatakan dengan cara lain, mengapa kita tidak menggunakan semua energi kita untuk melakukan pengujian *black box*? Jawabannya ada pada sifat catat prangkat lunak :

- a. Kesalahan logis dan asumsi yang tidak benar berbanding terbalik dengan probablitas jalur program yang akan dieksekusi. Kesalahan cenderung muncul dalam kerja kita pada saat kita mendesain dan mengimplementasikan fungsi,

- kondisi atau control yang berbeda di luar mainstraim. Pemrosesan setiap hari cenderung dipahamindengan baik sementara pemrosesan “kasus khusus” cenderung brantakan.
- b. Kita sering percaya bahwa jalur logis mungkin tidak akan dieksekusi bila pada kenyataannya akan dieksekusi pada baris *reguler*. Aliran logika dari suatu program kadang-kadang bersifat konterintuitif yang berarti asumsi kita tidak serasi mengenai aliran dan data kontrol dapat menyebabkan kita membuat kesalahan desain yang akan terungkap hanya setelah pengujian jalur mulai.
- c. Kesalahan *tipografis* adalah *random*. Bila sebuah program di terjemahkan kedalam kode sumber Bahasa pemrograman maka dimungkinkan akan terjadi banyak kesalahan pengetikan. Beberapa akan ditemukan dengan mekanisme pengecekan sintaks, tapi yang lainnya akan tetap tidak terdeteksi sampai pengujian mulai. Masing-masing alasan tersebut memberikan suatu argument untuk melakukan pengujian *white box* . pengujian *black box*, tidak peduli seberapa cermat dilakukan, dapan menangkap bentuk kesalahan tersebut.

2.5. Peralatan Pendukung

Peralatan pendukung merupakan alat yang digunakan digunakan untuk membangun aplikasi xplorer Bumiayu menggunakan *windows7* spesifikasi *software aplikasi* pendukung sebagai berikut.

- a. *jdk-8u60-windows-i586*. (Apllication 185,160 KB)

JDK adalah Perangkat lunak yang digunakan untuk melakukan proses kompilasi dari kode java ke bytecode yang dapat dimengerti dan dapat dijalankan oleh JRE (Java Runtime Envirotment). JDK wajib terinstall pada komputer yang

akan melakukan proses pembuatan aplikasi berbasis java, namun tidak wajib terinstall di komputer yang akan menjalankan aplikasi yang dibangun dengan java. Ada dua produk utama dalam *platform* keluarga *Java SE*, *Java Runtime Environment (JRE)* dan *Java Development Kit (JDK)*.

1) *Java Runtime Environment (JRE)*

JRE menyediakan perpustakaan, *Java Virtual Machine*, dan komponen lainnya untuk menjalankan applet dan aplikasi yang ditulis dalam bahasa pemrograman *Java*. Selain itu, dua teknologi penyebaran kunci adalah bagian dari JRE: *Java Plug-in*, yang memungkinkan untuk menjalankan applet di *browser* populer; dan *Java Web Start*, yang menyebarkan aplikasi mandiri melalui jaringan. JRE tidak mengandung alat dan utilitas seperti *compiler* atau *debugger* untuk mengembangkan applet dan aplikasi.

2) *Java Development Kit (JDK)*

JDK merupakan superset dari JRE, dan berisi segala sesuatu yang ada di JRE, ditambah alat-alat seperti *compiler* dan *debugger* yang diperlukan untuk mengembangkan applet dan aplikasi. Diagram konseptual atas menggambarkan semua komponen teknologi dalam *platform Java SE* dan bagaimana mereka cocok bersama.



```
cmd: Select C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\SASAX>java -version
java version "1.8.0_111"
Java(TM) SE Runtime Environment (build 1.8.0_111-b14)
Java HotSpot(TM) Client VM (build 25.111-b14, mixed mode)

C:\Users\SASAX>
```

Gambar 2.3 *Java version*

Sumber: <http://www.oracle.com/technetwork/java/javase/tech/index.html>

b. Android Studio

Android Studio adalah sebuah IDE untuk pengembangan aplikasi diplatform Android. Sama seperti kombinasi antara Eclipse dan Android Developer Tools (ADT). Saat ini usia Android Studio masih tergolong muda, baru versi 0.2.3 (masih *early acces preview*). Berikut ini adalah sifat dari Android Studio :

1. *Multi-platform*
2. Solaris, AIX, HP-UX dan Mac OS X.
3. *Multi-language*

Android Studio dikembangkan dengan bahasa pemrograman Java, akan tetapi Android Studio mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.

4. *Multi-role*

Selain sebagai IDE untuk pengembangan aplikasi, Android Studio digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.



Gambar 2.4 *Android Studio v2.2.2*

Sumber: <https://developer.android.com/studio/intro/index.html>

c. *Genymotion-2.8.1-vbox. (Application 157,663 KB)*

Genymotion tidak dapat dijalankan dari mesin virtual. Memang, dibutuhkan akses langsung ke perangkat keras untuk memberikan akselerasi *OpenGL*. Menjalankan *Genymotion* dari mesin virtual mencegah untuk langsung mengakses komponen *hardware*. Oleh karena itu, *Genymotion* tidak sesuai ketika diinstal dari *VMWare*, *Parallels*, *Hyper-V* atau *VirtualPC*.

Untuk menjalankan perangkat virtual, harus menginstal *Oracle Vm Virtualbox*, *Genymotion* mungkin bekerja dengan versi *Oracle VM Virtualbox*.



Gambar 2.5 Emulator *Genymotion 2.8.1*

Sumber: https://docs.genymotion.com/Content/01_Get_Started/Requirements.htm

d. **VirtualBox**

VirtualBox adalah sebuah software yang berfungsi untuk metode pembelajaran dalam menginstall OS (Operating System) secara virtual. Jadi pada intinya anda jika ingin belajar menginstall PC tidak perlu membongkar dan install ulang, cukup menggunakan **VirtualBox** anda pun bisa menginstall OS ke dalam PC. Hanya saja sifatnya virtual, tetapi tidak akan jauh beda

dengan yang aslinya. Kata virtualisasi juga merujuk ke kamus Oxford (Convert "something" to a computer-generated simulation of reality) yang artinya Mengubah atau mengkonversi ke bentuk simulasi dari bentuk yang real/nyata.



Gambar 2.6 *VirtualBox*

Sumber : <http://www.anaktkj.net/2016/07/pengertian-manfaat-dan-fungsi.html>

Fungsi VirtualBox

1. Dapat mencoba operating sistem masih dalam tahap uji atau rilis
2. Dapat mencoba dan merasakan operating sistem yang berbeda dengan operating sistem yang utama
3. Dapat mencoba untuk membuat sebuah simulasi jaringan dan
4. Dapat juga untuk menguji dan mencoba simulasi sistem operating system security

Manfaat virtualBok

1. Menghemat uang dalam hal ini dapat di artikan anda tidak perlu membeli hardware atau computer baru untuk mencoba install ulang dan memakai banyak operating system.

2. Dapat menginstall operating beberapa operating sistem secara bersamaan tanpa harus memperainkan harddisk.
3. Dapat juga bermanfaat bagi orang yang awam untuk system pembelajaran tanpa harus memindah dan mengcopy data yang ada pada harddisk.

e. *Adobe Photoshop*

Adobe Photoshop adalah program aplikasi desain yang berguna untuk mendesain gambar, mengedit *image* grafis, dan mengolah foto digital. Versi terbaru program ini adalah *Adobe Photoshop CS6* (Creative Suite 6) Beta, yang dirilis pada tanggal 21 Maret 2012. Software *Adobe Photoshop* dibuat dan dikembangkan oleh *Adobe Systems Incorporated*, sebuah vendor / perusahaan terkemuka di bidang pengembangan perangkat lunak grafis dan multimedia.



Gambar II 2.7 Photoshop

Sumber : <http://info-program-komputer.co.id>

2.6. GPS atau *Google Maps*

GPS atau *Google maps*, merupakan sebuah alat atau sistem yang dapat digunakan untuk menginformasikan penggunaanya berada (secara global) di permukaan bumi yang berbasiskan satelit. Data dikirim dari satelit berupa sinyal radio dengan data digital. Dimanapun posisi saat ini, maka GPS bisa membantu

menunjukkan arah, selama masih terlihat langit. Layanan GPS ini tersedia gratis, bahkan tidak perlu mengeluarkan biaya apapun kecuali membeli GPS receivernya. Awalnya GPS hanya digunakan hanya untuk kepentingan militer, tapi pada tahun 1980-an dapat digunakan untuk kepentingan sipil. GPS dapat digunakan dimanapun juga dalam 24 jam. Posisi unit GPS akan ditentukan berdasarkan titik-titik koordinat derajat lintang dan bujur.

GPS atau google maps, menurut (winardi, 2013:10) adalah sistem untuk menentukan letak di permukaan bumi dengan bantuan penyelarasan (*synchronizatio*) sinyal satelit. Sistem ini menggunakan 24 satelit yang mengirimkan sinyal gelombang mikro ke bumi. Sinyal diterima oleh alat penerima dipermukaan, dan digunakan untuk menentukan letak, kecepatan, arah dan waktu. Sistem ini serupa dengan GPS antara lain GLONASS Rusia, Galileo Eropa, IRNSS India.

GPS atau google maps, menurut (Abidin *et al*, 2011:20) adalah sistem satelit navigasi dan penentu posisi yang memberikan posisi dan kecepatan tiga dimensi dan informasi waktu, secara kontinu di seluruh dunia tanpa tergantung kepada waktu dan cuaca GPS atau *Global Positioning System*, merupakan sebuah alat atau sistem yang dapat digunakan untuk mengkonfirmasi penggunaannya dimana dia berada di permukaan bumi yang berbasis satelit. Data dikirim dari satelit berupa sinyal radio dengan alat digital.

Dimanapun berada maka, GPS bisa membantu menunjukkan arah , selama melihat langit GPS adalah sistem navigasi yang berbasis satelit yang saling berhubungan yang berada di orbitnya. Satelit-satelit itu milik Departement pertahanan Amerika Serikat (*Departement of Defense*) yang pertama kali dikenalkan mulai tahun 1978 dan pada tahun 1994 sudah memakai 24 satelit.

1. Kelebihan dan Kelemahan GPS atau Google Maps

- a. Proses navigasi kendaraan lebih mudah dan cepat
- b. Sangat membantu meningkatkan tracking di dunia militer
- c. Mudah dalam mengidentifikasi setiap lokasi yang ada di permukaan bumi serta mengetahui kondisinya secara *real time*

- d. Lebih ringkas dan mudah digunakan dibandingkan dengan peta konvensional
- e. Penggunaan GPS *tracking* bisa meningkatkan keamanan dan kenyamanan anda dalam menjaga kendaraan atau objek lainnya. Bila hal-hal yang tidak diinginkan terjadi, anda bisa dengan mudah melacak posisi objek yang dipasangkan alat GPS *tracking*.

2. Manfaat GPS atau *Google Maps*

Adanya fasilitas tambahan selain sebagai mesin pencari pada layanan map ini, tentu sangat membantu para pengguna internet. Hal ini khususnya dalam proses pencarian sebuah lokasi yang masih asing. Karena dengan adanya perangkat lunak ini akan memberikan beberapa manfaat di antaranya adalah :

1. Mempercepat pencarian sebuah lokasi dalam waktu yang singkat. Karena dengan teknologi digital sistem pencarian akan berlangsung dengan cepat. Dengan layanan ini, seseorang dapat menemukan tempat yang ia maksudkan. Sangat mudah sekali, bahkan untuk tempat yang terkategori sebagai tempat terpencil pun masih dapat diakses.
2. Membantu seseorang yang sedang bepergian untuk mencari jalan yang cepat pada lokasi yang hendak ditujunya. Hal ini adalah hal lain yang menjadi kelebihan dari adanya layanan ini.

Banyak sekali para orang yang mau bepergian yang tak mengerti dengan jelas arah jalan yang akan mengantarkannya ke tempat tujuan yang dimaksudkan. Banyak dari mereka yang tersesat karena tidak tahu jalan. Bertanya pada setiap orang yang ditemui di jalan pun bukan menjadi solusi yang mudah untuk dilakukan.

Maka dari itu, layanan ini membantu orang dengan tipe seperti ini. Sebelum bepergian, mereka cukup untuk mengakses situs ini, mengetik alamat atau nama tempat yang akan dituju. Dan dalam hitungan detik, layanan ini akan memberi tahu dimana lokasi tempat yang dituju serta penempatannya dari tempat-tempat yang lain yang ada di sekitarnya.

Jadi, layanan ini akan menunjukkan jalan mana saja yang harus ditempuh agar sampai ke tempat tujuan. Sehingga tentunya tak akan tersesat lagi. Dan bertanya pada orang yang ditemui di jalan hanya pada saat yang urgent saja. Tentunya akan lebih teroptimalkan lagi.

3. Mempermudah sistem penyimpanan peta. Karena dengan teknologi digital menjadikan kita tidak perlu ruang yang khusus untuk menyimpan sebuah peta. Sebab, dalam perangkat lunak ini data disimpan dalam bentuk digital.

Bagi orang yang selalu bergelut dengan duani digital, tentu layanan peta digital ini sangatlah membantu. Seseorang tak harus repot membawa sebuah peta yang lengkap dan tentunya tak berukuran kecil untuk menentukan tempat yang dimaksudkan. dengan layanan ini, cukup dengan mengakses situs penyediannya, maka layanan peta digital ini sudah dapat tersedia dan membantu kerja dari pencarinya.

4. Bisa diakses dari mana saja. Dengan teknologi internet, menjadikan kita bisa mengakses peta tersebut dari berbagai tempat yang memiliki fasilitas internet. hal ini adalah kemudahan lain yang ada di dalam layanan ini.

Tak terbatas oleh waktu selama ara penggunaanya terhubung dalam layanan internet, mereka tetap dapat untuk mengakses layanan ini. Juga tak terbatas oleh waktu, kapan pun tetap dapat mengaksesnya.

5. Mengetahui tempat tempat baru yang mungkin belum kita ketahui sebelumnya. Hal ini tentu sangat membantu ketika kita akan menuju ke tempat tersebut. sebelum melakukan perjalanan kita sudah mendapatkan bekal yang cukup tentang arah jalan yang akan ditempuh untuk sampai kepada tempat tersebut.

Terkadang memang sulti bagi seseorang untuk menuju tempat yang belum pernah ia ketahui atau datangi sebelumnya. Mungkin pada jaman terdahulu, seseorang meminta kepada orang yang lebih tahu atau sudah pernah mengunjungi tempat tersebut untuk membuatkan peta atau rute untuk menuju ke tempat tersebut layaknya rute itulah yang ada di dalam layanan ini. Terdapat jalan atau rute yang akan menghantar para pencarinya menuju tempat yang dimaksudkan.

6. Adanya pembaharuan data yang lebih cepat daripada menggunakan peta konvensional. Pembaharuan yang ada di dalam layanan ini tentunya dilakukan dengan sangat cepat, lebih cepat dibandingkan dengan peta konvensional yang ada.

Jika dengan peta konvensional dibutuhkan penelitian langsung ke lapangan tempat terjadinya perubahan nama tempat atau pun kondisi tempat maka dengan adanya layanan digital ini, hal tersebut tak harus dilakukan.