

Hak Cipta Diindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

BAB II

LANDASAN TEORI

2.1 Robot

Kata *robot* berasal dari bahasa *Czech*, *robota* yang berarti ‘bekerja’. Kata *robot* diperkenalkan oleh Karel Capek saat mementaskan *RUR (Rossum’s Universal Robots)* pada tahun 1921. Islam sendiri telah lama menggunakan dan mengembangkan teknologi robot. Pada Zaman Nabi Muhammad SAW, telah menciptakan mesin perang yang telah diberi roda dan dapat melontarkan bom. Lalu Al-Jazari (1136-1206) seorang ilmuwan islam pada dinasti *Artuqid* yang pertama kali menciptakan robot *humanoid* berbentuk band musik dan jam gajah. Kosep robotika Al-Jazari masih gunakan pada masa modern ini (Budiharto, 2014).

2.1.1 Jenis Robot

Robot dibuat dan didisain sesuai kebutuhan pengguna. Menurut (Budiharto, 2014) robot hingga saat ini, secara umum dibagi menjadi beberapa jenis sebagai berikut:

1. Robot manipulator

Robot manipulator banyak digunakan untuk kebutuhan industri, manipulator merupakan sebuah rangkaian mekanik *robot* yang terdiri dari sendi (*joint*) yang terhubung dengan lengan (*link*). Sendi tersebut adalah akuator yang menggerakkan lengan *robot* untuk mengerjakan tugas seperti manusia.



Gambar 2.1 Robot manipulator

Sumber : http://www.kuka-robotics.com/NR/rdonlyres/PR_KR1000.jpg

Hak Cipta Diindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

2. *Robot mobil (mobile robot)*

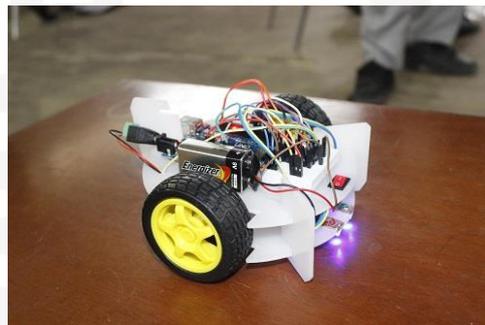
Mobile robot adalah robot yang dapat bergerak dan berjalan bebas dibedakan berdasarkan media jelajah sebagai berikut:

a. *Robot daratan (ground robot)*

Terdapat 2 jenis robot daratan (*ground robot*) yaitu:

i. *Robot beroda*

Robot beroda adalah *robot* yang paling sering dijumpai. Roda adalah teknik paling lama digunakan, mudah, dan efisien untuk menggerakkan *robot*. Roda juga mudah diperoleh, digunakan dan juga mudah untuk memasangnya pada robot. Roda yang digunakan pada robot bermacam-macam ada yang roda dua, roda empat, roda enam, atau roda *caterpillar* dan dipilih sesuai medan yang dilintasi robot.



Gambar 2.2 Robot beroda

ii. *Robot berkaki*

Robot berkaki adalah robot yang meniru perilaku makhluk hidup seperti berjalan, berlari, menaiki tangga. Contoh robot *soccer humanoid* dalam lomba kontes robot indonesia.



Gambar 2.3 Robot berkaki

Hak Cipta Diindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

b. *Robot air (submarine robot)*

ROV (Remotely Operated Vehicles) adalah salah satu jenis *robot submarine*. *Robot* ini kedap air dan biasa digunakan untuk melakukan pencarian didalam air contohnya pencarian pesawat yang jatuh dilautan yang luas. Dengan menggunakan robot ini pencarian lebih mudah karena memiliki sensor, kamera dan lengan (*arm*). Kontrol robot *ROV* berada dipermukaan laut dan data yang didapat akan ditampilkan pada monitor.



Gambar 2.4 Robot ROV

Sumber : <http://oceanservice.noaa.gov/facts/roveastbankgps.jpg>

c. *Robot terbang (aerial robot)*

Drone quadcopter merupakan salah satu jenis robot terbang. Robot ini dapat dikendalikan jarak jauh dan digunakan untuk kebutuhan fotografi pengambilan gambar diketinggian, mempermudah pencarian objek dan untuk pengiriman barang. *Robot drone quadcopter* ini populer karena stabil menggunakan 4 motor *brushless*.



Gambar 2.5 Robot terbang

2.1.2 Sistem Robot dan Fungsinya

Sistem pada robot terdiri dari beberapa bagian dan memiliki fungsi yang berbeda-beda, sistem robot tersebut terdiri dari:

1. Controller

Controller adalah rangkaian elektronik yang terdiri dari sistem minimum *processor*, memori, dan antarmuka *Input* dan *output*. *Controller* digunakan untuk menyelesaikan permasalahan elektronika yang sederhana seperti menghidupkan *LED* sampai permasalahan yang kompleks seperti pengendalian robot (Kadir, 2012). Contoh *controller* dapat dilihat pada gambar 2.6.



Gambar 2.6 Mikrokontroler 328P

2. Akuator

Adalah bagian yang berfungsi untuk penggerak robot dari perintah yang diberikan oleh *Input*. Piranti akuator menggunakan elektronik dan mekanik yang menghasilkan daya gerak. Motor DC adalah jenis dari akuator mekanik dan terdiri dari 2 bagian utama, yaitu rotor dengan stator (Budiharto, 2014). Contoh akuator dapat dilihat pada gambar 2.7.



Gambar 2.7 Motor DC

3. Driver motor

Adalah komponen penguat arus untuk mengendalikan motor listrik seperti motor DC dan *servo*. *Driver* motor dapat juga dihubungkan pada *controller*

Hak Cipta Diindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

untuk pengendalian gerak robot (Kadir, 2012). Contoh *driver* motor dapat dilihat pada gambar 2.8.



Gambar 2.8 Modul driver motor L298

4. Komunikasi

Komunikasi pada robot ada 2 jenis menggunakan kabel atau *wireless*. Contoh perangkat *wireless* seperti RF (radio frekuensi), *bluetooth*, *wifi* dan alat komunikasi lainnya. *Bluetooth* merupakan perangkat yang mudah digunakan serta *support* dengan *smartphone Android* (Budiharto, 2014). Contoh modul *bluetooth* dapat dilihat pada gambar 2.9.



Gambar 2.9 Modul bluetooth hc-05

2.2 Citra Digital

Citra digital adalah suatu matriks yang terdiri dari baris dan kolom, setiap pasang indeks baris dan kolom menyatakan suatu titik pada citra. Nilai dari setiap matriks menyatakan nilai kecerahan titik tersebut. Titik-titik tersebut dinamakan sebagai elemen citra atau piksel. Citra digital dapat didefinisikan sebagai fungsi fungsi dua variabel $f(x,y)$ dan x,y adalah koordinat spasial dan nilai $f(x,y)$ merupakan intensitas citra pada koordinat tersebut. Masing-masing citra digital memiliki ciri-ciri tersendiri misalnya, ukuran citra, resolusi, dan format citra lainnya (Mulyanto, 2009).

2.2.1 Citra Biner (Monokrom)

Citra biner terdiri dari dua buah warna yakni hitam putih. Hitam bernilai 1 dan putih bernilai 0. Untuk menyimpan kedua warna ini dibutuhkan 1 bit di memori. Meskipun pada saat ini komputer dapat mengolah citra hitam putih maupun citra

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

berwarna, tetapi citra biner masih tetap dipertahankan keberadaannya karena memiliki beberapa keuntungan. Adapun keuntungan dari citra biner adalah:

1. Tidak menghabiskan banyak memori karena memori yang dibutuhkan hanya 1 bit.
2. Waktu yang dibutuhkan untuk melakukan pemrosesan lebih singkat daripada citra hitam-putih karena pada citra biner operasi yang dilakukan berupa operasi logika (*and, or, not*) daripada operasi aritmatika bilangan bulat.

Susunan *pixel* citra biner (monokrom) dapat dilihat pada gambar berikut:

				=	0	1	1	1	1
				=	0	1	0	0	0
				=	0	1	1	1	1
				=	0	1	0	0	0
				=	0	1	0	0	0

Gambar 2.10 Susunan *pixel* citra biner

Citra monokrom dapat dilihat pada gambar berikut:



Gambar 2.11 Citra Monokrom

2.2.2 Citra Warna

Setiap *pixel* pada citra yang terdiri dari tiga warna dasar yakni *Red*, *Green*, *Blue* (RGB). Kedalaman *pixel* citra yang digunakan mempengaruhi banyaknya warna yang digunakan. Setiap warna dasar menggunakan penyimpanan sebanyak 8 bit = 1 *byte*, yang mempunyai kombinasi warna yang sangat banyak. Jadi satu *pixel* pada citra warna terdiri dari 3 *byte*. Citra warna dapat dilihat pada gambar berikut.



Gambar 2.12 Citra Warna

2.3 Computer Vision

Adalah salah satu cabang ilmu pengetahuan yang bertujuan untuk membuat suatu keputusan yang berguna mengenai objek fisik nyata dan keadaan berdasarkan sebuah gambar atau citra. *Computer vision* merupakan kombinasi antara (Shapiro & Stockman, 2000):

1. Pengolahan Citra

Pengolahan Citra (*Image Processing*) merupakan bidang yang berhubungan dengan proses perubahan pada citra agar mendapatkan kualitas citra yang lebih baik.

2. Pengenalan Pola

Pengenalan Pola (*Pattern Recognition*) merupakan bidang yang berhubungan dengan proses identifikasi objek pada citra, yang bertujuan untuk mendapatkan informasi yang disampaikan oleh citra.

Hak Cipta Diindungi Undang-Undang

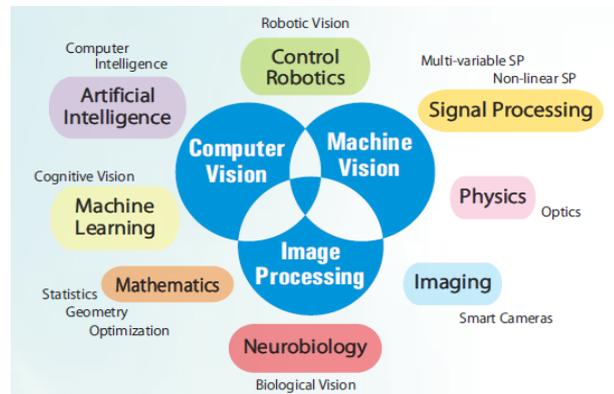
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Gambar ini menampilkan relasi antara *computer vision* dengan bidang riset lainnya:



Gambar 2.13 Relasi *computer vision*

Sumber : (Shapiro & Stockman, 2000)

2.4 Robot Vision

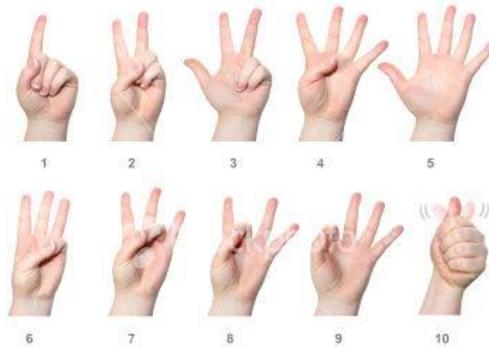
Robot vision merupakan teknologi penerapan *computer vision* pada *robot*. *Robot* membutuhkan informasi *vision* (kamera) untuk memutuskan sebuah aksi. *Vision* pada *robot* menjadi sangat penting karena informasi yang diberikan lebih detail dibanding jika hanya menggunakan sensor jarak, sensor inframerah dan *limit sensor*. Penerapan *vision* pada *robot* antara lain sebagai alat bantu navigasi, mencari objek yang diinginkan, inspeksi lingkungan, dan pengenalan pola visual contohnya pengenalan citra isyarat tangan (Budiharto, 2014).

Pada umumnya sebuah robot berbasis *vision* memiliki karakteristik (Budiharto, 2015):

1. *Sensing*, robot harus dapat mendeteksi lingkungan sekitar seperti halangan, panas, suara, dan citra.
2. Mampu bergerak, robot pada umumnya bergerak menggunakan roda, kaki, terbang atau berenang.
3. Cerdas, robot memiliki kecerdasan buatan untuk memutuskan aksi secara tepat dan akurat.
4. Membutuhkan energi yang memadai, robot membutuhkan *power supply* yang memadai agar unit pengontrolan dan aktuator dapat menjalankan fungsinya dengan baik.

2.5 Isyarat Tangan

Isyarat tangan digunakan untuk berkomunikasi tanpa berbicara hanya mengandalkan gerakan dan bentuk jari tangan (Branson, 1996). Isyarat yang digunakan pada penelitian ini adalah isyarat tangan kiri dan menggunakan pola isyarat bilangan yang biasa digunakan sehari-hari seperti 1, 2, 3, 4, 5, 6 dan 7. contoh isyarat tangan bilangan dapat dilihat dibawah ini.



Gambar 2.14 Isyarat bilangan

Sumber : (Hasan, 2013).

2.6 Mikrokontroller Arduino UNO

Arduino terbagi menjadi dua, yaitu *hardware* dan *software*. *Hardware* merupakan papan *Input* dan *output* (I/O) dan *software* meliputi penulis program berupa *Integrated Development Environment (IDE)*, papan Arduino diprogram dengan cara menyambungkan kabel serial ke PC yang terkoneksi dengan *software* IDE. Menggunakan bahasa program C Arduino yang mengadopsi bahasa C AVR. Arduino merupakan perangkat *opensource* (Kadir, 2012).



Gambar 2.15 Arduino UNO

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

2.6.1 Hardware: Arduino Uno (ATMega328)

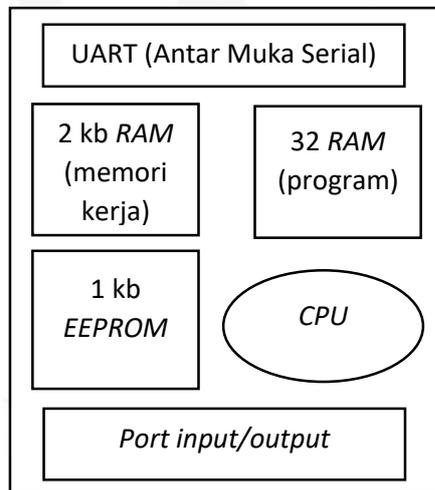
Robot vision yang dijadikan sebagai media penelitian ini menggunakan papan *mikrokontroller* Arduino Uno yang berbasis ATMega328. Berikut spesifikasi dari Arduino Uno (Kadir, 2012):

Tabel 2.1 Spesifikasi Arduino UNO

Mikrokontroller	ATMEGA 328
Input Voltage	7-12V
Digital I/O pins	14 (6 PWM output)
Analog I/O pins	6
Flash Memory	32 KB (Atmega 328) 0.5 KB (boot loader)
EEPROM	1 KB
Clock Speed	16 Mhz

(Kadir, 2012) Menggambarkan diagram blok sederhana *mikrokontroller*

Arduino UNO :



Gambar 2.16 Blok diagram Arduino

1. Memory kerja 2 KB RAM yang digunakan oleh variable-variabel dalam program, bersifat *volatile* (hilang saat daya diputuskan).

Hak Cipta Dilindungi Undang-Undang

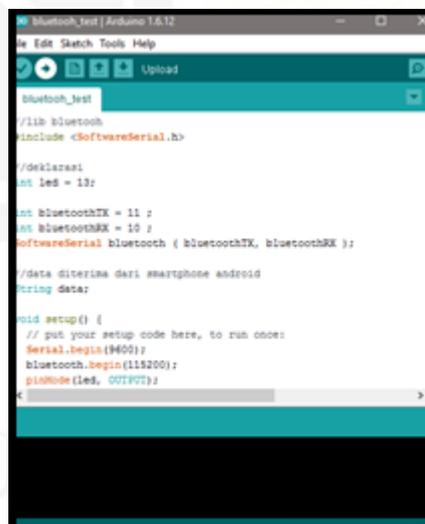
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

2. *Flash memory* 32 KB RAM yang bersifat *non-volatile* untuk menyimpan program yang di-*upload* dari komputer dan juga *bootloader*, yaitu program yang dijalankan CPU saat daya dinyalakan.
3. *EEPROM* 1KB untuk menyimpan data yang tidak boleh hilang saat daya diputuskan. Bersifat *non-volatile*.
4. CPU untuk menjalankan setiap instruksi program.
5. Port *Input* dan *output* untuk menerima dan mengeluarkan data digital atau analog melalui pin-pin yang ada.

2.6.2 Software: IDE Arduino 1.6.0

IDE Arduino terdiri dari *editor* program, *compiler*, dan *uploader*. *Editor* program berfungsi sebagai tempat bagi pengguna untuk menulis dan mengedit program karena *mikrokontroler* hanya memahami kode biner, maka diperlukan *compiler* untuk mengubah kode program menjadi kode biner.

Agar kode biner bisa dimuat ke dalam *mikrokontroler*, maka dibutuhkan sebuah modul yang bernama *boothloader*. Kode program sering juga disebut dengan istilah *sketch* (Kadir, 2012).



Gambar 2.17 IDE Arduino

2.7 Android

Android adalah sebuah sistem operasi pada handphone yang bersifat terbuka dan berbasis pada sistem operasi Linux (Safaat H, 2012). Pada awalnya Android dikembangkan oleh sebuah perusahaan *startup* yang bernama Android, Inc. Pada

Hak Cipta Diindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

tahun 2005 Android diakuisi oleh Google. Dan Google ingin Android untuk terbuka dan gratis (*opensource*). Oleh karena itu, setiap orang dapat menggunakan dan mengembangkan Android dengan mengunduh *source code* Android secara lengkap.

Beberapa versi Android sampai saat ini dapat dilihat pada Tabel 2.2 berikut:

Tabel 2.2 Versi Android

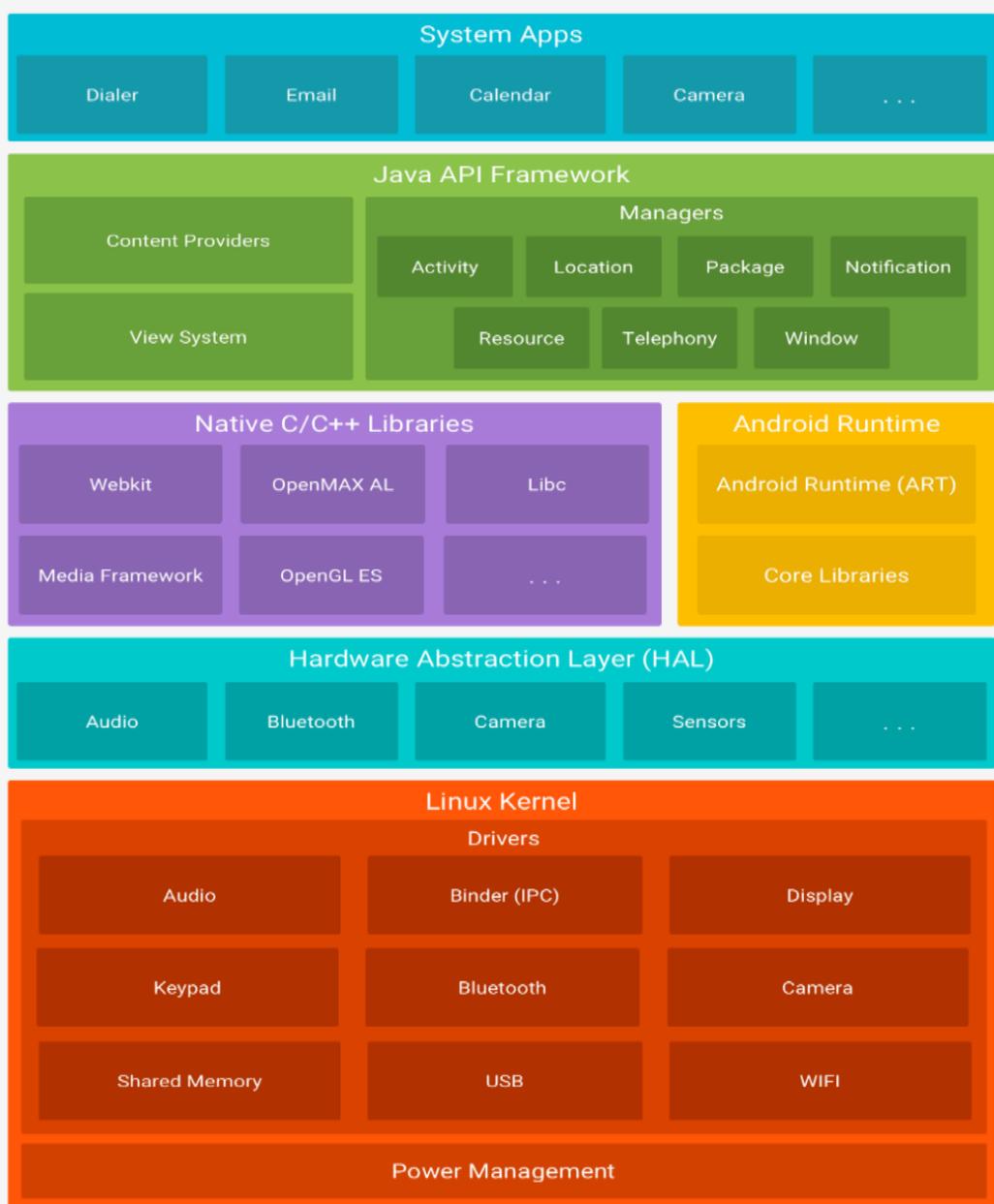
Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	1.7%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	1.6%
4.1.x	Jelly Bean	16	6.0%
4.2.x	Jelly Bean	17	8.3%
4.3	Jelly Bean	18	2.4%
4.4	KitKat	19	29.2%
5.0	Lollipop	21	14.1%
5.1	Lollipop	22	21.4%
6.0	Marshmallow	23	15.2%

Sumber : <https://developer.Android.com/about/dashboards/versi-platform>

Kelebihan dari Android adalah:

1. Bersifat *opensource* dapat digunakan dan dikembangkan secara gratis.
2. Sudah memiliki fitur yang lengkap seperti kamera, penyimpanan SQLite, konektivitas *bluetooth*, *GPS*, *proximity sensor*, *accelerometer* sensor, *digital compass* dan fitru lainnya.
3. Menyediakan dukungan pengembangan aplikasi dengan membuat *software* Android Studio yang digunakan untuk membangun aplikasi Android secara mudah.

Arsitektur Android terdiri dari beberapa layer yang membentuk sistem operasi Android (Safaat H, 2012), dapat dilihat pada gambar:



Gambar 2.18 Arsitektur *Platform Android*

Sumber : <https://developer.Android.com/guide/arsitektur-platform>

Pada gambar diatas menjelaskan bahwa sistem operasi Android secara kasar dibagi menjadi lima bagian dalam empat layer utama (Safaat H, 2012):

- Hak Cipta Diindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
 2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.



Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

1. *Linux Kernel*. *Fondasi platform* Android adalah kernel Linux. Pada layer ini mengandung *driver* untuk komponen *hardware* seperti kamera, *bluetooth*, wifi dan audio dari perangkat *smartphone* Android.
2. *Hardware Abstraction Layer* (HAL) merupakan modul *hardware*. Ketika API berkeja mengakses kamera, maka sistem Android akan memuat modul untuk *hardware* tersebut.
3. Android runtime pada layer yang sama dengan *libraries*, Android runtime menyediakan sekumpulan *library* inti yang memungkinkan para pengembang untuk menulis atau membuat aplikasi Android menggunakan bahasa pemrograman Java. Android *runtime* juga berisi 24 *Dalvik virtual machine*, yang memungkinkan setiap aplikasi Android untuk berjalan dalam prosesnya sendiri. Dalvik adalah mesin Android dan dioptimalkan untuk perangkat *mobile* bertenaga baterai dengan keterbatasan memory dan CPU.
4. *Native C/C++ Libraries* pada layer ini mengandung seluruh *source code* yang merupakan fitur utama sistem OS Android. Contoh *WebKit library* yang menyediakan fungsi untuk *web browsing*. *SQLite library* menyediakan layanan database sehingga aplikasi dapat menyimpan data.
5. *Java API Framework* membuka berbagai kemampuan dari sistem operasi Android untuk para pengembang sehingga mereka dapat menggunakannya dalam aplikasi mereka.
6. *System Apps* Pada layer atas ini, kita dapat menemukan aplikasi umum pada perangkat Android yaitu seperti *Phone*, *Contacts*, *Browser* dan juga aplikasi yang telah diunduh dan dipasang dari *Playstore*.

2.8 OpenCV

Opensouce Computer Vision (OpenCV) pada penelitian ini sebagai pengenalan isyarat tangan menggunakan metode *Support Vector Machine (SVM)*. *Library OpenCV* telah memiliki lebih dari 2500 algoritma yang sudah teroptimasi. Algoritma ini bisa digunakan untuk mendeteksi dan mengenal wajah, mengidentifikasi objek, mengklasifikasi tindakan manusia, melacak pergerakan objek, mengekstraksi model 3D dari objek, memproduksi poin 3D dari *stereo camera*, mencari gambar yang sama dari *database* gambar, mengikuti pergerakan mata, dan pengenalan pola isyarat tangan.

Hak Cipta Diindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

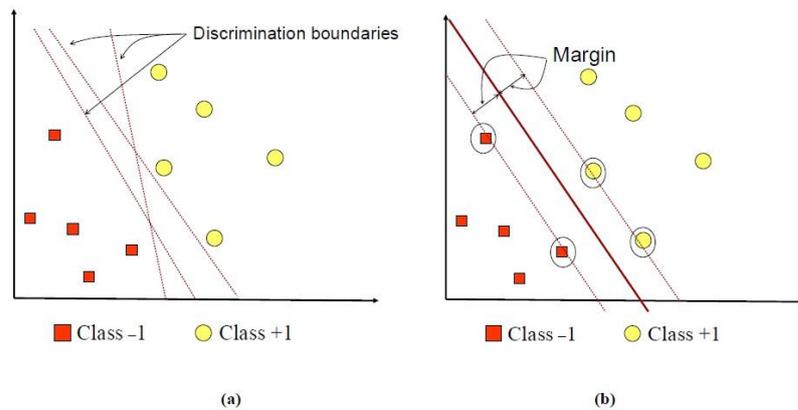
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Kelebihan OpenCV yaitu *opensource* dapat digunakan muti bahasa pemrograman yaitu C++, C, Python dan Java dan mendukung mult *platform* yaitu Windows, Linux, Android, dan Mac OS (Shapiro & Stockman, 2000).

2.9 Support Vector Machine (SVM)

Vapnik pada tahun 1992 pertama kali dikenalkan sebagai gabungan konsep-konsep unggulan dalam metode *pattren recognition*. SVM merupakan metode *pattren recognition* yang masih baru. Namun, kemampuannya dalam berbagai aplikasinya menempatkan pilihan yang pertama dalam *pattren recognition*.

SVM adalah metode *machine learning* yang bekerja atas prinsip *Structural Risk Minimization (SRM)* dengan tujuan menemukan *hyperplane* terbaik yang memisahkan dua buah *class* pada *Input space*. Untuk mengetahui didalam citra tersebut terdapat manusia atau tidak, digunakan *SVM classifier* untuk memisahkan *human* dan *nonhuman*. Pada *SVM classifier* dan algoritma klasifikasi yang berusaha memisahkan sebuah *hyperlane* optimal (Shawe-Taylor & Cristianini, 2000).



Gambar 2.19 SVM mencari *hyperlane* terbaik memisahkan *class -1* dan *+1*

Sumber : (Shawe-Taylor & Cristianini, 2000)

2.9.1 Algoritma Support vector machine

Langkah awal suatu algoritma SVM adalah pendefinisian persamaan suatu *hyperplane* pemisah yang dituliskan dengan.

$$W \cdot X + b = 0$$

Algoritma 2.1 Persamaan *hyperplane*

Hak Cipta Dilindungi Undang-Undang

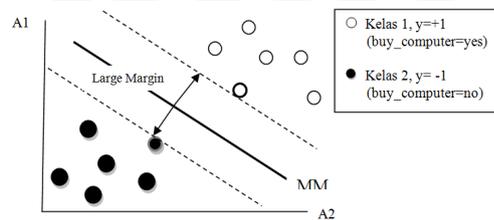
1. Diarangi mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

W adalah suatu bobot vektor, yaitu $W = \{W_1, W_2, \dots, W_n\}$; n adalah jumlah atribut dan b merupakan suatu skalar yang disebut dengan bias (Nuswantoro, 2013). Jika berdasarkan pada atribut A_1, A_2 dengan permisalan tupel pelatihan $X = (x_1, x_2)$, x_1 dan x_2 merupakan nilai dari atribut A_1 dan A_2 , dan jika b dianggap sebagai suatu bobot tambahan w_0 , maka persamaan suatu *hyperplane* pemisah dapat ditulis ulang sebagai berikut.

$$w_0 + w_1x_1 + w_2x_2$$

Algoritma 2.2 Persamaan hyperplane W

Setelah persamaan dapat didefinisikan, nilai x_1 dan x_2 dapat dimasukkan ke dalam persamaan untuk mencari bobot w_1, w_2 , dan w_0 atau b .



Gambar 2.20 Pemisah 2 class data dengan margin maximum

Pada gambar diatas *SVM* menemukan *hyperplane* pemisah maksimum, yaitu *hyperplane* yang mempunyai jarak maksimum antara tupel pelatihan terdekat. *Support vector* ditunjukkan dengan batasan tebal pada titik tupel. Dengan demikian, setiap titik yang terletak di atas *hyperplane* pemisah memenuhi rumus.

$$w_0 + w_1x_1 + w_2x_2 > 0$$

Algoritma 2.3 Hyperplane pemisah maksimum

Sedangkan, titik yang terletak di bawah *hyperplane* pemisah memenuhi rumus.

$$w_0 + w_1x_1 + w_2x_2 < 0$$

Algoritma 2.4 Titik di bawah hyperplane pemisah

Melihat dua kondisi di atas, maka didapatkan dua persamaan *hyperplane* yaitu.

$$H_1: w_0 + w_1x_1 + w_2x_2 \geq 1 \text{ untuk } y_i = +1$$

$$H_2: w_0 + w_1x_1 + w_2x_2 \leq -1 \text{ untuk } y_i = -1$$

Algoritma 2.5 Dua persamaan hyperplane

Hak Cipta Diindungi Undang-Undang

1. Diarangi mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Diarangi mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Perumusan model SVM menggunakan trik matematika yaitu formula *Lagrangian*. Berdasarkan *Lagrangian formulation*, Maksimal Margin *Hyperplane* (MMH) dapat ditulis ulang sebagai suatu batas keputusan (*decision boundary*) yaitu.

$$d(X^T) = \sum_{i=1}^1 y_i \alpha_i X_i X^T + b_0$$

Algoritma 2.6 Lagrangian formulation

y_i adalah label kelas dari *support vector* X_i . X^T merupakan suatu tupel test. α_i dan b_0 adalah parameter numerik yang ditentukan secara otomatis oleh optimalisasi algoritma SVM dan l adalah jumlah *vector support*.

Adanya *hyperplane* yang maksimum mampu memberikan akurasi yang lebih baik pada data yang dapat dipisahkan secara linier, namun hal tersebut tidak berlaku bagi data yang tidak dapat dipisahkan secara *linier*. Model pembelajaran SVM memperkenalkan istilah penalti untuk klasifikasi kesalahan dalam fungsi objektif dengan menggunakan parameter biaya. Dengan adanya parameter biaya terhadap kesalahan, maka fungsi optimasi SVM menjadi.

$$\min \frac{1}{2} |W|^2 + C \sum_{i=1}^{1n} \xi_i$$

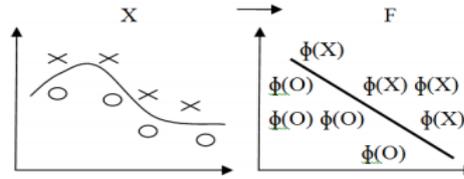
Algoritma 2.7 Fungsi optimasi SVM

$\xi_i \geq 0, 1 \leq m \leq l$ merupakan variabel slack untuk memungkinkan kesalahan beberapa klasifikasi dan C yang disebut sebagai parameter biaya untuk mengontrol keseimbangan antara margin dan kesalahan klasifikasi. Dengan demikian pembatas pada dua kelas diberi suatu tambahan berupa variable slack ξ_i sehingga margin pembatas menjadi:

$$x_i w + b \geq +1 - \xi_1 \text{ untuk } y_i = +1$$

$$x_i w + b \leq -1 + \xi_1 \text{ untuk } y_i = -1$$

Algoritma 2.8 Margin pembatas



Gambar 2.21 Mengubah *problem* yang tidak *linier* menjadi *linier*

2.9.2 *Pattern Recognition Menggunakan SVM*

Menurut (Nugroho, 2003) Konsep *SVM* dijelaskan secara sederhana adalah usaha mencari *hyperline* terbaik yang berfungsi untuk pemisah dari dua buah *class* pada *Input space*. Gambar (a) memperlihatkan beberapa *pattern* yang merupakan anggota dari dua buah *class* : +1 dan -1. *Pattern* yang tergabung pada *class* -1 disimbolkan dengan warna merah (kotak), sedangkan *pattern* pada *class* +1, disimbolkan dengan warna kuning (lingkaran). Klasifikasi adalah usaha menemukan garis (*hyperplane*) yang memisahkan antara kedua kelompok tersebut. Berbagai alternatif garis pemisah (*discrimination boundaries*) ditunjukkan pada gambar (a). Untuk mencari *Hyperplane* pemisah terbaik antara kedua *class* dicari dengan mengukur margin dengan titik maksimalnya. margin adalah jarak antara *hyperplane* dengan *pattern* terdekat dari masing-masing *class*. *Pattern* yang paling dekat ini disebut sebagai *support vector*.

Garis solid pada gambar 1-b menunjukkan *hyperplane* yang terbaik, yaitu yang terletak tepat pada tengah-tengah kedua *class*, sedangkan titik merah dan kuning yang berada dalam lingkaran hitam adalah *support vector*. Usaha untuk mencari lokasi *hyperplane* ini merupakan inti dari proses pembelajaran pada *SVM*.

2.9.3 *Karakteristik Support Vector Machine (SVM)*

Menurut (Nugroho, 2003), karakteristik *support vector machine* adalah sebagai berikut:

1. Prinsipnya *SVM* merupakan *linear classifier*.
2. *Pattern recognition* dilakukan dengan mentransformasikan data pada *Input space* ke ruang yang berdimensi lebih tinggi, dan optimisasi dilakukan pada ruang vector yang baru tersebut. Hal ini membedakan *SVM* dari solusi *pattern recognition* pada umumnya, yang melakukan optimisasi parameter



Hak Cipta Dilindungi Undang-Undang

1. Diarangi mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

pada ruang hasil transformasi yang berdimensi lebih rendah dari pada dimensi *Input space*.

3. Menerapkan strategi *Structural Risk Minimization (SRM)*.
4. Prinsip kerja *SVM* pada dasarnya hanya mampu menangani klasifikasi dua kelas.

2.9.4 Kelebihan *Support Vector Machine (SVM)*

Beberapa kelebihan dari metode *SVM* menurut (Nugroho, 2003) adalah sebagai berikut:

1. *Generalisasi* didefinisikan sebagai kemampuan suatu metode untuk mengklasifikasikan suatu *pattern*, yang tidak termasuk data yang dipakai dalam fase pembelajaran metode itu.
2. *Curse of dimensionality* didefinisikan sebagai masalah yang dihadapi suatu metode *pattern recognition* dalam mengestimasi parameter dikarenakan jumlah sampel data yang *relative* lebih sedikit dibandingkan dengan dimensional ruang vektor data tersebut.
3. *Feasibility SVM* dapat diimplementasikan relative mudah, karena proses penentuan *support vector* dapat dirumuskan dalam *QP problem*.

2.9.5 Kekurangan *Support Vector Machine (SVM)*

Beberapa kekurangan dari metode *SVM* menurut (Nugroho, 2003) adalah sebagai berikut:

1. Sulit dipakai *problem* berskala besar. Dalam hal ini dimaksudkan dengan jumlah sampel yang diolah.
2. *SVM* secara teoritik dikembangkan untuk *problem* klasifikasi dengan dua kelas. Dewasa ini *SVM* telah dimodifikasi agar dapat menyelesaikan masalah dengan lebih dari dua kelas.

2.10 *Unified Modelling Language (UML)*

UML adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berorientasi objek. Pemodelan (*modeling*) digunakan untuk penyederhanaan

permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami (Fowler, 2005).

Berdasarkan pendapat yang dikemukakan di atas dapat ditarik kesimpulan bahwa *Unified Modelling Language (UML)* adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasikan, menspesifikasikan, membangun dan dokumentasi dari sebuah sistem pengembangan perangkat lunak berbasis objek (*Object Oriented Programming*).

2.10.1 Konsep Dasar UML

Untuk dapat memahami *UML* diperlukan pemahaman tentang konsep bahasa pemodelan dan tiga elemen utama *UML*. Konsep utama *UML* yaitu benda (*things*) dan hubungan (*relation*).

1. Benda (*Things*)

Benda atau Objek merupakan bagian paling statik dari sebuah model, yang menjelaskan elemen-elemen lainnya dari sebuah konsep. Bentuk dari beberapa objek:

1. *Classes*, sekelompok dari *object* yang mempunyai attribute, operasi, dan hubungan yang semantik.
2. *Interfaces*, antar-muka yang menghubungkan dan melayani antar kelas dan atau elemen dan mendefinisikan sebuah kelompok dari spesifikasi pengoperasian.
3. *Collaboration*, interaksi dari sebuah kumpulan kelas-kelas atau elemen-elemen yang bekerja secara bersama-sama.
4. *Use cases*, pembentuk tingkah laku objek dalam sebuah model serta direalisasikan oleh sebuah *collaborations*.
5. *Nodes*, bentuk fisik dari elemen-elemen yang ada pada saat dijalankannya sebuah *system*

2. Hubungan (*Relationship*)

Ada 4 macam hubungan dalam penggunaan *UML*, yaitu:

1. *Dependency*, hubungan semantik antara dua objek yang mana sebuah objek berubah mengakibatkan objek satunya akan berubah pula.

Hak Cipta Diindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengummumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

2. *Association*, hubungan antar benda secara struktural yang terhubung diantara objek dalam kesatuan objek.
3. *Generalizations*, hubungan khusus dalam objek anak yang menggantikan objek induk dan memberikan pengaruhnya dalam hal struktur dan tingkah lakunya kepada objek induk
4. *Realizations*, hubungan semantik antar pengelompokkan yang menjamin adanya ikatan diantaranya yang diwujudkan diantara *interface* dan kelas atau *elements*, serta antara *use cases* dan *collaborations*.

2.10.4 Jenis-jenis diagram UML

Fungsi diagram-diagram dalam *UML* adalah untuk menggambarkan permasalahan maupun solusi dari permasalahan suatu model. Diagram tersebut dibagi menjadi 13 diagram, adalah sebagai berikut:

1. *Usecase diagram*

Usecase adalah abstraksi dari interaksi antara *system* dan *actor*. *Usecase* bekerja dengan cara mendeskripsikan tipe interaksi antara *user* sebuah *system* dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah *system* dipakai.

Diagram *Usecase* berguna dalam tiga hal:

- A. Menjelaskan fasilitas yang ada (*requirement*)
- B. Komunikasi dengan klien
- C. Membuat test dari kasus-kasus secara umum

2. *Activity Diagram*

Activity diagram menyediakan analisis dengan kemampuan untuk memodelkan proses dalam suatu sistem informasi. *Activity* diagram dapat digunakan untuk alur kerja model, *usecase* individual, atau logika keputusan yang terkandung dalam metode individual. *Activity* diagram juga menyediakan pendekatan untuk proses pemodelan paralel.

Pada dasarnya, diagram aktifitas canggih dan merupakan diagram aliran data yang terbaru. Secara teknis, diagram aktivitas menggabungkan ide-ide proses pemodelan dengan teknik yang berbeda termasuk model acara, *statecharts*, dan *Petri Nets*.

Hak Cipta Diindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

3. **Package Diagram**

Package diagram utamanya digunakan untuk mengelompokkan elemen diagram *UML* yang berlainan secara bersama-sama ke dalam tingkat pembangunan yang lebih tinggi yaitu berupa sebuah paket. Diagram paket pada dasarnya adalah diagram kelas yang hanya menampilkan paket, disamping kelas, dan hubungan ketergantungan, disamping hubungan khas yang ditampilkan pada diagram kelas.

4. **Statechart Diagram**

Statechart diagram digunakan untuk memodelkan perilaku dinamis satu kelas atau objek. *Statechart* diagram memperlihatkan urutan keadaan sesaat (*state*) yang dilalui sebuah objek, kejadian yang menyebabkan sebuah transisi dari suatu *state* atau aktivitas kepada yang lainnya. *Statechart* diagram khusus digunakan untuk memodelkan tahap-tahap diskrit dari sebuah siklus hidup objek, sedangkan *activity* diagram paling cocok untuk memodelkan urutan aktifitas dalam suatu proses.

5. **Sequence Diagram**

Sequence diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *sequence* diagram adalah gambaran tahap demi tahap yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *usecase* diagram.

6. **Class diagram**

Class diagram, memberikan pandangan secara luas dari suatu sistem dengan menunjukkan kelas-kelasnya dan hubungan mereka. *Class* diagram mempunyai 3 macam *relationships* (hubungan), sebagai berikut:

- A. *Association*, suatu hubungan antara bagian dari dua kelas yang terjadi jika salah satu bagian dari kelas mengetahui kelas yang lain dalam melakukan suatu kegiatan..
- B. *Aggregation*, hubungan *association* adalah salah satu kelasnya merupakan bagian dari suatu kumpulan dan memiliki titik pusat yang mencakup keseluruhan bagian.
- C. *Generalization*, hubungan turunan dengan mengasumsikan satu kelas merupakan suatu kelas super dari kelas yang lain.

Hak Cipta Diindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

7. *Collaboration* Diagram

Collaboration diagram menggambarkan interaksi antar objek seperti *sequence* diagram, tetapi lebih menekankan pada peran masing-masing objek. Setiap *message* memiliki *sequence number*, *message* dari level tertinggi memiliki nomor 1. Diagram membawa informasi yang sama dengan diagram *sequence*, tetapi lebih memusatkan atau memfokuskan pada kegiatan obyek dari waktu pesan itu dikirimkan.

8. *Composite Structure* Diagram

Diagram struktur komposit adalah diagram yang menunjukkan struktur *internal classifier*, termasuk poin interaksinya ke bagian lain dari *system*. Hal ini menunjukkan konfigurasi dan hubungan bagian, yang bersama-sama melakukan perilaku *classifier*. Diagram struktur komposit merupakan jenis diagram struktur yang statis dalam *UML*, yang menggambarkan struktur *internal* kelas dan kolaborasi.

Struktur komposit dapat digunakan untuk menjelaskan:

- A. Struktur dari bagian-bagian yang saling berkaitan;
- B. *Run-time* struktur yang saling berhubungan.

9. *Object* Diagram

Object diagram merupakan sebuah gambaran tentang objek-objek dalam sebuah *system* pada satu titik waktu. Karena lebih menonjolkan perintah-perintah dari pada *class*, *object* diagram lebih sering disebut sebagai sebuah diagram perintah.

10. *Timing* Diagram

Timing Diagram adalah bentuk lain dari *interaction* diagram, fokus utamanya lebih ke waktu. *Timing* diagram sangat berdaya guna dalam menunjukkan faktor pembatas waktu diantara perubahan *state* pada objek yang berbeda.

11. *Component* Diagram

Component diagram ini bila dikombinasikan dengan diagram penyebaran dapat digunakan untuk menggambarkan distribusi fisik dari modul perangkat lunak melalui jaringan. Misalnya, ketika merancang sistem *client-server*, hal ini berguna untuk menunjukkan mana kelas atau paket

Hak Cipta Diindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

kelas akan berada pada *node* klien dan mana yang akan berada di *server*. *Component* diagram juga dapat berguna dalam merancang dan mengembangkan sistem berbasis komponen. Karena berfokus pada analisis sistem berorientasi objek dan desain.

12. Deployment Diagram

Deployment diagram menggambarkan detail bagaimana komponen di *deploy* dalam infrastruktur *system*, komponen akan terletak (pada mesin, *server* atau piranti keras), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *server*, dan hal-hal lain yang bersifat fisik. Hubungan antar *node* (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini.

13. Interaction Overview Diagram

Interaction Overview diagram adalah penggabungan antara *activity* diagram dengan *sequence* diagram. *Interaction Overview* diagram dapat dianggap sebagai *activity* diagram, semua aktivitas diganti dengan sedikit *sequence* diagram, atau bisa juga dianggap sebagai *sequence* diagram yang dirincikan dengan notasi *activity* diagram yang digunakan untuk menunjukkan aliran pengawasan.

2.11 Java

Java merupakan bahasa pemrograman berorientasi objek yang bersifat *open source* dan bisa dijalankan di berbagai *platform*. Java mengadopsi sintaks dari bahasa C dan C++. Setiap program yang ditulis dengan bahasa Java awalnya dikompilasi menjadi sebuah kode objek yang disebut *bytecode*. Hasil dari *bytecode* adalah file berekstensi *.class*, selanjutnya file *.class* tersebut akan diterjemahkan baris demi baris (*interpreter*).

Dalam terminologi Java terdapat istilah “*write once, run anywhere*”. Ini berarti bahwa sekali menulis program Java dan melakukan kompilasi terhadap program Java tersebut, maka *bytecode*nya bisa dijalankan di dalam *platform* manapun selama *platform* tersebut terdapat *Java Virtual Machine (JVM)* (Kadir, 2012).

2.12 Penelitian Terdahulu

Terdapat beberapa penelitian terdahulu yang melakukan penelitian tentang robot *vision* dan pengenalan pola tangan (*hand recognition*) adalah (Kaura, 2013) *Gesture Controlled Robot using Image Processing*, pada penelitian tersebut dirancang robot pelayan (*service robot*) yang dikendalikan menggunakan pengenalan pola isyarat tangan, agar interaksi lebih alami dan mudah. Robot yang dirancang diberi kamera dan modul *wifi* untuk mengirim gambar ke PC setelah diproses perintah akan dikirim kembali ke robot agar robot bergerak sesuai dengan perintah. Hasil dari penelitian tersebut robot dapat mengenali isyarat tangan, namun karena pengiriman tersebut dikirim secara *real-time* pengiriman gambar sering terganggu dan terkadang terjadi *overload*.

Penelitian selanjutnya dilakukan oleh (Nasser, 2011) *Real-time Hand Gesture Detection and Recognition Using Bag-of-Features and Support Vector Machine (SVM) Techniques*. Penelitian tersebut melakukan pengenalan pola isyarat tangan secara *real-time* dengan metode pengenalan *SVM*. Penelitian tersebut membandingkan akurasi dan kecatan dengan *Input* citra dan ukuran yang berbeda. Didapatkan hasil akurasi 96,23% dengan ukuran *Input* data latih 50x50 *pixel* yang berisi data isyarat tangan saja dengan *backgorund* tembok berwarna putih.

Penelitian dilakukan oleh (Sayyid, 2015) *Kontrol Mobil Robot Menggunakan Hand Gesture Recognition Dengan Metode Adaptive Neuro-Fuzzy Interference System (ANFIS)*. Robot dikontrol dengan pengenalan isyarat tangan angka satu untuk maju, dua untuk mundur, tiga untuk belok kiri, empat untuk belok kanan dan lima stop. Hasil penelitian tersebut robot berhasil dikontrol dengan pengenalan isyarat tangan dengan rata-rata *error* 13,3%.

Penelitian survei yang dilakukan oleh (Hasan, 2013) *Vision based hand gesture recognition for human computer interaction: a survey*. Diketahui bahwa tangan adalah media pengenalan yang banyak dipilih pada penelitian sebelumnya, dibandingkan bagian tubuh lainnya dan juga pengenalan tangan statik atau tidak bergerak mayoritas menggunakan *SVM* sebagai metode pengenalan citranya.

Penelitian sebelumnya dalam bidang robotik pada jurusan Teknik Informatika UIN SUSKA Riau dilakukan oleh (Musridho, 2014) dengan judul analisa performa algoritma *line maze solving* pada jalur lengkung dan zig-zag.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.