



BAB II LANDASAN TEORI

2.1 Konsep Dasar Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk satu kesatuan. Komponen-komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap sistem tidak peduli betapapun kecilnya, selalu mengandung komponen-komponen atau subsistem-subsistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai suatu sistem yang lebih besar yang disebut supra sistem. Misalnya suatu perusahaan dapat disebut dengan suatu sistem dan industri disebut dengan supra sistem. Kalau dipandang industri sebagai sistem, maka perusahaan disebut dengan subsistem (Rosa, 2011).

2.1.1 Pengertian Sistem

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau menyelesaikan suatu sasaran tertentu. Prosedur itu sendiri adalah urutan yang tepat dari tahapan-tahapan instruksi yang menerangkan apa (*what*) yang harus dikerjakan, siapa (*who*) yang mengerjakannya, kapan (*when*) dikerjakan, dan bagaimana (*how*) mengerjakannya (Caputo, 2014).

Dalam pendekatan elemen/komponen, sistem didefinisikan sebagai sekumpulan elemen yang saling berinteraksi satu dengan yang lain untuk mencapai suatu tujuan tertentu dengan menerima proses *input* dan menghasilkan *output* dalam proses transformasi yang teratur. Dengan demikian pengertian sistem dapat disimpulkan sebagai suatu prosedur atau elemen yang saling berhubungan satu sama lain dimana dalam sebuah sistem terdapat suatu masukan, proses dan keluaran, untuk mencapai tujuan yang diharapkan (Rosa, 2011).



2.1.2 Karakteristik Sistem

Suatu sistem mempunyai karakteristik agar sistem dapat dibedakan dengan sistem yang lain. Berikut ini macam-macam karakteristik suatu sistem diantaranya (Rosa, 2011) :

1. Komponen sistem (*component*), dimana suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk satu kesatuan. Suatu sistem merupakan salah satu dari komponen sistem lain yang lebih besar, maka disebut subsistem, sedangkan sistem yang lebih besar tersebut adalah lingkungannya.
2. Batas sistem (*boundary*) merupakan pembatas atau pemisah suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.
3. Lingkungan luar sistem (*environment*) merupakan sesuatu di luar batas dari sistem yang dapat mempengaruhi operasi sistem, baik pengaruh yang menguntungkan ataupun yang merugikan.
4. Penghubung sistem (*interface*) merupakan hal yang sangat penting, sebab tanpa adanya penghubung, sistem akan berisi kumpulan subsistem yang berdiri sendiri dan tidak saling berkaitan. Maka dari itu penghubung dapat juga didefinisikan sebagai tempat dimana komponen atau sistem dan lingkungannya bertemu atau berinteraksi.
5. Masukan sistem (*input*) merupakan energi yang dimasukkan ke dalam sistem, dimana masukan tersebut dapat berupa bahan yang dimasukkan agar sistem tersebut dapat beroperasi (*maintenance input*), dan masukan yang diproses untuk mendapat keluaran (*signal input*).
6. Keluaran sistem (*output*) merupakan hasil dari pemrosesan, yang berupa informasi sebagai masukan pada sistem lain atau hanya sebagai sisa pembuangan.
7. Pengolah sistem (*process*) merupakan bagian yang melakukan perubahan dari masukan untuk menjadi keluaran yang diinginkan.
8. Sasaran sistem merupakan suatu tujuan yang ingin dicapai dalam sistem, agar sistem menjadi terarah dan terkendali.

Hak Cipta Dilindungi Undang-Undang

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

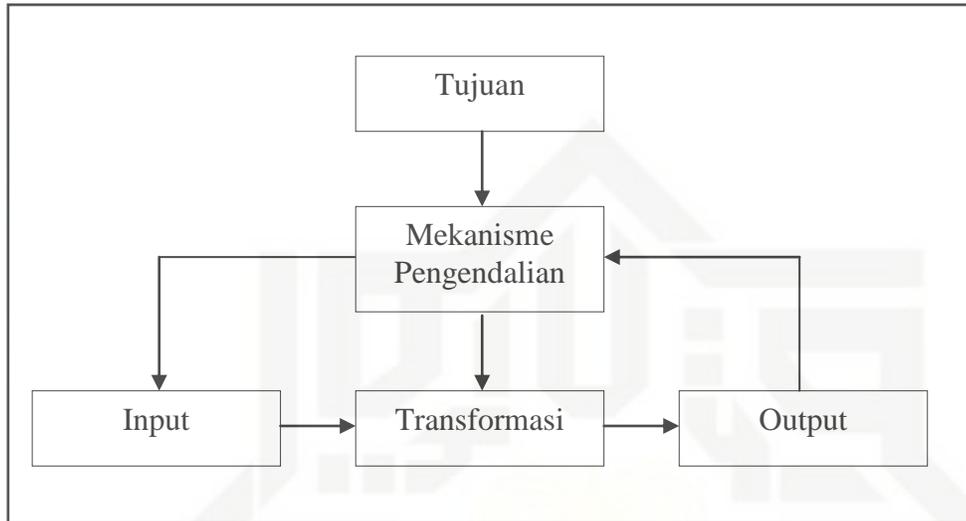
- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

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

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.

9. Mekanisme pengendalian dan umpan balik mekanisme pengendalian (*control mechanism*) diwujudkan dengan menggunakan umpan balik (*feedback*), yang mencuplik keluaran. Tidak semua sistem memiliki kombinasi elemen-elemen yang sama, tetapi susunan dasarnya sama.



Gambar 2.1 Bagian-Bagian Komponen dari Suatu Sistem

2.1.3 Klasifikasi Sistem

Sistem dapat diklasifikasikan dalam beberapa sudut pandang sebagai berikut (Rosa, 2011) :

1. Sistem Abstrak dan Sistem Fisik

Sistem abstrak (*abstract system*) adalah sistem yang berupa pemikiran atau gagasan yang tidak tampak secara fisik. Misalnya, sistem agama/teologi. Sistem fisik (*physical system*) adalah sistem yang ada secara fisik dan dapat dilihat dengan mata. Misalnya sistem komputer, sistem akuntansi, dan sistem transportasi.

2. Sistem Alamiah dan Sistem Buatan

Sistem alamiah (*natural system*) adalah sistem yang terjadi karena proses alam, bukan buatan manusia. Misalnya sistem tata surya, sistem rotasi bumi.



Sistem buatan manusia (*human made system*) adalah sistem yang terjadi melalui rancangan atau campur tangan manusia. Misalnya sistem komputer, sistem transportasi.

3. Sistem Tertentu dan Tak Tentu

Sistem tertentu (*deterministic system*) adalah sistem yang operasinya dapat diprediksi secara cepat dan interaksi di antara bagian-bagiannya dapat dideteksi dengan pasti. Misalnya, sistem komputer karena operasinya dapat diprediksi berdasarkan program yang dijalankan. Sistem tak tentu (*probabilistic system*) adalah sistem yang hasilnya tidak dapat diprediksi karena mengandung unsur probabilitas. Misalnya, sistem persediaan, sistem pemilu.

4. Sistem Tertutup Dan Terbuka

Sistem tertutup (*closed system*) adalah sistem yang tidak berhubungan dengan lingkungan di luar sistem. Sebenarnya sistem tertutup tidak ada, yang ada adalah relatif tertutup. Sistem terbuka (*open system*) adalah sistem yang berhubungan dengan lingkungan luar dan dapat terpengaruh dengan keadaan lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan output untuk subsistem yang lain. Misalnya sistem arloji, sistem ATM.

2.2 Sistem Pemantauan

Pemantauan visual dimulai pada akhir abad 19 untuk membantu petugas penjara dalam penemuan metode melarikan diri. Tidak sampai pertengahan abad 20 bahwa pengawasan diperluas untuk mencakup keamanan properti dan orang-orang. Teori di balik pemantauan visual yang didirikan mempunyai empat faktor kunci yang lazim (Caputo, 2014). Faktor-faktornya adalah:

1. Pencegahan

Jika seorang penjahat potensial menyadari kemungkinan diawasi dan direkam, dia dapat menentukan bahwa ada risiko dideteksi. Negara- negara di seluruh dunia menggunakan *video surveillance* (pengawasan video), fokus penggunaannya sebagian besar pada transportasi umum dan pilih area publik.

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.



86% dari instalasi internasional dimaksudkan untuk mencegah dan mendeteksi pencurian, dan 39% juga berfungsi sebagai pencegah kejahatan kekerasan.

2. Efisiensi

Meninjau rekaman *video surveillance* (pengawasan video) pada saat yang sama seperti menonton pemantauan langsung memberikan informasi tambahan mengenai situasi, yang memungkinkan pengguna untuk membuat keputusan yang lebih baik. Tergantung pada jumlah kamera keamanan dan lokasi mereka, melihat video secara langsung dan diarsipkan dapat mengkonfirmasi akan kegiatan ilegal yang dilakukan pelanggan, atau tersangka sebelum didekati oleh petugas keamanan.

3. Capable guardian

Dalam *The American Sociologist Reviews "Social Change and Crime Rate Trends: A Routine Activity Approach,"* oleh Lawrence Cohen dan Marcus Felson, penulis menyarankan bahwa pencegahan kejahatan termasuk kehadiran *video surveillance* yang "mampu" mengawasi guardian. Aktor atau *guardian* tidak perlu untuk hadir, hanya menonton, dan hari ini bahkan tidak menonton hanya pengarsipan tetapi menggunakan teknologi yang cerdas.

4. Deteksi

Deteksi adalah faktor keberhasilan yang lebih tinggi, memberikan bukti nyata bahwa *camera surveillance* (pengawasan kamera) bekerja. Britania terkenal sistem pengawasan video, menyediakan penegakan hukum dengan kemampuan untuk mengikuti siapa pun di seluruh kota London dengan menggunakan lebih dari 200.000 kamera.

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 mengumunkan 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.

2.3 Motion

Motion merupakan suatu aplikasi yang mampu memonitoring sinyal video dari sebuah kamera dan mampu mendeteksi perubahan yang terjadi pada potongan video. *Motion* berkerja secara penuh dalam mode *text* dan tidak memiliki *interface* (antarmuka). *Motion* dikembangkan pertama kali oleh Jeroen Vreeken dan kemudian dilanjutkan oleh Folkert van Heusden dan Kenneth Lavrsen. Aplikasi *motion* sendiri ditulis menggunakan bahasa C dan memiliki *output* berupa gambar jpg ataupun video mpg (Lavrsen, 2013).

Motion bekerja dengan membandingkan intensitas pixel dari gambar baru dengan gambar referensi (gambar lama). Dalam proses perbandingan ini warna dalam suatu gambar tidak diperlukan dan hanya diambil citra hitam putihnya saja. Aplikasi *motion* memiliki fitur antara lain (1) *Live Streaming Webcam*, (2) *Capture* gambar saat terdeteksi gerakan, (3) *Motion Detecion*, dan (4) Pengontrolan *Threshold*. (Lavrsen, 2013)

2.3.1 Live Streaming Webcam

Webcam menghasilkan *streaming video* dalam format MJPEG. Pada *webcam maxrate* dan *webcam quality* adalah pilihan penting untuk membatasi beban pada *server* dan *link*. Tidak disarankan untuk menetapkan nilai terlalu tinggi, kecuali jika hanya ingin menggunakannya pada *localhost* atau pada LAN internal. Pilihan *webcam quality* setara dengan tingkat kualitas untuk *jpeg pictures*.



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

2.3.2 Capture Gambar Saat Terdeteksi Gerakan

Bagian ini, *motion* mengontrol bagaimana menghasilkan gambar saat mendeteksi gerakan.

```
#####
# Image File Output
#####

#Output 'normal' pictures when motion is detected (default: on)
output_normal on

#Output pictures with only the pixels moving object (green ghost images) (default: off)
output_motion off

#The quality (in percent) to be used by the jpeg compression (default: 75)
quality 75
```

Gambar 2.2 Pengaturan Menangkap Gambar Dengan Motion (Lavrson, 2013)

2.3.3 Motion Detection

Motion dapat bekerja melakukan deteksi gerakan atau *motion detection*. Dalam penggunaannya terdapat beberapa kontrol untuk mendeteksi gerakan. Dalam penelitian ini penulis menggunakan control pada areadeteksi.

1. Area Deteksi

Mendeteksi daerah gerakan di pusat adalah pengaturan *default*. Sebuah *script* (on_area_detected) segera dimulai ketika pusat gerakan terdeteksi disalah satu wilayah yang diberikan.

1	2	3
4	5	6
7	8	9

Gambar 2.3 Area Deteksi Gerakan (Lavrson, 2013)



2.3.4 Threshold

Ambang batas atau *threshold* digunakan untuk menyatakan gerak. Ambang batas adalah jumlah piksel yang berubah dihitung setelah *noise filtering*, *masking*, *despeckle*, dan *labelling*.

Pilihan *threshold* adalah pengaturan deteksi yang paling penting. Ketika gerakan berjalan *threshold* akan membandingkan *frame* gambar ini dengan *frame* sebelumnya dan menghitung jumlah piksel yang berubah setelah diproses gambar dengan *noise filtering*, *masking*, *despeckle* dan *labelling*. Jika terjadi perubahan piksel dari yang ditentukan oleh 'ambang dapat dianggap bahwa telah terdeteksi adanya gerakan (Lavrson, 2013).

2.4 Metodologi Pengembangan Sistem

Pemrakarsa metodologi biasanya hanya mempublikasikan langkah-langkah apa yang harus dilakukan dan bagaimana urutan pengerjaannya. Filosofi yang ada dibalik sebuah metodologi biasanya tidak disertakan dalam publikasi. Hal ini tentunya sangat menyulitkan bagi pengguna dalam memahami secara utuh terhadap suatu metodologi (Prasetyo, 2010).

Proses membangun perangkat lunak komputer dan sistem informasi telah selalu ditentukan oleh metodologi pengembangan yang berbeda. Sebuah metodologi pengembangan perangkat lunak mengacu pada kerangka yang digunakan untuk merencanakan, mengelola, dan mengontrol proses pengembangan sistem informasi (Ville, 2010).

2.4.1 Pengertian SDLC (*System Development Life Cycle*)

Model SDLC atau Sekuensial Linier sering disebut juga Model Air Terjun. Model ini mengusulkan sebuah pendekatan perkembangan perangkat lunak yang sistematis dan sekuensial yang dimulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian, dan pemeliharaan. Model ini disusun bertingkat, setiap tahap dalam model ini dilakukan berurutan, satu sebelum yang lainnya. Model ini biasanya digunakan untuk membuat sebuah software dalam skala besar dan yang akan dipakai dalam waktu yang lama (Saputra, 2012).



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.

memberikan masukan ke *Analist*, *User Operation Worker* memberikan masukan ke *Analist*, *Sistem Management* memberikan masukan ke *Analist*.

b. *Analyzing System Needs*

Menganalisis kebutuhan sistem dengan menguraikan suatu sistem informasi yang utuh ke dalam komponen-komponennya untuk mengevaluasi peluang, dan hambatan yang terjadi. Ini dilakukan dengan menggunakan *dataflow diagram* sehingga dapat diketahui input, proses dan output dari sistem, serta kamus data untuk mendaftarkan item data dan spesifikasi yang digunakan dalam sistem. Selanjutnya membuat keputusan yang terstruktur untuk memperoleh kondisi dan tindakan alternatif melalui tabel keputusan atau pohon keputusan. Pada tahap ini juga dilakukan pembuatan proposal sistem yang berisikan ringkasan apa yang telah ditemukan tentang pengguna, kelebihan dan kekurangan sistem yang ada, menyediakan keuntungan dan biaya analisis, membuat rekomendasi tentang apa yang harus dilakukan. Yang terlibat adalah *Analist* bertugas melakukan pengumpulan data, menganalisis, dan observasi terhadap sistem yang telah ada yang akan digunakan sebagai informasi dalam perencanaan sistem, *User Management* memberikan masukan ke *Analist*, *Sistem Management* memberikan masukan ke *Analist*.

3. Fase Perancangan Sistem secara Umum

Aktivitas yang dilakukan pada fase ini adalah mendesain prosedur bagi pengguna untuk entri data secara akurat dan efektif, mendesain *interface* (antarmuka) bagi pengguna, mendesain *database* yang akan digunakan untuk menyimpan data yang diperlukan oleh pengambil keputusan, mendesain output baik *onscreen* atau tercetak, terakhir desain kontrol dan prosedur *backup*. Yang terlibat yaitu sistem analisis memberikan dokumen yang berisikan usulan dan laporan final desain kepada pemakai informasi,

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.

mengerjakan pohon keputusan, sistem desainer mengerjakan sistem kontrol, berkas dan *database*

4. Fase Kontruksi atau Implementasi

Fase ini terdiri dari:

a. *Developing and Documenting Software*

Tahap ini dikatakan sebagai fase mengembangkan dan mendokumentasikan perangkat lunak dimana aktivitas yang dilakukan adalah mengembangkan perangkat lunak melalui kerjasama sistem analis dengan programmer, mengembangkan dokumentasi melalui kerjasama antara sistem analis dengan pengguna. *Programmer* membuat dokumentasi dari perangkat lunak baik prosedur manual, pertanyaan yang paling sering ditanyakan oleh *user* (*Frequently Asked Questions*) karena pengguna terlibat dari awal pengembangan sistem, "*Read Me*" file.

Yang terlibat yaitu analis mendesain program komputer menggunakan struktur *charts* dan *pseudo code*, *programmer* mengerjakan program berdasarkan desain dari analis.

b. *Testing and Maintaining The Sistem*

Sebelum suatu sistem informasi dapat digunakan maka perlu dilakukan pengujian yang dilakukan oleh *programmer* dan sistem analis saja, kalau terjadi masalah maka segera dapat dilakukan perbaikan sistem, pengetesan dilakukan dengan memakai sampel data aktual. Disinilah fase perawatan terhadap sistem dimulai. Yang terlibat yaitu analis melakukan pengujian dari program yang dibuat, apakah sudah sesuai dengan dokumentasi yang sudah dibuat, *programmer* melakukan perbaikan jika ada kesalahan atau ada yang tidak sesuai dengan dokumentasi yang dibuat analis, *user management* ikut melakukan pengujian program.

c. *Implemeting and Evaluating The Sistem*

Pada fase ini dilakukan instalasi sistem, konversi sistem, dan evaluasi sistem. Pada tahap ini juga sudah masuk ke fase perawatan sistem dengan melakukan *review* dan evaluasi sistem. Yang terlibat adalah analis



2.4.3 Kelebihan dan Kekurangan System Development Life Cycle

1. Perawatan (*Maintenance*)

Fase perawatan telah dimulai dari :

 - a. *Testing and Maintaining The Sistem*

Dilakukan pengetasan sistem apakah sistem ada yang mengalami kegagalan (*bug*) dan sekaligus dilakukan perawatan terhadap sistem. Yang terlibat adalah *analist* melakukan pengujian dari program yang dibuat, apakah sudah sesuai dengan dokumentasi yang sudah dibuat, *programmer* melakukan perbaikan jika ada kesalahan atau ada yang tidak sesuai dengan dokumentasi yang dibuat *analist*, *user management* ikut melakukan pengujian program
 - b. *Implementing and Evaluating The Sistem*

Pada fase ini dilakukan pelatihan terhadap karyawan untuk menggunakan sistem , konversi sistem, dan evaluasi sistem. Yang terlibat yaitu analis memberikan pelatihan kepada karyawan dan melakukan konversi sistem, *programmer* melakukan instalasi, menambah fitur dan memperbaiki program.

2.4.3 Kelebihan dan Kekurangan System Development Life Cycle

1. Kelebihan
 - a. Mudah diaplikasikan.
 - b. Memberikan *template* tentang metode analisis, desain, pengkodean, pengujian, dan pemeliharaan.
2. Kekurangan
 - a. Jarang sekali proyek riil mengikuti aliran sekuensial yang dianjurkan model karena model ini bisa melakukan itersi tidak langsung.
 - b. Pelanggan sulit untuk menyatakan kebutuhan secara eksplisit sehingga sulit untuk megakomodasi ketidakpastian pada saat awal proyek.



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.

- c. Pelanggan harus bersikap sabar karena harus menunggu sampai akhir proyek dilalui. Sebuah kesalahan jika tidak diketahui dari awal akan menjadi masalah besar karena harus mengulang dari awal.
- d. Pengembang sering malakukan penundaan yang tidak perlu karena anggota tim proyek harus menunggu tim lain untuk melengkapi tugas karena memiliki ketergantungan hal ini menyebabkan penggunaan waktu tidak efisien.

2.5 Metodologi Pengumpulan Data

Pengumpulan data dilakukan untuk memperoleh informasi yang dibutuhkan dalam rangka mencapai tujuan penelitian (Gulo, 2002). Data berasal dalam dua bentuk utama, tergantung pada kedekatan dengan *event* yang direkam. Data yang telah diamati, *experienced*, atau direkam mendekati dengan event, dan terdekat untuk mendapat kebenaran, disebut data primer. Sumber-sumber tertulis yang menafsirkan atau merekam data primer disebut sumber sekunder (Walliman, 2011).

2.5.1 Data Primer

Terdapat beberapa metode dasar yang digunakan untuk mengumpulkan data primer. Yang utama adalah mengajukan pertanyaan, melakukan wawancara, mengamati tanpa terlibat, membenamkan diri dalam situasi, melakukan eksperimen, memanipulasi model. Dalam penelitian ini dilakukan wawancara dan observasi atau mengamati tanpa terlibat (*observing without getting involved*) (Walliman, 2011).



2.5.1.1 Observasi

Observasi adalah metode pengumpulan data melalui *observing without getting involved* (mengamati tanpa terlibat) dan bukan mengajukan pertanyaan. Tujuannya adalah untuk mengambil pandangan terpisah dari fenomena, dan menjadi 'tak terlihat', baik dalam fakta atau efek (seperti *being ignored* pada orang lain). Pengamatan dapat digunakan untuk merekam data tentang acara dan kegiatan, dan kondisi alam benda seperti bangunan atau artefak. Jenis observasi sering disebut sebagai *survey* (Walliman, 2011)

2.5.1.2 Wawancara

Wawancara adalah bentuk komunikasi langsung antara peneliti dan responden. Komunikasi berlangsung dalam bentuk Tanya-jawab dalam hubungan tatap muka, sehingga gerak dan mimik responden merupakan pola media yang melengkapi kata-kata secara verbal (Gulo, 2002)

Wawancara sangat berguna ketika data kualitatif diperlukan. Terdapat tiga jenis wawancara, yaitu wawancara terstruktur, wawancara tidak terstruktur dan wawancara semi-terstruktur. Dalam penelitian ini penulis menggunakan semi-terstruktur, dimana terdapat sesi wawancara terstruktur dan tidak terstruktur dengan standar tertentu dan pertanyaan terbuka (Walliman, 2011).

2.5.2 Data Sekunder

Semua studi penelitian memerlukan data sekunder untuk latar belakang penelitian, sebagai peneliti pasti akan perlu untuk memastikan konteks penelitian baik mengenai pertanyaan atau masalah pada penelitian, dan juga mendapatkan ide dari teori saat ini dengan menggunakan data. Keuntungan menggunakan set data sekunder adalah hasil data telah diproduksi oleh tim peneliti ahli, dan sering dengan anggaran besar dan sumber daya yang luas, di luar sarana dan kemampuan mahasiswa sendiri.

Terdapat banyak jenis dari data sekunder, sumber dokumenter utama berupa bahan tertulis dan non-tertulis, dan data survei dalam bentuk informasi statistik (Walliman, 2011)



2.5.2.1 Studi Pustaka

Studi Pustaka adalah pencarian sumber-sumber atau opini pakar tentang suatu hal yang berkaitan dengan tujuan penelitian. Dengan kata lain, studi pustaka merupakan pengkajian beberapa sumber pustaka yang umumnya terdapat di perpustakaan yang terkait dengan variabel-variabel utama atau topik sebuah penelitian. Maka dapat dikatakan bahwa hampir semua jenis penelitian memasukkan studi pustaka sebagai salah satu langkah yang ditempuh dalam keseluruhan penelitian (Patrisius, 2015).

2.6 UML (*Unified Modelling Language*)

Pada perkembangan teknik perkembangan berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modelling Language* (UML). UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung (Rosa, 2011).

Unified Modelling Language (UML) adalah satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek. UML memiliki beberapa diagram, diantaranya (Whitten, 2004) :

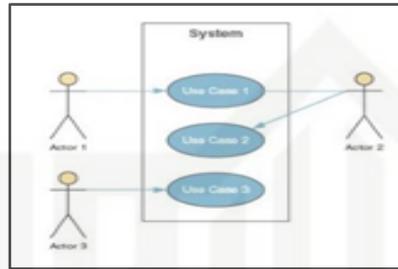
1. *User Case Diagram*

Use case atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

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

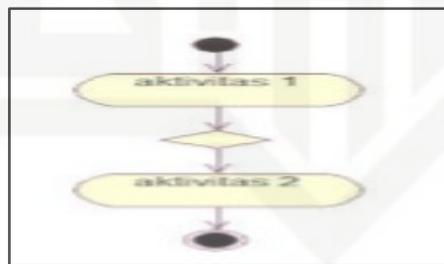
- a. Aktor merupakan orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
- b. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antarunit atau aktor.



Gambar 2.4 Contoh *Use Case Diagram*

2. *Activity Diagram*

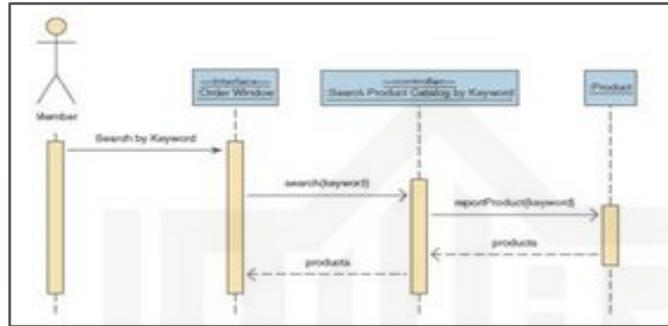
Activity Diagram memodelkan langkah-langkah proses atau kegiatan sistem. Diagram ini serupa dengan *flowchart* di mana secara grafis diagram ini menggambarkan aliran sekuensial dari kegiatan proses bisnis atau sebuah *use case*. Diagram ini berbeda dari *flowchart* dimana diagram ini menyediakan sebuah mekanisme untuk menggambarkan kegiatan yang tampak secara parallel.



Gambar 2.5 Contoh *Activity Diagram*

3. Sequence Diagram

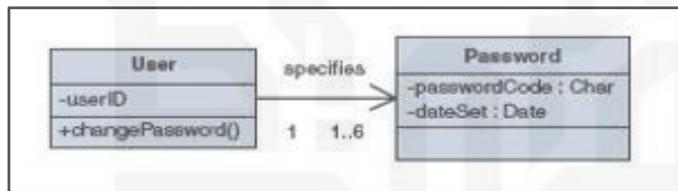
Sequence Diagram menggambarkan dengan sangat detil bagaimana sebuah objek berinteraksi satu sama lain sepanjang waktu. Diagram ini memodelkan logika sebuah *use case* dengan cara interaksi pesan di antara objek-objek dalam rangkaian waktu.



Gambar 2.6 *Sequence Diagram* Mencari Produk Katalog Dengan *Keyword*

4. Class Diagram

Class Diagram menggambarkan struktur objek sistem. Diagram ini menunjukkan kelas objek yang menyusun sistem dan juga berhubungan antara kelas objek tersebut.



Gambar 2.7 Contoh *Class Diagram*

2.7 Raspberry Pi

Raspberry Pi adalah sebuah SBC (*single-board computer*) seukuran kartu kredit. *Raspberry Pi* telah dilengkapi dengan semua fungsi layaknya sebuah komputer lengkap, menggunakan SOC (*System-on-a-Chip*) ARM yang dikemas dan diintegrasikan di atas PCB (papan sirkuit). *Raspberry Pi* ini mampu bekerja layaknya komputer pada umumnya dengan kemampuan untuk menjalankan sistem operasi Linux dan aplikasinya seperti *Libre Office*, multimedia (*audio* dan *video*), peramban *web*, ataupun *programming*. *Raspberry Pi* dapat menampilkan gambar ke TV/HDTV menggunakan koneksi HDMI ataupun TV standar menggunakan kabel RJ45 (Prihantono, 2013).

Terlebih lagi, *Raspberry Pi* memiliki kemampuan untuk berinteraksi dengan dunia luar, dan telah digunakan dalam berbagai macam proyek pembuat digital, dari mesin musik dan detektor orangtua untuk cuaca stasiun dan *tweeting birdhouse* dengan kamera infra-merah.

2.7.1 Raspberry Pi 3 Model B

Raspberry Pi 3 memberikan kapasitas 3 kali pengolahan yang lebih cepat dibandingkan dengan generasi sebelumnya. Ini generasi ketiga *Raspberry Pi* yang memiliki *upgrade Broadcom Prosesor BCM2837*, yang merupakan *ARM Cortex-A53* yang kuat berbasis *quad-core prosesor* dan sudah mendukung prosesor 64-bit yang berjalan pada 1.2 GHz, memiliki RAM 1 *GByte*. *Board* ini juga ditingkatkan dalam hal konektifitasnya, seperti penambahan *modul WI-FI dan Bluetooth onboard*. Untuk Spesifikasi lebih lengkap dapat dilihat pada Tabel 2.1 dan Tabel 2.2

Tabel 2.1 : Spesifikasi *Raspberry Pi 3 Model B*

Specifications	
Chip	Broadcom BCM2836 SoC
Core Architecture	64-bit Quad-Core ARM Cortex-A53
CPU	1.2 GHz
GPU	Dual core Videocore IV® Multimedia co-processor Provides Open GL ES 2.0, hardware-accelerated OpenVG,



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

© Hak cipta milik UIN Suska Riau	and 1080p30 H.264 high-profile decode Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure
Memory	1GB LPDDR2
Operating System	Boots from Micro SD Card, running a version of the linux operating system
Dimension	85 x 56 x 17 mm
Power	Micro USB socket 5V, 2.5A

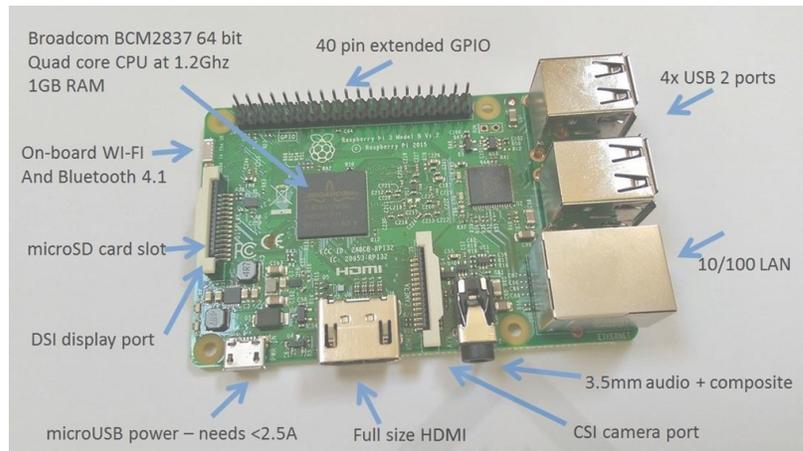
(Sumber: uk.rs-online.com)

Tabel 2.2 : Konektor Raspberry Pi 3 Model B

Connectors	
Ethernet	10/100 BaseT Ethernet socket
Video Output	HDMI (rev 1.3 and 1.4)
Audio Output	3.5mm jack, HDMI
USB	4 x USB 2.0 Connector
GPIO Connector	40-pin 2.54mm (100 mil) expansion header: 2x20 strip
	Memiliki 27 GPIO pin dengan 3,3 V, 5 V dan GND supply lines
Chip Antenna	WiFi and Bluetooth 4.1
Camera Connector	15-pin MIPI Camera Serial Interface (CSI-2)
Display Connector	Display Serial Interface (DSI) 15 way flat flex cable connector
	with two data lanes and a clock lane
Memory Card Slot	Micro SDIO

(Sumber: uk.rs-online.com)

- 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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.



Gambar 2.8 Arsitektur *Raspberry Pi 3 Model B*
(Sumber: tech.scargill.net/raspberry-pi-3-grand-opening)

2.8 Pi Camera

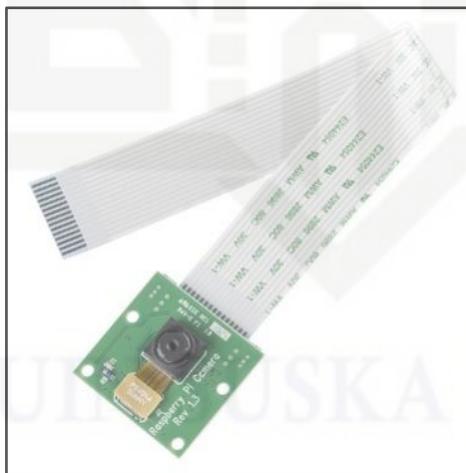
Sejak peluncuran *Raspberry Pi* model A, orang di seluruh dunia sedang menjajaki dalam berbagai cara. Ilmuwan komputer dan peneliti telah meneliti kamera *USB* dengan *Raspberry Pi*, bagaimanapun, tidak semua kamera *USB* didukung dengan *Raspberry Pi*, sehingga yayasan *Raspberry Pi* memutuskan untuk membangun sebuah modul kamera yang dapat langsung terhubung dengan konektor *CSI Raspberry Pi*. modul kamera 5MP yang mampu menangkap video 1080p dan gambar diam. Modul berukuran sekitar 25mm * 20mm * 9mm dan berat hanya lebih dari 3 gram, yang membuatnya sempurna untuk *mobile* atau aplikasi lain di mana ukuran dan berat memainkan peran penting (Samarth Shah, 2015)

Ada beberapa metode penyiaran yang biasa dilakukan. Metode yang paling umum adalah menggunakan *software* yang dapat mengubah gambar menjadi file dengan format JPEG, dan mengunggah ke *web server* menggunakan *File Transfer Protocol* (FTP).



Tabel 2.3 : Spesifikasi Pi Camera Modul

<i>Product Name</i>	<i>Pi Camera Module</i>
<i>Image Sensor</i>	<i>Omnivision 5640 CMOS image sensor in a auto-fouces module with integral IR filter</i>
<i>Still Resolution</i>	<i>5 megapixels</i>
<i>Acctive Array Size</i>	<i>2592 X 1944</i>
<i>Max Frame Rate</i>	<i>1080P 30fps@24Mhz</i>
<i>Picture Formats</i>	<i>PEG PNG YUV420 RGB888</i>
<i>Video Formats</i>	<i>Raw h.264</i>
<i>Connection</i>	<i>40 Pin FPC to the Camera Sensor Interface (CSI-0)</i>
<i>Image Control Functions</i>	<ul style="list-style-type: none"> - <i>Automatic exposure control (AEC)</i> - <i>Automatic white balance (AWB)</i> - <i>Automatic black level calibration (ABLC)</i> - <i>Automatic band filter, Mirror and flip</i>
<i>Temp Range</i>	<ul style="list-style-type: none"> - <i>Operating: -30°C to 70°C</i> - <i>Stable Image: 0°C to 50°C</i>
<i>Lens Size</i>	<i>1/4"</i>
<i>Dimension</i>	<i>36 x 32 x 10mm</i>
<i>Weight</i>	<i>5g</i>



Gambar 2.9 Pi Camera v1.3
(Sumber: www.sparkfun.com)

- 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 mengumumakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.



- 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.

2.9 Bash Shell

Bash adalah *shell*, atau *command language interpreter* atau dalam bahasa Indonesia adalah penerjemah bahasa perintah, untuk sistem operasi GNU. *Bash* sendiri adalah singkatan untuk 'Bourne-Again Shell', dimana Stephen Bourne adalah penulis pertama untuk *Unix shell* `sh`, yang muncul dalam *Seventh Edition Bell Labs Research* versi Unix.

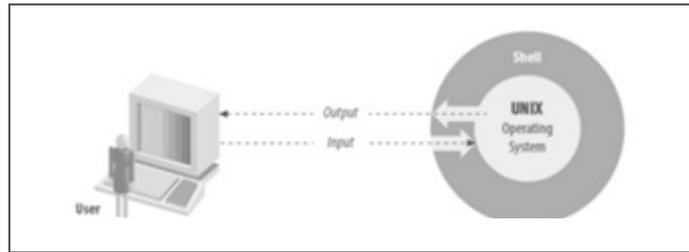
Pada dasarnya, *shell* hanyalah sebuah prosesor makro yang mengeksekusi perintah. Istilah prosesor makro berarti fungsi di mana teks dan simbol diperluas untuk membuat ekspresi yang lebih besar. Sebuah *shell Unix* merupakan sebuah *interpreter* perintah dan bahasa pemrograman. Sebagai penerjemah perintah, *shell* menyediakan antarmuka pengguna untuk set kaya utilitas GNU. Fitur bahasa pemrograman memungkinkan utilitas ini untuk digabungkan. File yang berisi perintah dapat dibuat, dan menjadi perintah sendiri. Ini perintah baru memiliki status yang sama seperti perintah sistem di direktori seperti `/bin`, memungkinkan pengguna atau kelompok untuk membangun lingkungan kustom untuk mengotomatisasi tugas umum mereka (www.gnu.org, 2012).

Shell adalah sebuah *user interface* untuk sistem operasi UNIX, dalam pemrograman apapun yang perlu mengambil masukan atau *input* dari pengguna, akan diterjemahkan ke dalam instruksi sehingga sistem operasi dapat mengerti, dan menyampaikan *output* sistem operasi kembali ke pengguna.

Terdapat berbagai jenis antarmuka pengguna, *bash* termasuk dalam kategori yang paling umum, yang dikenal sebagai *character-based user interface*. Antarmuka ini menerima jenis baris perintah tekstual yang pengguna ketikkan; antarmuka ini biasanya diproduksi keluaran berbasis teks (Newham, 2005)

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.



Gambar 2.10 *Bash Shell*

2.10 TightVNC

Enterprise (2010) menyatakan bahwa TightVNC adalah software khusus yang dapat digunakan untuk mengendalikan komputer dari jarak jauh. Software ini berperan sebagai *remote control* yang memudahkan pelaksanaan tugas-tugas dari mana saja. TightVNC memiliki dua bagian, yaitu *server* dan *viewer*. *Server* berfungsi untuk membagi layar dari komputer tempat ia berjalan, sedangkan *viewer* akan menampilkan layar jarak jauh dari *server*. Berikut fasilitas baru yang dihadirkan oleh TightVNC:

1. *Scalling of Remote Desktop*

Dapat mengatur tampilan *remote desktop* sesuai pilihan, apakah berbentuk satu layar penuh atau lebih kecil dari ukuran yang sebenarnya.

2. *Automatic SSH Tunneling on Unix*

Pengguna Unix dapat menggunakan koneksi via *SSH/OpenSSH* secara otomatis.

3. *Enhanced Web Browser Access*

TightVNC dilengkapi dengan *Java Viewer* yang dapat diakses langsung melalui internet.

4. *Support for Video Driver*

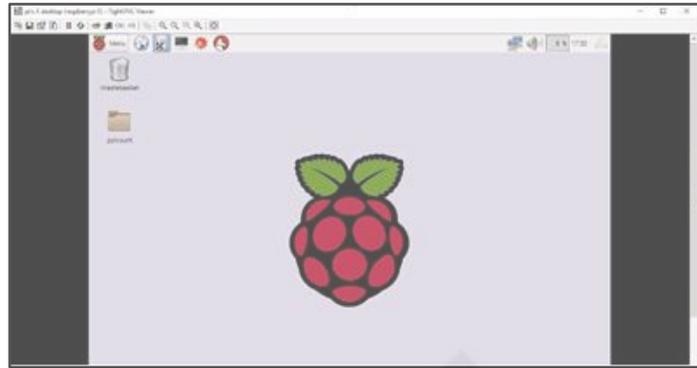
Server TightVNC dapat dihubungkan dengan video atau *webcam* yang terdapat pada komputer.

5. *Support for Two Passwords*

Dapat menggunakan dua password berbeda untuk bagian *server* dan *viewer*.

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



Gambar 2.11 Tampilan Desktop TightVNC

2.11 PuTTY

PuTTY adalah SSH dan *telnet client*, awalnya dikembangkan oleh Simon Tatham untuk platform Windows. PuTTY adalah perangkat lunak *open source* yang mempunyai dengan sumber-sumber kode dan dikembangkan dan didukung oleh sekelompok relawan. (<http://www.putty.org/>)

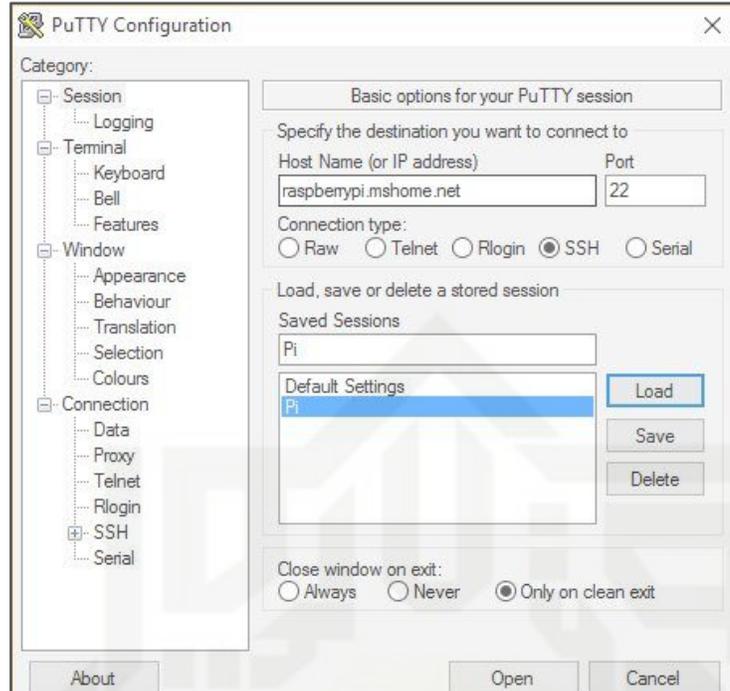
PuTTY adalah aplikasi klien *telnet* dan SSH yang paling populer untuk pengguna Windows yang merupakan *software remote console* / terminal yang digunakan untuk buat *remote connection* komputer melalui Port SSH atau sebagainya.

Sedangkan, *Secure Shell* atau SSH adalah protokol jaringan yang memungkinkan pertukaran data melalui saluran aman antara dua perangkat jaringan. Terutama banyak digunakan pada sistem berbasis Linux dan Unix untuk mengakses akun *shell*, SSH dirancang sebagai pengganti *Telnet* dan *shell remote* tak aman lainnya, yang mengirim informasi, terutama kata sandi, dalam bentuk teks sederhana yang membuatnya mudah untuk dicegat. Enkripsi yang digunakan oleh SSH menyediakan kerahasiaan dan integritas data melalui jaringan yang tidak aman seperti Internet. SSH menggunakan kriptografi kunci publik untuk mengotentikasi komputer remote dan membiarkan komputer *remote* untuk mengotentikasi pengguna, jika perlu. SSH juga biasanya digunakan untuk *login* ke mesin *remote* (Sugianto, 2013).



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



Gambar 2.12 Tampilan PuTTY

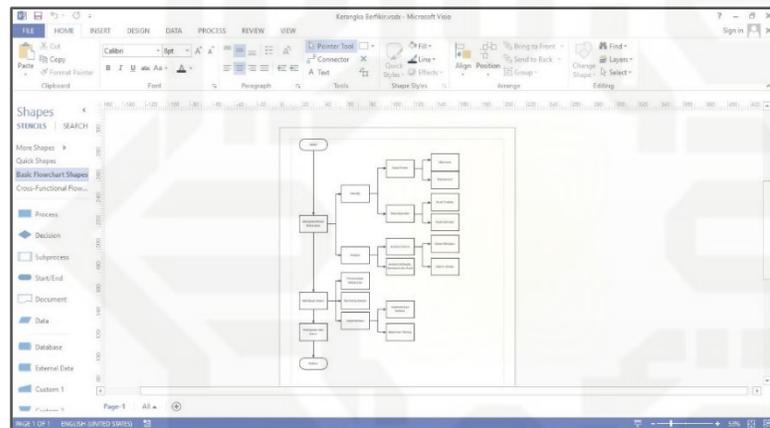
2.12 Black-Box Testing

Pengujian *black-box*, juga disebut pengujian perilaku atau *behavioral testing*, berfokus pada persyaratan fungsional perangkat lunak. Pengujian *black-box* berusaha untuk menemukan kesalahan pada beberapa kategori berikut: (1) fungsi-fungsi yang salah atau hilang, (2) kesalahan *interface* (antar muka), (3) kesalahan dalam struktur data atau akses *database* eksternal, (4) kesalahan pada performa atau *behavior*, dan (5) kesalahan pada inisialisasi dan terminasi (Pressman, 2010).

2.13 Microsoft Visio 2013

Visio adalah sebuah aplikasi yang memungkinkan untuk membuat diagram visual yang khas dan profesional yang dapat digunakan dalam berbagai pengaturan, mata pelajaran, dan profesi.

Ide dibalik Visio adalah untuk menyediakan alat-alat standar yang memungkinkan untuk dengan mudah merakit gambar atau diagram menggunakan blok bangunan dasar atau bentuk. Pendekatan ini untuk pembuatan diagram memungkinkan bagi siapa saja untuk membuat chart yang menakjubkan. Template membuat menjadi lebih mudah dengan menyediakan titik awal dasar yang menentukan format dan yang stensil digunakan.



Gambar 2.13 Tampilan Microsoft Visio 2013

2.14 JSON

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman *JavaScript*, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh *programmer* keluarga C termasuk C, C++, C#, *Java*, *JavaScript*, *Perl*, *Python* dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran data (www.json.org).



2.14.1 Struktur JSON

1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel *hash* (*hash table*), daftar berkunci (*keyed list*), atau *associative array*.
2. Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

2.15 Python

Python diciptakan pada awal 1990-an oleh Guido van Rossum di Stichting pusat matematika di Belanda sebagai penerus bahasa yang disebut ABC. Guido tetap penulis utama Python, meskipun itu mencakup banyak kontribusi dari orang lain.

Pada tahun 1995, Guido melanjutkan karyanya pada Python di Corporation untuk National Research Initiatives di Reston, Virginia di mana ia merilis beberapa versi perangkat lunak.

Pada Mei 2000, Guido dan tim pengembangan inti Python pindah ke BeOpen.com untuk membentuk tim BeOpen PythonLabs. Pada bulan Oktober tahun yang sama, tim *PythonLabs* pindah ke Digital Creations. Pada tahun 2001, *Python Software Foundation* (dibentuk, sebuah organisasi non-profit yang diciptakan khusus untuk memiliki Python terkait Kekayaan Intelektual. *Zope Corporation* adalah anggota mensponsori dari PSF.

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.