

BAB II

LANDASAN TEORI

Pada landasan teori ini akan dijelaskan mengenai teori – teori yang berhubungan dengan judul penulisan ini. Adapun landasan teori yang mendasari penulisan skripsi ini adalah :

2.1. APLIKASI

Aplikasi adalah program yang digunakan untuk melakukan pekerjaan tertentu, misalkan aplikasi untuk menghitung gaji suatu perusahaan dan aplikasi untuk kasir dalam swalayan. Biasanya aplikasi dibuat oleh perusahaan atas permintaan seseorang/lembaga perusahaan/perusahaan, baik perusahaan besar maupun kecil yang melayani pembuatan aplikasi untuk keperluan perusahaan, lembaga ataupun perorangan (Maryono, 2014).

2.2. KEMAMPUAN BELAJAR

Kemampuan belajar itu sendiri adalah dorongan internal dan eksternal pada siswa-siswa yang sedang belajar untuk mengadakan perubahan tingkah laku, pada umumnya dengan beberapa indikator atau unsur yang mendukung. (Uno, 2007).

Dalam hal belajar, kemampuan untuk melakukan pembelajaran dan menerima pelajaran itu bisa jadi karena dua faktor, yaitu:

1. Faktor Internal, merupakan faktor-faktor yang berasal dari dalam individu yang dapat mempengaruhi hasil belajar. Ada dua yaitu, faktor fisiologis yang berhubungan dengan kondisi fisik individu, seperti keadaan tonus jasmani dan keadaan fungsi jasmani. Dan faktor psikologis yang

berhubungan dengan kondisi psikologis seseorang, seperti kecerdasan, motivasi, minat, sikap dan bakat.

2. Faktor Eksternal adalah faktor yang datang dari luar diri individu yang dapat mempengaruhi proses belajar. Faktor eksternal dibagi menjadi menjadi dua, yaitu: lingkungan sosial, seperti: lingkungan sosial sekolah, lingkungan sosial masyarakat, lingkungan sosial keluarga. Dan lingkungan non sosial, seperti: lingkungan alamiah, faktor instrumental, dan faktor materi pelajaran.

Ada beberapa cara yang dapat membantu untuk meningkatkan kemampuan belajar bagi siswa-siswi, antara lain:

1. Membaca, memahami dan menulis.

Untuk meningkatkan kemampuan belajar, bisa terlebih dahulu untuk membaca pelajaran atau ilmu pengetahuan yang ada. Setelah membaca, pahami setiap pelajaran maupun ilmu pengetahuan yang sedang dipelajari, setelah itu tulis kembali pelajaran maupun ilmu pengetahuan yang dipelajari sesuai kemampuan masing-masing, lakukan hal ini secara rutin. Dengan begitu, secara tidak langsung, otak akan menyimpan setiap pelajaran maupun ilmu pengetahuan yang sudah pelajari.

2. Mencermati.

Setiap kali ada pelajaran maupun ilmu pengetahuan baru. Usahakan untuk mencermatinya dengan baik. Dengan begitu, otak akan berkembang dengan baik dalam mencermati setiap permasalahan yang ada.

3. Belajar Menghafal.

Dengan menghafal, otak akan dilatih untuk meningkatkan daya ingat. Karena, setiap pelajaran maupun ilmu pengetahuan yang baru akan menambah kekuatan memori dalam mengingat.

4. Mengajarkan.

Kebanyakan orang yang sukses dalam hidupnya dan bisa selalu mempunyai ilmu pengetahuan yang baru untuk dibicarakan adalah mereka yang mau untuk berbagi ilmu dengan orang lain. Mau mengajarkan ilmu pengetahuan kepada orang lain banyak memiliki manfaat. Selain bisa mengingat kembali pelajaran maupun ilmu pengetahuan yang dimiliki, juga bisa secara tidak langsung mengembangkan ilmu yang telah dimiliki. (Karya Abadi, 2014)

2.2.1. Tes Kemampuan Belajar

Beberapa penilaian yang terdapat di dalam tes yang menghasilkan kecenderungan kemampuan belajar adalah sebagai berikut:

1. Kemampuan Verbal

Kemampuan verbal, yaitu kemampuan memahami dan menggunakan bahasa, baik secara lisan maupun tulisan. Tes yang diujikan seperti tes lawan kata (antonim), Bahasa Inggris dan Bahasa Indonesia.

2. Kemampuan Numerik

Kemampuan numerik, yaitu kemampuan, ketepatan dan ketelitian dalam berhitung. Tes yang diujikan adalah matematika dan deret angka.

3. Kemampuan Spasial

Kemampuan spasial, yaitu kemampuan merancang, menggambarkan dan membayangkan suatu benda atau objek secara tepat. Tes yang diujikan adalah tes gambar.

4. Kemampuan *Reasoning*

Kemampuan *reasoning*, yaitu kemampuan memecahkan sebuah masalah. Tes yang diujikan adalah pengetahuan umum.

5. Kemampuan Memori

Kemampuan memori dan kreativitas berpikir, yaitu kemampuan daya ingat (menghafal) dan kreativitas berpikir pada saat menyelesaikan sebuah masalah. Tes yang diujikan adalah tes mengingat (*memory test*). (Leoni, 2008).

6. *Learning Disability*

Memiliki kemampuan tinggi pada bidang tertentu namun sekaligus juga memiliki kelemahan dalam bidang lain yang berbeda. Mengalami kesulitan belajar dan cenderung malas. Tes yang diujikan semua bernilai rendah. (Cruickshank, 1980).

2.3. ANDROID

2.3.1. Pengertian Android

Android merupakan sistem operasi berbasis Linux yang bersifat terbuka (*open source*) dan dirancang untuk perangkat seluler layar sentuh seperti *smartphone* dan komputer tablet. (Sherief, 2014).

Android bersifat *open source* atau sumber terbuka artinya android menyediakan *platform* bagi para *developer* atau pengembang untuk mengembangkan aplikasi android. Android merupakan platform yang lengkap, terbuka dan bebas artinya :

1. Lengkap

Para designer dapat melakukan pendekatan yang komprehensif ketika mereka sedang mengembangkan *platform* android. Sistem operasinya aman dan banyak menyediakan *tools* dalam membangun *software* dan memungkinkan peluang pengembangan aplikasi.

2. Terbuka

Platform android disediakan melalui lisensi terbuka (*open source*) sehingga pengembang dapat dengan bebas mengembangkan aplikasi.

3. Bebas

Tidak ada lisensi atau biaya royalti untuk dikembangkan pada *platform* android. Tidak ada biaya keanggotaan diperlukan. Tidak diperlukan biaya pengujian, aplikasi android dapat didistribusikan dan diperdagangkan dalam bentuk apapun (Safaat, 2011).

2.3.2. Sejarah Sistem Operasi Android

Android dikembangkan oleh Android, Inc., didirikan pada bulan Oktober 2003 di Palo Alto, California, oleh Andy Rubin (pendiri *Danger*), Rich Miner (pendiri *Wildfire Communications, Inc.*), Nick Sears (mantan VP *T-Mobile*) dan Chris White (kepala desain dan pengembangan

antarmuka webTV) untuk mengembangkan “perangkat *smartphone* yang lebih sadar akan lokasi dan preferensi penggunanya”. Awal tujuan pengembangan Android yaitu untuk mengembangkan sebuah sistem operasi canggih yang ditujukan untuk kamera digital, namun pasar untuk perangkat kamera digital tidak cukup besar dan pengembangan Android lalu dialihkan bagi pasar *smartphone* untuk menyaingi Symbian dan Windows Mobile (iPhone Apple belum dirilis saat itu). Android Inc. dioperasikan secara diam-diam, hanya diungkapkan para pengembang sedang menciptakan sebuah perangkat lunak untuk *smartphone*. Pada tahun yang sama, Rubin kehabisan uang, Steve Perlman, seorang teman dekat Rubin meminjaminya \$10.000 tunai dan menolak tawaran saham di perusahaan.

Google mengakuisisi Android Inc. pada tanggal 17 Agustus 2005, menjadikannya sebagai anak perusahaan yang sepenuhnya dimiliki oleh Google. Pendiri Android Inc. seperti Rubin, Miner dan White tetap bekerja di perusahaan setelah diakuisisi oleh Google. Di Google, tim yang dipimpin oleh Rubin mulai mengembangkan *platform smartphone* menggunakan kernel Linux. Google memasarkan *platform* tersebut kepada produsen perangkat seluler dan operator nirkabel, dengan janji bahwa mereka menyediakan sistem yang fleksibel dan bisa diperbarui. Google telah menyeleksi beberapa mitra perusahaan perangkat lunak dan perangkat keras, serta mengisyaratkan kepada operator seluler bahwa kerjasama ini terbuka bagi siapapun yang ingin berpartisipasi.

Pada tanggal 5 November 2007, *Open Handset Alliance* (OHA) didirikan. OHA bertujuan untuk mengembangkan standar terbuka bagi

perangkat seluler. Saat itu, Android diresmikan sebagai produk pertamanya; sebuah *platform* perangkat seluler yang menggunakan kernel Linux versi 2.6. Telepon seluler komersial pertama yang menggunakan sistem operasi Android adalah HTC Dream, yang diluncurkan pada 22 Oktober 2008.

Pada tahun 2010, Google merilis seri Nexus; perangkat *smartphone* dan tablet dengan OS Android yang diproduksi oleh HTC, LG dan Samsung. HTC bekerjasama dengan Google dalam merilis produk *smartphone* Nexus pertama, yakni Nexus One. Seri ini telah diperbarui dengan perangkat yang lebih baru, misalnya telepon pintar Nexus 4 dan tablet Nexus 10 yang diproduksi oleh LG dan Samsung. Pada 15 Oktober 2014, Google mengumumkan Nexus 6 dan Nexus 9 yang diproduksi oleh Motorola dan HTC. Pada 13 Maret 2013, Larry Page memberitahukan dalam postingan blognya bahwa Andy Rubin telah pindah dari divisi Android untuk mengerjakan proyek-proyek baru di Google. Ia digantikan oleh Sundar Pichai, yang sebelumnya menjabat sebagai kepala divisi Google Chrome, yang mengembangkan Chrome Os.

Sejak tahun 2008, Android terus melakukan sejumlah pembaruan untuk meningkatkan kinerja sistem operasi. Setiap versi utama yang dirilis dinamakan secara alfabetis berdasarkan nama-nama makanan pencuci mulut atau cemilan bergula; misalnya, versi 1.5 bernama *Cupcake*, yang kemudian diikuti oleh versi 1.6 *Donut*. Versi terbaru adalah 5.0 *Lollipop*, yang dirilis pada 15 Oktober 2014.

Tampilan Android didasarkan pada manipulasi langsung, menggunakan masukan sentuh yang serupa tindakan di dunia nyata, seperti

menggesek, mengetuk, mencubit, dan membalikkan cubitan untuk manipulasi obyek di layar. Pada bulan Oktober 2012, terdapat 700.000 aplikasi yang tersedia untuk Android, dan sekitar dua puluh lima juta aplikasi telah diunduh dari *Google Play*, toko aplikasi utama Android. Sebuah survei pada bulan April-Mei 2013 menemukan bahwa Android adalah *platform* paling populer bagi para pengembang, digunakan oleh 71% pengembang aplikasi seluler. Dan pada tanggal 3 September 2013, satu miliar perangkat Android telah diaktifkan.

Sifat Android yang terbuka telah membuat bermunculannya sejumlah besar komunitas pengembang aplikasi, dengan menambahkan fitur-fitur baru bagi Android pada perangkat yang secara resmi dirilis dengan menggunakan sistem operasi lain. (Sherief, 2014)

2.3.3. Versi Android

Pada perkembangannya Android selalu mengembangkan sistem operasi yang ditanamkan dalam *smartphone*. Berikut perkembangan sistem operasi yang dipergunakan oleh Android :

1. Android versi 1.1
2. Android versi 1.5 (*Cupcake*)
3. Android versi 1.6 (*Donut*)
4. Android versi 2.0/2.1 (*Éclair*)
5. Android versi 2.2 (*Froyo*)
6. Android versi 2.3 (*Gingerbread*)
7. Android versi 3.0/3.1 (*Honeycomb*)
8. Android versi 4.0/4.1 (*ICS : Ice Cream Sandwich*)

9. Android versi 4.2/4.3 (*Jelly Bean*)
10. Android versi 4.4 (*Kitkat*)
11. Android versi 5.0 (*lollipop*)

2.3.4. Arsitektur Android

Sistem operasi android dibangun berdasarkan kernel linux, dan memiliki arsitektur sebagai berikut (Amiral, 2011) :

1. Applications

Lapisan ini adalah lapisan aplikasi, serangkaian aplikasi akan terdapat pada perangkat *mobile*. Aplikasi ini telah terdapat pada Android termasuk kalender, kontak, SMS, dan lain sebagainya. Aplikasi – aplikasi ini ditulis dengan bahasa pemrograman Java.

2. Application Framework

Pengembangan aplikasi memiliki akses penuh ke Android sama dengan aplikasi ini telah tersedia. Pengembang dapat dengan mudah mengakses informasi lokasi, mengatur alarm, menambah pemberitahuan ke status bar dan lain sebagainya. Arsitektur aplikasi ini dirancang untuk menyederhanakan penggunaan kembali komponen, aplikasi apapun yang dapat mempublikasikan kemampuan dan aplikasi lainnya dapat menggunakan kemampuan mereka sesuai batas keamanan. Dasar dari aplikasi adalah seperangkat layanan sistem, yaitu berbagai view yang digunakan untuk membangun *user interface*, *content provider* yang memungkinkan aplikasi berbagi data *Resource Manager*

menyediakan akses ke kode seperti *graphic*, *string* dan *layout notification manager* yang akan membuat aplikasi dapat menampilkan tanda pada status bar dan *Activity Manager* yang berguna mengatur hidup dari aplikasi.

3. *Libraries*

Satu *set libraries* dalam bahasa C/C++ yang digunakan oleh berbagai komponen pada sistem Android.

4. *Android Runtime*

Satu *set libraries* inti yang menyediakan sebagian besar fungsi yang tersedia di *libraries* inti dari bahasa pemrograman java. Setiap aplikasi akan berjalan sebagai proses sendiri pada Dalvik Virtual Machine.

5. Linux Kernel

Android tergantung pada Linux versi 2.6 untuk layanan sistem inti seperti keamanan, manajemen memori, manajemen proses, *network stack*, dan *model driver*. Kernel juga bertindak sebagai lapisan antara *hardware* dan seluruh *software*.

2.3.5. Fitur Android

Fitur yang tersedia di Android adalah (Amiral, 2011)

1. Framework Aplikasi : memungkinkan penggunaan dan penghapusan komponen yang tersedia.
2. Dalvik Virtual Machine : mesin virtual dioptimalkan untuk perangkat mobile.
3. Grafik : Grafik di 2D dan 3D berdasarkan pustaka *OpenGL*.

4. SQLite : Untuk menyimpan data.
5. Mendukung media : audio, video dan berbagai format gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF).
6. *GSM, Bluetooth, EDGE, 3G dan WiFi (hardware dependent), Camera, Global Positioning System (GPS), Compass dan accelerometer (hardware dependent).*
7. Lingkungan pengembangan yang kaya, termasuk *emulator*, peralatan *debugging*, dan *plugin* untuk Android Studio.

2.4. SISTEM PENDUKUNG KEPUTUSAN

Sistem Pendukung Keputusan (SPK) biasanya dibangun untuk mendukung solusi atas suatu masalah atau untuk suatu peluang. Aplikasi Sistem Pendukung Keputusan (SPK) digunakan dalam pengambilan keputusan. Aplikasi Sistem Pendukung Keputusan (SPK) menggunakan CBIS (*Computer Based Information Systems*) yang fleksibel, interaktif, dan dapat diadaptasi, yang dikembangkan untuk mendukung solusi atas masalah manajemen spesifikasi yang tidak terstruktur.

Menurut Bonczek, dkk., (1980) dalam buku *Decision Support Systems And Intelligent Systems* (Turban, 2005: 137) mendefinisikan sistem pendukung keputusan sebagai sistem berbasis komputer yang terdiri dari tiga komponen yang saling berinteraksi, sistem bahasa (mekanisme untuk memberikan komunikasi antara pengguna dan komponen sistem pendukung keputusan lain), sistem pengetahuan (repository pengetahuan domain masalah yang ada pada sistem pendukung keputusan atau sebagai data atau sebagai produser) dan sistem

pemrosesan masalah (hubungan antara dua komponen lainnya, terdiri dari satu atau lebih kapasitas manipulasi masalah umum yang diperlukan untuk pengambilan keputusan).

2.4.1. Pengambilan Keputusan

Pengambilan keputusan merupakan proses pemilihan *alternative* tindakan untuk mencapai tujuan atau sasaran tertentu. Pengambilan keputusan dilakukan dengan pendekatan sistematis terhadap permasalahan melalui proses pengumpulan data menjadi informasi serta ditambah dengan faktor – faktor yang perlu dipertimbangkan dalam pengambilan keputusan.

Pengambilan Keputusan terdapat beberapa tahapan, dibawah ini tahap-tahapan pengambilan yang harus dilalui dalam proses pengambilan keputusan sebagai berikut :

1. Tahap Pemahaman (*Intelligence Phace*)

Tahap ini merupakan proses penelusuran dan pendeteksian dari lingkup problematika serta proses pengenalan masalah. Data masukan diperoleh, diproses dan diuji dalam rangka mengidentifikasi masalah.

2. Tahap Perancangan (*Design Phace*)

Tahap ini merupakan proses pengembangan dan pencarian alternatif tindakan / solusi yang dapat diambil. Tersebut merupakan representasi kejadian nyata yang disederhanakan, sehingga diperlukan proses validasi dan verifikasi untuk mengetahui keakuratan model dalam meneliti masalah yang ada.

3. Tahap Pemilihan (*Choice Phace*)

Tahap ini dilakukan pemilihan terhadap diantara berbagai alternatif solusi yang dimunculkan pada tahap perencanaan agar ditentukan/ dengan memperhatikan kriteria – kriteria berdasarkan tujuan yang akan dicapai.

4. Tahap Impelementasi (*Implementation Phace*)

Tahap ini dilakukan penerapan terhadap rancangan sistem yang telah dibuat pada tahap perancangan serta pelaksanaan alternatif tindakan yang telah dipilih pada tahap pemilihan.(Simon, 2012)

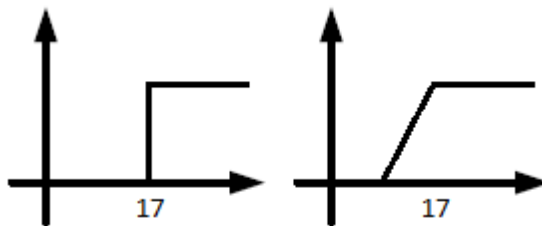
2.5. FUZZY TSUKAMOTO

2.5.1. Pengertian Logika *Fuzzy*

Dalam bahasa inggris, *fuzzy* mempunyai arti kabur atau tidak jelas.Jadi, logika *fuzzy* adalah logika yang kabur, atau mengandung unsur ketidakpastian.

Pada logika biasa, yaitu logika tegas, kita hanya mengenal dua nilai, salah atau benar, 0 atau 1.Sedangkan logika *fuzzy* mengenal nilai antara benar dan salah.Kebenaran dalam logika *fuzzy* dapat dinyatakan dalam derajat kebenaran yang nilainya antara 0 sampai 1.

Misalnya dalam kehidupan sehari-hari, dewasa didefinisikan dengan berusia 17 tahun ke atas. Jikamenggunakan logika tegas, seseorang yang berusia 17 tahun kurang 1 hari akan didefinisikan sebagai tidak dewasa. Namun dalam logika *fuzzy*, orang tersebut dapat dinyatakan dengan hampir dewasa.



Gambar 2.4 Logika tegas (kiri) dan logika *fuzzy* (kanan)

Sistem *fuzzy* merupakan penduga numerik yang terstruktur dan dinamis. Sistem ini mempunyai kemampuan untuk mengembangkan sistem intelijen dalam lingkungan yang tidak pasti. Sistem ini menduga suatu fungsi dengan logika *fuzzy*. Dalam logika *fuzzy* terdapat beberapa proses yaitu penentuan himpunan *fuzzy*, penerapan aturan IF-THEN dan proses inferensi *fuzzy*.

Ada beberapa metode untuk merepresentasikan hasil logika *fuzzy* yaitu metode *Tsukamoto*, *Sugeno* dan *Mamdani*. Pada metode *Tsukamoto*, setiap konsekuen direpresentasikan dengan himpunan *fuzzy* dengan fungsi keanggotaan monoton. Output hasil inferensi masing-masing aturan adalah z , berupa himpunan biasa (*crisp*) yang ditetapkan berdasarkan predikatnya. Hasil akhir diperoleh dengan menggunakan rata-rata terbobotnya. (Afan, 2012).

2.5.2. Sejarah Logika *Fuzzy*

Sistem *fuzzy* pertama kali diperkenalkan oleh Prof. L.A. Zadeh, seorang peneliti dari Universitas California dari Berkeley pada tahun 1965. Logika *fuzzy* dikembangkan dari teori himpunan *fuzzy*.

2.5.3. Himpunan *Fuzzy*

Himpunan *fuzzy* merupakan suatu group yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *fuzzy*. Pada himpunan tegas (*crisp*), nilai keanggotaan suatu item x dalam suatu himpunan A , yang sering ditulis dengan $f_A[x]$, memiliki dua kemungkinan, yaitu : Satu (1), yang berarti bahwa suatu item menjadi anggota dalam suatu himpunan atau Nol (0), yang berarti bahwa suatu item tidak menjadi anggota dalam suatu himpunan.

Pada himpunan *fuzzy* nilai keanggotaan terletak pada rentang 0 sampai 1. Apabila x memiliki nilai keanggotaan *fuzzy* $f_A[x] = 0$ berarti x tidak menjadi anggota himpunan A , demikian pula apabila x memiliki nilai keanggotaan *fuzzy* $f_A[x] = 1$ berarti x menjadi anggota penuh pada himpunan A .

Kemiripan antara keanggotaan *fuzzy* dengan probabilitas terkadang menimbulkan kerancuan, karena memiliki nilai pada interval $[0,1]$, namun interpretasi nilainya sangat berbeda. Keanggotaan *fuzzy* memberikan suatu ukuran terhadap pendapat atau keputusan, sedangkan probabilitas mengindikasikan proporsi terhadap keseringan suatu hasil bernilai benar dalam jangka panjang.

2.5.4. Kelebihan dan Kekurangan Logika *Fuzzy*

Logika *fuzzy* memiliki beberapa keunggulan, antara lain sebagai berikut.

1. Konsep logika *fuzzy* mudah dimengerti. Konsep matematis yang mendasari penalaran logika *fuzzy* sangat sederhana dan mudah dimengerti.
2. Logika *fuzzy* sangat fleksibel.
3. Logika *fuzzy* memiliki toleransi terhadap data-data yang tidak tepat.
4. Logika *fuzzy* mampu memodelkan fungsi nonlinear yang kompleks.
5. Logika *fuzzy* dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.
6. Logika *fuzzy* dapat bekerja sama dengan teknik-teknik kendali secara konvensional.
7. Logika *fuzzy* didasarkan pada bahas alami.

Sementara itu, dalam pengaplikasiannya, logika *fuzzy* juga memiliki beberapa kelebihan, antara lain sebagai berikut.

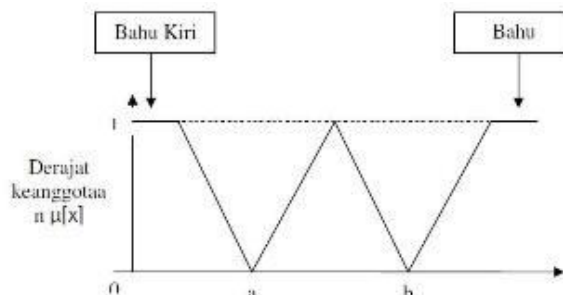
1. Daya gunanya dianggap lebih baik daripada teknik kendali yang pernah ada.
2. Pengendali *fuzzy* terkenal karena keandalannya.
3. Mudah diperbaiki.
4. Pengendali *fuzzy* memberikan pengendalian yang sangat baik dibandingkan teknik lain
5. Usaha dan dana yang dibutuhkan kecil.

Selain itu, logika *fuzzy* juga memiliki kekurangan, terutama dalam penerapannya. Kekurangan-kekurangan tersebut antara lain:

1. Para enjiner dan ilmuwan generasi sebelumnya dan sekarang banyak yang tidak mengenal teori kendali *fuzzy*, meskipun secara teknik praktis mereka memiliki pengalaman untuk menggunakan teknologi dan perkakas kontrol yang sudah ada.
2. Belum banyak terdapat kursus/balai pendidikan dan buku-buku teks yang menjangkau setiap tingkat pendidikan (*undergraduate*, *postgraduate*, dan *on site training*)
3. Hingga kini belum ada pengetahuan sistematis yang baku dan seragam tentang metodologi pemecahan problema kendali menggunakan pengendali *fuzzy*.
4. Belum adanya metode umum untuk mengembangkan dan implementasi pengendali *fuzzy*.

2.5.5. Fungsi Keanggotaan

Fungsi keanggotaan adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaan yang memiliki nilai interval antara 0 dan 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Salah satu representasi fungsi keanggotaan dalam *fuzzy* yang akan dipakai adalah representasi kurva bentuk bahu. Kurva yang bentuknya seperti bahu di sisi paling kanan dan paling kirinya. Himpunan *fuzzy* 'bahu', bukan segitiga, digunakan untuk mengakhiri variabel suatu daerah *fuzzy*.



Gambar 2.5 Representasi Kurva Bentuk Bahu

Ada dua keadaan himpunan *fuzzy* yang linear. Pertama, kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol $[0]$ bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi. Kedua, merupakan kebalikan yang pertama. Garis lurus dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih rendah.

2.5.6. Metode Tsukamoto

Dalam membangun sebuah sistem *fuzzy* dikenal beberapa metode penalaran, antara lain: metode Tsukamoto, metode Mamdani dan metode Sugeno. Pada metode Tsukamoto, setiap konsekuensi pada aturan yang berbentuk IF-THEN harus direpresentasikan dengan suatu himpunan *fuzzy* dengan fungsi keanggotaan yang monoton. Sebagai hasilnya, output hasil inferensi dari tiap-tiap aturan diberikan dengan tegas (*crisp*) berdasarkan α -predikat (*fire strength*). Hasil akhirnya diperoleh dengan menggunakan rata-rata terbobot. Misal ada 2 variabel input, var-1(x) dan var-2(y) serta 1 variabel output var-3(z), dimana var-1 terbagi atas 2 himpunan yaitu A1 dan A2 dan var-2 terbagi atas himpunan B1 dan B2.

Sedangkan var-3 juga terbagi atas 2 himpunan yaitu C1 dan C2.

(Kusumadewi dan Purnomo, 2010)

Ada dua aturan yang digunakan yaitu:

[R1] IF (x is A1) and (y is B2) THEN (z is C1)

[R2] IF (x is A2) and (y is B1) THEN (z is C2)

Keterangan:

R1 : Aturan *fuzzy*

x : Variabel pinjaman

α_1 : Himpunan pinjaman tertinggi

α_2 : Himpunan pinjaman terendah

And : Operator yang digunakan

y : Variabel jaminan

B1 : Himpunan jaminan tertinggi

B2 : Himpunan jaminan terendah

THEN : Operator yang digunakan

Z : Variabel Penghasilan (nilai crisp)

C1 : Himpunan penghasilan tertinggi (harus monoton)

C2 : Himpunan penghasilan terendah (harus monoton)

2.6. JAVA

Java menurut definisi dari Sun adalah nama sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer *standalone* ataupun pada lingkungan jaringan. Java adalah generasi kedua dari *java platform*, java berdiri diatas sebuah mesin interpreter yang diberi nama *java virtual machine*

(JVM). JVM inilah yang akan membaca *bytecode* dalam *file.class* dari suatu program sebagai representasi langsung program yang berisi bahasa mesin. Oleh karena itu, bahasa java disebut sebagai bahasa pemrograman yang *portable* karena dapat dijalankan pada berbagai sistem operasi, asalkan pada sistem operasi terdapat JVM (Shalahudin, 2010).

2.7. ANDROID STUDIO

Android studio merupakan sebuah integrated Development Environment (IDE) untuk platform android. Android studio ini diumumkan pada tanggal 16 mei 2013 pada konferensi Google I/O oleh produk manajer Google, Ellie Powers. Android studio bersifat free di bawah Apache License 2.0. Android studio awalnya dimulai dengan versi 1.0 pada bulan mei 2013, kemudian dibuat versi Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak yang dapat dijalankan di semua platform (*platform independent*). Berikut ini adalah sifat dari Eclipse :

1. *Multi Platform* : Target sistem operasi Eclipse adalah *Microsoft Windows, Linux, Solaris, AIX, HP-UX* dan *Mac OS X*.
2. *Multi Language* : Eclipse dikembangkan dengan bahasa pemrograman *java*, akan tetapi eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti *C/C++, Cobol, Python, Perl, PHP* dan sebagainya.
3. *Multi Role* : Selain sebagai IDE untuk pengembangan aplikasi, eclipse pun bisa digunakan untuk aktivitas dalam pengembangan

perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web dan lain sebagainya (Safaat, 2011).

2.8. ANDROID SDK EMULATOR

Android SDK merupakan *tools* bagi para *programmer* yang ingin mengembangkan aplikasi berbasis google android. Android SDK mencakup seperangkat alat pengembangan yang komprehensif. Android SDK terdiri dari *debugger*, *libraries*, *handset emulator*, dokumentasi, contoh kode dan tutorial. Saat ini android sudah mendukung arsitektur x86 pada Linux (distribusi linux apapun untuk desktop modern). *Mac OS X* 10.4.8 atau lebih, *Windows XP* atau *Vista*.

Persyaratan mencakup JDK, *Apache Ant* dan *Python 2.2* atau yang lebih baru. IDE yang didukung secara resmi adalah *Eclipse 3.2* atau lebih dengan menggunakan *plugin Android Development Tools (ADT)*. Dengan ini pengembang dapat menggunakan *text editor* untuk mengedit file *Java* dan *XML*, serta menggunakan peralatan *command line* untuk menciptakan, membangun, melakukan *debug* aplikasi Android (misalnya *reboot*, menginstall paket perangkat lunak dengan jarak jauh) (Haidi, 2013).

2.9. SQLite

SQLite adalah sebuah *embedded database* yang sangat terkenal karena menggabungkan antarmuka SQL dengan memori yang sangat kecil dan memori yang sangat baik. (Murphy, 2010).

SQL adalah sebuah *open source database* yang sudah ada cukup lama, cukup stabil, dan sangat terkenal pada perangkat kecil, termasuk android (Gargenta, 2011)

Untuk android, SQLite dijadikan satu di dalam Android *runtime*, sehingga setiap aplikasi android dapat membuat basis data SQLite. Karena menggunakan antarmuka SQL, cukup mudah digunakan untuk orang-orang dengan pengalaman lain yang berbasis database SQL (Murphy, 2010).

Terdapat beberapa alasan mengapa SQLite sangat cocok untuk pengembangan aplikasi Android (Murphy, 2010).

1. *Database* dengan konfigurasi nol. Artinya tidak ada konfigurasi *database* untuk para *developer*, ini membuatnya relatif mudah digunakan.
2. Tidak memiliki *server*. Tidak ada proses database SQLite yang berjalan. Pada dasarnya satu set *libraries* menyediakan fungsionalitas *database*.
3. *Single file database*. Ini membuat keamanan *database* secara langsung.
4. *Opensource*. Hal ini membuat developer mudah dalam pengembangan aplikasi.

2.10. UNIFIED MODELLING LANGUAGE (UML)

Unified Modelling Language(UML) adalah standar bahasa yang digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisa dan design, serta menggambarkan arsitektur dalam pemograman berorientasi objek (Shalahudin, 2011).

2.10.1. Use Case Diagram

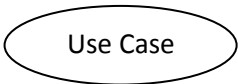
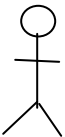
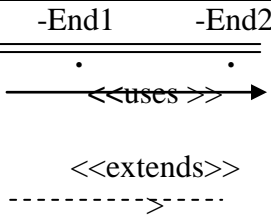
Use case atau diagram use case merupakan pemodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *Use case* digunakan untuk mengetahui fungsi apa saja yang ada dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case*, yaitu pendefinisian apa yang disebut aktor dan *use case*.

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.

Use case merupakan fungsionalitas yang disediakan sistem sebagai unit – unit yang saling bertukar pesan antar unit atau aktor.

Tabel 2.1 Simbol *use case* diagram

No.	Simbol	Nama	Deskripsi
1.		<i>Use case</i>	Menggambarkan proses / kegiatan yang dapat dilakukan oleh aktor
2.		Aktor/ <i>actor</i>	Menggambarkan entitas / subyek yang dapat melakukan suatu proses
3.		<i>Relation</i>	Relasi antara case dengan actor ataupun case lain

2.10.2. Class Diagram

Diagram kelas atau *class* diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut :

1. Kelas main

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan

2. Kelas yang menangani tampilan sistem

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai

3. Kelas yang diambil dari pendefinisian use case

Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*

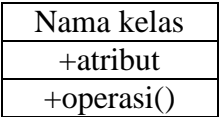
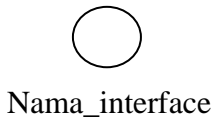
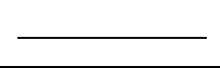
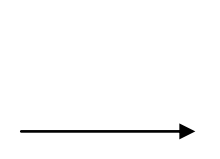
4. Kelas yang diambil dari pendefinisian data

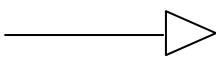
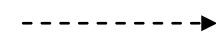
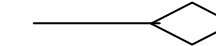
Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Jenis – jenis kelas di atas juga dapat digabungkan satu sama lain sesuai dengan pertimbangan yang dianggap baik asalkan fungsi-fungsi yang sebaiknya ada pada struktur kelas tetap ada. Susunan kelas juga dapat ditambahkan kelas utilitas seperti koneksi ke basis data, membaca *file* teks, dan lain sebagainya sesuai kebutuhan.

Dalam mendefinisikan metode yang ada di dalam kelas perlu memperhatikan apa yang disebut dengan *cohesion* dan *coupling*. *Cohesion* adalah ukuran seberapa dekat dengan keterkaitan instruksi di dalam sebuah metode terkait satu sama lain sedangkan *coupling* adalah ukuran seberapa dekat keterkaitan instruksi antara metode yang satu dengan metode yang lain dalam sebuah kelas. Sebagai aturan secara umum maka sebuah metode yang dibuat harus memiliki kadar *cohesion* yang kuat dan kadar *coupling* yang lemah.

Tabel 2.2 Simbol *class* diagram

No.	Simbol	Nama	Deskripsi
1.		Kelas / <i>Class</i>	Kelas pada struktur sistem
2.		Antarmuka / <i>Interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3.		Asosiasi / <i>association</i>	Asosiasi biasanya juga disertai dengan <i>multiplicity</i>
4.		Asosiasi berarah / <i>directed association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, biasanya juga disertai dengan <i>multiplicity</i>

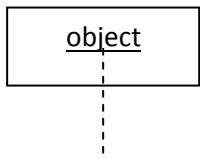
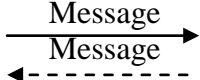
No.	Simbol	Nama	Deskripsi
5.		Generalisasi	Relasi antar kelas dengan makna generalisasi spesialisasi (Umum Khusus)
6.		Kebergantungan / <i>dependency</i>	Relasi antar dengan makna kebergantungan antar kelas
7.		Agregasi / <i>aggregation</i>	Relasi antar kelas dengan makna semua bagian ()

2.10.3. Sequence Diagram

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlihat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstiasi menjadi objek itu.

Banyaknya diagram sekuen yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram yang sekuen yang harus dibuat juga semakin banyak.


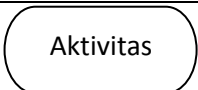
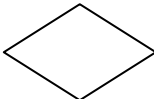


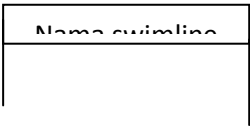
Tabel 2.3 Simbol *sequence* diagram

No.	Simbol	Nama	Deskripsi
1.		Object	Menggambarkan pos-pos obyek yang pengirim dan menerima <i>message</i>
2.		<i>Message</i>	Menggambarkan aliran pesan yang dikirim oleh pos-pos obyek

2.10.4. Activity Diagram

Diagram aktivitas atau activity diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Tabel 2.4 Simbol *activity* diagram

No.	Simbol	Nama	Deskripsi
1.		Status awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
2.		Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
3.		Percabangan / <i>decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
4.		Penggabungan / <i>join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
5.		Status akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
6.		Swimline	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

2.11. COREL DRAW X4

Corel Draw adalah sebuah program komputer yang melakukan editing pada garis vector. Program ini dibuat oleh Corel, sebuah perusahaan software yang berkantor pusat di Ottawa, Kanada. Corel Draw memiliki kegunaan untuk mengolah gambar, oleh karena itu banyak digunakan pada pekerjaan dalam bidang publikasi atau percetakan ataupun pekerjaan di bidang lain yang membutuhkan proses visualisasi (Nurhayati, 2012).

Suatu program yang baik tentu saja memiliki keunggulan yang signifikan. Beberapa keunggulan program Corel Draw antara lain adalah (Nurhayati, 2012) :

1. Gambar yang dihasilkan dengan vector atau berbasis vector bisa ditekan pada tingkat yang paling rendah namun hasilnya tidak kalah dengan gambar yang berbasis bitmap atau raster.
2. Penggunaan Corel Draw, terutama pada tool-tool yang ada di dalamnya sangat mudah dipahami oleh penggunanya, bahkan oleh orang yang baru pertama menggunakannya.
3. Corel Draw sangat baik untuk mengkolaborasikan antara tulisan dengan gambar, seperti layaknya *Adobe Photoshop*.
4. Banyaknya pengguna Corel Draw, membuat adanya komunitas dengan jumlah anggota yang besar. Hal ini akan membuat kita tidak kesulitan jika ingin mempelajari Corel Draw lebih mendalam karena banyak ditemukan tutorial, tips dan trik yang diproduksi oleh pengguna lain.

2.12. KAJIAN PUSTAKA

Kajian pustaka berisi tentang penelitian terdahulu yang pernah dilakukan oleh peneliti lain sesuai dengan judul penelitian ini. Kajian penelitian disajikan dalam bentuk tabel, sebagai berikut:

Tabel 2.5 Penelitian yang Relevan

Peneliti	Judul	Isi	Manfaat
Merry Christinne Steviani Adistiya, Januar Wibowo dan Julianto Lemantara (2012)	Sistem Pakar Penentuan Minat dan Bakat Anak Umur 5-10 Tahun	Implementasi sistem yang dibuat digunakan untuk mengetahui bakat dan minat anak-anak usia 5-10 tahun. Penelitian ini menggunakan metode Fuzzy Tsukamoto dan menerapkan PHP sebagai bahasa pemrograman dan MySQL sebagai penyimpan basis data	Membantu para orang tua mengetahui akan bakat dan minat anak-anak mereka tanpa perlu memaksakan kemauan orang tua dan mengetahui tentang bakat dan minat anak – anak mereka, tanpa mengeluarkan dana
Fiby Nur Afiana (2014)	Perancangan Test Kecerdasan Minat dan Bakat Anak Menggunakan Metode Fuzzy Logic	Penentuan bakat dan minat pada anak PAUD menggunakan metode Fuzzy Logic khususnya Fuzzy Tsukamoto dan menerapkan PHP sebagai bahasa pemrograman dan MySQL sebagai penyimpan basis data. Penelitian ini hanya menghasilkan 2 kecerdasan yaitu kecerdasan linguistic dan visual-spasial.	Membantu orang tua untuk mengetahui bagaimana minat dan bakat anak sejak dini untuk mempersiapkan penerus bangsa yang berkualitas tanpa bantuan tenaga ahli dalam hal ini psikolog anak yang jumlahnya masih sangat minim dan mahal biaya konsultasi