

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1. Tinjauan Pustaka

Untuk membantu proses penelitian karya ini, diperlukan proses kerangka pemikiran yang didasari dari referensi studi pustaka penelitian dan pembuatan karya terdahulu yang memiliki hubungan dengan implementasi MBTI dan metodologi *Kanban*.

Salah satunya, memiliki judul “*Persona Traits Identification based on Myers-Briggs Type Indicator(MBTI) - A Text Classification Approach*” yang disusun oleh Srilakshmi Bharadwaj, Srinidhi Sridhar, Rahul Choudhary, dan Ramamoorthy Srinath. Penelitian jurnal ini memiliki tujuan mengidentifikasi jenis kepribadian dan kondisi Kesehatan mental dari analisis kumpulan kalimat, seperti esai, untuk membantu inspirasi dari seorang penulis, membantu proses penerimaan pekerjaan, dan membantu memberi rekomendasi sebuah produk yang disesuaikan dengan hasil. Metodologi yang dipakai untuk keberlangsungan penelitian jurnal ini yaitu, *machine learning* dengan model algoritma Naïve Bayes dan SVM, serta *deep learning* dengan model algoritma *Neural Network*. Hasil yang didapatkan dari penelitian jurnal ini, model Naïve Bayes dapat memiliki hasil terakurat dengan *feature vectors* LIWC dan Emolex, model SVM dengan hasil akurasi tertinggi sebesar 88% dari model lainnya dengan menggunakan *feature vectors* TF-IDF, Emolex, ConceptNet, dan LIWC, terakhir model *Neural Network* dapat memiliki hasil terakurat dengan *feature vectors* SVD dan LIWC [13].

Jurnal referensi literatur kedua memiliki judul “Implementasi Website Sebagai Media Informasi Dan Promosi Pada Pondok Pesantren Putra-Putri Ad-dainuriyah 2 Semarang”. Jurnal ini disusun oleh Andik Prakasa Hadi dan Faiz

Abdul Rokhman. Penelitian jurnal ini memiliki tujuan menyebarkan informasi mengenai tempat pondok pesantren putra-putri addainuriyah 2 semarang lewat *platform web*. Metodologi yang digunakan pada jurnal ini merupakan *research-based method* yang bersifat *sustainability Reseacrh & Development*. Hasil dari penelitian ini, menurut uji coba lapangan dari dua tempat, menerima skor $\pm 80\%$ jawaban sangat baik untuk semua kriteria *web* [7].

Jurnal referensi ketiga memiliki judul “*Extrovert and Introvert Classification based on Myers-Briggs Type Indicator (MBTI) using Support Vector Machine (SVM)*”. Jurnal ini disusun oleh Muhammad Nurfauzi Sahono, Fiqie Ulya Sidi-astahta, Guruh Fajar Shidik, Ahmad Zainul Fanani, Muljono, Safira Nuraisha, dan Erba Lutfina. Penelitian jurnal ini memiliki tujuan untuk mengklasifikasi kepribadian dan karakteristik keaktifan antara *ekstrovert* atau *introvert* seseorang dari sebuah *post* media sosial orang tersebut. Jurnal ini menggunakan metodologi *machine learning* dengan algoritma SVM. Hasil yang didapatkan dari penelitian ini setelah model SVM menggunakan *feature vector* TF-IDF, sebuah *post* media sosial dapat ditentukan sifat *introvert* dan *ekstrovert* dari *user* pengunggah *post* dengan keakuratan $\pm 74\%$ dan keakuratan menjadi lebih tinggi jika *post* memiliki *ematicon* dan *mention* akun lain [14].

Jurnal referensi literatur keempat memiliki judul “Rancang Bangun *Website* Persona (*Personality Analyst*) Menggunakan *Myers-Briggs Type Indicator* Berbasis *Framework* Laravel”. Jurnal ini disusun oleh Ega Wahyu Cahyono dan Andi Iwan Nurhidayat. Tujuan dari jurnal ini untuk mendokumentasi pembuatan karya *website* yang dapat membantu seseorang, khususnya seorang *user* melamar pekerjaan, untuk mengetahui kepribadian dan pekerjaan yang cocok. Jurnal ini menggunakan metodologi teori MBTI untuk tes psikologi dan menggunakan metode pengembangan *Waterfall*. Jurnal ini menghasilkan aplikasi *web* pelamaran pekerjaan dengan dua jenis *user*, yaitu *user* biasa yang cocok untuk pelamar pekerjaan, dan *user admin* untuk penerima pelamar untuk sebuah perusahaan [15].

Jurnal referensi kelima memiliki judul “Metode *Agile* Dalam Perancangan Sistem Prediksi Prevalensi *Stunting* di Indonesia”. Jurnal ini disusun oleh Danang Arbian Sulisty, Yogie Susdyastama Putra, dan Suastika Yulia Riska. Tujuan dari jurnal ini untuk mengimplementasi *Agile* untuk pembuatan sistem yang dapat mendeteksi penyakit *Stunting* pada balita. Jurnal ini menggunakan metodologi pengembangan *Agile*. Penelitian jurnal ini menghasilkan aplikasi *web* untuk membantu *user* memprediksi penyebaran *stunting* dan dapat membuat akun untuk mengetahui dan menyimpan hasil prediksi yang dilihat dari daerah provinsi dengan kemudahan perubahan kebutuhan sistem dan kesederhanaan sistem [16].

Jurnal referensi literatur keenam memiliki judul “Implementasi *Cloud Computing Technology* dan Dampaknya Terhadap Kelangsungan Bisnis Perusahaan Dengan Menggunakan Metode *Agile* dan Studi Literatur”. Jurnal ini disusun oleh Eri Riana. Tujuan dari jurnal ini untuk meneliti implementasi teknologi *cloud* dengan layanan *Google Cloud Platform* terhadap sistem sebuah perusahaan. Jurnal ini menggunakan metodologi pengembangan *Agile* terhadap penelitian dan studi literatur mengenai teknologi *cloud*. Penelitian jurnal ini menghasilkan sebuah infrastruktur sistem manajemen *cloud* untuk membantu perusahaan kecil dalam perkembangan usaha dan penghematan biaya dengan implementasi dengan kelebihan kapasitas ruang data yang fleksibel dan *multi-platform* [17].

Jurnal referensi literatur ketujuh memiliki judul “Pengembangan Aplikasi *E-Catalog Augmented Reality* Sebagai Media Pemasaran Property”. Jurnal ini disusun oleh Muhammad Rizal dan Muhammad Rusmin. Tujuan dari jurnal ini untuk mengimplementasi teknologi *Augmented Reality* pada aplikasi pemasaran properti berbasis *android*. Jurnal ini menggunakan metode pengembangan *Agile*. Penelitian jurnal ini dapat menghasilkan aplikasi *Android* dan *IOS e-catalog* media promosi penjualan properti bersifat *augmented reality* dengan skor penilaian kelayakan aplikasi sebesar $\pm 86\%$ [18].

Jurnal referensi literatur kedelapan memiliki judul “Sistem Informasi *Multi-platform* Repositori Data Dosen Berbasis *React* Menggunakan *Framework Kanban*”. Jurnal ini disusun oleh Habibie Ed Dien, Putra Prima Arhandi, Aditya Eka Pradana P., dan Deby Silvia A. Tujuan dari jurnal ini untuk mengembangkan aplikasi *mobile* dan *website* yang dapat mengelola data-data dokumen yang diperlukan seorang dosen, seperti biodata, kepangkatan, penelitian, sertifikasi, dll. Jurnal ini menggunakan metode pengembangan *Agile* dengan *Framework Kanban*. Penelitian jurnal ini dapat menghasilkan aplikasi sistem repositori Beban Kerja Dosen untuk keberlangsungan Tri Dharma Perguruan Tinggi bersifat *multi-platform* menggunakan bahasa pemrograman *Javascript*, *React JS* untuk *platform web* dan *React Native* untuk *platform mobile*, dan *web service API* untuk saling menghubungkan *platform* aplikasi [19].

Jurnal referensi literatur kesembilan memiliki judul “Penilaian Bahasa Isyarat Ba-haru untuk Pengajaran dan Pembelajaran Solat bagi Orang Pekak”. Jurnal ini disusun oleh Siti Ramna Khamaruddin, Nurul Asiah Fasehah Muhammad, Norakyai-ree Mohd Raus, Adnan Mohamed Yusoff, Mahyuddin Hashim, Noornajihan Ja’afar, dan Mohd Nur Adzam Rasdi. Tujuan dari penelitian ini untuk menilai 63 gerakan isyarat untuk pelaksanaan ibadah sholat apakah valid untuk digunakan oleh seorang dengan disabilitas tunarungu. Jurnal ini menggunakan penilaian *Cohen Kappa*. Penelitian jurnal ini memberi hasil penilaian gerakan – gerakan isyarat dengan nilai 0.87, sehingga menurut proporsi penilaian validitas *Cohen Kappa*, nilai dinyatakan Sangat Baik untuk pelaksanaan ibadah sholat bagi seseorang yang memiliki disabilitas tunarungu [20].

Jurnal referensi literatur kesepuluh memiliki judul “Uji Validitas Aplikasi *SI-Book* Menggunakan *Spss* Dengan *Kom-Binasi* Metode *R-Tabel* Dan *Cohen’s Kappa*”. Jurnal ini disusun oleh Rizky Parluka, Putra Dwi Wira Gardha Yuni ahans, Rayhan Saneval Arhinza, Vito Fausta Majid, dan Muhammad Ghifari Alfian. Tujuan dari penelitian ini untuk menilai kualitas dan informasi yang diberikan oleh aplikasi *SI-Book*. Jurnal ini menggunakan *SPSS* dengan metode *R-Table*

dan penilaian *Cohen Kappa*. Hasil dari penelitian menilai kualitas dan validitas dari aplikasi *SI-Book*. Penilaian hasil validasi dengan metode *R-Table* menunjukkan hasil dalam persen-tase yaitu 100% valid dari soal no 1 hingga 10, serta hasil validasi dengan metode *Cohen's Kappa* menunjukkan hasil 0,793 yang berarti bahwa aplikasi *Si-Book* dapat dikategorikan kuat, dengan rincian 7 kali *error* dan 23 kali berhasil [21].

Berdasarkan semua referensi penelitian terdahulu yang diambil sebagai referensi, kesimpulan yang dapat ditarik merupakan pembuatan karya aplikasi *web* dengan menggunakan metode *Agile* dengan *framework Kanban* dan menggunakan bahasa pemrograman *Javascript* dengan *framework Express JS* dan *React JS* belum pernah dilakukan. Menurut referensi di atas, jurnal penelitian ke-1, ke-3, dan ke-4 diambil aspek implementasi tes MBTI ke dalam *web*, walaupun metodologi berdeda. Lalu, jurnal penelitian ke-2, ke-5, dan sisanya diambil aspek metodologi yang digunakan untuk proses pengembangan, walaupun studi kasus masing-masing berbeda.

Tabel 2.1. Tinjauan Pustaka

Judul	Penulis	Metode Penelitian	Hasil
<i>Persona Traits Identification based on Myers-Briggs Type Indicator(MBTI) - A Text Classification Approach</i> [13]	Bharadwaj, Srilakshmi; Sridhar, Srinidhi; Choudhary, Rahul; Srinath, Ramamoorthy.	Naïve Bayes, SVM, dan Neural Network	Dengan <i>librabry</i> dari LICW dan TF-IDF yang digunakan sebagai <i>vector</i> , Algoritma SVM dapat mencapai hasil tertinggi dengan kategori E/I 85.0%, S/N 88.2%, T/F 86.1%, dan J/P 79.0%. Walaupun dengan <i>pre-processing</i> SVD, kategori I/E dan S/N tercapai tertinggi dengan Neural Network. Tetapi, dengan tambahan <i>vector ConceptNet</i> dan <i>em-SenticNet</i> , SVM kembali menjadi tertinggi.
Implementasi <i>Website</i> Sebagai Media Informasi Dan Promosi Pada Pondok Pesantren Putra-Putri Addainuriyah 2 Semarang [7]	Prakasa Hadi, Andik; Abdul Rokhman, Faiz	<i>Research-Based Method</i>	Aplikasi <i>web</i> dapat menampilkan informasi lengkap mengenai pondok pesantren putra-putri Addainuriyah 2 Semarang dengan hasil skor total uji coba $\geq 80\%$ setuju dalam kualitas baik dari kriteria deain antar muka, kemudahan penggunaan, dan kualitas

Judul	Penulis	Metode Penelitian	Hasil
			informasi yang didapatkan dari uji coba lapangan sebanyak 2 kali.
<i>Extrovert and Introvert Classification based on Myers-Briggs Type Indicator(MBTI) using Support Vector Machine (SVM) [14]</i>	Sahono, Muhammad Nurfauzi; Sidiastahta, Fiqie Ulya; Shidik, Guruh Fajar; Fanani, Ahmad Zainul; Muljono; Nuraisha, Safira; Lutfina, Erba.	SVM	Dengan metode TF IDF, akurasi hasil terhitung sekitar 73,9%. Tetapi dengan <i>Non-Word Feature Extraction</i> , seperti emoticon, akurasi hasil terbaru meningkat 84.0%.

Judul	Penulis	Metode Penelitian	Hasil
Rancang Bangun <i>Website Persona (Personality Analyst)</i> Menggunakan <i>Myers-Briggs Type Indicator</i> Berbasis <i>Framework Laravel</i> [15]	Cahyono, Ega Wahyu; Nurhidayat, Andi Iwan.	<i>Waterfall</i>	Hasil situs memiliki 2 jenis <i>user</i> yang cocok untuk pelamar pekerja dan karyawan yang mengurus pelamaran. Seorang pelamar pekerja dapat melakukan tes yang sudah disediakan dari awal dan atau melakukan tes yang disediakan oleh suatu perusahaan. <i>User</i> dari pihak perusahaan mendapatkan menu yang berbeda untuk dapat mengurus hasil-hasil tes, profil perusahaan, dll.
Metode <i>Agile</i> Dalam Perancangan Sistem Prediksi Prevalensi <i>Stunting</i> Di Indonesia [16]	Sulistyo; Arbian, Danang; Putra, Yogie Susdyastama; Riska, Suastika Yulia.	<i>Agile</i>	Hasil situs dapat membantu <i>user</i> untuk memprediksi penyebaran <i>stunting</i> . <i>User</i> dapat membuat akun untuk mengetahui dan menyimpan hasil prediksi yang dilihat dari daerah provinsi.
Implementasi <i>Cloud Computing Technology</i> dan Dampaknya Terhadap Kelangsungan Bisnis Perusahaan	Riana, Eri.	<i>Agile</i>	Hasil dari penelitian, untuk mengadopsi teknologi <i>cloud</i> bagi perusahaan memerlukan melewati 9 tahap siklus hidup

Judul	Penulis	Metode Penelitian	Hasil
Dengan Menggunakan Metode <i>Agile</i> dan Studi Literatur [17]			adopsi <i>cloud</i> . Dimana ke-9 tahap tersebut dibagi menjadi 2 bagian, yaitu tahapan utama dan tahapan pendukung. Keuntungan yang bisa didapatkan dari pengadopsian teknologi <i>cloud</i> yaitu, pengurangan biaya infrastruktur, fleksibilitas pengaksesan data, sistem penyimpanan data-data otomatis, dll.
Pengembangan Aplikasi <i>E-Catalog Augmented Reality</i> Sebagai Media Pemasaran Property [18]	Muhammad Rizal; Muhammad Rusmin.	<i>Agile</i>	Hasil dari pembuatan karya aplikasi ARTalog, dapat menampilkan model 3d secara <i>augmented reality</i> di atas permukaan yang memiliki gambar yang sesuai dengan ketentuan. Dari pengujian aplikasi menggunakan 4 <i>smartphone</i> , semuanya dapat merespon gambar tertentu dan menampilkan model di atasnya. Dari hasil pengujian terhadap 10 responden, total rata-rata menyatakan semua aspek aplikasi

Judul	Penulis	Metode Penelitian	Hasil
			mendapatkan respon baik dari para responden.
Sistem Informasi <i>Multiplatform</i> Repositori Data Dosen Berbasis <i>React</i> Menggunakan <i>Framework Kanban</i> [19]	Ed Dien, Habibie; Prima Arhandi, Putra; Eka Pradana P., Aditya; Silvia A., Deby	<i>Kanban</i>	Sistem informasi memiliki dua <i>role user</i> . Role pertama merupakan <i>user</i> Dosen, sedangkan <i>role</i> kedua merupakan <i>admin</i> . <i>User</i> Dosen dapat membuat akun dengan mendaftarkan di aplikasi atau bisa didaftarkan oleh <i>admin</i> . Lalu, <i>user</i> Dosen baru dapat menambahkan informasi akun, seperti biodata, dan menambahkan dokumen, seperti sertifikat. Seorang <i>admin</i> dapat menambahkan

Judul	Penulis	Metode Penelitian	Hasil
Penilaian Bahasa Isyarat Baru untuk Pengajaran dan Pembelajaran Solat bagi Orang Pekak [20]	Khamaruddin, Siti Ramna; Muhamad, Nurul Asiah Fasehah; Mohd Raus, Norakyai-ree; Mohamed Yusoff, Adnan; Hashim, Mahyuddin; Ja'afar, Noornajihan; Rasdi, Mohd Nur Adzam;	<i>Cohen Kappa</i>	Hasil penelitian menganalisis validitas 63 gerakan bahasa isyarat untuk pelaksanaan ibadah sholat bagi orang dengan disabilitas tunarungu. Penelitian memiliki 11 penilai ahli, dengan 5 penilai seorang ahli agama dan 6 penilai seorang ahli bidang bahasa isyarat. Hasil penilaian 63 gerakan isyarat dari validitas para ahli berupa 0.87. Berdasarkan proporsi penilain validitas, hasil dinyatakan Sangat Baik untuk pelaksanaan ibadah sholat bagi seseorang yang memiliki disabilitas tunarungu.

Judul	Penulis	Metode Penelitian	Hasil
Uji Validitas Aplikasi Si-Book Menggunakan SPSS dengan kombinasi metode <i>R-Table</i> dan <i>Cohen's Kappa</i> . [21]	Parlika, Rizky; Yuniahans, Putra Dwi Wira Gardha; Arhinza, Rayhan Saneval; Majid, Vito Fausta; Alfian, Muhammad Ghifari;	SPSS, <i>R-Table</i> , <i>Cohen Kappa</i>	Hasil dari penelitian menilai kualitas dan validitas dari aplikasi SI-Book. Penilaian hasil validasi dengan metode <i>R-Table</i> menunjukkan hasil dalam persentase yaitu 100% valid dari soal no 1 hingga 10, serta hasil validasi dengan metode <i>Cohen's Kappa</i> menunjukkan hasil 0,793 yang berarti bahwa aplikasi Si-Book dapat dikategorikan kuat, dengan rincian 7 kali error dan 23 kali berhasil.

2.2. Landasan Teori

2.2.1. Aplikasi *Web*

Aplikasi *web* memiliki dua jenis, yaitu *web* statis dan *web* dinamis. *Web* statis masih menggunakan *tech stack* minimal, seperti HTML, sehingga setiap perkembangan dan pengaturan informasi memerlukan perubahan program. *Web* dinamis mulai menggunakan bahasa bersifat *server-side*, seperti PHP, sehingga setiap perubahan informasi dapat dilakukan dari dalam *web* dan tidak perlu memerlukan perubahan program dikarenakan informasi yang ditampilkan terhubung dengan *database* [22].

2.2.2. *Node JS*

Node JS adalah lingkungan pengembangan bersifat *open-source* serta *multi-platform* untuk mengembangkan aplikasi *web server-side* menggunakan *JavaScript*. Dengan arsitektur *event-driven* dan API I/O *non-blocking*-nya, dapat mengoptimalkan sebuah aplikasi *web* menjadi *real-time* yang memerlukan tingkat *throughput* dan skalabilitas tinggi. Berbeda dengan kerangka kerja pengembangan *web* sebelumnya yang didasarkan pada model *stateless*, yaitu informasi sesi tidak disimpan antar permintaan pengguna, *Node JS* memungkinkan komunikasi *bidirectional real-time* antara *client* dan *server*, sehingga mereka dapat bebas bertukar data [23].

2.2.3. *React JS*

React JS merupakan sebuah perpustakaan pengembangan *JavaScript* yang dipakai untuk mengembangkan antarmuka aplikasi *web*. Perpustakaan pengembangan ini tersedia secara terbuka untuk memudahkan proses pengembangan jika ingin membuat aplikasi *web* bersifat *Single Page Rendering*. Dengan *React JS*, kita bisa mengembangkan aplikasi besar dengan

memungkinkan skalabilitas besar dan dapat mengolah antarmuka dengan data yang sering berubah dengan mudah [24].

2.2.4. *Myers-Briggs Type Indicator*

Dengan singkatan MBTI, merupakan teori yang memberikan kategori pada jenis-jenis kepribadian manusia. Teori ini diciptakan oleh Isable Myers dan ibunya Katherine Briggs yang terinspirasi dari teori kepribadian Carl Jung pada masa Perang Dunia II dengan tujuan utama membantu penduduk menentukan karir cocok untuk para tentara rekrut yang tidak dikirim ke medan perang [25]. MBTI memiliki 8 kategori utama, lalu 4 buah akan digabungkan dalam sekuens tak terulang dan membentuk 16 sub kategori yang menyatakan karakteristik kepribadian [26]. Penjelasan dari ke-8 kategori utama, yaitu:

a) *Introversion* dan *Extraversion*:

Kedua Kategori ini sering dipakai untuk menilai kepribadian dari tingkat keaktifan seseorang. *Extraversion* (E) atau sering disebut *Ekstrovert* merupakan karakteristik orang yang selalu aktif dalam segala kegiatan. Mulai dari sosialisasi kepada orang lain, menyukai aktifitas luar rumah. Sedangkan *Introversion* (I) atau sering disebut *Introvert* merupakan kebalikan dari *Ekstrovert*.

b) *Intuition* dan *Sensing*:

Kedua kategori ini digunakan untuk menilai cara seseorang memproses informasi yang didapatkan. Seseorang dengan kategori *Intuition* (N) memproses informasi dari pola faktor dan hubungannya. Seorang *Intuition* berpikir teoritis dan konseptual, sehingga mereka inovatif dan imajinatif. Orang-orang ini suka memikirkan mengenai hal-hal apa saja yang dapat dicapai pada masa depan. Sedangkan *Sensing* (S) mengolah informasi berdasarkan factual yang realistis. Orang-orang ini mengambil Langkah dari pengalaman dan metode yang terbukti dan

efisien, sehingga dapat menjadi aplikatif. Seorang *Sensing* memilih untuk memikirkan masa yang sedang terjadi.

c) *Perceiving* dan *Judging*:

Kategori ini menilai langkah seseorang dalam sesuatu situasi dan kondisi. *Perceiving* (P) merupakan orang yang melakukan tanpa perencanaan. Orang-orang ini lebih suka melakukan tanpa berpikir, sehingga mereka fleksibel dan spontan. Seorang *Perceiving* sangat menyukai ketidakpastian dalam aksi mereka. Sedangkan *Judging* (J) merupakan kebalikan dari *Perceiving*. Seorang *Judging* akan membuat rencana dalam semua hal. Sehingga seorang *Judging* sistematis dalam mengambil Langkah dan tidak menyukai hal-hal yang mendadak.

d) *Feeling* dan *Thinking*:

Kategori ini menilai seseorang dalam mengambil keputusan. Seorang *Feeling* (F) menggunakan perasaan dan nilai keteguhan yang dimilikinya. Seorang *Feeling* terkesan harmonis dan dapat menjaga suatu hubungan. Sedangkan seseorang *Thinking* (T) menggunakan logika dan Analisa. Seorang *Thinking* terkesan analitis dan konsisten, sehingga dapat mempertahankan standar.

<p>ISTJ</p> <p>"DOING WHAT SHOULD BE DONE"</p> <p>Organizer • Compulsive Private • Trustworthy Rules 'n Regs • Practical</p> <p>MOST RESPONSIBLE</p>	<p>ISFJ</p> <p>"A HIGH SENSE OF DUTY"</p> <p>Amiable • Works Behind the Scenes Ready to Sacrifice • Accountable Prefers "Doing"</p> <p>MOST LOYAL</p>	<p>INFJ</p> <p>"AN INSPIRATION TO OTHERS"</p> <p>Reflective/Introspective Quietly Caring • Creative Linguistically Gifted • Psychic</p> <p>MOST CONTEMPLATIVE</p>	<p>INTJ</p> <p>"EVERYTHING HAS ROOM FOR IMPROVEMENT"</p> <p>Theory Based • Skeptical • "My Way" High Need for Competency Sees World as Chessboard</p> <p>MOST INDEPENDENT</p>
<p>ISTP</p> <p>"READY TO TRY ANYTHING ONCE"</p> <p>Very Observant • Cool and Aloof Hands-on Practically • Unpretentious Ready for what Happens</p> <p>MOST PRAGMATIC</p>	<p>ISFP</p> <p>"SEES MUCH BUT SHARES LITTLE"</p> <p>Warm and Sensitive • Unassuming Short Range Planner • Good Team Member In Touch with Self and Nature</p> <p>MOST ARTISTIC</p>	<p>INFP</p> <p>"PERFORMING NOBLE SERVICE TO AID SOCIETY"</p> <p>Strict Personal Values Seeks Inner Order/Peace Creative • Non-Directive • Reserved</p> <p>MOST IDEALISTIC</p>	<p>INTP</p> <p>"A LOVE OF PROBLEM SOLVING"</p> <p>Challenges others to Think Absent-minded Professor Competency Needs • Socially Cautious</p> <p>MOST CONCEPTUAL</p>
<p>ESTP</p> <p>"THE ULTIMATE REALIST"</p> <p>Unconventional Approach • Fun Gregarious • Lives for Here and Now Good at Problem Solving</p> <p>MOST SPONTANEOUS</p>	<p>ESFP</p> <p>"YOU ONLY GO AROUND ONCE IN LIFE"</p> <p>Sociable • Spontaneous Loves Surprises • Cuts Red Tape Juggles Multiple Projects/Events Quip Master</p> <p>MOST GENEROUS</p>	<p>ENFP</p> <p>"GIVING LIFE AN EXTRA SQUEEZE"</p> <p>People Oriented • Creative Seeks Harmony • Life of Party More Starts than Finishes</p> <p>MOST OPTIMISTIC</p>	<p>ENTP</p> <p>"ONE EXCITING CHALLENGE AFTER ANOTHER"</p> <p>Argues Both Sides of a Point to Learn Brinkmanship • Tests the Limits Enthusiastic • New Ideas</p> <p>MOST INVENTIVE</p>
<p>ESTJ</p> <p>"LIFE'S ADMINISTRATORS"</p> <p>Order and Structure • Sociable Opinionated • Results Driven Producer • Traditional</p> <p>MOST HARD CHARGING</p>	<p>ESFJ</p> <p>"HOST AND HOSTESSES OF THE WORLD"</p> <p>Gracious • Good Interpersonal Skills Thoughtful • Appropriate Eager to Please</p> <p>MOST HARMONIZING</p>	<p>ENFJ</p> <p>"SMOOTH TALKING PERSUADER"</p> <p>Charismatic • Compassionate Possibilities for People Ignores the Unpleasant • Idealistic</p> <p>MOST PERSUASIVE</p>	<p>ENTJ</p> <p>"LIFE'S NATURAL LEADERS"</p> <p>Visionary • Gregarious • Argumentative Systems Planners • Take Charge Low Tolerance for Incompetency</p> <p>MOST COMMANDING</p>

© Otto Kroeger Associates, 1997

Gambar 2.1. 16 kepribadian MBTI [13]

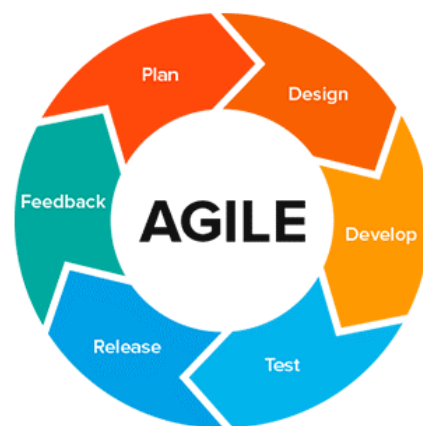
Dari kedelapan kategori di atas, terbentuklah 16 susunan kepribadian manusia yang masing-masing memiliki satu dari dua masing-masing 8 kategori utama.

2.2.5. *Kanban*

Kanban merupakan *framework* dari metodologi pengembangan perangkat lunak *Agile*. *Agile* tercipta berdasarkan sebuah manifesto yang dibentuk dari hasil musyawarah oleh 17 *software engineer* di resort ski, Utah [27]. Manifesto ini berisikan 4 pernyataan utama mengenai *Agile*, yaitu:

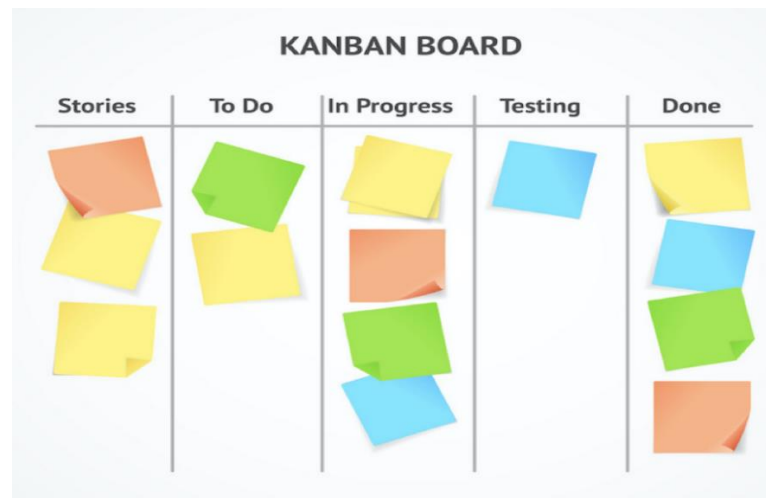
- a. *Individuals and interactions over processes and tools*
- b. *Working software over comprehensive documentation*
- c. *Customer collaboration over contract negotiation*
- d. *Responding to change over following a plan*

Metodologi *Agile* membawa kembali aspek-aspek tahapan pengembangan perangkat lunak, seperti manajemen proyek, desain perangkat lunak, dst, lalu disederhanakan dan dioptimalkan sedemikian rupa untuk kemudahan penggunaan dan memajukan aspek kesinambungan terstruktur [28].



Gambar 2.2. Metodologi *Agile*

Metodologi *Kanban* merupakan *framework* metodologi yang diadaptasi dari *Agile* untuk meningkatkan kualitas proses pengembangan perangkat lunak dari waktu ke waktu. Dibandingkan *framework Agile* lainnya, di dalam *Kanban* dapat memberikan gambaran jelas tentang aksi apa saja yang harus dilakukan, mengidentifikasi pembuangan tenaga kerja yang mengakibatkan penurunan efisiensi, dan cara meningkatkan kualitas dari masalah-masalah yang teidentifikasi dengan cepat dan efektif [28].



Gambar 2.3. Kanban Board

Kanban memiliki berbagai tahap. Tetapi, jenis tahap yang sering digunakan merupakan tahap penentuan fitur atau yang sering disebut *Backlog*, tahap fitur yang sudah ditentukan dan siap dikembangkan, tahap pengembangan fitur, tahap untuk menguji fungsionalitas fitur, dan tahap terkakhir jika pada tahap pengujian lancar maka fitur dinyatakan selesai [28].

2.2.6. *Cohen Kappa*

Koefisien *Cohen Kappa* merupakan penilaian indikator *Inter-Rater Agreement* yang biasa dipakai dalam menentukan hikmat dari total kesepakatan validitas terhadap suatu kategorikal deskriptif kualitatif yang dinilai oleh dua penilai yang ahli dalam bidang kategori tersebut [29]. Penilaian *Cohen Kappa* diciptakan oleh Jason Kappa dengan bertujuan untuk mengetahui nilai kecocokan indikator dari data yang bertema dominan dalam data bersifat kualitatif seperti dalam wawancara atau form kuesioner. Menurut Widhiarso (2015), tingkatan nilai untuk menentukan persetujuan validasi dalam *Cohen Kappa* antara lain [20] [21]:

Tabel 2.2. Tingkatan nilai *Cohen Kappa*

Indeks Kappa	Persetujuan
0.00	Sangat Lemah
0.00 - 0.20	Lemah
0.21 – 0.40	Sederhana Lemah
0.41 – 0.60	Sederhana
0.61 – 0.80	Baik
0.81 – 1.00	Sangat Baik

Untuk mencari nilai dari koefisien *Cohen Kappa*, rumus yang digunakan yaitu [30],

$$K = \frac{Po - Pc}{1 - Pc} \quad (2.1)$$

Dalam rumus di atas, K merupakan koefisien *Cohen Kappa*. Pa merupakan Proporsi kesepakatan teramati. Pc merupakan proporsi kesepakatan harapan. Pa didapatkan dengan menghitung total kesepakatan data valid dan data tidak valid. Po bisa didapatkan dengan menghitung *mean* dari

semua jumlah yang sama-sama disetujui. Untuk membantu mencari nilai P_o dan P_c , format proporsi persetujuan validitas dapat dilihat pada Tabel 2.3.

Tabel 2.3. Proporsi persetujuan validitas

		Penilai		Total
		Valid	Tidak Valid	
Penilai 2	Valid	A	B	Total
	Tidak Valid	C	D	Total
		Total	Total	

Untuk Mencari nilai P_o , rumus yang digunakan yaitu,

$$P_o = \frac{(A + C)}{(A + B + C + D)} \quad (2.2)$$

Untuk Mencari nilai P_c , rumus yang digunakan yaitu,

$$P_c = \frac{(A + B)(A + C) (C + D)(B + D)}{(A + B + C + D)} \quad (2.3)$$

2.2.7. PostgreSQL

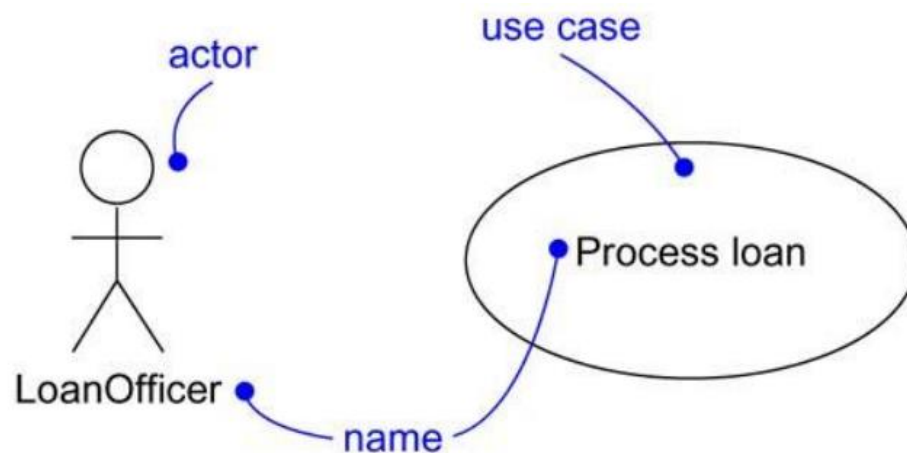
PostgreSQL merupakan sistem *object relational database management system* yang terbuka atau sering disebut dengan *open source*. Itu berarti struktur dan bahasa *database* ini sama dengan sistem lain yang mengimplementasikannya. *PostgreSQL* menekankan ekstensibilitas, kreativitas, serta kompatibilitas untuk lintas platform [31].

2.2.8. *Unified Modeling Language*

Unified Modeling Language (UML) adalah struktur standar menulis cetak biru perangkat lunak yang digunakan untuk memvisualisasikan, menentukan, membangun, dan mendokumentasikan artefak dari sistem intensif perangkat lunak. UML cocok untuk merancang berbagai sistem, mulai dari sistem informasi perusahaan besar hingga aplikasi web terdistribusi dan bahkan sistem tertanam waktu nyata dengan persyaratan kinerja yang ketat. Beberapa diagram yang terdapat dalam UML yaitu *Use Case Diagram*, *Activity Diagram*, dan *Sequence Diagram* [32].

2.2.9. *Use Case Diagram*

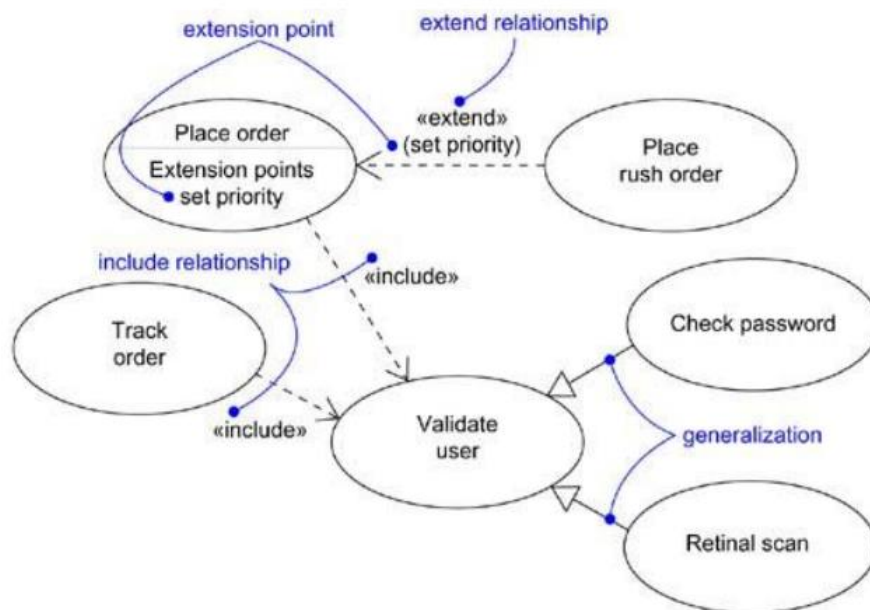
Use Case Diagram adalah diagram untuk memodelkan aspek dinamis dari sistem. *Use case diagram* adalah pusat untuk memodelkan perilaku sistem, subsistem, atau kelas. Masing-masing menunjukkan serangkaian *use case* dan *actor* serta hubungannya. *Use Case Diagram* memiliki beberapa komponen, yaitu *Use Case* dan *Actor* [32].



Gambar 2.4. *Actor dan Use Case*

Menurut gambar 2.4, terdapat komponen *use case* yang berbentuk oval dan *actor* yang berbentuk *stickman*. Komponen *use case* menentukan perilaku sistem atau bagian dari sistem dan merupakan deskripsi serangkaian urutan tindakan, termasuk varian, yang dilakukan sistem untuk menghasilkan hasil nilai yang dapat diamati bagi seorang aktor. *Use case* diterapkan untuk menangkap perilaku yang diinginkan dari sistem yang sedang dikembangkan tanpa harus menentukan bagaimana perilaku itu diimplementasikan [32].

Komponen *actor* mewakili serangkaian peran yang koheren yang diinteraksi oleh pengguna saat berinteraksi dengan *use case*. Biasanya, komponen *actor* mewakili peran yang dimainkan manusia, perangkat keras, atau bahkan sistem lain berinteraksi dengan sistem lain [32].



Gambar 2.5. *Generalization, include, dan extend*

Hubungan antar *use case* diatur dengan menentukan *generalization, include, dan extend*. Hubungan antar *use case* diterapkan untuk

memfaktorkan perilaku umum (dengan menarik perilaku tersebut dari *use case* lain yang disertakan) dan untuk memfaktorkan varian (dengan mendorong perilaku tersebut ke *use case* lain yang memperluasnya) [32].

Generalization adalah hubungan antara hal umum (disebut *superclass* atau *parent*) dan jenis yang lebih spesifik dari hal itu (disebut *subclass* atau *child*). Contoh, ditemukan *Window* kelas umum dengan jenisnya yang lebih spesifik, *MultiPaneWindow*. Dengan *generalization* hubungan dari *child* ke *parent*, *child* (*MultiPaneWindow*) akan mewarisi semua struktur dan perilaku *parent* (*Window*). *Child* bahkan dapat menambahkan struktur dan perilaku baru, atau mungkin mengubah perilaku *parent*. Dalam hubungan *generalization*, contoh *child* dapat digunakan kapan saja contoh orang tua beraksi dan artinya anak dapat menggantikan orang tua [32].

Sebuah hubungan *include* antara *use case* berartikan dasar *use case* secara eksplisit menggabungkan perilaku *use case* lain di lokasi yang ditentukan di dasar. *Use case* yang disertakan tidak pernah berdiri sendiri, tetapi hanya dibuat sebagai bagian dari basis yang lebih besar yang menyertakannya. Bisa dianggap bahwa *include* sebagai perilaku menarik *use case* dasar dari *use case* pemasok. Hubungan *include* digunakan untuk menghindari pendeskripsian alur peristiwa yang sama beberapa kali, dengan menempatkan perilaku umum dalam *use case* sendiri [32].



Hubungan *extend* antara *use case* berartikan *use case* dasar secara implisit menggabungkan perilaku *use case* lain di lokasi yang ditentukan secara tidak langsung oleh *extend use case*. *Use case* dasar mungkin berdiri sendiri, tetapi dalam kondisi tertentu, perilakunya dapat diperluas oleh perilaku *use case* lainnya. *Use case* dasar ini dapat diperluas hanya pada titik-titik tertentu yang disebut, tidak mengherankan, titik *extend*-nya. Hubungan *extend* digunakan untuk memodelkan bagian dari *use case* yang mungkin dilihat pengguna sebagai perilaku sistem opsional, sehingga dapat memisahkan perilaku opsional dari perilaku wajib. Hubungan *extend* juga dapat


menghubungkan perluasan untuk memodelkan subaliran terpisah yang dijalankan hanya dalam kondisi tertentu [32].

2.2.10. *Activity Diagram*

Activity diagram digunakan untuk memodelkan aspek dinamis dari suatu sistem. Sebagian besar, ini melibatkan pemodelan langkah-langkah berurutan dalam proses komputasi. Dengan *activity diagram*, memungkinkan juga untuk memodelkan aliran objek saat bergerak dari satu keadaan ke keadaan lain pada titik yang berbeda dalam aliran kontrol. *Activity diagram* dapat berdiri sendiri untuk memvisualisasikan, menentukan, membangun, dan mendokumentasikan dinamika masyarakat objek, atau dapat digunakan untuk memodelkan aliran kontrol suatu operasi. Sementara *activity diagram* menekankan aliran kontrol dari objek ke objek, diagram aktivitas menekankan aliran kontrol dari aktivitas ke aktivitas. Aktivitas adalah eksekusi nonatomik yang sedang berlangsung di dalam mesin negara. Aktivitas pada akhirnya menghasilkan beberapa tindakan, yang terdiri dari komputasi atom yang dapat dieksekusi yang menghasilkan perubahan status sistem atau pengembalian nilai [32].

Tabel 2.4. Komponen *activity diagram* [32]

Komponen	Keterangan
	Komponen <i>activity</i> yang berbentuk persegi dengan sudut lengkung digunakan untuk mewakili serangkaian tindakan
	Komponen <i>action</i> yang berbentuk persegi dengan sudut lengkung dan tinggi yang lebih pendek mewakili sebuah aksi yang sedang dilakukan

Komponen	Keterangan
	Komponen <i>control flow</i> yang berbentuk arah panah digunakan untuk menunjukkan alur dari aksi yang sedang dieksekusi




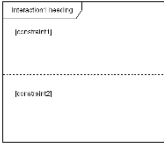
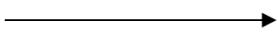
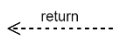
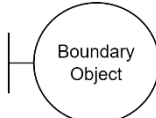
Action state dan *activity state* hanyalah jenis status khusus dalam mesin status. Saat *user* memasuki *action state* atau *activity state*, *user* tersebut cukup melakukan *action state* atau *activity state* tersebut, lalu saat *user* selesai, kontrol beralih ke *action state* atau *activity state* berikutnya. Oleh karena itu, *activity state* agak singkat. *Activity state* secara semantik setara dengan memperluas grafik *activity state* (dan secara transitif) di tempat hingga *user* hanya melihat *action state*. Meskipun demikian, *activity state* penting karena membantu *user* memecah komputasi kompleks menjadi beberapa bagian, dengan cara yang sama seperti *user* menggunakan operasi untuk mengelompokkan dan menggunakan kembali ekspresi. Ketika *action state* atau *activity state* dari suatu keadaan selesai, aliran kontrol segera beralih ke *action state* atau keadaan *activity state* berikutnya. *User* menentukan alur ini dengan menggunakan transisi untuk menunjukkan jalur dari satu *action state* atau *activity state* ke *action state* atau *activity state* berikutnya. Di UML, *user* merepresentasikan transisi sebagai garis terarah sederhana [32],

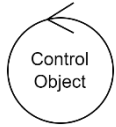
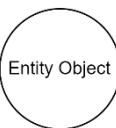
2.2.11. *Sequence Diagram*

Sequence diagram digunakan untuk memodelkan aspek dinamis dari suatu sistem. Sebagian besar, ini melibatkan pemodelan contoh konkret atau prototipe dari kelas, antarmuka, komponen, dan *node*, bersama dengan pesan yang dikirim di antara mereka, semuanya dalam konteks

skenario yang mengilustrasikan perilaku. *Sequence diagram* dapat berdiri sendiri untuk memvisualisasikan, menentukan, membangun, dan mendokumentasikan dinamika masyarakat objek tertentu, atau mereka dapat digunakan untuk memodelkan satu aliran kontrol tertentu dari kasus penggunaan. *Sequence diagram* tidak hanya penting untuk memodelkan aspek dinamis dari suatu sistem, tetapi juga untuk membangun sistem yang dapat dieksekusi melalui rekayasa maju dan mundur [32].

Tabel 2.5. Komponen *sequence diagram* [32]

Komponen	Keterangan
	Komponen aktor mewakili pengguna yang terlibat dalam interaksi dengan sistem, baik secara internal maupun eksternal
	Komponen <i>Activation box</i> yang berbentuk persegi panjang mewakili durasi waktu yang dibutuhkan oleh suatu objek untuk menyelesaikan tugas tertentu. Semakin lama waktu yang dibutuhkan, maka <i>activation box</i> akan semakin panjang
	Komponen <i>Lifelines</i> berbentuk garis putus-putus, digunakan untuk menunjukkan urutan kejadian yang terjadi pada objek selama pembuatan <i>diagram</i>
	<i>Option loop</i> adalah komponen <i>diagram</i> yang sangat berguna untuk menggambarkan bagaimana sistem berperilaku dalam situasi tertentu, terutama jika ada percabangan atau alternatif jalur yang harus dipertimbangkan
	<i>Synchronous message</i> adalah salah satu komponen <i>diagram</i> yang dilambangkan dengan simbol panah tebal ke kanan, dan digunakan untuk menunjukkan bahwa pengirim harus menunggu <i>respon</i> pesan sebelum melanjutkan
	<i>Reply message</i> adalah salah satu komponen <i>diagram</i> yang dilambangkan dengan simbol panah putus-putus ke kiri, dan berfungsi sebagai balasan atas suatu panggilan tertentu
	Komponen <i>boundary</i> dalam <i>diagram</i> biasanya terletak pada batas luar sistem, dan dapat berupa antarmuka pengguna atau alat lain yang digunakan untuk berinteraksi antara pengguna dan sistem

Komponen	Keterangan
 <p data-bbox="430 388 544 451">Control Object</p>	<p data-bbox="690 367 1372 451">Komponen <i>control</i> dalam sebuah sistem bertanggung jawab untuk mengatur alur informasi dalam skenario bisnis, dan dapat mempengaruhi perilaku teknis dari sistem</p>
 <p data-bbox="430 535 544 598">Entity Object</p>	<p data-bbox="690 514 1372 598"><i>Entity</i> adalah komponen sistem yang berperan menyimpan informasi atau data, dan biasanya direpresentasikan sebagai objek model atau <i>beans</i></p>

Di dalam *sequence diagram*, pertama ada garis hidup objek. Garis hidup objek adalah garis putus-putus vertikal yang mewakili keberadaan suatu objek selama periode waktu tertentu. Sebagian besar objek yang muncul dalam diagram interaksi akan tetap ada selama interaksi berlangsung, jadi semua objek ini disejajarkan di bagian atas diagram, dengan garis hidupnya digambar dari bagian atas diagram ke bagian bawah. Objek dapat dibuat selama interaksi. Garis hidup mereka dimulai dengan penerimaan pesan yang distereotipkan sebagai kreasi. Objek dapat dihancurkan selama interaksi. Garis hidup mereka diakhiri dengan penerimaan pesan yang distereotipkan sebagai kehancuran [32].

Kedua, ada fokus kontrol. Fokus kontrol adalah persegi panjang tipis dan tinggi yang menunjukkan periode waktu selama objek melakukan tindakan, baik secara langsung maupun melalui prosedur bawahan. Bagian atas persegi panjang disejajarkan dengan awal aksi dan dapat ditandai dengan pesan balasan. Anda dapat menampilkan kumpulan fokus kontrol dengan menumpuk fokus kontrol lain sedikit ke kanan induknya. Jika Anda ingin lebih tepat tentang di mana fokus kontrol berada, Anda juga dapat menaungi wilayah persegi panjang di mana metode objek sebenarnya menghitung [32].

2.2.12. *Unit Testing*

Unit testing biasanya dilakukan oleh *developer*. Dengan *unit testing*, memungkinkan untuk menguji metode dan kelas individual, tetapi umumnya *developer* tidak menguji konfigurasi program yang lebih besar. Seorang *developer* juga biasanya tidak menguji antarmuka atau interaksi pustaka, kecuali yang mungkin benar-benar digunakan oleh metode pengembangan. Karena disaat sedang melakukan *unit testing*, seorang *developer* tahu bagaimana semua metode ditulis, seperti apa data seharusnya, apa tanda tangan metode itu, dan apa nilai dan tipe yang dikembalikan seharusnya. Ini dikenal sebagai pengujian kotak putih. Ini harus benar-benar disebut pengujian kotak transparan, karena asumsinya adalah Anda dapat melihat semua detail kode yang sedang diuji [33].