

BAB II LANDASAN TEORI

2.1 Tinjauan Pustaka

Studi tabel digunakan sebagai bahan pertimbangan yang memiliki kaitannya dengan penelitian yang sedang dilakukan saat ini. Penelitian terdahulu yang digunakan oleh penulis dapat dilihat pada table 2.1:

Tabel 2. 1 Studi Table

1	Nama(tahun)	Abdul Azis , Ito Setiawan & Arif Risqiantoro (2019).
	Judul	Aplikasi <i>E-Voting</i> Untuk Pemilihan Kepala dusun Berbasis <i>Website</i>
	Tujuan Penelitian	Tujuan dari penelitian ini adalah menghasilkan sebuah Aplikasi Pemilihan Kepala dusun Berbasis <i>Web</i> sebagai solusi untuk sistem Pemilihan Kepala dusun Tayem Timur yang <i>efektif</i> menambah partisipasi pemilih dengan memudahkan pemilih menggunakan hak pilihnya, menghilangkan suara tidak sah, menanggulangi resiko kecurangan dan serta dapat menekan biaya penyelenggaraan kegiatan Pilkades.
	Metode/ <i>Tools</i>	Sistem dikembangkan menggunakan metode <i>prototype</i> .
	Hasil Penelitian	Hasil pada penelitian ini adalah Aplikasi Pemilihan Kepala dusun Berbasis <i>Web</i> untuk Desa Tayem Timur.
	Kesimpulan/Saran	Kesimpulan pada penelitian ini telah berhasil membangun Aplikasi Pemilihan Kepala dusun Berbasis <i>Web</i> di Desa Tayem Timur. Berdasarkan <i>User Acceptance Testing</i> Aplikasi Pemilihan Kepala dusun Berbasis <i>Web</i> ini telah sesuai dengan perancangan yang telah ditetapkan pada tahap desain

		<p>sistem serta sesuai dengan keinginan pelanggan (Panitia Pemilihan Kepala dusun Tayem Timur).</p> <p>Saran dari penelitian ini Aplikasi ini perlu dikembangkan dalam segi desain antarmuka agar lebih menarik dan interaktif. Aplikasi ini belum menerapkan sistem keamanan terenkripsi untuk menjamin keamanan dari tindak peretasan data.</p>
2	Nama(tahun)	Angga Widwan Krismanto, Rifki Adhitama & Novian Adi Prasetyo (2022).
	Judul	Rancang Bangun Aplikasi <i>E-Voting</i> Pemilihan Ketua Pondok Pesantren Berbasis <i>Web</i> Menggunakan Metode <i>Waterfall</i>
	Tujuan Penelitian	Tujuan dari penelitian ini adalah untuk mengalih fungsikan pemilihan manual menuju pemilihan dengan memanfaatkan sistem digital, untuk mempermudah kinerja KPUP (Komisi Pemilihan Umum Pondok) dalam mengelola hasil suara pemilihan ketua Pondok Pesantren Darussalam Dukuhwaluh-Purwokerto.
	Metode/ <i>Tools</i>	Sistem dikembangkan menggunakan metode <i>waterfall</i> .
	Hasil Penelitian	Berdasarkan hasil data <i>user</i> yang telah peneliti lakukan menggunakan metode <i>black box testing</i> dapat disimpulkan bahwa sistem yang sedang dikembangkan berupa <i>voting</i> berbasis <i>web</i> calon Lurah Pondok Pesantren Darussalam Dukuhwaluh Purwokerto (PILKALUR) menjawab rumusan masalah dari penelitian.
	Kesimpulan/Saran	Kesimpulan pada penelitian ini dari hasil pengujian sistem menggunakan <i>black box testing</i> , 60 fitur yang di uji sedikitnya ada dua diantaranya yang gagal di jalankan, dengan demikian dapat disimpulkan dari hasil pengujian bahwa aplikasi <i>E-Voting</i> layak untuk digunakan. Semua fungsi yang terdapat pada Aplikasi

		<i>E-Voting</i> berbasis web sudah sesuai dengan perancangan yang sudah disusun.
3	Nama(tahun)	Adi Hermawansyah & Rosmiati Nur (2019)
	Judul	Perancangan Sistem Informasi <i>E-Voting</i> Pemilihan Ketua RT Dengan <i>verifikasi</i> Nomor Induk Keluarga Berbasis <i>web</i> .
	Tujuan Penelitian	Tujuan penelitian ini adalah untuk mempermudah warga dalam memilih ketua RT dimanapun sehingga hasil suara yang diharapkan lebih maksimal agar kegiatan pemilihan ketua RT ini dapat berjalan dengan <i>optimal</i> dan <i>efisien</i> .
	Metode/ <i>Tools</i>	Sistem dikembangkan menggunakan metode SDLC (<i>System Development Lift Cycle</i>).
	Hasil Penelitian	Hasil penelitian ini adalah melalui beberapa pengujian ini aplikasi <i>E-Voting</i> berbasis web dapat berfungsi sesuai dengan yang diharapkan serta aplikasi yang dibangun telah cukup memenuhi tujuan awal dan kedepanya dapat dijadikan sebuah projek yang berguna bagi masyarakat meskipun masih ada terdapat beberapa kelebihan dan kelemahan.
	Kesimpulan/Saran	Kesimpulan pada penelitian ini dalam melakukan perancangan dan membangun sistem informasi <i>E-Voting</i> ini menggunakan database PHP dan Mysql sebagai penyimpanan data, pengolahan data warga, pemilihan serta perhitungan. Sistem <i>E-Voting</i> ini mengurangi permasalahan pelaksanaan <i>voting konvensional</i> sebelumnya. Saran pada penelitian ini untuk pengembangan lebih lanjut sebaiknya ditambahkan tampilan yang lebih

		menarik lagi pada webnya tanpa mengurangi kemudahan pengguna.
4	Nama(tahun)	Amelia Melati Putri, Elzy Novianti, Suri Wulandari, Muhammad Fadli Ansyari, Muhammad Rzky Fadillah, Muhammad Luthfi Hamzah (2022).
	Judul	Perancangan Sistem Informasi <i>E-Voting</i> untuk Pemilihan Ketua OSIS Menggunakan <i>Agile Method</i> .
	Tujuan Penelitian	Tujuan dari penelitian ini adalah dengan memanfaatkan teknologi terkomputerisasi pelaksanaan pemilihan kepala OSIS (<i>E-Voting</i>) berbasis web sanggup menciptakan pemilihan kepala OSIS lebih mudah.
	Metode/ <i>Tools</i>	Sistem dikembangkan menggunakan metode <i>Agile Method</i> .
	Hasil Penelitian	Hasil penelitian ini berupa perancangan <i>web</i> menuntaskan kasus-kasus mengenai proses pemilihan Ketua OSIS bisa dilaksanakan secara <i>efektif, efisien, cepat & transparan</i> .
	Kesimpulan/Saran	Kesimpulan pada penelitian ini proses pemilihan dapat menghemat waktu dan biaya. Harapannya dengan sistem ini proses pemilihan dapat berjalan secara <i>efektif dan efisien</i> .
5	Nama(tahun)	Jelita Mutiara Setyo Budi & I Kadek Dwi Nuryana (2020).
	Judul	Rancang Bangun Aplikasi <i>E-Voting</i> Berbasis <i>Web</i> Menggunakan <i>Framework Laravel</i> .
	Tujuan Penelitian	Tujuan dari penelitian ini adalah merancang dan mengimplementasikan aplikasi <i>E-Voting</i> Berbasis <i>Web</i> menggunakan <i>Framework Laravel</i> dalam pemilihan umum ketua OSIS dan menggunakan metode yang mudah sehingga dapat membantu dalam pengembangan sistem aplikasi.

Metode/ <i>Tools</i>	Sistem dikembangkan menggunakan metode <i>Spiral</i> .
Hasil Penelitian	Sebuah aplikasi Rancang Bangun Aplikasi <i>E-Voting</i> Berbasis <i>Web</i> Menggunakan <i>Framework Laravel</i> (Studi Kasus : Pemilihan Ketua OSIS di SMA Negeri 1 Kebomas) yang bertujuan untuk mengembangkan sistem <i>voting</i> secara elektronik berbasis <i>website</i> menggunakan <i>framework laravel</i> di SMA Negeri 1 Kebomas Gresik.
Kesimpulan/Saran	<p>Kesimpulan pada penelitian ini aplikasi <i>E-Voting</i> pemilihan Ketua OSIS berbasis web yang dibuat dengan bahasa pemrograman PHP dan <i>Laravel</i> sebagai <i>framework</i>. Aplikasi ini menggunakan MySQL sebagai database dan dirancang dengan metode <i>spiral</i>. Dengan adanya aplikasi <i>E-Voting</i> ini maka pelaksanaan pemungutan suara akan lebih ekonomis dan efisien sehingga memudahkan pelaksana kegiatan maupun pihak sekolah. Dengan digunakannya aplikasi <i>E-Voting</i>, secara tidak langsung kita juga berupaya dalam mengimplementasikan kecanggihan teknologi dalam kehidupan sehari-hari dan juga turut andil dalam mengurangi limbah kertas.</p> <p>Saran pada penelitian ini adalah aplikasi ini sebaiknya perlu dikembangkan menjadi berbasis <i>mobile</i>, dengan tujuan memudahkan dalam melakukan kegiatan <i>E-Voting</i> di <i>smartphone</i> masingmasing siswa, mengingat zaman sekarang memiliki <i>smartphone</i> sudah menjadi kebutuhan. Sistem keamanan pada aplikasi ini disarankan untuk dikembangkan menjadi lebih baik untuk mengantisipasi kecurangan-kecurangan yang mungkin dapat terjadi saat kegiatan pemungutan suara dilaksanakan.</p>

2.2 E-Voting

E-Voting (Saputro et al., 2022) secara umum adalah penggunaan teknologi komputer pada pelaksanaan *Voting*. menjelaskan secara umum sejarah, jenis *Electronic Voting*, keuntungan dan kerugian dalam penggunaannya. Pilihan teknologi yang digunakan dalam implementasi dari *e Voting* sangat bervariasi, seperti penggunaan *smart card* untuk otentikasi pemilih, penggunaan internet sebagai sistem pemungutan suara, penggunaan *touch screen* sebagai pengganti kartu suara, dan masih banyak variasi teknologi yang digunakan.

2.3 Pemilihan

Pemilihan (Fariyanto dan Ulum, 2021) untuk menghimpun keinginan masyarakat untuk menentukan solusi yang terbaik dalam menyelesaikan masalah. Sistem pemilihan umum yang dikenal masyarakat adalah pemilihan atau pilkada yang diselenggarakan oleh Komite Pemilihan Umum (KPU), namun kegiatan pemilihan biasanya dilakukan dalam skala yang lebih kecil, seperti pemilihan kepala dusun. Pemilihan kepala dusun biasanya dilaksanakan setelah masa jabatan kepala dusun berakhir.

2.4 Kepala dusun

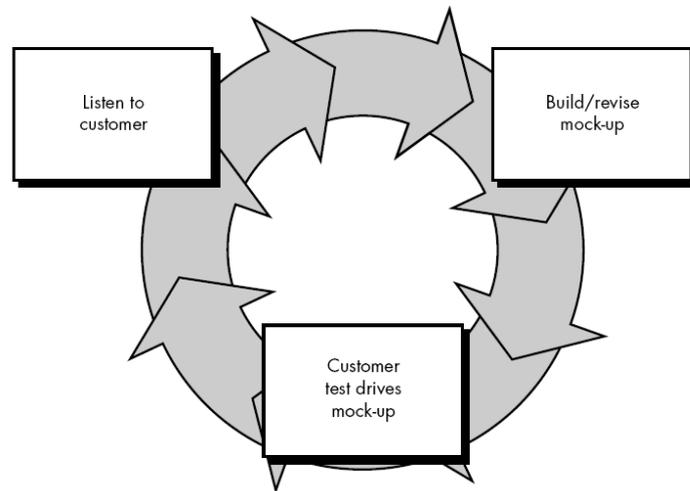
Kepala dusun (Parjiyana et al., 2019) bertugas melaksanakan keputusan desa di dusunnya masing-masing, kemudian melaksanakan kebijakan dari kepala desa di wilayah dusunnya masing-masing.

2.5 Website

Website (Novitasari et al., 2021) sejumlah halaman *web* yang memiliki topik saling terkait, terkadang disertai pula dengan berkas-berkas gambar, video, atau berkas lainnya. Situs *web* merupakan kumpulan dari halaman *web* yang sudah dipublikasikan di jaringan internet dan memiliki domain atau URL (*Unified Resource Locator*) yang dapat diakses semua pengguna internet dengan cara mengetikkan alamatnya.

2.6 Metode Pengembangan Sistem *Prototype*

Model *prototype* (Yoko et al., 2019) sebuah metode yang mengharuskan pengembang perangkat lunak membuat sebuah *mockup* berupa *model* aplikasi, dimana pengguna tidak bisa menyajikan informasi secara jelas mengenai kebutuhan yang sesuai dengan keinginannya. Hasil dari model *prototype* berupa mockup yang akan menjadi rujukan *model desain* yang akan digunakan. Sistem dengan model *prototype* memungkinkan pengguna agar mengetahui seperti apa tahapan sistem dibuat sehingga sistem mampu beroperasi dengan baik. Metode *prototype* digunakan pada penelitian ini dimaksudkan untuk mendapatkan representasi dari pemodelan aplikasi yang akan dibuat. Rancangan aplikasi awal mulanya berbentuk *mockup* selanjutnya akan dievaluasi oleh pengguna. Setelah mockup dievaluasi pengguna tahap selanjutnya *mockup* menjadi bahan rujukan bagi pengembang *software* untuk merancang aplikasi.



Gambar 2. 1 Model *Prototype* (Rosa A.s & M. Shalahuddin, 2019)

1. *Listen To Customer*

Pada tahap ini melakukan identifikasi kebutuhan pekon Suka Banjar atau pengguna sistem untuk memperoleh informasi tentang keinginan ataupun masalah yang terjadi di pekon Suka Banjar. Informasi yang didapatkan yaitu roses pendataan masyarakat yang berhak memilih masih dilakukan secara manual dengan cara panitia pemilihan mendatangi rumah-rumah pemilih secara satu persatu, data pemilih dicatat menggunakan *Microsoft Excel* berdasarkan urutan RT/RW masing-masing, pendataan pun harus diselesaikan paling lambat 7 (tujuh) hari sebelum pemungutan suara, setelah itu hasil pendataan tersebut dicetak dan disahkan oleh ketua panitia pemilihan dan sekretaris panitia pemilihan. Selanjutnya informasi yang didapatkan yaitu tentang proses pemilihan dilakukan, proses pemilihan masih dilakukan secara manual yaitu dengan cara masyarakat mendatangi tempat pemungutan suara (TPS) masing-masing yang telah ditetapkan panitia pemilihan dengan membawa surat keterangan memiliki hak suara, setelah itu masyarakat mendatangi meja panitia yang ada di TPS dan akan didata ulang oleh panitia

berdasarkan NIK dan nomor KK, setelah di data panitia masyarakat akan mendapatkan surat kertas suara yang di berikan oleh panitia, lalu masyarakat akan masuk kedalam bilik pencoblosan yang telah dibuat oleh panitia, setelah masyarakat melakukan pencoblosan masyarakat akan melipat kembali kertas surat suara dan memasukkan kertas suara tersebut kedalam kotak suara yang telah disediakan oleh panitia, lalu masyarakat akan mencelupkan jari pada tinta yang telah disiapkan oleh panitia sebagai bukti bahwa telah menyelesaikan pencoblosan dengan menggunakan hak suaranya. Setelah semua masyarakat selesai melakukan pencoblosan panitia akan mengumpulkan semua kertas surat suara di kantor desa, setelah itu akan dilakukan perhitungan kertas surat suara secara satu persatu oleh panitia yang disaksikan oleh semua panitia pemilihan dan beberapa masyarakat yang telah di pilih dan ditetapkan sebagai saksi pemilihan. Proses selanjutnya pengesahan hasil pemilihan, sekretaris panitia pemilihan akan membuat berita acara pemilihan yang akan ditetapkan oleh Badan Permusyawaratan Desa (BPD), BPD akan menandatangani berita acara tersebut dan menyerahkan hasil pemilihan beserta berita acara kepada Wali Kota melalui camat, berita acara dan hasil pemilihan yang telah ditetapkan oleh BPD akan disahkan oleh Wali Kota sebagai kepala dusun terpilih.

2. Build and Resive Mock-up

Pada tahap ini membuat rancangan prototype yang sesuai dengan kebutuhan pengguna sistem dengan menggunakan UML (*Unified Modeling Language*) dan

rancangan antarmuka sesuai yang diinginkan pengguna sistem berdasarkan masalah yang terjadi.

3. *Customer Test Drives Mock-up*

Tahapan ini dilakukan pengujian sistem dan evaluasi mengenai sistem yang telah dibuat apakah semua yang ada pada sistem telah memenuhi kebutuhan pengguna atau belum. Jika hasil yang di dapat belum dapat memenuhi kebutuhan pengguna sistem maka pengembang sistem akan memperbaiki *prototype* tersebut.

2.7 Pengujian ISO 9126

ISO-9126 (Supriyono, 2019) *standard* pengukuran jaminan kualitas perangkat lunak. Ide dasarnya adalah menentukan dan mengevaluasi produk perangkat lunak untuk mengetahui kualitas *internal* dan kualitas *eksternal* serta hubungannya dengan atribut kualitas. Untuk memberikan kerangka kerja kualitas perangkat lunak, *Organisasi Internasional untuk Standardisasi* (ISO) dan *Komisi Teknis Elektro Internasional* (IEC) telah mengembangkan standar ISO 9126 untuk rekayasa Perangkat Lunak. Kualitas produk, untuk menyediakan spesifikasi dan model evaluasi yang *komprehensif* untuk kualitas produk perangkat lunak. Kualitas dianggap sebagai salah satu *driver* paling penting dari keberhasilan perangkat lunak. Dalam ISO 9126, totalitas atribut kualitas diklasifikasikan ke dalam struktur *hierarki* karakteristik pohon dan sub-karakteristik. Ini menetapkan enam karakteristik: Fungsi, Keandalan, Kegunaan, *Efisiensi*, *Maintabilitas* dan *Portabilitas*.

Pada penelitian ini pengujian ISO 9126 hanya dilakukan 2(dua) pengujian (*functionality, usability*). Berikut table karakteristik dan sub-karakteristik aspek pengujian ISO 9126 :

Tabel 2. 2 Tabel Karakteristik dan Sub-Karakteristik

Karakteristik	Sub-Karakteristik
<i>Functionality</i>	<p><i>Suitability</i></p> <p>Bisakah perangkat lunak melakukan tugas yang dibutuhkan?</p>
	<p><i>Accurateness</i></p> <p>Bisakah perangkat lunak menghasilkan hasil yang diharapkan?</p>
	<p><i>Security</i></p> <p>Bisakah perangkat lunak berinteraksi dengan sistem lain?</p>
	<p><i>Interoperability</i></p> <p>Apakah perangkat lunak dilengkapi dengan tindakan pengamanan?</p>
<i>Reability</i>	<p><i>Maturity</i></p> <p>Bisakah sebagian besar kesalahan dihilangkan dari waktu ke waktu?</p>
	<p><i>Fault</i></p> <p>Tolerance Bisakah perangkat lunak menangani kesalahan?</p>
	<p><i>Recoverability</i></p>

	Apakah perangkat lunak dapat bekerja dan mengembalikan data?
<i>Usability</i>	<i>Understandability</i> Bisakah perangkat lunak dipahami dengan mudah?
	<i>Learnability</i> Bisakah perangkat lunak dipelajari dengan mudah?
	<i>Operability</i> Bisakah perangkat lunak dioperasikan dengan upaya minimal?
	<i>Attractiveness</i> Apakah perangkat lunak memiliki antarmuka yang menarik?
<i>Efficiency</i>	<i>Time behavior</i> Apakah perangkat lunak berperilaku tepat waktu
<i>Portability</i>	<i>Adaptability</i> Bisakah perangkat lunak diadaptasi dengan mudah?
	<i>Installability</i> Bisakah perangkat lunak diinstal dengan mudah?
	<i>Coexistence</i>

	Bisakah perangkat lunak bekerja dengan sistem perangkat lunak yang ada?
	<i>Replaceability</i> Bisakah perangkat lunak diganti dengan produk serupa?

- a. *Functionality* (Fungsionalitas) adalah kemampuan perangkat lunak dalam menyediakan fungsi sesuai dengan kebutuhan pengguna ketika digunakan dalam kondisi yang spesifik. Sebuah website harus mampu diakses oleh pengguna dengan lingkungan sistem yang berbeda tanpa mengurangi fungsi yang ada.
- b. *Reliability* (Kehandalan) adalah kemampuan perangkat lunak untuk mempertahankan kinerjanya ketika digunakan dalam kondisi tertentu. Kehandalan suatu perangkat lunak dapat ditentukan dari jumlah masukan yang dapat menyebabkan kegagalan pada saat sedang dijalankan. Hal ini dapat diamati oleh pengguna.
- c. *Usability* (Kebergunaan) berkaitan dengan apakah produk dapat mencapai tujuan tertentu secara efektif, efisien, dan memperoleh kepuasan setelah digunakan. Aspek kebergunaan dapat diukur dengan menggunakan instrumen berupa kuisisioner. Kuisisioner ini akan diisi oleh pengguna setelah mereka mencoba menggunakan aplikasi. Kemudahan penggunaan adalah derajat tentang bagaimana kemudahan perangkat lunak digunakan, dimana hal ini sering diindikasikan menggunakan sub-atribut kemudahan untuk dipahami, kemudahan untuk dipelajari, dan operabilitas.

- d. *Efficiency* (Efisiensi) berkaitan dengan kemampuan perangkat lunak untuk memberikan kinerja yang sesuai terhadap jumlah sumber daya yang digunakan pada saat keadaan tersebut. Apabila ingin menciptakan perangkat lunak yang handal, aspek *efficiency* harus benar-benar diperhatikan. Penggunaan sumber daya yang tidak efisien, misalnya menggunakan algoritma yang tidak tepat dapat menyebabkan kinerja perangkat lunak menjadi lamban.
- e. *Portability* (Portabilitas) berkaitan dengan kemampuan suatu perangkat lunak beroperasi atau bekerja dalam lingkungan yang berbeda-beda. Untuk menguji tingkat portabilitas dari aplikasi berbasis web, aplikasi tersebut dicoba dijalankan menggunakan browser yang berbeda. Selain itu, percobaan menggunakan perangkat yang berbeda juga dapat menentukan tingkat portabilitas dari software tersebut.
- f. *Maintainability* (Pemeliharaan) berkaitan dengan kemampuan perangkat lunak untuk dimodifikasi. Modifikasi meliputi koreksi, perbaikan atau adaptasi terhadap perubahan lingkungan, persyaratan, dan spesifikasi fungsional (Pamungkas, 2018).

2.7.1. Skala Pengukuran

Metode Skala *Likert* (Setyawan dan Atapukan, 2018) metode yang digunakan untuk mengukur tingkat kepuasan pengguna menggunakan skala *likert*. Skala *likert* memiliki empat atau lebih butir-butir pertanyaan yang dikombinasikan sehingga membentuk sebuah skor atau nilai yang mempersentasikan sifat individu, misalkan pengetahuan, sikap dan perilaku. Skala *likert* dapat juga dikatakan sebagai skala *psikometrik* yang umum digunakan dalam kuesioner dan merupakan skala

yang paling banyak digunakan untuk penelitian. Skala *likert* dapat diperlihatkan pada Tabel 2.3 :

Tabel 2. 3 Skala Likert

Skala	Kategori
5	Sangat Setuju
4	Setuju
3	Cukup
2	Kurang Setuju
1	Tidak Setuju

Hasil perhitungan responden akan dihitung presentase kelayakan dengan menggunakan perhitungan :

$$\text{Hasil testing (100\%)} = \frac{\text{Skor penilaian (f)}}{\text{Skor tertinggi (n)}} \times 100\%$$

Presentase kelayakan yang diperoleh kemudian dibandingkan dengan table konversi yang berpedoman pada acuan konversi nilai :

Tabel 2. 4 Skala Konversi Nilai

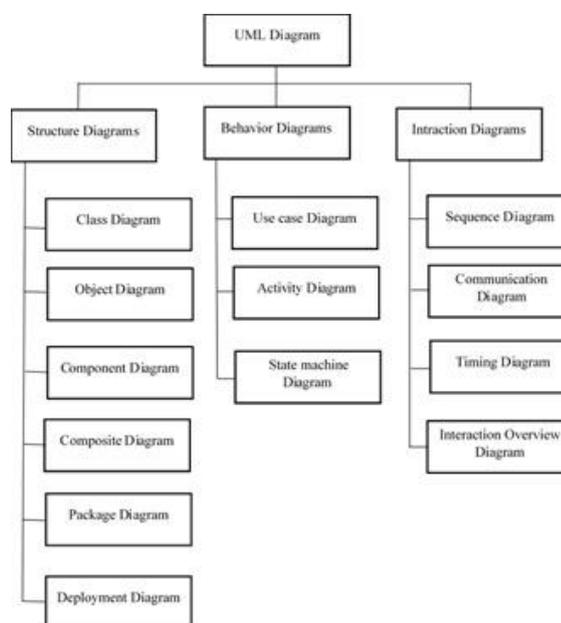
Presentase Pencapaian (%)	Interpretasi
$90 \leq x$	Sangat Baik
$80 \leq x < 90$	Baik
$70 \leq x < 80$	Cukup
$60 \leq x < 70$	Kurang

X < 60	Sangat Kurang
--------	---------------

Keterangan : x = presentase hasil pengujian

2.8 Unified Modeling Language (UML)

Pada UML 2.3 (Rosa A.s & M. Shalahuddin, 2019) terdiri dari 13 macam diagram yang dikelompokkan pada 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar dibawah ini.



Gambar 2. 2 Diagram UML (Rosa A.s & M. Shalahuddin, 2019)

Berikut ini penjelasan singkat dari pembagian kategori tersebut.

- *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
- *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.

- *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun *interaksi* antar *subsistem* pada suatu sistem.

Dari penjelasan diatas pada penelitian ini saya menggunakan 2(dua) diagram yaitu diagram *use case* diagram, *class* diagram. Berikut penjelasan dari masing-masing diagram :

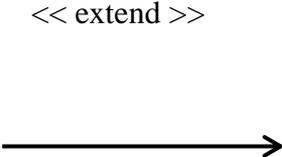
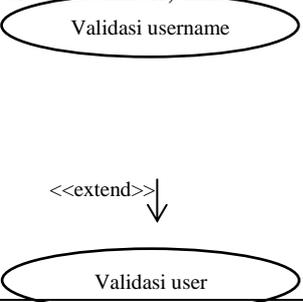
2.8.1 Use case Diagram

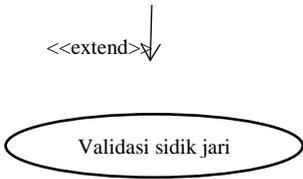
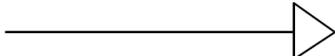
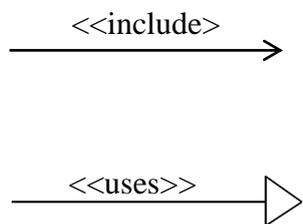
Use case atau diagram *use case* (Rosa A.s & M. Shalahuddin, 2019) pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antar satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa 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*.

- Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar 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.

Berikut adalah simbol yang ada pada diagram *use case* :

Tabel 2. 5 Simbol Use Case Diagram (Rosa A.s & M. Shalahuddin, 2019)

Simbol	Deskripsi
<p data-bbox="316 510 435 544"><i>Use case</i></p> 	<p data-bbox="853 510 1356 712">Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau actor, biasanya dinyatakan dengan kata kerja di awal frase nama <i>use case</i>.</p>
<p data-bbox="316 745 475 779">Aktor/<i>actor</i></p> 	<p data-bbox="853 745 1356 936">Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi</p> <p data-bbox="853 969 1356 1216">yang akan di itu sendiri, walaupun symbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
<p data-bbox="316 1249 515 1283"><i>Ektensi/extend</i></p> 	<p data-bbox="853 1249 1356 1619">Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu, mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misal</p> 

	 <p>Arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.</p>
<p>Generalisasi/<i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya,</p>
<p>Menggunakan/<i>include/uses</i></p> 	<p>Relasi <i>use case</i> tambahkan ke sebuah <i>use case</i> di mana <i>use case</i> ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsi atau sebagai syarat dijalankan <i>use case</i> ini</p> <p>Ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i> :</p> <ul style="list-style-type: none"> • <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahkan dijalankan. • <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan telah

	<p>dijalankan sebelum <i>use case</i> tambahan dijalankan.</p> <p>Ketika interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>
<p>Asosiasi /association</p> <p>_____</p>	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dan aktor</p>

2.8.2 Class Diagram

Diagram kelas atau *class diagram* (Rosa A.s & M. Shalahuddin, 2019) menggambarkan struktur sistem dari segi definisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

- Atribut merupakan *variabel-variabel* yang dimiliki suatu kelas
- Operasi atau metode adalah fungsi-fungsi yang dimiliki suatu kelas

Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

- Kelas main

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.
- Kelas yang menangani tampilan sistem (*view*)

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.
- Kelas yang diambil dari pendefinisian *use case* (*controller*)
- Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.

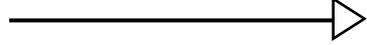
- Kelas yang diambil dari pendefinisian data(*model*)

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

Berikut adalah simbol yang ada pada diagram kelas :

Tabel 2. 6 Simbol *Class* Diagram (Rosa A.s & M. Shalahuddin, 2019)

Simbol	Deskripsi
<p>Kelas</p> 	<p>Kelas pada struktur sistem</p>
<p>Antarmuka/<i>Interface</i></p> 	<p>Sama dengan konsep <i>Interface</i> dalam pemrograman berorientasi objek.</p>
<p>Asosiasi /<i>association</i></p> 	<p>Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>.</p>
<p>Asosiasi berarah/<i>directed association</i></p> 	<p>Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>.</p>

Generalisasi 	Relasi antar kelas dengan makna <i>generalisasi-spesialisasi</i> (umum khusus).
Kebergunaan/ <i>dependency</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas.
Agregasi/ <i>aggregation</i> 	Relasi antar kelas dengan makna semua- bagian (<i>whole part</i>).

2.8.3 Activity Diagram

Diagram *activity* atau *activity* diagram (Rosa A.s & M. Shalahuddin, 2019) menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah system atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas system bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

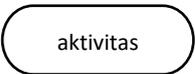
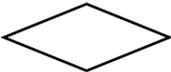
Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut :

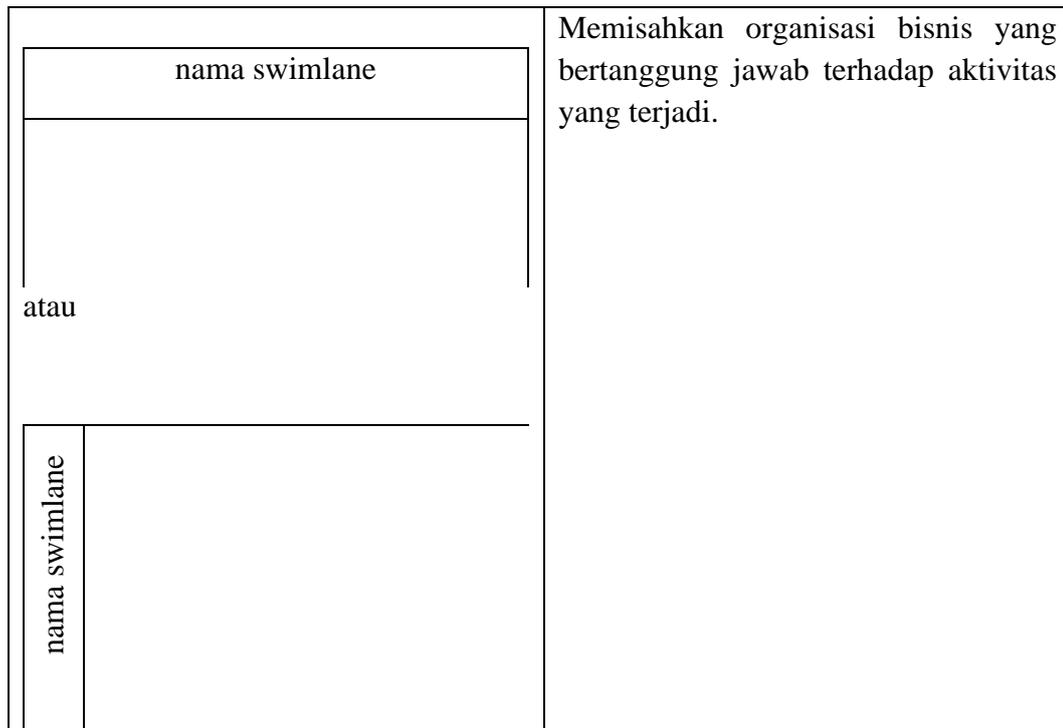
- Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
- Urutan atau pengelompokan tampilan dari sistem/*user Interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
- Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.

- Rancangan menu yang ditampilkan pada perangkat lunak.

Berikut adalah simbol-simbol yang ada pada diagram aktivitas :

Tabel 2. 7 Simbol *Activity Diagram* (Rosa A.s & M. Shalahuddin, 2019)

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

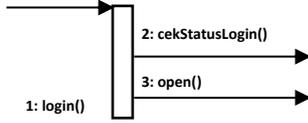


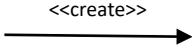
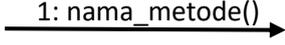
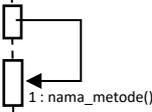
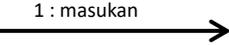
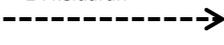
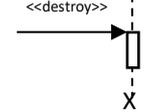
2.8.4 Sequence Diagram

Sequence diagram atau diagram sekuen (Rosa A.s & M. Shalahuddin, 2019) menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan *message* yang dikirimkan dan diterima antar objek. Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksinya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Berikut ini adalah simbol-simbol *Sequence diagram*, seperti yang terlihat pada table 2.7 di bawah ini :

Tabel 2. 8 Simbol *Sequence Diagram* (Rosa A.s & M. Shalahuddin, 2019)

Simbol	Deskripsi
<p>Aktor</p>  <p>Atau</p> <div style="border: 1px solid black; padding: 2px; width: fit-content;"> nama aktor . </div> <p>Tanpa waktu aktif</p>	<p>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; biasanya dinyatakan menggunakan kata benda di awal <i>frase</i> nama aktor.</p>
<p>garis hidup / <i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek.</p>
<p>objek</p> <div style="border: 1px solid black; padding: 2px; width: fit-content;"> nama objek : nama kelas </div>	<p>Menyatakan objek yang berinteraksi pesan.</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p>  <pre> sequenceDiagram actor Actor participant Object as participant Target as Actor->>Object: 1: login() activate Object Object->>Target: 2: cekStatusLogin() Object->>Target: 3: open() deactivate Object </pre> <p>Maka <code>cekStatusLogin()</code> dan <code>open()</code> dilakukan di dalam metode <code>login()</code></p> <p>Aktor tidak memiliki waktu aktif</p>

<p>pesan tipe create</p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.</p>
<p>Pesan tipe call</p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,</p>  <p>Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.</p>
<p>Pesan tipe send</p> 	<p>Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.</p>
<p>Pesan tipe return</p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.</p>
<p>Pesan tipe destroy</p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>.</p>

2.9 Alat Pengembangan WEB

2.9.1 XAMPP

Xampp (Sarwindah, 2018) suatu bundel *web server* yang populer digunakan untuk coba-coba di *windows* karena kemudahan instalisasinya. Xampp merupakan perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*) yang terdiri atas *Apache HTTP Server*, *Mysql* database dan penerjemahan bahasa yang ditulis dengan bahasa pemrograman PHP. Nama Xampp merupakan singkatan dari empat sistem operasi yaitu *Apache*, *Mysql*, *PHP*, dan *Perl*. Program ini tersedia dalam GNU dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis.

2.9.2 MySQL

MySQL (*My Structure Query Language*) (Wicaksono et al., 2021) *software* RDBMS yang dapat mengolah database dengan cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak user dan dapat melakukan suatu proses secara sinkron atau berbarengan. Dari pengertian diatas Penulis menyimpulkan bahwa MySQL adalah database yang digunakan oleh Pemrograman aplikasi yang sama dengan PHP yang isinya kode untuk menjalankan aplikasi yang akan dibuat.

2.9.3 Hyperteks Markup Language (HTML)

Hypertext Markup Language (Putra et al., 2019) suatu bahasa markah yang dipakai dalam membuat suatu tampilan web, berbagai informasi didalam sebuah penjelajahan Internet dan format *hiperteks* sederhana, ditulis dengan berkas ASCII

agar bisa menghasilkan tampilan yang terintegrasi. berkas yang dimuat dalam perangkat lunak pengolahan data dan disimpan kedalam sebuah format ASCII agar bisa dapat menjadi halaman sebuah web dengan perintah HTML. Berawal dari bahasa yang banyak dipakai sebelumnya didalam dunia percetakan dan juga dunia penerbitan yang biasa disebut dengan *Standard Generalized Markup Language* atau SGML, HTML menjadi sebuah standar yang dipakai secara luas dalam menampilkan halaman *web*.

2.9.4 Cascading Style Sheet (CSS)

Cascading Style Sheets atau bisa di sebut CSS (Febrianto et al., 2021) sekumpulan perintah untuk menghasilkan sebuah tampilan halaman *website*, CSS beroperasi melalui tag `<Style>` dengan adanya CSS pada HTML maka pengaturan warna dan tampilan akan menjadi lebih mudah, peran CSS sangatlah penting pada *website* tanpa adanya CSS, maka tampilan *website* akan dirasakan membosankan.

2.9.5 Hypertext Preprocessor (PHP)

Hypertext Preprocessor (Febrianto et al., 2021) atau yang biasa di sebut PHP merupakan bahasa pemrograman web yang mana *hypertext preprocessor* ini digunakan untuk memasukan sebuah alur logika dan biasanya sebagai penghubung dengan database MySQL agar data-data yang ada di *database* dapat di tampilkan pada halaman *web*.

2.9.6 JavaScript

Javascript (Febrianto et al., 2021) bahasa yang berbentuk kumpulan skrip yang fungsinya digunakan untuk menambahkan interaksi antara halaman *web* dengan pengguna halaman *web*. *Javascript* dijalankan pada sisi klien yang akan

memberikan kemampuan fitur-fitur tambahan halaman web yang lebih baik dibandingkan fitur-fitur yang terdapat pada HTML.

2.9.7 Framework CodeIgniter

Codeigniter (Febrianto et al., 2021) sebuah php *framework* yang berupa kumpulan *folder* dan *file* php, java script,css,txt dan file berbasis *web* lainnya dengan setting tertentu untuk menggunakannya dan menyediakan *library* dan *helper* yang dapat di manfaatkan di dalam pemrograman php.

2.9.8 Model View Control (MVC)

Konsep MVC (Febrianto et al., 2021) konsep pemisahan antara *logic* dengan tampilan dan *database*. Manfaat konsep ini adalah, membuat *coding logic* lebih *simple*, karena sudah di pisah dengan *code* untuk tampilan dan membuat *programmer* dapat bekerja secara terpisah dengan *designer*. *Programmer* mengerjakan *logic*, sedangkan *designer* berkeutatan dengan *design* dan tampilan.

1. *Model* merupakan *code* struktur data. *Model* berisi fungsi di dalam pengolahan *database*. *Script* Sql masuk di sini.
2. *View* merupakan *code* untuk menampilkan tampilan suatu program. Tampilan dapat berupa *web page*, *header*, *footer* dan apa saja yang berjenis tampilan.
3. *Controller* merupakan *code* untuk *logic*, algoritma dan sebagai penghubung antara *model*,*view*, dan sumber lain yang di perlukan untuk mengolah HTTP *request* dan *generate web page*.

1.9.10 Adobe Dreamweaver

Adobe Dreamweaver (Krismiaji, 2019) aplikasi desain dan pengembangan *web* yang mempunyai *editor visual* WYSIWYG (*what you see is what you get*).

Dreamweaver memiliki fitur *browser* yang terintegrasi untuk melihat halaman *web* yang dikembangkan di jendela pratinjau program sendiri agar konten memungkinkan untuk terbuka di *web browser* yang telah terinstal.

2.10 Uji Kelayakan Sistem Menggunakan TELOS

TELOS memberikan tinjauan komprehensif dan menyeluruh untuk membuat studi kelayakan secara umum, TELOS framework dapat memastikan proses pengambilan keputusan berbasis pada data (Prambayun dan Maharani, 2020). Kerangka kerja TELOS memiliki lima faktor kelayakan yang dapat dilihat pada tabel 2.9 :

Tabel 2. 9 Faktor Kelayakan TELOS

No	Kriteria
1	Kelayakan Teknik (<i>Technical Feasibility</i>)
2	Kelayakan Ekonomi (<i>Economic Feasibility</i>)
3	Kelayakan Hukum (<i>Legal</i>)
4	Kelayakan <i>Operational</i>
5	Kelayakan Jadwal (<i>Schedule</i>)