

BAB III

LANDASAN TEORI

3.1 Web

3.1.1 Pengertian

Web atau yang sering disebut *Word Wide Web* dan disingkat menjadi *www* adalah suatu metode untuk menampilkan informasi di internet, baik berupa teks, gambar, suara maupun video yang interaktif dan mempunyai kelebihan untuk menghubungkan (*link*) satu dokumen dengan dokumen lainnya (*hypertext*) yang dapat diakses melalui sebuah browser. Browser adalah perangkat lunak untuk mengakses halaman – halaman web, seperti Mozilla Firefox, Chrome, Opera, dan lain-lain.

Informasi yang disajikan melalui browser dibangun dengan Bahasa semi pemrograman HTML (*HyperText Markup Language*), dan kemudian ditingkatkan fungsinya dengan menyisipkan kode – kode Bahasa pemrograman web, seperti PHP, ASP, JSP, dan lain – lain, sehingga mampu menampilkan informasi yang lebih interaktif dan dinamis serta terhubung dengan database.

Beberapa istilah yang berhubungan dengan *www* sering membingungkan seorang pemula internet. Istilah – istilah tersebut adalah:

- *www*, adalah metode untuk menampilkan informasi di internet yang dapat diakses melalui sebuah browser.
- Halaman web (*web page*), adalah sekelompok informasi yang tampil pada *www*, informasi ditampilkan berdasarkan sebuah URL (*Uniform Resource Locator*).
- URL, adalah alamat lengkap sebuah informasi yang dapat diakses melalui browser, misalnya <http://www.ephi.web.id/images/ephi.jpg>. *http* adalah protocol untuk layanan web, *www* melambangkan layanan web, *ephi.web.id* adalah domain, *images* adalah nama folder, dan *ephi.jpg* adalah nama file
- Situs Web (*Website*) adalah kumpulan dari halaman web yang terdapat pada satu domain, yang terdiri dari dua atau lebih halaman web.

- *Homepage* adalah halaman awal sebuah website yang diakses berdasarkan domainnya. Misalnya anda mengakses web dengan alamat www.ephi.web.id, maka halaman yang tampil pertama kali disebut homepage, namun jika anda telah mengklik link – link yang lain pada halaman tersebut, maka yang anda buka adalah halaman webnya.
- Domain adalah nama unik yang dimiliki oleh setiap *website* untuk memudahkan manusia dalam mengingat sebuah website, misalnya ephi.web.id, lintai.net, rantaunet.org, dan lain – lain.

Seiring dengan perkembangan teknologi informasi yang begitu cepat, website juga mengalami perkembangan yang sangat pesat. Faktor utama yang membuat website begitu cepat berkembang adalah karena penyebaran informasi melalui website sangat cepat dan mencakup area yang luas (dunia), tidak dibatasi oleh jarak dan waktu, disamping itu, sekarang juga lagi trend pembuatan website – website pribadi atau *blogger*, disamping *e-banking*, *e-commerce*, *e-learning* dan lain sebagainya (Yuhefizar, 2008).

Teknologi web telah merubah cara organisasi dalam menjalankan bisnis dan komunikasi. Untuk pengembangan aplikasi, teknologi web telah menyajikan dunia baru dari rekayasa perangkat lunak (*software engineering*) dengan teknik – teknik baru, tool baru, lingkungan perancangan, dan penyebaran yang baru. Teknologi ini juga memungkinkan organisasi untuk menyampaikan aplikasi web dengan lebih mudah dan lebih cepat serta menyediakan metode yang lebih efisien untuk melakukan pemeliharaan. Sebagai hasilnya, organisasi lebih tanggap terhadap kebutuhan pengguna dan lebih cepat untuk menyesuaikan aplikasi kepada pengguna yang lebih spesifik (Simarmata, 2010).

3.2 Aplikasi Web

3.2.1 Pengertian

Dalam rekayasa perangkat lunak, suatu aplikasi web adalah suatu aplikasi yang diakses menggunakan penjelajah web melalui suatu jaringan seperti Internet atau intranet. Ia juga merupakan suatu aplikasi perangkat lunak komputer yang dikodekan dalam bahasa yang didukung penjelajah web (seperti

ASP, Perl, Java, Java Script, PHP, Python, Ruby, dll) dan bergantung pada penjelajah tersebut untuk menampilkan aplikasi.

Aplikasi web menjadi populer karena kemudahan tersedianya aplikasi klien untuk mengaksesnya, penjelajah web, yang kadang disebut sebagai suatu thin client (klien tipis). Kemampuan untuk memperbarui dan memelihara aplikasi web tanpa harus mendistribusikan dan menginstalasi perangkat lunak pada kemungkinan ribuan komputer klien merupakan alasan kunci popularitasnya. Aplikasi web yang umum misalnya webmail, toko ritel, lelang online, wiki, papan diskusi, weblog.

Banyak keuntungan yang diberikan oleh Aplikasi berbasis Web daripada aplikasi berbasis desktop, sehingga aplikasi berbasis web telah diadopsi oleh perusahaan sebagai bagian dari strategi teknologi informasinya, karena beberapa alasan:

1. Akses informasi mudah.
2. Setup server lebih mudah.
3. Informasi mudah didistribusikan.
4. Bebas platform, informasi dapat disajikan oleh browser web pada sistem operasi mana saja karena adanya standar dokumen berbagai tipe data dapat disajikan.

3.2.2 Usabilitas Aplikasi Web

Usabilitas adalah salah satu factor kualitas yang paling penting untuk aplikasi Web. Usabilitas yang kurang pada aplikasi akan menyebabkan pengguna tidak dapat mengoperasikannya, terutama pada aplikasi Web. Tujuan utama adalah untuk merancang aplikasi Web, kita harus mempertimbangkan konteks kebutuhan pengguna dan pemakaian umum. Selain itu, kita juga harus menentukan cara manusia memproses informasi, persepsi, memori, dan perhatian yang mempengaruhi interaksi.

Meskipun sejumlah petunjuk rancangan secara mendetail telah tersedia untuk mendukung rancangan aplikasi Web yang dapat digunakan, usabilitas tidak dapat dicapai “dalam satu tembakan”. Dengan kata lain, semua fase dari proses pengembangan harus berulang dan pengulangan ini harus dikontrol

dengan acuan seberapa baik tujuan khusus yang akan dicapai. Salah satu aktivitas utama untuk hal ini adalah pengevaluasian perkembangan usability, dengan menerapkan metode berbasis pengguna dan metode ahli. Aturan umum untuk aplikasi Web adalah bahwa pendekatan berbasis pengguna tradisional lebih sulit dilakukan daripada perangkat lunak konvensional. Oleh karena itu, banyak proyek menggunakan aplikasi lain berdasarkan model abstrak, termasuk analisis on-line dari tingkah laku pengguna yang tidak diketahui (Simarmata, 2010).

Menurut Herman dkk (2017) berpendapat aplikasi hibrid mengatasi kekurangan yang terdapat pada aplikasi berbasis web dengan menambahkan fitur – fitur yang disediakan oleh aplikasi native. Aplikasi hibrid dapat di deploy pada berbagai platform sekaligus, sehingga dapat meminimalisasi usaha dalam proses pembuatan aplikasi. Pada umumnya aplikasi hibrid dibuat dengan menggunakan Bahasa yang biasa digunakan untuk membuat aplikasi berbasis web, yaitu HTML, CSS, dan Javascript. Kelebihan aplikasi *hybrid* adalah:

1. Dapat di deploy pada berbagai jenis platform dengan source code sama tanpa melakukan proses porting.
2. Memiliki akses yang luas terhadap API platform, walaupun tidak seluas aplikasi native, namun masih lebih luas dibandingkan aplikasi berbasis web.

Sedangkan kekurangannya adalah:

1. Performa tidak sebaik aplikasi native.
2. Tampilan antarmuka pengguna terkadang memiliki perbedaan dengan tampilan native platform, sehingga dapat menimbulkan inkonsistensi pada tampilan antarmuka pengguna.

3.3 Metode Agile

3.3.1 Pengertian

Agile Development Methods adalah sekelompok metodologi pengembangan perangkat lunak yang didasarkan pada prinsip-prinsip yang sama atau pengembangan sistem jangka pendek yang memerlukan adaptasi cepat dari pengembang terhadap perubahan dalam bentuk apapun. *Agile development*

methods merupakan salah satu dari Metodologi pengembangan perangkat lunak yang digunakan dalam pengembangan perangkat lunak.

Menurut Raharjana (2017), “agile merupakan pendekatan yang iterative dan evolusioner yang dilakukan dengan mengedepankan kolaborasi serta menggunakan dokumen formal yang terbatas dan tepat untuk membangun perangkat lunak yang berkualitas dalam hal biaya yang efektif serta waktu yang sesuai dengan kebutuhan stakeholder yang bias berubah – ubah.

Menurut Suryantara (2017), metode agile merupakan metodolgi yang cepat. Beberapa metodologi agile yang berkembang saat ini antara lain:

- *eXtreme Programming (XP).*
- SCRUM.
- *Adaptive Software Development (ASD).*
- *Dynamic System Development Method (DSDM).*
- *Feature Driven Development (FDD).*
- *Crystal Light Methodology Family.*
- *Pragmatic Programming.*
- *Open Source Software Development (OSSD).*

3.3.2 Prinsip-prinsip Metode Agile

Agile Alliance (Pressman, 2010) mendefinisikan 12 prinsip untuk mencapai proses yang termasuk dalam agility, yaitu:

1. Prioritas utama adalah memuaskan klien dengan menghasilkan perangkat lunak secara cepat.
2. Menyambut perubahan kebutuhan walaupun terlambat dalam pengembangan perangkat lunak. Proses Agile memanfaatkan perubahan untuk keuntungan kompetitif klien.
3. Menghasilkan perangkat lunak yang bekerja secara rutin dari jangka waktu beberapa minggu sampai beberapa bulan dengan preferensi kepada jangka waktu yang lebih pendek.
4. Rekan bisnis dan pengembang perangkat lunak harus bekerja sama setiap hari dan sepanjang proyek.

5. Mengembangkan proyek di sekitar individual yang termotivasi. Memberi lingkungan dan dukungan yang dibutuhkan dan mempercayai untuk menyelesaikan pekerjaan dengan baik.
6. Metode yang paling efisien dan efektif untuk menyampaikan informasi dari dan dalam tim pengembangan perangkat lunak adalah dengan komunikasi secara langsung.
7. Perangkat lunak yang bekerja adalah ukuran utama kemajuan.
8. Proses Agile menggalakkan pengembangan berkelanjutan. Sponsor, pengembang, dan pengguna akan dapat mempertahankan kecepatan tetap secara berkelanjutan.
9. Perhatian yang berkesinambungan terhadap keunggulan teknis dan rancangan yang baik meningkatkan *Agility*.
10. Kesederhanaan adalah hal sangat yang amat penting.
11. Arsitektur, kebutuhan, dan rancangan perangkat lunak terbaik muncul dari tim yang dapat mengorganisir diri sendiri.
12. Secara berkala, tim pengembang berefleksi tentang bagaimana untuk menjadi lebih efektif, kemudian menyesuaikan dan menyelaraskan kebiasaan bekerjanya.

3.3.3 Metodologi Model eXtreme Programming

XP (eXtreme Programming) merupakan salah satu metodologi rekayasa perangkat lunak yang banyak digunakan untuk mengembangkan aplikasi oleh para developer. XP diperkenalkan oleh Kent Beck ketika ia ditunjuk untuk menangani sebuah proyek penggajian dari Chrysler yang dikenal dengan C3 (*Chrysler Comprehensive Compensation*). Proyek ini dimulai sekitar maret 1996. Proyek tersebut terancam gagal karena rumitnya sistem yang dibuat dan kegagalan pada saat memasuki tahap uji sistem (testing). Pihak Chrysler akhirnya menyewa Kent Beck sebagai konsultan di bidang software engineering. Kemudian ia dikenal sebagai pencetus XP. Kent Beck bersama rekannya, yaitu Ron Jeffries, diberi kewenangan oleh Chrysler untuk melakukan berbagai perubahan di proyek C3 untuk membuat proyek tersebut lebih efisien, lebih

fleksibel. Akhirnya Kent Beck dan Ron Jeffries berhasil menyelesaikan proyek C3 dengan menerapkan berbagai metode pada saat pengembangannya.

Kumpulan metode selama menangani proyek C3 kemudian dikenal sebagai metodologi *eXtreme Programming* (XP). XP memiliki kesan kumpulan ide lama yang sederhana, dan tidak ada efek apapun pada pengembangan aplikasi. Kent Beck mengakui dan menegaskan bahwa XP tidak selalu cocok (sesuai) untuk setiap proyek pengembangan perangkat lunak. Namun, XP memiliki kelebihan yaitu sesuai untuk proyek yang memiliki *dynamic requirements* atau proyek yang memiliki berbagai requirements yang tidak jelas dari klien.

XP sangat cocok untuk pengembangan proyek yang memerlukan adaptasi cepat dalam perubahan – perubahan yang terjadi selama pengembangan aplikasi. XP juga cocok untuk anggota tim yang tidak terlalu banyak dan berada pada lokasi yang sama dalam pengembangan sistem (Suryantara, 2017).

Aspek dasar yang perlu diperhatikan bila menggunakan XP dalam pengembangan aplikasi. Nilai – nilai aspek dasar tersebut adalah:

1. The Planning Game

Pada tahap ini XP mirip dengan metodologi RAD (*Rapid Application Development*). RAD menggunakan proses yang pendek dan cepat, mengutamakan aspek teknik. Proses ini juga menentukan requirements dibutuhkan penggunaan teknik score card.

2. Small Releases

Developer akan menyelesaikan setiap bagian dari perangkat lunak, kemudian dipresentasikan kepada perusahaan (klien). Apabila disetujui maka bagian tersebut diterapkan ke seluruh aplikasi, kemudian dilakukan tes awal pada penerapan aplikasi tersebut.

3. Metaphor

Metaphor menggambarkan visi, tujuan pengembangan perangkat lunak secara luas. Metaphor juga bertujuan menjaga komunikasi antara klien dengan *developer* agar tetap baik.

4. **Simple Design**

XP di desain sesederhana mungkin untuk menghindari pengembangan sistem yang sulit dimengerti. Bila terjadi perubahan pada desain, maka resiko kegagalan desain dapat diminimalkan.

5. **Refactoring**

Refactoring merupakan suatu aspek khusus pada metodologi XP. Proses ini mengubah kode program dari perangkat lunak dengan tujuan meningkatkan kualitas dari struktur program. Sehingga *developer* tidak harus mengulang proses desain.

6. **Testing**

Tim developer harus membuat tes yang akan digunakan untuk menjalankan software dan mengembangkannya. Pada saat proses coding selesai, software tersebut akan diuji menggunakan model tes yang telah dibuat sebelumnya.

7. **Pair Programming**

Merupakan aspek dimana dua orang programmer bersama – sama dalam satu tempat atau komputer, mereka saling berinteraksi dan berdiskusi tentang permasalahan dan penyelesaiannya.

8. **Collective Ownership**

Antar programmer saling berbagi informasi sehingga tidak ada ketergantungan terhadap satu orang programmer.

9. **Coding Standart**

Antar programmer harus sepakat dalam memahami script/program. Sehingga programmer yang terlibat dalam pengembangan proyek memiliki pemahaman yang sama.

10. **Continues Integration**

Developer diwajibkan untuk melakukan build agar kesalahan pada program dapat dideksi dan diperbaiki secara secepat mungkin. Waktu maksimal yang dibutuhkan untuk build dalam XP adalah satu hari.

11. 40-Hours Week

Menurut pandangan Kent Beck, waktu ideal seorang programmer bekerja dalam satu hari adalah 8 jam dan 5 hari dalam satu minggu.

12. On-side Customer

Bila menggunakan XP dalam pengembangan sistem maka sebaiknya melibatkan klien dalam proses build dan tes yang dilakukan. Klien maka dapat memberikan masukan dan koreksi atas aplikasi yang dibuat sehingga aplikasi yang dibuat sesuai dengan kebutuhan klien.

3.3.4 Keuntungan dan Kerugian eXtreme Programming

Setiap metodologi memiliki keuntungan dan kerugian. XP juga memiliki keuntungan dan kerugian. Adapun keuntungan dan kerugian dalam metodologi XP adalah:

Keuntungan:

- ✓ Dalam hal ini XP menjalin komunikasi/interaksi yang baik dengan pelanggan (klien) pada pengembangan aplikasi.
- ✓ Saling menghargai antar – developer dan meningkatkan komunikasi.
- ✓ Dapat menjadi pembelajaran bagi orang lain (klien).
- ✓ Klien mendapatkan umpan balik yang akurat mengenai aplikasi yang dibuat.
- ✓ Dengan XP dapat mengubah pemikiran pelanggan terhadap aplikasi yang dibuat.
- ✓ Developer tidak berkerja secara berlebihan, karena XP memfokuskan pada aplikasi/program yang dibuat.
- ✓ Dengan XP dapat membuat keputusan – keputusan yang bersifat teknikal

Kerugian:

- Perubahan selalu diterima sehingga developer harus selalu siap dengan perubahan – perubahan dari klien.
- Di awal – awal pengembangan sistem tidak bias membuat code (program) yang detail (prinsip *simplicity*).

- XP hanya mengerjakan satu proyek dan dijalankan oleh satu tim. Tidak efektif mengerjakan proyek yang berbeda secara bersamaan (paralel).
- XP tidak dapat dikerjakan bila developer saling terpisah jauh dengan klien, karena mengakibatkan terhambatnya komunikasi antara developer dengan klien.
- XP belum terbukti dapat bekerja pada sistem yang mempunyai masalah skalabilitas. Skalabilitas adalah tingkat perencanaan kerja yang diperlukan berdasarkan ukuran proyek, kompleksitas proyek, dan ukuran tim.

3.4 Cloud Computing

3.4.1 Pengertian

Cloud computing merupakan evolusi yang mengadopsi virtualization, service – oriented architecture and utility computing. Cloud computing memungkinkan konsumen teknologi untuk memikirkan komputasi secara efektif dengan biaya minimal dan dapat diandalkan. Hal lain yang juga tidak perlu lagi dikhawatirkan oleh pengguna adalah tentang bagaimana membangunnya, cara kerjanya, siapa yang mengoperasikan atau di mana harus meletakkannya.

Virtualization adalah pencipta versi virtual (bukan actual) terhadap sumber daya teknologi informasi, seperti sistem operasi, server perangkat penyimpanan (storage) atau sumber daya jaringan. Virtualisasi dapat dilihat sebagai bagian dari tren secara menyeluruh dari *enterprise* TI yang mencakup autonomic computing. Autonomic computing merupakan sebuah scenario di mana lingkungan TI akan mampu melakukan pengelolaan sendiri (self management) didasarkan pada aktivitas yang dirasakan dan utility computing. Kekuatan pemrosesan computer dianggap sebagai sebuah utilitas yang memungkinkan klien membayar sesuai yang diperlukan.

Dalam perspektif teknologi komunikasi sendiri, cloud computing atau komputasi cloud dapat diartikan sebagai suatu teknologi yang memanfaatkan internet resource untuk komputasi yang dapat direquest oleh pengguna dan merupakan sebuah layanan dengan pusat server bersifat virtual atau berada dalam cloud (internet) itu sendiri. Secara sederhana cloud computing dapat

didefinisikan sebagai “layanan teknologi informasi yang bias dimanfaatkan atau diakses oleh pelanggannya melalui jaringan internet”.

3.4.2 Cara Kerja Cloud

Cloud computing bekerja secara *client – server*, menggunakan *protocol* web *browser*. Cloud menyediakan server berbasis aplikasi dan semua layanan data kepada pengguna, dengan output ditampilkan pada perangkat klien. Jika pengguna ingin membuat dokumen menggunakan pengolah data, misalnya cloud menyediakan aplikasi yang cocok yang berjalan pada server yang menampilkan pekerjaan yang dilakukan oleh pengguna pada layar web browser klien.

Memori yang dialokasikan untuk web browser sistem klien digunakan untuk membuat data aplikasi muncul di layar sistem klien, tetapi semua perhitungan dan perubahan dicatat oleh server, dan hasil akhir termasuk file yang dibuat atau diubah secara permanen disimpan pada server cloud. Kinerja dari aplikasi cloud tergantung pada kecepatan akses jaringan, dan kehandalan serta kecepatan pemrosesan perangkat klien. Sejak layanan cloud berbasis web, bekerja pada platform, termasuk linux, Macintosh, dan computer Windows. Ponsel smart, bantalan dan perangkat tablet dengan internet dan mengakses *World Wide Web* juga menyediakan layanan cloud untuk *telecommuting* dan pengguna ponsel. Sebuah penyedia layanan mungkin mempunyai kekuatan pemrosesan dari beberapa computer remote dalam cloud untuk mencapai tugas – tugas rutin seperti back up sejumlah besar data, pengolah data, atau pekerjaan komputasi secara intensif. Tugas – tugas ini biasanya mungkin sulit, memakan waktu, atau mahal untuk pengguna individu atau perusahaan kecil untuk menyelesaikan, terutama dengan sumber daya komputasi yang terbatas dan dana. Dengan komputasi cloud, klien hanya memerlukan komputer sederhana, seperti netbook, dirancang dengan komputasi *cloud* dalam pikiran, atau bahkan smartphone, dengan koneksi ke Internet, atau jaringan perusahaan, dalam rangka untuk membuat permintaan data dari cloud, maka istilah “perangkat lunak sebagai layanan” (SaaS). Perhitungan dan penyimpanan dibagi antara computer remote untuk menangani volume besar dari kedua, sehingga klien tidak perlu

membeli perangkat keras mahal atau perangkat lunak untuk menangani tugas. Hasil dari tugas pengolahan dikembalikan ke klien melalui jaringan, tergantung pada kecepatan koneksi internet.

3.4.3 Syarat Cloud Computing

1. **On-Demand Self-Services**

Sebuah layanan cloud computing harus dapat dimanfaatkan oleh pengguna melalui mekanisme swalayan dan langsung tersedia pada saat dibutuhkan. Campur tangan penyedia layanan sangat minim.

2. **Broadband Network Access**

Sebuah layanan cloud computing harus dapat diakses dari mana saja, kapan saja, dengan apa pun, asalkan kita terhubung ke jaringan layanan atau internet.

3. **Resource Pooling**

Sebuah layanan cloud computing harus tersedia secara terpusat dan dapat membagi sumber daya secara efisien. Karena cloud computing digunakan bersama – sama oleh berbagai pelanggan, penyedia layanan harus dapat membagi beban secara efisien, sehingga sistem dapat dimanfaatkan secara maksimal.

4. **Rapid Elasticity**

Sebuah layanan cloud computing harus dapat menaikkan atau menurunkan kapasitas sesuai kebutuhan. Misalnya, sebuah website berita dalam jaringan cloud computing, maka apabila terjadi peningkatan traffic karena ada berita penting, maka kapasitas harus dapat dinaikkan dengan cepat.

5. **Measured Service**

Sebuah layanan *cloud computing* harus disediakan secara terukur, karena nantinya akan digunakan dalam proses pembayaran. Layanan cloud computing dibayar sesuai penggunaan sehingga harus terukur dengan baik (Jamil et al, 2016).

3.4.4 Karakteristik Cloud Computing

Menurut Ferdiana (2016), karakteristik *Cloud Computing* didefinisikan dengan kategori sebagai berikut:

1. **Software as a Service (SaaS)**

Layanan aplikasi berbasis cloud yang diakses melalui internet. SaaS dirikan dengan model berlangganan sebuah layanan aplikasi tertentu untuk memperoleh manfaatnya. Office 365, OneDrive, dan CRM Online adalah contoh layanan SaaS. Analogi SaaS layaknya tinggal di hotel, kemudian diberi berbagai fasilitas yang sudah ditetapkan. Pada kategori ini, penggalan berinteraksi dengan sumber daya secara tidak langsung melalui aplikasi. Contohnya pengguna OneDrive tidak bias melakukan konfigurasi secara detail akan kebutuhan ruang penyimpanannya. Yang dilakukan adalah berlangganan paket dengan ruang penyimpanan yang dibutuhkan.

2. **Platform as a Service (PaaS)**

PaaS adalah layanan solusi berbasis cloud yang diakses melalui internet. PaaS memberikan solusi untuk semua kebutuhan, mulai dari infrastruktur hingga perangkat pengembangan. Microsoft Azure adalah salah satu contoh dari PaaS. Analogi PaaS layaknya mengontrak rumah beserta isinya dan juga diizinkan memodifikasi untuk kebutuhan tertentu. Pada kategori ini, pelanggan berinteraksi langsung dengan sumber daya melalui panel khusus. Contohnya pengguna azure menggunakan Azure portal untuk mengelolanya.

3. **Infrastructure as a Services (IaaS)**

IaaS adalah layanan sewa infrastruktur layaknya hosting konvensional. Perbedaannya dengan hosting konvensional adalah fleksibilitas untuk alokasi infrastruktur tanpa melibatkan pihak pengelola. Microsoft Azure juga memiliki layanan IaaS bersamaan dengan solusi PaaS-nya. Analogi IaaS layaknya menyewa tanah untuk kemudian dibangun rumah sesuai dengan kebutuhannya. Pada kategori ini, pelanggan berinteraksi layaknya pengguna PaaS.

Cloud Computing sendiri dalam pelaksanaannya terbagi menjadi tiga jenis implementasi, yakni:

- *Public Cloud*. *Cloud Computing* diimplementasikan oleh organisasi dengan berlangganan melalui pihak penyedia *Cloud Computing*, seperti Microsoft Azure, Amazon, dan yang lain. Skenario pelaksanaan ini cocok bagi organisasi yang hendak mengadopsi Cloud secara cepat.
- *Private Cloud*. *Cloud Computing* diimplementasikan oleh organisasi dengan menyiapkan infrastruktur dan layanannya sendiri untuk kemudian digunakan bagi kebutuhan organisasi. Skenario pelaksanaan ini sangat cocok bagi organisasi besar yang ingin secara penuh mengontrol sumber dayanya secara terpusat dan memenuhi aspek Cloud. *Microsoft System Center* dan *VMWare* adalah beberapa contoh penyedia solusi *private cloud*.
- *Hybrid Cloud*. Merupakan solusi yang mengintegrasikan antara *Private Cloud/On-Premise* dengan *Public Cloud*. Hal ini ditandai dengan organisasi menggunakan dua layanan. Hal ini menjamin organisasi untuk memiliki control dan juga memiliki fleksibilitas sumber daya. Berbagai penyedia *Cloud Computing* seperti Microsoft dan Amazon dapat mengintegrasikan antara kedua *Cloud* tersebut.

3.5 Pengujian Alpha dan Beta

Apabila Perangkat Lunak dibuat untuk pelanggan maka dapat dilakukan acceptance test sehingga memungkinkan pelanggan untuk memvalidasi seluruh keperluan. Pengujian dilakukan karena memungkinkan pelanggan menemukan kesalahan yang lebih rinci dan membiasakan pelanggan memahami perangkat lunak yang telah dibuat.

3.5.1 Pengujian Alpha

Dilakukan pada sisi pengembang oleh seorang pelanggan. Pengujian ini dilakukan terhadap perangkat lunak untuk memastikan bahwa perangkat lunak dapat berjalan dengan benar sesuai dengan kebutuhan dan tujuan yang

diharapkan. Pengujian Alpha berfokus pada persyaratan fungsionalitas dari perangkat lunak (Sidiq, 2013).

3.5.2 Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan secara objektif, dimana pengujian dilakukan secara langsung terhadap pengguna, dengan memberikan kuisioner mengenai tanggapan pengguna atas perangkat lunak yang telah dibangun. Pengujian beta dilakukan terhadap usability sistem (Sidiq, 2013).

Definisi usability menurut ISO 9241:11 (1998) adalah sejauh mana suatu produk dapat digunakan oleh pengguna tertentu untuk mencapai target yang ditetapkan dengan efektivitas, efisiensi dan mencapai kepuasan penggunaan dalam konteks tertentu. Konteks penggunaan terdiri dari pengguna, tugas, peralatan (hardware, software dan material). Berdasarkan definisi tersebut usability diukur berdasarkan komponen:

1. Efisiensi (*efficiency*): bagaimana pengguna dapat mencapai sebuah tujuan yang diinginkan.
2. Efektifitas (*effectiveness*): sumber daya yang digunakan dalam mencapai tujuan dan hasil yang telah dicapai.
3. Kepuasan (*satisfaction*): sejauh mana penggunaan merasakan kepuasan terhadap suatu produk yang telah digunakan.

3.5.3 Rancangan Variabel dan indikator

Menurut (Putri Rosalinda, 2018) Untuk mendapatkan data kuisioner yang baik, diperlukan pengukuran terhadap variabel yang telah disusun sebelumnya. Untuk melakukan pengukuran tersebut diperlukan sebuah indikator untuk memperjelas variabel tersebut. Variabel yang digunakan dalam pembuatan kuisioner yaitu *efficiency*, *effectiveness*, dan *satisfaction*.

Tabel 3.1 Variabel dan Indikator

Variabel	Indikator
<i>efficiency</i> (Efisiensi)	Waktu dalam menyelesaikan <i>task</i>
	<i>Time to learn</i>
	Waktu yang dibutuhkan untuk mengkoreksi error
<i>effectivenees</i> (Efektifitas)	persentase pencapaian yang telah dicapai
	kesesuaian dari jumlah fitur yang ada dalam sistem aplikasi pergi bareng
	presentase fungsi yang telah dipelajari
	presentase kesalahan yang berhasil dikoreksi
<i>Satisfaction</i> (Kepuasan)	Skala penilaian untuk kepuasan User Interface
	Skala penilaian dengan kekuatan fitur yang diberikan dalam sistem aplikasi pergi bareng
	Skala penilaian penggunaan untuk kemudahan belajar
	Skala penilaian untuk penanganan error pada sistem aplikasi pergi bareng

3.5.4 Skala Likert

Skala *Likert* adalah suatu skala psikometrik yang umum digunakan dalam kuesioner. Skala Likert merupakan skala yang paling banyak digunakan dalam riset berupa survey seperti mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial. Dengan Skala Likert, variabel yang akan diukur dijabarkan menjadi indikator variabel. Kemudian indikator tersebut dijadikan sebagai titik tolak untuk menyusun item-item instrumen yang dapat berupa pertanyaan atau pernyataan. Skala Likert dirancang untuk meyakinkan responden menjawab dalam berbagai tingkatan pada setiap butir pertanyaan atau pernyataan yang terdapat dalam kuesioner. Data tentang dimensi dari variabel-variabel yang dianalisis dalam penelitian ini yang ditujukan kepada responden menggunakan skala 1 sampai dengan 5 untuk mendapatkan data yang bersifat *original* dan diberi skor seperti pada tabel 3.2.

Tabel 3.2 Nilai Kuisisioner

PK	STS	TS	N	S	SS
Nilai	1	2	3	4	5

Keterangan:

PK = Pertanyaan Kuisisioner

STS = Sangat Tidak Setuju

TS = Tidak Setuju

N = Netral

S = Setuju

SS = Sangat Setuju