

BAB II

TINJAUAN PUSTAKA

2.1 Pengertian *Website*

Website atau disingkat *web*, dapat diartikan sekumpulan halaman yang terdiri atas beberapa laman yang berisi informasi dalam bentuk data digital, baik berupa teks, gambar, video, audio dan animasi lainnya yang disediakan melalui jalur koneksi internet. Lebih jelasnya, *website* merupakan halaman-halaman yang berisi informasi yang ditampilkan oleh browser, seperti Mozilla Firefox, Google Chrome atau yang lainnya, sedangkan internet adalah jaringan yang digunakan untuk mengirim informasi pada *website* [1].

Dalam perkebangannya hingga era saat ini, teknologi berbasis *web* telah memiliki kurang lebih tiga *phase* perkembangan/generasi yakni dimulai dari *Web 1.0*, *Web 2.0*, *Web 3.0/web semantic* [2].

1. *Web 1.0*

Merupakan teknologi *Web* generasi pertama yang merupakan revolusi baru di dunia Internet karena telah mengubah cara kerja dunia industri dan media. Secara umum, *Web 1.0* dikembangkan untuk pengaksesan informasi dan memiliki sifat yang sedikit interaktif. Secara garis besar, sifat *Web 1.0* adalah *Read*.

2. *Web 2.0*

Istilah *Web 2.0* pertama kalinya diperkenalkan oleh O'Reilly Media pada tahun 2004 sebagai teknologi *Web* generasi kedua yang mengedepankan kolaborasi dan *sharing* informasi secara daring. Sifat *web 2.0* adalah *Read-Write*. Yang menjadi kunci perbedaan dalam *Web 2.0* dan *Web 1.0* adalah keterbatasan pada *Web 1.0* yang mengharuskan pengguna internet untuk datang ke dalam *website* tersebut dan melihat satu persatu konten di dalamnya. Sedangkan *Web 2.0* memungkinkan pengguna internet dapat melihat konten suatu *website* tanpa harus berkunjung ke alamat situs yang bersangkutan. Kemampuan *web 2.0* dalam melakukan aktivitas *drag and drop*, *auto complete*, *chat*, *voice* dan dapat dilakukan layaknya aplikasi desktop.

3. *Web 3.0 / Web Semantic*

Web 3.0 adalah sekumpulan teknologi yang menawarkan cara baru yang efisien dalam membantu komputer mengorganisasi dan menarik kesimpulan dari data daring. *Web 3.0* menawarkan metode yang efisien dalam membantu komputer mengorganisasi dan menarik kesimpulan dari data daring. Konsep ini dapat diandaikan sebuah *website* sebagai sebuah intelektualitas buatan (*Artificial Intelligence*).

2.2 Bahasa Pemrograman *Website*

Bahasa pemrograman merupakan bahasa yang dapat dipahami oleh komputer. Ada banyak bahasa pemrograman yang memiliki fungsi berbeda-beda, diantaranya bahasa pemrograman untuk membuat aplikasi desktop, membuat game, membuat aplikasi *web*, membuat aplikasi *handphone*, dan sebagainya. Dalam perkembangan *website*, ada banyak jenis bahasa pemrograman yang dapat digunakan, beberapa diantaranya adalah HTML, CSS, Javascript dan PHP [1].

2.2.1 HTML

HTML singkatan dari *Hyper Text Markup Language*, yaitu skrip yang berupa tag-tag untuk membuat dan mengatur struktur *website* [1]. Disebut *hypertext* karena didalam HTML sebuah *text* biasa dapat berfungsi lain, kita dapat membuatnya menjadi *link* yang dapat berpindah dari satu halaman ke halaman lainnya hanya dengan meng-klik *text* tersebut. Kemampuan *text* inilah yang dinamakan *Hyper Text*, walaupun pada implementasinya nanti tidak hanya *text* yang dapat dijadikan *link*. Disebut *Markup Language* karena bahasa HTML menggunakan tanda (*mark*), untuk menandai bagian-bagian dari *text*. Misalnya, *text* yang berada di antara tanda tertentu akan menjadi tebal, dan jika berada di antara tanda lainnya akan tampak besar. Tanda ini kenal sebagai HTML tag [2]. Beberapa tugas utama HTML dalam membangun *website*, diantaranya sebagai berikut [1]:

1. Menentukan *layout website*.
2. Memformat teks dasar, seperti pengaturan paragraf dan format font.

3. Membuat *list*.
4. Membuat tabel.
5. Menyisipkan gambar, video, dan audio.
6. Membuat *link*.
7. Membuat formulir.

2.2.2 CSS

CSS singkatan dari *Cascading Style Sheets*, yaitu skrip yang digunakan untuk mengatur desain *website*. Walaupun HTML mempunyai kemampuan untuk mengatur tampilan *website*, namun kemampuannya sangat terbatas [1]. Fungsi utama dari css ialah mendesain, merancang, merubah, dan juga membentuk halaman pada *website* ataupun *blog* dan isi dari halaman *website* itu ialah tag-tag html, logikanya css itu bisa mengubah tag-tag html yang sederhana sehingga bisa menjadi lebih fungsional dan juga menarik [3].

2.2.3 PHP

PHP singkatan dari *Hypertext Preprocessor* yang merupakan *server-side programming*, yaitu bahasa pemrograman yang diproses di sisi *server*. Fungsi utama PHP dalam membangun *website* adalah untuk melakukan pengolahan data pada *database*. Data *website* akan dimasukkan ke *database*, diedit, dihapus dan ditampilkan pada *website* yang diatur oleh PHP [1].

Dibandingkan bahasa pemrograman lainnya PHP memiliki kelebihan lainnya. Hal itu terbukti dari adanya standar yang menjadikan PHP menjadi dasar bagi programmer *web* diseluruh dunia. Dimana *market share* penggunaan bahasa pemrograman *server side* untuk mayoritas *website* diseluruh dunia dipegang PHP dibandingkan bahasa pemrograman lainnya. Salah satu kelebihan PHP adalah tidak melakukan kompilasi dalam penggunaannya. Selain itu PHP juga dapat mudah ditemukan mulai dari apache, IIS, Lighttpd hingga Xitami dengan konfigurasi yang relatif mudah. Kelebihan lainnya dari PHP adalah pengembangan yang lebih mudah. Hal itu dikarenakan terdapat banyak milis-milis dan *developer* yang membantu dalam proses pengembangan. Selanjutnya, PHP

sebagai bahasa *open source* dapat digunakan dalam berbagai mesin baik Linux, Windows, Unix/ Macintosh serta dapat dijalankan secara *runtime* melalui *console* [4].

2.2.4 Javascript

Berbeda dengan php yang diproses di sisi *server*, javascript diproses pada komputer klien. Karena pemrosesannya dilakukan di komputer klien, membuat javascript lebih interaktif dibanding php. Peran javascript dalam membuat *website* adalah memberikan efek animasi yang menarik dan interaktivitas dalam penanganan *event* yang dilakukan oleh pengguna *website* [1].

2.3 Software yang dibutuhkan

Dalam proses pembuatan *website*, ada beberapa macam *software* yang dibutuhkan, yaitu sebagai berikut :

1. Sublime Text

Sublime Text Editor adalah *text editor* yang di buat untuk mempermudah pekerjaan programmer membantu programmer dalam mengerjakan tugasnya. Sublime merupakan *text editor* yang di gunakan untuk banyak sekali bahasa pemrograman dan bahasa markup. Sublime *text editor* juga mendukung penambahan *plugin*, banyak *plugin* yang telah tersedia yang bisa kita gunakan sesuai keinginan dan kebutuhan. Sublime di bangun dengan menggunakan python dan hingga saat ini sublime telah mencapai versi 3 [5].

2. Paket Apache dan Phpmyadmin

Apache berfungsi sebagai *web server*, yaitu tempat menyimpan file-file php dan file-file lainnya yang diperlukan *website*, sedangkan Phpmyadmin merupakan aplikasi berbasis *web* yang digunakan untuk membuat *database* MySQL sebagai tempat untuk menyimpan data-data *website*. Keduanya biasanya sudah disediakan dalam satu paket aplikasi seperti Appserv atau Xampp. Xampp adalah salah satu paket installer yang berisi Apache yang merupakan *web server* tempat menyimpan file-file yang diperlukan *website*, dan Phpmyadmin sebagai aplikasi yang digunakan untuk perancangan *database* MySQL [1].

3. Laravel

Laravel adalah sebuah PHP *Framework* yang dirancang untuk pengembang yang membutuhkan sebuah *toolkit* yang sederhana dan elegan untuk menciptakan aplikasi *web* berfitur lengkap. Laravel diciptakan oleh Taylor Otwell pada tahun 2011 [6]. Laravel dirilis dibawah lisensi MIT dengan kode sumber yang sudah disediakan oleh Github, sama seperti *framework-framework* yang lain, Laravel dibangun dengan konsep MVC (*Model-View-Controller*), kemudian Laravel dilengkapi juga *command line tool* yang bernama "Artisan" yang bisa digunakan untuk *packaging bundle* dan instalasi *bundle* melalui *command prompt* [7]. Laravel merupakan *software open source* yang memiliki fitur-fitur untuk membantu pengembangan *web*. Selain itu, *website* yang dibangun dengan menggunakan Laravel ini juga aman dari berbagai serangan yang terjadi pada situs *web*. Lalu, *script code* yang dibuat pada Laravel ini juga sangat rapih, karena setiap *script code* yang dibuat akan dipisah sesuai dengan fungsinya masing-masing, seperti *script code* untuk *front end* akan ditempatkan pada *blade engine template* dan untuk *back end* akan ditempatkan pada *controller* [6]. Beberapa fitur yang dimiliki oleh *framework* Laravel adalah sebagai berikut [7] :

- a. *Bundles* yaitu sebuah fitur dengan sistem pengemasan modular dan berbagai *bundle* telah tersedia untuk digunakan dalam aplikasi Anda.
- b. *Eloquent ORM* merupakan penerapan PHP lanjutan dari pola "*active record*" menyediakan metode internal untuk mengatasi kendala hubungan antara objek *database*. Pembangunan *query* Laravel Fluent didukung *Eloquent*.
- c. *Application Logic* merupakan bagian dari aplikasi yang dikembangkan baik menggunakan *Controllers* maupun sebagai bagian dari deklarasi *Route*. Sintaks yang digunakan untuk mendefinisikannya mirip dengan yang digunakan oleh *framework Sinatra*.
- d. *Reverse Routing*, mendefinisikan hubungan antara *Link* dan *Route*, sehingga jika suatu saat ada perubahan pada *route* secara otomatis akan tersambung dengan *link* yang relevan. Ketika *Link* yang dibuat dengan menggunakan nama-nama dari *Route* yang ada, secara otomatis Laravel akan membuat URI yang sesuai.

- e. *Restful Controllers*, memberikan sebuah *option* (pilihan) untuk memisahkan logika dalam melayani HTTP GET dan permintaan POST.
- f. *Class Auto Loading*, menyediakan otomatis *loading* untuk *class-class* PHP, tanpa membutuhkan pemeriksaan manual terhadap jalur masuknya. Fitur ini mencegah *loading* yang tidak perlu.
- g. *View Composers* adalah kode unit *logical* yang dapat dijalankan ketika sebuah *View* di *load*.
- h. *IoC Container* memungkinkan untuk objek baru yang dihasilkan dengan mengikuti prinsip control pembalik, dengan pilihan contoh dan referensi dari objek baru sebagai Singletons.
- i. *Migrations* menyediakan versi sistem *control* untuk skema *database*, sehingga memungkinkan untuk menghubungkan perubahan pada basis kode aplikasi dan keperluan yang dibutuhkan dalam merubah tata letak *database*, mempermudah dalam penempatan dan memperbarui aplikasi.
- j. Unit Testing, mempunyai peran penting dalam *framework* Laravel, dimana unit testing ini mempunyai banyak tes untuk mendeteksi dan mencegah regresi. Unit testing dapat dijalankan melalui fitur "*artisan command-line*".
- k. *Automatic pagination* menyederhanakan tugas dari penerapan halaman, menggantikan penerapan yang manual dengan metode otomatis yang terintegrasi ke Laravel.

2.4 Masjid

2.4.1 Pengertian Masjid

Masyarakat menyebut masjid adalah rumah Allah *subhanahu wa ta'ala* yang difungsikan untuk menunaikan shalat. Dari segi bahasa, kata masjid terambil dari akar kata *sajada-sujud*, yang berarti *patuh*, *taat*, serta *tunduk* dengan penuh hormat dan *ta'dzim*. Dalam pengertian sehari-hari, masjid merupakan bangunan tempat shalat kaum Muslim. Tetapi, karena akar katanya mengandung makna *tunduk* dan *patuh*, maka hakikat

masjid adalah tempat melakukan segala aktivitas yang mengandung kepatuhan kepada Allah semata [8].

2.4.2 Fungsi Masjid dan Kegiatan-Kegiatan yang Biasa Diadakan di Masjid

Selain tempat ibadah masjid juga merupakan pusat kehidupan komunitas muslim. Kegiatan-kegiatan perayaan hari besar, diskusi, kajian agama, ceramah dan belajar Al Qur'an sering dilaksanakan di Masjid. Bahkan dalam sejarah Islam, masjid turut memegang peranan dalam aktivitas sosial kemasyarakatan hingga kemiliteran. Berikut ini adalah beberapa fungsi masjid yaitu [9]:

1. Fungsi Keagamaan

Semua muslim yang telah baligh atau dewasa harus menunaikan shalat lima kali sehari. Walaupun beberapa masjid hanya dibuka pada hari Jumat, tetapi masjid yang lainnya menjadi tempat shalat sehari-hari. Pada hari Jumat, semua muslim laki-laki yang telah dewasa diharuskan pergi ke masjid untuk menunaikan shalat ke masjid,

Shalat jenazah, biasanya juga diadakan di masjid. Shalat jenazah dilakukan untuk muslim yang telah meninggal, dengan dipimpin seorang imam. Shalat jenazah dilakukan di area sekitar masjid. Ketika gerhana matahari muncul, kaum Muslimin juga mengadakan shalat khusuf untuk mengingat kebesaran Allah. Pada dua hari raya atau 'idain, yaitu Idul Fitri dan Idul Adha umat Muslim juga melakukan shalat.

2. Fungsi Sosial

Masjid di banyak kota di Indonesia menjadi bagian tidak terpisahkan dari alun-alun dan pusat kegiatan masyarakat. Banyak pemimpin Muslim setelah wafatnya Nabi Muhammad saw, berlomba-lomba untuk membangun masjid. Seperti kota Mekkah dan Madinah yang berdiri di sekitar Masjidil Haram dan Masjid Nabawi, kota Karbala juga dibangun di dekat makam Husain bin Ali. Kota Isfahan, Iran dikenal dengan Masjid Imam-nya yang menjadi pusat kegiatan masyarakat. Pada akhir abad ke-17, Syah Abbas I dari dinasti Safawi di Iran mengubah kota Isfahan menjadi salah satu kota terbagus di dunia dengan membangun Masjid Syah dan Masjid Syaikh Lutfallah di pusat kota. Ini menjadikan kota Isfahan memiliki lapangan pusat kota yang terbesar

di dunia. Lapangan ini berfungsi sebagai pasar bahkan tempat olahraga. Masjid biasa digunakan sebagai tempat perkumpulan umat Islam. Biasanya perkembangan jumlah masjid di daerah pinggiran kota, lebih besar dibanding di daerah kota.

3. Fungsi Pendidikan

Fungsi utama masjid yang lainnya adalah sebagai tempat pendidikan. Beberapa masjid, terutama masjid yang didanai oleh pemerintah, biasanya menyediakan tempat belajar baik ilmu keislaman maupun ilmu umum. Sekolah ini memiliki tingkatan dari dasar sampai menengah, walaupun ada beberapa sekolah yang menyediakan tingkat tinggi. Beberapa masjid biasanya menyediakan pendidikan paruh waktu, biasanya setelah subuh, maupun pada sore hari. Pendidikan di masjid ditujukan untuk segala usia, dan mencakup seluruh pelajaran, mulai dari keislaman sampai sains. Selain itu, tujuan adanya pendidikan di masjid adalah untuk mendekatkan generasi muda kepada masjid.

Pelajaran membaca Qur'an dan bahasa Arab sering sekali dijadikan pelajaran di beberapa negara berpenduduk Muslim di daerah luar Arab, termasuk Indonesia. Kelas-kelas untuk muallaf, atau orang yang baru masuk Islam juga disediakan di masjid-masjid di Eropa dan Amerika Serikat, dimana perkembangan agama Islam melaju dengan sangat pesat. Beberapa masjid juga menyediakan pengajaran tentang hukum Islam secara mendalam. Madrasah, walaupun letaknya agak berpisah dari masjid, tapi tersedia bagi umat Islam untuk mempelajari ilmu keislaman.

Berikut ini beberapa kegiatan-kegiatan yang biasa dilakukan di masjid yaitu :

1. Kegiatan Kajian Rutin (Majelis Ilmu)

Dalam Islam, majelis ilmu adalah berkumpulnya beberapa orang di suatu tempat (utamanya di masjid) yang bersih dan baik untuk membahas atau mempelajari ilmu syar'i. Dari pengertian tersebut, majelis ilmu tentu berbeda dengan majelis ghibah atau majelis gosip. Majelis ilmu berbeda dengan majelis senda gurau dan guyonan yang isinya hanya tertawaan dan candaan saja. Ingat, disebut dengan majelis ilmu atau

majelis kajian jika yang dibahas di dalamnya adalah ilmu. Mendahulukan ilmu yang wajib, kemudian ilmu yang fardhu kifayah dan ilmu-ilmu yang baik setelahnya [10].

2. Kegiatan Halaqah

Halaqah adalah sebuah istilah yang ada hubungannya dengan dunia pendidikan, khususnya pendidikan atau pengajaran Islam (tarbiyah Islamiyah). Istilah halaqah (lingkaran) biasanya digunakan untuk menggambarkan sekelompok kecil Muslim yang secara rutin mengkaji ajaran Islam. dapat disimpulkan bahwa halaqah merupakan sekumpulan individu muslim yang bersungguh-sungguh dan berusaha untuk tolong menolong sesama anggota halaqah untuk mempelajari, memahami, dan mengamalkan Islam secara menyeluruh yang berlandaskan al-Qur'an dan Sunnah Rasulullah [11]. Biasanya halaqah atau liqo banyak dilakukan di masjid-masjid dan diikuti mulai dari anak remaja maupun orang-orang tua.

3. Kegiatan Tahsin (Ilmu Tajwid)

Tujuan mempelajari Ilmu Tajwid atau Tahsin adalah agar dapat membaca ayat-ayat al-Quran secara betul (fasih) sesuai dengan yang diajarkan oleh Nabi saw.. Dengan kata lain, agar dapat memelihara lisan dari kesalahan-kesalahan ketika membaca kitab Allah Ta'ala [12]. Kegiatan ini juga sering diadakan di masjid-masjid dan yang mengikuti kegiatan ini mulai dari anak-anak yang berusia tujuh tahun sampai orang dewasa.

4. Kegiatan bulan Ramadan

Masjid, pada bulan Ramadan mengakomodasi umat Muslim untuk beribadah pada bulan Ramadan. Pada bulan Ramadan, masjid-masjid biasanya menyelenggarakan acara pengajian yang amat diminati oleh masyarakat. Ada beberapa masjid yang juga menyediakan makanan untuk sahur. Masjid-masjid biasanya mengundang kaum fakir miskin untuk datang menikmati sahur atau iftar di masjid. Hal ini dilakukan sebagai amal shaleh pada bulan Ramadan. Pada malam hari setelah shalat Isya digelar, umat Muslim disunahkan untuk melaksanakan salat Tarawih berjamaah di masjid. Setelah salat Tarawih, ada beberapa orang yang akan membacakan Al-Qur'an. Pada sepuluh hari terakhir di bulan Ramadan, masjid-masjid besar akan menyelenggarakan

I'tikaf, yaitu sunnah Nabi Muhammad SAW. untuk berdiam diri di Masjid (mengkhususkan hari-hari terakhir ramadan guna meningkatkan amal ibadah) dan memperbanyak mengingat Allah SWT.

5. Kegiatan Amal

Rukun ketiga dalam Rukun Islam adalah zakat. Setiap muslim yang mampu wajib menzakati hartanya sebanyak 2.5% dari jumlah hartanya. Masjid, sebagai pusat dari komunitas umat Islam, menjadi tempat penyaluran zakat bagi yatim piatu dan fakir miskin. Pada saat Idul Fitri, masjid menjadi tempat penyaluran zakat fitrah dan membentuk panitia amil zakat. Panitia zakat, biasanya di bentuk secara lokal oleh orang-orang atau para jemaah yang hidup di sekitar lingkungan masjid dan begitu pula dalam pengelolaannya. Namun, untuk masjid-masjid besar seperti di pusat kota, biasanya langsung ditangani oleh pemerintah daerah setempat.

6. Kegiatan Pengumpulan Dana

Masjid juga menjadi tempat kegiatan untuk mengumpulkan dana. Masjid juga sering mengadakan bazar, dimana umat Islam dapat membeli alat-alat ibadah maupun buku-buku Islam. Masjid tanah liat di Djenné, Mali, secara tahunan mengadakan festival untuk merekonstruksi dan membenah ulang masjid [9].

2.5 Kepengurusan Masjid

Pengurus masjid adalah tenaga penggerak bagi jamaahnya dalam upaya memakmurkan masjid, ini berarti pengurus menjadi salah satu faktor penentu dalam upaya mencapai pemakmuran masjid. Meskipun potensi jamaah masjid sangat besar seperti pendidikan yang tinggi, wawasan yang luas, ketrampilan yang banyak, dana yang cukup dan sarana atau fasilitas hidup yang memadai, tetap saja tidak membuat masjid menjadi makmur bila mereka tidak digerakkan. Disinilah pengurus harus mampu menjadi penggerak, karenanya pengurus masjid harus dinamis dan aktif, jangan berdiam diri saja menunggu partisipasi jamaah. Pengurus juga harus kreatif dan tidak boleh bersikap putus asa serta masa bodoh bila jamaahnya kurang aktif. Ketika seseorang telah menyatakan

kesediannya menjadi pengurus masjid, semestinya dia merasa bertanggungjawab terhadap kemakmuran masjid yang diurusnya itu [13].

2.6 Metode *Prototype*

Prototype adalah suatu proses yang memungkinkan *developer* membuat sebuah model *software*, metode ini baik digunakan apabila *client* tidak bisa memberikan informasi yang maksimal mengenai kebutuhan yang diinginkan. Secara ideal *prototype* berfungsi sebagai sebuah mekanisme untuk mengidentifikasi kebutuhan *software*, bila *prototype* yang sedang bekerja dibangun pengembangannya harus menggunakan fragmen-fragmen program yang ada atau mengaplikasikan alat-alat bantu (contohnya : *report generator*, *window manager* dll) dimana memungkinkan program yang bekerja untuk dimunculkan secara tepat. *Prototyping* model sendiri mempunyai tujuan yaitu mengembangkan model awal *software* menjadi sebuah sistem yang final. *Prototype* dibedakan menjadi 2, yaitu [14]:

1. *Paper Prototype*, menggambarkan interaksi manusia dan mesin dalam sebuah bentuk yang memungkinkan *user* mengerti bagaimana interaksi itu terjadi.
2. *Working Prototype*, adalah *prototype* yang mengimplementasikan beberapa bagian dari fungsi *software* yang diinginkan seperti pada pendekatan pengembangan *software* yang diinginkan seperti pada pendekatan pengembangan *software*. Model ini dimulai dengan :
 - a. Pengumpulan kebutuhan *developer* dan *customer*
 - b. Menentukan semua tujuan *software*

Berikut ini adalah tahapan-tahapan dari *prototype* :

1. Pengumpulan Kebutuhan

Developer dan klien akan bertemu terlebih dahulu dan kemudian menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan berikutnya.

2. Perancangan

Perancangan dilakukan dengan cepat dan rancangan tersebut mewakili semua aspek *software* yang diketahui, dan rancangan ini menjadi dasar pembuatan *prototype*.

3. Menentukan Apakah *Prototype* Dapat Diterima

Pengembang mendemonstrasikan prototipe kepada pengguna untuk mengetahui apakah telah memberikan hasil yang memuaskan. Jika ya, langkah 4 akan diambil. Jika tidak, prototipe direvisi dengan mengulang kembali langkah 1, 2, dan 3 dengan pemahaman yang lebih baik mengenai kebutuhan pengguna.

4. Mengkodekan Sistem

Dalam tahap ini *prototyping* yang sudah di sepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai.

5. Evaluasi Sistem

Pelanggan mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan. Jika ya, sistem dapat digunakan dan jika tidak, ulangi langkah 2 dan 3.

2.6.1 Jenis-Jenis Prototipe

Kata prototipe banyak digunakan untuk berbagai macam hal. Daripada berupaya menyatukan semua kegunaan ini kedalam satu definisi atau berusaha memerintahkan satu pendekatan yang benar terhadap suatu topik *prototyping* yang kontroversial, kami menggambarkan bagaimana masing-masing dari beberapa konsepsi *prototyping* sangat berguna bila diterapkandalam suatu situasi tertentu [15].

1. Prototipe *Patched-up*

Jenis *prototyping* yang pertama ini berkaitan dengan penyusunan sistem yang bekerja namun *patch* atau *patch*-bersama-sama. Contohnya dalam sistem informasi adalah model kerja yang memiliki semua fitur yang diperlukan namun tidak efisien. Dalam hal *prototyping* ini, pengguna dapat berinteraksi dengan sistem sesuai dengan antarmukanya serta jenis keluaran yang tersedia. Pencarian dan penyimpanan

informasi bisa menjadi sangat tidak efisien, karena program-programnya ditulis secara cepat dengan tujuan agar bisa berfungsi, atau lebih dari sekedar efisien. Contoh lain untuk prototipe *patch-up* adalah suatu sistem informasi yang memiliki semua prototipe yang diajukan, tetapi menjadi model dasar yang sebenarnya akan ditingkatkan.

2. Prototipe Non-operasional

Konsepsi *prototype* kedua adalah prototipe dari model skala nganggur yang disusun untuk menguji beberapa rancangan tertentu. Satu model skala nganggur dari suatu sistem informasi bisa diproduksi bila pengkodean yang diperlukan oleh aplikasi terlalu mahal untuk prototipe kecuali bila ada gagasan yang dianggap sangat bermanfaat dari sistem bisa diperoleh melalui *prototyping* masukan dan keluaran saja. Mengenai pemrosesan ini, karena masalah biaya dan waktu, tidak akan diprototipekan. Beberapa keputusan mengenai kegunaan sistem masih dapat dibuat berdasarkan masukan dan keluaran yang diprototipekan.

3. Prototipe *First-of-Series*

Konsepsi *prototyping* ketiga melibatkan penciptaan suatu model skala lengkap pertama dari sistem, yang disebut pilot. Jenis *prototyping* semacam ini digunakan ketika merencanakan banyak versi dari sistem informasi yang sama. Model kerja skala lengkap ini memungkinkan pengguna bisa mendapat pengalaman interaksi yang realistis dengan sistem baru, yang sebelumnya meminimalkan biaya untuk menanggulangi masalah-masalah yang ada. Contohnya ditemukan dalam instalasi perbankan untuk pentransferan dana secara elektronik. Pertama-tama, dipasang sebuah prototipe berskala lengkap di salah satu dari dua lokasi yang ada, dan apabila berhasil, dipasang duplikat-duplikatnya di semua lokasi berdasarkan pola-pola yang digunakan konsumen serta faktor-faktor penting lainnya.

4. Prototipe Fitur-fitur Terpilih

Konsepsi *prototyping* keempat berkaitan dengan pembangunan suatu model operasional yang mencakup beberapa, tapi tidak semua, fitur-fitur yang dimiliki sistem final. Analoginya, ibarat sebuah pusat perbelanjaan retail baru yang akan

dibuka sebelum pembangunan seluruh bagian toko selesai. Pada pusat retail yang baru dibangun tersebut, fungsi utamanya ialah agar orang bisa membeli beberapa barang tertentu. Bila melakukan *prototyping* sistem informasi dengan cara ini, meski tidak semua, fitur-fitur intinya harus dimasukkan. Bila *prototyping* jenis ini dilakukan, sistem akan bisa tercapai sesuai modul sehingga bila fitur-fitur yang dibuat prototipenya dianggap berhasil, maka akan dipadukan ke dalam sistem final yang besar tanpa melakukan pekerjaan yang besar dalam antarmuka.

2.6.2 Kelebihan dan Kelemahan *Prototype*

Adapun kelebihan dari *prototype* adalah sebagai berikut [14]:

1. Komunikasi akan terjalin baik antara pengembang dan pelanggan.
2. Pengembang dapat bekerja lebih baik dalam menentukan kebutuhan setiap pelanggannya.
3. Pelanggan berperan aktif dalam proses pengembangan sistem.
4. Lebih menghemat waktu dalam pengembangan sistem.
5. Penerapan menjadi lebih mudah karena pemakai mengetahui apa yang diharapkannya.

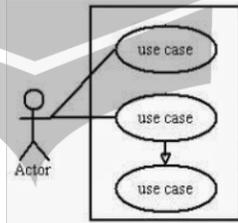
Adapun kelemahan dari *prototype* adalah sebagai berikut :

1. Pelanggan kadang tidak melihat atau menyadari bahwa perangkat lunak yang ada belum mencantumkan kualitas perangkat lunak secara keseluruhan dan juga belum memikirkan kemampuan pemeliharaan untuk jangka waktu lama.
2. Pengembang biasanya ingin cepat menyelesaikan proyek sehingga menggunakan algoritma dan bahasa pemrograman yang sederhana untuk membuat *prototyping* lebih cepat selesai tanpa memikirkan lebih lanjut bahwa program tersebut hanya merupakan sebuah kerangka kerja (*blueprint*) dari sistem.
3. Hubungan pelanggan dengan computer yang disediakan mungkin tidak mencerminkan teknik perancangan yang baik dan benar.

2.7 Teknik Pengembangan Sistem

2.7.1 Pengertian Use Case

Use case diagram adalah diagram yang membantu dalam menyusun requirement sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua fitur yang ada pada sistem. *Use case* diagram menggunakan fungsionalitas yang diharapkan dari sebuah sistem. *Use case* diagram menekankan apa yang dibuat sistem, dan bukan bagaimana. Sebuah *use case* merepresentasikan sebuah interaksi antara *actor* dengan sistem [16].

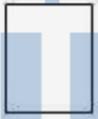
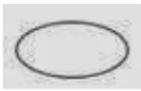


2.7.2 Simbol dan Notasi

Berikut adalah notasi-notasi yang ada pada *use case* diagram [16]:

Tabel 2. 1 Simbol dan Notasi *Use Case*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).

3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use cases</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk

			menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

2.7.3 Basis Data

Sebuah basis data adalah sebuah kumpulan data yang saling berhubungan secara logis, dan merupakan sebuah penjelasan dari data tersebut, yang di desain untuk menemukan data yang diutuhkan oleh sebuah organisasi. Di dalam basis data, semua data diintegrasikan dengan menghindari duplikasi data. Basis data dapat digunakan oleh banyak department dan pemakai. Basis data tidak hanya memegang data operasional organisasi, tetapi juga penjelasan mengenai data tersebut. Karena alasan tersebut basis data juga dideskripsikan sebagai kumpulan data yang saling terintegrasi. Basis data juga merupakan sekumpulan elemen data terintegrasi yang secara logika saling berhubungan. Basis data mengonsolidasikan berbagai catatan yang terlebih dahulu disimpan dalam file-file terpisah ke dalam satu gabungan umum elemen data yang menyediakan data untuk banyak aplikasi. Elemen data mendeskripsikan entitas-entitas dan hubungan antara entitas-entitas tersebut.

Desain basis data adalah proses membuat desain yang akan mendukung operasional dan tujuan perusahaan. Tujuan desain basis data adalah:

1. Menggambarkan relasi data antara data yang dibutuhkan oleh aplikasi dan *user view*.
2. Menyediakan model data yang mendukung seluruh transaksi yang diperlukan.
3. Menspesifikasikan desain dengan struktur yang sesuai dengan kebutuhan sistem.

Ada beberapa pendekatan yang dapat digunakan dalam mendesain basis data, yaitu [17]:

1. *Top-down*

Diawali dengan membuat data model. Pendekatan *Top-down* dapat diilustrasikan menggunakan *entity-relationship* (ER) model yang *high level*, lalu mengidentifikasi *entity*, dan *relationship* antar-*entity* organisasi. Pendekatan ini sesuai bagi basis data yang kompleks.

2. *Bottom-up*

Dimulai dari level dasar *attribute* (*property entity* dan *relationship*), menganalisa hubungan antar-*attribute*, mengelompokkannya dalam suatu relasi yang menggambarkan tipe *entity* dan relasi antara *entity*. Pendekatan ini sesuai bagi basis data dengan jumlah *attribute* yang sedikit.

3. *Inside-out*

Mirip seperti pendekatan *Bottom-up*, perbedaannya adalah tahap awal mengidentifikasi *major entity* lalu menguraikannya menjadi *entity* relasi-relasi dan *attribute-attribute* yang berhubungan dengan *major entity*.

4. *Mixed*

Menggunakan pendekatan *Bottom-up* dan *Top-down*.

UNIVERSITAS
MIKROSKIL