

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 TINJAUAN PUSTAKA

Presensi atau daftar hadir adalah formulir berisi data identitas dan validasi dengan paraf atau tandatangan seseorang yang digunakan sebagai bukti kehadiran/keikutsertaan seseorang tersebut dalam suatu acara/kegiatan (Badunk, 2013). Terkadang di sebuah perusahaan seseorang atau karyawan melakukan presensi saat datang dan pulang dari kerja. Selain itu, setiap perusahaan atau instansi memiliki sistem presensi yang berbeda-beda.

Sistem presensi yang terkomputerisasi sudah banyak dikembangkan sebelumnya dengan perangkat lunak dan metode yang berbeda-beda. Sistem yang dikembangkan juga bervariasi, berbasis *website*, *fingerprint*, maupun *desktop*.

Penelitian yang dilakukan oleh Muhammad, dkk. (2013) menjelaskan bahwa sistem presensi MONIKUL V.1 yang telah dikembangkan dapat melakukan monitoring absensi *fingerprint* untuk perkuliahan yang ada pada Jurusan Sistem Informasi ITS. Sistem ini juga dapat melakukan pengolahan jadwal belajar mengajar terkait absensi *fingerprint*. Selain itu pemanfaatan teknologi ini dapat meringankan beban TU jurusan untuk rekapitulasi absensi saat akhir semester. Melalui uji coba performa menggunakan *apache benchmark* pada lingkungan uji coba, MONIKUL V.1 dapat menangani *request* lebih dari 1000 *user* berbeda secara bersamaan.

Dengan adanya sistem ini memungkinkan semua civitas yang tergabung di dalamnya dapat melihat dan merekap absensinya. Civitas yang dimaksud adalah

user yang tidak lain adalah Dosen, Mahasiswa, dan Admin. *User* tersebut memiliki privilege berbeda tergantung jabatan *user* tersebut (Muhammad dkk., 2013).

Selain itu, penelitian yang dilakukan oleh Zakaria dan Fandi (2008) mengungkapkan bahwa PDA dapat membantu penggunanya dalam melakukan aktivitas yang sebelumnya sulit untuk dilakukan. Salah satunya adalah untuk melakukan presensi perkuliahan, mayoritas masih menggunakan cara manual, yaitu dengan mengedarkan kertas presensi yang ditandatangani oleh mahasiswa. Sehingga PDA juga mampu melakukan presensi mahasiswa oleh dosen yang bersangkutan, dan langsung dikirim ke aplikasi *desktop* di suatu *Personal Computer* (PC) untuk kemudian disimpan ke dalam *database*.

Dari hasil penelitian tersebut diketahui bahwa dalam proses presensi mahasiswa, kita dapat menggunakan PDA sebagai media untuk presensi. Dan dalam proses manajemen presensi kita dapat menggunakan *Web Servis* dalam pengembangannya. Dalam Proses manajemen kita bisa mengetahui beberapa data salah satunya adalah data presensi mahasiswa.

Berdasarkan pengujian dan analisis yang telah dilakukan oleh masing-masing peneliti, kedua peneliti tersebut membangun beberapa sistem presensi menggunakan beberapa media. Namun, masih terdapat kelemahan pada dua penelitian tersebut. Penelitian Muhammad, dkk. (2013) memiliki kelemahan pada pengadaan alat *fingerprint* yang terlalu mahal. Sehingga, sangat tidak efektif jika diterapkan pada sebuah instansi besar seperti universitas. Penelitian Zakaria dan Fandi (2008) juga memiliki kelemahan pada sistem *security* yang digunakan.

Aplikasi PDA yang digunakan belum mempunyai identitas yang unik. Sehingga, sangat mudah dicurangi jika tidak ada sistem otentikasi yang aman. Untuk itu, aplikasi presensi menggunakan PDA akan lebih baik jika dikembangkan menggunakan sistem yang unik seperti *two factor authentication*.

Terdapat penelitian lain yang dilakukan oleh Gusman (2013) mengenai *Analysis and Implementation of Two Factor Authentication with QR Code in Web-Based Application*. Dia mengungkapkan bahwa proses otentikasi pengguna dalam layanan *Internet banking*, selain menggunakan *password* miliknya sendiri, pengguna juga menggunakan *token* yang akan membangkitkan suatu *password* sekali pakai (*one time password*) agar dapat terotentikasi dalam menggunakan layanan *Internet banking*. Pada layanan *Internet banking* suatu *token* bekerja berdasarkan prinsip *challenge* dan *response*, dimana *web server* akan membangkitkan suatu *challenge* yang nantinya akan diproses oleh *token* dan menghasilkan suatu *response* yang digunakan untuk melakukan otentikasi.

Berdasarkan penelitian Gusman (2013), penggunaan *token* dapat mempercepat dan mempermudah pengguna dalam melakukan proses otentikasi. Pada *token* biasanya pengguna diharuskan memasukkan kode *challenge* (berdasarkan hasil *generated random string*) dan kode *response* secara manual. Sehingga pengguna lebih terjamin keamanannya saat melakukan *logon* ke sistem karena aplikasi telah menerapkan *two factor authentication*.

Hasil dari pengujian yang dilakukan oleh Gusman (2013) adalah pencurian terhadap *token* pengguna tidak dapat berguna, karena *token* juga dilindungi dengan *password* untuk menggunakannya. Aktivitas pencurian terhadap informasi

pengguna pada aplikasi *token* juga tidak berguna sebab informasi tersebut dienkripsi. Pencurian informasi *username* dan *QR code* juga tidak berguna, sebab informasi tersebut akan segera tidak berlaku saat pengguna berhasil terotentikasi (*one time password*). Selain itu proses otentikasi dibatasi hanya dapat dilakukan dalam batas waktu tertentu.

Melihat potensi token sebagai pengaman, maka token juga dapat dikembangkan pada sistem presensi menggunakan PDA. Dengan menggunakan teknologi *web service*, sistem presensi online dapat dikembangkan pada dua *client*, yaitu dosen dan mahasiswa. *Client* untuk dosen digunakan untuk men-*generate token* sedangkan *client* untuk mahasiswa digunakan untuk *enroll token*. Sehingga mahasiswa dapat melakukan presensi menggunakan *token* yang telah di-*generate* oleh dosen.

2.2 LANDASAN TEORI

2.2.1 Definisi Pengelolaan Data Presensi

Pengelolaan adalah serangkaian aktivitas-aktivitas koordinasi yang mencakup perencanaan, pengorganisasian, pengarahan, pengendalian, penempatan serta pengambilan keputusan untuk menghasilkan suatu produk dan jasa yang efektif dan efisien Sikula (1973). Berdasarkan penelitian Sikula (1973), peneliti dapat menyimpulkan bahwa pengolahan data presensi adalah suatu proses kegiatan pencatatan terhadap setiap presensi dengan tujuan untuk mengetahui data dan laporan berkaitan dengan presensi masuknya karyawan, presensi pulangny karyawan, dan bentuk transaksi yang dilakukan pada kurun waktu tertentu. Tujuan

dari pengolahan data presensi adalah agar dapat mengetahui data dan informasi yang berkaitan dengan informasi presensi.

2.2.2 Intranet

Intranet adalah jaringan komputer yang saling berhubungan atau tersambung yang digunakan oleh suatu sistem organisasi maupun lembaga. Atau Definisi Intranet yang lain adalah Intranet merupakan suatu jaringan komputer yang berbasis protokol TCP/IP, layaknya jaringan internet hanya saja penggunaannya yang dibatasi atau lebih tertutup jadi tidak semua pengguna atau orang dapat secara mudah mengakses jaringan intranet serta hanya orang atau pengguna tertentu saja yang dapat masuk dan menggunakan jaringan intranet.

Dapat disimpulkan Fungsi Intranet adalah berfungsi mengkomunikasikan komputer satu dengan yang lain, persis seperti internet tetapi memiliki layanannya yang terbatas, tak seluas dan seberagam di jaringan internet.

Perbedaan antara *internet* dan *intranet* sebagai berikut:

Jaringan *internet*:

- 1) Sifat yang terbuka, yang artinya dapat di akses oleh siapa saja asalkan dapat terhubung dengan jaringan internet.
- 2) Jangkauan yang luas dan tak terbatas, yang artinya dimana si pengguna dapat menjangkau seluruh belahan dunia tanpa batasan.

Jaringan *intranet*:

- 1) Sifatnya tak terbuka, artinya dimana hanya dapat diakses oleh orang-orang atau pengguna yang memiliki akses terhadap jaringan intranet.
- 2) Jangkauan yang luas tapi terbatas, artinya dimana hanya menjangkau pada area-area yang di perlukan untuk jaringan intranet saja.

2.2.3 Aplikasi Web

Aplikasi *web* merupakan sebuah aplikasi yang menggunakan teknologi *browser* untuk menjalankan aplikasi dan diakses melalui jaringan komputer (Remick, 2011). Sedangkan menurut Wulandari (2015), aplikasi *web* merupakan aplikasi yang dapat diakses menggunakan *web browser* dengan melalui jaringan *internet* maupun *intranet*.

Aplikasi *web* juga merupakan suatu perangkat lunak komputer yang dikodekan dalam bahasa pemrograman yang mendukung perangkat lunak berbasis *web* seperti HTML, JavaScript, CSS, Ruby, Python, Php, Java dan bahasa pemrograman lainnya.

Aplikasi berbasis *web* dapat digunakan untuk berbagai macam tujuan yang berbeda. Sebagai contoh aplikasi berbasis *web* dapat digunakan untuk membuat *invoice* dan memberikan cara yang mudah dalam penyimpanan data di *database*. Aplikasi ini juga dapat dipergunakan untuk mengatur pengelolaan, karena fitur tersebut sangat berguna khususnya bagi mereka yang berbisnis ritel. Bukan hanya itu, Aplikasi berbasis *web* juga dapat bekerja memonitoring sistem dalam hal tampilan.

2.2.4 Two Factor Authentication

Otentikasi adalah proses membuktikan suatu identitas (Kessler, 1998).

Otentikasi bertujuan membuktikan keabsahan seseorang dalam menggunakan suatu layanan. Metode otentikasi bisa dilihat dalam 4 kategori metode (Pamadya, 2011):

- 1) *Something you know*, merupakan metode yang paling umum dengan mengandalkan *password*.
- 2) *Something you have*, merupakan faktor tambahan untuk membuat otentikasi menjadi lebih aman dengan mengandalkan barang yang sifatnya unik, contohnya: kartu magnetik/*smartcard*, *USB token*, *token hardware*, dan sebagainya.
- 3) *Something you are*, merupakan metode yang mengandalkan keunikan bagianbagian tubuh yang tidak Mungkin ada pada orang lain, seperti Sidik jari, Sidik retina, Dan suara.
- 4) *Something you do*, melibatkan bahwa setiap user dalam melakukan sesuatu dengan cara yang berbeda. Contoh: Penggunaan analisis suara (*voice recognition*), dan analisis tulisan tangan.

Pada aplikasi yang kritis dan sensitif seperti transaksi keuangan, satu metode otentikasi saja tidak cukup. Oleh karena itu muncul istilah 2FA (*Two Factor Authentication*) yang merupakan sistem otentikasi yang menggunakan 2 faktor (metode) yang berbeda. Metode otentikasi dapat dikombinasikan untuk meningkatkan keamanan, salah satu contohnya adalah dengan kombinasi “*something you have*” berupa kartu ATM dengan “*something you know*” berupa PIN. Kombinasi ini merupakan kombinasi yang paling banyak dipakai.

Contoh kasus lain adalah ketika anda berbelanja di pasar modern dan membayar dengan kartu, tanpa disadari anda telah memakai lebih dari satu faktor autentikasi. Faktor yang pertama adalah “*Something You Have*” yaitu kartu debit/kredit anda. Faktor kedua adalah “*Something You Know*”, ketika anda diminta memasukkan PIN ke dalam mesin EDC. Bahkan mungkin ada faktor ketiga yaitu “*Something You Can*”, ketika anda diminta menanda-tangani nota pembayaran yang dicetak mesin EDC.

Internet banking juga menggunakan *two factor authentication* dengan mengombinasikan “*something you know*” berupa *password* dan “*something you have*” berupa *hardware token* (*keyBCA* atau *Token Mandiri*).

2.2.5 Token

Token merupakan unit atau elemen dasar dalam bahasa komputer (seperti ‘kata’ di bahasa manusia), dimana unit tersebut tidak terbagi lagi. Token merupakan bagian dari hasil pemecahan sumber yaitu penerjemahan program *lexeme* pada saat melakukan *scanner* (Soleh, 2013). Karakter didalam token dikategorikan sebagai salah satu dari lima kelas token yang menggambarkan fungsi mereka (*identifier*, *keyword*, *literal string*, *operator*, *separator*) sesuai dengan aturan bahasa pemrograman.

Menurut Oktafia (2008), *token* merepresentasikan nama:

- 1) *Identifier* (nama variabel, fungsi dan tipe atau nama yang didefinisikan pemakai).

- 2) *Keyword*
- 3) *Literal String*
- 4) *Operator*
- 5) *Label*
- 6) Simbol tanda (tanda kurung, koma dan titik koma).

Penelitian ini, hanya menggunakan beberapa karakter dalam penerapannya, yaitu:

- 1) Angka (1, 2, 3, 4, 5, 6, 7, 8, 9, 0)
- 2) Huruf besar (A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z)
- 3) Huruf kecil (a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z)

Token ini terdiri dari delapan *digit random keyword*. Proses *generate random token* terjadi setelah pengguna (dosen) meng-klik tombol *create* pada sesi perkuliahan. Setelah tombol *create* diklik, sistem akan melakukan *random* dari daftar karakter yang disebutkan. Jika *token* yang dibuat sama dengan *token* sesi lain di hari yang sama, maka sistem akan melakukan pengulangan dalam proses *generate random token*. Setelah sistem mendapatkan *token* yang unik, *token* akan ditampilkan pada halaman *detail* sesi perkuliahan. Sehingga *token* ini nantinya akan digunakan oleh mahasiswa untuk melakukan presensi.

2.2.6 Lightweight Directory Access Protocol (LDAP) Server

Lightweight Directory Access protocol (LDAP) merupakan protokol aplikasi yang mengatur mekanisme manajemen dan pengaksesan layanan informasi direktori terdistribusi melalui jaringan IP. Informasi tersebut dapat berupa *people*, *organizations*, *roles*, *services* dan banyak entitas lainnya. LDAP menggunakan model *client-server*, di mana *client* mengirimkan identifier data kepada *server* dengan menggunakan protokol TCP/IP dan *server* mencoba mencari data tersebut pada DIT (*Directory Information Tree*) yang tersimpan di *server*. Apabila data ditemukan, maka hasilnya akan dikirimkan ke *client* tersebut, namun bila tidak, maka hasilnya berupa *pointer* ke *server* lain yang menyimpan data yang dicari. LDAP memiliki bentuk struktur yang berhirarki, bukan berformat kolom dan baris sebagaimana *database* normal, sehingga memudahkan untuk memasukkan sejumlah besar detail yang mirip dalam bentuk yang terorganisir. LDAP memiliki dua layanan utama, yaitu:

- 1) Slapd: LDAP *daemon* yang melayani permintaan dari klient, *query* dan berkomunikasi dengan *backend database*.
- 2) Slurp: Replikasi *daemon* yang berfungsi melayani replikasi data agar terus terjadi sinkronisasi data antara klien dan *server*.

2.2.7 UML

Unified Modeling Language (UML) adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya (Nugroho et al, 2014). UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat *tool* untuk mendukung pengembangan sistem

tersebut. UML mulai diperkenalkan oleh Object Management Group, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an (Nugroho et al, 2014). Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP (Nugroho et al, 2014). UML merupakan dasar bagi perangkat (*tool*) desain berorientasi objek dari IBM (Nugroho et al, 2014).

Berikut ini adalah definisi dari 4 *diagram* UML menurut Handayani (2014) yaitu:

1) *Use Case Diagram*

Use Case Diagram merupakan inti fungsionalitas koheren yang diekspresikan sebagai transaksi-transaksi yang terjadi antara *actor* dan *system*.

2) *Sequence Diagram*

Secara grafis menggambarkan bagaimana objek berinteraksi dengan satu sama lain melalui pesan pada sekuensi sebuah *use case* atau operasi.

3) *Activity Diagram*

Secara grafis digunakan untuk menggambarkan rangkaian aliran aktivitas baik proses bisnis maupun *use case*. *Activity diagram* dapat juga digunakan untuk memodelkan *action* yang akan dilakukan saat sebuah operasi dieksekusi, dan memodelkan hasil dari *action* tersebut.

2.2.8 Visual Studio 2015

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi

console, aplikasi Windows, ataupun aplikasi Web. Visual Studio mencakup kompiler, SDK, Integrated Development Environment (IDE), dan dokumentasi (umumnya berupa MSDN Library). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe.

Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas Windows) ataupun *managed code* (dalam bentuk Microsoft Intermediate Language di atas .NET Framework). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi Silverlight, aplikasi Windows Mobile (yang berjalan di atas .NET Compact Framework).

Microsoft Visual Studio 2015 Enterprise dapat membuat dan mengedit aplikasi lebih mudah dan cepat sehingga meningkatkan produktifitas dalam pembuatan aplikasi. Microsoft Visual Studio 2015 Enterprise sudah mendukung sistem operasi windows terbaru yaitu windows 10. Sehingga software ini sangat mendukung dan mudah bagi instansi yang ingin mengembangkan aplikasinya ke versi terbaru.

2.2.9 ASP.NET (MVC)

ASP.NET MVC adalah kerangka kerja untuk membangun aplikasi *web* yang menerapkan pola Model-View-Controller umum untuk kerangka ASP.NET (Galloway et al, 2014). Microsoft telah merilis Visual Studio 2015 Enterprise. Ada banyak fitur baru yang diperkenalkan dengan rilis ini dari Visual Studio. Salah

satunya adalah MVC 6, itu adalah versi terbaru dari MVC. MVC adalah pola arsitektur yang memisahkan aplikasi menjadi tiga bagian utama yang disebut Model, View dan Controller. MVC adalah cara yang diakui untuk mengembangkan aplikasi.

MVC 5 adalah sebuah *update* terbaru dari Visual Studio dengan kelebihan dapat membangun sebuah *web* dinamis atau situs *data-driven*. Visual Studio telah menambahkan fitur menarik seperti *pages application* , *mobile optimization* dan *adaptive rendering*.

Menurut Joshi (2013) ada beberapa karakteristik penting dalam MVC 5 seperti yang diberikan di bawah ini:

- 1) One ASP.NET
- 2) ASP.NET Identity
- 3) Bootstrap
- 4) Filter Authentication
- 5) Filter Authorization

2.2.10 Microsoft SQL Server Management Studio

Microsoft SQL Server adalah sistem manajemen *database* relasional yang dikembangkan oleh Microsoft. Sebagai database, Microsoft SQL Server adalah produk *software* yang fungsi utamanya adalah untuk menyimpan dan mengambil data seperti yang diminta oleh aplikasi perangkat lunak lain, baik *localhost* atau pada komputer yang sama maupun melalui jaringan (pada komputer lain dalam jaringan, termasuk jaringan *internet*). Banyak edisi yang berbeda dari Microsoft SQL

Server ditujukan untuk pengguna yang berbeda dan untuk beban kerja yang berbeda pula (mulai dari aplikasi kecil yang menyimpan dan mengambil data di komputer yang sama ataupun untuk jutaan pengguna dan komputer yang mengakses data dalam jumlah besar dari *internet* pada waktu yang sama).

Microsoft SQL Server termasuk DBMS profesional. Beberapa pesaing seperti MySQL, Oracle, telah mengembangkan *software* serupa dalam beberapa tahun terakhir, tetapi Microsoft SQL Server lebih mudah digunakan dan memiliki lebih banyak fitur. Pemicunya antara lain adalah dukungan penuh dari Microsoft. Perangkat lunak yang ditawarkan oleh Microsoft juga menawarkan integrasi yang erat dengan .NET framework, dan ini tidak dimiliki oleh produk lain (Aiska, 2011).

2.2.11 Black Box testing

Dalam pengujian perangkat lunak ada dua yaitu *white box testing* dan *black box testing*. Dari kedua metode itu, pada skripsi dipilih menggunakan *black box testing* karena dianggap lebih tepat dibanding *white box testing*. Perangkat lunak memerlukan seperangkat tes untuk pencarian kesalahan fungsi-fungsi dalam aplikasi sehingga dalam hal ini *black box testing* lebih sesuai. Pengujian ini digunakan untuk mengetahui apakah fungsi-fungsi dalam perangkat lunak sudah sesuai dengan yang diharapkan.

Menurut Pressman (2010), *black box testing* berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan engineer untuk memperoleh input yang sepenuhnya akan melaksanakan persyaratan fungsional untuk sebuah program. *Black box testing* berusaha untuk menemukan kesalahan dalam kategori berikut:

- a. Fungsi yang tidak benar atau fungsi yang hilang.
- b. Kesalahan antarmuka.
- c. Kesalahan dalam struktur data atau akses database eksternal.
- d. Kesalahan kinerja.
- e. Kesalahan inisialisasi dan pemutusan kesalahan.