

Pembuatan Sistem Informasi Kehadiran Pegawai Non-PNS Berbasis Web Pada Pusat Pelayanan Teknologi (Pusyantek) BPPT

(Building Attendance Web-Based Information System for Non Civil Servant at Center for Technology Service (Pusyantek) BPPT)

Rachma Fauziah Krismayanti¹, Sri Nurdiati², Agung Widyo Utomo³

¹ Sekolah Vokasi Institut Pertanian Bogor, Jl. Kumpang No. 14, Bogor, Indonesia

² Universitas IPB, Jl. Raya Dramaga Kampus IPB Dramaga, Bogor, Indonesia

³ Pusat Pelayanan Teknologi BPPT, Jl. MH Thamrin No. 8, Jakarta Pusat, Indonesia

Email : rachmafrau@gmail.com

ABSTRACT

Center for Technology Service (Pusyantek) in Agency for The Assesment and Application of Technology (BPPT) has two kinds of employee, there are civil and non-civil servants. The civil servants already have the official website for attendance system, but the non-civil servants don't have the official website. The presence process for employee has engaged by the fingerprint machine, but still need help from Microsoft Excel. Administrator creates a recapitulation constantly for a period. This process can cause an inaccurate data, inefficient time management, and others. Employees also hard to check the presence's datas and complete the absences. With such problems, administrator needs attendance system that can help administrator and non-civil servants. The system was built by incremental method because the client already know requirements that the system necessary. Incremental method consist of five steps, those are communication, planning, modelling, construction, and deployment. System building involved two users, admin and employee. The system building engage by two iteration, which is first increment with 22 functions and second increment with 28 functions. System building successfully implemented attendance data management feature, attendance feature, filling permit's data feature, and filling leave's data feature.

Key words: incremental method, attendance information system, BPPT, Pusyantek

ABSTRAK

Badan Pengkajian dan Penerapan Teknologi (BPPT) khususnya di Pusat Pelayanan Teknologi (Pusyantek) memiliki tenaga kerja yang terdiri atas pegawai PNS dan non-PNS. Pegawai PNS sudah memiliki web resmi untuk sistem kehadiran, namun pegawai non-PNS belum memiliki. Proses kehadiran sudah dilakukan dengan mesin *fingerprint*, namun pengolahannya masih dilakukan secara manual menggunakan bantuan aplikasi Microsoft Excel. Pengolah melakukan pengolahan untuk menghasilkan rekap secara berkala per periodenya. Hal tersebut bisa menyebabkan data yang tidak akurat, pemanfaatan waktu yang tidak efisien, dan lain-lain. Pegawai pun sulit untuk melihat data kehadiran dan melengkapi data ketidakhadiran. Dengan adanya

permasalahan tersebut, pengolah mengharapkan adanya sistem kehadiran berbasis web yang memudahkan pengolah dan pegawai non-PNS Pusyantek. Sistem dibuat dengan menerapkan alur kerja yang terstruktur dari metode pengembangan *incremental*. Metode tersebut digunakan karena klien sudah mengetahui kebutuhan apa saja yang diperlukan. Metode *incremental* terdiri atas lima tahapan, yaitu komunikasi, perencanaan, pemodelan, pembuatan, serta penyebaran dan umpan balik. Pembuatan sistem melibatkan dua *user*, yaitu admin dan pegawai. Pembuatan sistem menghasilkan dua kali iterasi yaitu *increment* pertama dengan 22 fungsi dan *increment* kedua dengan 28 fungsi. Pembuatan sistem berhasil mengimplementasikan fitur pengelolaan data kehadiran, fitur kehadiran, fitur kelengkapan data pengajuan izin, dan fitur kelengkapan data pengajuan cuti.

Kata kunci : metode incremental, sistem informasi kehadiran, BPPT, Pusyantek

PENDAHULUAN

Badan Pengkajian dan Penerapan Teknologi (BPPT), tepatnya di Pusat Pelayanan Teknologi (Pusyantek) adalah unit utama yang memiliki peran dalam bidang teknologi. Saat ini, kebutuhan sistem berbasis *website* semakin meningkat. Sistem informasi berbasis web menjadi pilihan bagi perusahaan sebagai wadah pelayanan untuk kebutuhan internal maupun eksternal.

Badan Pengkajian dan Penerapan Teknologi (BPPT), khususnya pada Pusat Pelayanan Teknologi (Pusyantek) memiliki sumber daya manusia yang memadai. Pegawai di Pusyantek terdiri atas pegawai PNS dan non-PNS. Pegawai PNS Pusyantek sudah memiliki web resmi untuk kehadiran, sedangkan pegawai non-PNS masih belum tersedia. Penerapan kehadiran untuk pegawai non-PNS sudah menggunakan mesin kehadiran *fingerprint*, namun pengolahannya masih dilakukan secara manual. Keluaran dari aplikasi bawaan mesin *fingerprint* tidak dapat diubah sesuai kebutuhan. Maka dari itu, setiap periodenya pengolah harus mengolah kehadiran melalui Microsoft Excel dengan memasukkan rumus ke dalamnya. Selain itu, pegawai sulit untuk melihat kehadiran pegawai karena belum ada web yang menampilkan kehadiran. Pegawai juga kesulitan dalam melengkapi data keterangan izin dan cuti.

Diperlukan sistem yang memudahkan pengolah dalam mengolah kehadiran pegawai non-PNS dan memudahkan pegawai dalam memeriksa kehadiran, melakukan pelengkapan data izin, dan data cuti. Sistem ini diharapkan memiliki fitur pengelolaan data pegawai, pemasukan data kehadiran oleh admin, pelengkapan data izin, pelengkapan data cuti, serta rekapitulasi data.

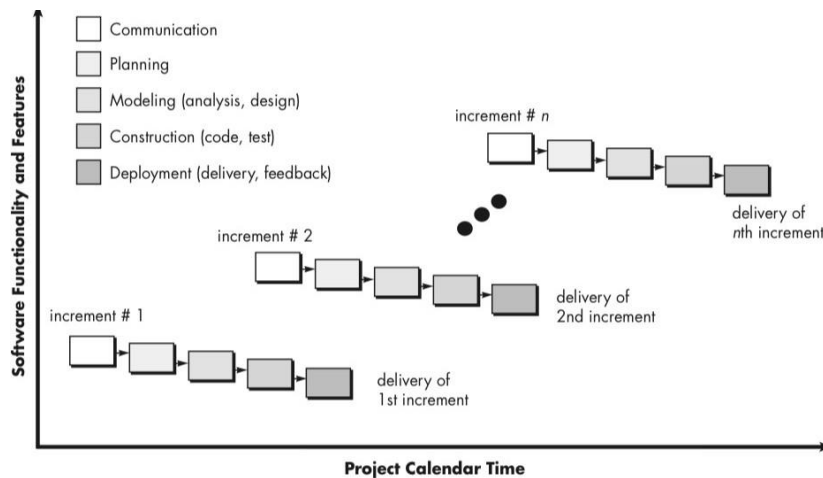
Tujuan dibuatnya sistem informasi kehadiran ini di antaranya untuk menyediakan fitur pengelolaan data kehadiran pegawai, fitur kehadiran pegawai, fitur kelengkapan data izin, dan fitur kelengkapan data cuti.

METODE PENELITIAN

Pembuatan sistem menggunakan metode *incremental*. Alasan

penggunaan metode ini karena metode pengembangannya sederhana, resiko kegagalannya lebih rendah, klien sudah mengetahui kebutuhan yang dibutuhkan, serta klien juga ikut andil dalam memberikan masukannya.

Metode *incremental* adalah pengembangan secara bertahap yang dilakukan dengan cara membagi ke dalam subsistem yang bisa berjalan secara paralel. Menurut Roger Pressman (Pressman 2010) terdapat lima tahapan yang dilakukan, yaitu: 1) komunikasi, 2) perencanaan, 3) pemodelan, 4) pembuatan, dan 5) penyebaran. Berdasarkan Gambar 2, lima tahapan merupakan implementasi dari satu subsistem, yang nantinya tahapan-tahapan tersebut menjadi satu kesatuan yang membentuk *increment 1*, *increment 2*, dan seterusnya. Tahapan metode *incremental* dapat dilihat pada Gambar 1.



Gambar 1 Tahapan metodologi *incremental* (Pressman2010)

Tahapan pertama komunikasi antara pengguna dan pengembang. Tahap ini harus dilakukan sebelum melakukan analisis, pemodelan, atau spesifikasi (Pressman 2010). Pengguna memiliki masalah yang bisa diselesaikan dengan sistem berbasis komputer, maka dilakukan komunikasi agar pengembang bisa mendapatkan sasaran yang akan dicapai, memahami masalah, dan menentukan kebutuhan apa saja yang dapat membantu memahami fungsi dan fitur sistem.

Tahapan kedua dibuat suatu perencanaan sistem serta batasan ruang lingkungannya, meliputi manajemen dan praktek teknis yang memudahkan pengembang untuk menjabarkan *road map* dalam mencapai tujuan dan sasaran (Pressman 2010). Perencanaan dilakukan dengan cara membuat rencana tugas- tugas yang hendak dikerjakan, kebutuhan , produk yang dihasilkan, dan jadwal.

Tahap ketiga pemodelan, untuk mempermudah memahami kebutuhan secara menyeluruh. Standar pemodelan yang digunakan adalah *Unified Modeling Language* (UML) yaitu menggunakan *use case diagram*, *class diagram*, dan *activity diagram*. Diagram-diagram ini membantu dalam menyetarakan spesifikasi kebutuhan.

Tahapan keempat adalah pembuatan, proses menghasilkan *increment* berdasarkan pemodelan yang telah dibuat sebelumnya.

Tujuannya untuk mendefinisikan spesifikasi kebutuhan. Aktivitas mencakup kegiatan penulisan kode dan uji perangkat lunak hingga siap diberikan kepada klien.

Tahapan kelima adalah penyebaran, sistem yang telah dibuat akan diberikan kepada klien atau pengguna, kemudian pengguna akan mengevaluasi apakah sudah sesuai dengan kebutuhan fungsional atau belum. Setelah mengevaluasi, pengguna akan memberikan *feedback* terhadap sistem yang sudah dibuat.

HASIL DAN PEMBAHASAN

Pembuatan sistem menghasilkan dua iterasi *increment*, yaitu *increment* pertama dan *increment* kedua. *Increment* pertama menghasilkan 22 fungsi dan *increment* kedua menghasilkan 28 fungsi baru.

1. *Increment* pertama

1.1 Komunikasi

Pusyantek BPPT memiliki jadwal masuk dan keluar yang berbeda pada hari kerja Senin sampai Jumat. Hari Senin sampai dengan Kamis pegawai masuk pukul 07.30 dan keluar pukul 16.00, dengan durasi kerja tujuh jam tiga puluh menit ditambah satu jam istirahat. Berbeda dengan hari jumat yang jam masuk pukul 07.30 dan keluar pukul 16.30, dengan durasi kerja tujuh jam tiga puluh menit ditambah satu setengah jam istirahat.

Proses kehadiran pegawai non PNS Pusyantek dimulai dengan *tap* jari ke mesin *fingerprint*, yaitu sebelum jam masuk dan sesudah jam keluar yang sudah ditentukan. Kemudian data kehadiran dari mesin masuk ke komputer yang sudah ada aplikasi mesin *fingerprint*. Pengolah mengolah data yang sudah tersimpan dalam dengan cara mengekspor data dalam format *excel*. Pengolah mengolah data pegawai satu persatu menggunakan *Microsoft Excel* dengan memasukkan rumus. Proses tersebut dilakukan satu per satu untuk 25 pegawai yang ada di Pusyantek setiap periode. Dalam satu *file* berisi 500 baris data kehadiran dan 29 kolom. Tentu hal ini menimbulkan beberapa masalah. yaitu menimbulkan data yang tidak akurat, pemanfaatan waktu yang tidak efisien, dan lain-lain.

Dengan adanya permasalahan di atas, pengolah mengharapkan adanya aplikasi berbasis *website* yang memudahkan pengolah dalam mengolah data kehadiran. Pengolah pun berharap agar pegawai dapat dengan mudah mengakses kehadiran, mengajukan izin, dan mengajukan cuti. Sistem yang diharapkan pengolah adalah pengolah bisa langsung mengunggah *file* data kehadiran ke dalam sistem dan sistem akan langsung mengolah secara otomatis kolom apa saja yang diperlukan tanpa harus diolah terlebih dahulu.

1.2 Perencanaan

Tahap ini dilakukan untuk menganalisis kebutuhan dan masalah untuk mendapatkan fungsi yang akan dikembangkan. Penamaan kode untuk kebutuhan adalah SIAP-XXX singkatan dari kode fungsi, representasi dari kebutuhan fungsional dan SIAP-XXX-X kebutuhan sub-fungsional, SIAP diambil dari nama sistem. Kebutuhan data dapat dilihat pada Tabel 1,

kebutuhan fungsional pada Tabel 2, dan kebutuhan non fungsional pada Tabel 3.

Tabel 1 Analisis kebutuhan data *increment* satu

Nama	Primary	Store	Deskripsi
Absensi	Absensi_id	siap	Berisi informasi data kehadiran pegawai
Bulan	Bulan_id	siap	Berisi informasi periode bulan
Cuti	Cuti_id	siap	Berisi informasi pengajuan cuti pegawai
Izin	Izin_id	siap	Berisi informasi pengajuan izin pegawai
Jabatan	Jabatan_id	siap	Berisi informasi jabatan pegawai
Lampiran	Lampiran_id	siap	Berisi lampiran pengajuan izin dan cuti
Pegawai	Peg_no_akun	siap	Berisi informasi data pegawai
Tahun	Tahun_id	siap	Berisi informasi periode tahun
User	Username	siap	Berisi informasi user

Tabel 2 Analisis kebutuhan fungsional *increment* satu

Kode	Fungsional	Deskripsi
SIAP-001	<i>Login</i>	Pegawai, admin melakukan <i>login</i>
SIAP-002	<i>Logout</i>	Pegawai, admin melakukan <i>logout</i>
SIAP-003	Lihat absensi	Pegawai melihat daftar kehadiran
SIAP-004	Kelola pengajuan izin	Pegawai mengelola pengajuan izin
SIAP-004-1	Lihat pengajuan izin	Pegawai melihat pengajuan izin
SIAP-004-2	Tambah pengajuan izin	Pegawai menambah pengajuan izin
SIAP-004-3	Ubah pengajuan izin	Pegawai mengubah pengajuan izin
SIAP-004-4	Hapus pengajuan izin	Pegawai menghapus pengajuan izin
SIAP-004-5	Cetak pengajuan izin	Pegawai mencetak pengajuan izin
SIAP-005	Kelola pengajuan cuti	Pegawai mengelola pengajuan cuti
SIAP-005-1	Lihat pengajuan cuti	Pegawai melihat pengajuan cuti
SIAP-005-2	Tambah pengajuan cuti	Pegawai menambah pengajuan cuti
SIAP-005-3	Ubah pengajuan cuti	Pegawai mengubah pengajuan cuti
SIAP-005-4	Hapus pengajuan cuti	Pegawai menghapus pengajuan cuti
SIAP-005-5	Cetak pengajuan cuti	Pegawai mencetak pengajuan cuti
SIAP-006	Kelola persetujuan	Pegawai mengelola persetujuan
SIAP-006-1	Unggah persetujuan	Pegawai mengunggah persetujuan
SIAP-006-2	Hapus persetujuan	Pegawai menghapus persetujuan
SIAP-007	Kelola profil pegawai	Pegawai mengelola profil
SIAP-007-1	Ubah <i>username</i>	Pegawai mengubah <i>username</i>
SIAP-007-2	Ubah <i>password</i>	Pegawai mengubah <i>password</i>
SIAP-007-3	Ubah nama	Pegawai mengubah nama

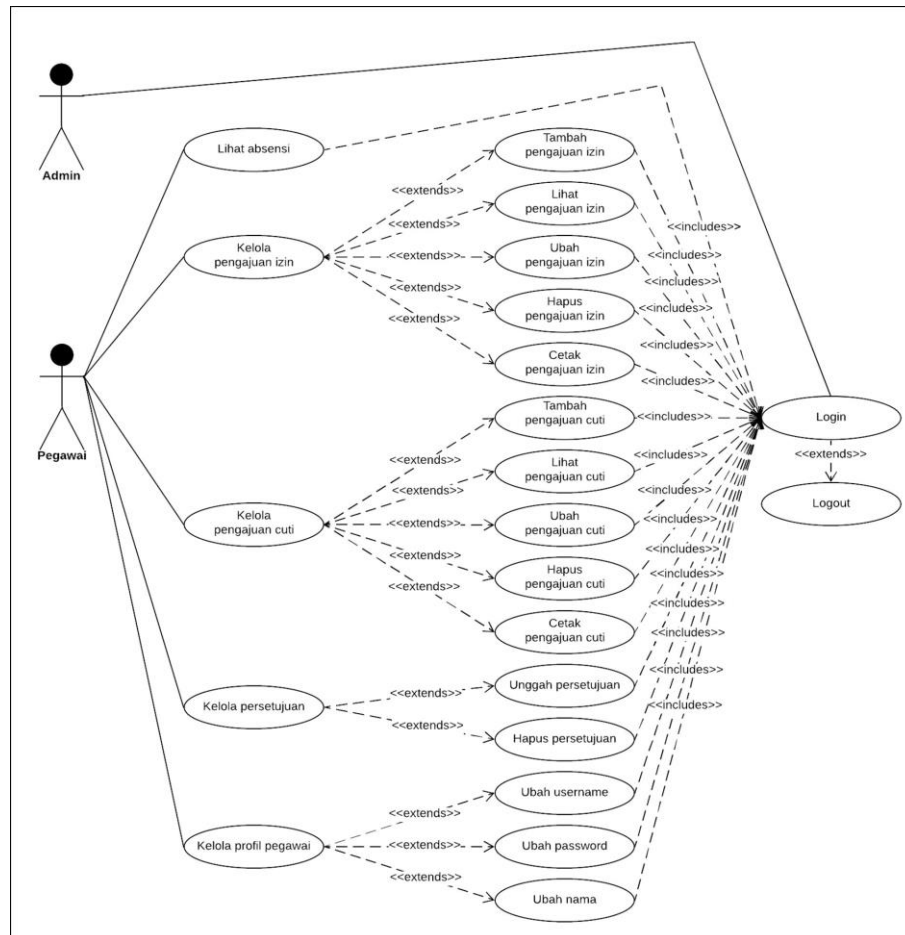
Tabel 3 Analisis kebutuhan non fungsional *increment* satu

Kode	Parameter	Deskripsi
SIAP-008	<i>Communication</i>	Bahasa Indonesia
SIAP-009	<i>Security</i>	Hak akses
SIAP-010	<i>Availability</i>	24 jam sehari, 7 hari seminggu.

1.3 Pemodelan

1.3.1 Use Case Diagram

Pada *incremental* pertama didahulukan pengerjaan fungsi untuk *user* pegawai. *Use case diagram* dapat dilihat pada Gambar 2. *Use case diagram* dijabarkan dengan deskripsi *use case*, salah satunya dapat dilihat pada Tabel 4.



Gambar 2 Use case diagram increment satu

1.3.2 Class Diagram

Perancangan *database* untuk sistem yang akan dibuat menghasilkan suatu *class diagram*. *Class diagram* berisikan rancangan relasi antara tabel sudah dianalisis keperluannya untuk sistem, dapat dilihat pada Gambar 3.

1.3.3 Antarmuka

Rancangan antarmuka bertujuan untuk memberikan gambaran seperti apa tampilan yang akan dibuat, penempatan *header* dan konten pada sistem. Salah satu rancangan antarmuka, yaitu absensi pegawai dapat dilihat pada Gambar 4.

1.3.4 Masukan

Rancangan masukan bertujuan mendeskripsikan masukan agar meminimalisir kesalahan pada pemasukan data. Salah satu rancangan masukan, yaitu tambah pengajuan izin dapat dilihat pada Gambar 5.

1.3.5 Proses

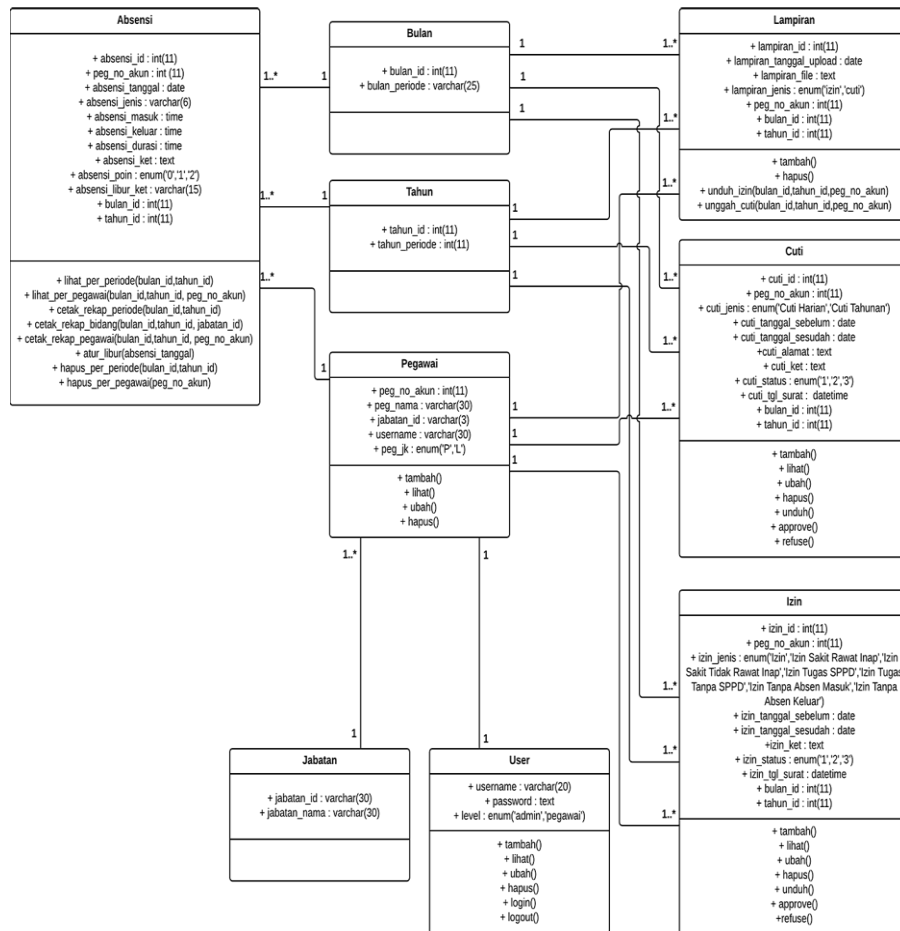
Rancangan proses adalah alur kerja dari *use case* yang sudah dibuat untuk menggambarkan aliran proses dan menghasilkan *activity diagram*. Salah satu *activity diagram* untuk sistem ini dapat dilihat pada Gambar 6.

1.3.6 Keluaran

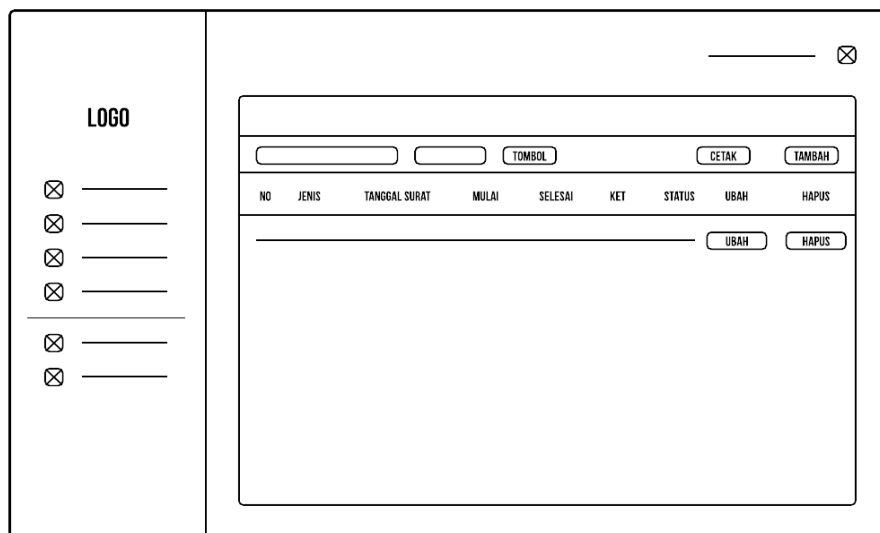
Rancangan keluaran dihasilkan dari *activity diagram* yang sudah dibuat sebelumnya. Rancangan keluaran berfungsi untuk menggambarkan hasil dari proses yang akan ditampilkan oleh sistem.

Tabel 4 Deskripsi *use case* tambah pengajuan izin di *increment* satu

<i>Use Case</i>	Deskripsi
<i>Use Case Name</i>	Tambah pengajuan izin
<i>Goal in Context</i>	Pegawai bisa menambahkan pengajuan izin dengan mengisi <i>form</i> yang sudah ada
<i>Precondition</i>	Masuk ke halaman pengajuan izin dengan periode <i>default</i> bulan dan tahun sekarang
<i>Successful End Condition</i>	Pegawai berhasil menambah pengajuan izin, selanjutnya system akan menampilkan data yang baru ditambahkan ke daftar data pengajuan izin
<i>Failed End Condition</i>	Muncul <i>warning</i> untuk mengisi <i>field</i> yang kosong
<i>Actor</i>	Pegawai
<i>Trigger</i>	Pegawai mengisi <i>form</i> yang berisi jenis, tanggal sebelum, tanggal sesudah, dan keterangan izin. Setelah selesai mengisi, pegawai memilih simpan
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Menekan tombol tambah 2. Mengisi jenis izin 3. Mengisi tanggal sebelum dan sesudah 4. Mengisi keterangan jika perlu 5. Menekan tombol simpan atau batal
<i>Extension</i>	<ol style="list-style-type: none"> 1. Jika <i>filed</i> masih kosong, akan muncul <i>warning</i> 2. Jika menekan batal maka fungsi dibatalkan

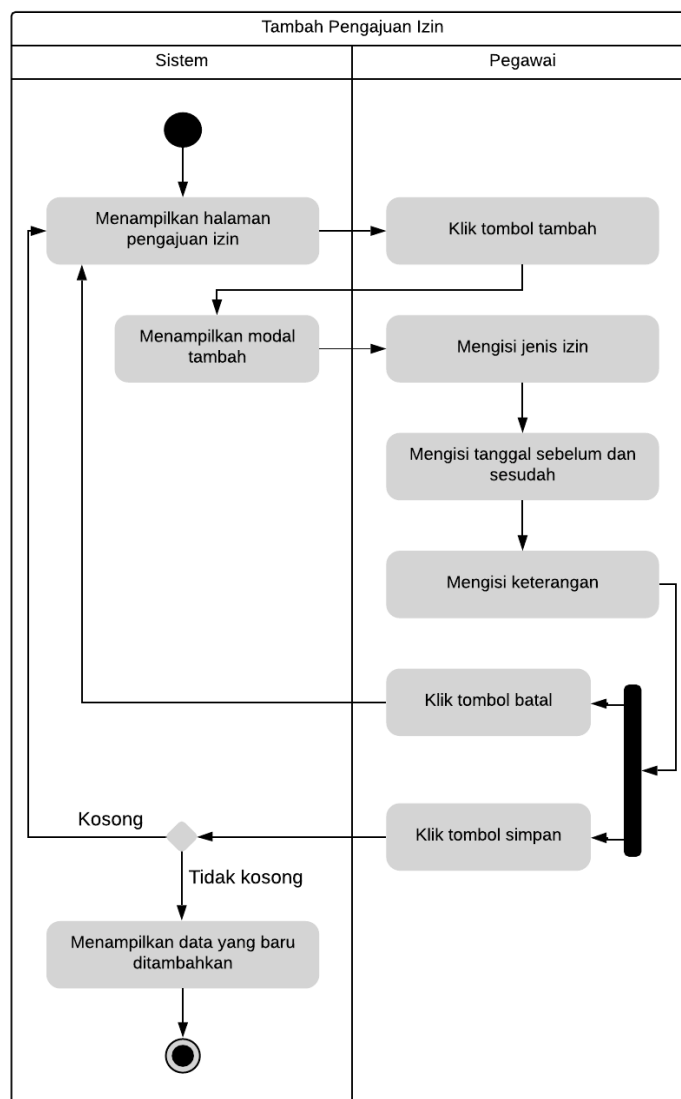


Gambar 3 Class diagram increment satu



Gambar 4 Rancangan antarmuka pengajuan izin increment satu

Gambar 5 Rancangan masukan tambah pengajuan izin *increment satu*



Gambar 6 Rancangan *activity diagram* tambah pengajuan izin

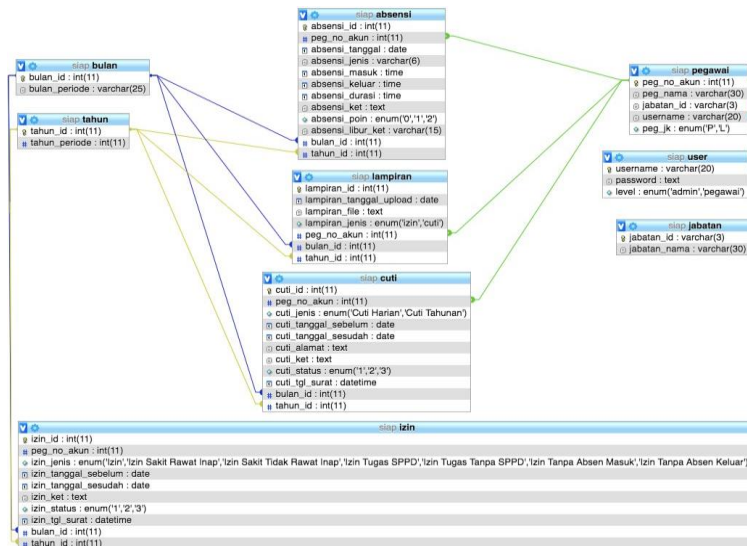
1.4 Pembuatan

Pembuatan sistem menggunakan laptop Macbook Air. Perangkat lunak yang digunakan dalam pembuatan sistem ini, yaitu:

- Sistem operasi : macOS High Sierra
- Bahasa pemrograman : PHP
- DBMS : MySQL
- Web Server : Apache
- Framework : CodeIgniter
- Web browser : Google Chrome dan Safari
- Text editor : Sublime Text

1.4.1 Implementasi Database

Tahap ini menghasilkan implementasi *database* berdasarkan kebutuhan data yang telah dibuat pada tahap pemodelan. Hasil implementasi dapat dilihat pada Gambar 7.



Gambar 7 Implementasi database

1.4.2 Implementasi Antarmuka

Tahapan implementasi antarmuka merupakan proses untuk menghasilkan implementasi dari antarmuka yang sudah dibuat sebelumnya dengan melakukan pengkodean. Implementasi antarmuka absensi dapat dilihat pada Gambar 8.

1.4.3 Implementasi Masukan

Tahapan implementasi masukan merupakan proses menghasilkan masukan yang sudah dirancang sebelumnya. Salah satu implementasi masukan, yaitu implementasi tambah pengajuan izin dapat dilihat pada Gambar 9.

1.4.4 Implementasi Proses

Tahapan implementasi proses untuk sistem ini menggunakan bantuan dari *framework* CodeIgniter dengan bahasa pemrograman PHP dan *editor* Sublime Text. *Framework* CodeIgniter menggunakan pola yang bernama *model-view-controller*, dimana pada proses implementasinya melakukan pengkodean yang dipisahkan menjadi *model*, *view*, dan *controller*.

1.4.5 Implementasi Keluaran

Implementasi keluaran pada sistem ini menghasilkan keluaran berupa tabel dan laporan rekap berformat PDF. Salah satu implementasi keluaran, yaitu implementasi keluaran absensi dapat dilihat pada Gambar 10.

NO	TANGGAL	JAM MASUK	JAM KELUAR	JAM DURASI	POKOK	KET
1	2019-01-10	-	-	-	0	-
2	2019-01-11	06:57:00	15:53:00	07:26:00	2	Hadir
3	2019-01-14	07:14:00	17:40:00	07:30:00	2	Hadir
4	2019-01-15	-	-	-	0	-

Gambar 8 Implementasi antarmuka absensi

Gambar 9 Implementasi masukan tambah pengajuan izin

NO	JENIS IZIN	TANGGAL SURAT	MULAI	SELESAI	KET	STATUS	UBAH	HAPUS
1	Izin	2019-06-02	2019-01-11	2019-01-11	-	Sudah disetujui	Ubah	Hapus
2	Izin Sakit Rawat Inap	2019-06-02	2019-01-12	2019-01-12	-	Ditolak	Ubah	Hapus
3	Izin Sakit Tidak Rawat Inap	2019-06-02	2019-01-13	2019-01-13	-	Belum disetujui	Ubah	Hapus
4	Izin Tugas SPPD	2019-06-02	2019-01-14	2019-01-14	-	Belum disetujui	Ubah	Hapus
5	Izin Tanpa Absen Pagi	2019-06-02	2019-01-16	2019-01-16	-	Belum disetujui	Ubah	Hapus
6	Izin Tanpa Absen Pulang	2019-06-02	2019-01-17	2019-01-17	-	Belum disetujui	Ubah	Hapus
No	Jenis Izin	Tanggal Surat	Mulai	Selesai	Ket	Status	Ubah	Hapus

Gambar 10 Implementasi keluaran pengajuan izin

1.5 Penyebaran dan Umpan Balik

Pada tahap ini dilakukan presentasi di hadapan klien berupa demo sistem yang berguna untuk pengujian terhadap sistem. Pengujian bertujuan untuk mencari kesalahan saat menjalankan sistem. Pengujian pada tahap ini menggunakan metode *black box*. Pengujian *login* dapat dilihat pada Tabel 5.

Tabel 1 Pengujian fungsi *login* di *increment* satu

Nilai Masukan	Skenario Pengujian	Hasil yang Diharapkan	Hasil
Benar	Pengguna mengisi <i>form</i> dengan masukan yang benar, klik masuk	Berhasil masuk sesuai hak akses	Sukses
Salah	Pengguna mengisi <i>form</i> dengan masukan <i>username</i> atau <i>password</i> salah, klik masuk	Gagal masuk, muncul <i>alert</i>	Sukses
Kosong	Pengguna tidak mengisi <i>form</i> , klik tombol masuk	Muncul <i>alert</i> jika <i>form</i> kosong	Sukses
<i>Username</i> kosong	Pengguna mengisi <i>form</i> dengan <i>username</i> kosong dan <i>password</i> terisi, klik tombol masuk	Muncul <i>alert</i> jika <i>username</i> kosong	Sukses
<i>Password</i> kosong	Pengguna mengisi <i>form</i> dengan <i>username</i> terisi dan <i>password</i> kosong, klik tombol masuk	Muncul <i>alert</i> jika <i>password</i> kosong	Sukses

2. *Increment* kedua

2.1 Komunikasi

Pada tahap ini dilakukan konsultasi lanjutan mengenai fungsi tambahan yang hendak dibuat pada *increment* kedua. Sistem yang diharapkan pengolah yaitu pengolah dapat langsung mengunggah *file* data kehadiran ke dalam sistem dan sistem akan langsung mengolah secara otomatis kolom apa saja yang diperlukan tanpa harus diolah. Setelah melalui beberapa kali konsultasi didapatkan kesimpulan kebutuhan yang diinginkan oleh klien. Berikut fitur yang diharapkan pengolah:

- Pengolah dapat mengekspor *file* kehadiran ke sistem dalam format *.xlsx*.
- Pengolah dapat melihat data kehadiran pegawai per periode (hari, tanggal, jam masuk, jam keluar, durasi, poin, dan keterangan.)
- Pengolah dapat melihat dan mencetak rekap kehadiran (nama pegawai, jumlah masuk, sakit, izin, cuti, tanpa keterangan, tidak lengkap, SPPD, tanpa SPPD, total durasi, kinerja kerja, dan poin) tiap periode.
- Pengolah menyetujui atau menolak pengajuan izin dan cuti yang dibuat.

2.2 Perencanaan

Pada tahap perencanaan *increment* kedua, didapatkan analisis kebutuhan data untuk *user* admin yang berjumlah 28 fungsi. Berikut adalah hasil yang didapatkan dari tahap komunikasi yang dilakukan pada tahap sebelumnya. Pada kebutuhan data, ditambahkan satu tabel bernama *upload_absensi* yang berguna untuk menyimpan data unggah absensi periode, dapat dilihat pada Tabel 6. Kebutuhan fungsional pada sistem kehadiran dapat dilihat pada Tabel 7.

Tabel 6 Analisis kebutuhan data di *increment* dua

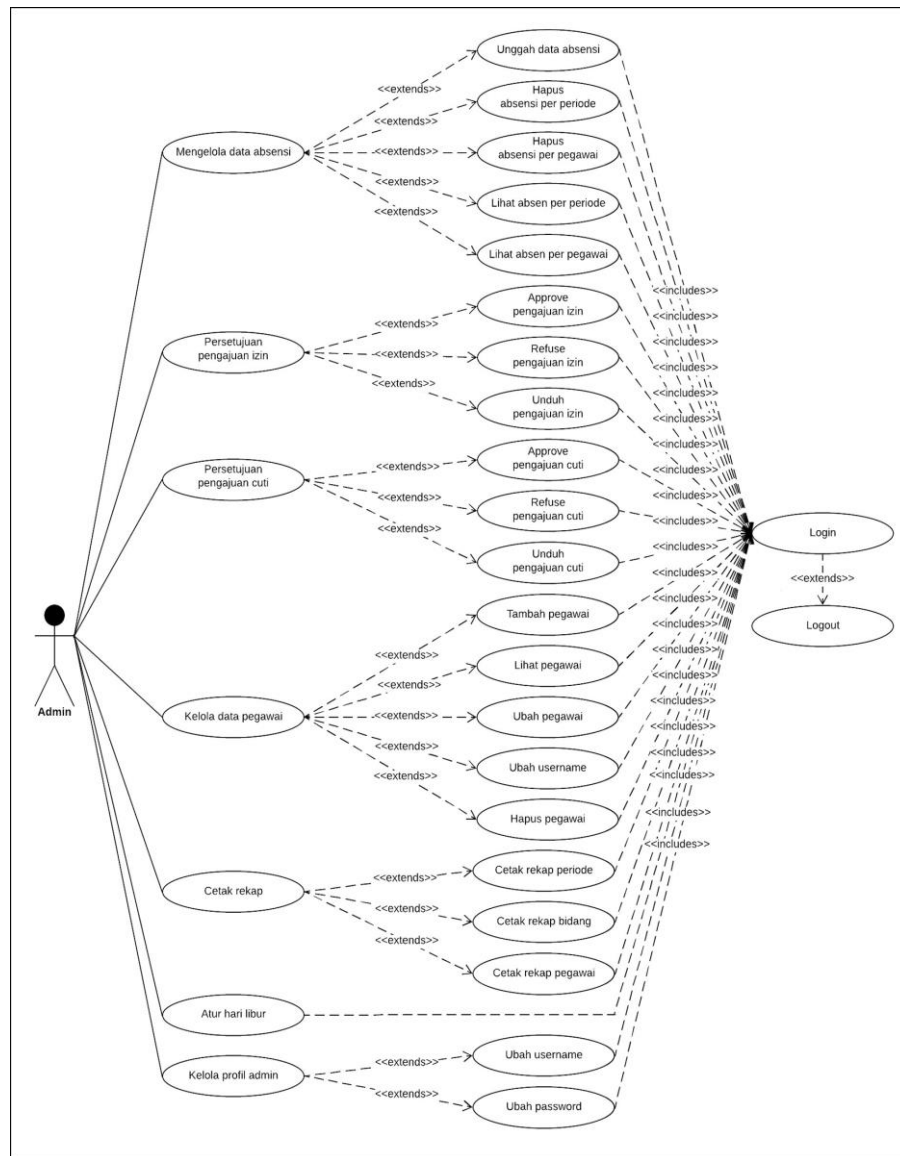
Nama	Primary Store	Deskripsi
<i>Upload_absensi</i>	<i>Upload_id</i> siap	Informasi unggahan absensi periode

Tabel 7 Analisis kebutuhan fungsional di *increment* dua

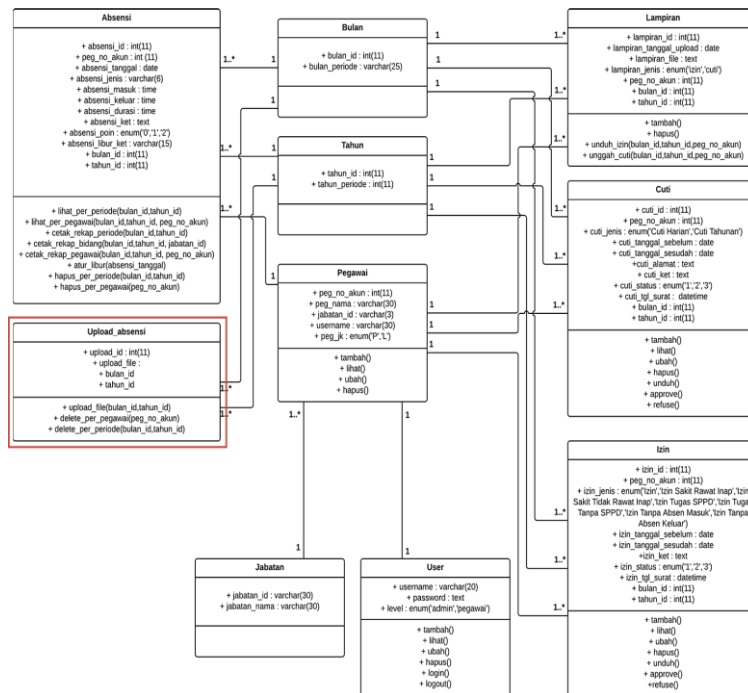
Kode	Fungsional	Deskripsi
SIAP-011	Kelola data absensi	Admin mengelola data kehadiran
SIAP-011-1	Unggah data absensi	Admin mengimport data kehadiran
SIAP-011-2	Hapus absensi periode	Admin menghapus absensi periode
SIAP-011-3	Hapus absensi pegawai	Admin menghapus absensi pegawai
SIAP-011-4	Lihat absensi per periode	Admin melihat data kehadiran
SIAP-011-5	Lihat absensi per pegawai	Admin melihat data kehadiran
SIAP-012	Persetujuan pengajuan izin	Admin mengelola pengajuan cuti
SIAP-012-1	<i>Approve</i> pengajuan izin	Admin menyetujui pengajuan cuti
SIAP-012-2	<i>Refuse</i> pengajuan izin	Admin menolak pengajuan cuti
SIAP-012-3	Unduh pengajuan izin	Admin mengunduh pengajuan cuti
SIAP-013	Persetujuan pengajuan cuti	Admin mengelola pengajuan cuti
SIAP-013-1	<i>Approve</i> pengajuan cuti	Admin menyetujui pengajuan cuti
SIAP-013-2	<i>Refuse</i> pengajuan cuti	Admin menolak pengajuan cuti
SIAP-013-3	Unduh pengajuan cuti	Admin mengunduh pengajuan cuti
SIAP-014	Kelola data pegawai	Admin mengelola data pegawai
SIAP-014-1	Lihat pegawai	Admin menilai pegawai
SIAP-014-2	Tambah pegawai	Admin menambah data pegawai
SIAP-014-3	Ubah pegawai	Admin mengubah data pegawai
SIAP-014-4	Ubah <i>username</i> pegawai	Admin mengubah <i>username</i>
SIAP-014-5	Hapus pegawai	Admin menghapus pegawai
SIAP-015	Cetak rekap	Admin mencetak rekap
SIAP-015-1	Cetak rekap periode	Admin mencetak rekap per periode
SIAP-015-2	Cetak rekap bidang	Admin mencetak rekap per bidang
SIAP-015-3	Cetak rekap pegawai	Admin mencetak rekap per pegawai
SIAP-016	Atur hari libur	Admin mengatur hari libur
SIAP-017	Kelola profil admin	Admin mengelola profil admin
SIAP-017-1	Ubah <i>username</i>	Admin mengubah <i>username</i>
SIAP-017-2	Ubah <i>password</i>	Admin mengubah <i>password</i>

2.3 Pemodelan

Pada tahap pemodelan, dilakukan pemodelan kembali untuk iterasi yang kedua. Pada pemodelan iterasi kedua ditambahkan beberapa fungsi baru. Fungsi baru yang telah dianalisis sebelumnya dibentuk menjadi *class diagram*, antarmuka, masukan, proses dalam bentuk *activity diagram*, dan keluaran. *Use case diagram* untuk *increment* kedua dapat dilihat pada Gambar 12. *Class diagram* untuk *inrement* kedua mengakibatkan bertambahnya satu tabel, yaitu tabel *upload_absensi*. *Class diagram* dapat dilihat pada Gambar 13.



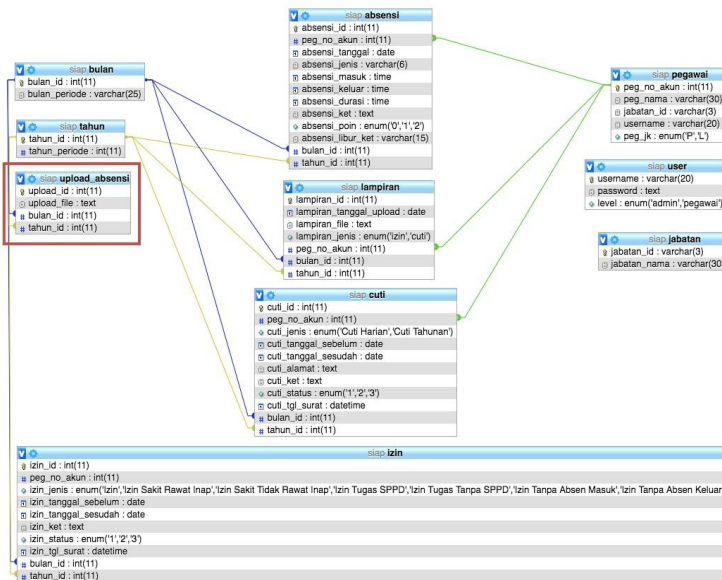
Gambar 11 Use case diagram increment dua



Gambar 12 Class diagram increment dua

2.4 Pembuatan

Sama seperti pada *inrement* pertama, dilakukan implementasi untuk *database*, proses, antarmuka, masukan, proses, dan keluaran. Implementasi pada *database* mengakibatkan penambahan tabel baru yaitu *upload_absensi* yang dapat dilihat pada Gambar 14.



Gambar 13 Implementasi *database* increment dua

2.5 Penyebaran dan Umpan Balik

Pada tahap ini juga dilakukan presentasi di hadapan klien berupa demo sistem untuk pengujian terhadap sistem seperti pada *increment pertama*.

SIMPULAN

Sistem Informasi Kehadiran Pegawai Non-PNS di Pusyantek BPPT merupakan sistem informasi yang mencakup kebutuhan kehadiran pegawai non- PNS. Setelah melewati tahap metode *incremental*, dapat disimpulkan bahwa fitur pengelolaan data kehadiran pegawai, fitur kehadiran pegawai, fitur kelengkapan data izin, dan fitur kelengkapan data cuti berhasil dibuat.

SARAN

Saran untuk sistem informasi kehadiran pegawai yaitu agar sistem dapat langsung terhubung dengan mesin *fingerpint*. Dengan demikian, pengolah tidak perlu mengunggah *file* data kehadiran ke dalam sistem secara manual. Sistem ini tidak menutup kemungkinan untuk dapat dikembangkan dengan menambahkan fitur yang lebih kompleks ke depannya.

DAFTAR PUSTAKA

- Brand AW. 2013. *PHP, MySQL, Javascript & HTML5 All-In-One For Dummies*. New Jersey(US): John Wiley & Sons.
- CodeIgniter. *Welcome to CodeIgniter* [Internet]. [Diunduh 2019 Februari 17]. Tersedia pada: www.codeigniter.com/user_guide/general/welcome.html.
- CodeIgniter. *Model-View-Controller* [Internet]. [Diunduh 2019 Juni 12]. Tersedia pada: www.codeigniter.com/user_guide/overview/mvc.html.
- Miles R, Hamilton K. 2006. *Learning UML 2.0*. United States of America(US) : O'Reilly Media, Inc.