BAB IV

ANALISIS DAN PERANCANGAN SISTEM

4.1. Analisis Sistem Yang Berjalan

Analisis sistem adalah penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan, sehingga dapat diusulkan perbaikan-perbaikan. Langkah-langkah analisis sistem antara lain :

1. *Identify*, yaitu memahami masalah.
2. *Understand*, yaitu memahami kerja dari sistem yang ada.
3. *Analyze,* yaitu menganalisis sistem.
4. *Report,* yaitu membuat laporan hasil analisis.

Tahap analisis ini sangat penting karena dalam tahap ini apabila terdapat kesalahan, maka akan menyebabkan kesalahan terhadap tahap selanjutnya. Maka perlu tingkat ketelitian dan kecermatan yang tinggi untuk mendapatkan kualitas kerja sistem yang baik. Analisis yang sedang berjalan bertujuan untuk mengetahui lebih jelas bagaimana cara kerja sistem tersebut dan masalah yang dihadapi sistem untuk dijadikan landasan usulan perancangan sistem.

4.1.1. Analisis Dokumen

Analisis Dokumen ini dilakukan guna untuk mengetahui dokumen apa saja yang menjadi *input*, proses, dan *output* dari sistem ini. Dokumen *input* adalah

dokumen yang akan diproses oleh sistem yang biasanya dilakukan oleh entitas luar sistem. Dokumen Proses adalah dokumen yang diperlukan oleh sistem dalam melakukan kegiatan pendafataran dan pengajuan, sedangkan yang menjadi Dokumen *Output* adalah dokumen yang dihasilkan oleh proses olahan sistem, yang berasal dari dalam sistem ke entitas luar. Dan dokumen-dokumen yang digunakan sebenarnya cukup banyak, baik itu yang berupa data dalam bentuk simpanan (*database*) maupun yang bisa dijadikan laporan maupun dokumen dalam bentuk fisik (masukan dan keluaran). Namun yang akan dibahas dan dijelaskan dalam analisa ini adalah dokumen yang dirasa cukup sering digunakan dan diperlukan dalam aktivitas karyawan di Mitra Dinamika Konsultan, dokumen tersebut diantaranya adalah :

1. Dokumen modul kerja.
2. Formulir informasi klien.
3. Slip gaji karyawan.

Tabel 4.1.

Analisis Dokumen

No	Nama Dokumen	Uraian
1	Dokumen modul kerja	a. Deskripsi : Dokumen ini merupakan dokumen yang berisikan informasi seputar modul kerja yang diberikan oleh *Project Leader* kepada *Programmer*. b. Fungsi : Sebagai kumpulan informasi yang bertujuan untuk membantu *Programmer* dalam

		mengerjakan program aplikasi yang sedang dibuat. c. Sumber : *Project Leader* d. Distribusi : *Programmer* e. Atribut : nomor_modul, nama_modul, tanggal_keluar_modul, keterangan.
2	Formulir informasi klien	a. Deskripsi : Sebuah formulir yang bertuliskan informasi dari klien seputar pertanyaan, keluhan dan permohonan perbaikan Program Aplikasi buatan Mitra Dinamika Konsultan yang telah digunakan oleh klien. b. Fungsi : Sebagai media untuk mencatat suatu informasi dari klien, agar *Programmer* Mitra Dinamika Konsultan dapat melihatnya dan segera mengerjakan apa menjadi kebutuhan dari klien tersebut. c. Sumber : Klien d. Distribusi : *Project Leader* dan *Programmer* e. Atribut : nomor_formulir, nomor_induk_klien, nama_klien, tanggal_pencatatan, keterangan
3	Slip gaji karyawan	a. Deskripsi : Merupakan sebuah slip gaji yang diberikan oleh *Project Leader* Mitra Dinamika Konsultan kepada para karyawannya setiap bulannya.

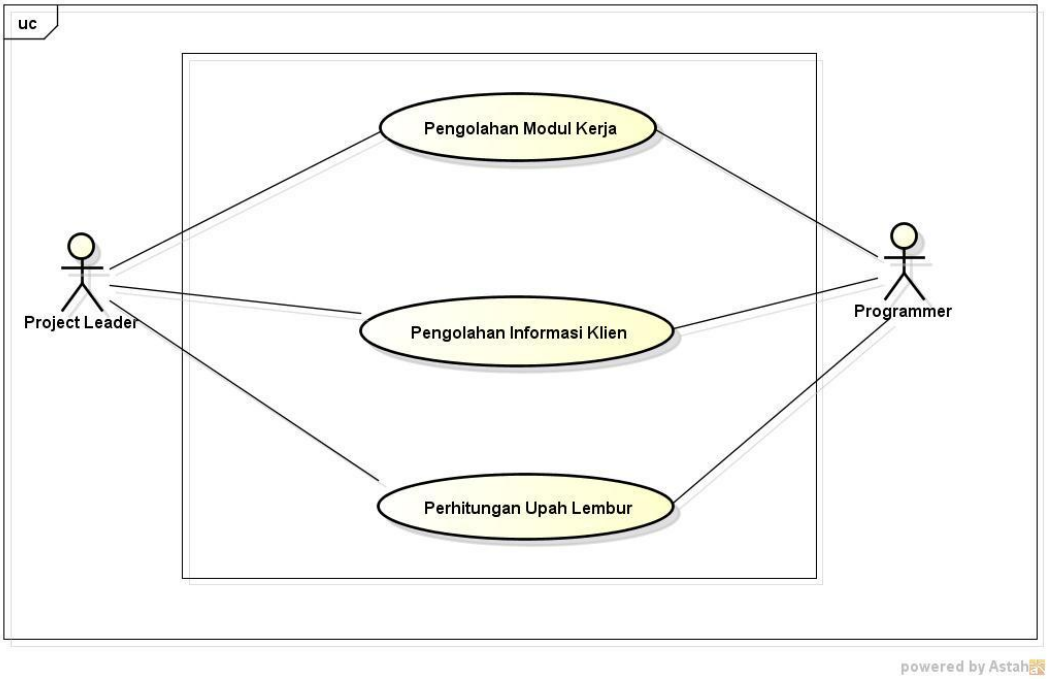
		b. Fungsi : Sebagai bukti gaji karyawan yang bertuliskan jumah gaji karyawan bersangkutan c. Sumber : *Project Leader* d. Distribusi : Karyawan e. Atribut : nomor_slip_gaji, bulan_slip_gaji, tanggal_penerimaan_slip_gaji, nomor_induk_karyawan, nama_karyawan, gaji_pokok, upah_lembur

4.1.2. Analisis Prosedur Yang Sedang Berjalan

Analisis sistem yang sedang berjalan bertujuan untuk mengetahui lebih jelas bagaimana cara kerja sistem tersebut dan masalah yang dihadapi sistem. Analisis sistem ini dapat dijadikan sebagai suatu landasan usulan perancangan sistem yang sedang berjalan yang dilakukan berdasarkan urutan kejadian yang ada serta dari urutan kejadian tersebut. Berikut merupakan analisis sistem yang sedang berjalan.

4.1.2.1. *Usecase Diagram*

Usecase Diagram dari sistem yang sedang berjalan, digambarkan sebagai berikut :



Gambar 4.1. *Usecase Diagram* Sistem Yang Sedang Berjalan

Dalam *usecase diagram* diatas, dapat terlihat terdapat 3 proses inti yang sedang berjalan di Mitra Dinamika Konsultan. Untuk lebih jelasnya dapat dijabarkan sebagai berikut :

1. Proses pertama, yaitu pengolahan modul kerja dimana *Project Leader* sebagai *Actor* yang hanya menerbitkan modul-modul melalui *e-mail* kepada para *Programmer* di Mitra Dinamika Konsultan. Lalu sesudah *Programmer* tersebut menerima modul tersebut, lalu mereka cek dan menyetujui untuk dikerjakan.
2. Proses kedua, yaitu pengolahan informasi klien dimana klien menyampaikan informasi seputar pertanyaan, keluhan dan permohonan perbaikan Program Aplikasi buatan Mitra Dinamika Konsultan yang telah digunakan oleh klien

kepada *Project Leader* atau kepada *Programmer*. Penyampaian informasi tersebut dapat langsung bertatap muka pada saat meeting atau dengan menggunakan telepon. Lalu *Project Leader* dan *Programmer* akan menangani kebutuhan klien tersebut dan kemudian dibuatlah suatu laporan atau dokumentasinya. Setelah itu *Project Leader* akan menyampaikan hasil pengerjaannya itu kepada klien dan klien menerimanya.

3. Proses ketiga, yaitu perhitungan upah lembur dimana para karyawan mencatat waktu lemburnya dan kemudian mengajukan klaim tersebut dengan cara mengirimkannya melalui *e-mail*. Lalu setelah itu *Project Leader* akan menghitungnya, dan *Project Leader* akan memberikannya sekalian pada saat gajian tiba.

4.1.2.2. Skenario *Activity Diagram*

a) Skenario *Activity Diagram* Sistem Pengolahan Modul Kerja Yang Sedang Berjalan

1. Nama *Usecase* : Pengolahan Modul Kerja
2. Aktor : *Project Leader* dan *Programmer*
3. Tipe : *Primary*
4. Tujuan : Melakukan pengolahan modul agar tersampaikan dengan jelas.

Tabel 4.2.

Skenario Pengolahan Modul Kerja Yang Sedang Berjalan

No : A01	
Nama	Pengolahan Modul Kerja

Tujuan	Melakukan pengolahan modul agar tersampaikan dengan jelas	
Deskripsi		
Tipe	*Primary*	
Aktor	*Project Leader* dan *Programmer*	
Skenario Utama		
Kondisi Awal	-	
Aksi Aktor		**Reaksi Sistem**
1. *Project Leader* membuat modul kerja		
2. *Project Leader* menerbitkannya melalui *e-mail* kepada *Programmer* yang bersangkutan		
3. *Programmer* akan menerima modul tersebut untuk dicek : a) Jika "tidak sesuai", maka kembali ke proses (1) b) Jika "sesuai", maka lanjut ke proses berikutnya		
4. *Programmer* tersebut akan menyetujui untuk mengerjakan modul tersebut		
5. *Programmer* akan mengerjakan modul tersebut		
Kondisi Akhir		Modul kerja terselesaikan dengan baik

b) Skenario *Activity Diagram* Sistem Pengolahan Informasi Klien Yang Sedang Berjalan

1. Nama *Usecase* : Pengolahan Informasi Klien.
2. Aktor : Klien, *Project Leader* dan *Programmer*.
3. Tipe : *Primary.*
4. Tujuan : Agar keinginan klien terpenuhi.

Tabel 4.3.

Skenario Pengolahan Informasi Klien Yang Sedang Berjalan

No : A02		
Nama	Pengolahan Informasi Klien	
Tujuan	Agar keinginan klien terpenuhi	
Deskripsi		
Tipe	*Primary*	
Aktor	Klien, *Project Leader* dan *Programmer*	
Skenario Utama		
Kondisi Awal	-	
Aksi Aktor		**Reaksi Sistem**
1. Klien menyampaikan informasi seputar pertanyaan, keluhan dan permohonan perbaikan Program Aplikasi buatan Mitra Dinamika Konsultan yang telah digunakan		
2. *Project Leader* atau *Programmer* mencatat informasi tersebut pada sebuah formulir khusus untuk informasi-informasi klien		

3. *Project Leader* dan *Programmer* akan mengerjakan kebutuhan klien tersebut	
4. Setelah selesai, *Project Leader* dan *Programmer* membuat suatu laporan atau dokumentasinya	
5. Klien menerima hasil pengerjaan tersebut	
Kondisi Akhir	Kebutuhan klien melalui informasi yang disampaikan tersebut dapat terpenuhi

c) Skenario *Activity Diagram* Sistem Perhitungan Upah Lembur Yang Sedang Berjalan

1. Nama *Usecase* : Perhitungan Upah Lembur.
2. Aktor : *Project Leader* dan karyawan.
3. Tipe : *Primary.*
4. Tujuan : Karyawan dapat menerima upah lembur.

Tabel 4.4.

Skenario Perhitungan Upah Lembur Yang Sedang Berjalan

No : A03	
Nama	Perhitungan Upah Lembur
Tujuan	Karyawan dapat menerima upah lembur

Deskripsi	
Tipe	*Primary*
Aktor	*Project Leader* dan Karyawan
Skenario Utama	
Kondisi Awal	-

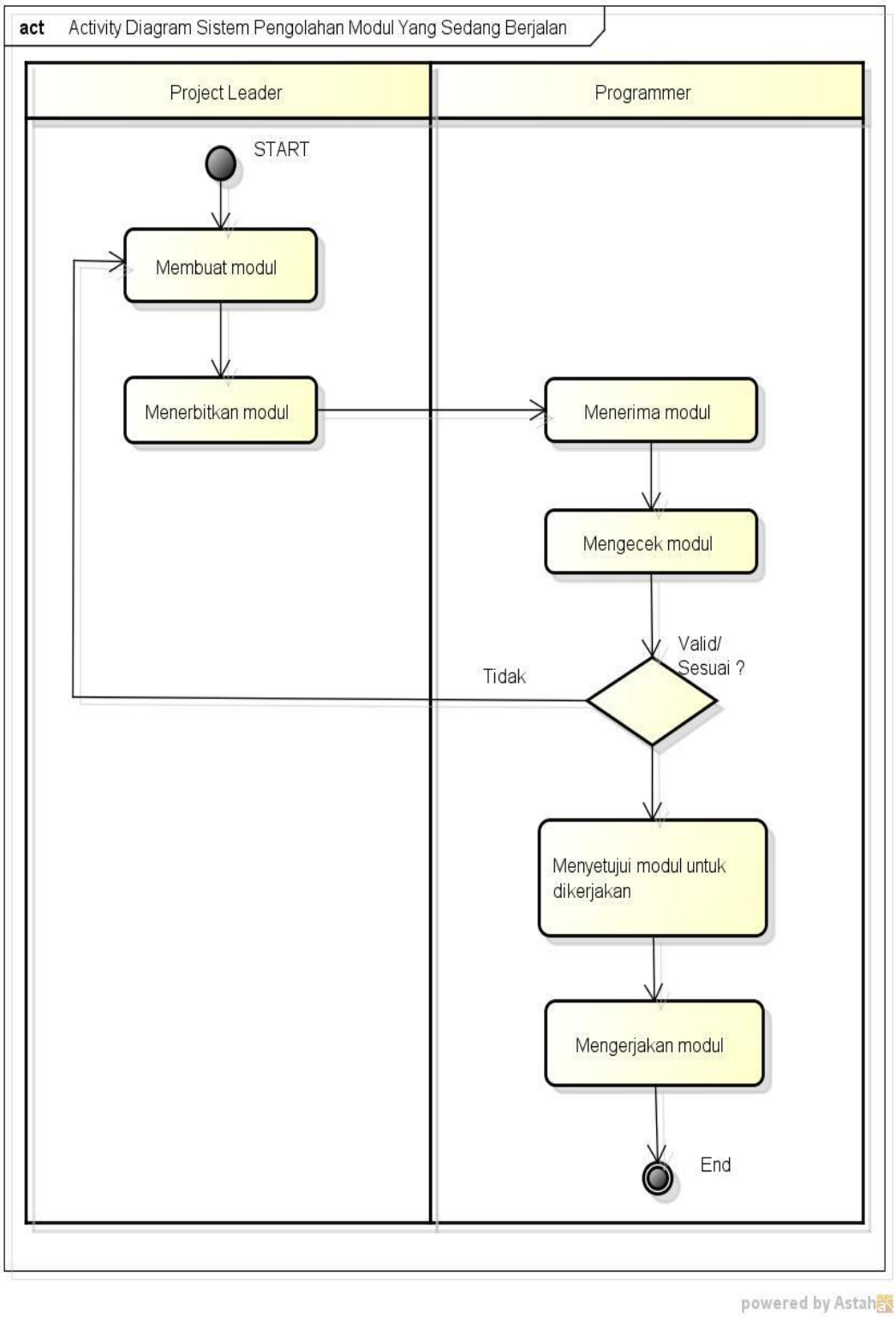
Aksi Aktor	**Reaksi Sistem**
1. Karyawan mencatat waktu lembur/ klaim lembur	
2. Karyawan mengajukan klaim tersebut dengan cara mengirimkannya melalui *e-mail*.	
	3. *Project Leader* akan menghitung klaim-klaim karyawan
	4. *Project Leader* akan memberikan upah lembur tersebur sekalian pada saat waktu gajian
5. Karyawan menerima upah lembur tersebut	
Kondisi Akhir	Perhitungan upah lembur dapat dilakukan dengan baik dan karyawan dapat menerimanya.

4.1.2.3. *Activity Diagram*

1. *Activity Diagram* Sistem Pengolahan Modul Yang Sedang Berjalan

Proses pengolahan modul di Mitra Dinamika Konsultan secara umum dapat dijelaskan sebagai berikut :

a) *Project Leader* akan membuat modul tersebut.
b) Lalu *Project Leader* menerbitkannya kepada *Programmer* yang bersangkutan. Modul tersebut akan dikirim menggunakan *e-mail* perusahaan. Contoh alamat *e-mail* karyawannya : *erick.hermawan@mitradinamikakonsultan.com*.
c) Lalu *Programmer* akan menerima modul tersebut untuk dicek, apabila modul tersebut memang sesuai maka kemudian *Programmer* tersebut akan menyetujuinya dengan membalas *e-mail*. Apabila modul tersebut tidak sesuai/ salah kirim, maka *Programmer* juga akan membalas *e-mail* tersebut dengan memberikan alasan mengapa modul tersebut tidak sesuai.
d) Dan kemudian *Programmer* akan mengerjakan modul tersebut.

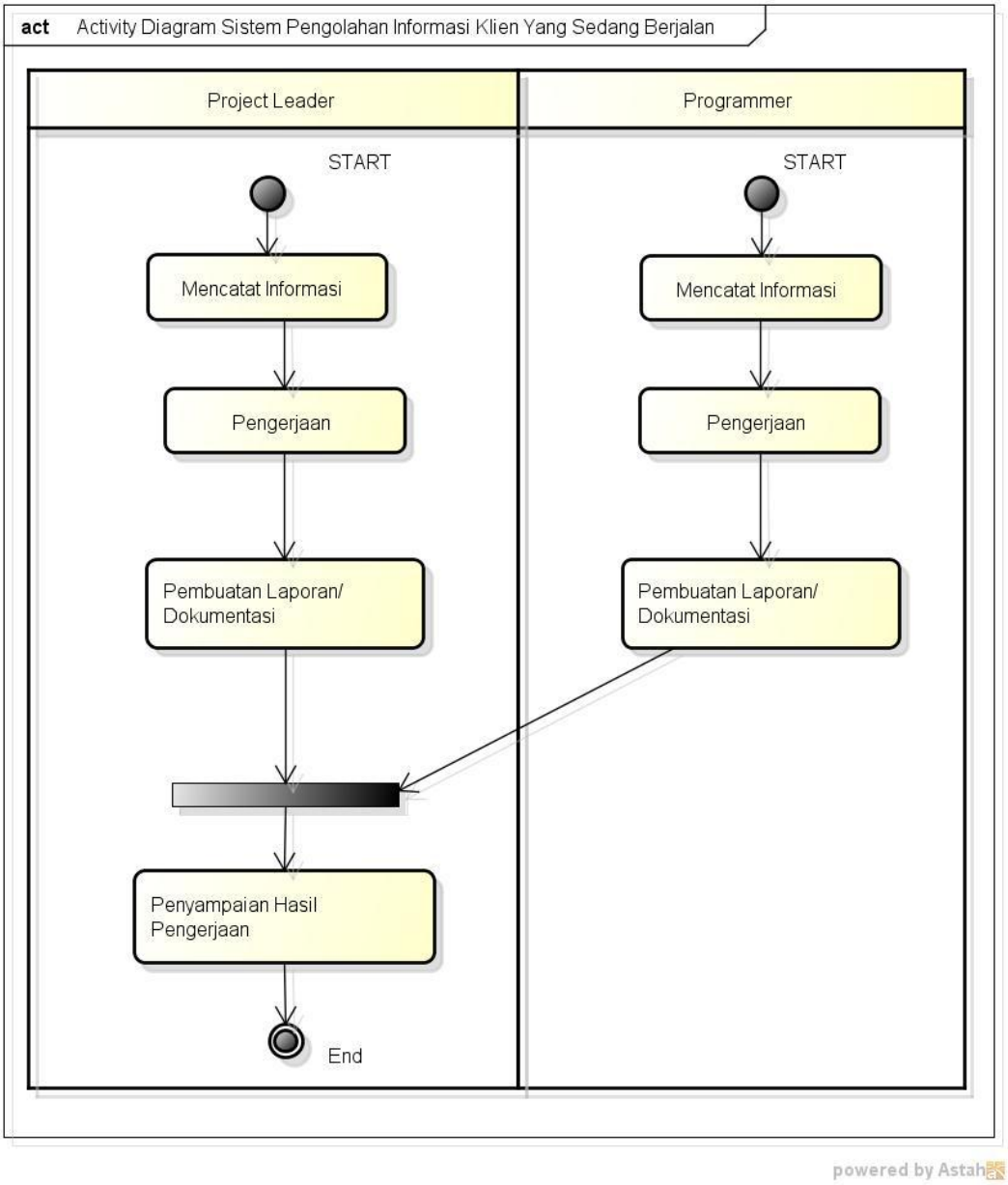


Gambar 4.2. *Activity Diagram* Pengolahan Modul Yang Sedang Berjalan

2. *Activity Diagram* Sistem Pengolahan Informasi Klien Yang Sedang Berjalan

Proses pengolahan informasi klien di Mitra Dinamika Konsultan secara umum dapat dijelaskan sebagai berikut :

a) Dimulai dari klien yang menyampaikan informasi seputar pertanyaan, keluhan dan permohonan perbaikan Program Aplikasi buatan Mitra Dinamika Konsultan yang telah digunakan oleh klien kepada *Project Leader* atau kepada *Programmer*. Penyampaian informasi tersebut dapat langsung bertatap muka pada saat meeting atau dengan menggunakan telepon.
b) Lalu *Project Leader* dan *Programmer* akan mencatat informasi tersebut pada sebuah formulir khusus untuk informasi-informasi klien.
c) Kemudian *Project Leader* dan Programmer akan menangani kebutuhan klien tersebut dengan mengerjakan bagian mana yang menjadi keinginan klien tersebut.
d) Dan setelah selesai pengerjaannya, dibuatlah suatu laporan atau dokumentasinya.
e) Lalu *Project Leader* akan menyampaikan hasil pengerjaan tersebut kepada klien.
f) Klien menerima hasil pengerjaan tersebut.



Gambar 4.3. *Activity Diagram* Pengolahan Informasi Klien Yang Sedang Berjalan

3. *Activity Diagram* Sistem Perhitungan Upah Lembur Yang Sedang Berjalan

Proses perhitungan upah lembur di Mitra Dinamika Konsultan secara umum dapat dijelaskan sebagai berikut :

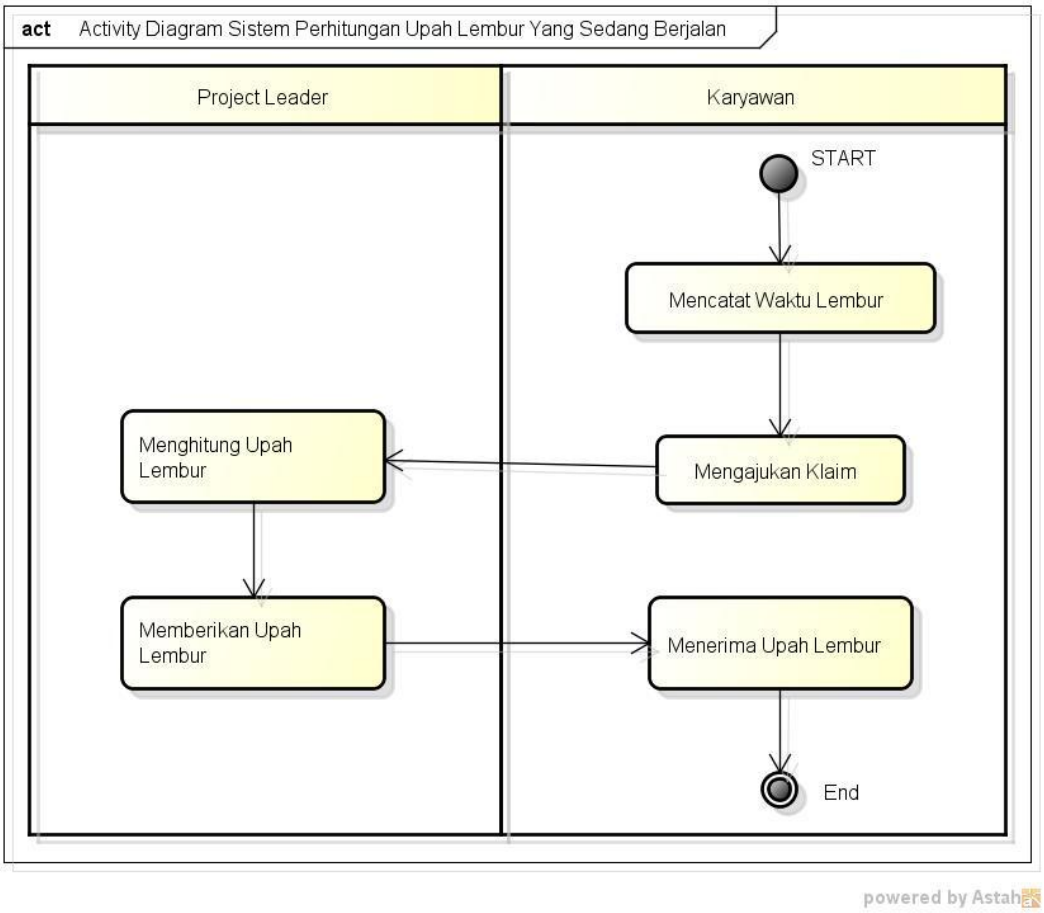
a) Para *Programmer* mencatat waktu lemburnya.

b) Kemudian mereka mengajukan klaim tersebut dengan cara mengirimkannya melalui *e-mail*.

c) Lalu setelah itu *Project Leader* akan menghitungnya.

d) Kemudian *Project Leader* akan memberikannya sekalian pada saat gajian tiba.

e) Para *Programmer* akan menerima upah lembur tersebut.



Gambar 4.4. *Activity Diagram* Perhitungan Upah Lembur Yang Sedang Berjalan

4.1.3. Kesimpulan Analisis Prosedur Yang Sedang Berjalan

Berdasarkan penjabaran analisis mengenai sistem yang sedang berjalan diatas, dapat terlihat bahwa sistem-sistem tersebut masih terpisah dan berjalan sendiri-sendiri. Kebanyakan proses-prosesnya tersebut memerlukan koneksi *internet* untuk mengirimkan maupun menerima *e-mail*. Sehingga akan mengalami masalah apabila koneksi *internet* tersebut mengalami gangguan, sehingga dapat mengganggu proses kerja di Mitra Dinamika Konsultan.

Belum adanya sistem-sistem yang saling terkait dalam sebuah Program Aplikasi khusus yang mampu menangani sistem-sistem tersebut agar terintegrasi dengan baik. Dan belum adanya sistem absensi yang diterapkan di Mitra Dinamika Konsultan.

Dari analisis tersebut, maka perlunya dibuatkan suatu Program Aplikasi yang dapat mengintegrasikan sistem-sistem tersebut menjadi satu dan sekaligus menerapkan suatu sistem absensi. Program Aplikasi tersebut akan memudahkan *Project Leader* dan *Programmer* dalam mengolah modul-modul kerja, mengolah informasi klien, perhitungan upah lembur akan semakin jelas dan dapat terkontrol dengan baik serta sistem absensi tersebut yang akan membuat para karyawan Mitra Dinamika Konsultan menjadi disiplin dalam bekerja. Semua kegiatan sistem-sistem yang sedang berjalan tersebut dapat tercakup, diakses dan diselesaikan melalui Program Aplikasi yang akan dibuat.

4.2. Perancangan Sistem

Adapun tujuan dan gambaran dari perancangan sistem yang dikembangkan, yaitu:

4.2.1. Tujuan Perancangan

Perancangan sistem ini dilakukan setelah tahap analisis sistem berjalan dikerjakan, dalam hal ini telah didapatkan gambaran dengan jelas apa yang harus dikerjakan. Pada dasarnya tahap perancangan ini bertujuan untuk:

1. Untuk memenuhi kebutuhan kepada pemakai sistem.
2. Untuk memberikan gambaran yang jelas dan rancang bangun yang lengkap kepada pemogram komputer dan ahli-ahli teknik lainnya seputar program aplikasi yang akan dibuat.

4.2.2. Gambaran Umum Sistem Yang Diusulkan

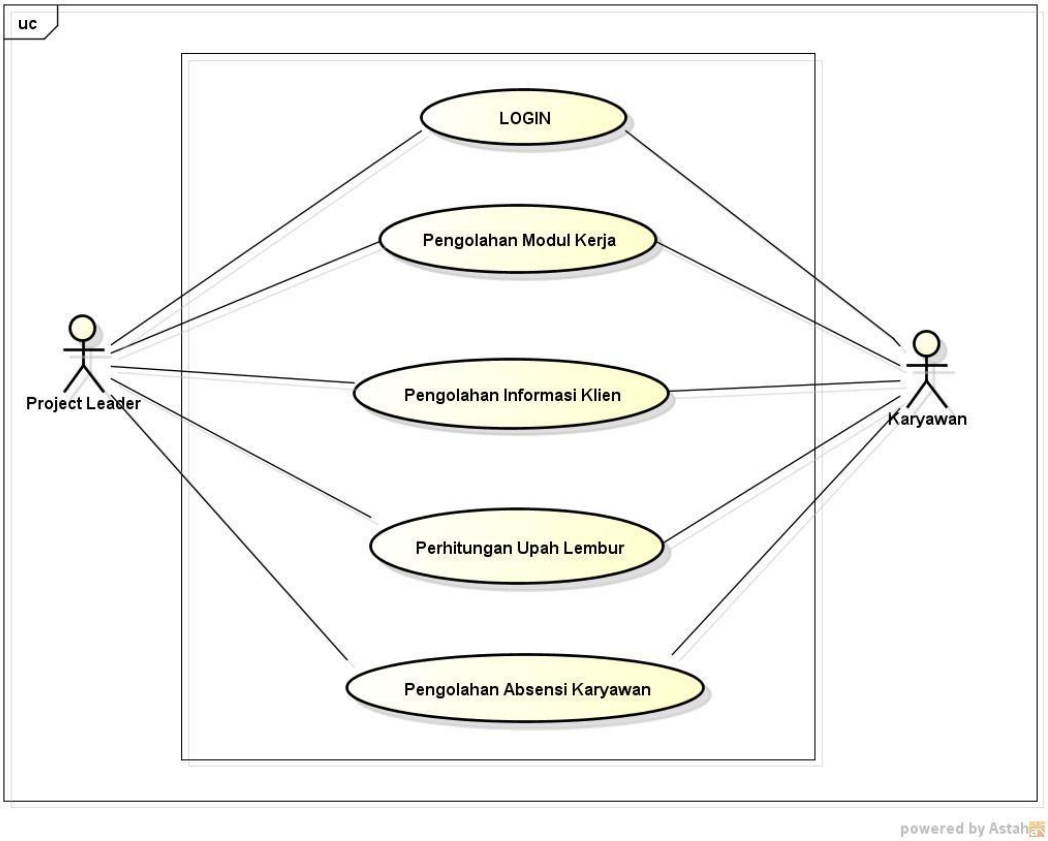
Perancangan aplikasi pembantu dan pendukung kinerja karyawan ini menggunakan bahasa UML (*Unified Modeling Language*). Mulai dari pembuatan rancangan *Usecase* nya hingga *Deployment* Diagramnya. Secara umum, proses tersebut dimulai dari penentuan arsitektur utama dari sistem yang ingin dirancang dan dibuat diagram alur proses pendaftaran sebagai Admin (Administrator/Operator) yang mana dalam kasus ini adalah *Project Manager* itu sendiri dan dilanjutkan alur proses utamanya, yang berupa pengolahan modul kerja, pengolahan informasi klien, perhitungan upah lembur hingga pengolahan absensi karyawan.

4.2.3. Perancangan Prosedur Yang Diusulkan

Usulan atau rancangan dari analisis sistem yang akan dikembangkan adalah sebagai berikut :

4.2.3.1. *Usecase Diagram*

Usecase diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *Usecase* mempresentasikan sebuah interaksi antara Aktor dengan sistem. Berikut adalah perancangan *Usecase* nya:



Gambar 4.5. *Usecase Diagram* Sistem Yang Diusulkan

Dalam *usecase diagram* diatas, dapat terlihat terdapat 5 proses inti yang akan diusulkan di Mitra Dinamika Konsultan. Untuk lebih jelasnya dapat dijabarkan sebagai berikut :

4. Proses pertama, yaitu aktivitas *login* yang terlebih dahulu harus dilakukan oleh *Project Leader* dan karyawan untuk dapat masuk ke Program Aplikasi ini.
5. Proses kedua, yaitu pengolahan modul kerja dimana *Project Leader* sebagai *Actor* yang hanya menerbitkan modul-modul melalui program aplikasi tersebut kepada para karyawan di Mitra Dinamika Konsultan. Lalu sesudah itu karyawan tersebut menerima modul, lalu mereka cek dan menyetujui untuk dikerjakan.
6. Proses ketiga, yaitu pengolahan informasi klien dimana klien menyampaikan informasi seputar pertanyaan, keluhan dan permohonan perbaikan Program Aplikasi buatan Mitra Dinamika Konsultan yang telah digunakan oleh klien kepada *Project Leader* atau kepada karyawan. Penyampaian informasi tersebut dapat langsung bertatap muka pada saat meeting atau dengan menggunakan telepon. Lalu *Project Leader* dan karyawan akan menangani kebutuhan klien tersebut dan kemudian dibuatlah suatu laporan atau dokumentasinya. Setelah itu *Project Leader* akan menyampaikan hasil pengerjaannya itu kepada klien dan klien menerimanya. Proses tersebut akan diinputkan kedalam program aplikasi ini agar dapat didokumentasikan untuk memudahkan proses penyampaian informasi kepada klien, informasi

perkembangan pengerjaannya hingga mendapatkan outputan berupa laporan pengerjaannya.

7. Proses keempat, yaitu perhitungan upah lembur dimana para karyawan mencatat waktu lemburnya di dalam program aplikasi ini dan kemudian mengajukan klaim tersebut dengan cara mensubmitkan . Lalu setelah itu klaim tersebut di program aplikasi ini dan *Project Leader* akan menghitungnya, lalu *Project Leader* akan memberikannya sekalian pada saat gajian tiba. *Output* yang akan dihasilkan dari proses ini adalah laporan upah lembur karyawan perbulannya.
8. Dan terakhir yaitu proses kelima adalah pengolahan absensi karyawan, dimana sistem ini sebelumnya belum ada di Mitra Dinamika Konsultan, oleh karena itu sistem absensi ini akan sangat penting untuk diterapkan, karena dapat meningkatkan kedisiplinan dari para karyawannya. Karyawan-karyawan Mitra Dinamika Konsultan akan menginputkan data kehadiran setiap hari kerja dengan menggunakan program aplikasi ini disaat mereka datang di pagi hari untuk memulai kerja. Fitur ini juga nantinya akan dapat diakses melalui *Handphone* para karyawan yang mempunyai fitur *wi-fi* nya, dimana khusus penginputan data kehadiran dapat diakses melalui *Handphone*. Outputan yang akan dihasilkan dari proses ini adalah rekapan absensi.

4.2.3.2. Skenario *Diagram*

Skenario yang diusulkan adalah sebagai berikut :

1. Skenario *Diagram Login* Yang Diusulkan
   a. Nama *Usecase* : *Login.*

b. Aktor : *Project Leader* dan *Karyawan.*

c. Tipe : *Primary.*

d. Tujuan : Agar dapat masuk dan mengakses Program Aplikasi.

Tabel 4.5.

Skenario *Login* Yang Diusulkan

No : A01		
Nama	*Login*	
Tujuan	Agar dapat masuk dan mengakses Program Aplikasi	
Deskripsi		
Tipe	*Primary*	
Aktor	*Project Leader* dan Karyawan	
Skenario Utama		
Kondisi Awal	-	
Aksi Aktor		**Reaksi Sistem**
6. *Project Leader* / Karyawan memasukkan *username* dan *password*		
		7. Sistem akan mengecek *username* dan *password*
8. Mendapatkan pesan dari sistem : a) Apabila *username / password* tidak *valid*, maka memasukkan *username/ password* lagi atau keluar dari program aplikasi		

b) Apabila *username/ password valid*, maka *Project Leader*/ Karyawan dapat mengakses Program Aplikasi	
Kondisi Akhir	*Project Leader* / Karyawan dapat masuk dan mengakses Program Aplikasi

2. Skenario *Diagram* Sistem Pengolahan Modul Kerja Yang Diusulkan

a. Nama *Usecase* : Pengolahan Modul Kerja.

b. Aktor : *Project Leader* dan *Karyawan.*

c. Tipe : *Primary.*

d. Tujuan : Melakukan pengolahan modul agar tersampaikan dengan jelas.

Tabel 4.6.

Skenario Pengolahan Modul Kerja Yang Diusulkan

No : A02		
Nama	Pengolahan Modul Kerja	
Tujuan	Melakukan pengolahan modul agar tersampaikan dengan jelas	
Deskripsi		
Tipe	*Primary*	
Aktor	*Project Leader* dan Karyawan	
Skenario Utama		
Kondisi Awal	-	
Aksi Aktor		**Reaksi Sistem**

1. *Project Leader* membuat modul kerja	
2. *Project Leader* menerbitkannya melalui Form Modul Kerja di Program Aplikasi kepada Karyawan yang bersangkutan beserta nama modul dan penjelasannya	
	3. Sistem dapat menyimpan, mengubah,menghapus, mencari dan menyortir modul kerja beserta penjelasannya
4. Karyawan akan mencari modul untuk dirinya dengan memasukkan *username* dan *password* kemudian dapat mengunduh modul tersebut untuk dicek : c) Jika "tidak sesuai", maka kembali ke proses (1) d) Jika "sesuai", maka lanjut ke proses berikutnya	
5. Karyawan tersebut akan menyetujui untuk mengerjakan modul tersebut	
6. Karyawan akan mengerjakan modul tersebut	
Kondisi Akhir	Modul kerja terselesaikan dengan baik

3. Skenario *Diagram* Sistem Pengolahan Informasi Klien Yang Diusulkan

a. Nama *Usecase* : Pengolahan Informasi Klien.

b. Aktor : Klien, *Project Leader* dan Karyawan.

c. Tipe : *Primary.*

d. Tujuan : Agar keinginan klien terpenuhi.

Tabel 4.7.

Skenario Pengolahan Informasi Klien Yang Diusulkan

<table>
<tr><td colspan="3">No : A03</td></tr>
<tr><td>Nama</td><td colspan="2">Pengolahan Informasi Klien</td></tr>
<tr><td>Tujuan</td><td colspan="2">Agar keinginan klien terpenuhi</td></tr>
<tr><td colspan="3">Deskripsi</td></tr>
<tr><td>Tipe</td><td colspan="2">Primary</td></tr>
<tr><td>Aktor</td><td colspan="2">Klien, Project Leader dan Karyawan</td></tr>
<tr><td colspan="3">Skenario Utama</td></tr>
<tr><td>Kondisi Awal</td><td colspan="2">-</td></tr>
<tr><td colspan="2">Aksi Aktor</td><td>Reaksi Sistem</td></tr>
<tr><td colspan="2">6. Proses pengolahan informasi klien dimulai pada saat klien sepakat bekerjasama dengan pihak Mitra Dinamika Konsultan dan
klien dapat menyampaikan informasi seputar pertanyaan, keluhan dan permohonan perbaikan Program Aplikasi buatan Mitra Dinamika Konsultan yang telah digunakan</td><td></td></tr>
<tr><td colspan="2">7. Project Leader atau Karyawan menginputkan informasi tersebut pada Form Informasi Klien di Program Aplikasi</td><td></td></tr>
</table>

	8. Sistem dapat menyimpan, mengubah,menghapus, mencari dan menyortir informasi-informasi Klien
9. *Project Leader* dan Karyawan akan mengerjakan kebutuhan klien tersebut	
	10. Sistem akan mengolah informasi-informasi yang sudah masuk untuk mengubahnya menjadi laporan yang dapat didownload
11. Klien menerima hasil pengerjaan tersebut	
Kondisi Akhir	Kebutuhan klien melalui informasi yang disampaikan tersebut dapat terpenuhi

4. Skenario Diagram Sistem Perhitungan Upah Lembur Yang Diusulkan

   a. Nama *Usecase* : Perhitungan Upah Lembur.

   b. Aktor : *Project Leader* dan Karyawan.

   c. Tipe : *Primary.*

   d. Tujuan : Karyawan dapat menerima upah lembur.

Tabel 4.8.

Skenario Perhitungan Upah Lembur Yang Diusulkan

<table>
<tr><td colspan="3">No : A04</td></tr>
<tr><td>Nama</td><td colspan="2">Perhitungan Upah Lembur</td></tr>
<tr><td>Tujuan</td><td colspan="2">Karyawan dapat menerima upah lembur</td></tr>
<tr><td colspan="3">Deskripsi</td></tr>
<tr><td>Tipe</td><td colspan="2">Primary</td></tr>
<tr><td>Aktor</td><td colspan="2">Project Leader dan Karyawan</td></tr>
<tr><td colspan="3">Skenario Utama</td></tr>
<tr><td>Kondisi Awal</td><td colspan="2">-</td></tr>
<tr><td colspan="2">Aksi Aktor</td><td>Reaksi Sistem</td></tr>
<tr><td colspan="2">6. Karyawan menginputkan waktu lembur/ klaim lemburnya pada Form Klaim Upah Lembur di Program Aplikasi.</td><td></td></tr>
<tr><td colspan="2"></td><td>7. Sistem dapat menyimpan, mengubah,menghapus, mencari dan menyortir data pengajuan klaim lembur Karyawan</td></tr>
<tr><td colspan="2"></td><td>8. Sistem akan menjadikan rekapan klaim tersebut menjadi laporan yang akan tersampaikan ke Project Leader</td></tr>
</table>

9. *Project Leader* akan memberikan upah lembur tersebut pada saat waktu gajian	
10. Karyawan menerima upah lembur tersebut	
Kondisi Akhir	Perhitungan upah lembur dapat dilakukan dengan baik dan karyawan dapat menerimanya.

5. Skenario *Diagram* Sistem Pengolahan Absensi Karyawan Yang Diusulkan

a. Nama *Usecase* : Pengolahan Absensi Karyawan.

b. Aktor : *Project Leader* dan Karyawan.

c. Tipe : *Primary.*

d. Tujuan : Data Absensi Karyawan dapat terekap dengan baik.

Tabel 4.9.

Skenario Pengolahan Absensi Karyawan Yang Diusulkan

No : A05		
Nama	Pengolahan Absensi Karyawan	
Tujuan	Data Absensi Karyawan dapat terekap dengan baik	
Deskripsi		
Tipe	*Primary*	
Aktor	*Project Leader* dan Karyawan	
Skenario Utama		
Kondisi Awal	-	
Aksi Aktor		**Reaksi Sistem**
1. Karyawan menginputkan absensi pada Form		

Absensi di Program Aplikasi setiap pagi pada waktu hari kerja	
	2. Sistem dapat menyimpan, mengubah,menghapus, mencari dan menyortir data absensi Karyawan. Dan akan menjadikannya laporan/ rekapan absensi karyawan.
3. *Project Leader* akan menerima laporan/ rekapan absensi karyawan	
Kondisi Akhir	Data absensi karyawan dapat terekap untuk meningkatkan kedisiplinan karyawannya.

4.2.3.3. *Activity Diagram*

Activity Diagram yang diusulkan adalah sebagai berikut :

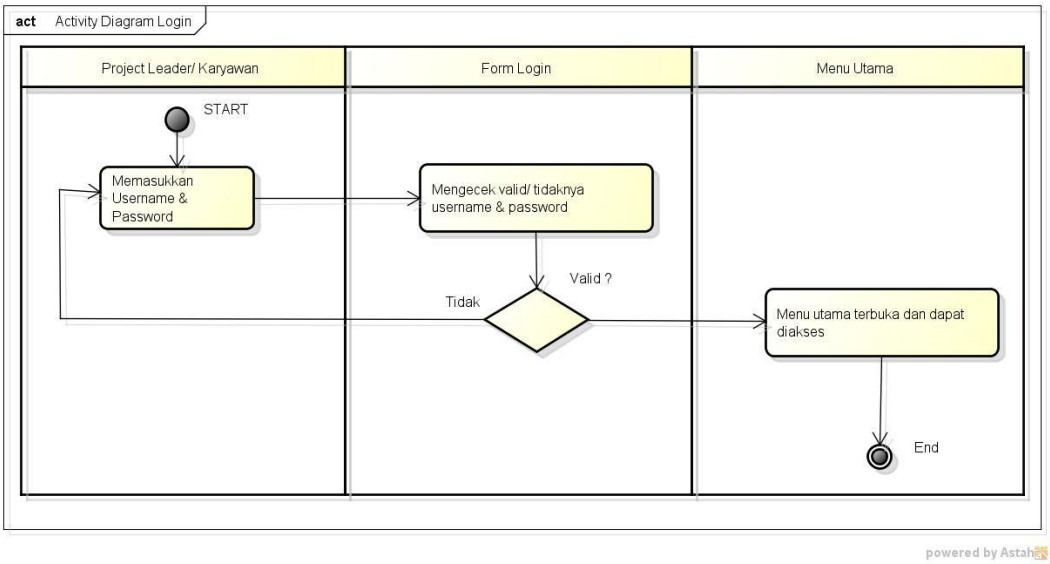
1. *Activity Diagram Login* Yang Diusulkan

Proses login yang diusulkan secara umum dapat dijelaskan sebagai berikut:

a. *Project Leader* atau karyawan *Username* dan *Password.*
b. Lalu di Form Login, akan mengecek apakah *Username* dan *Password valid/* tidak.

c. Apabila tidak valid, form login akan memberikan pesan, dan *Project Leader* atau karyawan dapat memasukkan kembali *Username* dan *Password* atau dapat keluar dari Program Aplikasi.

d. Apabila *valid*, maka masuk ke manu utama dan dapat mengakses Program Aplikasi tersebut.



Gambar 4.6. *Activity Diagram* Login Yang Diusulkan

2. Activity Diagram Sistem Pengolahan Modul Yang Diusulkan

Proses Pengolahan Modul yang diusulkan secara umum dapat dijelaskan sebagai berikut:

a. *Project Leader* membuat modul kerja.

b. *Project Leader* menerbitkannya melalui Form Modul Kerja di Program Aplikasi kepada Karyawan yang bersangkutan beserta nama modul dan penjelasannya.

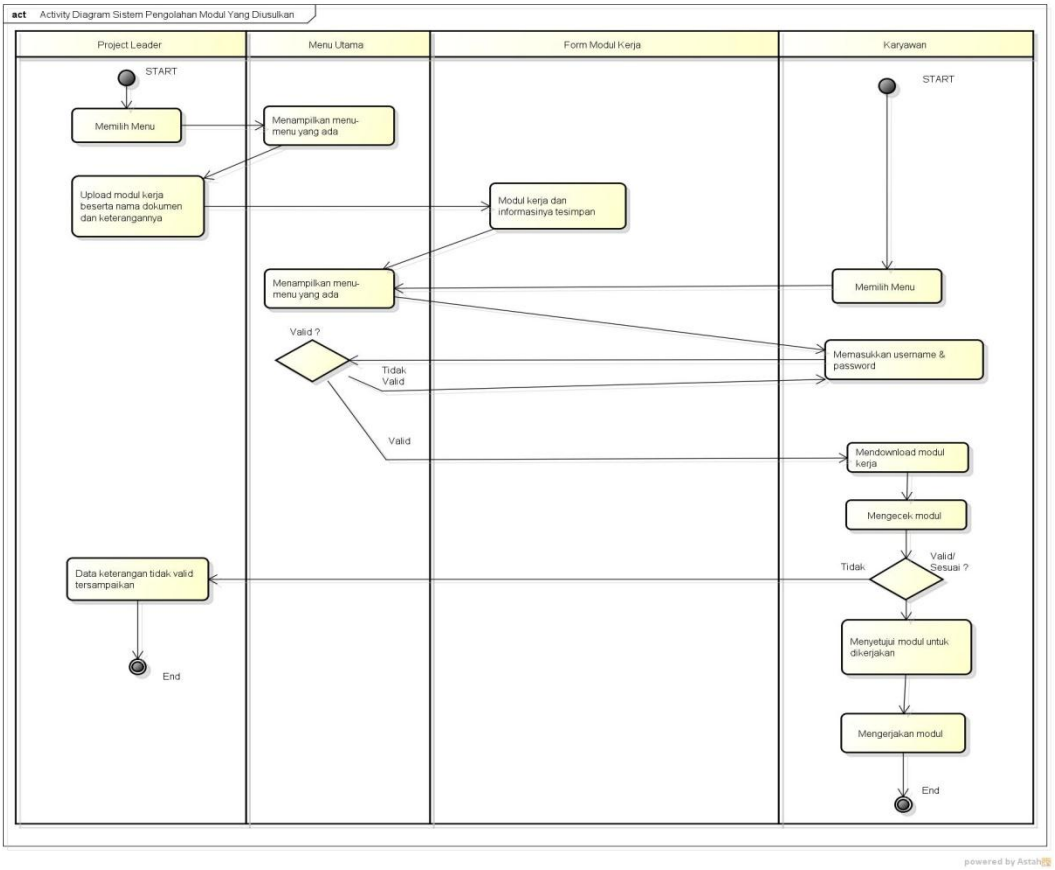
c. Sistem dapat menyimpan, mengubah,menghapus, mencari dan menyortir modul kerja beserta penjelasannya.

d. Karyawan akan mencari modul untuk dirinya dengan memasukkan *username* dan *password* terlebih dahulu kemudian dapat mengunduh modul tersebut untuk dicek.

1) Jika "tidak sesuai", maka kembali ke proses (1).
2) Jika "sesuai", maka lanjut ke proses berikutnya.

e. Karyawan tersebut akan menyetujui untuk mengerjakan modul tersebut.

f. Karyawan akan mengerjakan modul tersebut.

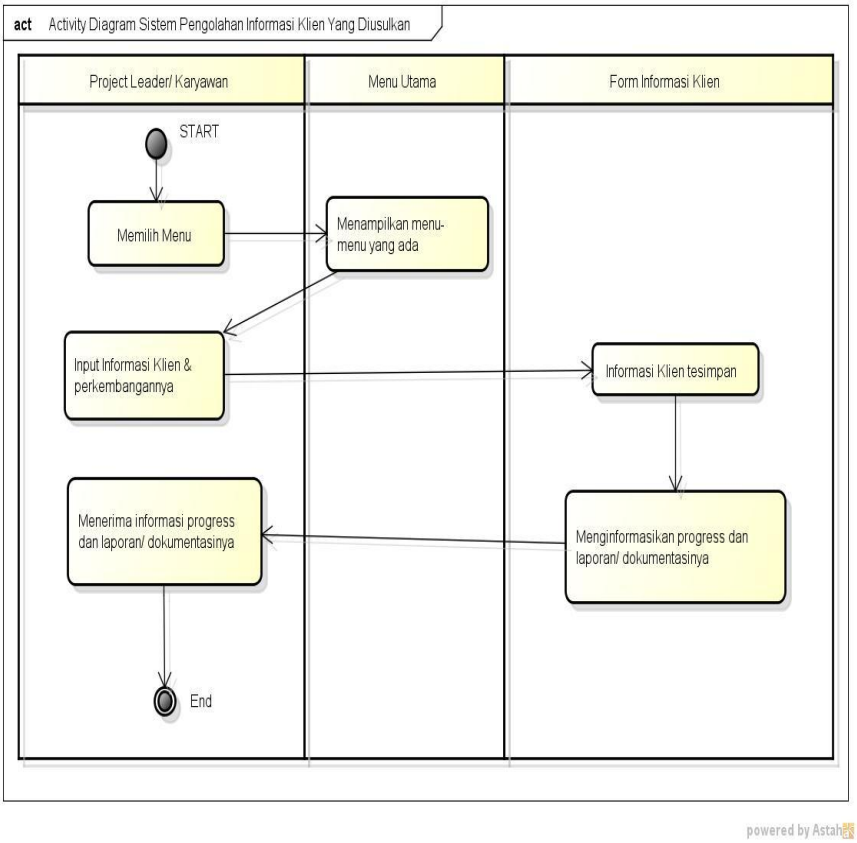


Gambar 4.7. *Activity Diagram* Pengolahan Modul Kerja Yang Diusulkan

3. *Activity Diagram* Sistem Pengolahan Informasi Klien Yang Diusulkan

Proses Pengolahan Informasi Klien yang diusulkan secara umum dapat dijelaskan sebagai berikut:

a. Proses pemgolahan informasi klien dimulai pada saat klien sepakat bekerjasama dengan pihak Mitra Dinamika Konsultan dan klien dapat menyampaikan informasi seputar pertanyaan, keluhan dan permohonan perbaikan Program Aplikasi buatan Mitra Dinamika Konsultan yang telah digunakanaikan Program Aplikasi buatan Mitra Dinamika Konsultan yang telah digunakan.
b. *Project Leader* atau Karyawan menginputkan informasi tersebut pada Form Informasi Klien di Program Aplikasi.
c. Sistem dapat menyimpan, mengubah,menghapus, mencari dan menyortir informasi-informasi Klien.
d. *Project Leader* dan Karyawan akan mengerjakan kebutuhan klien tersebut.
e. Sistem akan mengolah informasi-informasi yang sudah masuk untuk mengubahnya menjadi laporan yang dapat didownload.
f. Klien menerima hasil pengerjaan tersebut.



Gambar 4.8. *Activity Diagram* Pengolahan Informasi Klien Yang Diusulkan

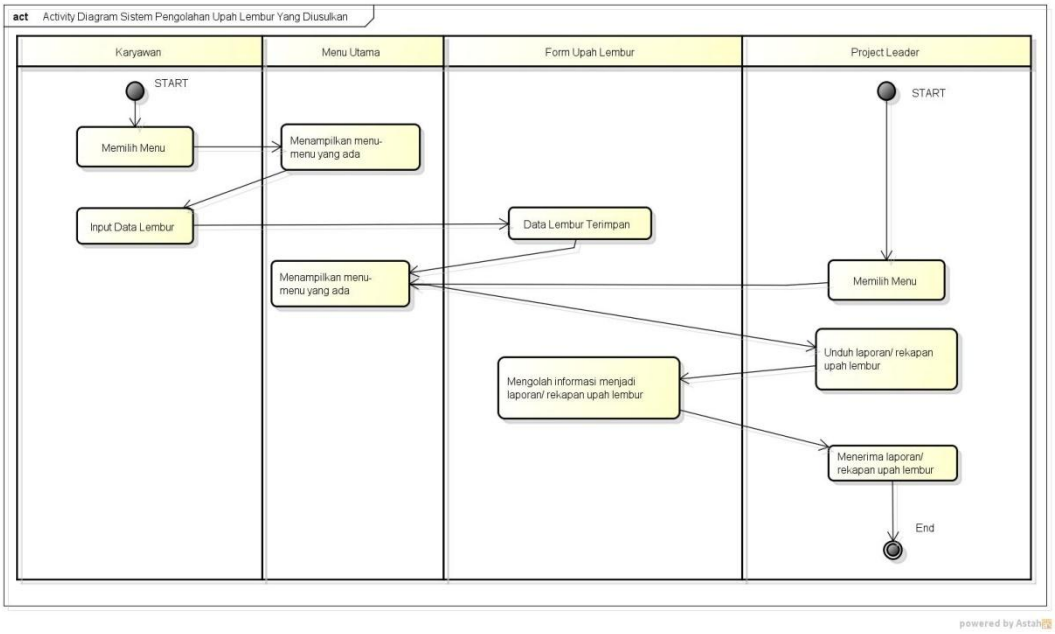
4. *Activity Diagram* Sistem Perhitungan Upah Lembur Yang Diusulkan

Proses Perhitungan Upah Lembur yang diusulkan secara umum dapat dijelaskan sebagai berikut:

a. Karyawan menginputkan waktu lembur/ klaim lembur nya pada Form Klaim Upah Lembur di Program Aplikasi.
b. Sistem dapat menyimpan, mengubah,menghapus, mencari dan menyortir data pengajuan klaim lembur Karyawan.
c. Sistem akan menjadikan rekapan klaim tersebut menjadi laporan yang akan tersampaikan ke *Project Leader.*

d. *Project Leader* akan memberikan upah lembur tersebut pada saat waktu gajian.

e. Karyawan menerima upah lembur tersebut.



Gambar 4.9. *Activity Diagram* Perhitungan Upah Lembur Yang Diusulkan

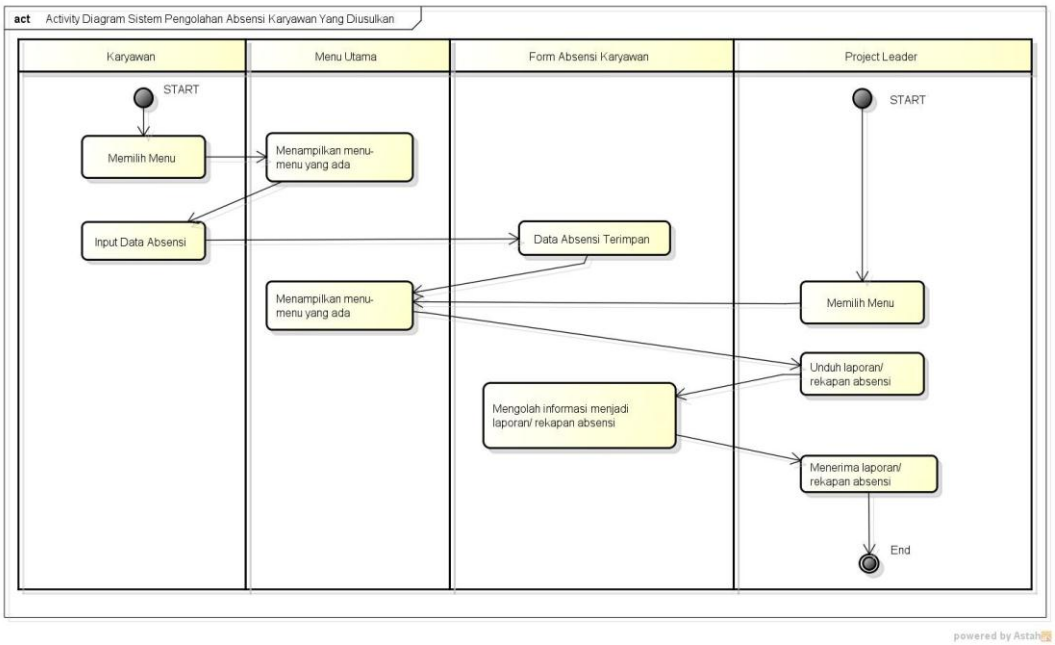
5. *Activity Diagram* Sistem Pengolahan Absensi Karyawan Yang Diusulkan

Proses Pengolahan Absensi Karyawan yang diusulkan secara umum dapat dijelaskan sebagai berikut:

a. Karyawan menginputkan absensi pada Form Absensi di Program Aplikasi setiap pagi pada waktu hari kerja.

b. Sistem dapat menyimpan, mengubah,menghapus, mencari dan menyortir data absensi Karyawan. Dan akan menjadikannya laporan/ rekapan absensi karyawan.

c. *Project Leader* akan menerima laporan/ rekapan absensi karyawan.



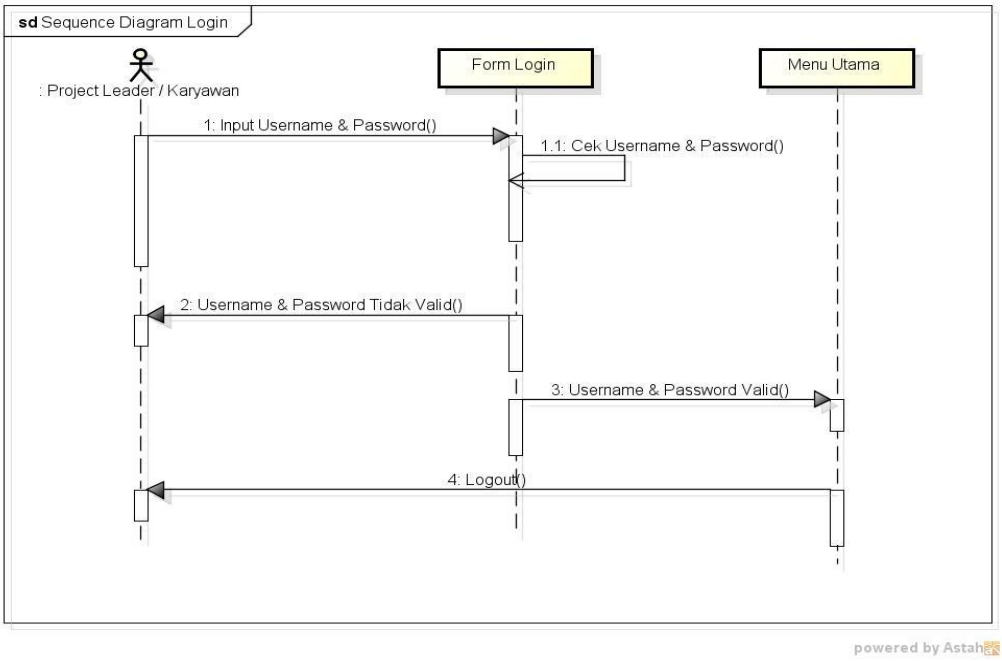
Gambar 4.10. *Activity Diagram* Pengolahan Absensi Karyawan Yang Diusulkan

4.2.3.4. *Sequence Diagram*

Sequence Diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan *output* tertentu. *Diagram* ini menunjukkan sejumlah contoh obyek dan *message* yang di letakkan di antara objek-objek di dalam usecase. Komponen utama *sequence diagram* terdiri dari objek yang di tulisakan dengan kotak segi empat bernama. *Message* di wakili oleh garis dengan tanda panah dan waktu yang di tunjukkan dengan progress *vertical*. Diawali dari apa yang mentriggerkan aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

1. *Sequence Diagram Login*

Gambar dibawah ini merupakan *Sequence Diagram Login* yang menggambarkan proses yang sedang terjadi dalam *Login Project Leader* atau karyawan. Prosesnya dimulai dengan admin masuk kedalam *form login* terlebih dahulu, kemudian selanjutnya *Project Leader* atau karyawan dapat melakukan *login* dengan memasukkan *Username* dan *Password* untuk dapat masuk kedalam sistem/ menu utama. Proses *Login* itu sendiri diikuti juga validasi apabila *Project Leader* atau karyawan salah memasukan *Username* dan *Password.* Dan diakhirkan dengan proses *logout*.

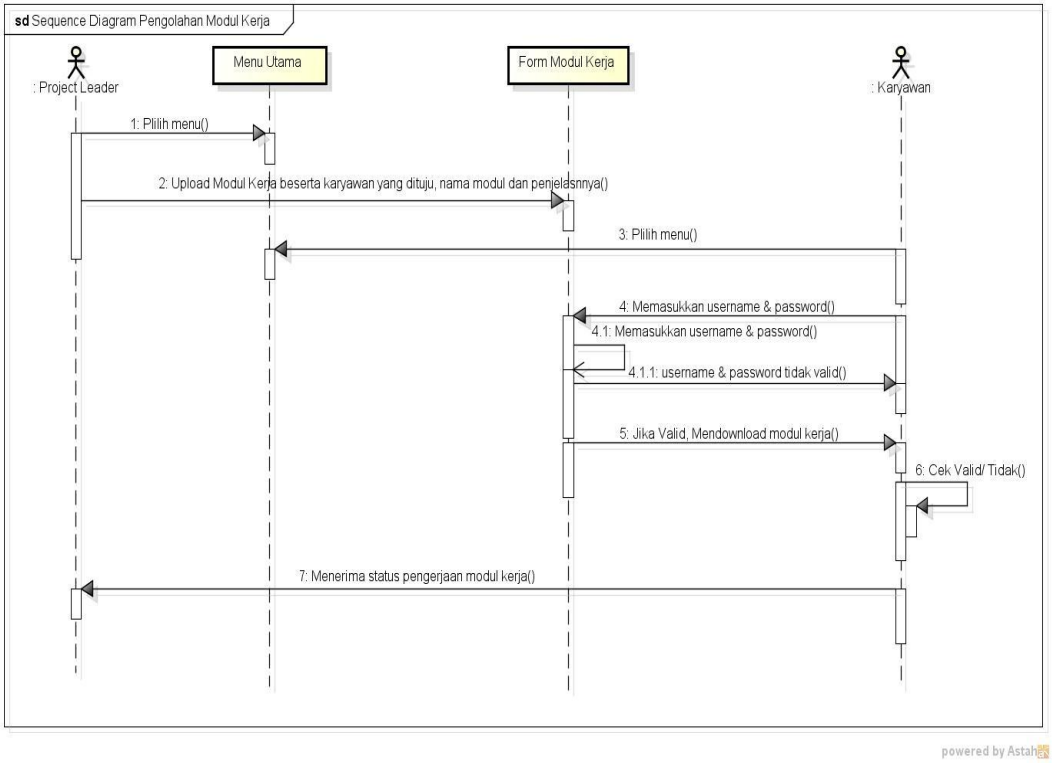


Gambar 4.11. *Sequence Diagram Login* Yang Diusulkan

2. *Sequence Diagram* Pengolahan Modul Kerja

Gambar dibawah ini merupakan *Sequence Diagram* pengolahan modul kerja, dimana *Project Leader* mengupload file yang berupa modul kerja tersebut ke Program Aplikasi yang akan dibuat untuk diteruskan kepada karyawan yang

bersangkutan. Lalu Karyawan tersebut mendownload *file* tersebut dengan memasukkan *username* & *password* untuk validasi dan apabila valid maka akaj dicek apakah modul kerja tersebut benar/ tidak ditujukan kepada karyawan yang bersangkutan. Apabila salah, maka karyawan akan tidak menyetujui pengerjaan dengan menyertakan informasi berupa alasannya. Dan apabila benar, maka karyawan akan menyetujui dan mengerjakan modul kerja tersebut.

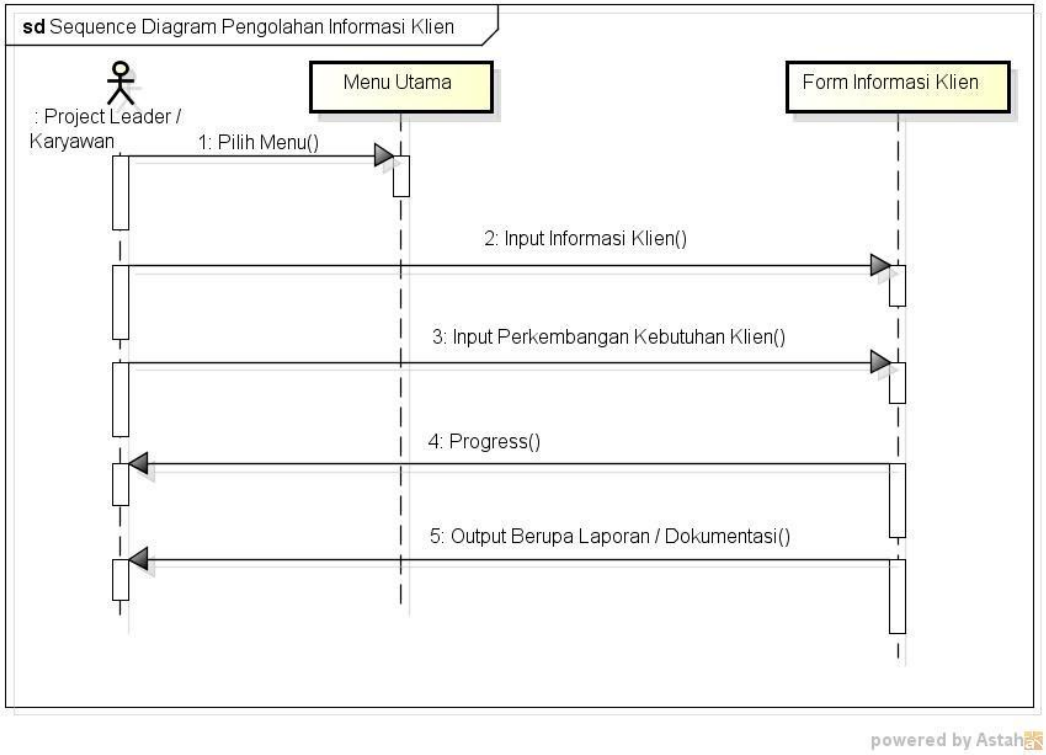


Gambar 4.12. *Sequence Diagram* Pengolahan Modul Kerja Yang Diusulkan

3. *Sequence Diagram* Pengolahan Informasi Klien

Gambar dibawah ini merupakan *Sequence Diagram* Pengolahan Informasi Klien, dimana informasi yang telah disampaikan klien kepada *Project Leader* atau kepada karyawan pertama-tama akan diinputkan kedalam program aplikasi, dan

apabila ada perkembangan selanjutnya dalam penanganan kebutuhan klien tersebut akan kembali diinputkan kedalam program aplikasi. Lalu informasi-informasi tersebut akan diolah dan akan menghasilkan outputan berupa laporan kepada klien dan dokumentasi yang akan disimpan oleh pihak Mitra Dinamika Konsultan.

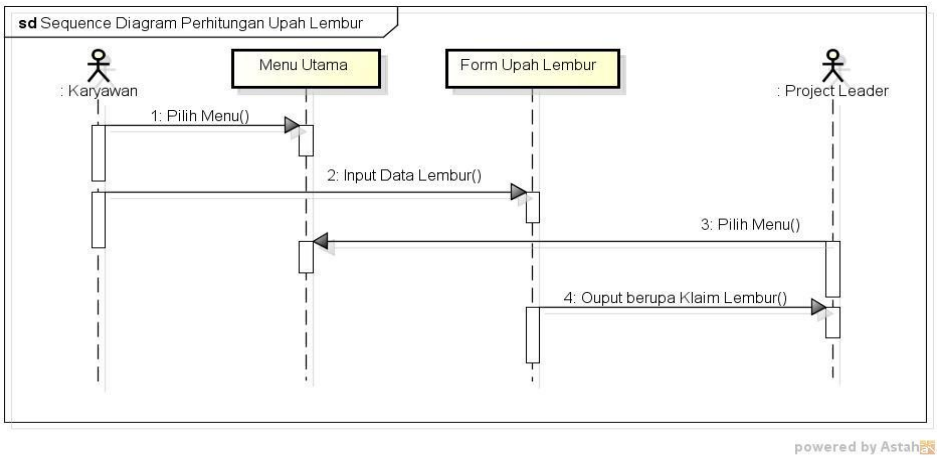


Gambar 4.13 *Sequence Diagram* Pengolahan Informasi Klien Yang Diusulkan

4. *Sequence Diagram* Perhitungan Upah Lembur

Gambar dibawah ini merupakan *Sequence Diagram* perhitungan waktu lembur, dimana data-data lembur karyawan yang berupa tanggal lembur, lama lembur, dan keterangan lembur akan diinputkan setiap kali karyawan lembur

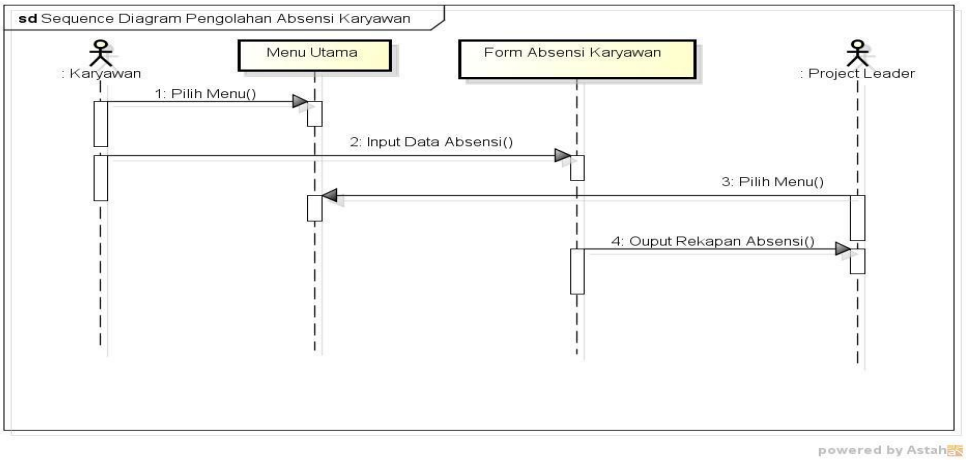
kerjannya. Lalu kumpulan data-data lembur tersebut akan menjadi outputan berupa klaim lembur yang akan di terima oleh *Project Leader* setiap bulannya.



Gambar 4.14. *Sequence Diagram* Perhitungan Waktu Lembur Yang Diusulkan

5. *Sequence Diagram* Pengolahan Absensi Karyawan

Gambar dibawah ini merupakan *Sequence Diagram* pengolahan absensi karyawan, dimana karyawan akan menginputkan data kehadiran setiap hari kerja dengan menggunakan program aplikasi ini disaat mereka datang di pagi hari untuk memulai kerja. Outputan yang akan dihasilkan dari proses ini adalah rekapan absensi kepada *Project Leader*.



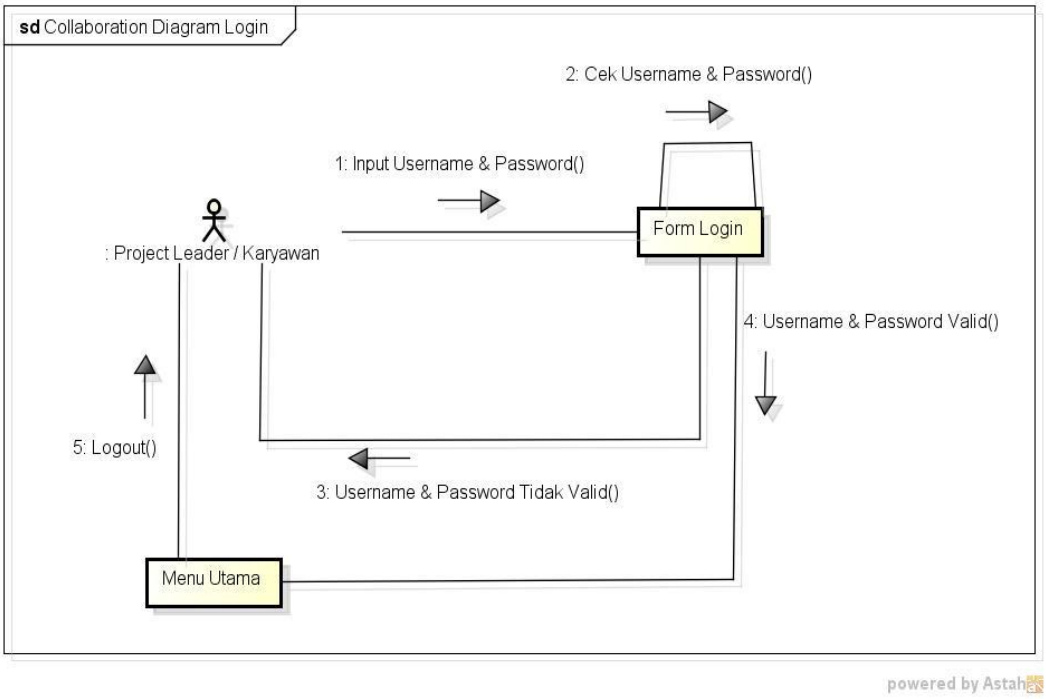
Gambar 4.15. *Sequence Diagram* Pengolahan Absensi Karyawan Yang Diusulkan

4.2.3.5. *Collaboration Diagram*

Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masingmasing objek dan bukan pada waktu penyampaian message. Setiap message memiliki sequence number, di mana *message* dari level tertinggi memiliki nomor 1. *Messages* dari level yang sama memiliki prefiks yang sama.

1. *Collaboration Diagram Login*

Collaboration diagram Login, menggambarkan skenario dari Aplikasi pembantu dan pendukung kinerja karyawan di Mitra Dinamika Konsultan antara *Project Leader* / karyawan dengan sistem dalam proses *login*. Dalam hal ini *Project Leade* / karyawan harus menginputka *Username* dan *Password* terlebih dahulu untuk masuk kedalam sistem. Berikut gambarnya :

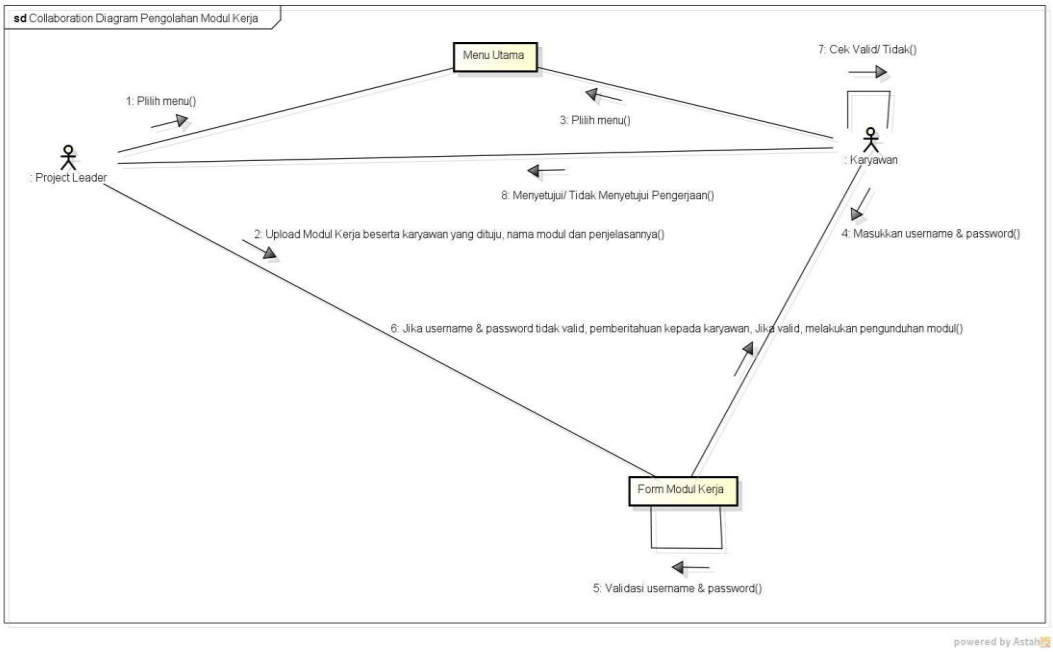


Gambar 4.16. *Collaboration Diagram Login* Yang Diusulkan

2. *Collaboration Diagram* Pengolahan Modul Kerja

Collaboration diagram Pengolahan Modul Kerja menggambarkan skenario dari Aplikasi pembantu dan pendukung kinerja karyawan di Mitra Dinamika Konsultan antara *Project Leader*, karyawan dengan sistem. Dalam hal ini *Project Leader* yang sudah mengupload modul kerja kedalam sistem program aplikasi yang akan dibuat agar dapat diterima oleh para karyawan yang bersangkutan. Lalu Karyawan tersebut mendownload file tersebut dengan memasukkan *username* dan *password* terlebih dahulu , apabila *username* dan *password valid*, maka modul dapat didownload untuk dicek apakah modul kerja tersebut benar/ tidak ditujukan kepada karyawan yang bersangkutan. Apabila salah, maka karyawan akan tidak menyetujui pengerjaan dengan menyertakan

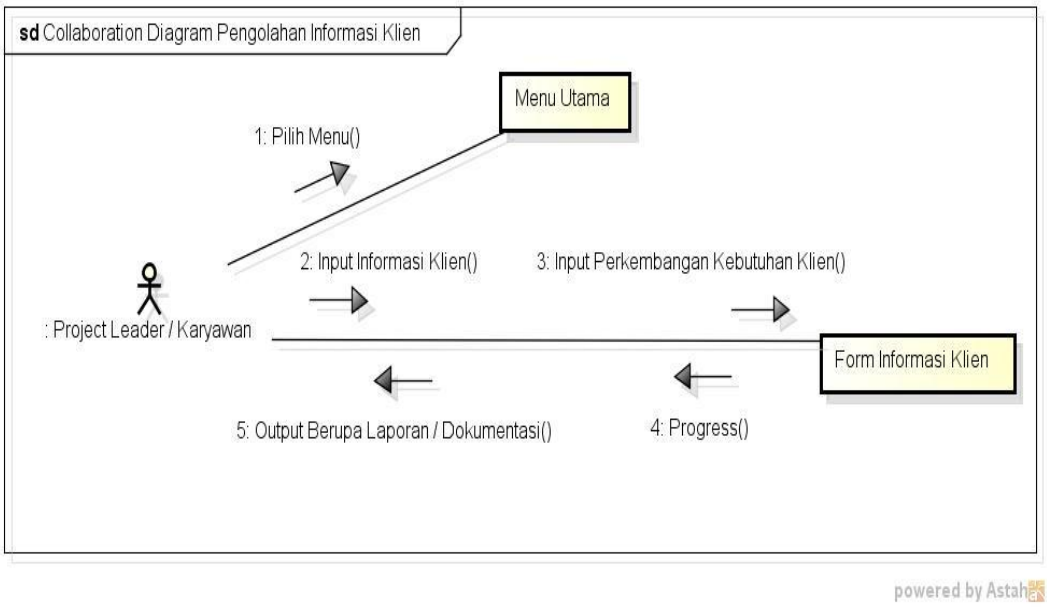
informasi berupa alasannya. Dan apabila benar, maka karyawan akan menyetujui dan mengerjakan modul kerja tersebut. Berikut gambarnya :



Gambar 4.17. *Collaboration Diagram* Pengolahan Modul Kerja Yang Diusulkan

3. *Collaboration Diagram* Pengolahan Informasi Klien

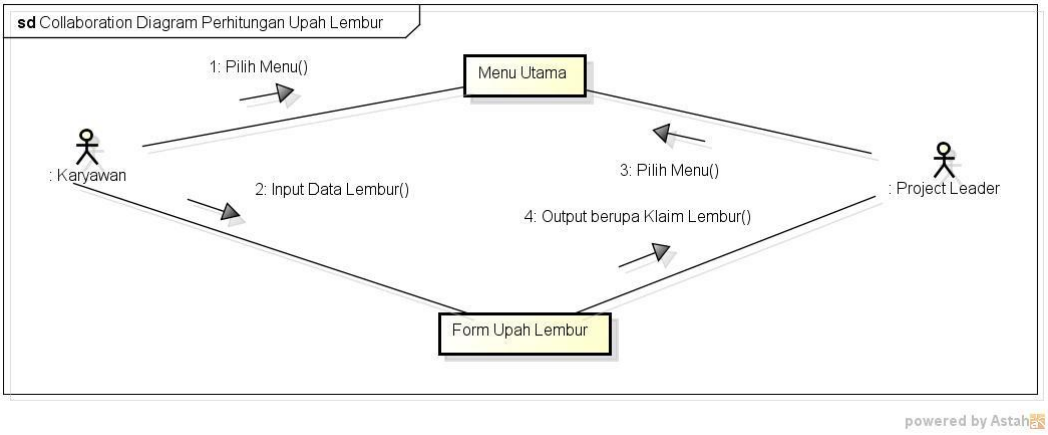
Collaboration diagram pengolahan informasi klien, menggambarkan skenario dari Aplikasi pembantu dan pendukung kinerja karyawan di Mitra Dinamika Konsultan antara *Project Leader* , karyawan dengan sistem. Dalam hal ini informasi yang telah disampaikan klien kepada *Project Leader* atau kepada karyawan pertama-tama akan diinputkan kedalam program aplikasi, dan apabila ada perkembangan selanjutnya dalam penanganan kebutuhan klien tersebut akan kembali diinputkan kedalam program aplikasi. Lalu informasi-informasi tersebut akan diolah dan akan menghasilkan outputan berupa laporan kepada klien dan dokumentasi yang akan disimpan oleh pihak Mitra Dinamika Konsultan. Berikut gambarnya :



Gambar 4.18. *Collaboration Diagram* Pengolahan Informasi Klien Yang Diusulkan

4. *Collaboration Diagram* Perhitungan Upah Lembur

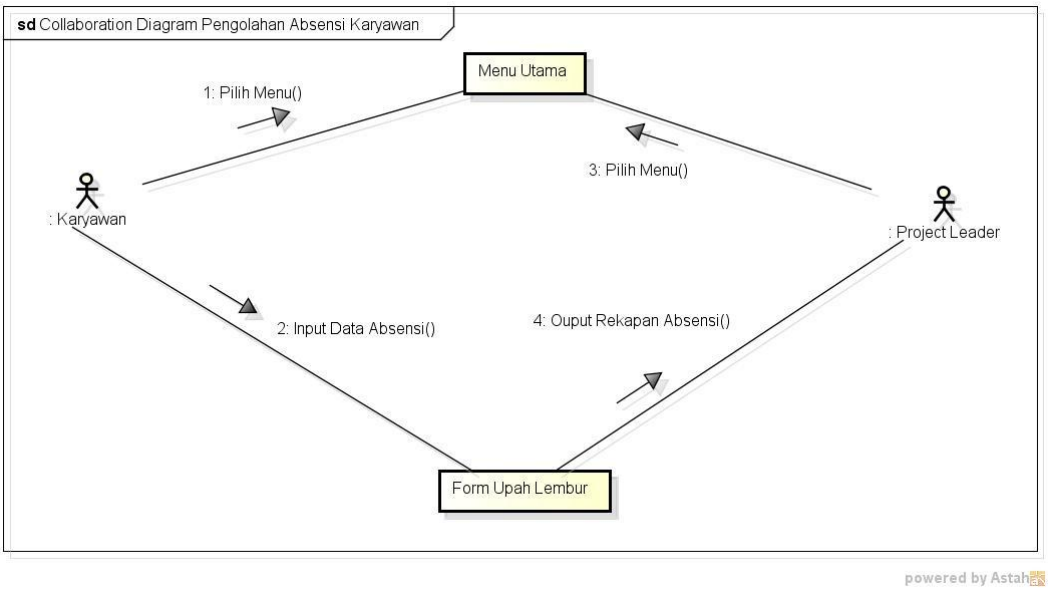
Collaboration diagram perhitungan upah lembur, menggambarkan skenario dari Aplikasi pembantu dan pendukung kinerja karyawan di Mitra Dinamika Konsultan antara *Project Leader* , karyawan dengan sistem. Dalam hal ini data-data lembur karyawan yang berupa tanggal lembur, lama lembur, dan keterangan lembur akan diinputkan setiap kali karyawan lembur kerjannya. Lalu kumpulan data-data lembur tersebut akan menjadi outputan berupa klaim lembur yang akan di terima oleh *Project Leader* setiap bulannya. Berikut gambarnya :



Gambar 4.19. *Collaboration Diagram* Perhitungan Upah Lembur Yang Diusulkan

5. *Collaboration Diagram* Pengolahan Absensi Karyawan

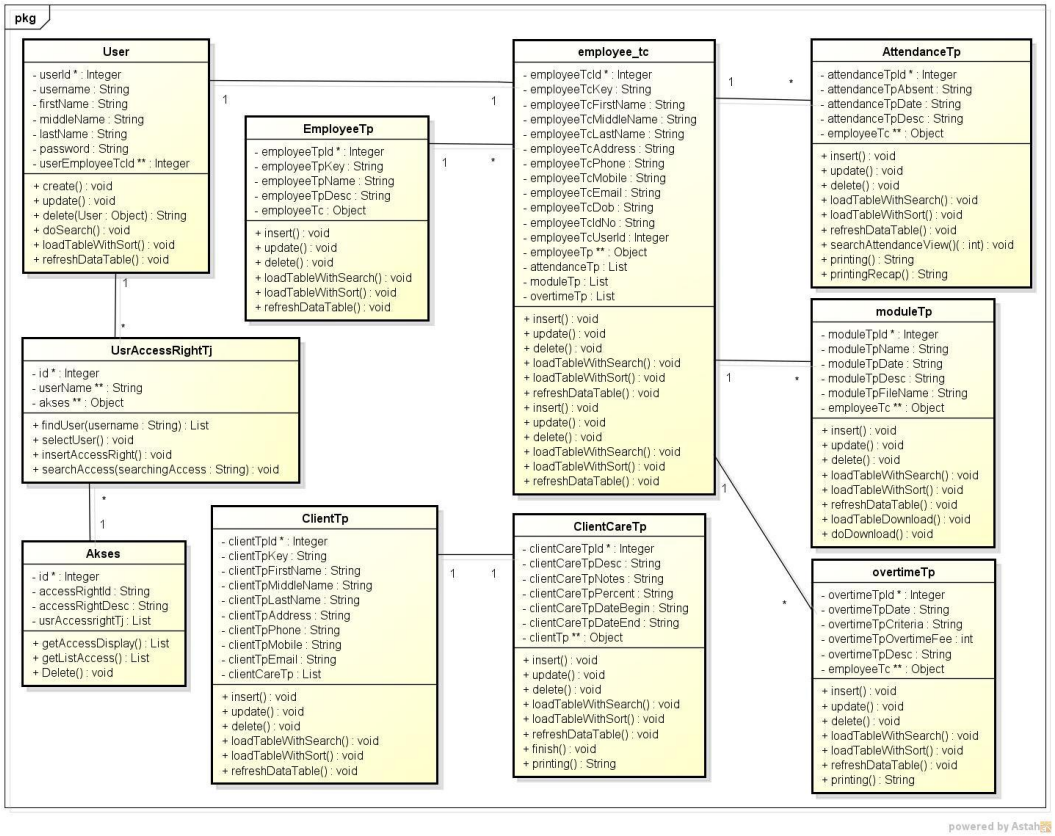
Collaboration diagram pengolahan absensi karyawan, menggambarkan skenario dari Aplikasi pembantu dan pendukung kinerja karyawan di Mitra Dinamika Konsultan antara *Project Leader* , karyawan dengan sistem. Dalam hal ini karyawan akan menginputkan data kehadiran setiap hari kerja dengan menggunakan program aplikasi ini disaat mereka datang di pagi hari untuk memulai kerja. Outputan yang akan dihasilkan dari proses ini adalah rekapan absensi kepada *Project Leader*. Berikut gambarnya :



Gambar 4.20. *Collaboration Diagram* Pengolahan Absensi Karyawan Yang Diusulkan

4.2.3.6. *Class Diagram*

Class Diagram adalah sebuah spesifikasi yang jika di instansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class Diagram* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaam tersebut (metoda/fungsi). *Class Diagram* menggambarkan struktur dan deskripsi *Class*, *Package*, dan dan *Object* beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain. Berikut adalah *Class Diagram* yang dibutuhkan dalam perancangan Program Aplikasi pembantu dan pendukung kinerja karyawan di Mitra Dinamika Konsultan :

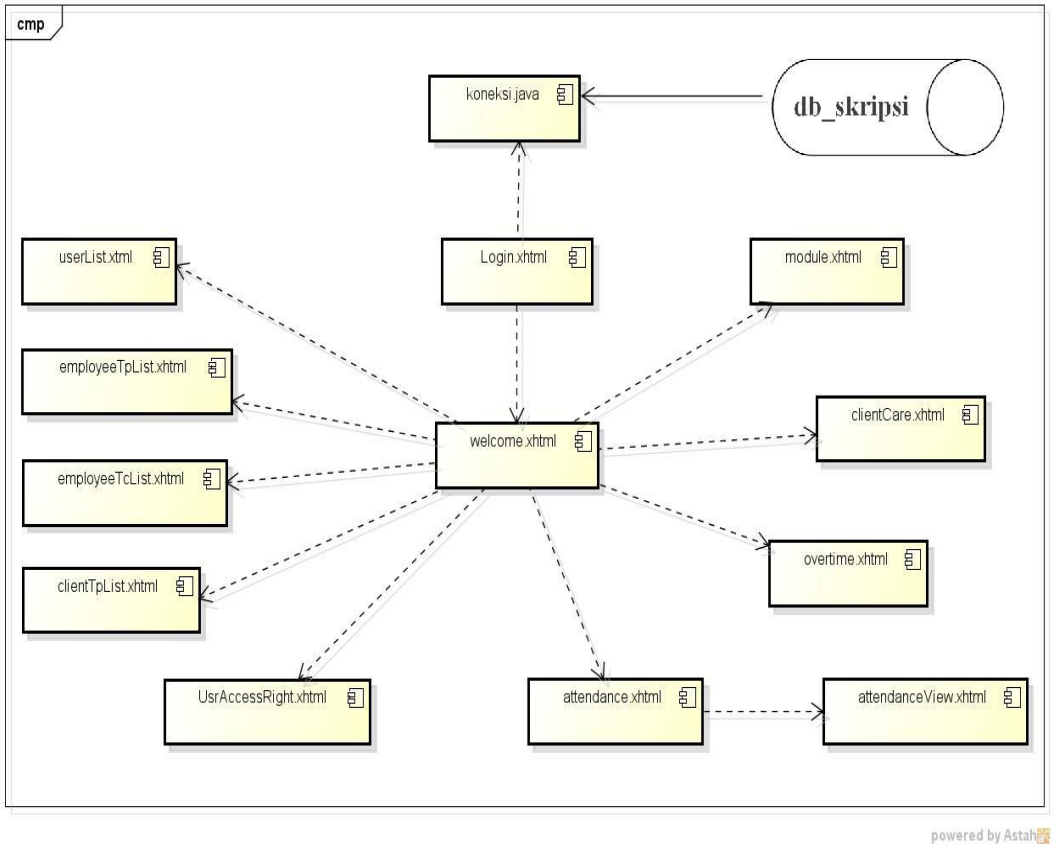


Gambar 4.21. *Class Diagram* Yang Diusulkan

4.2.3.7. *Component Diagram*

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun run *time*. Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang

disediakan sebuah komponen untuk komponen lain. Berikut gambar *Component diagram* yang diusulkan :

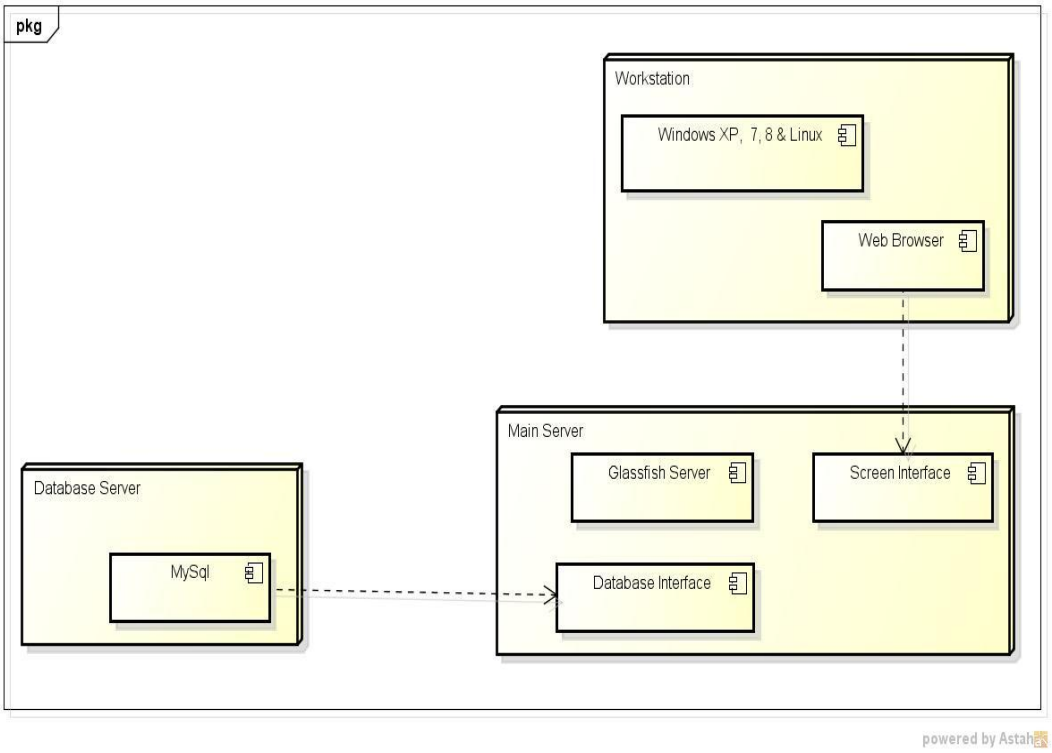


Gambar 4.22. *Component Diagram* Yang Diusulkan

4.2.3.8. *Deployment Diagram*

Deployment/ physical diagram menggambarkan detail bagaimana komponen di-deploy dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, *server* atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *server*, dan hal-hal lain yang bersifat fisikal. Sebuah *node* adalah *server*, *workstation*, atau piranti keras lain yang digunakan untuk men-deploy komponen dalam lingkungan sebenarnya. Hubungan antar node

(misalnya *TCP/ IP*) dan *requirement* dapat juga didefinisikan dalam diagram ini. . Berikut gambar *Deployment diagram* yang diusulkan :



Gambar 4.23 *Deployment Diagram* Yang Diusulkan

4.2.4. Perancangan Antar Muka

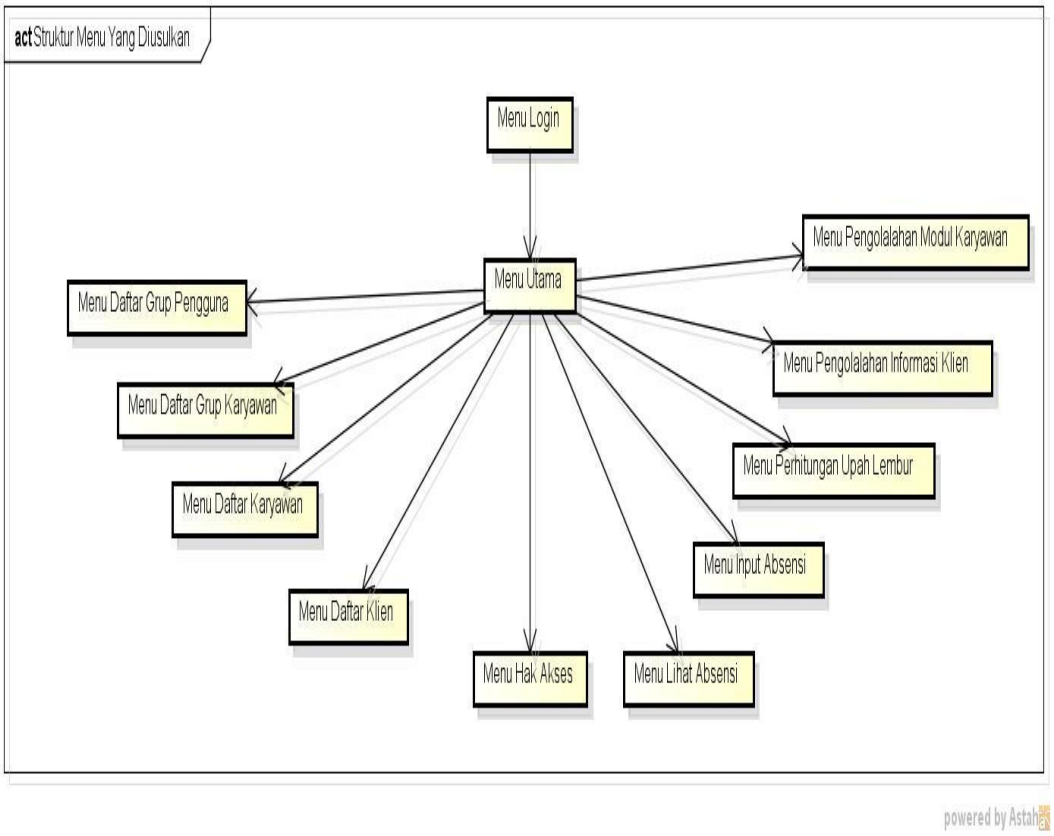
Pada perancangan antar muka atau *interface* ini akan menggambarkan rancangan tampilan yang ada dalam sistem yang diusulkan. Adapun yang akan dibahas adalah perancangan input dan perancangan *output*.

4.2.4.1. Struktur *Menu*

Struktur *menu* adalah bentuk umum dari suatu rancangan program untuk memudahkan pemakai dalam menjalankan program komputer sehingga pada saat

menjalankan program dikomputer, *user* tidak mengalami kesulitan lagi dalam memilih menu nyang diinginkan.

Struktur menu yang diusulkan adalah sebagai berikut :



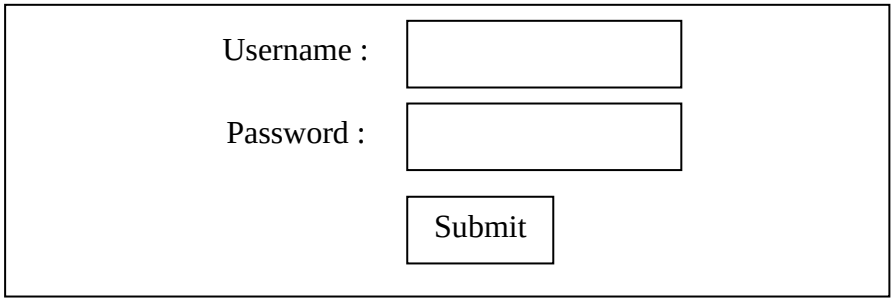
Gambar 4.24. Struktur *Menu* Yang Diusulkan

4.2.4.2. Perancangan *Input*

Perancangan ini dipergunakan bagi pengguna yang berkepentingan untuk menggunakan program aplikasi. Bagi yang menggunakan program aplikasi ini maka terlebih dahulu harus memasukkan *username* dan *password*.

1. Login

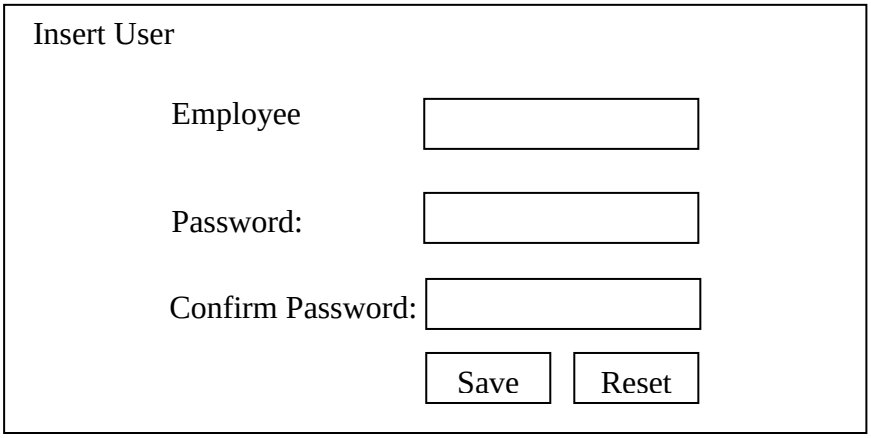
Tombol *Submit* di *Form* Inputan *Login* akan menvalidasi langsung ke *database* sistem, dimana tombol *Submit* diatas akan menampilkan pesan validasi *username* dan *password* dari *user*.



Gambar 4.25. *Form* Inputan *Login* Yang Diusulkan

2. Pendaftaran *User Maintenance*

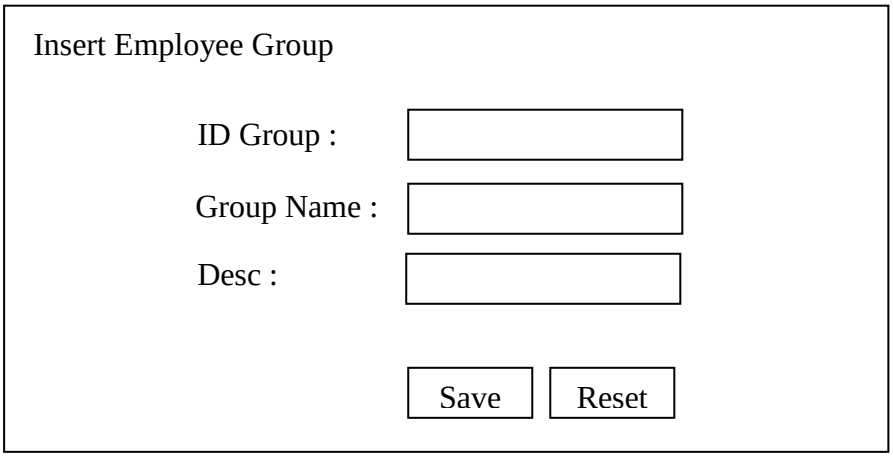
Tombol *Save* akan menyimpan data *User maintenance* langsung ke *database* sistem, sedangkan tombol *Reset* diatas akan mengulang pengisian *record*, apablia *user* salah memasukkan data/ ingin mengganti data, dengan mengklik tombol *Reset* tersebut, maka sistem otomatis akan menghapus inputan.



Gambar 4.26. *Form* Inputan *User Maintenance* Yang Diusulkan

3. Pendaftaran *Employee Group*

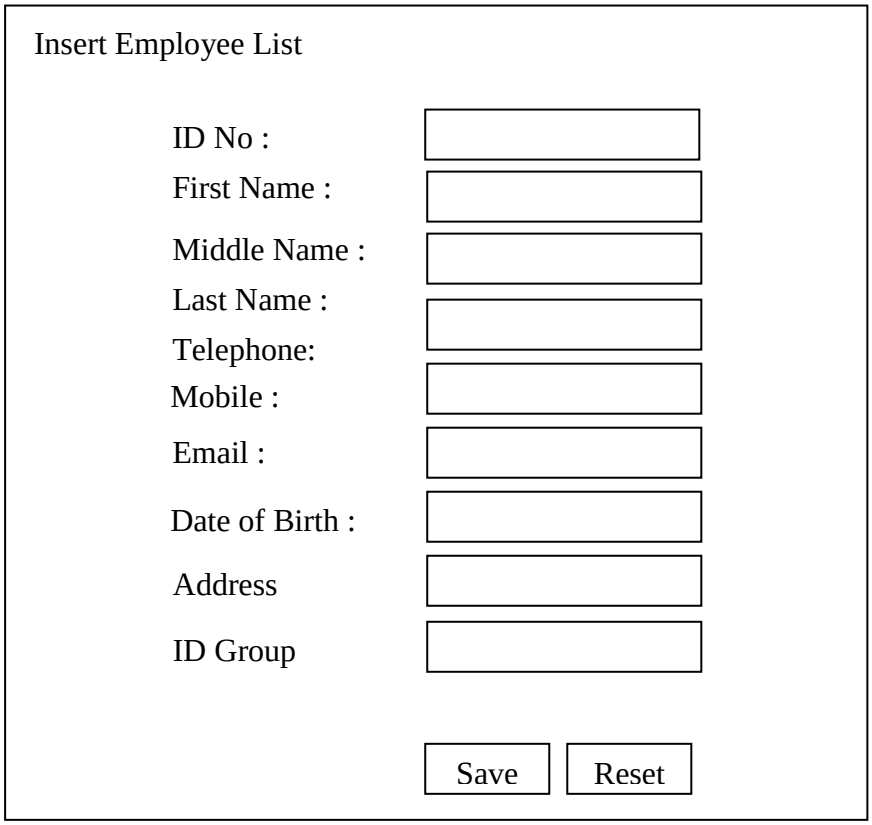
Tombol *Save* akan menyimpan data *Employee Group* langsung ke *database* sistem, sedangkan tombol *Reset* diatas akan mengulang pengisian *record*, apablia *user* salah memasukkan data/ ingin mengganti data, dengan mengklik tombol *Reset* tersebut, maka sistem otomatis akan menghapus inputan.



Gambar 4.27. *Form* Inputan *Employee Group* Yang Diusulkan

4. Pendaftaran *Employee List*

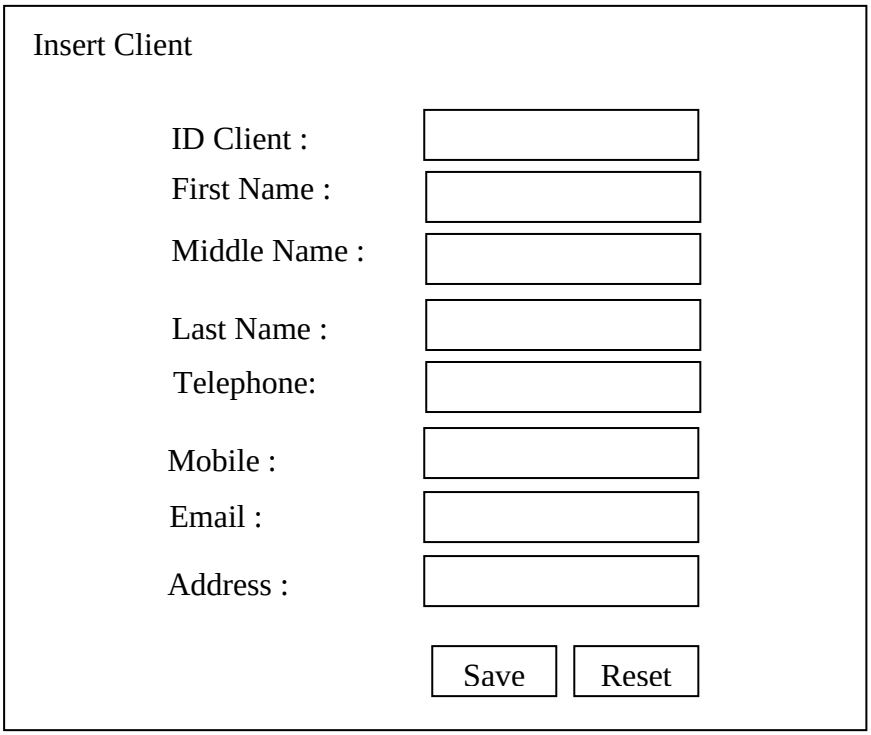
Tombol *Save* diatas akan menyimpan data *Employee List* langsung ke *database* sistem, sedangkan tombol *Reset* diatas akan mengulang pengisian *record*, apablia *user* salah memasukkan data/ ingin mengganti data, dengan mengklik tombol *Reset* tersebut, maka sistem otomatis akan menghapus inputan.



Gambar 4.28. *Form* Inputan *Employee List* Yang Diusulkan

5. Pendaftaran Klien

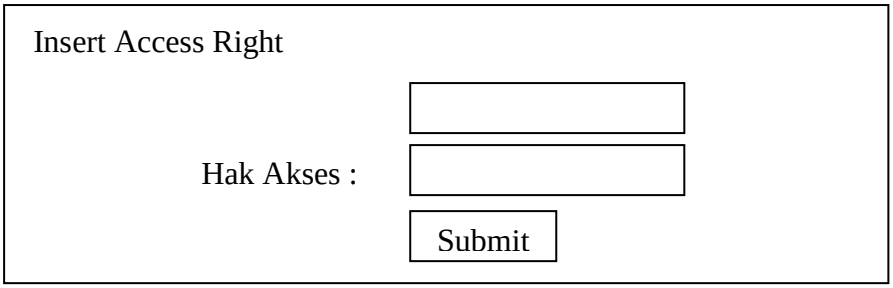
Tombol *Save* akan menyimpan data klien langsung ke *database* sistem, sedangkan tombol *Reset* diatas akan mengulang pengisian *record*, apablia *user* salah memasukkan data/ ingin mengganti data, dengan mengklik tombol *Reset* tersebut, maka sistem otomatis akan menghapus inputan.



Gambar 4.29. *Form* Pendaftaran Klien Yang Diusulkan

6. Pemberian Hak Akses

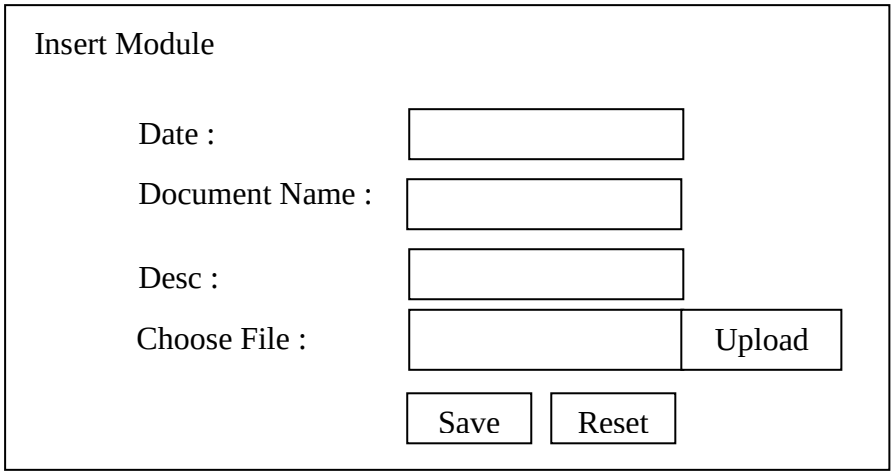
Tombol *Save* akan menyimpan data hak akses langsung ke *database* sistem, sedangkan tombol *Reset* diatas akan mengulang pengisian *record*, apablia *user* salah memasukkan data/ ingin mengganti data, dengan mengklik tombol *Reset* tersebut, maka sistem otomatis akan menghapus inputan.



Gambar 4.30. *Form* Pemberian Hak Akses Yang Diusulkan

7. Pengolahan Modul Karyawan

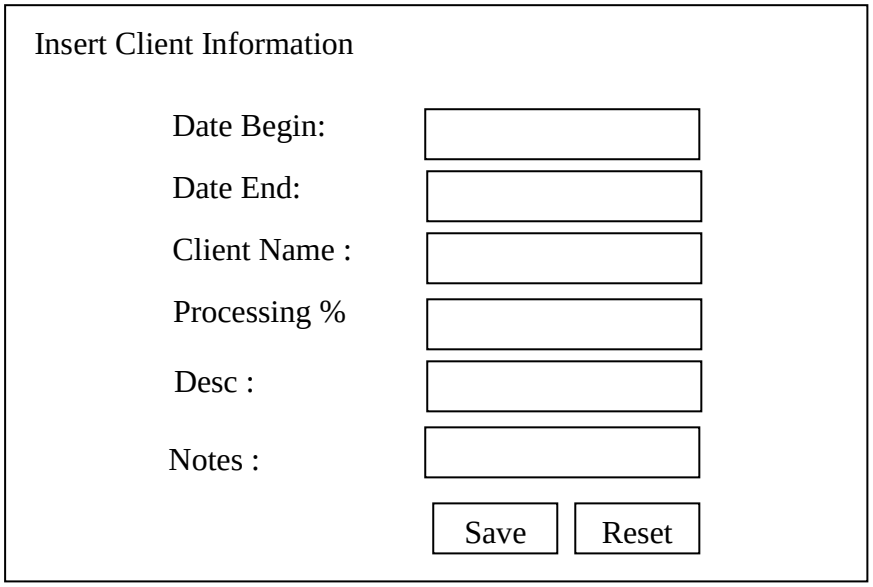
Tombol *Save* akan menyimpan data modul karyawan langsung ke *database* sistem, sedangkan tombol *Reset* diatas akan mengulang pengisian *record*, apablia *user* salah memasukkan data/ ingin mengganti data, dengan mengklik tombol *Reset* tersebut, maka sistem otomatis akan menghapus inputan.



Gambar 4.31. *Form* Pengolahan Modul Karyawan Yang Diusulkan

8. Pengolahan Informasi Klien

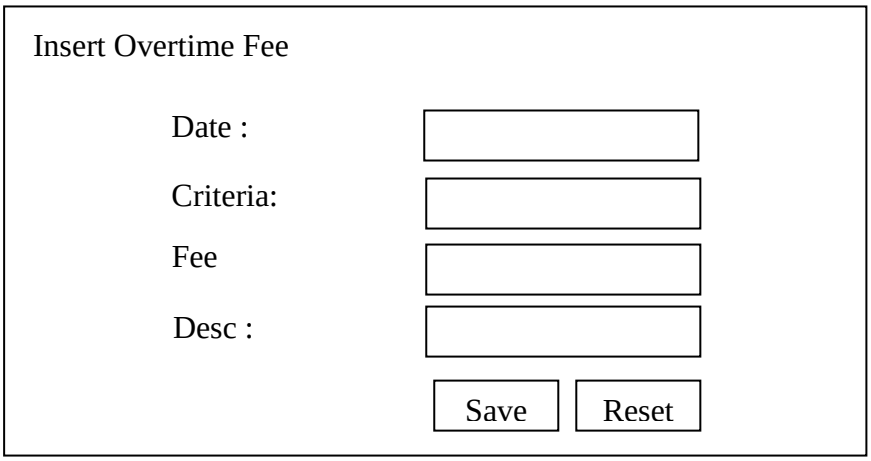
Tombol *Save* akan menyimpan data informasi klien langsung ke *database* sistem, sedangkan tombol *Reset* diatas akan mengulang pengisian *record*, apablia *user* salah memasukkan data/ ingin mengganti data, dengan mengklik tombol *Reset* tersebut, maka sistem otomatis akan menghapus inputan.



Gambar 4.32. *Form* Pengolahan Informasi Klien Yang Diusulkan

9. Pengolahan Upah Lembur

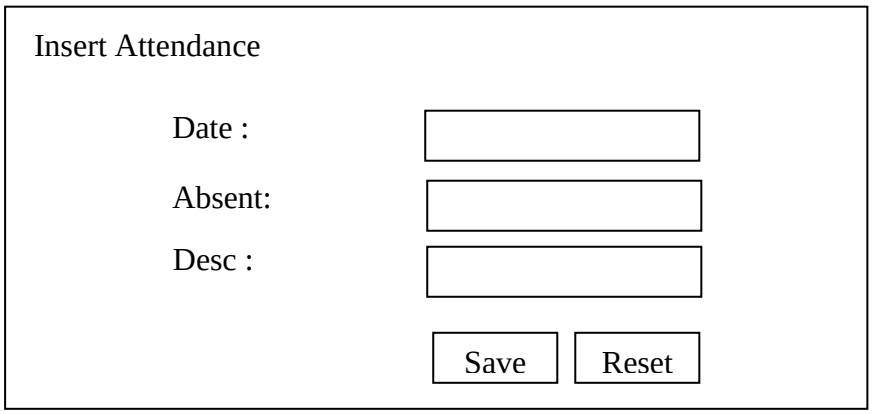
Tombol *Save* akan menyimpan data uaph lembur langsung ke *database* sistem, sedangkan tombol *Reset* diatas akan mengulang pengisian *record*, apablia *user* salah memasukkan data/ ingin mengganti data, dengan mengklik tombol *Reset* tersebut, maka sistem otomatis akan menghapus inputan.



Gambar 4.33. *Form* Pengolahan Upah Lembur Yang Diusulkan

10. *Input* Absensi Karyawan

Tombol *Save* akan menyimpan data absensi karyawan langsung ke *database* sistem, sedangkan tombol *Reset* diatas akan mengulang pengisian *record*, apablia *user* salah memasukkan data/ ingin mengganti data, dengan mengklik tombol *Reset* tersebut, maka sistem otomatis akan menghapus inputan.



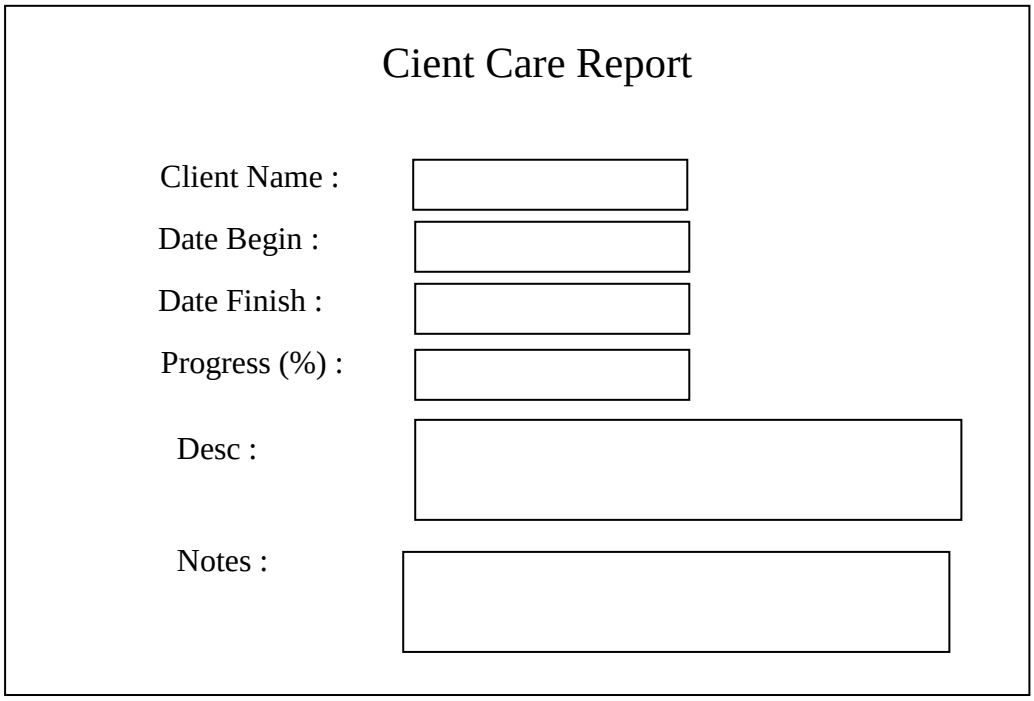
Gambar 4.34. *Form Input* Absensi Karyawan Yang Diusulkan

4.2.4.3. Perancangan *Output*

Rancangan *output* adalah produk dari sistem informasi yang dihasilkan setelah input dilaksanakan. Hasil dari rancangan input data terlihat dari media keluaran. Dalam rancangan *output*, sistem informasi ini juga memanfaatkan file database mengandung data-data yang terbilang cukup banyak sehingga membutuhkan tempat yang cukup luas untuk menggambarkan bagaimana sebenarnya rancangan *output* pada aplikasi yang akan dibuat.

1. Laporan Informasi Klien

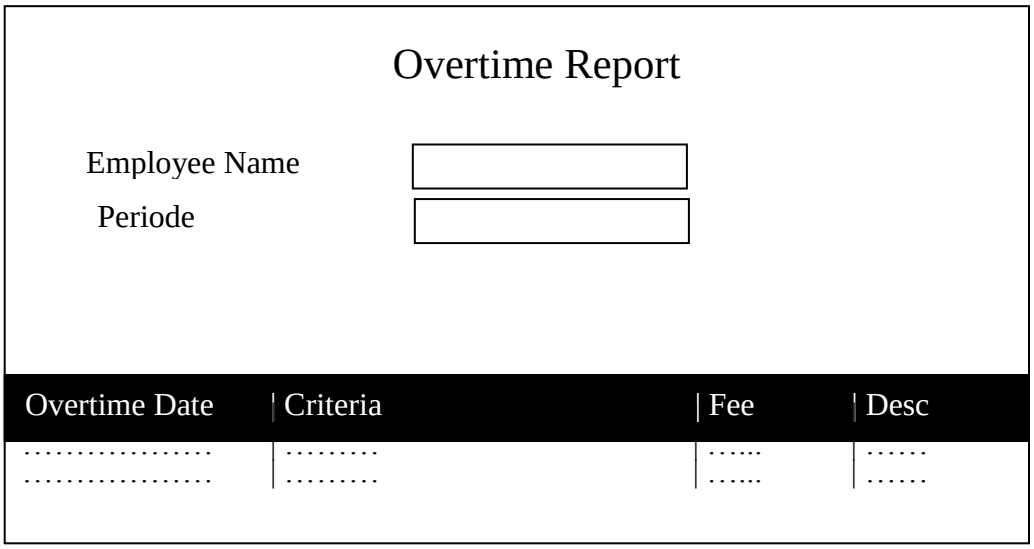
Perancangan *output* ini bertujuan menampilkan data informasi proses pengerjaan kebutuhan klien. Dimana *output* ini berbentuk laporan berformat .docx yang dapat langsung diunduh pada *Form* Informasi Klien di Program Aplikasinya.



Gambar 4.35. *Form Output* Laporan Informasi Klien Yang Diusulkan

2. Laporan Upah Lembur Karyawan

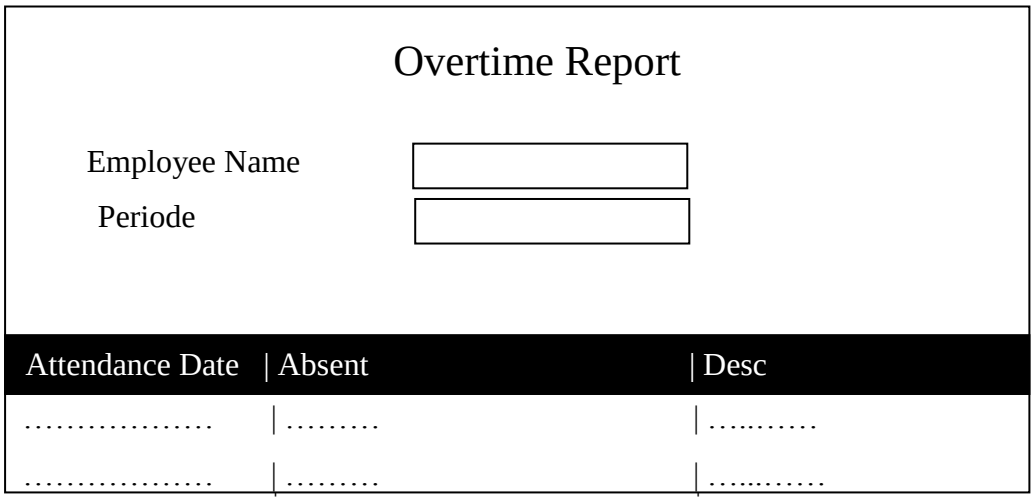
Perancangan *output* ini bertujuan menampilkan data rekapan upah lembur yang diajukan oleh karyawan. Dimana *output* ini berbentuk laporan berformat .docx yang dapat langsung diunduh pada *Form* Upah Lembur di Program Aplikasinya.



Gambar 4.36. *Form Output* Laporan Upah Lembur Yang Diusulkan

3. Laporan Absensi Karyawan

Perancangan *output* ini bertujuan menampilkan data absensi karyawan yang diajukan oleh karyawan. Dimana *output* ini berbentuk laporan berformat .pdf yang dapat langsung diunduh pada *Form* Absensi Karyawan di Program Aplikasinya.

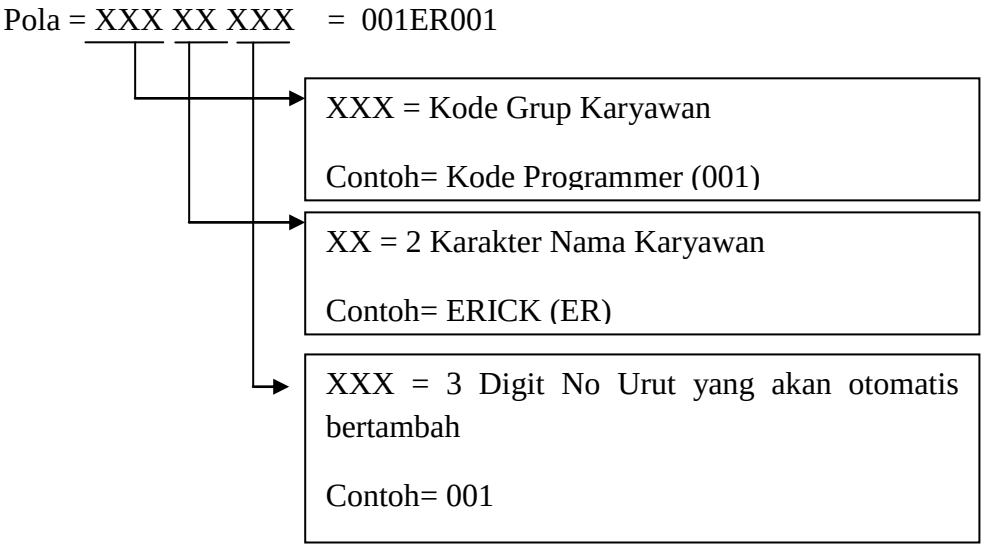


Gambar 4.37. *Form Output* Laporan Absensi Karyawan Yang Diusulkan

4.2.4.4. Kodefikasi

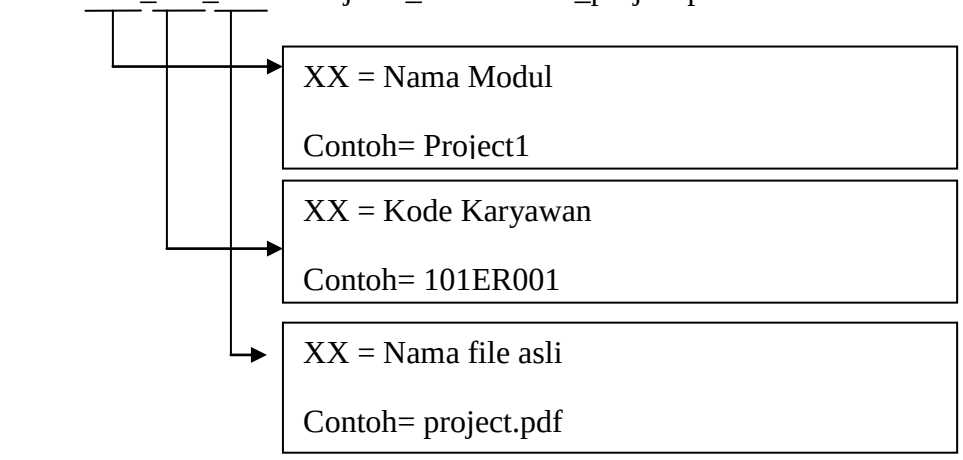
Merupakan pola kode otomatis yang dibentuk oleh sistem, dimana inputan yang dimasukkan akan otomatis digenerate oleh sistem. Contoh kedefikasi yang ada di Program Aplikasi ini adalah :

1. Kode Karyawan



Gambar 4.38. Struktur Kodefikasi Kode Karyawan

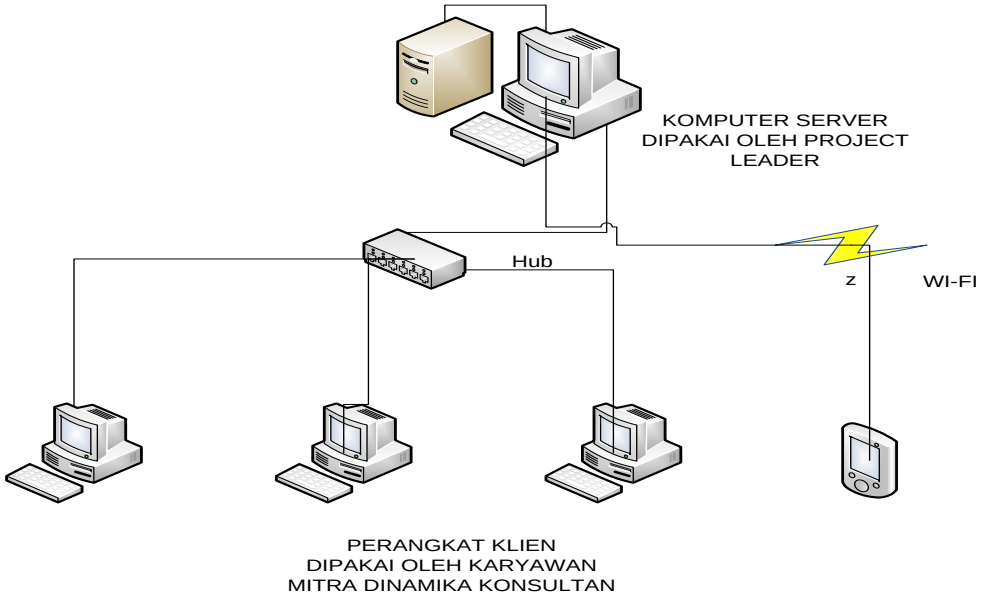
2. Nama File Modul



Gambar 4.39. Struktur Kodefikasi Nama File Modul

4.2.4.5. Perancangan Arsitektur Jaringan

Adapun arsitertur jaringan sebagai lingkungan pengembangan dan implementasi dari lingkungan dari aplikasi secara ringkas diilustrasikan oleh gambar berikut :



Gambar 4.40. Arsitektur Jaringan Yang Diusulkan

Secara umum, arsitektur tersebut dapat dijelaskan sebagai berikut :

1. Program Aplikasi ini akan di *install* di komputer *server*, yang mana akan menggunakan DBMS (*Database Management System*)
2. Masing-masing komputer klien akan mengakses aplikasi tersebut dengan menggunakan sistem jaringan komputer.