



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

PELAKSANAAN KERJA MAGANG

3.1. KEDUDUKAN KOORDINASI

Penulis dalam pelaksanaan kerja magang ini tidaklah mengerjakan proyek sistem informasi untuk pelanggan dari perusahaan ataupun bekerja sama dengan karyawan-karyawan perusahaan secara tim dan mengerjakan sebuah proyek sistem informasi, melainkan penulis bekerja secara independen untuk membuatkan sistem informasi berbasis web yang akan dipakai internal perusahaan dalam hal administrasi data karyawan dan data absensi. Koordinasi selama kerja magang ini dilakukan penulis langsung ke pemilik perusahaan, yaitu Bpk. Matius Kelvin selaku *Business Development* dan pemilik dari perusahaan. Beliau lah yang memberitahukan kebutuhan sistem yang akan dibuatkan, dan sekaligus berperan sebagai pembimbing lapangan yang mengarahkan penulis dalam hal pengembangan sistem apabila terdapat kesulitan selama pengerjaannya.

3.2. TUGAS YANG DILAKUKAN

Tugas yang penulis lakukan pertama kali adalah bertemu dengan Bpk. Matius Kelvin untuk membicarakan tentang sistem yang akan dibuatkan. Pada pertemuan pertama ini penulis mendapatkan gambaran secara garis besar mengenai sistem, tingkatan pengguna yang akan menggunakan sistem, dan fungsi-fungsi dari masing-masing tingkatan pengguna tersebut. Berdasarkan gambaran umum tersebut penulis diberi tugas untuk

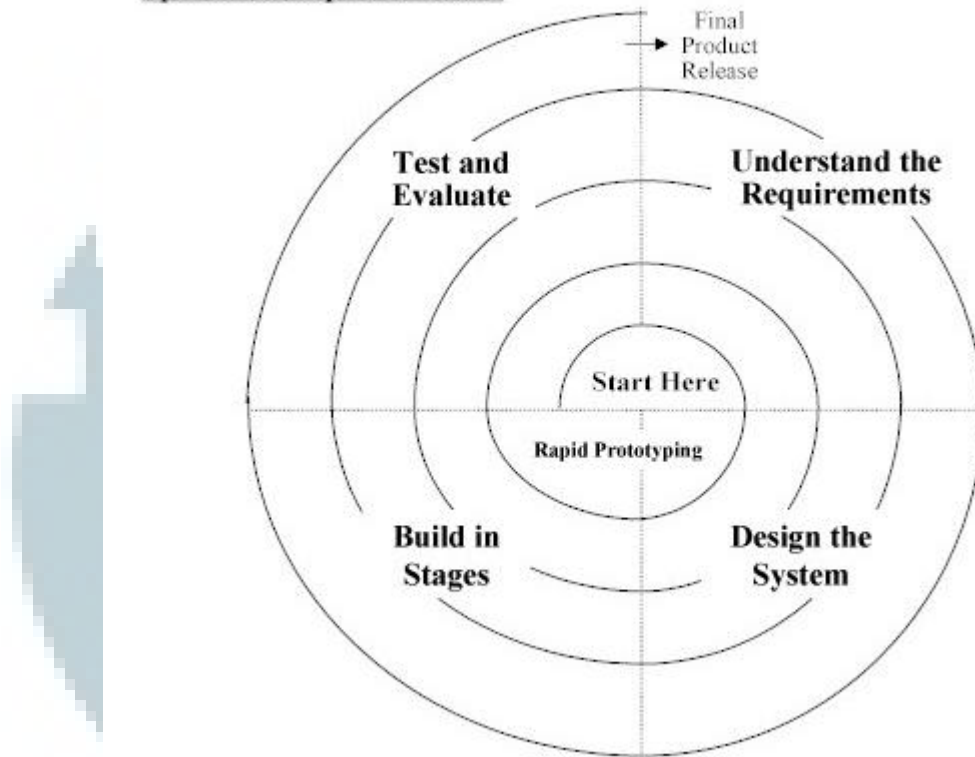
merancang aplikasi dan *database* sesuai dengan gambaran awal yang diberikan oleh pemilik perusahaan. Untuk pertemuan-pertemuan selanjutnya, penulis memperlihatkan hasil kerja sistem yang dibuat di *laptop* penulis secara bertahap kepada pemilik sesuai dengan hasil revisi pada pertemuan sebelumnya. Setelah sistem sudah sesuai dengan keinginan maka akan dilakukan rencana implementasi sistem ke *hosting* pada akhir bulan Desember, sehingga web dapat mulai diakses pada awal tahun 2014.

3.3. METODE PENGEMBANGAN SISTEM

3.3.1. Metode Pengembangan Sistem yang digunakan

Berdasarkan koordinasi yang dilakukan penulis selama kerja magang terkait pengembangan sistem ini, maka metodologi pengembangan sistem yang penulis gunakan adalah *Spiral*. Ini dikarenakan setiap tahapan yang dilakukan penulis selama pengembangan sistem itu dilakukan berulang dengan adanya prototipe sistem yang terus berkembang dan dievaluasi oleh pemilik perusahaan sampai sistem itu sudah benar-benar selesai dan sesuai keinginan pemilik, serta memenuhi kebutuhan proses bisnis perusahaan khususnya dalam hal administrasi data karyawan dan data absensi. Metode pengembangan sistem ini terdiri atas beberapa tahapan sesuai dengan (Chapman, 2004).

Spiral Development Model



Gambar 3.1 *Spiral Development Model*(Chapman, 2004).

1. Memahami Kebutuhan

Pada tahap ini dilakukan pengumpulan kebutuhan sistem, menganalisis dan memahami kebutuhan sistem, serta mengidentifikasi apabila terdapat masalah yang harus diperbaiki dari sistem.

2. Desain Sistem

Setelah requirement diterima, dilanjutkan ke tahap desain sistem berdasarkan requirement yang didapat pada tahapan sebelumnya.

3. Membangun sistem

Pada tahapan ini dibuatlah prototipe atas desain sistem atau bisa disebut tahapan *development*.

4. Tes dan Evaluasi

Pengguna mengevaluasi dan tes sistem, apakah terdapat kelemahan yang harus diperbaiki baik itu dari sisi tampilan, fungsi yang tidak sesuai dengan kebutuhan perusahaan, maupun adanya perubahan dan penambahan kebutuhan atas sistem dari pengguna.

3.3.2. Metode Pelaksanaan Penulis pada setiap Tahapan

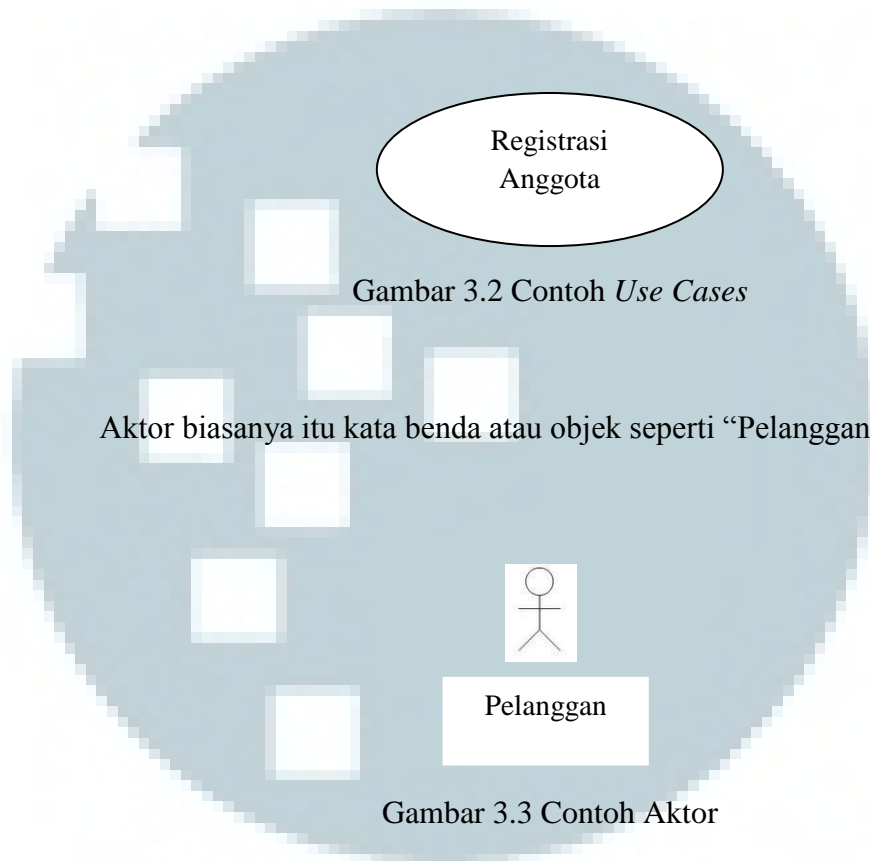
1. Memahami Kebutuhan

Pada tahap pengumpulan *requirement* ini penulis menggunakan metode *interview* yang tidak terstruktur. *Interview* tidak terstruktur ini maksudnya tanpa adanya standar gol atas hasil dari *interview* dan tanpa mempersiapkan pertanyaan terlebih dahulu (Gatewood, 2008, h. 504).

2. Desain Sistem

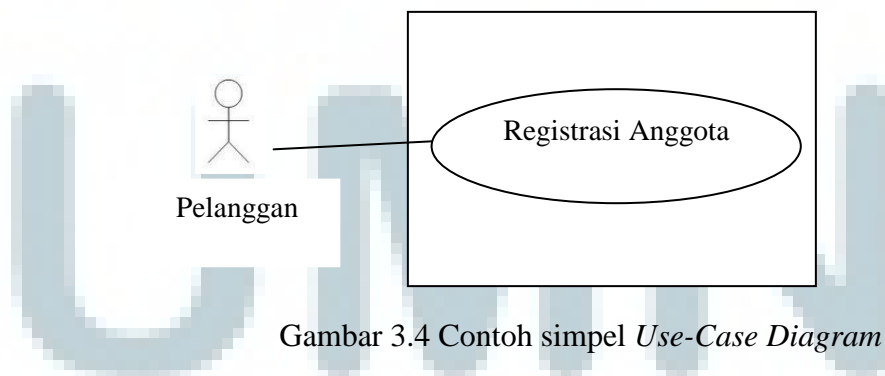
Untuk desain sistem, penulis menggunakan *Use-Case Diagram* untuk menjelaskan tingkah laku dari sistem. *Use-Case Diagram* ini terdiri atas *use cases*, aktor dan *associations*.

Use cases biasanya adalah proses atau kata kerja seperti “Registrasi Anggota”.



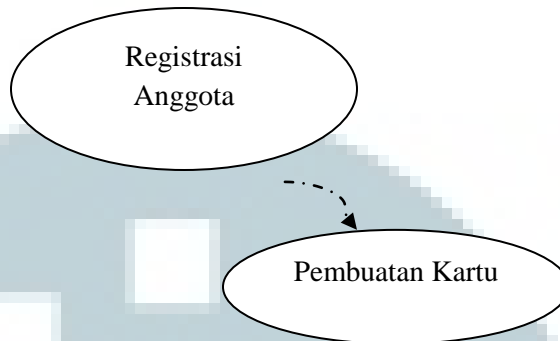
Gambar 3.2 Contoh *Use Cases*

Aktor biasanya itu kata benda atau objek seperti “Pelanggan”.



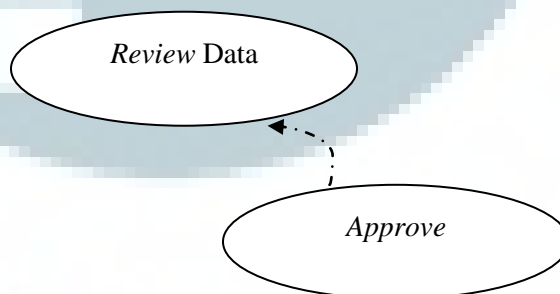
Gambar 3.4 Contoh simpel *Use-Case Diagram*

Pada gambar 3.4 dijelaskan bahwa terdapat seorang aktor yaitu Pelanggan yang mempunyai kemampuan untuk melakukan proses registrasi anggota (Bitner, 2003).



Gambar 3.5 Use Case Inclusion

Include statement ini maksudnya *use case* yang tidak diselesaikan oleh *use case* itu sendiri melainkan dengan *use case* lain juga. Pada gambar 2.4 menjelaskan bahwa registrasi anggota *include* pembuatan kartu, dimana artinya apabila registrasi anggota baru maka berikut dengan pembuatan kartu.



Gambar 3.6 Use Case Extension

Extend statement merupakan sebuah kemampuan untuk menambahkan fungsi tambahan di *use case* utama pada sebuah kondisi tertentu. Seperti *use-case* pada gambar 3.6 yang menjelaskan *approve* bisa dilakukan atau tidak ketika sedang me-*review* data. Dalam kasus ini *approve* bisa dilakukan tergantung pilihan aktor(Pilone, 2005).

Sedangkan untuk data yang didapat dari kebutuhan setiap proses terkait sistem, akan dinormalisasi, sehingga hasil dari normalisasi ini akan menunjukkan relasi antar tabel yang dibentuk dalam bentuk *ERD*.

Normalisasi merupakan proses bertahap yang memungkinkan pengguna untuk mengidentifikasi adanya kemungkinan masalah *update anomalies* dan meminimalisasikan atau bahkan menghilangkan potensi-potensi masalah tersebut. Biasanya normalisasi dilakukan pada tahap *first normal form*, *second normal form* dan *third normal form*, maka data sudah normal (Pratt, 2011).

ERD merupakan sebuah model untuk menunjukkan relasi *logical* dan interaksi antar entitas di dalam sebuah ekosistem sistem. *ERD* ini digunakan untuk membuat struktur *physical data*. Terdapat tiga tipe relasi antar entitas yaitu *one-to-one*, *one-to-many*, dan *many-to-many*. (Shelly, 2009).

3. Membangun Sistem

Untuk pengembangan sistem, penulis menggunakan *IDE Microsoft Visual Studio 2008* dengan bahasa pemrograman *VB.NET* dan untuk *database* menggunakan *Microsoft SQL SERVER 2005 Express Edition*.

IDE Microsoft Visual Studio ini digunakan untuk membuat konsol dan tampilan antarmuka aplikasi baik itu aplikasi desktop ataupun aplikasi web. Para pengguna bisa memilih bahasa pemrograman yang *Visual Studio* sediakan seperti *C++*, *C#*, dan *VB.NET* untuk pengembangan sistem (Wikipedia, 2013).

Microsoft SQL SERVER 2005 Express Edition ini didesain untuk menyediakan *database platform* yang mudah digunakan dalam hal pembuatan *database*, dan memudahkan untuk intergrasi dengan proyek *Visual Studio*(George, 2005).

4. Tes dan Evaluasi

Untuk *testing* penulis menggunakan metode *testing blackbox*. Metode ini mempertimbangkan *input*, *output* dan fungsi-fungsi dari sistem, apakah sudah sesuai dengan spesifikasi *requirement* atau skenario yang ditentukan.(Limaye 2009,h. 108). Pada tahapan ini sistem akan dites oleh pengguna di *laptop* penulis, jadi masih dijalankan di *localhost*. Sistem akan dievaluasi oleh pengguna apakah sudah sesuai atau ada yang harus direvisi, baik itu dari sisi tampilan dan fungsi.

3.3.3. Detil Pelaksanaan Penulis di Setiap Tahapan

1. Pertemuan Pertama

➤ Memahami Kebutuhan

Pada minggu pertama ini, penulis mendapatkan gambaran mengenai sistem yang dijelaskan oleh pemilik perusahaan. Penjelasan pertama yaitu mengenai tingkatan pengguna aplikasi berbasis web ini, pertama yaitu tingkatan superadmin (pemilik perusahaan), *admin* (karyawan *HRD*), dan *user* (karyawan *IT*). Dari masing-masing tingkatan pengguna itu memiliki tampilan dan fungsi yang berbeda sesuai tingkatannya.

i. Superadmin

Superadmin memiliki fungsi untuk membuat ID pengguna kepada karyawan yang tingkatan penggunaannya admin dan *user*. Pembuatan ID ini juga berikutan dengan pembuatan NIK yang dibuat berdasarkan tanggal masuk kerja si karyawan. Jadi apabila karyawan tersebut masuk kerja pada tanggal 10 Maret 2013 dan kebetulan dia karyawan pertama di bulan dan tahun itu, maka NIK yang dibentuk 03201301. 03 melambangkan bulan Maret, 2013 melambangkan tahun 2013, dan 01 melambangkan karyawan pertama. Untuk ID yang digunakan adalah *email* si karyawan, dan untuk *password* dibentuk secara acak terlebih dahulu yang terdiri atas lima karakter. Pemberitahuan NIK, ID pengguna, dan password karyawan akan otomatis dikirim melalui *email* setelah *user ID* dibuat.

Karyawan yang sudah dibuatkan *user ID* akan diminta untuk melakukan penggantian *password* terlebih dahulu pada saat pertama kali *login*, baik itu admin ataupun *user*. Pemilik memiliki aturan standar untuk *password*, yaitu minimal enam karakter, terdiri atas satu huruf besar, satu huruf kecil, dan satu spesial karakter. Lalu setelah berhasil ganti *password* maka akan langsung masuk ke sistem sesuai dengan otoritas pengguna.

Untuk tingkatan pengguna admin dan *user*, mereka sama-sama memiliki tampilan untuk melakukan pengisian data karyawan karena mereka sama-sama berstatus karyawan, tetapi hanya tugasnya saja yang berbeda. Data karyawan memiliki status data yaitu *draft*, *review* dan *final*. *Draft* berarti data masih dalam keadaan *editing* atau pengisian oleh karyawan, *review* berarti data sudah di-*submit* oleh

karyawan ke *HRD*, dan *final* berarti data karyawan tersebut sudah di-*approve* oleh *HRD*. Data karyawan ini bisa disimpan terlebih dahulu, jadi status masih *editing* atau *draft*, dan bisa juga langsung *submit* dimana merubah kondisi data menjadi *review* dan menunggu data diproses oleh pihak *HRD*. Proses *submit* akan memberikan notifikasi melalui email kepada *HRD* dengan isi notifikasi siapa yang melakukan *submit* data sehingga data dapat langsung diproses. Pengisian data karyawan ini terdiri atas data umum seperti nama depan, nama belakang, alamat tempat tinggal sekarang, alamat identitas, nomor handphone pertama dan nomor handphone kedua, nomor telepon rumah, jenis kelamin, tempat dan tanggal lahir, email, dan hobi. Selain itu terdapat data spesifik juga seperti data pendidikan formal yang terdiri atas nama institusi, tingkat pendidikan, tahun masuk dan tahun keluar, data pendidikan nonformal yang terdiri atas nama *training*, agen atau instansi yang menyelenggarakan, tahun masuk dan tahun keluar, dan keluarga. Keluarga di sini maksudnya data keluarga karyawan, seperti nama, jenis kelamin, tempat dan tanggal lahir, pekerjaan, dan hubungannya dengan karyawan. Karena *user ID* menggunakan *email*, maka apabila terjadi perubahan *email* oleh karyawan, ini akan mempengaruhi *user ID* untuk *login* juga.

Tingkatan pengguna admin memiliki fungsi-fungsi lebih yang pengguna user tidak miliki seperti.

ii. Admin (karyawan *HRD*)

Admin memiliki tampilan untuk melakukan pencarian data karyawan berdasarkan NIK, nama lengkap, dan juga status data (*draft,review,final*). Untuk keadaan data karyawan yang berstatus *review*, dalam arti data ini sudah di-*submit* dan menunggu dari sisi

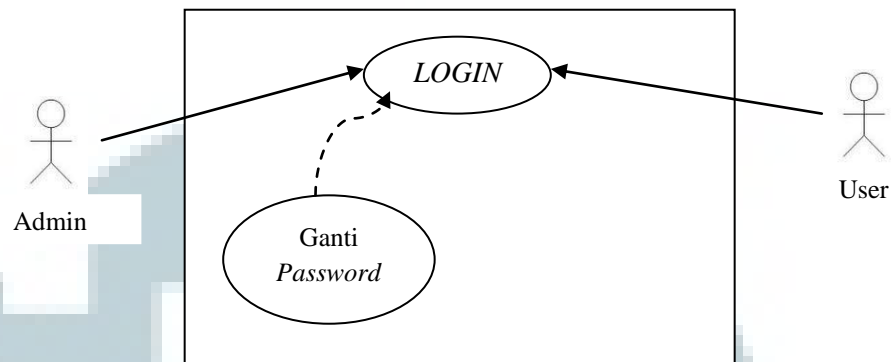
admin untuk melakukan *approve* atas data tersebut atau *reject* karena data masih belum lengkap. Tingkatan pengguna admin ini bisa melakukan perubahan data karyawan baik itu data yang berkondisi *draft*, *review* ataupun *final*. Jadi admin bisa melakukan perubahan data karyawan jika terdapat data yang salah, dan sama halnya apabila terjadi perubahan email dari sisi admin maka akan mempengaruhi *user ID* untuk *login* si karyawan tersebut.

Admin juga memiliki tampilan untuk membuatkan *user ID* untuk karyawan IT atau yang tingkatan penggunaanya *user*. Admin tidak bisa membuatkan *user ID* untuk sesama admin. Proses pembuatannya sama dengan sisi superadmin, tetapi bedanya tidak ada pemilihan tingkatan pengguna yang dibuatkan, jadi sudah pasti *user ID* untuk tingkatan pengguna *user*.

➤ Desain Sistem

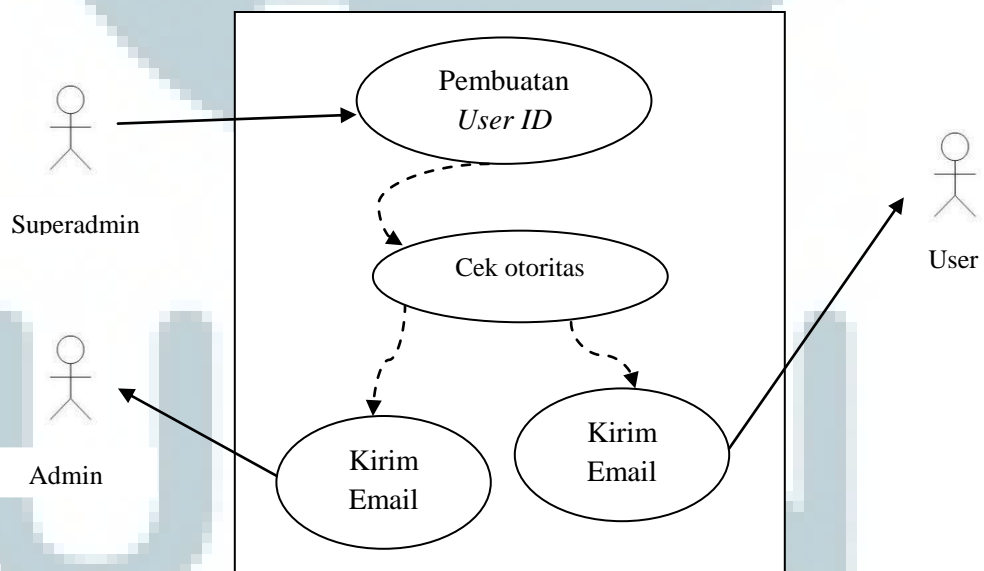
Berdasarkan gambaran mengenai fungsi dari masing-masing pengguna, maka penulis membentuknya dalam bentuk usecase diagram untuk melihat fungsi dari masing-masing pengguna di dalam sistem secara umum.

U
M
M
N



Gambar 3.7 Usecase Diagram *Login Karyawan*

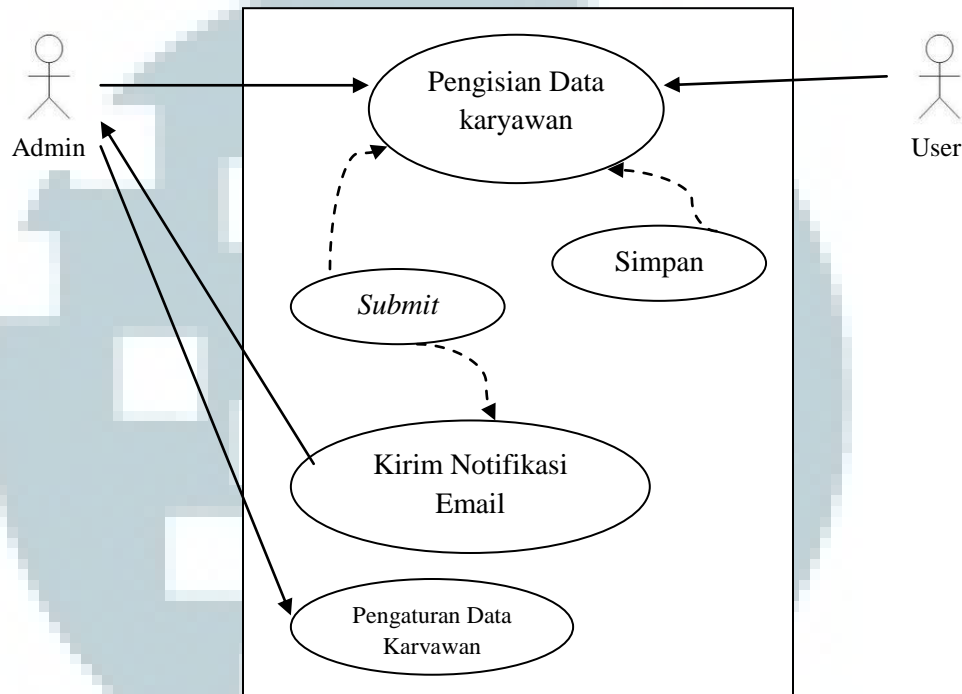
Gambar di atas menjelaskan bahwa karyawan dengan tingkatan pengguna admin ataupun *user* yang melakukan *login* pada pertama kali harus melakukan *ganti password* terlebih dahulu



Gambar 3.8 Usecase Diagram Superadmin Pembuatan *User ID*

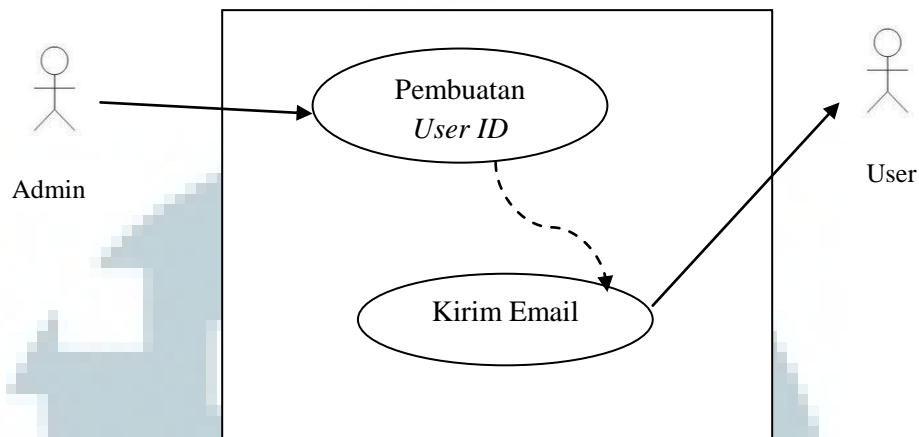
Gambar 3.8 menjelaskan fungsi dari superadmin untuk proses pembuatan *user ID* bagi karyawan, baik itu untuk pengguna admin

ataupun *user*. *Use-case* cek otoritas menjelaskan proses pengecekan apakah dibuatkan untuk tingkatan pengguna admin atau user.



Gambar 3.9 Usecase Diagram Administrasi Data Karyawan

Gambar di atas menjelaskan proses pengisian data karyawan. Setelah pengisian data selesai dilakukan karyawan bisa *submit* datanya ke *HRD* atau bisa melakukan menyimpan sementara datanya untuk dilanjutkan pengisiannya nanti. Untuk *use-case* pengaturan data karyawan menjelaskan fungsi dari admin untuk melakukan perubahan data karyawan apabila terdapat kesalahan data atau melakukan *approving* dan *rejecting* terhadap data karyawan yang sudah di-*submit*.



Gambar 3.10 Usecase Diagram Administrasi *User ID*

Gambar 3.10 menjelaskan fungsi dari sisi admin yang bisa membuat *user ID* untuk karyawan yang tingkatan penggunaanya *user*.

Berdasarkan kebutuhan data dari masing-masing fungsi pengguna di dalam sistem, maka perlu disimpan data-data seperti data karyawan, data untuk *login*, dan data peranan setiap tingkatan pengguna sebagai berikut.

Atribut	Tipe Data	Keterangan
Role_name	Varchar(10)	Nama peranan (superadmin, admin, dan <i>user</i>)

Tabel 3.1 *Roles*

Atribut	Tipe Data	Keterangan
User_id	Varchar(50)	Diisi dengan email untuk ID login.
Password	Varchar(150)	Password untuk login
Login_status	Varchar(10)	Untuk cek apakah pertama kali login atau tidak

Tabel 3.2 *Users*

Atribut	Tipe Data	Keterangan
First_name	Varchar(30)	Nama depan
Last_name	Varchar(30)	Nama belakang
Current_address	Varchar(100)	Alamat sekarang
Address_id	Varchar(100)	Alamat identitas diri
Phone1	Varchar(20)	Nomor handphone 1
Phone2	Varchar(20)	Nomor handphone 2
Home_number	Varchar(20)	Nomor telepon rumah
Gender	Varchar(6)	Jenis kelamin
Date_of_birth	Datetime	Tanggal, bulan dan tahun lahir
Place_of_birth	Varchar(20)	Tempat lahir
Email	Varchar(50)	Email
Hobby	Varchar(75)	Hobi
Status	Varchar(6)	Untuk mengetahui status data karyawan apakah <i>draft</i> , <i>review</i> atau <i>final</i>

Table 3.3 *Employees*

Atribut	Tipe Data	Keterangan
Institution_name	Varchar(50)	Nama institusi
Stage	Varchar(15)	Tingkatan pendidikan
Year_in	Varchar(4)	Tahun mulai
Year_out	Varchar(4)	Tahun selesai

Tabel 3.4 *Formal Educations*

Atribut	Tipe Data	Keterangan
Name_of_training	Varchar(50)	Nama pelatihan
Agency	Varchar(50)	Penyelenggara
Year_in	Varchar(4)	Tahun mulai
Year_out	Varchar(4)	Tahun selesai

Tabel 3.5 *Non Formal Educations*

Atribut	Tipe Data	Keterangan
Name	Varchar(50)	Nama lengkap
Gender	Varchar(6)	Jenis kelamin
Place_of_birth	Varchar(20)	Tempat lahir
Date_of_birth	Datetime	Tanggal, bulan dan tahun lahir
Job	Varchar(30)	Pekerjaan
Relationship	Varchar(50)	Hubungan dengan karyawan

Tabel 3.6 *Family*

NORMALISASI

#ROLES

(**role_id (PK)**, role_name)

#USERS

(**user_id (PK)**, password, **NIK (FK)**, **role_id (FK)**, login_status)

#EMPLOYEES

UNF

(**NIK**, {institution_name, stage, year_in, year_out, name_of_training, agency, year_in, year_out, name, gender, place_of_birth, date_of_birth, job, relationship}, first_name, last_name, current_address, address_id, phone1, phone2, home_number, gender, date_of_birth, place_of_birth, email, hobby, status)

1NF

(**NIK**, institution_name, stage, year_in, year_out, name_of_training, agency, year_in, year_out, name, gender, place_of_birth, date_of_birth, job, relationship, first_name, last_name, current_address, address_id, phone1, phone2, home_number, gender, date_of_birth, place_of_birth, email, hobby, status)

2NF

(**NIK**, {institution_name, stage, year_in, year_out}, {name_of_training, agency, year_in, year_out}, {name, gender, place_of_birth, date_of_birth, job, relationship}, first_name, last_name, current_address, address_id, phone1, phone2,

home_number, gender, date_of_birth, place_of_birth, email, hobby, status)

3NF

#EMPLOYEES

(NIK (PK), first_name, last_name, current_address, address_id, phone1, phone2, home_number, gender, date_of_birth, place_of_birth, email, hobby, status)

#FORMAL_EDUCATIONS

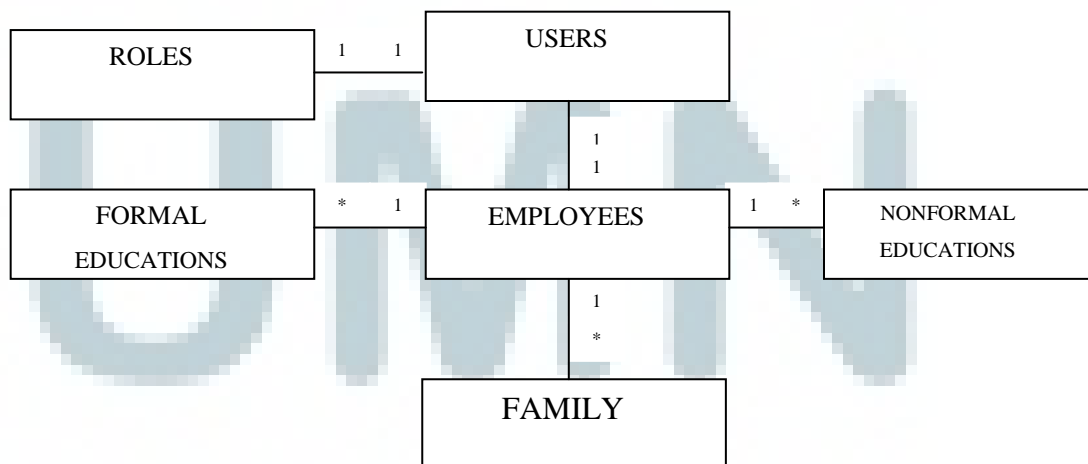
(NIK (FK), institution_name, stage, year_in, year_out)

#NONFORMAL_EDUCATIONS

(NIK (FK), name_of_training, agency, year_in, year_out)

#FAMILY

(NIK (FK), name, gender, place_of_birth, date_of_birth, job, relationship)



Gambar 3.11 ERD

Dari hasil normalisasi data didapat relasi antar tabel pada gambar 3.11 yang dijelaskan sebagai berikut.

Tabel *employees* mempunyai hubungan kardinalitas *one-to-one* dengan table *users*, ini berarti satu orang karyawan mempunyai sebuah *user ID* dan satu *user ID* dimiliki oleh satu orang karyawan. Untuk *Users* dengan *Roles* hubungan kardinalitasnya juga *one-to-one*, dimana sebuah *user ID* mempunyai satu tingkatan pengguna.

Untuk relasi antar tabel *employees* dengan *formal educations*, *nonformal educations*, dan *family*, hubungan kardinalitasnya adalah *one-to-many*. Jadi seorang karyawan bisa mempunyai banyak data di tabel *formal educations*, *nonformal educations*, dan *family*, sedangkan satu baris data di tabel *formal*, *nonformal* dan *family* hanya dimiliki oleh seorang karyawan.

➤ **Membangun Sistem**

Di sini dimulai tahap pembuatan database sesuai desain *database*, dan proses pembuatan sistem dimulai dengan pembentukan halaman-halaman untuk ketiga tingkatan pengguna. Halaman yang ada untuk masing-masing tingkatan pengguna dibuat berdasarkan desain *Use-case Diagram* yang sudah dibuat.

➤ **Tes dan Evaluasi**

Pada tahap ini tes untuk pengisian data atas skenario tertentu belum dilakukan, jadi masih revisi atas tampilan sistem dan kekurangan sistem.

Revisi

- i. Penambahan data nomor NPWP dan nomor jamsostek di data karyawan
- ii. Menambahkan site map path
- iii. Memperbaiki *GUI* yang tadinya masih desain penulis, sekarang disesuaikan dengan keinginan pemilik perusahaan.
- iv. Penambahan fungsi untuk menonaktifkan karyawan dari sisi admin.

2. Pertemuan Kedua

➤ Memahami Kebutuhan

Untuk fungsi menonaktifkan karyawan, itu ada di sisi admin, dimana dinonaktifkan dengan cara menginputkan tanggal berhentinya karyawan.

➤ Desain Sistem

Penambahan data nomor NPWP dan nomor jamsostek, dan tanggal berhenti, serta status karyawan aktif atau tidak aktif pada tabel 3.3

➤ Membangun Sistem

Merubah desain tampilan sesuai dengan keinginan pemilik perusahaan.

Berikut contoh tampilan yang disesuaikan dengan keinginan pemilik.

accelist^Å

Login	
User ID :	<input type="text"/>
Password:	<input type="password"/>
<input type="button" value="Login"/>	

Gambar 3.12 *Login*

accelist^Å

Administration > Create User ID

Administration ▶

Gambar 3.13 *Header bagian Kiri*

PT Accelist Lentera Indonesia

Talavera Office Park, 28th floor
JL. T. B. Simatupang Kav. 22-26
Jakarta 12430 - Indonesia

Phone: +6221 7599 7932 | Fax: +6221 7599 9888
www.accelist.com | email: customer.service@accelist.com

Logout

Gambar 3.14 *Header bagian Kanan*

Please Change Your Password First

User ID

Password

New Password

New Password Confirmation

Gambar 3.15 Ganti *Password* Setelah *Login* Pertama Kali

NIK Full Name Status

NIK	Name	Status	Role Name	Employee Status	Detail
05201301	David Tan	draft	hrd	active	Detail

Gambar 3.16 Pencarian Data Karyawan

INSERT DATA

User ID(Email)

Hire Date

First Name

Middle Name

Last Name

NIK

Password

Gambar 3.17 Pembuatan *ID* Pengguna

NIK : 05201301 **EDITING(DRAFT)**

GENERAL INFORMATION

First Name *	<input type="text" value="David"/>
Middle Name	<input type="text"/>
Last Name	<input type="text" value="Tan"/>
Title	<input type="text"/>
Place of Birth *	<input type="text"/>
Date of Birth *	<input type="text"/>

Gambar 3.18 Pengisian Data Karyawan

Menambahkan kotak teks untuk NPWP dan jamsostek ke halaman web terkait pengisian data karyawan.

No. NPWP	<input type="text"/>
If you have previously participated in Jamsostek? *	<input type="radio"/> Yes <input type="radio"/> No
No. KPJ *	<input type="text"/>

Gambar 3.19 Penambahan NPWP dan Jamsostek

Halaman admin terkait pengaturan data karyawan, ditambahkan *button* untuk menonaktifkan karyawan, dan untuk mengaktifkannya lagi di *button* halaman itu juga.

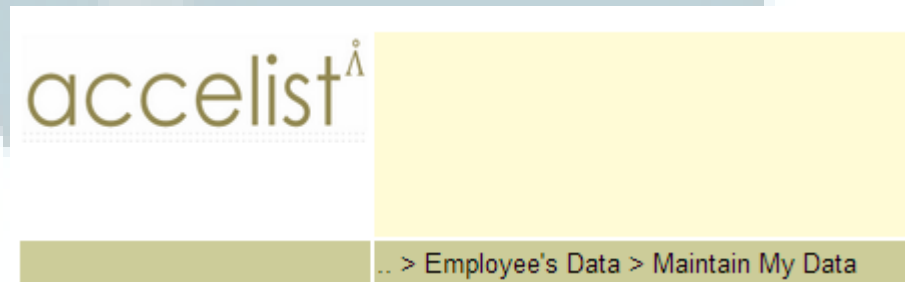
Resign Date

Gambar 3.20 Menonaktifkan Karyawan

Resign Date

Gambar 3.21 Mengaktifkan Karyawan

Menambahkan fungsi *site map path* ke web. *Site map path* ini menjelaskan posisi halaman yang sedang diakses pengguna.



Gambar 3.22 Site Map Path

➤ Tes dan Evaluasi

Tes dilakukan oleh Bpk. Matius Kelvin selaku pembimbing lapangan

Skenario	Hasil
Pengaksesan halaman web tanpa login	Tidak bisa akses halaman web tanpa login terlebih dahulu
Pengaksesan halaman web lain yang tidak sesuai otoritas	Pengaksesan halaman web sudah dijaga berdasarkan otoritas pengguna
Password baru tidak sesuai format yang ditentukan pemilik	Tidak bisa ganti password jika password baru tidak sesuai format yang ditentukan
User ID atau Email yang tidak sesuai format	Tidak bisa membuat user ID jika email formatnya belum benar

Tabel 3.7 Skenario *Testing*.

Revisi

- i. Perubahan data karyawan mengikuti kebutuhan data jamsostek
- ii. Enrkripsi data password yang ada di database
- iii. Menggunakan *ajax toolkit* untuk merubah kotak teks terkait pengisian tanggal seperti tanggal lahir, tanggal masuk kerja yang tadinya masih menggunakan *dropdownlist*, menjadi kotak teks yang bisa untuk melakukan pencarian seperti halnya kalender.
- iv. Penambahan fungsi, apabila terjadi perubahan data lagi pada saat status data karyawan itu *final* maka akan tercatat riwayat perubahan datanya di *database*.

3. Pertemuan Ketiga

➤ Memahami Kebutuhan

Penjelasan mengenai administrasi data absensi karyawan. Untuk tingkatan pengguna admin dan user sama-sama mempunyai tampilan untuk melakukan pengisian absensi. Terdapat dua jenis tampilan, pertama itu tampilan untuk langsung mengisi absensi pada bulan dan tahun sekarang, dan tampilan satu lagi untuk melakukan pencarian terhadap absennya berdasarkan bulan dan tahun yang diisikan. Pada saat melakukan absensi apabila dipilih tidak hadir, maka akan ada pilihan apakah hari itu libur, karyawan izin, sakit, dan lainnya. Jika dipilih lainnya maka akan ada kotak teks untuk mengisi sendiri alasan kenapa karyawan tidak hadir. Untuk metode penyimpanan datanya, ada simpan untuk sementara dan ada simpan sebagai komplit. Simpan sebagai komplit dapat dilakukan jika semua tanggal di bulan itu sudah terisi semua absensinya, berikut dengan penjelasan apabila karyawan tersebut tidak hadir.

Pencatatan sejarah perubahan data pada saat status data karyawan sudah *final*. Data yang dicatat adalah tanggal berubahnya data tersebut, data apa yang berubah, serta data lama dan data barunya.

➤ Desain Sistem

Penyesuaian data karyawan dengan data jamsostek yang ditambahkan ke tabel 3.3

Berikut data untuk pencatatan perubahan data pada saat kondisi data *final*.

Atribut	Tipe Data	Keterangan
Date_changed	Datetime	Tanggal berubahnya data
Field_changed	Varchar(50)	<i>Field</i> apa yang berubah datanya
Old_data	Varchar(50)	Data lama
New_data	Varchar(50)	Data baru

Tabel 3.8 *History Data Changes*

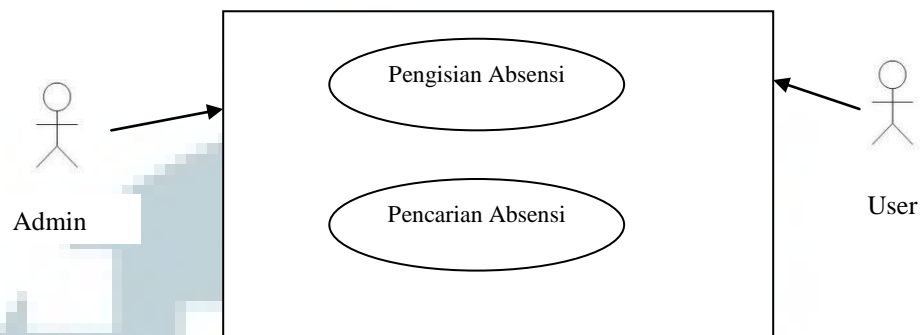
#HISTORY_DATA_CHANGES

(**NIK (FK)**, date_changed, field_changed, old_data, new_data)

Terkait penjelasan mengenai absensi, penulis meng-*generate* absensi untuk karyawan berdasarkan tanggal masuk kerja, jadi begitu karyawan itu dibuatkan *user ID*, otomatis akan dibuatkan *template* absensi dari tanggal masuk kerja sampai akhir bulan tahun ini. Maka dari itu dibutuhkan tambahan tabel kalender di database yang digunakan untuk *generate* absensi karyawan begitu karyawan itu dibuatkan user ID.

Atribut	Tipe Data	Keterangan
calendarDate	Datetime	Berisikan tanggal, bulan dan tahun

Tabel 3.9 *Calendar*



Gambar 3.23 Usecase Diagram Administrasi Data Absensi Karyawan

Gambar di atas menjelaskan fungsi tambahan untuk tingkatan pengguna admin dan *user* yang bisa mengisi absen dan melakukan pencarian terhadap absen nya sendiri di web.

Untuk kebutuhan data yang disimpan dalam database terkait pengisian absensi sebagai berikut.

Atribut	Tipe Data	Keterangan
Month_attendances	Varchar(10)	Bulan absen
Year_attendances	Varchar(4)	Tahun absen
Status_attendances	Varchar(10)	Kondisi absen, apakah komplit atau tidak.

Tabel 3.10 *Employee Attendances History*

Atribut	Tipe Data	Keterangan
Date_attendances	Datetime	Tanggal, bulan dan tahun
Emp_attendances	Varchar(3)	Kehadiran karyawan
Emp_expl	Varchar(7)	Alasan tidak hadir(sakit,ijin,libur,cuti

		atau lainnya)
Other	Varchar(50)	Alasan lain jikalau tidak hadir

Tabel 3.11 *Employee Attendances Detail*

NORMALISASI

UNF

(**id**, month_attendances, year_attendances, {date_attendances, emp_attendances, emp_expl, other}).

1NF

(**id**, month_attendances, year_attendances, date_attendances, emp_attendances, emp_expl, other)

2NF

(**id**, month_attendances, year_attendances, {date_attendances, emp_attendances, emp_expl, other}).

3NF

#EMP_ATTENDANCES_HIST

(**id (PK)**, month_attendances, year_attendances)

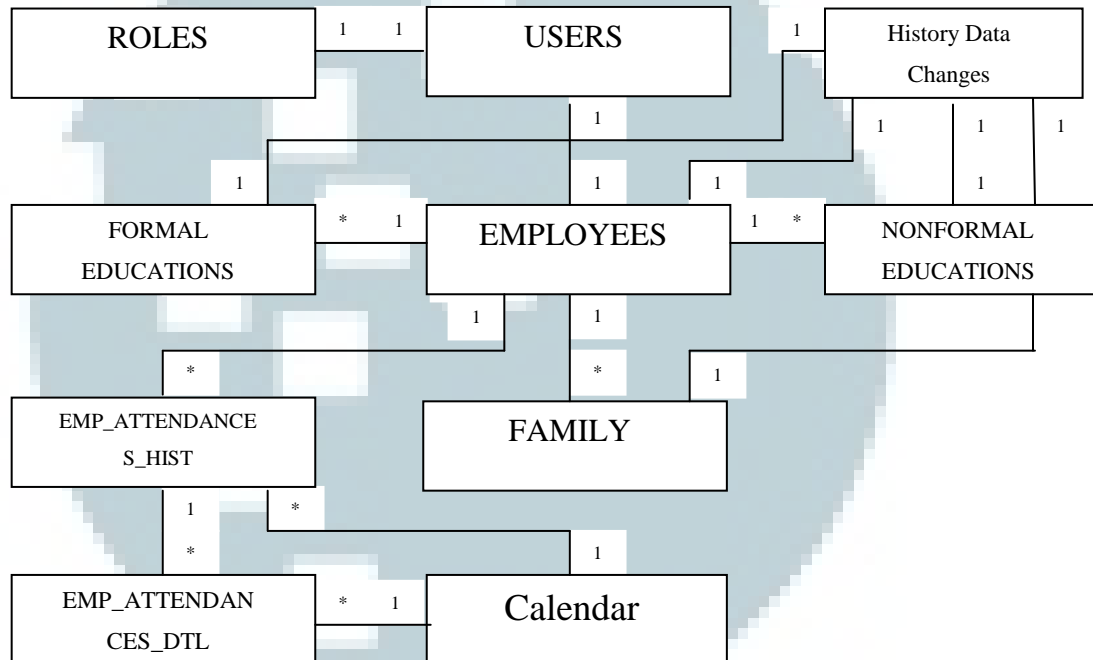
#EMP_ATTENDANCES_DTL

(**id (FK)**, date_attendances, emp_attendances, emp_expl, other)

Untuk menghubungkan antara absensi dengan karyawan maka menggunakan NIK sebagai penghubung antara tabel.

#EMP_ATTENDANCES_HIST

(**id (PK)**, **NIK (FK)**, month_attendances, year_attendances)



Gambar 3.24 ERD Revisi

Terdapat penambahan tabel *calendar* dan *history data changes* pada ERD. Hubungan kardinalitas antara *history data changes* dengan *employees*, *nonformal educations*, dan *family* adalah *one-to-one* ini dikarenakan setiap perubahan data baik itu dari tabel *employees*, *nonformal*, *formal* dan *family* akan tercatat satu rekor di tabel *history data changes* dan satu rekor di tabel *history data changes* merupakan pencatatan atas sebuah perubahan data di antara tabel *employees*, *nonformal*, *formal* dan *family*. Sedangkan hubungan kardinalitas antara tabel *calendar* dengan tabel absensi *history* dan absensi detail

adalah *one-to-many*, ini dikarenakan data-data terkait tanggal, bulan dan tahun di tabel absensi *history* ataupun detail dari absensi *history* di-generate berdasarkan isi dari tabel *calendar*.

➤ **Membangun Sistem**

Tambahan dua buah tampilan terkait absensi karyawan, tampilan untuk absensi bulan ini, dan tampilan untuk pencarian absensi berdasarkan bulan dan tahun.

Date	Attendance	Explanation	Other
01 Nov 2013	<input type="radio"/> yes <input checked="" type="radio"/> no	<input type="text" value="lainnya"/>	<input type="text" value="aaa"/>
02 Nov 2013	<input type="radio"/> yes <input checked="" type="radio"/> no	<input type="text" value="sakit"/>	
03 Nov 2013	<input type="radio"/> yes <input checked="" type="radio"/> no	<input type="text" value="libur"/>	
04 Nov 2013	<input type="radio"/> yes <input checked="" type="radio"/> no	<input type="text" value="ijin"/>	
05 Nov 2013	<input checked="" type="radio"/> yes <input type="radio"/> no		
06 Nov 2013	<input checked="" type="radio"/> yes <input type="radio"/> no		
07 Nov 2013	<input checked="" type="radio"/> yes <input type="radio"/> no		

Gambar 3.25 Pengisian Absensi Karyawan

From : To :

Months	Year	Status	Detail
MAY	2013	complete	<input type="button" value="Detail"/>
JUNE	2013	complete	<input type="button" value="Detail"/>

Gambar 3.26 Pencarian Absensi per Bulan dan Tahun

Data absensi yang tadi di-generate akan otomatis terisi hadir dan komplit sampai sebelum bulan dan tahun sekarang.

Password karyawan sudah dienkripsi di *database*.

Perubahan tampilan untuk halaman-halaman yang terkait dengan administrasi data karyawan, karena data sudah mengikuti data jamsostek.

Last Name	<input type="text" value="Tan"/>
Title	<input type="text"/>
Place of Birth *	<input type="text"/>
Date of Birth *	<input type="text"/>
Identity *	<input type="radio"/> KTP <input type="radio"/> Paspor
Identity Number *	<input type="text"/>
Valid Until *	<input type="text"/>
Address ID *	<input type="text"/>
Current Address *	<input type="text"/>
Zip Code *	<input type="text"/>
Mother's Maiden Name *	<input type="text"/>
Last Education *	<input type="radio"/> SD <input type="radio"/> D3 <input type="radio"/> S2 <input type="radio"/> SMP <input type="radio"/> S1 <input type="radio"/> S3 <input type="radio"/> SMA
Marital Status *	<input type="radio"/> Not Married <input type="radio"/> Married
Gender *	<input type="radio"/> Male <input type="radio"/> Female

Gambar 3.27 Contoh pengisian data karyawan sesuai data jamsostek

➤ Tes dan Evaluasi

Tes

Penambahan skenario tes perubahan tanggal kerja lagi begitu NIK sudah di-*generate* untuk pembuatan *user ID* karyawan di tabel 3.7. Hasil dari tes ini NIK harus sesuai dengan tanggal masuk kerja karyawan baru bisa dibuatkan *user ID*.

Revisi

- i. Penambahan user siapa yang sedang *login*

4. Pertemuan Keempat

➤ Memahami Kebutuhan

Menambahkan informasi mengenai user siapa yang sedang *login* di setiap halaman akses karyawan.

➤ Desain Sistem

-

➤ Membangun Sistem

Menambahkan label yang akan diisi dengan nama karyawan begitu halaman itu diakses sesuai dengan karyawan yang sedang *login*.

PT Accelist Lentera Indonesia

Talavera Office Park, 28th floor
JL. T. B. Simatupang Kav. 22-26
Jakarta 12430 - Indonesia

Phone: +6221 7599 7932 | Fax: +6221 7599 9888

www.accelist.com | email: customer.service@accelist.com

Logout

Welcome, David

Gambar 3.28 Nama Pengguna yang Login

➤ Tes dan Evaluasi

Tes

-

Revisi

i. Penambahan fungsi cetak absensi untuk Admin ataupun *User*.

ii. Admin bisa *maintain* data absensi jika ada kesalahan

5. Pertemuan Kelima

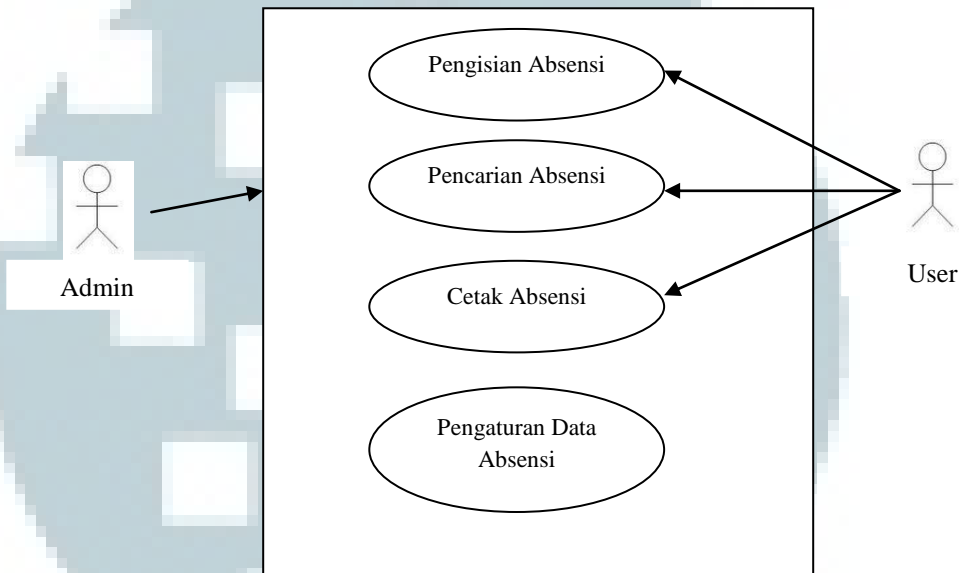
➤ Memahami Kebutuhan

Terdapat tambahan tampilan baru untuk tingkatan pengguna admin dan juga *user*. Proses dari cetak absensi ini dimulai dengan menginputkan periode tanggal absensi yang karyawan ingin cetak, diketahui dan disetujui oleh siapa, beserta dengan jabatannya.

Untuk admin bisa *maintain* data absensi, jadi jika ada kesalahan data absensi karyawan, maka admin memungkinkan untuk melakukan perubahan data dengan melakukan pencarian terhadap absensi

karyawan yang bersangkutan. Pencarian dilakukan berdasarkan nama, nik dan bulan, tahun absensi si karyawan tersebut.

➤ Desain Sistem



Gambar 3.29 Usecase Diagram Administrasi Data Absensi Karyawan

Gambar di atas menjelaskan terdapat penambahan *use-case* cetak absensi dan pengaturan data absensi. Untuk cetak absensi bisa dilakukan oleh tingkatan pengguna admin dan *user*, dimana mereka bisa mencetak absensi mereka masing-masing melalui web. Sedangkan pengaturan data absensi ini untuk sisi admin dalam hal melakukan perubahan pada data absensi karyawan apabila terdapat kesalahan.

➤ Membangun Sistem

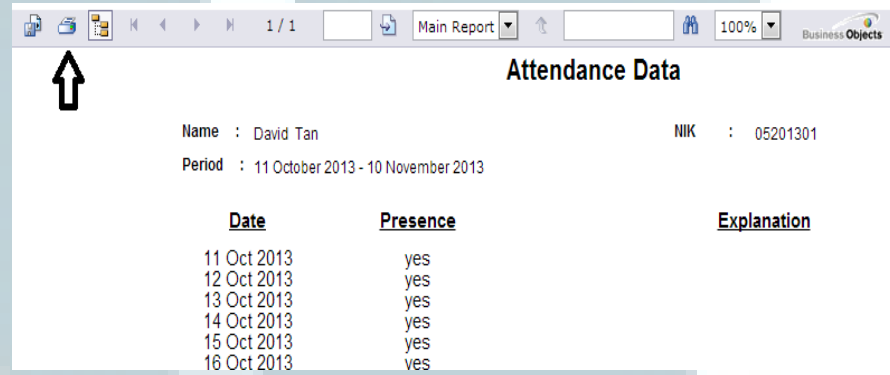
Membuat tampilan untuk cetak absen di kedua tingkatan pengguna admin dan user.

Name : David Tan
Period : 11 October 2013 - 11 November 2013

NIK : 05201301

Date	Presence	Explanation
11 Oct 2013	<input checked="" type="checkbox"/> yes <input type="checkbox"/> no	
12 Oct 2013	<input checked="" type="checkbox"/> yes <input type="checkbox"/> no	
13 Oct 2013	<input checked="" type="checkbox"/> yes <input type="checkbox"/> no	
14 Oct 2013	<input checked="" type="checkbox"/> yes <input type="checkbox"/> no	

Gambar 3.30 Hasil Pencarian Absensi yang Ingin Dicitak



Attendance Data

Name : David Tan
Period : 11 October 2013 - 10 November 2013

NIK : 05201301

Date	Presence	Explanation
11 Oct 2013	yes	
12 Oct 2013	yes	
13 Oct 2013	yes	
14 Oct 2013	yes	
15 Oct 2013	yes	
16 Oct 2013	ves	

Gambar 3.31 Cetak Absensi

UMMN

Attendance Data		
Name : David Tan	NIK : 05201301	
Period : 11 October 2013 - 10 November 2013		
<u>Date</u>	<u>Presence</u>	<u>Explanation</u>
11 Oct 2013	yes	
12 Oct 2013	yes	
13 Oct 2013	yes	
14 Oct 2013	yes	
15 Oct 2013	yes	
16 Oct 2013	yes	
17 Oct 2013	yes	
18 Oct 2013	yes	
19 Oct 2013	yes	
20 Oct 2013	yes	
21 Oct 2013	yes	
22 Oct 2013	yes	
23 Oct 2013	yes	
24 Oct 2013	yes	
25 Oct 2013	yes	
26 Oct 2013	yes	
27 Oct 2013	yes	
28 Oct 2013	yes	
29 Oct 2013	yes	
30 Oct 2013	yes	
31 Oct 2013	yes	
01 Nov 2013	no	
02 Nov 2013	no	sakit
03 Nov 2013	yes	
04 Nov 2013	yes	
05 Nov 2013	yes	
06 Nov 2013	yes	
07 Nov 2013	yes	
08 Nov 2013	yes	
09 Nov 2013	yes	
10 Nov 2013	yes	
Filled By	Known By	Approved By

Gambar 3.32 Hasil Cetak Absensi

Untuk admin ada tampilan baru untuk *maintain* absensi karyawan seperti gambar 3.33 dan 3.34.

Type to search Type to search

Full Name : NIK :

From : To :

Gambar 3.33 Pencarian Absensi Karyawan

Months	Year	Status	Update
JULY	2013	complete	<input type="button" value="Update"/>
AUGUST	2013	complete	<input type="button" value="Update"/>
SEPTEMBER	2013	complete	<input type="button" value="Update"/>
OCTOBER	2013	complete	<input type="button" value="Update"/>
NOVEMBER	2013	complete	<input type="button" value="Update"/>

Gambar 3.34 Hasil Pencarian Absensi yang Ingin di-Update

➤ Tes dan Evaluasi

Tes

Penambahan skenario tes perubahan email karyawan apakah mempengaruhi *user ID* atau tidak di tabel 3.7. Hasil dari tes ini apabila *email* berubah maka *user ID* untuk *login* juga akan berubah.

Revisi

- i. Tanggal masuk kerja karyawan bisa dirubah oleh admin
- ii. Penambahan captcha untuk login
- iii. Penambahan fungsi untuk langsung ke absensi bulan sebelumnya di halaman absensi bulan ini karyawan.

- iv. Untuk tampilan pada saat cetak absensi, pengisian data diketahui dan disetujui oleh, beserta dengan jabatannya apabila sudah pernah diisi akan otomatis terisi sesuai dengan pengisian terakhir.
- v. Untuk admin ada penambahan fungsi untuk penghapusan *User ID*
- vi. Penambahan fungsi ganti *password* dan lupa *password* bagi karyawan

6. Pertemuan Keenam

➤ Memahami Kebutuhan

Admin bisa merubah tanggal mulai kerja karyawan, ini bisa dilakukan untuk mencegah apabila terjadi kesalahan pada waktu pengisian tanggal mulai kerja.

Menambahkan captcha pada saat melakukan *login*, untuk mencegah adanya *bots* yang dapat membuat *server down*.

Untuk memudahkan pengguna dalam melakukan absensi maka pengguna bisa langsung ke absensi bulan sebelumnya pada saat melakukan absensi di bulan tertentu, tanpa harus melakukan pencarian absen terlebih dahulu.

Agar menghindari pengguna dalam melakukan penginputan berulang pada saat ingin mencetak absensi, karena biasanya dalam satu proyek karyawan itu berkoordinasi dengan orang yang sama. Jadi dalam pengisian diketahui oleh dan disetujui oleh beserta jabatannya itu tidak berulang.

Admin bisa melakukan penghapusan terhadap ID pengguna yang sudah dibuatkan, tetapi dengan satu kondisi yaitu data karyawan itu belum *ter-update* sama sekali.

Karyawan bisa melakukan perubahan *password* masing-masing di web demi keamanan *User ID*.

Karyawan juga bisa melakukan lupa *password*, untuk mengatasi apabila karyawan lupa *password*-nya. Proses lupa *password* ini dilakukan dengan memasukkan *user ID* atau *email* karyawan, lalu sistem akan meng-*generate password* baru untuk karyawan tersebut secara acak dan dikirim melalui *email* ke *user ID* tersebut.

➤ Desain Sistem

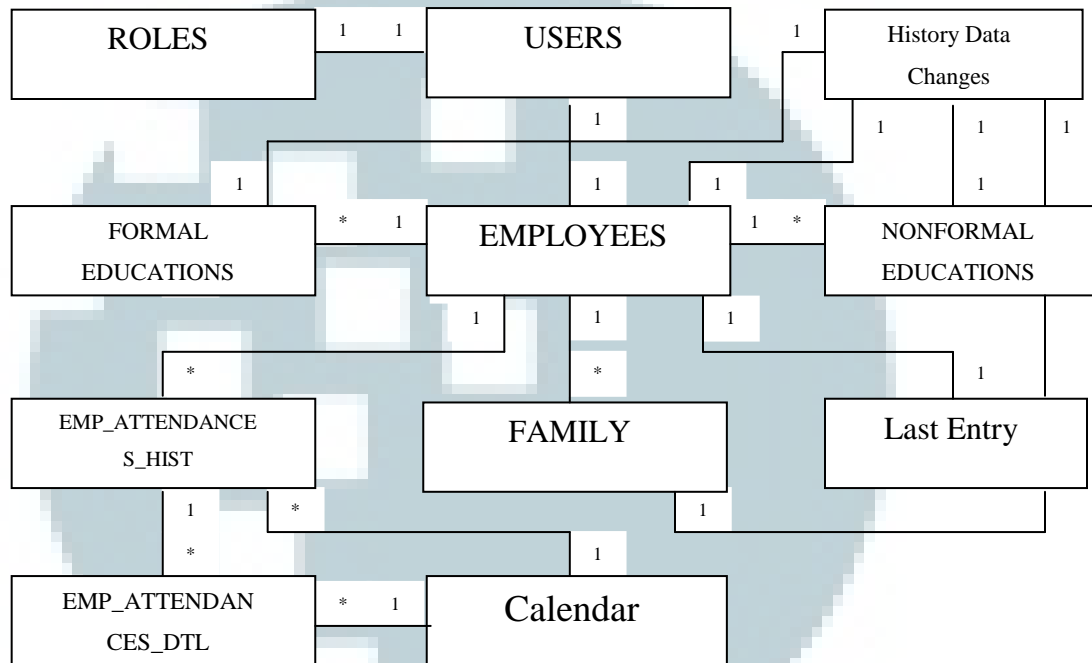
Untuk mengatasi masalah pencegahan input berulang terkait pencetakan data absensi. Maka penulis menambahkan tabel baru di database untuk masalah tersebut.

Atribut	Tipe Data	Keterangan
Known_by	Varchar(50)	Diketahui oleh
App_by	Varchar(50)	Disetujui oleh
Posisi_known	Varchar(50)	Posisi orang yang mengetahui proyek karyawan
Posisi_app	Varchar(50)	Posisi orang yang menyetujui proyek karyawan

Tabel 3.12 *Last Entry*

#LAST_ENTRY

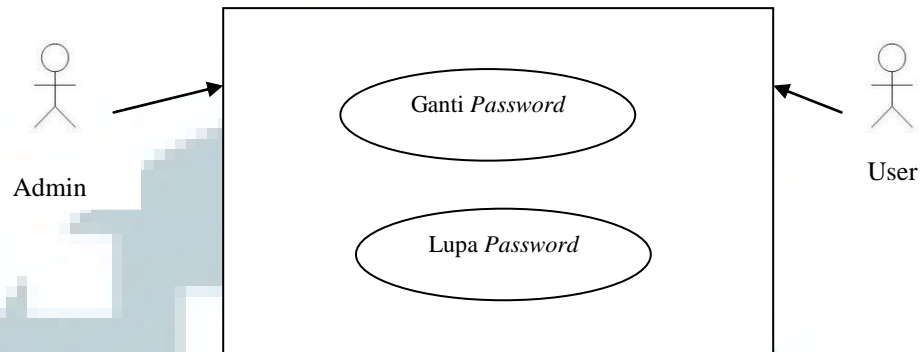
(NIK (FK), known_by, app_by, posisi_known, posisi_app)



Gambar 3.35 ERD Revisi

Gambar di atas menjelaskan penambahan tabel *last entry* untuk pencatatan sejarah cetak absensi karyawan.

Untuk mengatasi admin yang bisa menghapus *user ID* karyawan yang sudah dibuatkan, tetapi dengan kondisi data karyawan belum pernah dilakukan pengisian sama sekali, maka penulis menambahkan field *status_update* ke tabel 3.3 untuk mengetahui apakah data karyawan sudah pernah diisi atau belum.



Gambar 3.36 Usecase Diagram *Maintain Password*

Gambar 3.36 ini menjelaskan fungsi untuk mengganti password dan lupa password di web untuk karyawan Admin dan *User*.

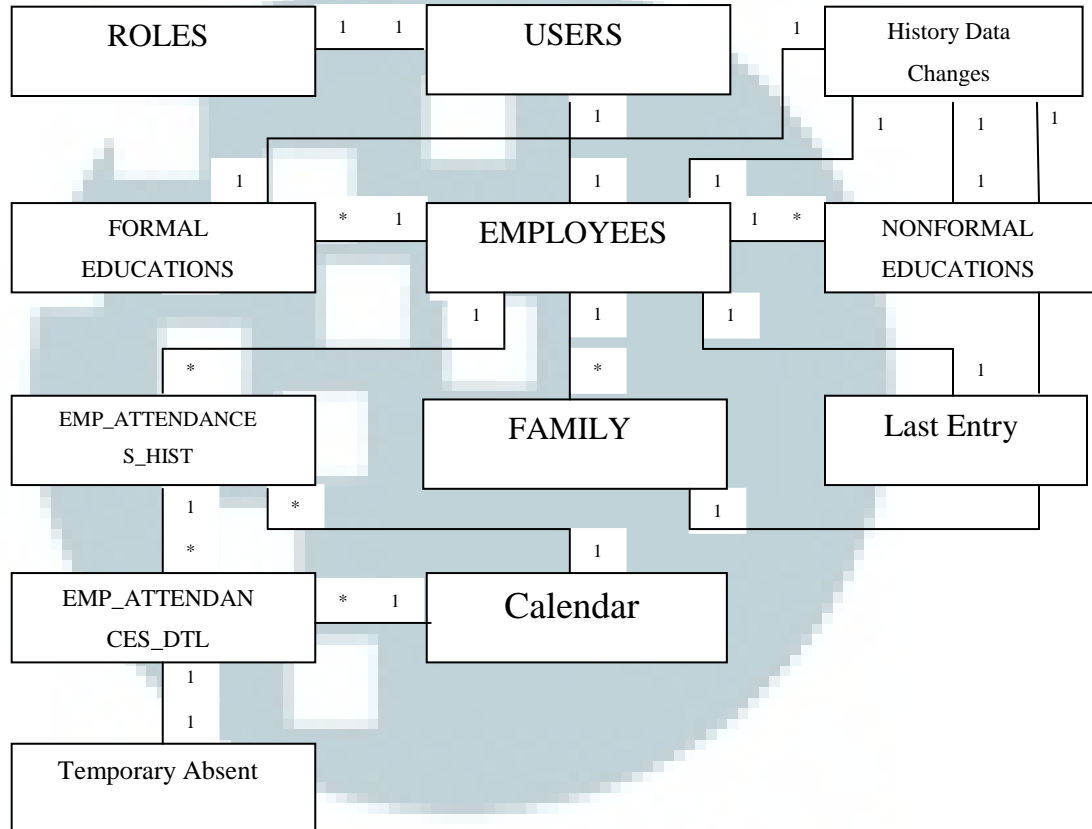
Perubahan tanggal kerja karyawan akan mempengaruhi *template* absensi karyawan, maka dari itu penulis membutuhkan tabel tampilan sementara yang terdiri atas tanggal absensi, kehadirannya, alasan jikalau tidak hadir, dan alasan lainnya si karyawan yang menampung sementara data absensi karyawan pada kondisi data absensi belum komplit.

Atribut	Tipe Data	Keterangan
Date_attendances	Datetime	Tanggal, bulan dan tahun
Emp_attendances	Varchar(3)	Kehadiran karyawan
Emp_expl	Varchar(7)	Alasan tidak hadir(sakit,ijin,libur,cuti atau lainnya)
Other	Varchar(50)	Alasan lain jikalau tidak hadir

Tabel 3.13 Temporary Absent

#TEMPORARY_ABSENT

(NIK (FK), date_attendances, emp_attendances, emp_expl, other)



Gambar 3.37 ERD Revisi

Terdapat penambahan tabel *temporary absent* untuk menimpa absensi yang sudah dilakukan karyawan ke *template* absensi barunya. Maka dari itu hubungan kardinalitasnya adalah *one-to-one* dimana satu rekor di tabel tampungan absen itu digunakan untuk menimpa 1 rekor di tabel absensi detail.

➤ Membangun Sistem

Tanggal masuk kerja di tampilan *maintain* data karyawan sisi admin bisa dirubah, dimana jika perubahannya beda bulan dan tahun dari tanggal kerja sebelumnya akan merubah NIK dan *template* absensi karyawan. Untuk itu penulis menyelesaikan masalah ini dengan menampung absensi yang status nya belum komplit, lalu *regenerate* absensi sesuai dengan tanggal masuk kerja baru si karyawan, dan menimpa tampilan absensi tadi ke *template* absensi yang baru.

Penambahan captcha di login



Gambar 3.38 Login dengan Captcha

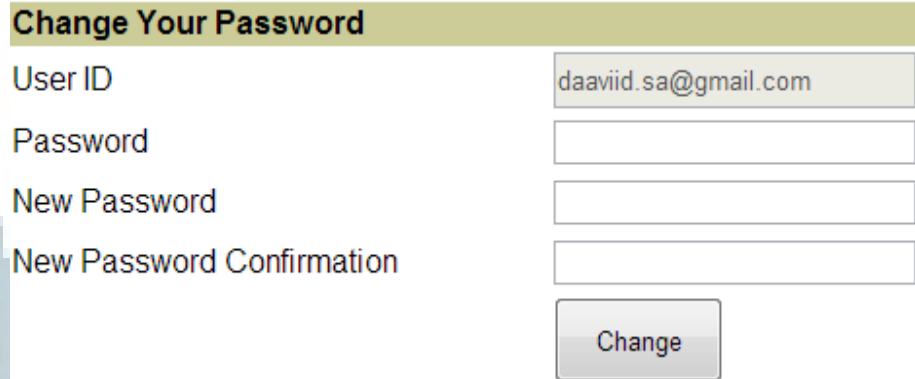
Menambahkan fungsi langsung ke bulan sebelumnya saat melakukan absensi.

[Previous Month](#)

Date	Attendance	Explanation
01 Nov 2013	<input type="radio"/> yes <input checked="" type="radio"/> no	<input type="text"/>
02 Nov 2013	<input type="radio"/> yes <input checked="" type="radio"/> no	sakit <input type="text"/>

Gambar 3.39 Pengisian Absen Dengan Fungsi Bulan Sebelumnya

Menambahkan tampilan ubah *password* untuk karyawan.



Change Your Password

User ID

Password

New Password

New Password Confirmation

Gambar 3.40 Penggantian *Password* Karyawan

Menambahkan tampilan untuk lupa *password* di awal tampilan *login*, jadi ada *link* menuju ke halaman lupa *password*. Di halaman ini terdiri atas penginputan *user ID* dan jika *user ID* itu memang ada, maka akan dikirimkan *password* baru melalui *email*.



accelist^Å

[Back to the Login Page](#)

Insert your User ID (Email)

Gambar 3.41 Lupa *Password*

Terdapat penambahan fungsi untuk admin dalam hal *maintain user ID*. Jadi admin bisa menghapus *user ID* yang belum pernah melakukan perubahan atau pengisian data karyawan.

NIK	Full Name	Updated
05201301	David Tan	no

Gambar 3.42 *Maintain User ID*

➤ Tes dan Evaluasi

Tes

Penambahan skenario tes perubahan tanggal kerja karyawan apakah mempengaruhi NIK atau tidak di tabel 3.7. Hasil dari tes ini adalah perubahan tanggal masuk kerja karyawan akan merubah NIK karyawan juga.

Revisi

- i. Apabila karyawan sudah pernah melakukan perubahan dan pengisian data karyawannya maka *button delete user ID karyawan* dari sisi admin tidak bisa dipilih.
- ii. Penambahan *captcha* di halaman lupa *password*.
- iii. Penambahan fungsi langsung ke bulan selanjutnya pada saat melakukan pengisian absen di bulan tertentu.

7. Pertemuan Ketujuh

➤ Memahami Kebutuhan

Pada saat kondisi data karyawan sudah pernah diisi atau dirubah, maka *button delete* di halaman admin untuk penghapusan *user ID* tidak bisa diklik.

Menambahkan captcha untuk mencegah *bots* yang lupa *password* di web untuk membuat *server down*.

Menambahkan fungsi untuk ke bulan selanjutnya pada saat karyawan melakukan absensi, sehingga memudahkan dan mempercepat dalam pengisian absen.

➤ Desain Sistem

-

➤ Membangun Sistem

Halaman admin terkait *maintain user ID* karyawan, apabila *user ID* itu sudah melakukan perubahan data karyawan maka status update menjadi sudah pernah, dan tidak bisa menghapus *user ID* nya. Sedangkan kebalikannya, apabila belum pernah melakukan perubahan data maka *user ID* tersebut masih bisa dihapus oleh Admin.

Updated?	Delete
no	Delete
yes	Delete

Gambar 3.43 Disable Button Delete user ID



Menambahkan captcha di halaman lupa password.



Gambar 3.44 Lupa Password dengan Captcha

Menambahkan fungsi langsung ke bulan berikutnya di halaman pengisian absen admin dan user.

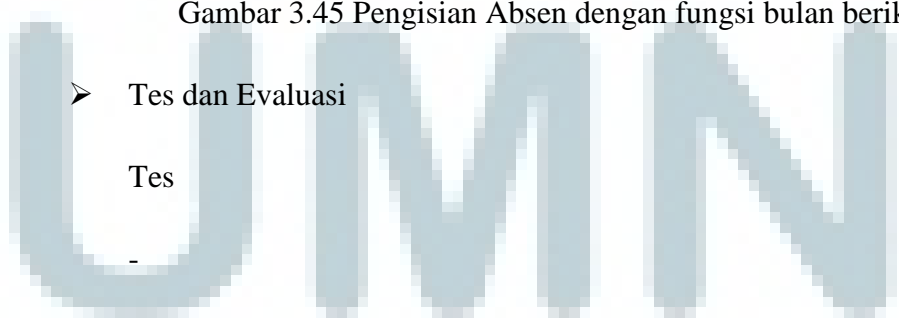
[Previous Month](#) [Next Month](#)

Date	Attendance	Explanation
01 Oct 2013	<input type="radio"/> yes <input checked="" type="radio"/> no	<input type="text" value="libur"/> ▾
02 Oct 2013	<input checked="" type="radio"/> yes <input type="radio"/> no	

Gambar 3.45 Pengisian Absen dengan fungsi bulan berikutnya

➤ Tes dan Evaluasi

Tes



Revisi

- i. Penambahan pilihan cuti pada saat karyawan memilih tidak hadir ketika melakukan absensi.
- ii. Penambahan fitur penghitungan total hadir karyawan setiap bulannya.
- iii. Admin ditambahkan fungsi untuk meng-*generate* absensi karyawan tahun berikutnya.
- iv. Penambahan tampilan untuk superadmin melihat data total kehadiran karyawan berdasarkan bulan dan tahun.

8. Pertemuan Kedelapan

➤ Memahami Kebutuhan

Apabila karyawan memilih cuti pada saat melakukan absensi, maka akan ada kotak teks untuk pengisian alasan karyawan ini cuti

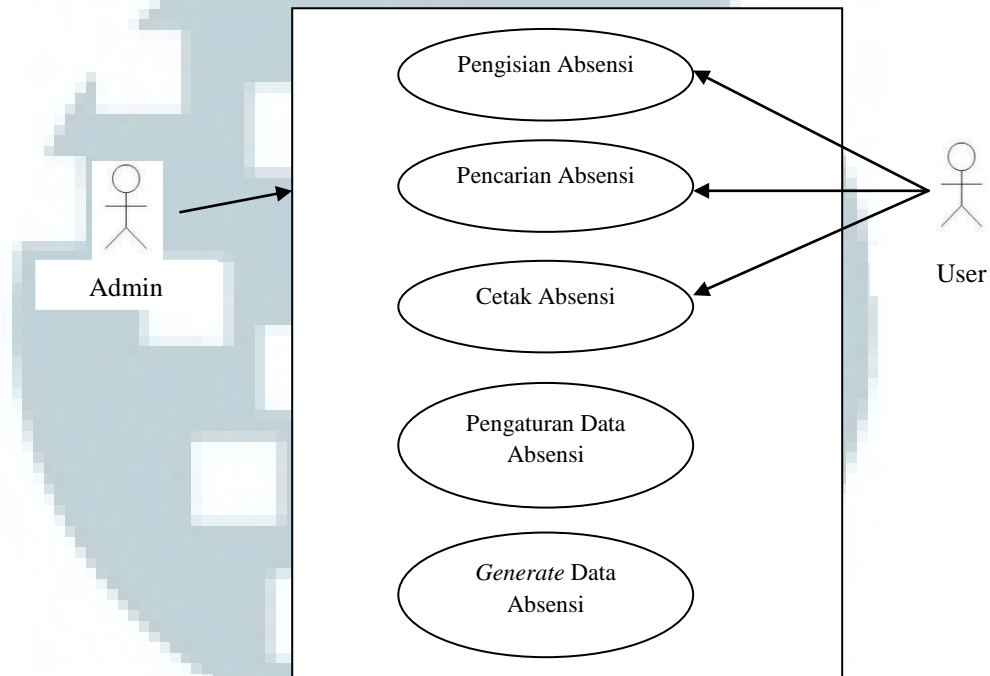
Karyawan bisa melihat total kehadiran setiap bulannya pada saat melakukan absensi.

Admin terdapat tampilan untuk *generate* absensi tahun berikutnya para karyawan. Ini dilakukan karena absensi karyawan yang tadi di-*generate* otomatis ketika karyawan dibuatkan *user ID* itu berakhir di akhir bulan tahun ini, dan untuk absensi tahun depan maka admin harus men-*generate* absensi tahun berikutnya untuk semua karyawan.

Pemilik perusahaan butuh untuk melihat total kehadiran setiap bulannya semua karyawannya untuk dikalikan dengan uang makan.

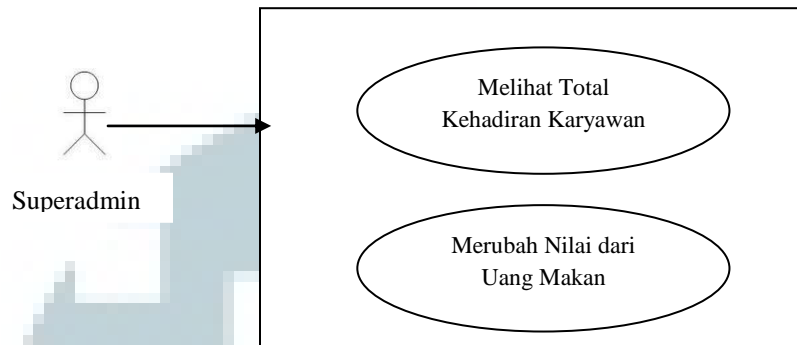
➤ Desain Sistem

Penambahan fungsi *generate* absensi per tahun oleh admin perusahaan.



Gambar 3.46 *Usecase Diagram* Administrasi Data Absensi Karyawan

Gambar di atas menjelaskan terdapat tambahan *use-case generate* data absensi untuk tingkatan pengguna admin. Fungsi dari *use-case* ini untuk meng-*generate* absensi karyawan tahun berikutnya.



Gambar 3.47 *Usecase Diagram* Superadmin Melihat Total Kehadiran dan Perubahan Nilai Uang Makan.

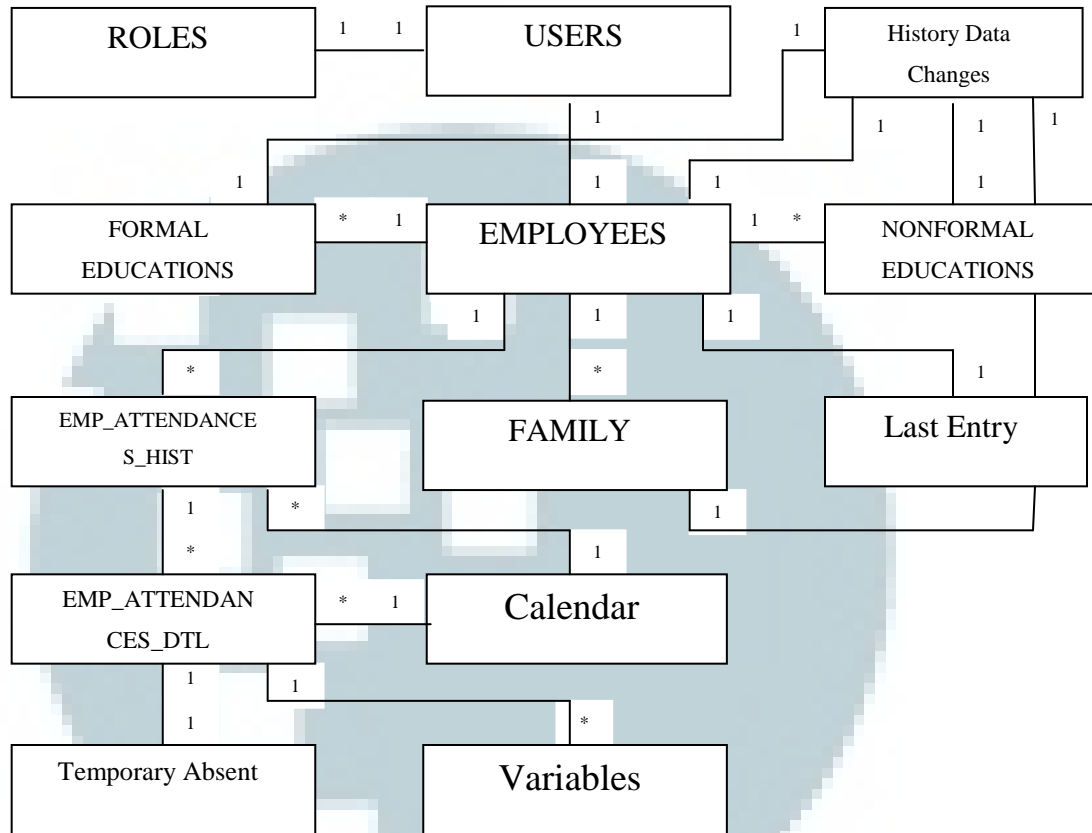
Gambar di atas menjelaskan terdapat tampilan baru untuk tingkatan pengguna superadmin atau pemilik untuk melihat total kehadiran karyawan per bulan dan tahun yang diisikan. Total kehadiran ini digunakan untuk dikalikan dengan uang makan, dimana nilai dari uang makan ini tetap dan bisa diubah oleh pemilik. Oleh karena itu dibutuhkan data uang makan di database untuk dikalikan dengan total kehadiran karyawan, dan ditambahkan halaman baru untuk pemilik merubah nilai dari uang makan tersebut.

Atribut	Tipe Data	Keterangan
Jenis_field	Varchar(30)	Jenis variabelnya, misalnya uang makan, uang transport dsb.
Nilai	Int	Nilai dari jenis field tersebut

Tabel 3.14 *Variables*

#VARIABLES

(jenis_field, nilai)



Gambar 3.48 ERD Revisi

Tambahan tabel *variables* di *ERD* untuk relasi dengan absensi detail. Jadi setiap total hadir per bulan dari tabel absensi detail bisa dikalikan dengan satu atau lebih variabel dari tabel *variables*, seperti uang makan, uang jalan dsb.

➤ **Membangun Sistem**

Menambahkan pilihan cuti ke halaman absensi admin dan user, dan jika cuti dipilih maka ada kotak teks untuk pengisian detail cutinya.

Explanation	Other
cuti	
cuti	
libur	
sakit	
ijin	
cuti	
lainnya	

Gambar 3.49 Pemilihan Cuti pada saat Absensi

Menambahkan fungsi penghitungan total hadir data absensi pada halaman pengisian data absensi per bulannya admin dan user.

[Previous Month](#) Next Month Total Attendance = 8

Date	Attendance
01 Nov 2013	<input type="radio"/> yes <input checked="" type="radio"/> no
02 Nov 2013	<input type="radio"/> yes <input checked="" type="radio"/> no

Gambar 3.50 Total Hadir Karyawan



Menambahkan tampilan untuk admin *generate* absensi karyawan tahun berikutnya.

YEARS	Generate
2013	<input type="button" value="Generate"/>
2014	<input type="button" value="Generate"/>

Gambar 3.51 Generate Absensi

Menambahkan halaman baru untuk superadmin melihat total kehadiran per bulan dan tahun setiap karyawannya, dan halaman untuk perubahan nilai uang makan.

Search By Month and Year: Welcome, **Matius Kelvin**

NIK	Name	Role Name	Month	Year	Total Attendances	Uang Makan	Total
05201301	David Tan	hrd	AUGUST	2013	31	50000	1550000

Gambar 3.52 Superadmin melihat total kehadiran karyawan

Description	Value	Edit
uang makan	<input type="text" value="50000"/>	<input type="button" value="Update"/> <input type="button" value="Cancel"/>

Gambar 3.53 Superadmin Merubah Nilai dari Uang Makan

➤ Tes dan Evaluasi

Tes

Penambahan skenario membuat *user ID* baru lagi setelah karyawan admin men-*generate* absensi tahun berikutnya untuk para karyawan di tabel 3.7. Hasil dari tes ini adalah absensi karyawan baru akan mengikuti periode absensi karyawan lainnya yang sudah ter-*generate* sampai tahun berikutnya.

Revisi

- i. Untuk fungsi langsung ke bulan berikutnya pada saat karyawan melakukan absensi itu berakhir di bulan sekarang.
- ii. Untuk fungsi dari admin dan user dalam melakukan pencarian absensi berdasarkan bulan dan tahun hanya sampai pada absensi bulan ini, bulan berikutnya belum bisa dibuka.

9. Pertemuan Kesembilan

➤ Memahami Kebutuhan

Karyawan dicegah untuk melakukan absensi pada bulan setelah bulan sekarang.

➤ Desain Sistem

➤ Membangun Sistem

Button *next month* akan tidak aktif ketika sudah sampai bulan sekarang di halaman pengisian absensi karyawan. Misalnya bulan ini adalah November maka tampilan akan seperti berikut.

[Previous Month](#) Next Month Total Attendance = 8

Date	Attendance
01 Nov 2013	<input type="radio"/> yes <input checked="" type="radio"/> no
02 Nov 2013	<input type="radio"/> yes <input checked="" type="radio"/> no

Gambar 3.54 Next Month Tidak Aktif

From : 2013/05 To : 2013/12 Search

Months	Year	Status	Detail
MAY	2013	complete	Detail
JUNE	2013	complete	Detail
JULY	2013	complete	Detail
AUGUST	2013	complete	Detail
SEPTEMBER	2013	complete	Detail
OCTOBER	2013	complete	Detail
NOVEMBER	2013		Detail
DECEMBER	2013		Detail

Gambar 3.55 Detil data absensi

Pada saat ingin klik detail data absensi Desember akan ada *alert* pemberitahuan bahwa absensi belum bisa diakses sekarang.

From : 2013/05 To : 2013/12 Search

Months	Year	Status	Detail
MAY	2013	complete	Detail
JUNE		complete	Detail
JULY		complete	Detail
AUGUST		complete	Detail
SEPTEMBER		complete	Detail
OCTOBER		complete	Detail
NOVEMBER	2013		Detail
DECEMBER	2013		Detail

Message

Absent Cannot be Opened Now

OK

Gambar 3.56 Notifikasi Pengaksesas Halaman Absen Melebihi Bulan dan Tahun Sekarang

➤ Tes dan Evaluasi

Sistem sudah sesuai keinginan pemilik perusahaan.

3.3.4. Hasil Akhir

Hasil akhir didapatkan penulis secara tahap demi tahap sesuai dengan tahapan SDLC yang penulis gunakan. Hasil desain sistem berupa *usecase diagram*, *table of attributes* dan ERD, serta hasil tes dari sistem terus berkembang sampai kondisi desain sistem sudah *final* dari pertemuan pertama sampai pertemuan ke Sembilan (Lihat lampiran A, B, C, dan D untuk hasil akhir). Sehingga pada pertemuan ke sembilan sistem sudah sesuai keinginan pemilik perusahaan dan pemilik perusahaan melakukan skenario tes terakhir dalam bentuk UAT (Lihat lampiran E) terkait pengesahaan bahwa sistem sudah benar-benar siap untuk dilakukan perencanaan implementasi pada akhir bulan Desember.

3.3.5. Kendala

Untuk kendala yang dialami penulis selama kerja magang adalah keterbatasan pengetahuan yang didapatkan di bangku perkuliahan demi menyelesaikan masalah-masalah yang ditemui selama kerja magang. Seperti halnya penggunaan database *MS SQL SERVER Express Edition*, bagaimana cara koneksi dari aplikasi ke *database*, penggunaan bahasa *Transact SQL* untuk pembuatan *trigger* dan *procedure* demi memenuhi kebutuhan proses dari sistem yang dibuatkan. Kendala lain yang ditemukan penulis adalah pada saat pembuatan kotak teks untuk pengisian tanggal lahir atau tanggal masuk kerja, dimana tidak ada objek dari visual studio untuk pembuatan *date time picker*. Selain itu juga penulis mendapatkan kendala untuk bahasa pemrograman web khususnya penggunaan *javascript*, dimana *javascript* ini dipakai oleh penulis untuk membuat web itu interaktif dari sisi *client*, jadi tanpa harus *load* ke *server* terlebih dahulu untuk melakukan sebuah proses.

3.3.6. Solusi

Berdasarkan kendala-kendala yang penulis temui selama proses kerja magang maka solusi yang penulis lakukan adalah menanyakan kepada pembimbing lapangan terkait masalah-masalah yang penulis temukan. Lalu pembimbing lapangan memberikan masukan saran terkait keterbatasan pengetahuan untuk penyelesaian masalah tersebut seperti penggunaan *trigger*, *procedure*, *javascript*, dan untuk masalah *date-time picker* penulis diberitahukan mengenai *ajax tool kit* agar terdapat objek baru di *visual studio* untuk *date-time picker*. Setelah mendapatkan masukan untuk penyelesaian masalah tersebut penulis melakukan eksplorasi sendiri untuk cara penggunaannya.

UMMN