

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Analisis dan perancangan sistem pada aplikasi presensi kegiatan dilakukan dengan menggunakan metode *System Development Life Cycle* (SDLC). Tahap yang dilakukan adalah tahapan perencanaan, analisis dan desain.

3.1. Perencanaan

Pada tahapan perencanaan, identifikasi masalah dilakukan untuk mengetahui permasalahan yang terdapat pada organisasi lebih dalam. Identifikasi masalah dilakukan dengan cara sebagai berikut:

1. Wawancara

Wawancara dilakukan secara langsung kepada mahasiswa dan karyawan yang sering mengadakan kegiatan di lingkungan Stikom Surabaya.

2. Pengamatan

Pengamatan dilakukan secara langsung ketika suatu kegiatan dilaksanakan. Pengamatan yang dilakukan juga mencakup pencarian data yang akan digunakan oleh sistem.

3. Studi literatur

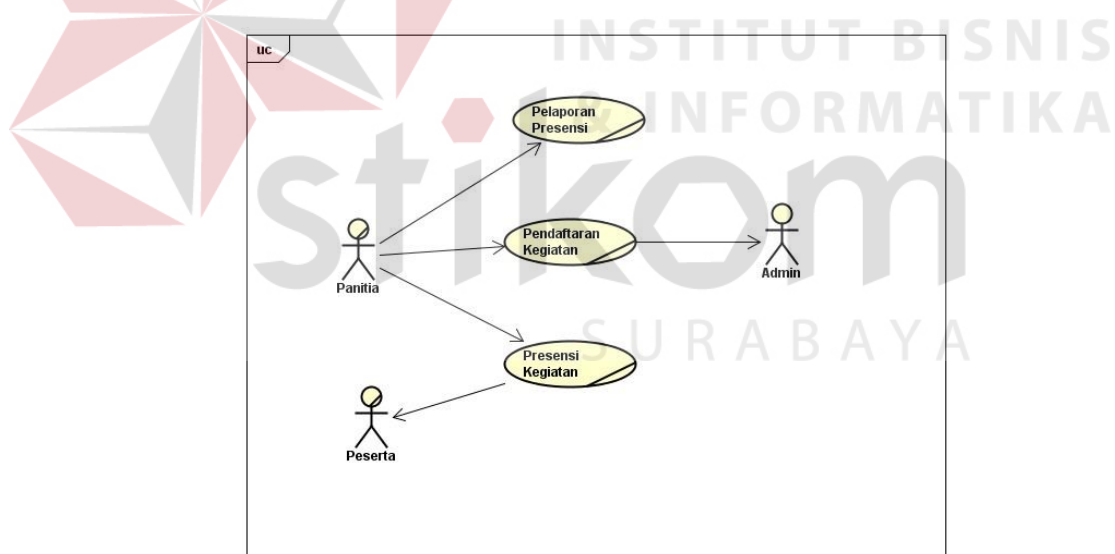
Studi literatur digunakan untuk mencari referensi informasi yang berhubungan dengan topik yang diusulkan. Studi literatur dilakukan dengan cara mengumpulkan buku, makalah, artikel atau karya tulis lain yang berhubungan dengan topik atau usulan yang diangkat.

3.2. Tahap Analisis

Analisis dan perancangan sistem dilakukan setelah melakukan identifikasi masalah. Langkah pertama yang dikerjakan adalah dengan membuat *Business Use Case* berdasarkan hasil pengamatan dan wawancara. Selanjutnya dibuat *Activity Diagram* berdasarkan *Business Use Case* yang telah dibuat. Pengembangan dari *Activity Diagram* adalah *System Use Case* yang menjelaskan lebih detail proses di dalam aplikasi. Dan kemudian dilanjutkan dengan membuat *Flow Of Event* untuk memberikan penjelasan lebih detil dari *System Use Case*.

3.2.1. Business Use Case

Gambar 3.1 adalah bentuk dari *Business Use Case* untuk aplikasi presensi kegiatan.



Gambar 3. 1 *Business Use Case* Presensi Kegiatan

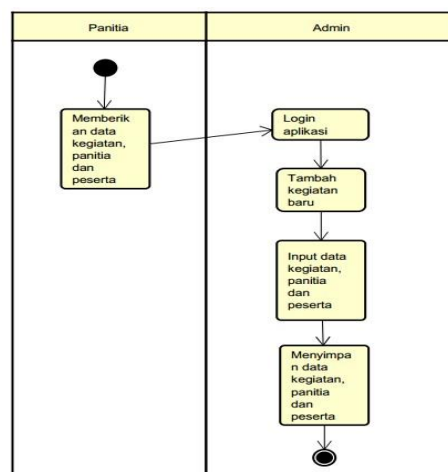
Pada Gambar 3.1 terdapat 2 aktor bisnis yaitu Panitia dan Peserta kegiatan serta yang menjadi pekerja bisnis adalah Admin dari aplikasi presensi

kegiatan. *Use case* yang terjadi adalah pendaftaran kegiatan, presensi kegiatan dan pelaporan presensi kegiatan.

Pada *use case* pendaftaran kegiatan, aktor bisnis Panitia melakukan pendaftaran kegiatan yang akan dilakukan kepada Admin. Maka kemudian Admin akan meminta data – data yang dibutuhkan seperti data panitia, data kegiatan serta data peserta untuk dimasukkan ke dalam aplikasi. *Use case* presensi kegiatan merupakan proses presensi kegiatan yang dilakukan setelah melakukan pendaftaran kegiatan. Presensi kegiatan dilakukan oleh Panitia dengan menggunakan perangkat Android yang menggunakan NFC untuk mengabsen peserta kegiatan yang hadir. Sedangkan pada *use case* pelaporan kegiatan, panitia melihat hasil presensi untuk mengetahui siapa saja dan berapa jumlah peserta kegiatan tersebut.

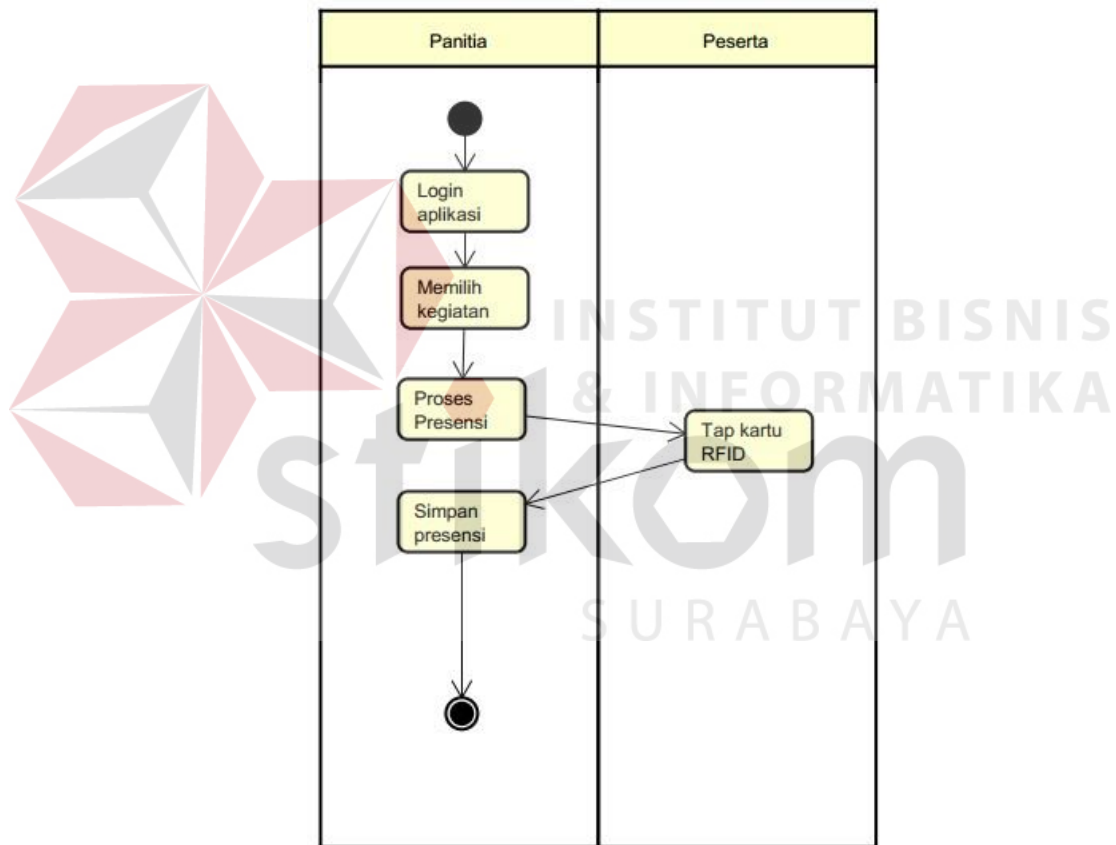
3.2.2. Activity Diagram

Activity Diagram atau diagram aktifitas menggambarkan lebih detil dari setiap *business use case*. Gambar 3.2, 3.3 dan 3.4 menyajikan penurunan *business use case* ke dalam diagram aktifitas.



Gambar 3. 2 Diagram Aktifitas Pendaftaran Kegiatan

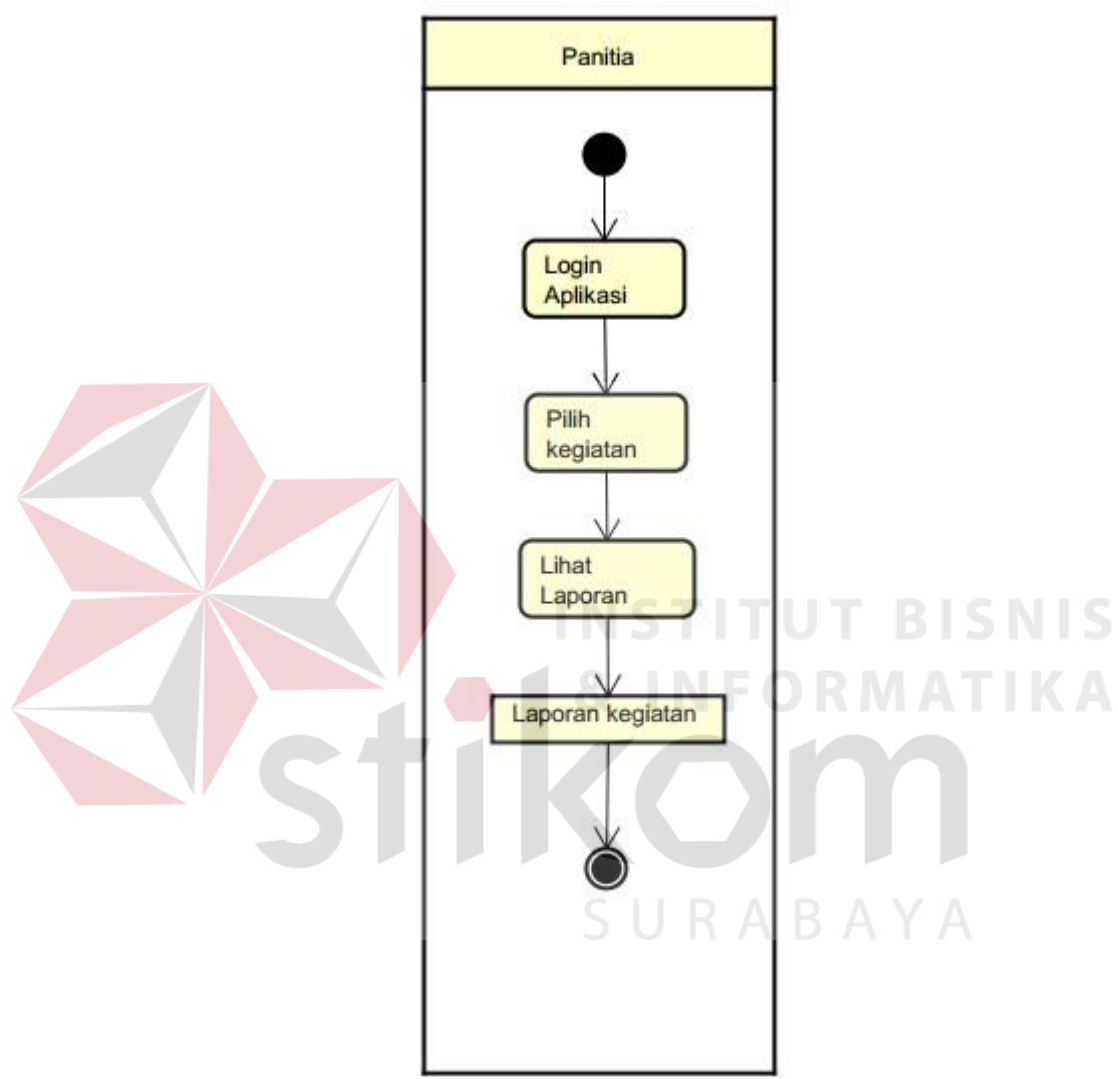
Pada Gambar 3.2 Diagram aktifitas pendaftaran kegiatan menjelaskan proses pendaftaran yang dilakukan oleh panitia kegiatan kepada admin aplikasi. Panitia memberikan data kegiatan yang berupa nama kegiatan, tempat kegiatan dilakukan, penyelenggara kegiatan, waktu kegiatan dimulai dan diakhiri, jenis peserta dan nama panitia kegiatan. Kemudian admin akan memasukkan data tersebut dan menyimpan ke dalam *database*. Proses berhenti setelah data kegiatan berhasil disimpan.



Gambar 3. 3 Diagram Aktifitas Presensi Kegiatan

Pada Gambar 3.3 Diagram aktifitas presensi kegiatan menunjukkan proses presensi kegiatan yang berlangsung. Panitia melakukan *login* terlebih dahulu ke dalam aplikasi. Kemudian panitia memilih kegiatan yang akan

dilakukan proses presensi. Setelah memilih kegiatan, panitia meminta peserta kegiatan untuk melakukan *tapping* kartu *RFID* ke perangkat pembaca *RFID*. Data kartu yang terbaca kemudian disimpan langsung ke dalam *database*.



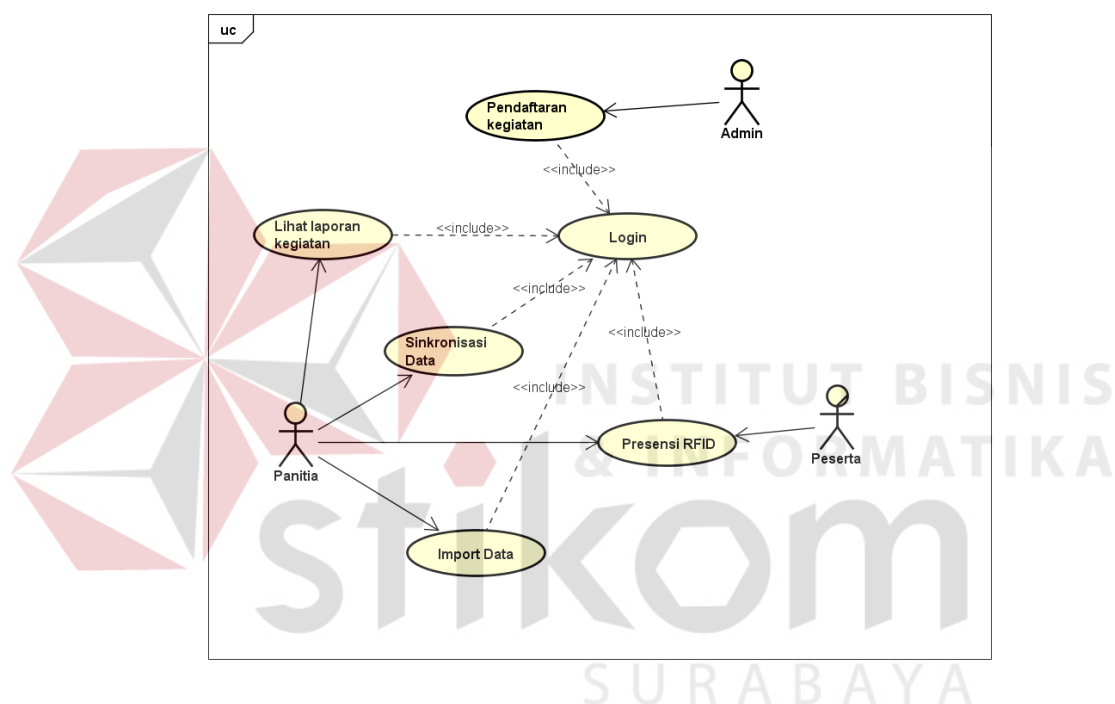
Gambar 3. 4 Diagram Aktivitas Pelaporan Kegiatan

Pada Gambar 3.4 Diagram aktivitas pelaporan kegiatan bertujuan untuk menjelaskan proses pelaporan hasil presensi kegiatan. Panitia terlebih dulu melakukan *login* ke dalam aplikasi. Selanjutnya panitia memilih kegiatan yang

akan dilihat laporan hasil presensinya dan kemudian mencetak hasil laporan tersebut.

3.2.3. System Use Case

System Use Case merupakan penurunan dari *Business Use Case* berdasarkan proses yang ada pada diagram aktifitas. Gambar 3.5 menunjukkan hasil penurunan *use case* bisnis ke dalam *use case* sistem.



Gambar 3. 5 *System Use Case* Aplikasi Presensi Kegiatan

Pada Gambar 3.5 *System Use Case Aplikasi Presensi Kegiatan* menjelaskan penurunan *use case* bisnis ke *use case* sistem menghasilkan *use case* baru yang menjadikan interaksi antar aktor dengan *use case* menjadi lebih jelas. Muncul beberapa *use case* tambahan yang didapat dari beberapa proses pada diagram aktifitas. *Use case Login* merupakan proses untuk mengenali pengguna aplikasi. Proses ini di-include oleh *use case* – *use case* lain karena proses *login* merupakan proses utama sebelum melakukan proses – proses yang lain. *Use case*

sinkronisasi data digunakan untuk melakukan sinkronisasi data antara aplikasi presensi *Android* dengan data yang berada pada *server*. *Use case import* data merupakan proses pengambilan data presensi *offline* dan di-merge dengan data pada presensi *online*.

3.2.4. Flow of Event

Flow of Event merupakan penjelasan dari proses – proses yang dilakukan dari *use case* yang terdapat pada system *use case*.

Tabel 3. 1 Tabel *Flow Of Event Login*

Nama Use Case	Login	
Tujuan	Untuk mengenali pengguna yang berhak menggunakan aplikasi.	
Prasyarat	Tidak ada	
Kondisi akhir sukses	Pengguna dikenali dan diarahkan ke halaman utama.	
Kondisi akhir gagal	Pengguna tidak dikenali maka aplikasi menampilkan pesan kesalahan.	
Aktor utama	Admin dan panitia	
Aktor sekunder	Tidak ada	
Pemicu	Pengguna membuka aplikasi	
Alur utama	Langkah	Aksi Utama
	1	Pengguna menginputkan <i>username</i> dan kata sandi
	2	Pengguna melakukan submit inputan
	3	Aplikasi melakukan pengecekan data

	4	Aplikasi mengarahkan pengguna ke halaman utama
	5	Use Case berakhir
Alur alternatif	Langkah	Aksi Alternatif
	4.1	Aplikasi menampilkan pesan error ketidaksesuaian isian, kembali ke langkah 1.

Tabel 3. 2 Tabel *Flow Of Event* Tambah Data Kegiatan

Nama Use Case	Tambah data kegiatan	
Tujuan	Untuk menambah data kegiatan yang ingin didaftarkan panitia	
Prasyarat	Login	
Kondisi akhir sukses	Data kegiatan berhasil ditambahkan	
Kondisi akhir gagal	Data kegiatan sudah ada di database	
Aktor utama	Admin	
Aktor sekunder	Panitia	
Pemicu	Admin memilih menu tambah kegiatan	
Alur utama		Aksi Utama
	1	Admin memasukkan data kegiatan, data panitia dan data peserta.
	2	Admin melakukan submit data yang diinputkan.
	3	Aplikasi mengecek data kegiatan yang diinputkan dengan data yang ada di database.
	4	Aplikasi menyimpan data kegiatan, data panitian dan data peserta.
	5	Aplikasi menampilkan pesan bahwa data telah disimpan.
	6	Use Case berakhir.

Alur alternatif	Langkah	Aksi Alternatif
	3.1	Aplikasi menampilkan pesan bahwa telah ada data kegiatan pada database yang hampir sama dengan data kegiatan yang diinputkan, kembali ke tahap 1.

Tabel 3. 3 Tabel *Flow Of Event* Lihat Laporan Kegiatan

Nama Use Case	Lihat laporan kegiatan	
Tujuan	Untuk melihat laporan hasil presensi kegiatan	
Prasyarat	Login	
Kondisi akhir sukses	Data kegiatan berhasil ditampilkan	
Kondisi akhir gagal	Data kegiatan tidak ditemukan	
Aktor utama	Panitia	
Aktor sekunder		
Pemicu	Panitia memilih menu lihat laporan kegiatan	
Alur utama	Langkah	Aksi Utama
	1	Panitia memilih kegiatan yang ingin dilihat laporan presensinya.
	2	Aplikasi mengecek data kegiatan sesuai yang dipilih panitia
	3	Aplikasi menampilkan laporan presensi kegiatan.
	4	Use Case berakhir.

Tabel 3. 4 Tabel *Flow Of Event* Presensi Kegiatan

Nama Use Case	Presensi Kegiatan
Tujuan	Untuk melakukan presensi kegiatan menggunakan NFC
Prasyarat	Login
Kondisi akhir sukses	Data presensi berhasil disimpan
Kondisi akhir gagal	Kartu RFID tidak dikenali

Aktor utama	Panitia	
Aktor sekunder	Peserta Kegiatan	
Pemicu	Panitia memilih menu presensi kegiatan	
Alur utama	Langkah	Aksi Utama
	1	Panitia memilih kegiatan.
	2	Aplikasi menampilkan data kegiatan berdasarkan pilihan panitia.
	3	Aplikasi <i>stand by</i> menunggu <i>tapping</i> kartu RFID dari peserta.
	4	Peserta melakukan <i>tapping</i> kartu RFID pada NFC.
	5	Aplikasi menerima nomor <i>tag</i> kartu.
	6	Panitia menyimpan seluruh data presensi peserta.
	7	Use Case berakhir.
Alur alternatif	Langkah	Aksi Alternatif
	5.1	Aplikasi menampilkan pesan kartu tidak dikenali, kembali ke tahap 4.

Tabel 3. 5 Tabel *Flow Of Event* Sinkronisasi Data

Nama Use Case	Sinkronisasi Data	
Tujuan	Untuk melakukan sinkronisasi data presensi	
Prasyarat	Login	
Kondisi akhir sukses	Data berhasil disinkronisasi	
Kondisi akhir gagal	Sinkronisasi data gagal	
Aktor utama	Panitia	
Aktor sekunder	Tidak ada	
Pemicu	Panitia memilih menu sinkronisasi data	
Alur utama	Langkah	Aksi Utama

	1	Panitia memilih kegiatan yang ingin dilakukan sinkronisasi data.
	2	Aplikasi melakukan sinkronisasi data pada kegiatan yang telah dipilih panitia
	3	Aplikasi menampilkan sinkronisasi data berhasil
	4	Use Case berakhir.
Alur alternatif	Langkah	Aksi Alternatif
	2.1	Aplikasi menampilkan pesan sinkronisasi gagal karena tidak ada sambungan ke server, kembali ke tahap 1.
	2.2	Aplikasi menampilkan pesan tidak ada data yang perlu dilakukan sinkronisasi, lompat ke tahap 4

3.3. Tahap Desain

Tahap desain menjelaskan secara spesifik bagaimana sistem atau aplikasi dibangun nantinya. Dalam tahap desain, dibuat *Sequence Diagram* untuk menggambarkan pesan yang dikirim antar komponen di dalam aplikasi. Setelah *Sequence Diagram*, maka *Collaboration Diagram* dibuat untuk menunjukkan hubungan antar komponen di dalam aplikasi. Dari hubungan tersebut, selanjutnya dibuat *Class Diagram* untuk mengetahui kelas – kelas apa saja yang diperlukan oleh aplikasi. *Component Diagram* menunjukkan kelas atau komponen mana yang perlu dilakukan kompilasi terlebih dulu. Dan *Deployment Diagram* menggambarkan bagaimana aplikasi akan diimplementasikan nantinya.

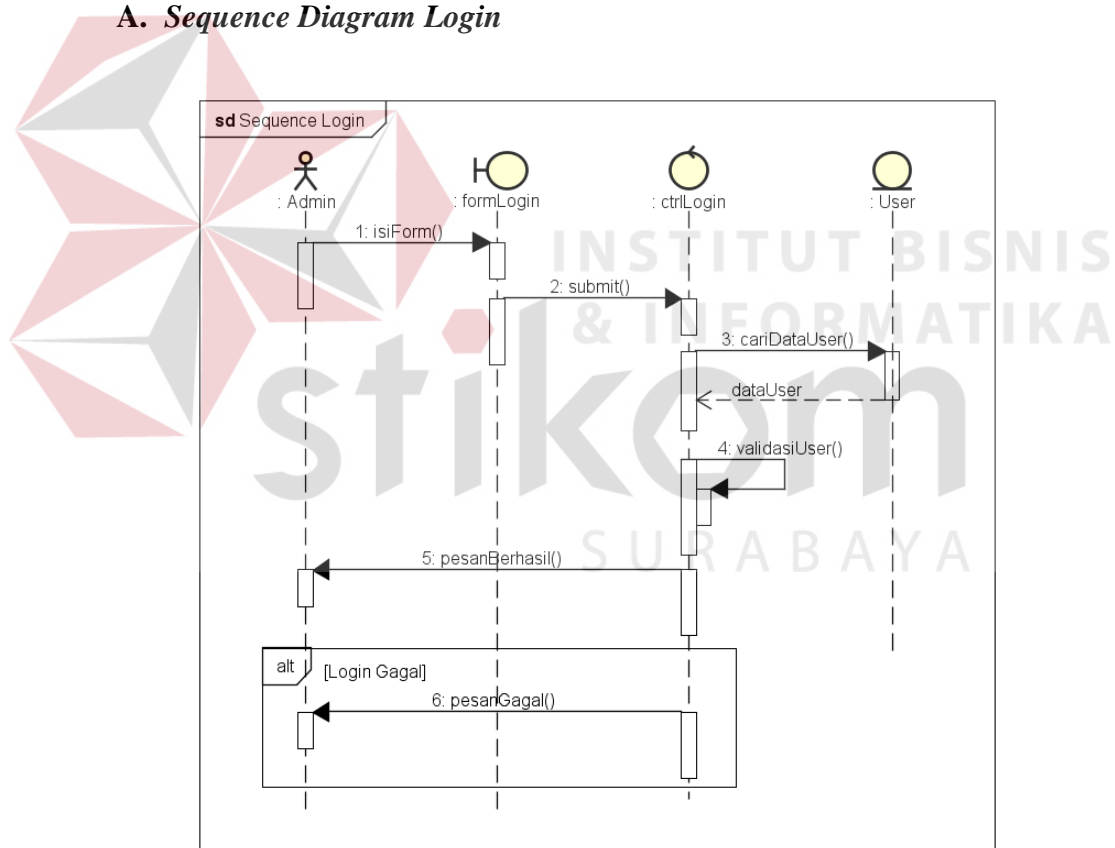
Conceptual Data Model dan *Physical Data Model* merupakan diagram yang menunjukkan hubungan antar entitas dan tabel – tabel yang diperlukan di dalam

database. Sedangkan desain input dan output menunjukkan rancangan *user interface* dari aplikasi yang akan dibuat.

3.3.1. Sequence Diagram

Sequence Diagram menggambarkan pesan yang dikirim antar komponen dengan aktor yang memulainya. *Sequence Diagram* dibuat berdasarkan *System Use Case* yang telah dibuat sebelumnya.

A. Sequence Diagram Login



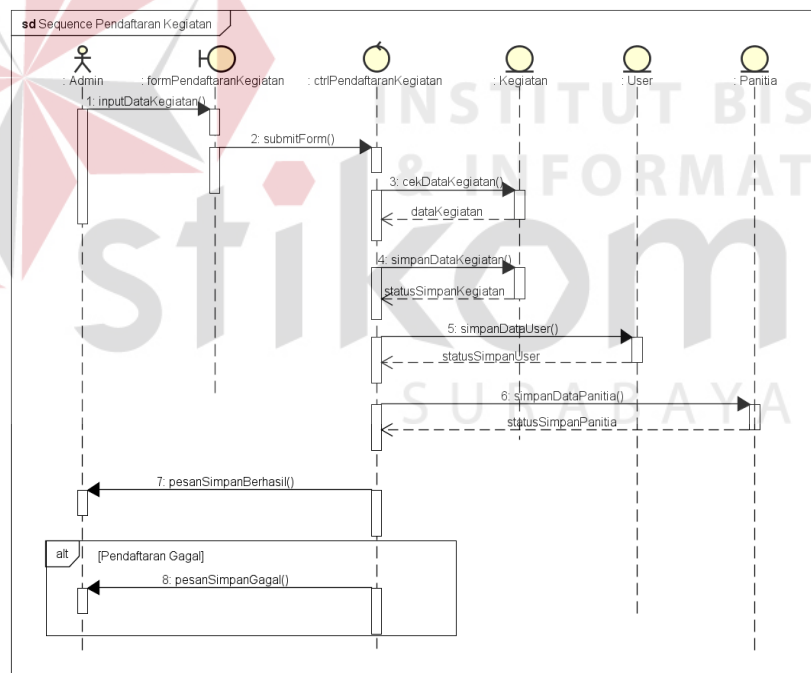
Gambar 3. 6 *Sequence Diagram Login*

Pada Gambar 3.6 *Sequence Diagram Login* menjelaskan proses login dimulai dari admin atau panitia dengan mengisi *username* dan *password* pada form

login. Setelah mengisi form *login*, admin atau panitia melakukan *submit form* yang membuat *form login* mengirimkan pesan kepada *controller* untuk dilakukan validasi. *Controller* kemudian mengirimkan pesan kepada entitas *User* untuk mencari data admin atau panitia pada *database*.

Jika data ditemukan, maka *controller* akan memberikan pesan bahwa *login* berhasil dan mengarahkan admin atau panitia ke halaman selanjutnya. Jika data tidak ditemukan, maka *controller* akan memberikan pesan bahwa *login* gagal dan meminta admin atau panitia untuk mengisi kembali *username* dan *password* yang benar.

B. Sequence Diagram Pendaftaran Kegiatan



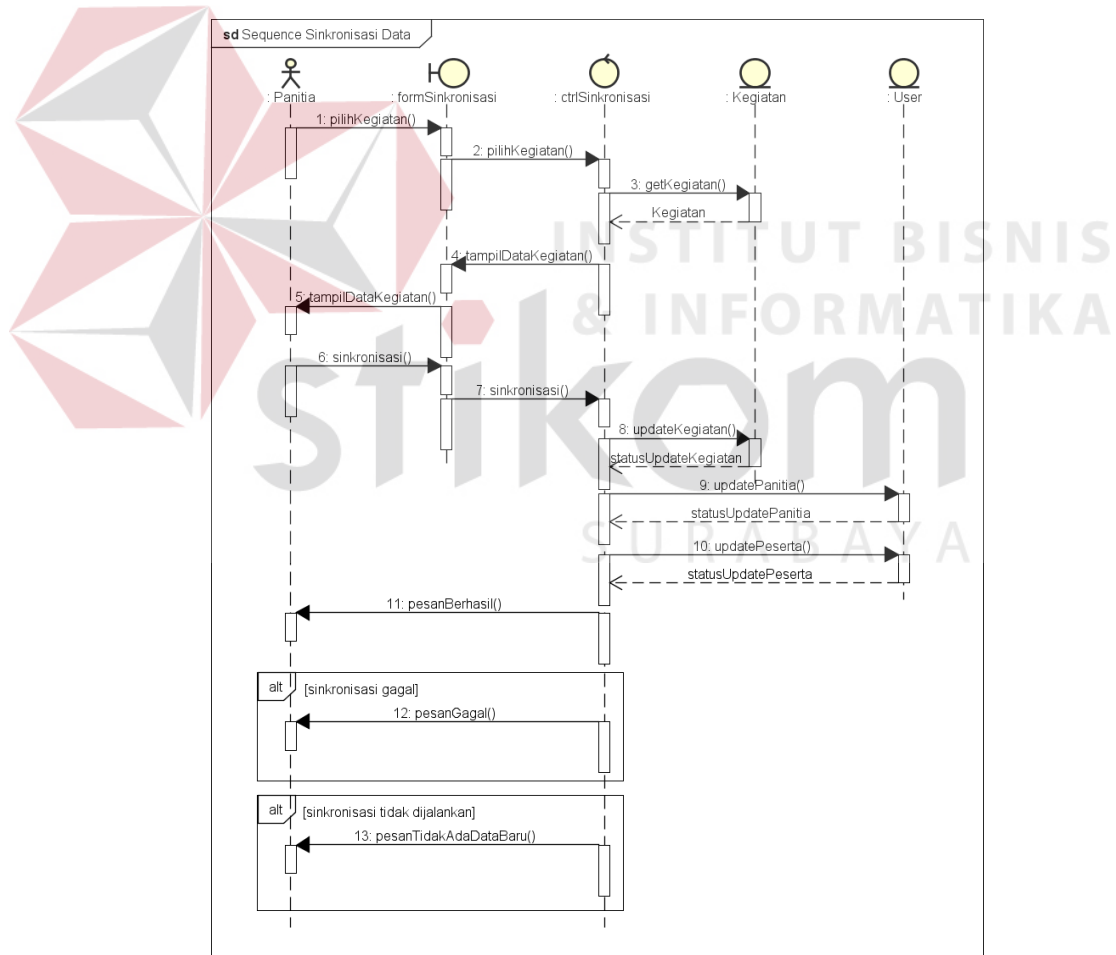
Gambar 3. 7 Sequence Diagram Pendaftaran Kegiatan

Pada Gambar 3.7 Sequence Diagram Pendaftaran Kegiatan menjelaskan proses pendaftaran kegiatan dimulai dari admin mengisi data kegiatan dan kemudian melakukan *submit form*. Data yang di-*submit* diterima oleh *controller*.

Controller akan mengecek apabila terdapat kesesuaian antara data yang diinputkan dengan data yang terdapat pada *database*. Bila tidak ditemukan data yang sama, selanjutnya *controller* akan menyimpan data kegiatan, data panitia dan data admin ke dalam *database*.

Jika proses penyimpanan data berhasil, *controller* akan memberikan pesan berhasil kepada admin. Dan jika proses penyimpanan gagal, maka *controller* akan memberikan pesan kesalahan kepada admin.

C. Sequence Diagram Sinkronisasi Data

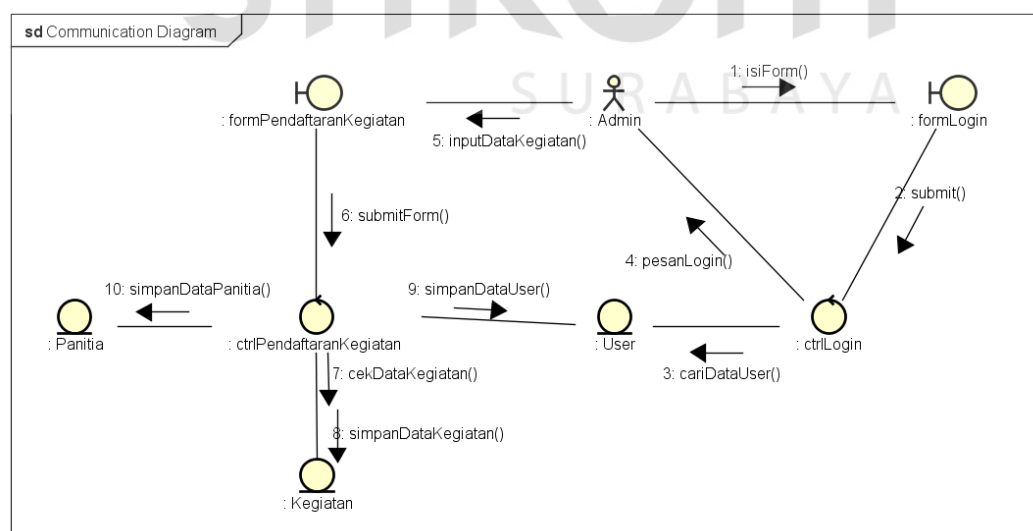


Gambar 3. 8 Sequence Diagram Sinkronisasi Data

Pada Gambar 3.8 *Sequence Diagram* Sinkronisasi Data menjelaskan proses sinkronisasi data yang dilakukan oleh panitia. Sinkronisasi data merupakan proses melakukan penyamaan data yang berada di server dengan data yang berada di perangkat lokal. Penyamaan data perlu dilakukan jika saat proses presensi, perangkat lokal tidak dapat terhubung ke server.

Panitia memilih kegiatan yang akan dilakukan sinkronisasi data di *form* sinkronisasi. Setelah dipilih, *controller* akan menampilkan data kegiatan yang dipilih dan menampilkannya ke *form* sinkronisasi. Kemudian panitia akan mengirim pesan untuk dilakukan sinkronisasi kepada *form* sinkronisasi yang selanjutnya mengirim pesan sinkronisasi kepada *controller*. *Controller* akan membandingkan data yang berada pada perangkat lokal dengan data pada server. Jika terdapat data baru, maka *controller* akan menambah data baru tersebut ke server.

3.3.2. Collaboration Diagram



Gambar 3. 9 Diagram Kolaborasi

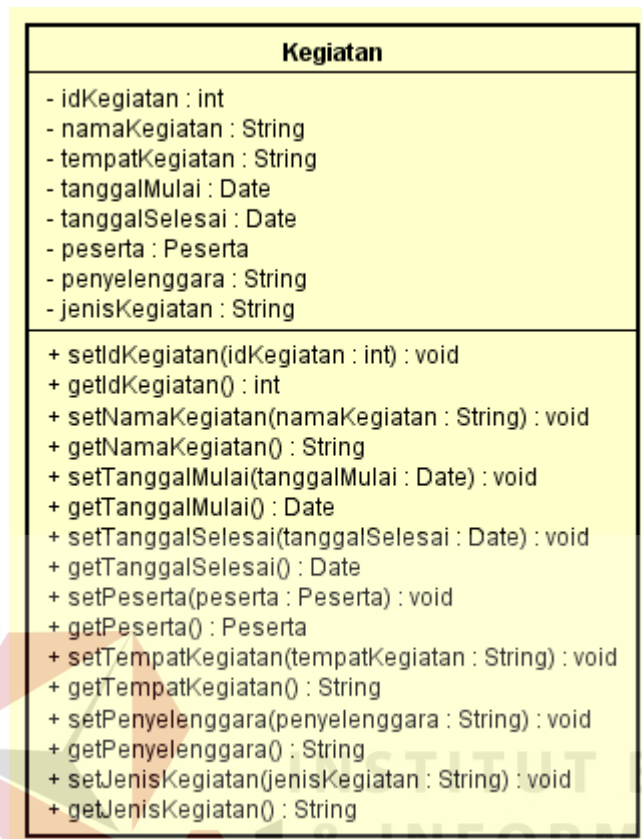
Pada Gambar 3.9 Diagram kolaborasi diatas menggambarkan hubungan antar aktor dengan obyek pada aplikasi dan pesan yang dikirimkan di dalamnya. Pesan – pesan yang dikirimkan tersebut nantinya akan dioleh oleh obyek – obyek tersebut untuk kemudian dikembalikan kembali pada aktor atau obyek lain yang mengirimnya.

3.3.3. *Class Diagram*

Class Diagram digunakan untuk menampilkan kelas – kelas yang saling berhubungan dalam sistem atau aplikasi. Sebuah aplikasi dapat terdiri dari beberapa diagram kelas yang saling berhubungan.

Pada aplikasi presensi ini, penggambaran diagram kelas menggunakan pendekatan *Model-View-Controller* (MVC). Dengan pendekatan tersebut, kelas yang dibuat dibagi menjadi beberapa bagian. Kelas yang termasuk dalam *Class Model* adalah kelas – kelas yang berhubungan dengan *database* atau mewakili entitas yang dibuat termasuk mengelola *business logic* aplikasi. Kelas yang termasuk dalam *Class View* adalah kelas – kelas yang berhubungan dengan tampilan atau *output* pada pengguna aplikasi. Dan kelas yang masuk ke dalam *Class Controller* merupakan kelas – kelas yang berfungsi untuk menghubungkan komunikasi antara kelas – kelas pada *Class View* dengan kelas – kelas yang berada pada *Class Model*.

A. Class Kegiatan



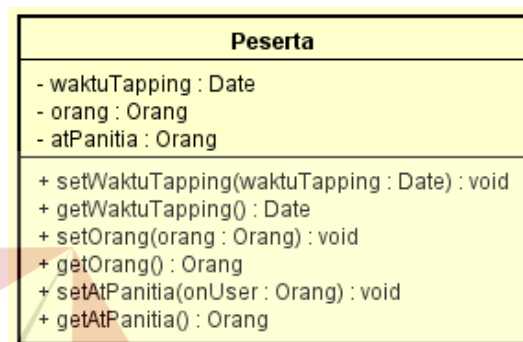
Gambar 3. 10 Diagram Kelas Kegiatan

Pada Gambar 3.10 Diagram Kelas Kegiatan termasuk dalam *Class Model*, dimana kelas ini yang akan melakukan interaksi dengan *database* yang berhubungan dengan data kegiatan. Atribut yang terdapat pada kelas kegiatan adalah `idKegiatan` yang berfungsi untuk menyimpan nomor unik kegiatan, `namaKegiatan` yang berfungsi untuk menyimpan nama kegiatan, `tempatKegiatan` yang berfungsi untuk menyimpan tempat kegiatan, `tanggalMulai` yang berfungsi untuk menyimpan tanggal mulai kegiatan, `tanggalSelesai` yang berfungsi untuk menyimpan tanggal selesai kegiatan, `peserta` yang berfungsi untuk menyimpan peserta kegiatan, `penyelenggara` yang berfungsi untuk menyimpan nama

penyelenggara dan jenisKegiatan yang berfungsi untuk menyimpan jenis kegiatan yang dilakukan.

Fungsi *getter* dan *setter* digunakan untuk mendapatkan dan mengisi nilai atribut – atribut yang terdapat pada kelas kegiatan karena atribut tersebut bersifat *private*.

B. Class Peserta

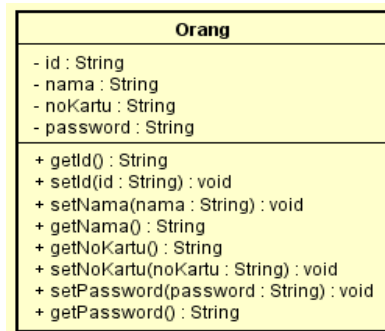


Gambar 3. 11 Diagram Kelas Peserta

Pada Gambar 3.11 Diagram Kelas peserta termasuk dalam *Class Model* dimana kelas ini berhubungan dengan data peserta kegiatan. Atribut yang dimiliki kelas peserta adalah waktuTapping yang berfungsi untuk mencatat waktu peserta kegiatan melakukan *tapping* kartu, orang yang merupakan *instance* dari kelas Orang yang berfungsi untuk menyimpan data orang yang melakukan *tapping* kartu, dan atPanitia yang berfungsi untuk menyimpan data panitia yang melakukan presensi.

Fungsi yang terdapat pada kelas ini juga merupakan fungsi *getter* dan *setter* yang bertujuan untuk mendapatkan dan mengisi nilai pada atribut kelas.

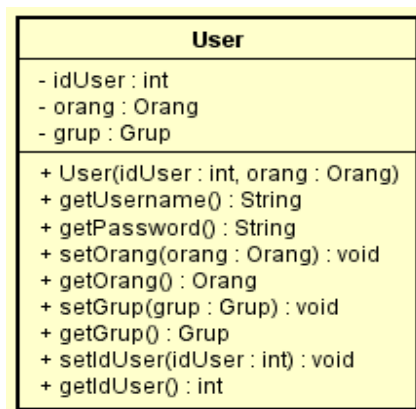
C. Class Orang



Gambar 3. 12 Diagram Kelas Orang

Pada Gambar 3.12 Diagram Kelas Orang termasuk dalam *Class Model* yang berguna untuk melakukan interaksi dengan *database* yang berhubungan dengan data orang. Atribut pada kelas ini adalah id yang berfungsi untuk menyimpan NIM atau NIK, nama yang berfungsi untuk menyimpan nama orang, noKartu yang berfungsi untuk menyimpan nomor unik kartu RFID yang dimiliki, dan password yang berfungsi untuk menyimpan kata sandi orang tersebut. Fungsi *getter* dan *setter* juga terdapat pada kelas ini untuk melakukan pengisian ataupun untuk mendapatkan nilai dari atribut kelas.

D. Class User

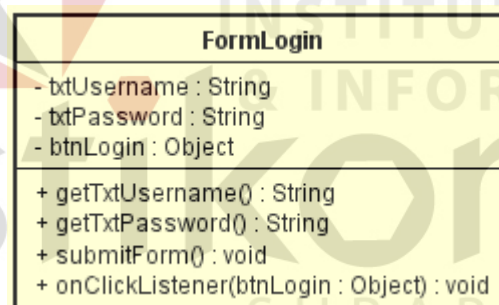


Gambar 3. 13 Diagram Kelas User

Pada Gambar 3.13 Diagram Kelas User juga termasuk dalam *Class Model* yang berhubungan dengan data pengguna aplikasi *web server*. Atribut yang terdapat pada kelas user adalah *idUser* yang berfungsi untuk menyimpan nomor unik user, orang yang merupakan *instance* dari kelas Orang yang berfungsi untuk menyimpan data orang yang terdaftar sebagai *user* dan grup yang berfungsi untuk menyimpan data grup user.

Selain fungsi *setter* dan *getter*, ada juga fungsi *constructor* *User()* dengan parameter *idUser* yang bertipe *int* dan orang yang bertipe *Orang*, dan fungsi *getUsername()* untuk mendapatkan id dari kelas Orang serta *getPassword()* untuk mendapatkan *password* dari kelas Orang.

E. Class FormLogin



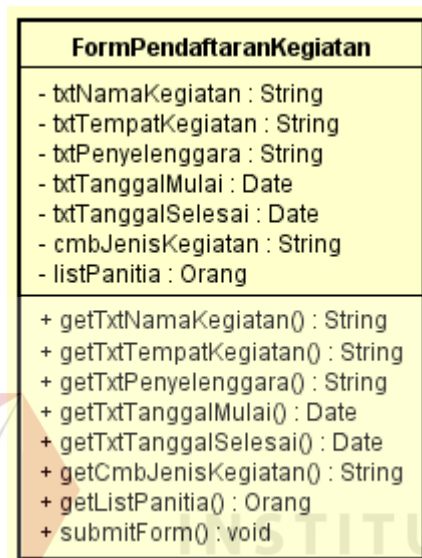
Gambar 3. 14 Diagram Kelas FormLogin

Pada Gambar 3.14 Diagram Kelas FormLogin merupakan kelas yang termasuk *Class View* karena kelas ini digunakan untuk menampilkan *user interface* login kepada pengguna aplikasi. Atribut pada kelas ini adalah *txtUsername* untuk menyimpan *username* pengguna, *txtPassword* untuk menampung *password* pengguna dan *btnLogin* yang merupakan tombol *login*.

Fungsi *getter* pada pada kelas ini untuk mendapatkan nilai dari *txtUsername* dan *txtPassword* yang diisi oleh pengguna. Fungsi *submitForm()*

digunakan untuk mengirimkan *username* dan *password* ke *controller*. Fungsi *onClickListener()* dipanggil ketika tombol *login* diklik oleh pengguna dan kemudian memanggil fungsi *submitForm()*.

F. *Class FormPendaftaranKegiatan*

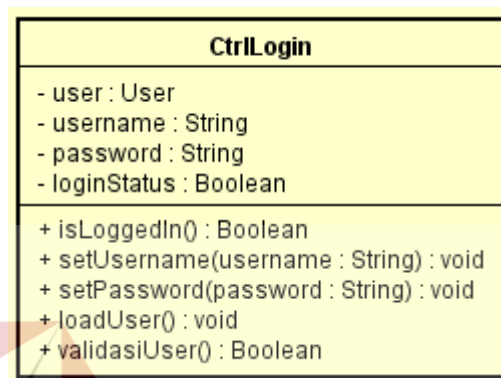


Gambar 3. 15 Diagram Kelas *FormPendaftaranKegiatan*

Pada Gambar 3.15 Diagram Kelas *FormPendaftaranKegiatan* termasuk dalam *Class View* karena kelas ini menampilkan halaman pendaftaran kegiatan kepada pengguna aplikasi. Atribut pada kelas ini adalah *txtNamaKegiatan* untuk menampung data nama kegiatan, *txtTempatKegiatan* untuk menampung data tempat kegiatan, *txtPenyelenggara* untuk menampung data penyelenggara, *txtTanggalMulai* untuk menampung data waktu mulai kegiatan, *txtTanggalSelesai* untuk menampung data waktu selesai kegiatan, *cmbJenisKegiatan* untuk menampung data jenis peserta kegiatan dan *listPanitia* untuk menampung data panitia kegiatan.

Fungsi *getter* dan *setter* kelas ini sama dengan *getter* dan *setter* pada kelas lain, yaitu untuk mengisi dan membaca nilai dari atribut – atribut yang terdapat pada kelas *FormPendaftaranKegiatan*. Fungsi *submitForm()* digunakan untuk mengirim isian pengguna ke kelas *controller*.

G. Class CtrlLogin

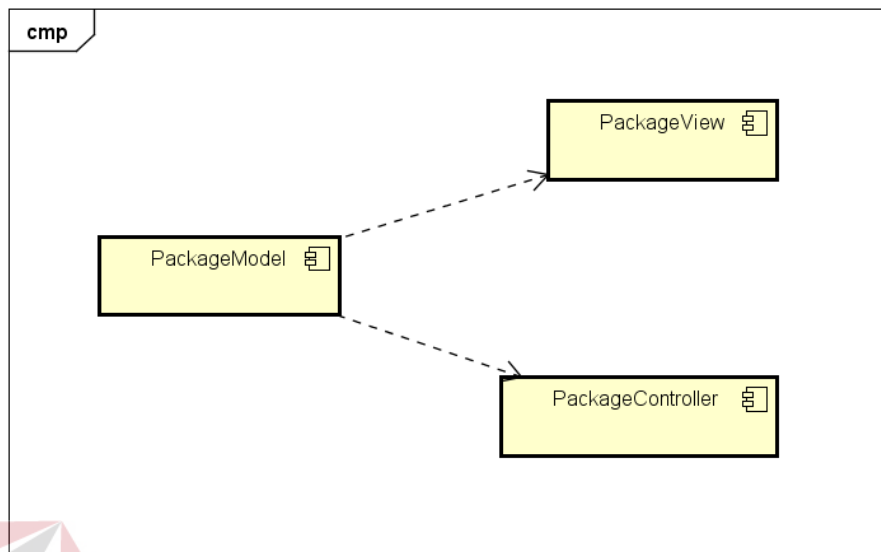


Gambar 3. 16 Diagram Kelas CtrlLogin

Pada Gambar 3.16 Diagram Kelas CtrlLogin adalah bagian dari *Controller Class* karena digunakan untuk menghubungkan kelas *FormLogin* dengan kelas *User*. Atribut yang terdapat pada kelas ini adalah *user* yang merupakan *instance* dari kelas *User*, *username* untuk menampung data *username* dari *FormLogin*, *password* untuk menampung data *password* dari *FormLogin* dan *LoginStatus* untuk menampung status *login* dari pengguna.

Fungsi *isLoggedIn()* digunakan untuk mengecek apakah suatu pengguna aplikasi sudah melakukan proses *login* atau belum. Fungsi *validasiUser()* digunakan untuk melakukan validasi isian *username* dan *password* dengan data yang ada pada *database*. Dan fungsi *loadUser()* digunakan untuk mengambil data user dari *database*.

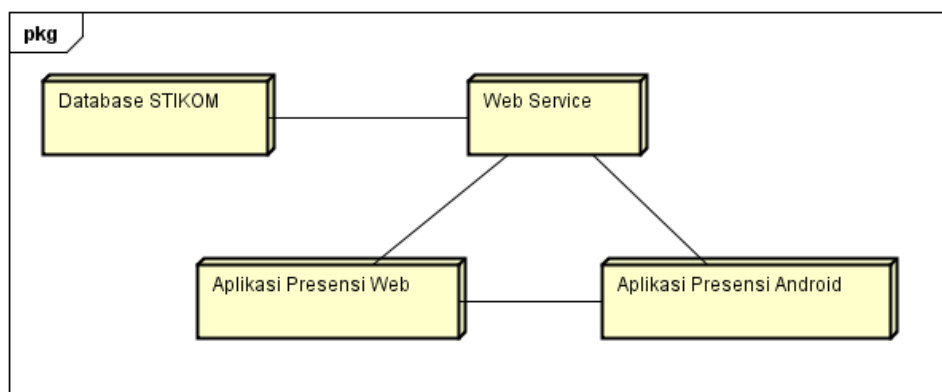
3.3.4 . Component Diagram



Gambar 3. 17 Diagram Komponen

Pada Gambar 3.17 Diagram komponen memberikan panduan bagi pengembang untuk mengetahui komponen mana yang harus terlebih dahulu dilakukan kompilasi. Gambar diatas menunjukkan bahwa komponen yang terdapat pada *package Model* harus dikompilasi terlebih dahulu karena memiliki dependensi terhadap komponen yang berada pada *package View* dan *package Controller*.

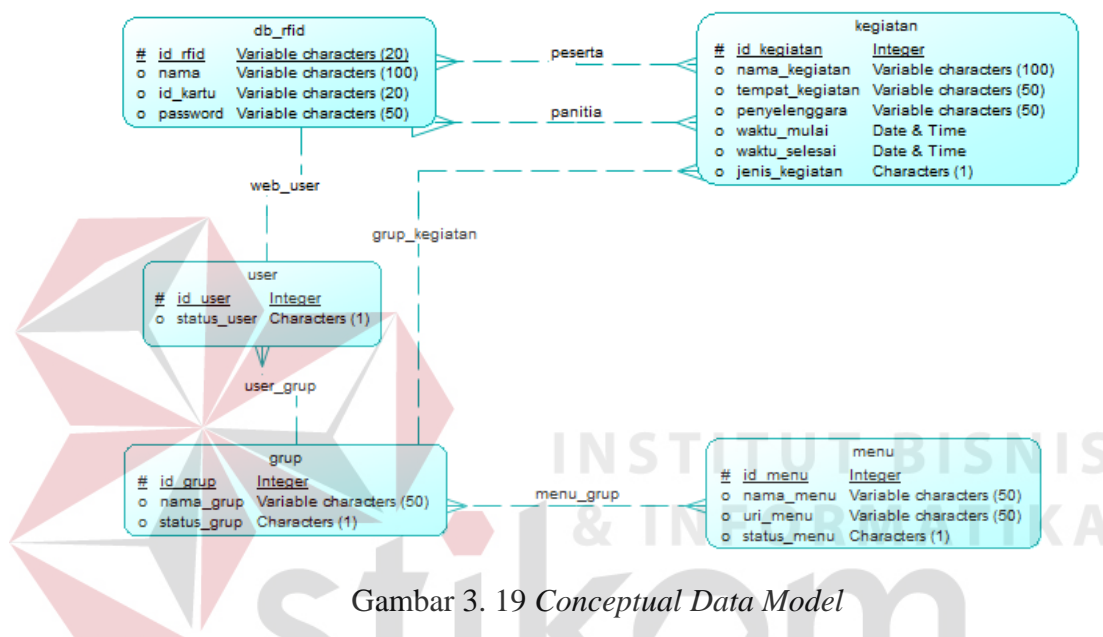
3.3.5. Deployment Diagram



Gambar 3. 18 Deployment Diagram

Pada Gambar 3.18 *Deployment diagram* diatas menggambarkan hubungan antar sistem yang dibangun dengan sistem lainnya. Dari gambar di atas, aplikasi presensi web dan aplikasi presensi *Android* berhubungan dengan *web service* untuk dapat mengakses *database* yang dimiliki Stikom Surabaya.

3.3.6. Conceptual Data Model



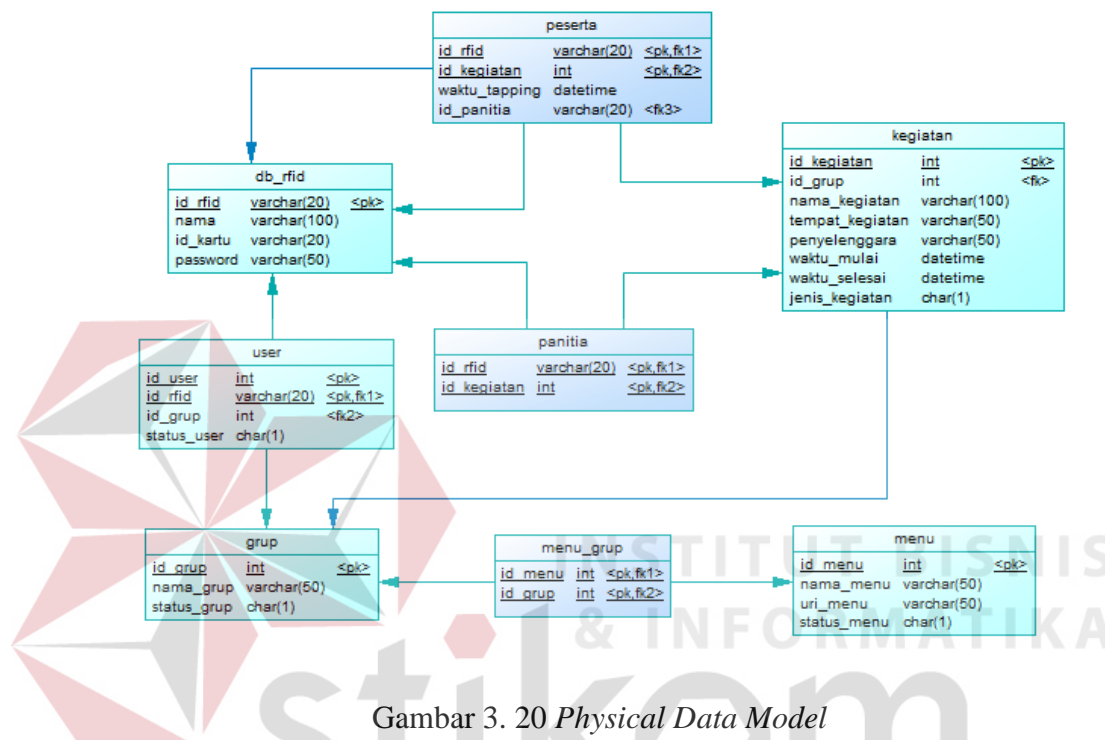
Gambar 3. 19 *Conceptual Data Model*

Pada Gambar 3.19 *Conceptual Data Model* diatas, ditunjukkan entitas utama yaitu kegiatan, db_rfid, user, grup dan menu. Entitas kegiatan adalah entitas yang memiliki atribut yang berhubungan dengan kegiatan. Entitas kegiatan memiliki relasi *many to many* dengan entitas db_rfid. Entitas db_rfid adalah entitas *web service* yang berasal dari *database* internal STIKOM Surabaya dengan atribut berupa *id_rfid*, *nama*, *id_kartu* dan *password*. Relasi yang dihasilkan antara entitas kegiatan dengan entitas db_rfid adalah relasi panitia dan relasi peserta.

Entitas user adalah entitas yang menyimpan data pengguna aplikasi yang berasal dari hubungan entitas db_rfid dengan entitas grup. Pada entitas ini terdapat *id_user* sebagai *primary key* dan *status_user*. Entitas grup memiliki atribut

id_grup, nama_grup, dan status_grup. Entitas ini memiliki relasi dengan entitas menu dimana entitas menu digunakan untuk memberikan hak akses terhadap pengguna aplikasi presensi web.

3.3.7. Physical Data Model



Gambar 3. 20 Physical Data Model

Pada Gambar 3.20 Physical Data Model (PDM) adalah bentuk desain yang akan digunakan dalam membuat skema basis data aplikasi nantinya. PDM merupakan hasil *generate* dari *Conceptual Data Model* pada sub bab sebelumnya yang menghasilkan beberapa tabel baru sesuai dengan hasil relasi antar entitas. Tabel baru tersebut dihasilkan dari relasi *many to many* dua entitas. Tabel baru hasil *generate* CDM adalah peserta, panitia, dan menu_grup.

3.3.8. Struktur Tabel

A. Tabel Kegiatan

Nama tabel : Kegiatan

Primary key : id_kegiatan

Foreign key : id_grup

Fungsi : Menyimpan data kegiatan

Tabel 3. 6 Tabel Kegiatan

No.	Field Name	Data Type	Length	Constraint	Foreign Key	
					On Field	On Table
1	id_kegiatan	integer	11	PK		
2	nama_kegiatan	varchar	100			
3	tempat_kegiatan	varchar	50			
4	penyelenggara	varchar	50			
5	waktu_mulai	datetime				
6	waktu_selesai	datetime				
7	jenis_kegiatan	char	1			
8	id_grup	integer	11		Id_grup	Grup

B. Tabel Panitia

Nama tabel : Panitia

Primary key : id_kegiatan, id_rfid

Foreign key : id_kegiatan, id_rfid

Fungsi : Menyimpan data panitia kegiatan

Tabel 3. 7 Tabel Panitia

No.	Field Name	Data Type	Length	Constraint	Foreign Key	
					On Field	On Table
1	id_kegiatan	integer	11	PK, FK	id_kegiatan	kegiatan
2	id_rfid	varchar	20	PK, FK	id_rfid	db_rfid

C. Tabel Peserta

Nama tabel : Peserta

Primary key : id_kegiatan, id_rfid

Foreign key : id_kegiatan, id_rfid

Fungsi : Menyimpan data peserta kegiatan

Tabel 3. 8 Tabel Peserta

No.	Field Name	Data Type	Length	Constraint	Foreign Key	
					On Field	On Table
1	id_kegiatan	integer	11	PK, FK	id_kegiatan	kegiatan
2	id_rfid	varchar	20	PK, FK	id_rfid	db_rfid
3	waktu_tapping	datetime				
4	id_panitia	varchar	20	FK	id_rfid	db_rfid

D. Tabel User

Nama tabel : User

Primary key : id_user

Foreign key : id_grup, id_rfid

Fungsi : Menyimpan data pengguna aplikasi presensi web

Tabel 3. 9 Tabel User

No.	Field Name	Data Type	Length	Constraint	Foreign Key	
					On Field	On Table
1	id_user	integer	11	PK		
2	id_rfid	varchar	20	PK, FK	id_rfid	db_rfid
3	id_grup	integer	11	FK	id_grup	grup
4	status_user	Char	1			

E. Tabel Group

Nama tabel : Grup

Primary key : id_grup

Foreign key : -

Fungsi : Menyimpan data grup pengguna aplikasi

Tabel 3. 10 Tabel Group

No.	Field Name	Data Type	Length	Constraint	Foreign Key	
					On Field	On Table
1	id_grup	integer	11	PK		
2	Nama_grup	varchar	50			
3	Status_grup	Char	1			

F. Group Menu

Nama tabel : Grup_menu

Primary key : id_grup, id_menu

Foreign key : id_grup, id_menu

Fungsi : Menyimpan data relasi antara tabel grup dan tabel menu

Tabel 3. 11 Tabel Grup_menu

No.	Field Name	Data Type	Length	Constraint	Foreign Key	
					On Field	On Table
1	id_grup	integer	11	PK, FK	id_grup	Grup
2	id_menu	Integer	11	PK, FK	id_menu	Menu

G. Tabel Menu

Nama tabel : Menu

Primary key : id_menu

Foreign key : -

Fungsi : Menyimpan data menu hak akses pengguna aplikasi presensi web

Tabel 3. 12 Tabel Menu

No.	Field Name	Data Type	Length	Constraint	Foreign Key	
					On Field	On Table
1	id_menu	integer	11	PK		
2	Nama_menu	Varchar	50			
3	Uri	Varchar	50			

No.	Field Name	Data Type	Length	Constraint	Foreign Key	
					On Field	On Table
4	Status_menu	Char	1			

3.3.9. Desain Input dan Output

Desain *input* dan *output* merupakan rancangan awal tampilan masukan maupun keluaran aplikasi. Desain *input* dan *output* menunjukkan bagaimana cara pengguna dapat berinteraksi dengan aplikasi yang dibuat. Berdasarkan desain *input* dan *output*, maka selanjutnya dapat dikerjakan tampilan aplikasi yang akan dibuat.

Desain *input* dan *output* pada aplikasi ini mencakup desain *input* dan *output* pada aplikasi presensi *web* dan aplikasi presensi *Android*.

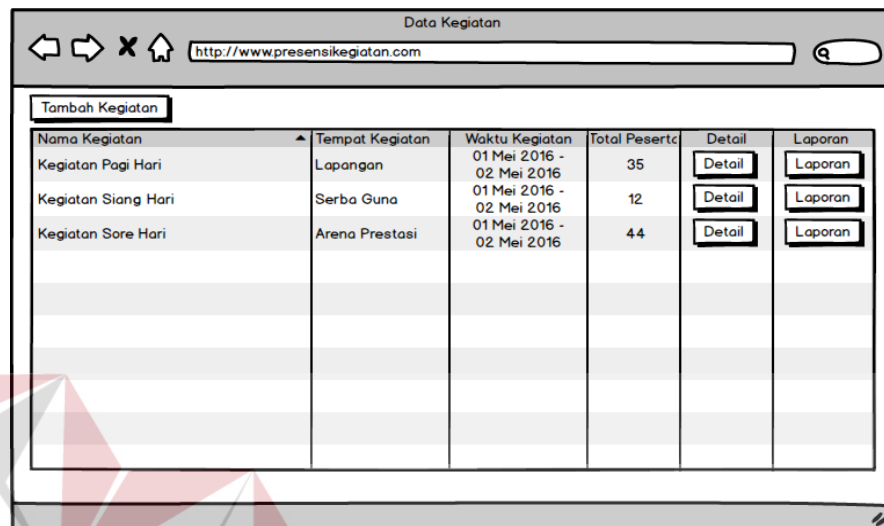
A. Desain Halaman *Login Web*

Gambar 3. 21 Desain Halaman *Login Web*

Pada Gambar 3.21 Halaman *login web* digunakan oleh admin atau panitia yang akan mengakses aplikasi presensi *web*. Pada tampilan *Login Web*, pengguna mengisi NIM atau NIK ke kolom isian *username* dan mengisi kata sandi

pada kolom isian *password*. Kemudian pengguna mengklik tombol Login untuk melakukan *submit* isian.

B. Desain Halaman Data Kegiatan

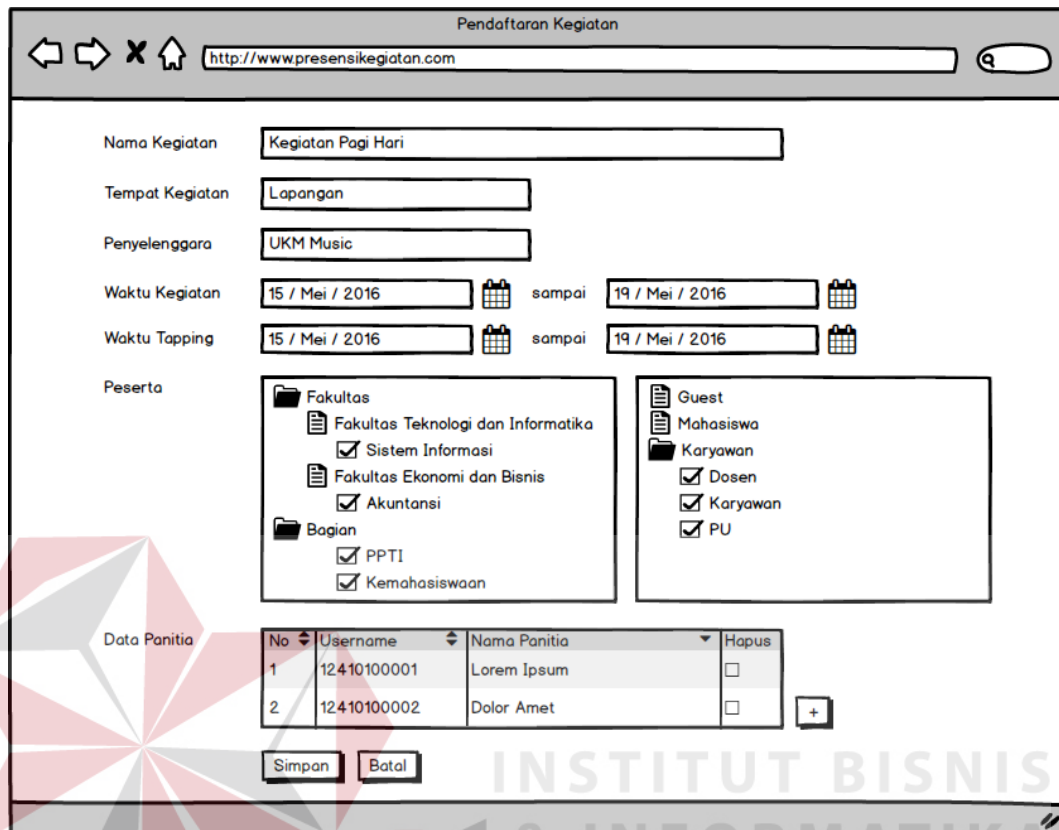


Nama Kegiatan	Tempat Kegiatan	Waktu Kegiatan	Total Peserta	Detail	Laporan
Kegiatan Pagi Hari	Lapangan	01 Mei 2016 - 02 Mei 2016	35	Detail	Laporan
Kegiatan Siang Hari	Serba Guna	01 Mei 2016 - 02 Mei 2016	12	Detail	Laporan
Kegiatan Sore Hari	Arena Prestasi	01 Mei 2016 - 02 Mei 2016	44	Detail	Laporan

Gambar 3. 22 Desain Halaman Data Kegiatan

Pada Gambar 3.22 Desain Halaman data kegiatan ditampilkan kepada pengguna setelah pengguna aplikasi berhasil melakukan *login*. Data kegiatan yang ditampilkan untuk pengguna disesuaikan dengan grup pengguna. Bagi pengguna aplikasi yang merupakan bagian dari panitia kegiatan, maka data kegiatan yang ditampilkan hanya kegiatan yang sedang berlangsung dimana panitia tersebut didaftarkan. Dari halaman ini, pengguna bisa melihat detail kegiatan dan juga peserta yang sudah melakukan presensi pada kegiatan tersebut. Selain itu, pengguna juga dapat mencetak laporan presensi kegiatan ke dalam bentuk pdf. Tombol tambah kegiatan digunakan oleh pengguna yang termasuk dalam *user* aplikasi. Sedangkan panitia tidak dapat menggunakan tombol tersebut.

C. Desain Halaman Pendaftaran Kegiatan



Pendaftaran Kegiatan

http://www.presensikegiatan.com

Nama Kegiatan:

Tempat Kegiatan:

Penyelenggara:

Waktu Kegiatan: sampai

Waktu Tapping: sampai

Peserta:

- ☒ Fakultas
 - ☒ Fakultas Teknologi dan Informatika
 - ☒ Sistem Informasi
 - ☒ Fakultas Ekonomi dan Bisnis
 - ☒ Akuntansi
- ☒ Bagian
 - ☒ PPTI
 - ☒ Kemahasiswaan
- ☒ Guest
 - ☒ Mahasiswa
- ☒ Karyawan
 - ☒ Dosen
 - ☒ Karyawan
 - ☒ PU

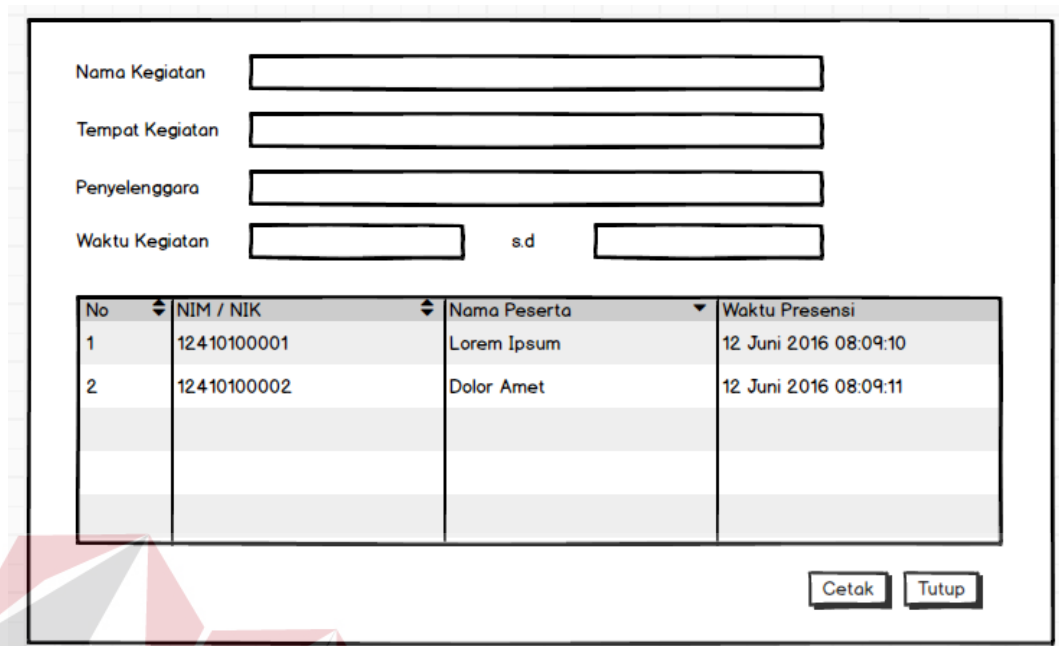
Data Panitia

No	Username	Nama Panitia	Hapus
1	12410100001	Lorem Ipsum	<input type="checkbox"/>
2	12410100002	Dolor Amet	<input type="checkbox"/>

Gambar 3. 23 Desain Halaman Pendaftaran Kegiatan

Pada Gambar 3.23 Desain Halaman pendaftaran kegiatan digunakan oleh pengguna yang terdaftar untuk mendaftarkan kegiatannya. Halaman ini tidak dapat diakses oleh pengguna yang terdaftar sebagai panitia. Pada halaman ini, pengguna mengisi nama kegiatan, tempat kegiatan berlangsung, penyelenggara kegiatan, tanggal mulai dan tanggal selesai kegiatan, waktu mulai presensi hingga waktu selesai presensi dan peserta kegiatan serta panitia yang bertugas melakukan proses presensi kegiatan. Setelah mengisi semua kolom isian, pengguna melakukan *submit form* dengan mengklik tombol simpan. Tombol batal digunakan untuk membatalkan pendaftaran dan kembali ke halaman utama atau *form data kegiatan*.

D. Desain Halaman Detail Kegiatan



Nama Kegiatan
 Tempat Kegiatan
 Penyelenggara
 Waktu Kegiatan s.d

No	NIM / NIK	Nama Peserta	Waktu Presensi
1	12410100001	Lorem Ipsum	12 Juni 2016 08:09:10
2	12410100002	Dolor Amet	12 Juni 2016 08:09:11

Gambar 3. 24 Desain Halaman Detail Kegiatan

Pada Gambar 3.24 Desain Halaman detail kegiatan ini digunakan untuk menampilkan detail kegiatan yang telah didaftarkan. Detail kegiatan menampilkan nama kegiatan, tempat kegiatan berlangsung, penyelenggara kegiatan, waktu kegiatan berlangsung dan daftar peserta kegiatan. Pada halaman ini juga terdapat tombol cetak untuk mencetak laporan kegiatan ke dalam bentuk pdf.

E. Desain Halaman *Login Android*



Gambar 3. 25 Desain Halaman Login Android

Pada Gambar 3.25 Desain Tampilan halaman *login* Android digunakan oleh panitia untuk dapat masuk ke dalam aplikasi presensi. Panitia mengisi NIM atau NIK pada kolom isian username dan kata sandi ke dalam isian *password*. Selanjutnya panitia mengklik tombol *login* untuk masuk ke dalam aplikasi.

Pengguna yang memiliki akses akan langsung diarahkan ke halaman utama aplikasi. Aplikasi akan menampilkan pesan kesalahan apabila *username*

atau *password* tidak sesuai atau pengguna tidak memiliki hak akses sebagai panitia kegiatan.

F. Desain Halaman *Login Android*



Gambar 3. 26 Desain Halaman Data Kegiatan *Android*

Pada Gambar 3.26 Desain halaman data kegiatan *android* menampilkan seluruh kegiatan yang sedang berlangsung sesuai dengan daftar kegiatan yang dimiliki oleh panitia yang sedang *login*. Panitia dapat memilih dari daftar kegiatan

yang ditampilkan untuk bisa masuk ke halaman presensi dan menampilkan data peserta dengan menekan tombol presensi.

G. Desain Halaman *Tapping Kegiatan Android*



Gambar 3. 27 Desain Halaman *Tapping Kegiatan Android*

Pada Gambar 3.27 Desain Halaman *Form tapping* kegiatan digunakan oleh panitia untuk melakukan proses presensi dengan menempelkan kartu *RFID* peserta ke area NFC perangkat. Nomor kartu yang terbaca oleh aplikasi kemudian ditampilkan pada kolom Kode Kartu. Jika aplikasi dapat tersambung ke *server*, maka aplikasi akan melakukan *query* NIM atau NIK dan nama peserta dan menampilkannya pada kolom NIM / NIK dan nama. Tombol simpan hanya

digunakan bila kartu tidak dapat dikenali dan panitia mengisi kolom NIM / NIK secara manual.

H. Desain Halaman Info Kegiatan *Android*



Gambar 3. 28 Desain Halaman Info Kegiatan *Android*

Pada Gambar 3.28 Desain Halaman info kegiatan ini menampilkan data mengenai kegiatan yang dipilih oleh panitia untuk dilakukan proses presensi. Pada halaman ini, ditampilkan nama kegiatan, tempat kegiatan, penyelenggara, waktu mulai kegiatan, waktu selesai kegiatan dan jenis peserta kegiatan.

I. Desain Halaman Data Peserta Kegiatan *Android*



Gambar 3. 29 Desain Halaman Data Peserta Kegiatan Android

Pada Gambar 3.29 Desain Halaman data presensi menampilkan seluruh data peserta yang telah melakukan proses presensi atau *tapping* ke aplikasi presensi. Halaman ini menampilkan NIM atau NIK peserta, nama peserta dan waktu saat peserta melakukan *tapping*. Data peserta yang ditampilkan sesuai dengan kegiatan yang dipilih dihalaman utama.

3.4. Tahap Pengujian

Pengujian dilakukan dengan membuat *test plan* berdasarkan fungsionalitas tiap *form* yang dirancang. Pengujian yang dilakukan menggunakan metode *Black Box Testing* dengan mengamati hasil keluaran dari tiap fungsi yang diuji coba.

3.4.1. Desain Pengujian Halaman Login

Tabel 3. 13 Tabel Desain Pengujian Halaman *Login*

No	Tujuan	Input	Output Diharapkan
1.	Menguji <i>textbox password</i>	Gabungan huruf dan angka	Karakter yang diinputkan tampil sebagai simbol *
2.	Menguji validasi isian	NIM / NIK dan kata sandi yang benar	Menampilkan halaman utama dan nama pengguna
3.	Menguji validasi isian	NIM / NIK dan kata sandi yang salah	Menampilkan pesan kesalahan
4	Menguji validasi hak akses	NIM / NIK dan kata sandi bukan panitia	Menampilkan pesan kesalahan hak akses

3.4.2. Desain Pengujian Halaman Data Kegiatan Web

Tabel 3. 14 Tabel Desain Pengujian Halaman Data Kegiatan *Web*

No	Tujuan	Input	Output Diharapkan
1.	Menampilkan Data Kegiatan		Data kegiatan tampil sesuai dengan hak akses
2.	Menguji visibilitas tombol tambah kegiatan	<i>Login</i> berdasarkan hak akses	Tombol login hanya muncul pada pengguna berstatus <i>user</i> dan tidak muncul pada panitia
3.	Menguji tombol tambah kegiatan	Tombol tambah kegiatan	Menampilkan halaman pendaftaran kegiatan

4.	Menguji tombol detail kegiatan	Tombol detail kegiatan	Menampilkan <i>form</i> detail kegiatan
5.	Menguji tombol cetak laporan	Tombol cetak laporan	Menampilkan halaman pdf hasil cetak laporan

3.4.3 . Desain Pengujian Halaman Pendaftaran Kegiatan

Tabel 3. 15 Tabel Desain Pengujian Halaman Pendaftaran Kegiatan *Web*

No	Tujuan	Input	Output Diharapkan
1.	Menguji tombol pilih waktu kegiatan	Tombol pilih waktu kegiatan	Tampil <i>datepicker</i> untuk memilih waktu kegiatan
2.	Menguji tombol tambah panitia	Tombol tambah panitia	Ada baris baru pada tabel panitia
3.	Menguji tombol simpan	Tombol simpan	Aplikasi menyimpan kegiatan dan mengosongkan <i>form</i> isian
4.	Menguji tombol batal	Tombol batal	Mengosongkan <i>form</i> isian dan kembali ke halaman utama

3.3.4. Desain Pengujian Halaman *Login Android*

Tabel 3. 16 Tabel Desain Pengujian Halaman *Login Android*

No	Tujuan	Input	Output Diharapkan
1.	Menguji <i>textbox password</i>	Password Data 1	Karakter yang diinputkan tampil sebagai simbol *
2.	Menguji validasi isian	Username Data 1, Password Data 1	Menampilkan halaman utama
3.	Menguji validasi isian	Username Data 1, Password Data 2	Menampilkan pesan kesalahan
4.	Menguji koneksi ke server		Menampilkan pesan tidak dapat terhubung ke server

3.3.5. Desain Pengujian Halaman Daftar Kegiatan *Android*

Tabel 3. 17 Tabel Desain Pengujian Halaman Daftar Kegiatan *Android*

No	Tujuan	Input	Output Diharapkan
1.	Menampilkan Data Kegiatan		Data kegiatan tampil sesuai dengan hak akses
2.	Menguji tombol buka presensi kegiatan	Tombol buka presensi	Menampilkan halaman presensi

3.3.6. Desain Pengujian Halaman Tapping Kegiatan *Android*

Tabel 3. 18 Tabel Desain Pengujian Halaman Tapping Kegiatan *Android*

No	Tujuan	Input	Output Diharapkan
1.	Menguji fungsi info kegiatan		Info kegiatan menampilkan data dengan benar
2.	Menguji tombol simpan manual	Tombol simpan manual	Menampilkan pesan data tersimpan

3.3.7. Desain Pengujian Halaman Data Peserta Kegiatan

Tabel 3. 19 Tabel Desain Pengujian Halaman Data Peserta Kegiatan

No	Tujuan	Input	Output Diharapkan
1.	Menampilkan Data Peserta		Data peserta tampil sesuai dengan kegiatan yang dipilih