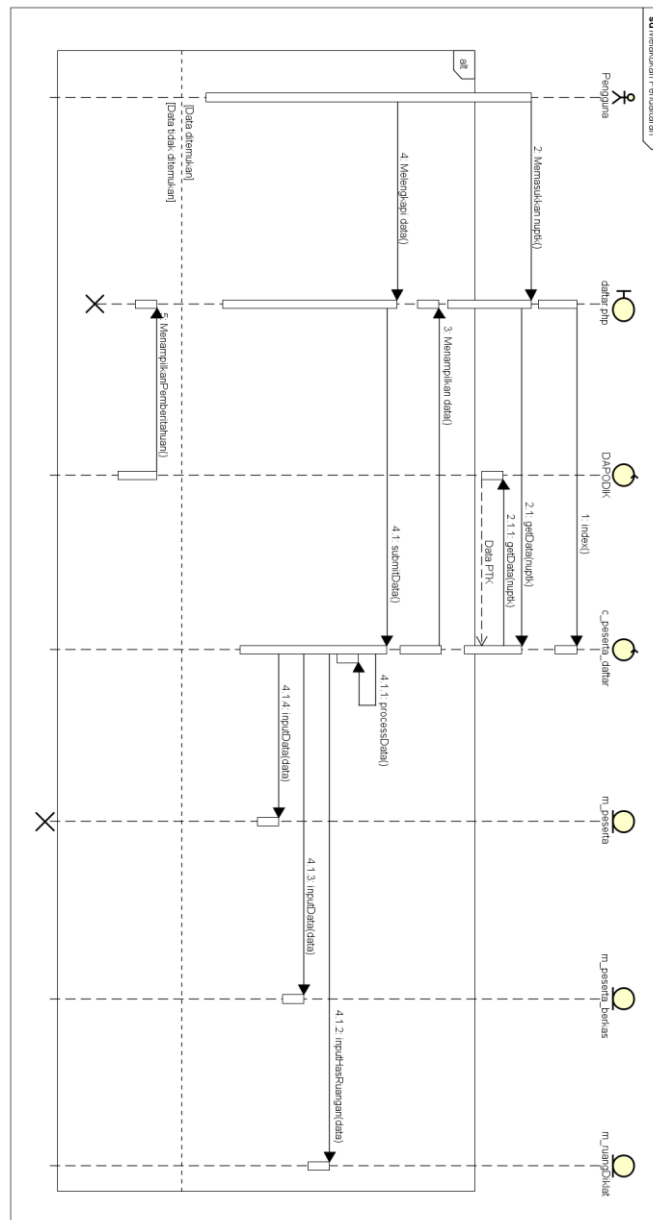


BAB 5 PERANCANGAN

Bab ini membahas perancangan Sistem Informasi Manajemen DIKLAT yang dilakukan dengan melakukan perancangan *sequence* diagram, *class* diagram dan perancangan database.

5.1 Perancangan *Sequence* Diagram

Sequence diagram adalah diagram yang menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek.

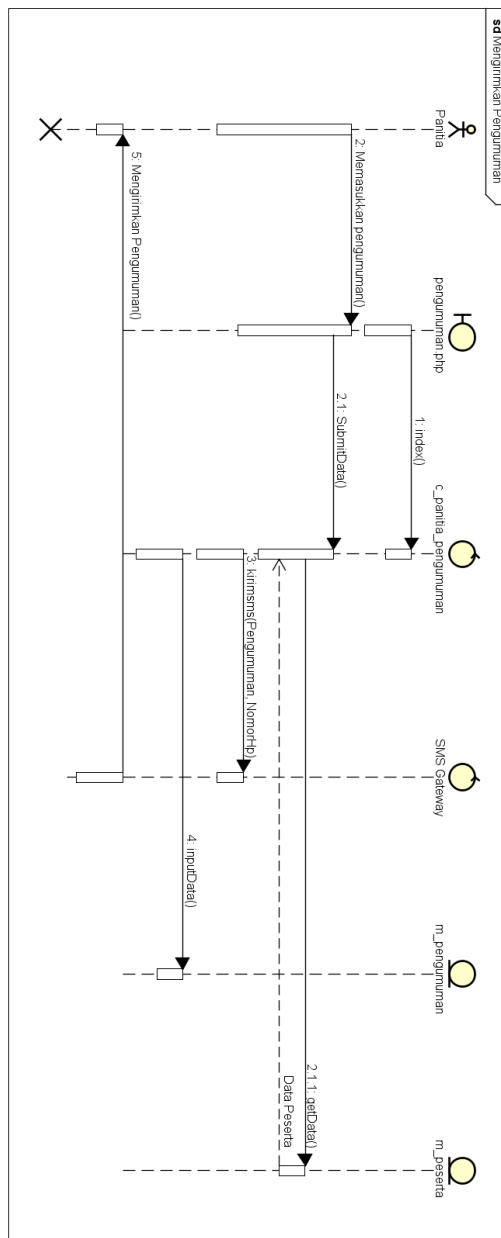


Gambar 5.1 *Sequence* Diagram Pendaftaran

5.1.1 Sequence Diagram Melakukan Pendaftaran

Sequence diagram melakukan pendaftaran adalah sebuah alur yang dijalankan oleh aktor pengguna. Aktor pengguna ini dapat melakukan pendaftaran pada sistem dengan memasukkan NUPTK pada sistem, kemudian sistem akan mengirimkan NUPTK tersebut ke aplikasi DAPODIK untuk mengetahui apakah pengguna tersebut terdaftar sebagai pendidik dan atau tenaga kependidikan. Selanjutnya jika pengguna terdaftar, sistem akan mendapatkan data peserta tersebut dan jika pengguna tidak terdaftar dalam aplikasi DAPODIK, maka sistem akan menampilkan pemberitahuan bahwa peserta tidak terdaftar dalam aplikasi DAPODIK dan peserta tidak dapat melanjutkan proses pendaftaran. Data yang dikirimkan aplikasi DAPODIK belum lengkap untuk digunakan dalam proses pendaftaran. Oleh karena itu, pengguna wajib melengkapi data tersebut. Setelah pengguna berhasil melakukan pendaftaran, panitia akan melakukan verifikasi untuk melihat apakah data peserta yang telah dimasukkan telah lengkap atau belum. Jika data peserta telah lengkap panitia menekan tombol verifikasi dan sistem akan mengirimkan *username* dan *password* yang dapat digunakan untuk pengguna masuk ke dalam sistem sebagai peserta dan dapat melakukan proses-proses lainnya. Alur tersebut digambarkan pada Gambar 5.1.

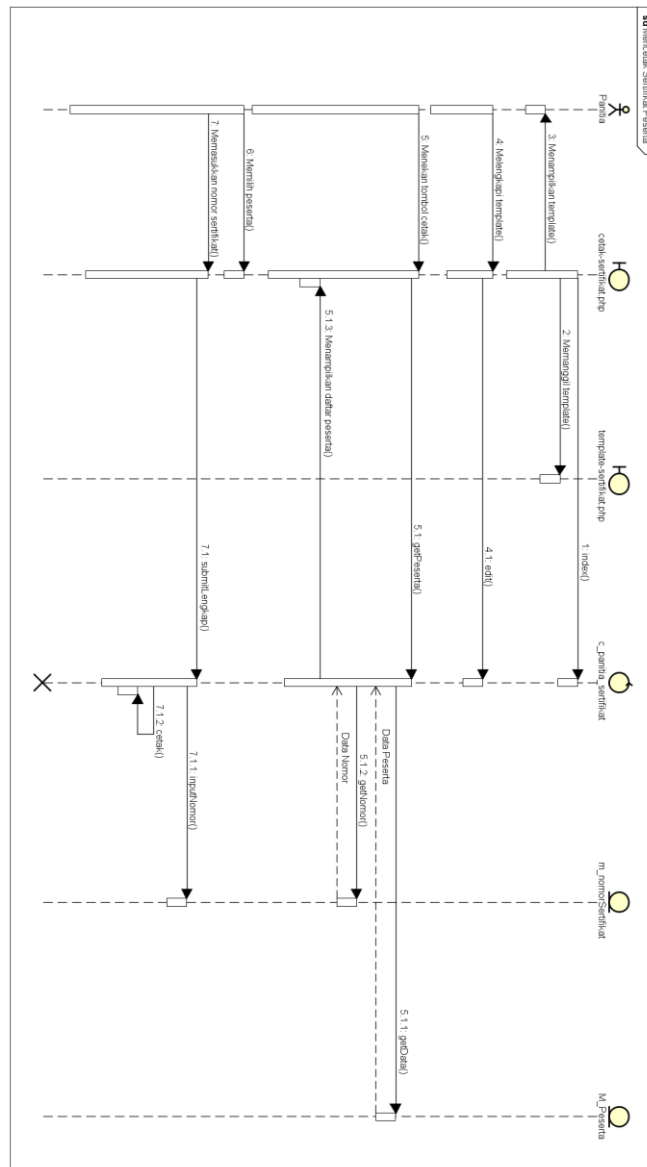
Dalam proses ini terdapat integrasi dengan aplikasi DAPODIK guna memudahkan sistem dalam membatasi siapa saja yang bisa melakukan proses ini. Aplikasi DAPODIK adalah aplikasi pengolahan data terpusat yang mengolah data peserta didik, pendidik, tenaga kependidikan, hingga profil sekolah-sekolah. Aplikasi ini digunakan secara nasional diseluruh indonesia untuk memudahkan operator sekolah (OPS) dalam menyatukan data kependidikan, memverifikasi data guru dan siswa dalam proses pencairan dana tunjangan atau beasiswa dan dapat juga digunakan oleh aplikasi lain untuk berbagai macam proses. Dalam pengembangan Sistem Informasi Manajemen Diklat ini, proses integrasi dilakukan melalui *Application Programming Interface* (API) dalam bentuk HTTP REST API. HTTP REST API adalah itegrasi melalui URL tertentu dimana URL tersebut akan mengembalikan data sesuai dengan permintaan. Namun, aplikasi DAPODIK yang digunakan adalah aplikasi buatan yang digunakaan sebagai contoh bukan aplikasi DAPODIK yang sebenarnya karena tidak mendapatkan hak akses untuk mengakses aplikasi DAPODIK yang sebenarnya.



Gambar 5.2 Sequence Diagram Mengirimkan informasi

5.1.2 Sequence Diagram Mengirimkan Pengumuman

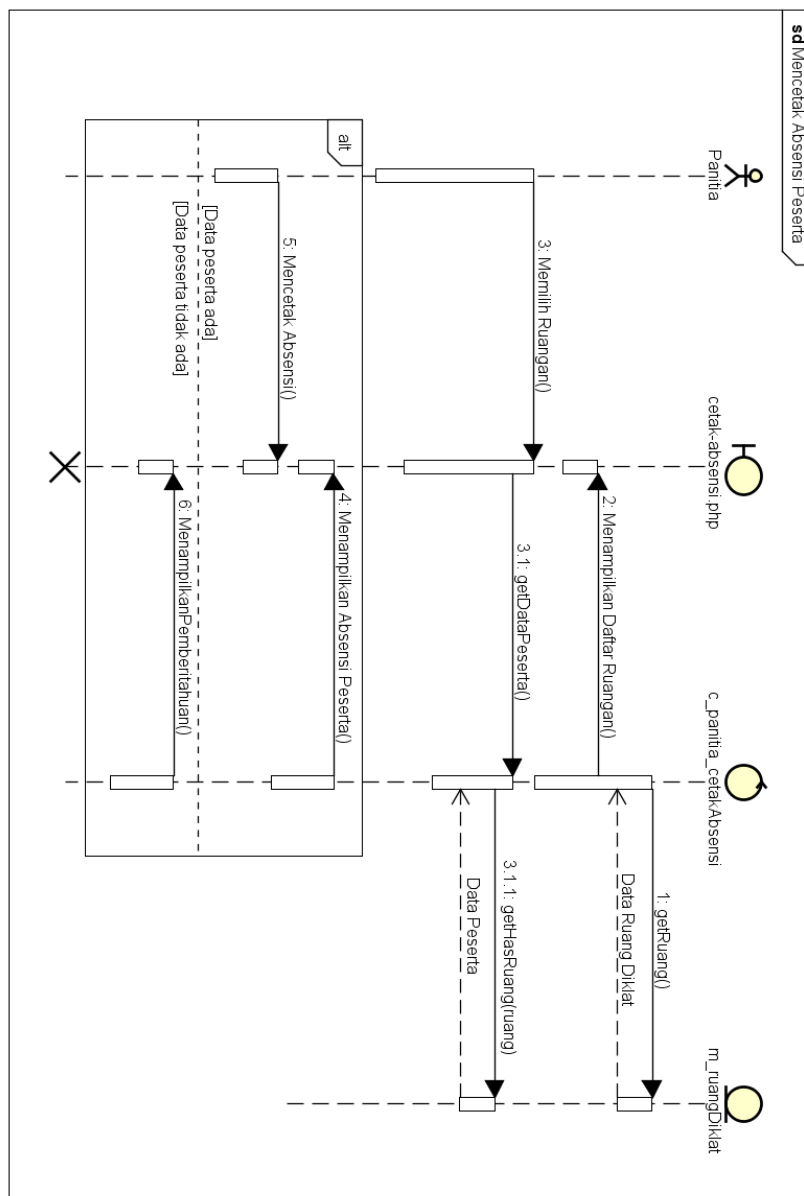
Sequence diagram mengirimkan pengumuman adalah sebuah alur yang dapat dijalankan oleh aktor panitia untuk dapat mengirimkan pengumuman kepada seluruh peserta yang sedang mengikuti kegiatan diklat. Pengumuman dikirimkan melalui sistem dengan berkomunikasi dengan aplikasi SMS gateway. Proses tersebut dimulai dari panitia memasukkan pengumuman yang akan dikirimkan, lalu sistem mengambil data nomor handphone peserta, kemudian sistem mengirimkan data pengumuman beserta nomor handphone peserta ke aplikasi SMS gateway. Selanjutnya aplikasi SMS gateway akan mengirimkan pengumuman tersebut kepada peserta yang disebarkan melalui SMS. Alur tersebut digambarkan pada Gambar 5.2.



Gambar 5.3 Sequence Diagram Mencetak Sertifikat Peserta

5.1.3 Sequence Diagram Mencetak Sertifikat Peserta

Sequence diagram mencetak sertifikat peserta merupakan sebuah proses yang dapat dijalankan oleh aktor panitia dimana panitia dapat mencetak sertifikat peserta dengan bantuan sistem. Proses ini dimulai dari panitia memasukkan gambar tanda tangan yang dibutuhkan pada sertifikat dan menyimpannya, lalu sistem mengambil seluruh data peserta, kemudian sistem menempelkan gambar tanda tangan tersebut beserta data peserta pada sertifikat dan menyimpannya. Selanjutnya panitia memilih sertifikat yang akan dicetak dan terakhir panitia menekan tombol cetak untuk mencetak sertifikat. Alur tersebut digambarkan pada Gambar 5.3.



Gambar 5.4 Sequence Diagram Mencetak Absensi Peserta

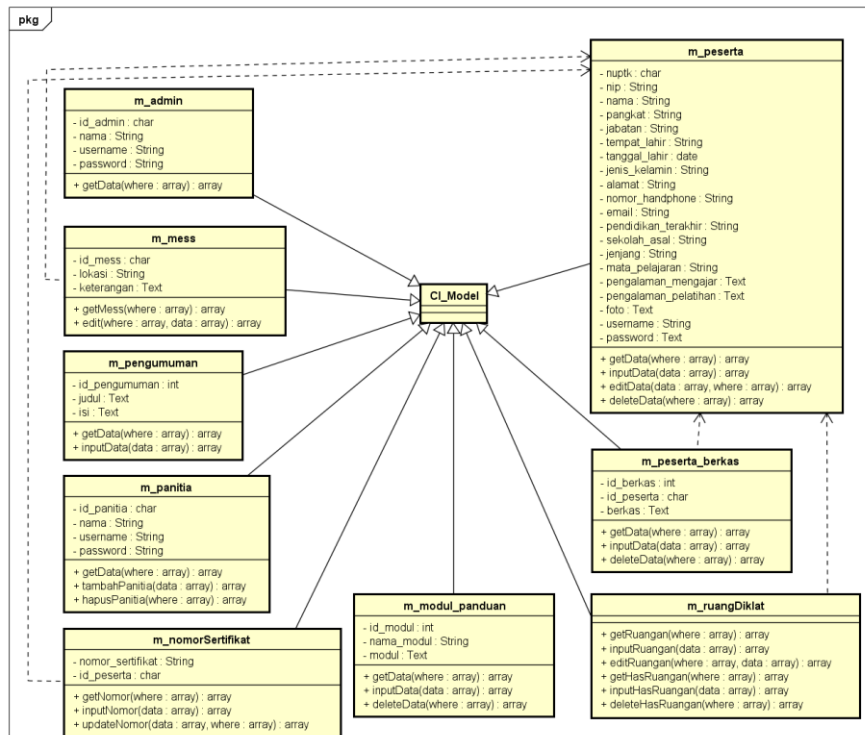
5.1.4 Sequence Diagram Mencetak Absensi Peserta

Sequence diagram mencetak absensi peserta merupakan proses yang dapat dijalankan oleh aktor panitia dimana panitia dapat mencetak absensi peserta dengan bantuan sistem. Proses ini dimulai dari sistem menampilkan daftar ruangan diklat yang tersedia, lalu paniti memilih ruangan mana yang absensinya akan dicetak. Setelah itu, sistem menampilkan absensi ruangan tersebut dan jika data peserta pada absensi tersebut tidak kosong, panitia dapat menekan tombol cetak untuk mencetak absensi. Namun jika data peserta pada absensi tersebut kosong, sistem akan menampilkan pesan data peserta pada ruangan tersebut masih kosong. Alur tersebut digambarkan pada Gambar 5.4.

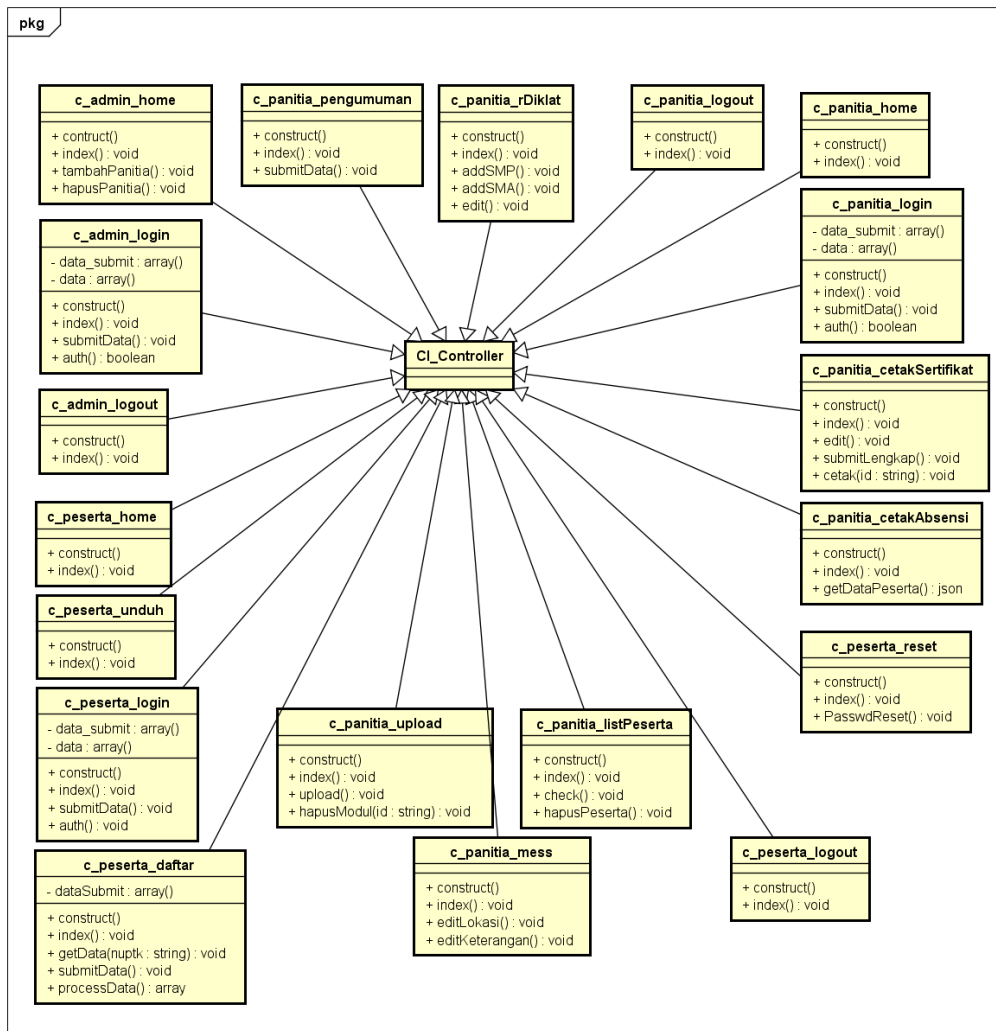
5.2 Perancangan Class Diagram

Class diagram adalah diagram yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Dalam pemodelan *class diagram* Sistem Informasi Manajemen Dikalt terdapat sembilan belas kelas yang berperan sebagai controller, dimana kelas-kelas tersebut merupakan *extend* dari kelas *CI_Controller*. Kelas-kelas tersebut digunakan dalam proses-proses yang dijalankan oleh masing-masing aktor. Aktor peserta terdapat enam kelas, yaitu *c_peserta_unduh*, *c_peserta_reset*, *c_peserta_logout*, *c_peserta_login*, *c_peserta_home*, dan *c_peserta_daftar*. Aktor panitia terdapat sepuluh kelas, yaitu *c_panitia_cetakAbsensi*, *c_panitia_cetakSertifikat*, *c_panitia_home*, *c_panitia_listPeserta*, *c_panitia_login*, *c_panitia_logout*, *c_panitia_mess*, *c_panitia_pengumuman*, *c_panitia_rDiklat*, dan *c_panitia_upload*. Sedangkan aktor admin hanya terdapat tiga kelas, yaitu *c_admin_home*, *c_admin_login*, dan *c_admin_logout*. Pemodelan tersebut digambarkan pada Gambar 5.6.

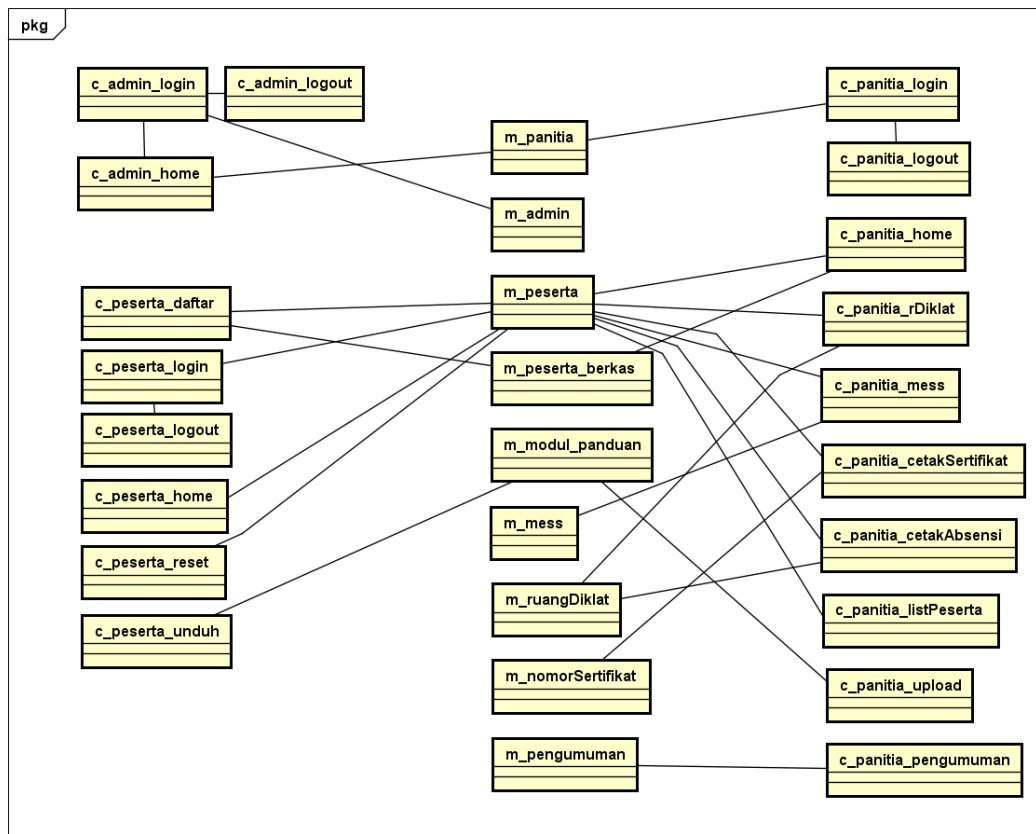
Selain kelas controller, dalam pemodelan *class diagram* Sistem Informasi Manajemen Dikalt terdapat sembilan kelas yang berperan sebagai model, dimana kelas-kelas tersebut merupakan *extend* dari kelas *CI_Model*. Tiap-tiap kelas tersebut merupakan digunakan dalam melakukan *query* ke dalam basis data. Kelas-kelas tersebut adalah *m_admin*, *m_mess*, *m_modul_panduan*, *m_nomorSertifikat*, *m_panitia*, *m_pengumuman*, *m_peserta*, *m_peserta_berkas*, dan *m_ruangDiklat*. Pemodelan tersebut digambarkan pada Gambar 5.5.



Gambar 5.5 Class Diagram Model



Gambar 5.6 Class Diagram Controller

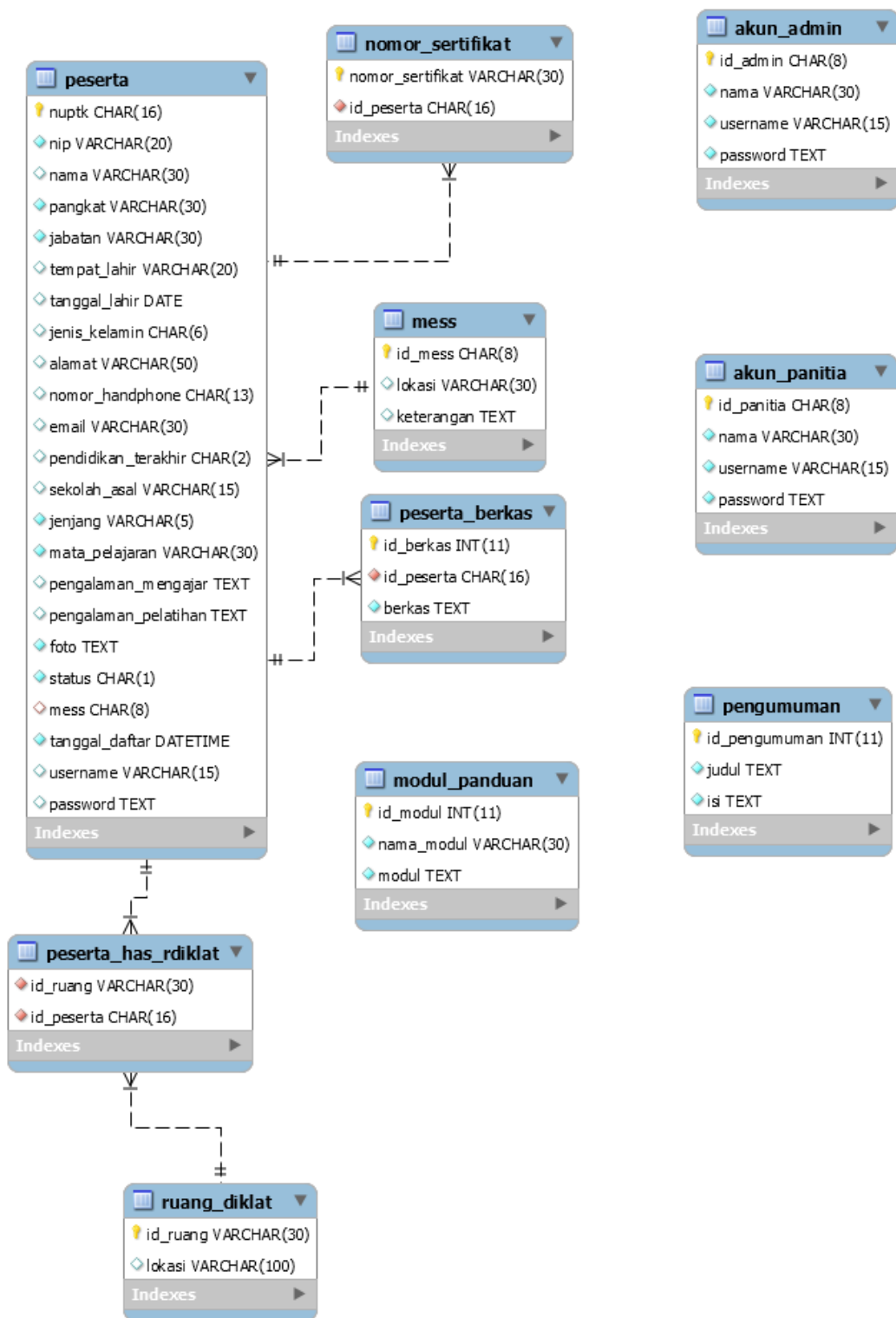


Gambar 5.7 Class Diagram Keseluruhan

Pada Gambar 5.7 merupakan gambar *class* diagram yang menjelaskan hubungan antar kelas secara keseluruhan baik itu kelas *controller* maupun kelas *model*. Setiap kelas dapat berinteraksi dengan lebih dari satu kelas yang lain. Kelas yang memiliki interaksi terbanyak adalah kelas *m_peserta*, dimana kelas tersebut berinteraksi dengan sepuluh kelas lain, seperti kelas *c_peserta_daftar*, *c_peserta_login*, *c_peserta_home*, *c_peserta_reset*, *c_panitia_home*, *c_panitia_rDiklat*, *c_panitia_mess*, *c_panitia_cetakSertifikat*, *c_panitia_cetakAbsensi*, dan *c_panitia_listPeserta*.

5.3 Perancangan *Physical Data Model*

Perancangan *physical data model* digunakan untuk memodelkan rancangan basis data yang dibutuhkan oleh sistem. Perancangan ini merancang semua struktur basis data mulai dari nama tabel, nama kolom, kedudukan kolom, tipe data pada kolom dan relasi tabel. Perancangan *physical data model* didapatkan dari perancangan *class* diagram dan pengamatan dokumen formulir pendaftaran pelatihan guru sasaran kurikulum 2013 (lampiran D). Pada Gambar 5.8 menggambarkan perancangan *physical data model* tersebut.



Gambar 5.8 Physical Data Model

5.4 Perancangan Komponen

Perancangan komponen menggambarkan Algoritme yang terdapat dalam suatu komponen perangkat lunak. Namun tidak semua komponen dalam perangkat lunak dijelaskan dalam perancangan ini, yang dijelaskan hanyalah proses pendaftaran. Proses pendaftaran dilakukan oleh kelas `c_peserta_daftar`, `m_peserta`, `m_peserta_berkas`, dan `m_ruangDiklat`. Dalam kelas-kelas tersebut terdapat beberapa fungsi yang digunakan dalam proses pendaftaran, yaitu fungsi `getData`, `submitData`, `processData`, `inputData`, `inputHasRuangan`.

5.4.1 Algoritme Fungsi `getData`

Tabel 5.1 Algoritme Fungsi `getData`

1	Mulai
2	Mengirimkan permintaan data peserta kepada aplikasi DAPODIK dan menyimpannya ke dalam variabel <code>\$getDataJson</code>
3	Merubah data yang diterima dari aplikasi DAPODIK yang sebelumnya dari bentuk json menjadi array dengan menggunakan fungsi <code>json_decode</code> dan disimpan ke dalam variabel <code>\$tmp</code>
4	Jika variabel <code>\$tmp</code> tidak sama dengan null
5	Isi dari variabel <code>\$tmp</code> disimpan ke dalam variabel array <code>\$data</code>
6	Membuat index 0, index label pada variabel array <code>\$data</code> tersebut berisikan data nuptk peserta dari index nuptk dari variabel <code>\$tmp</code>
7	Menampilkan seluruh data peserta yang sudah ditampung dalam variabel array <code>\$data</code> dalam bentuk string json
8	Lainnya
9	Menampilkan string dalam bentuk json yang berisikan null
10	Selesai

Fungsi `getData` pada Tabel 5.1 adalah fungsi yang digunakan untuk mendapatkan data peserta dari aplikasi DAPODIK dan kemudian disimpan dalam variabel tertentu untuk ditampilkan pada halaman pendaftaran. Fungsi ini diakses oleh `view` `daftar.php` ketika pengguna memasukkan NUPTK pada formulir pendaftaran.

5.4.2 Algoritme Fungsi submitData

Tabel 5.2 Algoritme Fungsi submitData

1	Mulai
2	Menyimpan data yang telah dimasukkan pengguna ke dalam atribut array dataSubmit
3	Mengatur konfigurasi pada library "upload" yang akan digunakan untuk mengunggah foto
4	Memanggil library "upload" dengan konfigurasi yang telah ditentukan sebelumnya
5	Melakukan inialisasi pada library "upload" yang telah dipanggil
6	Jika proses <i>upload</i> berhasil
7	Data upload disimpan dalam variabel \$tempData
8	Menambahkan data uplaod kedalam atribut dataSubmit
9	Lainnya
10	Menampilkan pesan "error" jika proses upload tidak berhasil
11	Memanggil fungsi processData dan disimpan ke dalam variabel \$dataHasRuangan
12	Menyimpan data peserta ke dalam database dengan mengakses fungsi inputData pada kelas m_peserta dengan atribut dataSubmit sebagai parameternya dan menyimpan kembalian dari fungsi tersebut jika terjadi kesalahan ke dalam variabel \$error[0]
13	Menyimpan hasil pengolahan data peserta yang didapatkan dari fungsi processData ke dalam database dengan mengakses fungsi inputHasRuangan dari kelas m_ruangDiklat dan menyimpan kembalian dari fungsi tersebut jika terjadi kesalahan ke dalam variabel \$error[1]
14	Jika variabel \$erorr[0] kosong
15	Melakukan proses <i>upload</i> data berkas kelengkapan peserta dan menampilkan pesan pendaftaran berhasil, username & password anda akan dikirimkan melalui email jika telah diverifikasi oleh panitia
16	Lainnya
17	Menampilkan pesan pendaftaran gagal anda telah terdaftar yang artinya proses penyimpanan data gagal dikarenakan data anda telah ada ke dalam sistem
18	Selesai

Fungsi submitData pada Tabel 5.2 adalah fungsi yang dijalankan oleh kelas *controller* `c_peserta_daftar` saat pengguna menekan tombol daftar pada halaman pendaftaran. Fungsi ini berfungsi untuk menampung data yang telah dimasukkan oleh peserta sekaligus mengunggah foto dan berkas yang diunggah oleh peserta.

5.4.3 Algoritme Fungsi ProcessData

Tabel 5.3 Algoritme Fungsi ProcessData

1	Mulai
2	Jika data jenjang mengajar yang dimasukkan pengguna adalah sd kelas 1
3	Maka id ruangan diklat yang akan disimpan adalah SD01
4	Jika data jenjang mengajar yang dimasukkan pengguna adalah sd kelas 2
5	Maka id ruangan diklat yang akan disimpan adalah SD02
6	Jika data jenjang mengajar yang dimasukkan pengguna adalah sd kelas 3
7	Maka id ruangan diklat yang akan disimpan adalah SD03
8	Jika data jenjang mengajar yang dimasukkan pengguna adalah sd kelas 4
9	Maka id ruangan diklat yang akan disimpan adalah SD04
10	Jika data jenjang mengajar yang dimasukkan pengguna adalah sd kelas 5
11	Maka id ruangan diklat yang akan disimpan adalah SD05
12	Jika data jenjang mengajar yang dimasukkan pengguna adalah sd kelas 6
13	Maka id ruangan diklat yang akan disimpan adalah SD06
14	Jika data jenjang mengajar yang dimasukkan pengguna adalah SMP
15	Maka akan memanggil id ruangan diklat yang telah disimpan sebelumnya di database sesuai mata pelajaran yang dimasukkan, dan id ruang diklat yang akan disimpan adalah SMP_data mata pelajaran yang dimasukkan
16	Jika id data ruangan yang disimpan ketika jenjang SMP kosong
17	Maka memasukkan data ruangan baru dengan id ruang SMP_mata pelajaran
18	Melakukan inisialisasi pada variabel dataHasRuangan untuk memasukkan data id_ruang yang telah diproses sebelumnya. Selanjutnya dataHasRuang tersebut akan dikeluarkan ketika fungsi ini dipanggil
19	Jika dataSubmit jenis kelamin "Pria"
20	Maka memanggil data lokasi mess peserta laki-laki dan menyimpannya dalam variabel mess
21	Menyimpan id mess yang didapatkan dari proses sebelumnya pada variabel id_mess
22	Lainnya jika jenis kelasmin dalam dataSubmit adalah "Wanita"
23	Maka memanggil data lokasi mess peserta perempuan dan menyimpannya dalam variabel mess
23	Menyimpan id mess yang didapatkan dari proses sebelumnya pada variabel id_mess
24	Mengatur kembalian fungsi yaitu variabel dataHasRuangan
25	Selesai

Fungsi `ProcessData` adalah fungsi yang dipanggil oleh fungsi `SubmitData` untuk melakukan pengolahan data peserta. Pengolahan tersebut meliputi penentuan ruangan diklat peserta dan penentuan lokasi *mess* peserta. Fungsi ini mengeluarkan id ruangan peserta yang telah ditentukan berdasarkan jenjang dan mata pelajaran yang dimasukkan oleh pengguna. Selain itu, ketika fungsi ini dijalankan juga menyimpan id *mess* pada data peserta sesuai hasil pengolahan yaitu sesuai jenis kelamin yang dimasukkan oleh pengguna. Algoritme fungsi ini dijelaskan pada Tabel 5.3.

5.4.4 Algoritme Fungsi inputData

Tabel 5.4 Algoritme Fungsi inputData

1	Mulai
2	Memanggil library <code>insert</code> yang disediakan oleh framework <code>codeigniter</code> dengan parameter string nama tabel pada <i>database</i> yang akan dimasukkan dan variabel data dalam bentuk array yang berisikan data-data yang akan dimasukkan
3	Mengatur kembalian fungsi yaitu berupa library <code>error</code> untuk mengetahui apakah terjadi error ketika fungsi ini dipanggil
4	Selesai

Fungsi `inputData` terdapat pada kelas `m_peserta` dan `m_peserta_berkas` dengan alur dan parameter yang sama. Fungsi `inputData` pada kelas `m_peserta` digunakan untuk memasukkan data peserta yang dimasukkan peserta pada proses pendaftaran dan fungsi `inputData` pada kelas `m_peserta_berkas` digunakan untuk memasukkan data berkas-berkas peserta yang dimasukkan dalam proses pendaftaran. Data berkas-berkas peserta yang akan dimasukkan dalam `inputData` kelas `m_peserta_berkas` hanyalah string lokasi dimana berkas-berkas tersebut disimpan di dalam sistem. Algoritme fungsi ini dijelaskan pada Tabel 5.4.

5.4.5 Algoritme Fungsi inputHasRuangan

Tabel 5.5 Algoritme Fungsi inputHasRuangan

1	Mulai
2	Memanggil library <code>insert</code> yang disediakan oleh framework <code>codeigniter</code> dengan parameter string <code>'peserta_has_rdiklat'</code> sebagai nama tabel pada <i>database</i> yang akan dimasukkan dan variabel data dalam bentuk array yang berisikan data-data yang akan dimasukkan
3	Mengatur kembalian fungsi yaitu berupa library <code>error</code> untuk mengetahui apakah terjadi error ketika fungsi ini dipanggil
4	Selesai

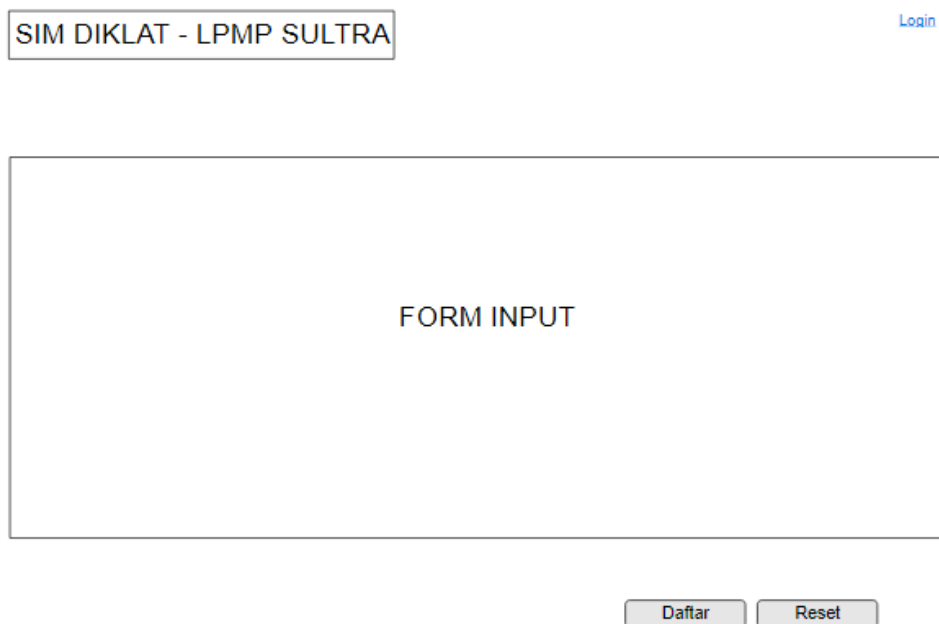
Fungsi `inputHasRuangan` terdapat pada kelas `m_ruangDiklat` yang digunakan untuk memasukkan hasil pengolahan data penentuan ruangan diklat peserta pada *database*. Data yang dimasukkan pada fungsi ini adalah id data peserta yang telah dimasukkan pada *database* dan id ruangan diklat yang telah ditentukan akan ditempati oleh peserta. Algoritme fungsi ini dijelaskan pada Tabel 5.5.

5.5 Perancangan Antarmuka

Perancangan antarmuka adalah menggambarkan desain kasar dari antarmuka sistem yang akan dikembangkan. Perancangan antarmuka Sistem Informasi Manajemen Diklat hanya menggambarkan antarmuka pada proses pendaftaran.

5.5.1 Halaman Daftar

Halaman pendaftaran adalah halaman yang pertama kali diakses oleh pengguna saat mengakses sistem. Halaman ini berfungsi untuk pengguna dapat melakukan pendaftaran dengan melengkapi data diri yang didapatkan dari aplikasi DADODIK. Perancangan antarmuka halaman daftar digambarkan pada Gambar 5.9.

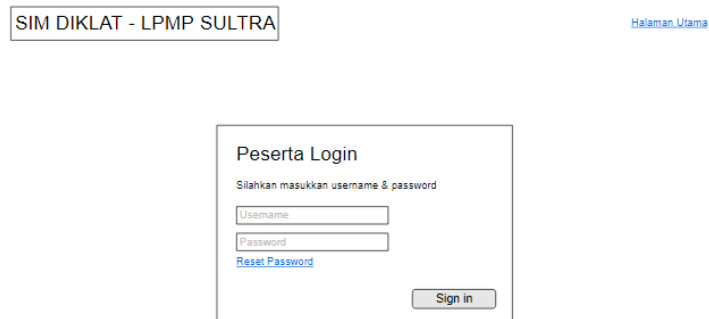


The image shows a wireframe of a registration page. At the top left, there is a header box containing the text "SIM DIKLAT - LPMP SULTRA". At the top right, there is a blue link labeled "Login". In the center of the page, there is a large rectangular box with the text "FORM INPUT" inside it. At the bottom of the page, there are two buttons: "Daftar" and "Reset".

Gambar 5.9 Perancangan Antarmuka Halaman Daftar

5.5.2 Halaman Login

Halaman login digunakan oleh pengguna untuk dapat masuk ke dalam sistem. Setelah pengguna masuk ke dalam sistem, pengguna kemudian menjadi peserta dan dapat melakukan beberapa proses selanjutnya seperti, melihat data peserta, mengubah data peserta, melihat lokasi ruang DIKLAT, dan melihat lokasi *mess*. Perancangan antarmuka halaman ini digambarkan pada Gambar 5.10.



SIM DIKLAT - LPMP SULTRA [Halaman Utama](#)

Peserta Login
Silahkan masukkan username & password

Username

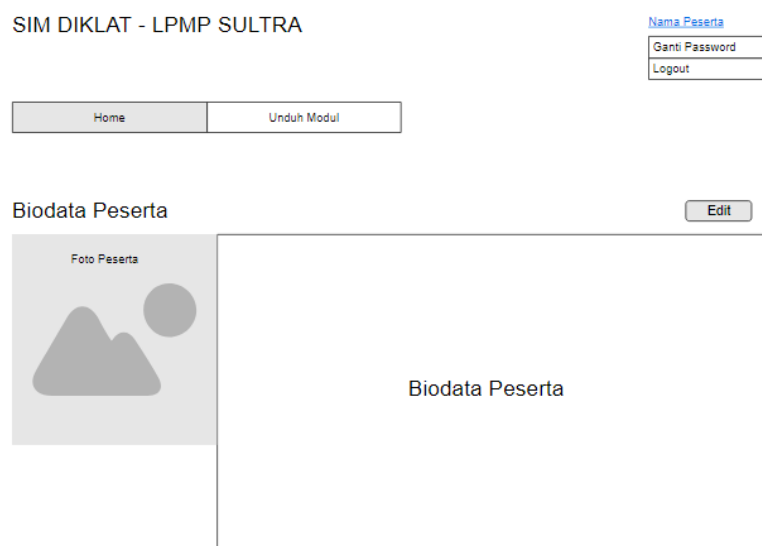
Password

[Reset Password](#)

Gambar 5.10 Perancangan Antarmuka Halaman Login

5.5.3 Halaman Home Peserta

Halaman home peserta adalah halaman yang pertama kali diakses peserta ketika telah melakukan login. Halaman tersebut menampilkan data peserta, foto peserta dan tombol untuk mengubah data peserta. Selain data peserta, halaman ini juga menampilkan lokasi ruangan diklat peserta, dan lokasi *mess* peserta yang didapatkan dari pengolahan data peserta pada proses pendaftaran. Perancangan antarmuka pada halaman ini digambarkan pada Gambar 5.11.



SIM DIKLAT - LPMP SULTRA [Nama Peserta](#)

Biodata Peserta

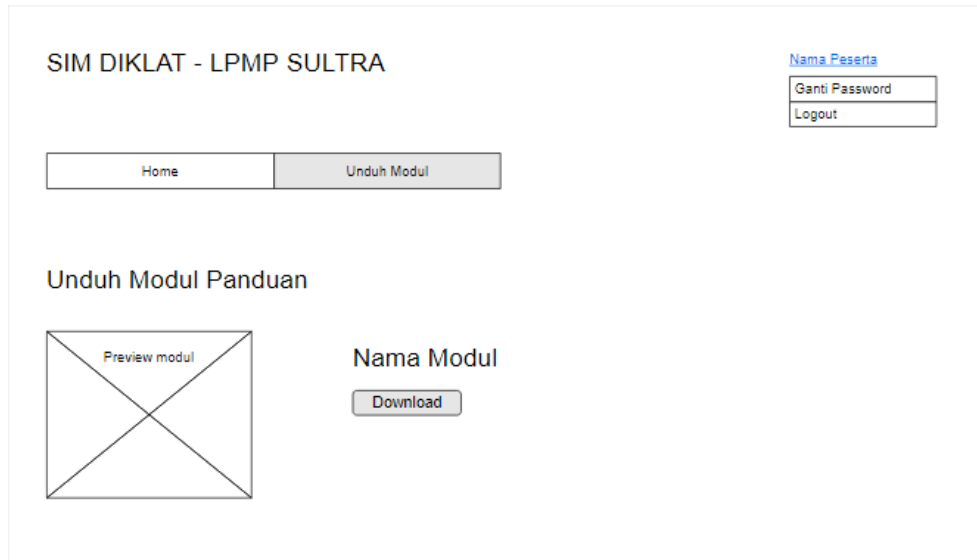
Foto Peserta

Biodata Peserta

Gambar 5.11 Perancangan Antarmuka Halaman Home Peserta

5.5.4 Halaman Unduh Modul Panduan

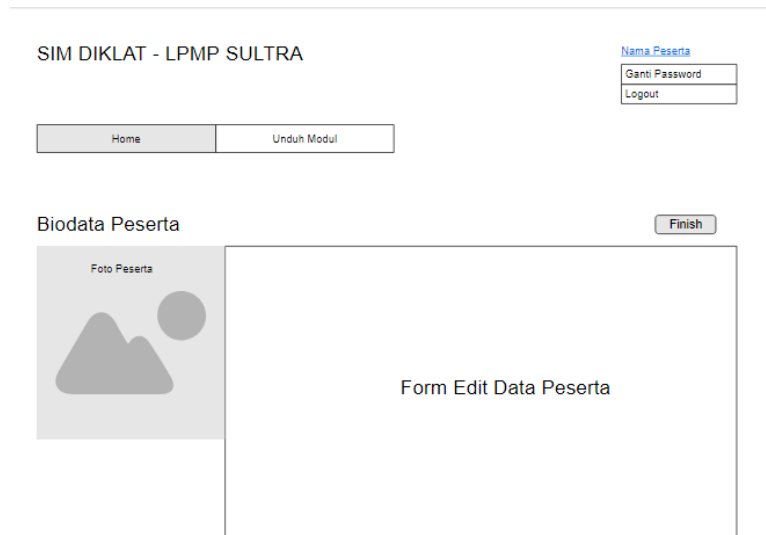
Halaman unduh modul panduan adalah halaman yang menampilkan *preview* dari kumpulan modul-modul panduan yang dapat di unduh oleh peserta. Dalam halaman tersebut terdapat tiga bagian utama, yaitu *preview* modul, nama modul yang akan diunduh, dan tombol *download* untuk mengunduh modul panduan. Perancangan antarmuka halaman ini digambarkan pada Gambar 5.12.



Gambar 5.12 Perancangan Antarmuka Halaman Unduh Modul Panduan

5.5.5 Halaman Edit Data Peserta

Halaman edit data peserta adalah halaman yang digunakan peserta untuk mengubah data yang telah dimasukkan dalam proses pendaftaran. Halaman ini disediakan oleh sistem untuk mencegah terjadinya kesalahan data peserta yang dimasukkan dalam proses pendaftaran. Data peserta yang dapat diubah adalah nomor *handphone*, *email*, pendidikan terakhir, sekolah asal, pengalaman mengajar, dan pengalaman pelatihan. Perancangan antarmuka pada halaman ini dijelaskan pada Gambar 5.13.



Gambar 5.13 Perancangan Antarmuka Halaman Edit Data Peserta

5.5.6 Halaman *Reset Password*

Halaman *reset password* digunakan oleh peserta untuk mengubah *password* yang didapatkan oleh sistem melalui *email*. *Password* dibuat oleh sistem secara acak sehingga sulit untuk diingat oleh peserta, sehingga halaman ini dibuat untuk memudahkan peserta mengingat *password* yang mereka buat sendiri. Perancangan antarmuka pada halaman ini digambarkan pada Gambar 5.14.



Gambar 5.14 Perancangan Antarmuka Halaman *Reset Password*