**Bab III**

**ANALISIS DAN PERANCANGAN**

Pengembangan perangkat lunak ini menggunakan metode *waterfall*, yang terdiri atas tahap studi literatur, analisis data, perancangan, implementasi, evaluasi, dan pemeliharaan. Pada bab ini akan dijelaskan secara rinci dua tahapan pengembangan, yaitu analisis dan perancangan.

**3.1 Analisis**

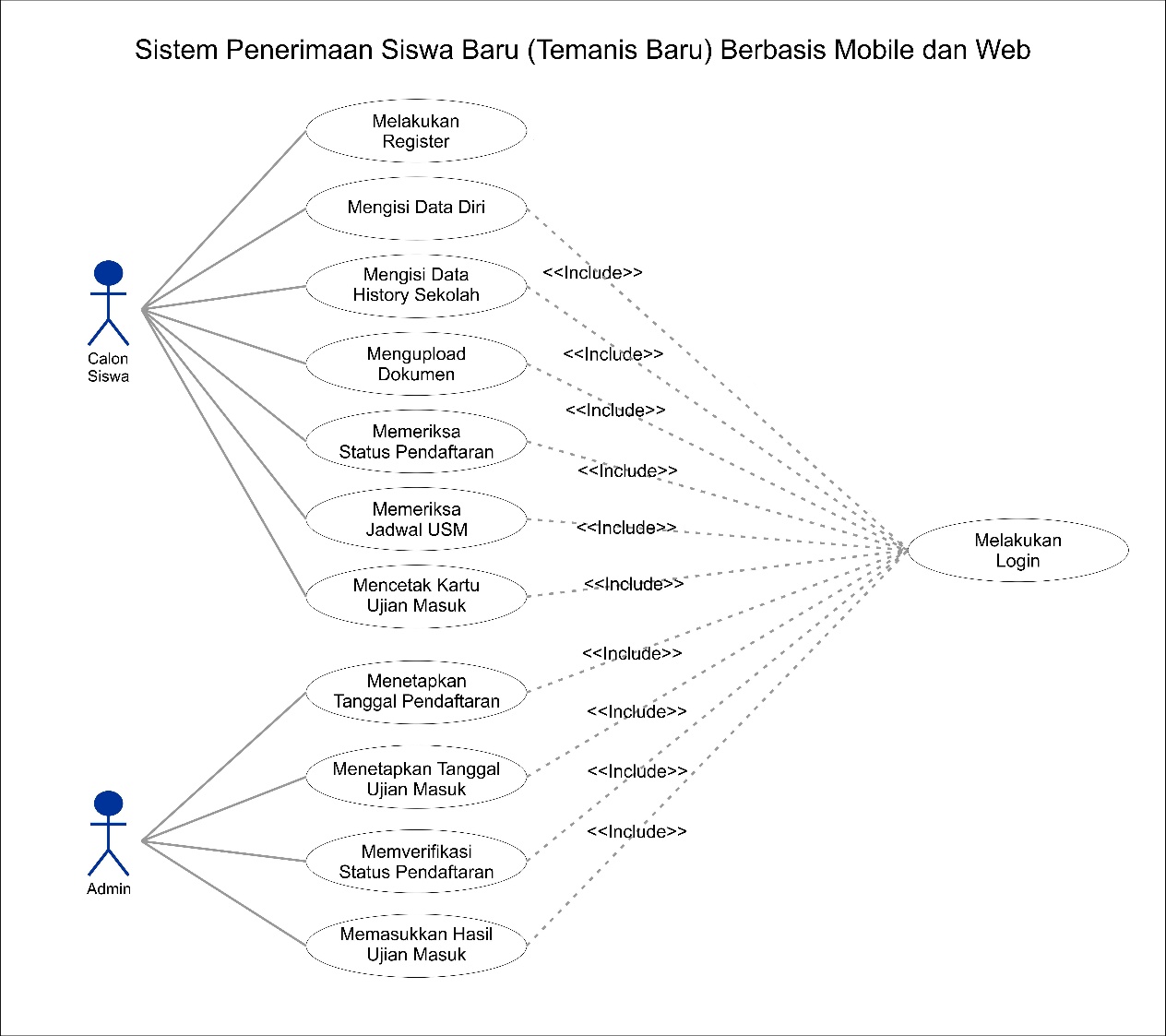
Pada subbab ini akan dijelaskan tahapan analisis yaitu analisis kebutuhan yang berupa analisis kebutuhan fungsional dengan menggunakan *use case diagram* dan analisis kebutuhan non fungsional. Berikut adalah penjelasan masing-masing analisis tersebut.

**3.1.1 Analisis Kebutuhan**

Analisis kebutuhan merupakan tahapan penting dalam pengembangan sistem untuk mengetahui secara detail sistem yang ingin dikembangkan. Kebutuhan sistem akan ditentukan oleh pemilik dan pengguna sistem. Analisis kebutuhan dibagi menjadi dua yaitu analisis kebutuhan fungsional yang merupakan spesifikasi inti mengenai hal-hal yang bisa dilakukan oleh sistem dan kebutuhan non fungsional yang merupakan komponen pendukung pada sistem.

**3.1.1.1 Analisis Kebutuhan Fungsional**

Analisis kebutuhan fungsional dapat digambarkan dengan menggunakan *use case diagram.*



Gambar 3.1 *Use Case Diagram*

Ada 2 aktor yang terlibat dalam sistem, yaitu calon siswa dan admin. Tabel 3.1 adalah definisi 2 aktor yang terlibat dalam sistem.

|  |  |  |
| --- | --- | --- |
| **No.** | **Aktor** | **Deskripsi** |
| 1. | Calon siswa | Aktor yang melakukan pendaftaran ke sekolah |
| 2. | Admin | Aktor yang memverifikasi pendaftaran dari calon siswa |

Tabel 3.1 Definisi Aktor

*Use case* menyatakan fungsi lengkap yang dilakukan oleh aktor dan menghasilkan nilai ke aktor serta menggambarkan di mana *level user* dan respon.

Tabel 3.2 menyatakan definisi *use case* pada Gambar 3.1.

|  |  |  |
| --- | --- | --- |
| **No.** | **Use Case** | **Definisi** |
| 1. | Melakukan *register* | Melakukan pembuatan akun baru agar terdaftar dalam sistem |
| 2. | Melakukan *login* | Melakukan *login* untuk masuk ke dalam sistem |
| 3. | Mengisi data diri | Melakukan pengisian terhadap data diri dari calon siswa yang bersangkutan |
| 4. | Mengisi data *history* sekolah | Melakukan pengisian terhadap data dari sekolah sebelumnya |
| 5. | Meng-*upload* dokumen | Melakukan *upload* dokumen yang dibutuhkan untuk proses pendaftaran |
| 6. | Memeriksa status pendaftaran | Melakukan pemeriksaan terhadap status pendaftaran |
| 7. | Memeriksa jadwal ujian saringan masuk | Melakukan pemeriksaan jadwal untuk mengikuti ujian saringan masuk |
| 8. | Mencetak kartu ujian masuk | Melakukan pencetakan kartu ujian masuk sebagai syarat mengikuti ujian |
| 9. | Menetapkan tanggal pendaftaran | Melakukan penetapan tanggal buka dan tutup pendaftaran |
| 10. | Memverifikasi status pendaftaran | Melakukan verifikasi terhadap *status form* pendaftaran calon siswa |
| 11. | Memasukkan hasil ujian masuk | Melakukan *input* hasil ujian masuk terkait kelulusan calon siswa |
| 12. | Menetapkan tanggal ujian saringan masuk | Melakukan penetapan tanggal ujian saringan masuk untuk calon siswa |

Tabel 3.2 Definisi *Use Case*

Selanjutnya adalah membuat skenario (*flow of event)* yang menggambarkan urutan interaksi aktor dengan sistem dari *use case.*

Tabel 3.3 menunjukkan skenario *use case* membuat akun baru

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nomor** | 1 |
| **Nama** | Membuat akun baru |
| **Tujuan** | Untuk melakukan pembuatan akun baru agar terdaftar dalam sistem |
| **Deskripsi** | |
| **Aktor** | Calon siswa |
| **Skenario Utama** | |
| **Kondisi Awal** | Tampilan halaman *register* |
| **Aksi Aktor** | **Reaksi Sistem** |
| Mengisi *email, password &* konfirmasi *password* | Sistem akan menampilkan kotak *email, password &* konfirmasi *password* yang harus diisi agar dapat terdaftar dalam sistem |
| Mencentang tombol “Saya bukan robot” | Sistem akan menampilkan 9 pilihan gambar yang harus dipilih secara benar sesuai dengan permintaan sistem |
| Menekan tombol daftar | Sistem akan memproses dan menyimpan data calon siswa ke dalam *database* |
| **Kondisi Akhir** | Tampilan halaman *login* |

Tabel 3.3 Skenario *use case* membuat akun baru

Tabel 3.4 menunjukkan skenario *use case* melakukan *login*

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nomor** | 2 |
| **Nama** | Melakukan *login* |
| **Tujuan** | Untuk masuk ke dalam sistem |
| **Deskripsi** | |
| **Aktor** | Calon siswa |
| **Skenario Utama** | |
| **Kondisi Awal** | Tampilan halaman login |
| **Aksi Aktor** | **Reaksi Sistem** |
| Mengisi *email & password* | Sistem akan menampilkan kotak *email & password* yang harus diisi dengan benar agar dapat masuk ke dalam sistem |
| Menekan tombol masuk | Sistem akan memproses dan melakukan pengecekan pada *database* |
| **Kondisi Akhir** | Tampilan halaman beranda calon siswa |

Tabel 3.4 Skenario *use case* melakukan *login*

Tabel 3.5 menunjukkan skenario *use case* mengisi data diri

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nomor** | 3 |
| **Nama** | Mengisi data diri |
| **Tujuan** | Untuk melakukan pengisian terhadap data diri |
| **Deskripsi** | |
| **Aktor** | Calon siswa |
| **Skenario Utama** | |
| **Kondisi Awal** | Tampilan halaman *form* pendaftaran (*tab* profil) |
| **Aksi Aktor** | **Reaksi Sistem** |
| Mengisi data diri | Sistem akan menampilkan kotak berisi identitas data diri yang harus dilengkapi |
| Menekan tombol *save* | Sistem akan memproses dan menyimpan data-data yang telah diisi |
| **Kondisi Akhir** | Tampilan halaman *form* pendaftaran (*tab* profil) |

Tabel 3.5 Skenario *use case* mengisi data diri

Tabel 3.6 menunjukkan skenario *use case* mengisi *data* *history* sekolah

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nomor** | 4 |
| **Nama** | Mengisi data *history* sekolah |
| **Tujuan** | Untuk melakukan pengisian terhadap data dari sekolah sebelumnya |
| **Deskripsi** | |
| **Aktor** | Calon siswa |
| **Skenario Utama** | |
| **Kondisi Awal** | Tampilan halaman *form* pendaftaran (*tab* tingkatan) |
| **Aksi Aktor** | **Reaksi Sistem** |
| Mengisi data *history* sekolah | Sistem akan menampilkan kotak berisi data *history* sekolah sebelumnya yang harus dilengkapi |
| Menekan tombol *save* | Sistem akan memproses dan menyimpan data-data yang telah diisi |
| **Kondisi Akhir** | Tampilan halaman *form* pendaftaran (*tab* tingkatan) |

Tabel 3.6 Skenario *use case* mengisi *data* *history* sekolah

Tabel 3.7 menunjukkan skenario *use case* meng*-upload* dokumen

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nomor** | 5 |
| **Nama** | Meng-*upload* dokumen |
| **Tujuan** | Melakukan *upload* dokumen yang dibutuhkan untuk proses pendaftaran |
| **Deskripsi** | |
| **Aktor** | Calon siswa |
| **Skenario Utama** | |
| **Kondisi Awal** | Tampilan halaman *form* pendaftaran (*tab* dokumen) |
| **Aksi Aktor** | **Reaksi Sistem** |
| Meng-*upload* dokumen | Sistem akan menampilkan kotak dimana *user* dapat meng-*upload* dokumen yang dibutuhkan |
| Menekan tombol *save* | Sistem akan memproses dan menyimpan data-data yang telah di-*upload* |
| Memeriksa status pendaftaran | Sistem akan menampilkan status siswa apakah *verified, pending, passed, failed* |
| **Kondisi Akhir** | Tampilan halaman *form* pendaftaran (*tab* dokumen) |

Tabel 3.7 Skenario *use case* meng*-upload* dokumen

Tabel 3.8 menunjukkan skenario *use case* memeriksa status pendaftaran

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nomor** | 6 |
| **Nama** | Memeriksa status pendaftaran |
| **Tujuan** | Untuk melakukan pemeriksaan terhadap status pendaftaran |
| **Deskripsi** | |
| **Aktor** | Calon siswa |
| **Skenario Utama** | |
| **Kondisi Awal** | Tampilan halaman *form* pendaftaran |
| **Aksi Aktor** | **Reaksi Sistem** |
| Memeriksa status pendaftaran | Sistem akan menampilkan status siswa apakah *verified, pending, passed, failed* |
| **Kondisi Akhir** | Tampilan halaman *form* pendaftaran |

Tabel 3.8 Skenario *use case* memeriksa status pendaftaran

Tabel 3.9 menunjukkan skenario *use case* memeriksa jadwal ujian saringan masuk

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nomor** | 7 |
| **Nama** | Memeriksa jadwal ujian saringan masuk |
| **Tujuan** | Untuk melakukan pemeriksaan jadwal untuk mengikuti ujian saringan masuk |
| **Deskripsi** | |
| **Aktor** | Calon siswa |
| **Skenario Utama** | |
| **Kondisi Awal** | Tampilan halaman ujian saringan masuk |
| **Aksi Aktor** | **Reaksi Sistem** |
| Memeriksa jadwal ujian masuk | Sistem akan menampilkan tanggal untuk jadwal ujian masuk |
| Mengklik tombol cetak | Sistem akan mencetak kartu ujian siswa |
| **Kondisi Akhir** | Tampilan halaman ujian saringan masuk |

Tabel 3.9 Skenario *use case* memeriksa jadwal ujian saringan masuk

Tabel 3.10 menunjukkan skenario *use case* mencetak kartu ujian masuk

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nomor** | 8 |
| **Nama** | Mencetak kartu ujian masuk |
| **Tujuan** | Untuk melakukan pencetakan kartu ujian masuk sebagai syarat mengikuti ujian |
| **Deskripsi** | |
| **Aktor** | Calon siswa |
| **Skenario Utama** | |
| **Kondisi Awal** | Tampilan halaman ujian saringan masuk |
| **Aksi Aktor** | **Reaksi Sistem** |
| Mengklik tombol cetak | Sistem akan mencetak kartu ujian siswa |
| **Kondisi Akhir** | Tampilan halaman ujian saringan masuk |

Tabel 3.10 Skenario *use case* mencetak kartu ujian masuk

Tabel 3.11 menunjukkan skenario *use case* menetapkan tanggal pendaftaran

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nomor** | 9 |
| **Nama** | Menetapkan tanggal pendaftaran |
| **Tujuan** | Melakukan penetapan tanggal buka dan tutup pendaftaran |
| **Deskripsi** | |
| **Aktor** | Admin |
| **Skenario Utama** | |
| **Kondisi Awal** | Tampilan halaman buka/tutup pendaftaran |
| **Aksi Aktor** | **Reaksi Sistem** |
| Menetapkan tanggal buka pendaftaran | Sistem akan menampilkan kalender yang akan dipilih admin sebagai tanggal pembukaan pendaftaran |
| Menetapkan tanggal tutup pendaftaran | Sistem akan menampilkan kalender yang akan dipilih admin sebagai batas tanggal pendaftaran |
| **Kondisi Akhir** | Tampilan halaman buka/tutup pendaftaran |

Tabel 3.11 Skenario *use case* menetapkan tanggal pendaftaran

Tabel 3.12 menunjukkan skenario *use case* memverifikasi status pendaftaran

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nomor** | 10 |
| **Nama** | Memverifikasi status pendaftaran |
| **Tujuan** | Melakukan verifikasi terhadap status form pendaftaran calon siswa |
| **Deskripsi** | |
| **Aktor** | Admin |
| **Skenario Utama** | |
| **Kondisi Awal** | Tampilan halaman verifikasi pendaftar |
| **Aksi Aktor** | **Reaksi Sistem** |
| Mengklik nama calon siswa untuk diverifikasi | Sistem akan menampilkan data calon siswa beserta dokumen yang telah diupload untuk diverifikasi |
| Memasukkan hasil ujian masuk dengan mengklik *pass / fail* | Sistem akan mengubah status murid sesuai dengan pilihan admin |
| **Kondisi Akhir** | Tampilan halaman verifikasi pendaftar |

Tabel 3.12 Skenario *use case* memverifikasi status pendaftaran

Tabel 3.13 menunjukkan skenario *use case* memasukkan hasil ujian masuk

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nomor** | 11 |
| **Nama** | Memasukkan hasil ujian masuk |
| **Tujuan** | Melakukan *input* hasil ujian masuk terkait kelulusan calon siswa |
| **Deskripsi** | |
| **Aktor** | Admin |
| **Skenario Utama** | |
| **Kondisi Awal** | Tampilan halaman verifikasi pendaftar |
| **Aksi Aktor** | **Reaksi Sistem** |
| Memasukkan hasil ujian masuk dengan mengklik *pass / fail* | Sistem akan mengubah status murid sesuai dengan pilihan admin |
| **Kondisi Akhir** | Tampilan halaman verifikasi pendaftar |

Tabel 3.13 Skenario *use case* memasukkan hasil ujian masuk

Tabel 3.14 menunjukkan skenario *use case* menetapkan tanggal ujian saringan masuk

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nomor** | 12 |
| **Nama** | Menetapkan tanggal ujian saringan masuk |
| **Tujuan** | Melakukan penetapan tanggal ujian saringan masuk untuk calon siswa |
| **Deskripsi** | |
| **Aktor** | Admin |
| **Skenario Utama** | |
| **Kondisi Awal** | Tampilan halaman ujian saringan masuk |
| **Aksi Aktor** | **Reaksi Sistem** |
| Menetapkan tanggal ujian saringan masuk | Sistem akan menampilkan kalender yang akan dipilih admin sebagai tanggal ujian saringan masuk |
| **Kondisi Akhir** | Tampilan halaman ujian saringan masuk |

Tabel 3.14 Skenario *use case* menetapkan tanggal ujian saringan masuk

**3.1.1.2 Analisis Kebutuhan Non Fungsional**

Adapun kebutuhan non fungsional dari sistem ini adalah tampilan/*design* yang tetap konsisten/sama walaupun dijalankan di berbagai *browser* seperti Mozilla Firefox, Google Chrome, Torch.

**3.2. Perancangan**

Tahapan perancangan akan merincikan gambaran sistem sebelum memasuki tahap implementasi. Perancangan pada TEMANIS BARU mencakup perancangan sistem usulan dengan *sequence diagram*. Kemudian perancangan tampilan ditampilan dengan *mockup* sedangkan perancangan basis data dengan ERD (Entity Relationship Diagram).

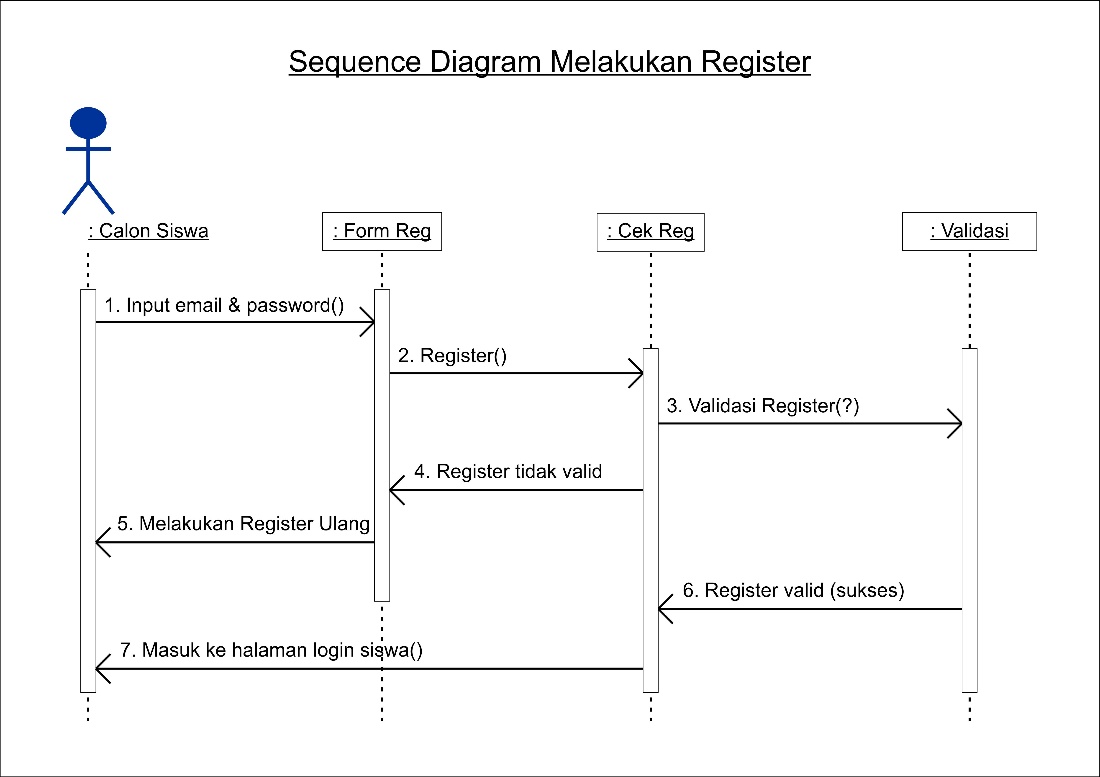
**3.2.1 Perancangan Sistem Usulan**

Perancangan sistem usulan dari TEMANIS BARU dapat digambarkan dengan *sequence diagram*. *Sequence diagram* adalah suatu diagram yang menggambarkan interaksi antar obyek dan mengindikasikan komunikasi diantara obyek-obyek tersebut.

Berikut *sequence diagram* dari sistem penerimaan siswa baru:

1. ***Sequence Diagram* Melakukan *Register***

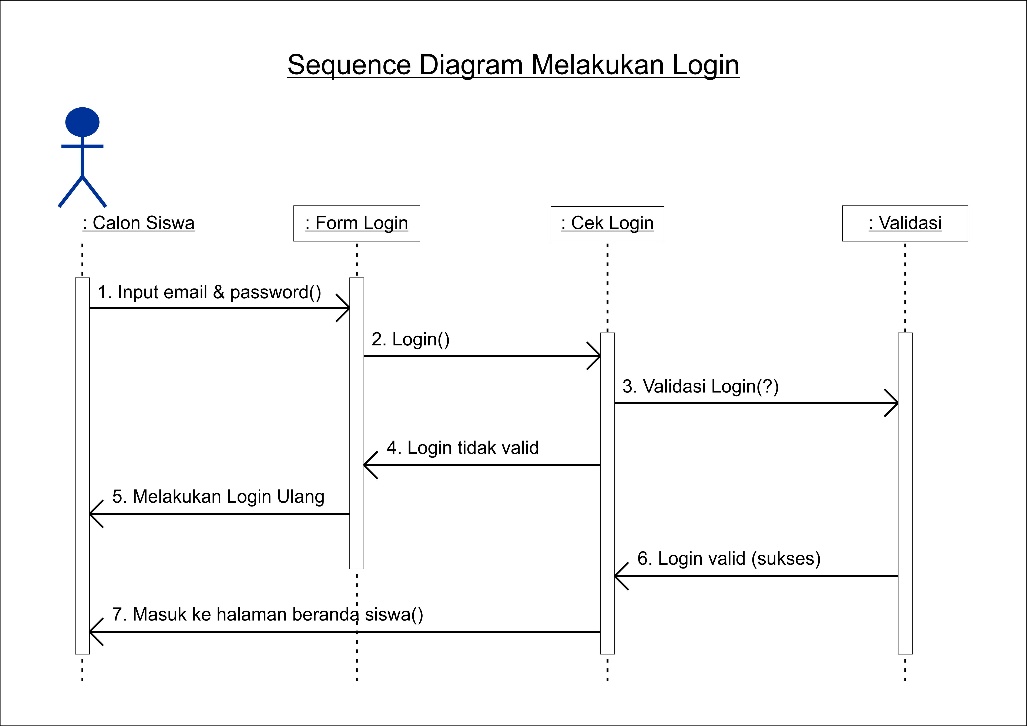
Gambar 3.2 menunjukkan alur dari *sequence diagram* melakukan *register* bagi calon siswa. Calon siswa melakukan r*egister* dengan mengisi *email* dan *password*. Apabila informasi yang diisi tidak *valid*, sistem akan meminta calon siswa untuk mengisi ulang sedangkan apabila *valid*, calon siswa akan diarahkan ke halaman *login*.



Gambar 3.2 *Sequence Diagram* Melakukan *Register*

1. ***Sequence Diagram* Melakukan *Login***

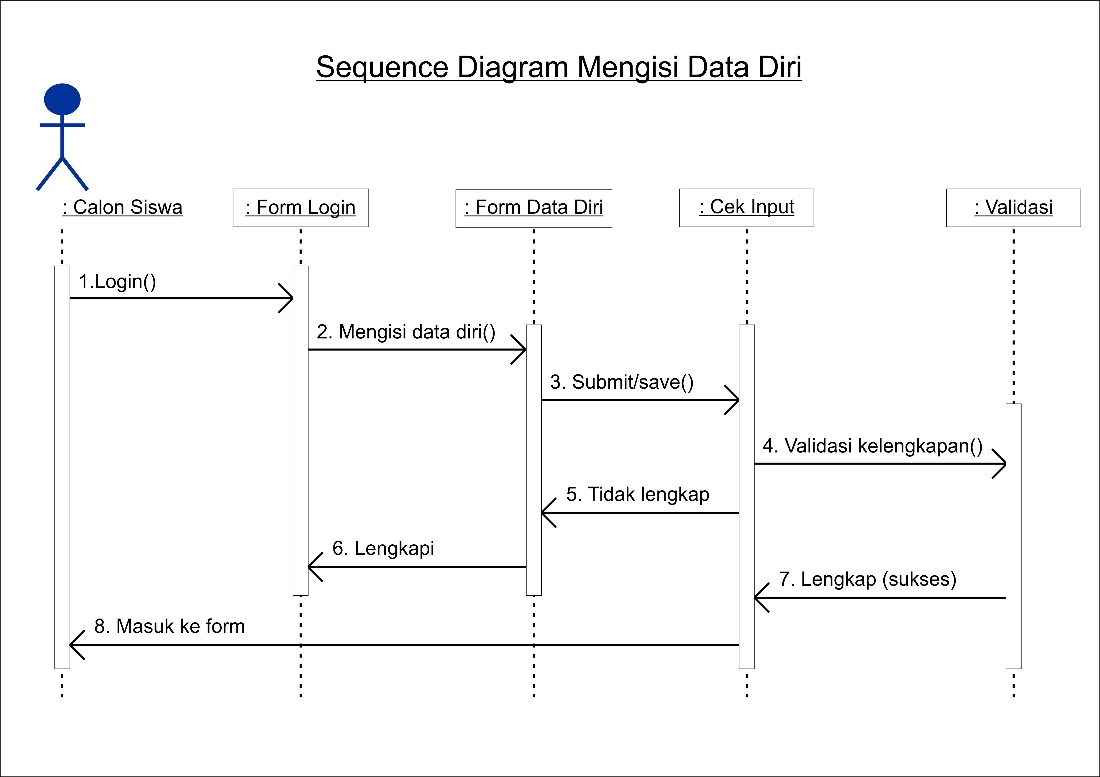
Gambar 3.3 menunjukkan alur dari *sequence diagram* melakukan *login* bagi calon siswa. Calon siswa yang telah melakukan *register* dapat melakukan *login* dengan mengisi *email* dan *password*. Apabila informasi yang diisi tidak *valid*, sistem akan meminta calon siswa untuk mengisi ulang sedangkan apabila *valid*, calon siswa akan diarahkan ke halaman beranda siswa.



Gambar 3.3 *Sequence Diagram* Melakukan *Login*

1. ***Sequence Diagram* Mengisi Data Diri**

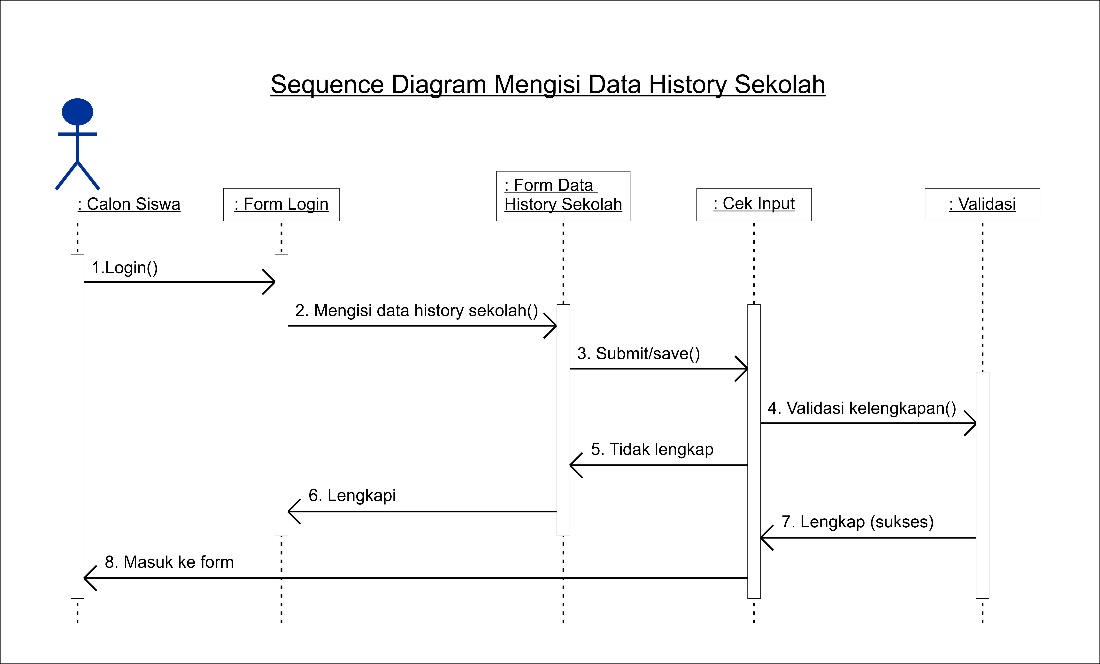
Gambar 3.4 menunjukkan alur dari *sequence diagram* mengisi data diri bagi calon siswa. Calon siswa dapat mengisi data diri dengan *login* terlebih dahulu kemudian memilih fitur Pendaftaran dan masuk ke dalam *tab* Profil (data diri). Apabila semua informasi wajib belum diisi lengkap, sistem akan meminta calon siswa untuk melengkapinya sedangkan apabila telah lengkap, tahap ini pun telah selesai.



Gambar 3.4 *Sequence Diagram* Mengisi Data Diri

1. ***Sequence Diagram* Mengisi *Data History* Sekolah**

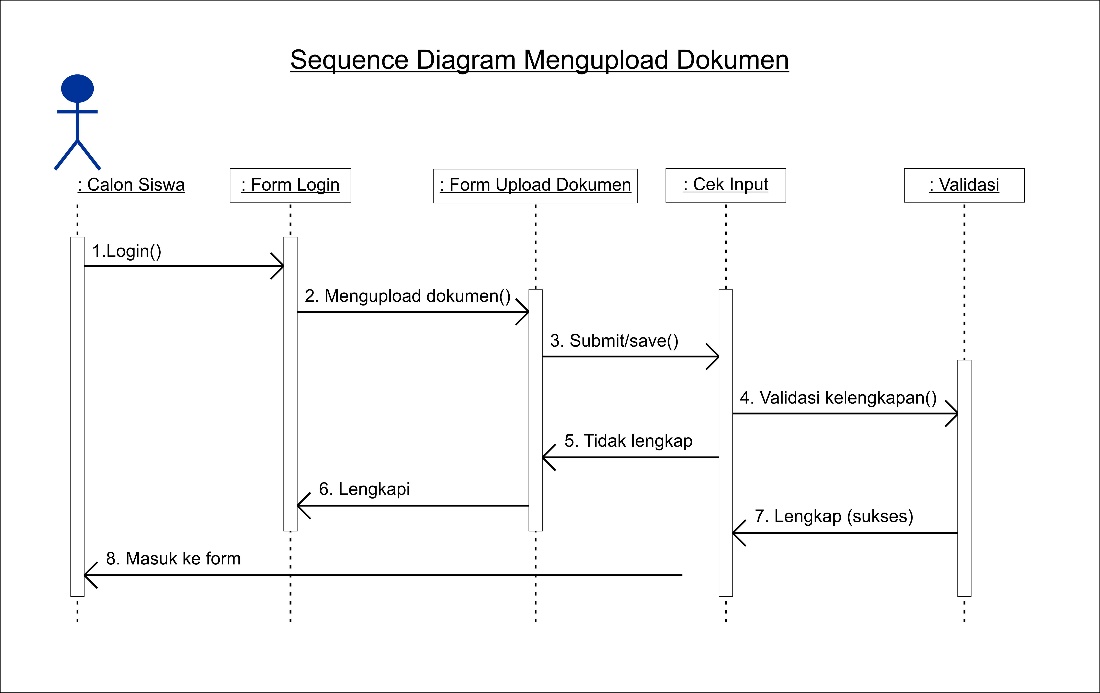
Gambar 3.5 menunjukkan alur dari *sequence diagram* mengisi data *history* sekolah bagi calon siswa. Calon siswa dapat mengisi data *history* sekolah dengan *login* terlebih dahulu kemudian memilih fitur Pendaftaran dan masuk ke dalam *tab* tingkatan (*history* sekolah). Apabila semua informasi wajib belum diisi lengkap, sistem akan meminta calon siswa untuk melengkapinya sedangkan apabila telah lengkap, tahap ini pun telah selesai.



Gambar 3.5 *Sequence Diagram* Mengisi Data *History* Sekolah

1. ***Sequence Diagram* Meng-*upload* Dokumen**

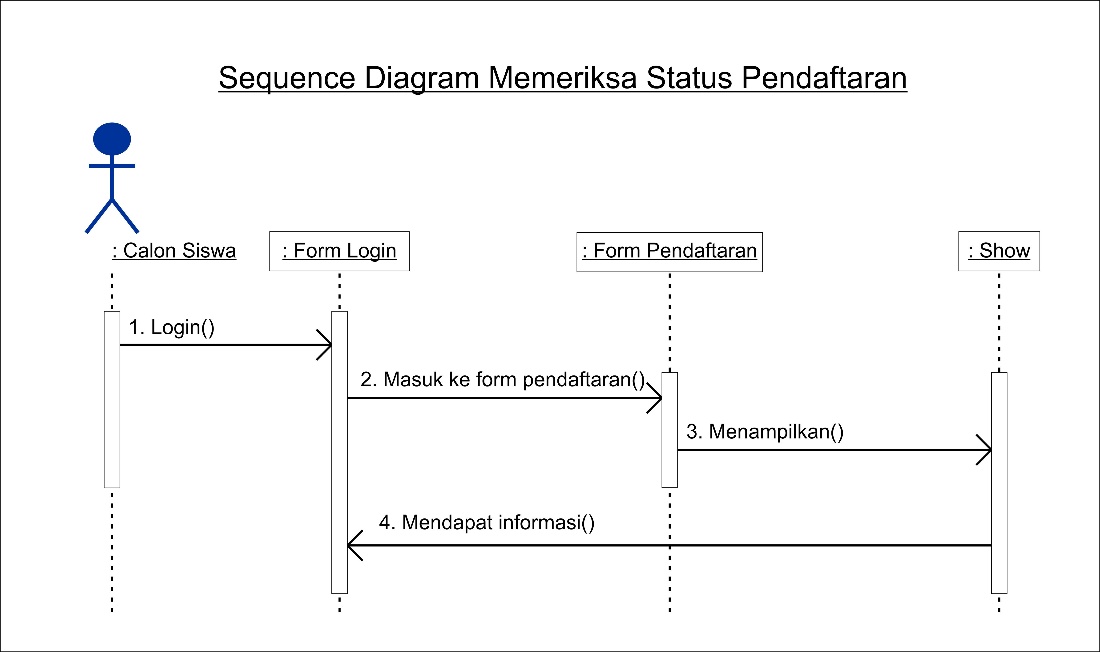
Gambar 3.6 menunjukkan alur dari *sequence diagram* meng-*upload* dokumen bagi calon siswa. Calon siswa dapat meng-*upload* dokumen dengan *login* terlebih dahulu kemudian memilih fitur Pendaftaran dan masuk ke dalam *tab* Dokumen (*upload* dokumen). Apabila semua dokumen wajib belum di-*upload*, sistem akan meminta calon siswa untuk melengkapinya sedangkan apabila telah lengkap, tahap ini pun telah selesai.



Gambar 3.6 *Sequence Diagram* Meng-*upload* Dokumen

1. ***Sequence Diagram* Memeriksa Status Pendaftaran**

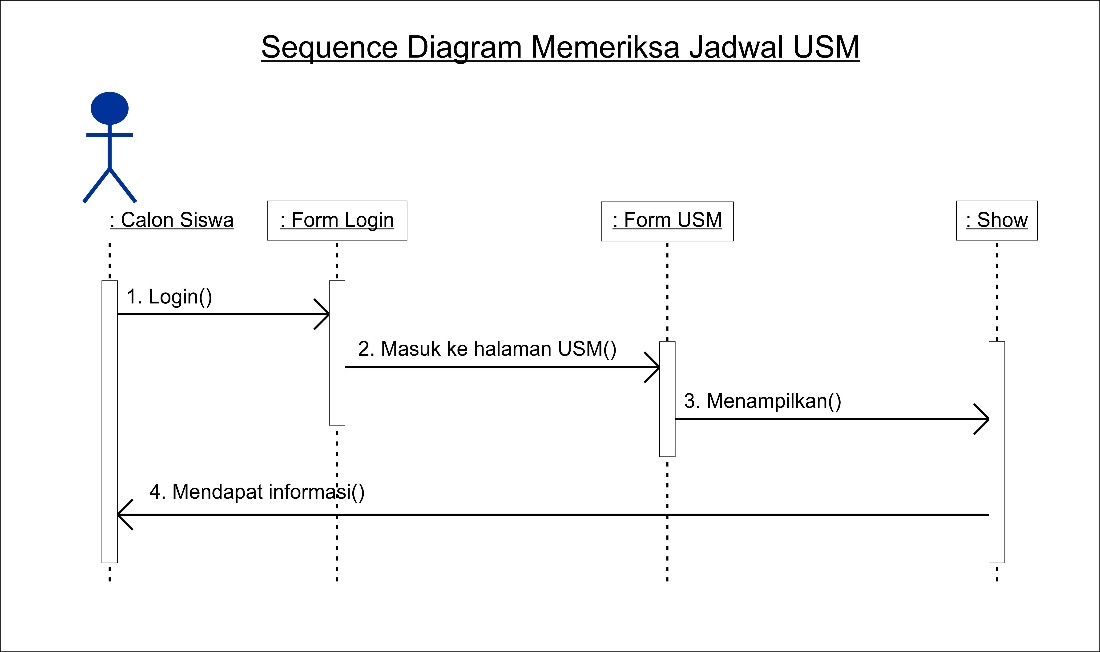
Gambar 3.7 menunjukkan alur dari *sequence diagram* memeriksa status pendaftaran calon siswa. Calon siswa dapat memeriksa status pendaftaran dengan *login* terlebih dahulu kemudian memilih fitur Pendaftaran. Di dalam *form* Pendaftaran pada setiap tab baik itu profil, tingkatan maupun dokumen, pada bagian bawah akan terdapat status pendaftaran calon siswa.



Gambar 3.7 *Sequence Diagram* Memeriksa *Status* Pendaftaran

1. ***Sequence Diagram* Memeriksa Jadwal Ujian Masuk**

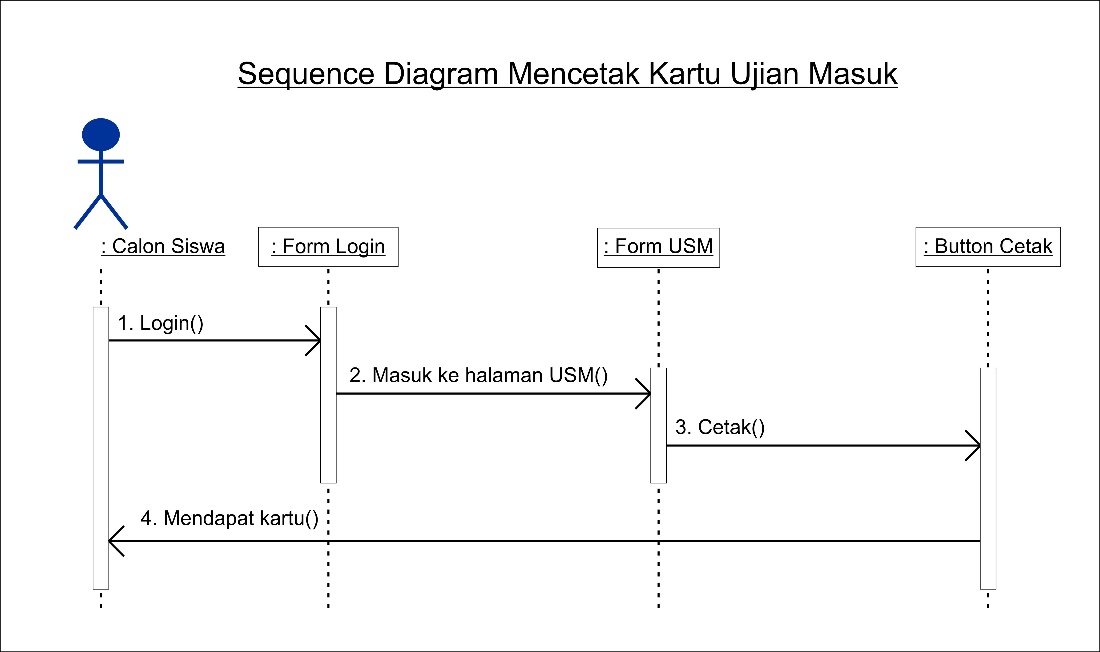
Gambar 3.8 menunjukkan alur dari *sequence diagram* memeriksa jadwal ujian saringan masuk calon siswa. Calon siswa dapat memeriksa jadwal ujian saringan masuk dengan *login* terlebih dahulu kemudian masuk ke dalam *form* USM. Di dalam form USM, akan ditampilkan jadwal lengkap ujian saringan masuk



Gambar 3.8 *Sequence Diagram* Memeriksa Jadwal Ujian Masuk

1. ***Sequence Diagam* Mencetak Kartu Ujian Masuk**

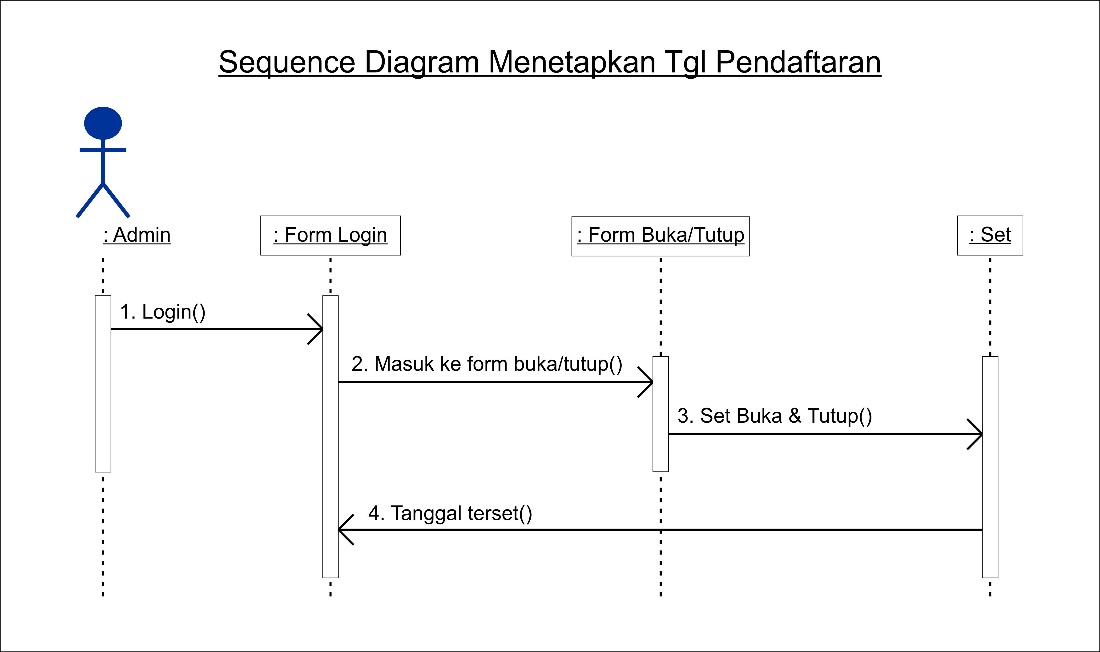
Gambar 3.9 menunjukkan alur dari *sequence diagram* mencetak kartu ujian masuk calon siswa. Calon siswa dapat mencetak kartu ujian masuk dengan *login* terlebih dahulu kemudian masuk ke dalam *form* USM. Di dalam *form* USM, terdapat fitur untuk mencetak kartu USM.



Gambar 3.9 *Sequence Diagram* Mencetak Kartu Ujian Masuk

1. ***Sequence Diagram* Menetapkan Tanggal Pendaftaran**

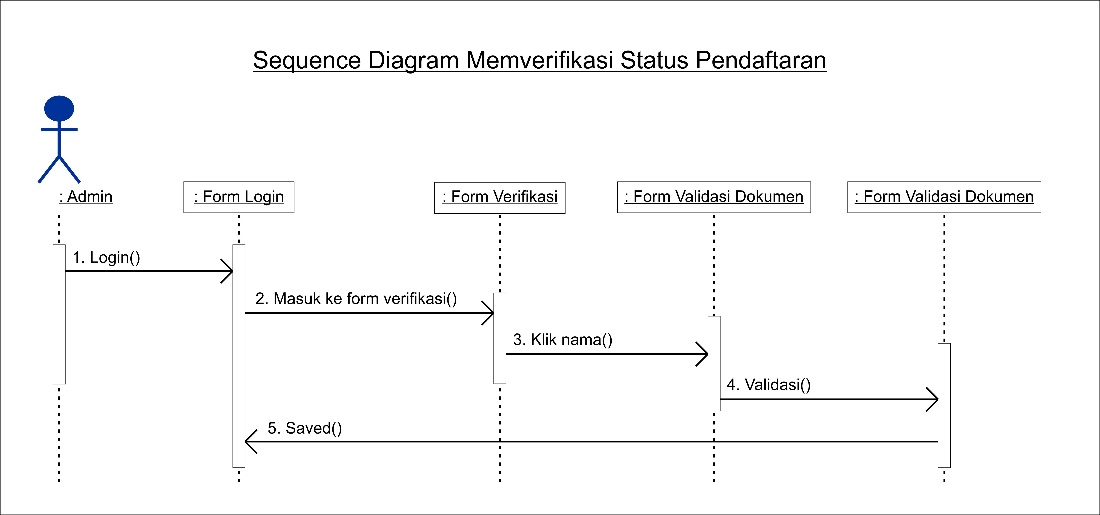
Gambar 3.10 menunjukkan alur dari *sequence diagram* menetapkan tanggal pendaftaran. Admin dapat menetapkan tanggal buka maupun tanggap tutup pendaftaran dengan *login* terlebih dahulu kemudian masuk ke dalam *form* Buka/Tutup Pendaftaran. Di dalam form ini akan tersedia fitur kalender untuk menetapkan tanggal buka dan tanggal tutup pendaftaran.



Gambar 3.10 *Sequence Diagram* Menetapkan Tanggal Pendaftaran

1. ***Sequence Diagram* Memverifikasi Status Pendaftaran**

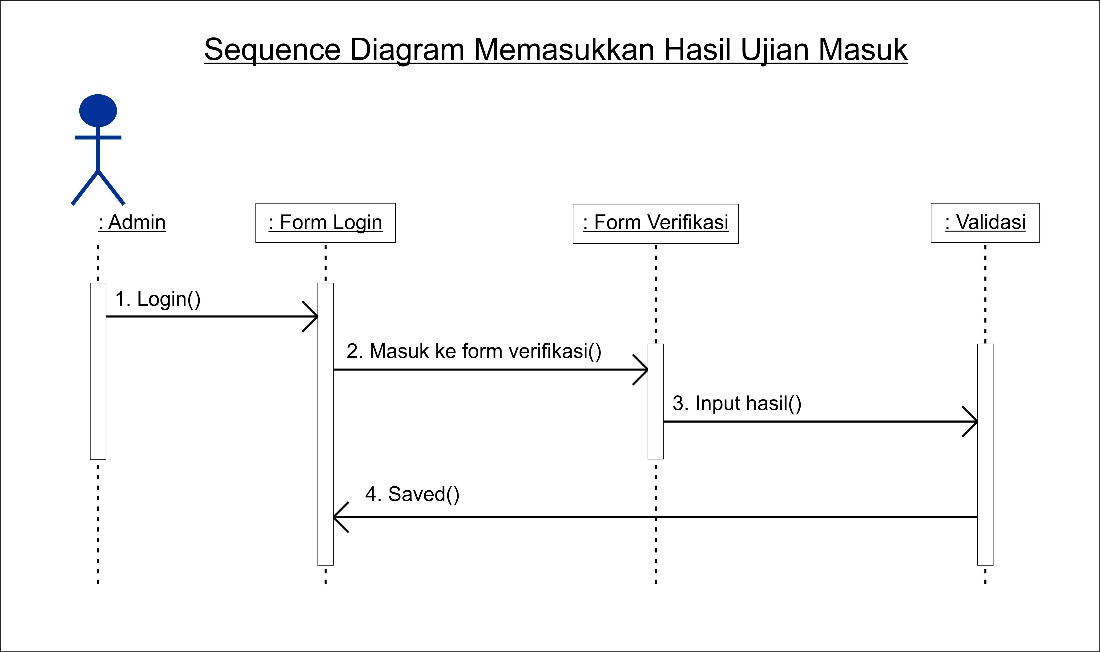
Gambar 3.11 menunjukkan alur dari *sequence diagram* meverifikasi status pendaftaran. Admin dapat memverifikasi status pendaftaran dengan *login* terlebih dahulu kemudian masuk ke dalam *form* Verifikasi Pendaftar. Di dalam form ini terdapat tabel daftar calon siswa dan admin dapat memverifikasi status pendaftaran dengan mengklik nama calon siswa yang akan diarahkan ke dalam form informasi calon siswa.



Gambar 3.11 *Sequence Diagram* Memverifikasi *Status* Pendaftaran

1. ***Sequence Diagram* Memasukkan Hasil Ujian Masuk**

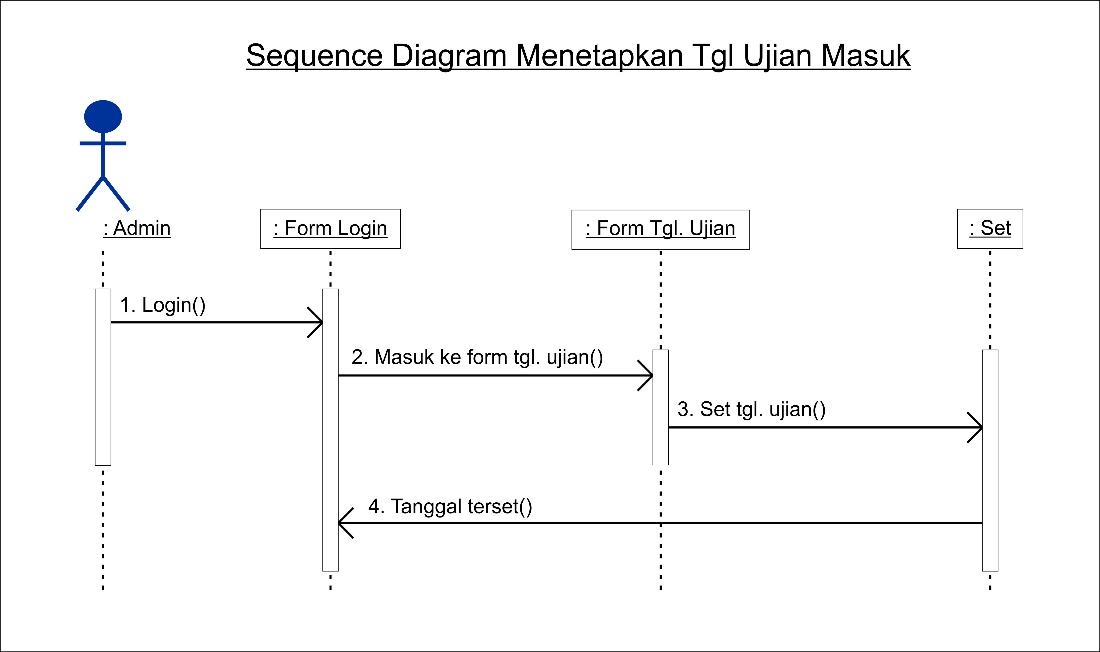
Gambar 3.12 menunjukkan alur dari *sequence diagram* memasukkan hasil ujian masuk. Admin dapat memasukkan hasil ujian masuk dengan *login* terlebih dahulu kemudian masuk ke dalam *form* Verifikasi Pendaftar. Di dalam form ini terdapat tabel daftar calon siswa dan admin dapat memasukkan hasil ujian masuk dengan mengubah status siswa baik *Passed* maupun *Failed*.



Gambar 3.12 *Sequence Diagram* Memasukkan Hasil Ujian Masuk

1. ***Sequence Diagram* Menetapkan Tanggal Ujian Masuk**

Gambar 3.13 menunjukkan alur dari *sequence diagram* menetapkan tanggal ujian masuk. Admin dapat menetapkan tanggal ujian masuk dengan *login* terlebih dahulu kemudian masuk ke dalam USM. Di dalam form ini akan tersedia fitur kalender untuk menetapkan tanggal buka ujian masuk.



**Gambar 3.13 *Sequence Diagram* Menetapkan Tanggal Ujian Masuk**

**3.2.2 Perancangan Tampilan**

Tampilan aplikasi TEMANIS BARU yang akan dikembangkan terdiri dari dua bagian perancangan yaitu Perancangan Tampilan Calon Siswa dan Perancangan Tampilan Admin.

**3.2.2.1 Perancangan Tampilan Calon Siswa**

Perancangan antar muka halaman utama calon siswa memiliki beberapa fungsi yang dapat dilakukan oleh calon siswa yang telah login yaitu:

1. Pendaftaran: merupakan halaman yang berisi *form* pendaftaran yang harus dilengkapi oleh calon siswa. *Form* pendaftaran ini terdiri dari 3 buah *tabs* yaitu mengisi data diri, mengisi data *history* sekolah dan meng-*upload* dokumen.
2. Ujian Saringan Masuk: merupakan halaman yang menampilkan jadwal ujian saringan masuk serta calon siswa dapat mencetak kartu ujian masuk di dalam halaman ini.

**3.2.2.2 Perancangan Tampilan Admin**

Perancangan antar muka halaman utama admin memiliki beberapa fungsi yang dapat dilakukan oleh admin yang telah login yaitu:

1. Buka/Tutup Pendaftaran: merupakan halaman dimana admin menetapkan tanggap buka pendaftaran maupun batas akhir pendaftaran.
2. Verifikasi Pendaftar: merupakan halaman yang digunakan admin untuk memverifikasi form pendaftaran setiap calon siswa serta memasukkan hasil ujian masuk.
3. Ujian Saringan Masuk: merupakan halaman dimana admin menetapkan tanggal ujian saringan masuk untuk calon siswa.

**3.2.3 Perancangan Antar Muka**

Perancangan antar muka merupakan tahapan untuk membuat tampilan atau *design* dari sistem yang akan dibuat. Rancangan tampilan yang dibuat meliputi rancangan struktur *menu*, rancangan *input* dan rancangan *output* dari sistem yang akan dibuat.

**3.2.3.1 Perancangan Tampilan Halaman Depan**

Gambar 3.14a dan Gambar 3.14b menunjukkan perancangan tampilan halaman depan dari sistem. Gambaran besar mengenai sistem TEMANIS BARU terlihat dari halaman depan.



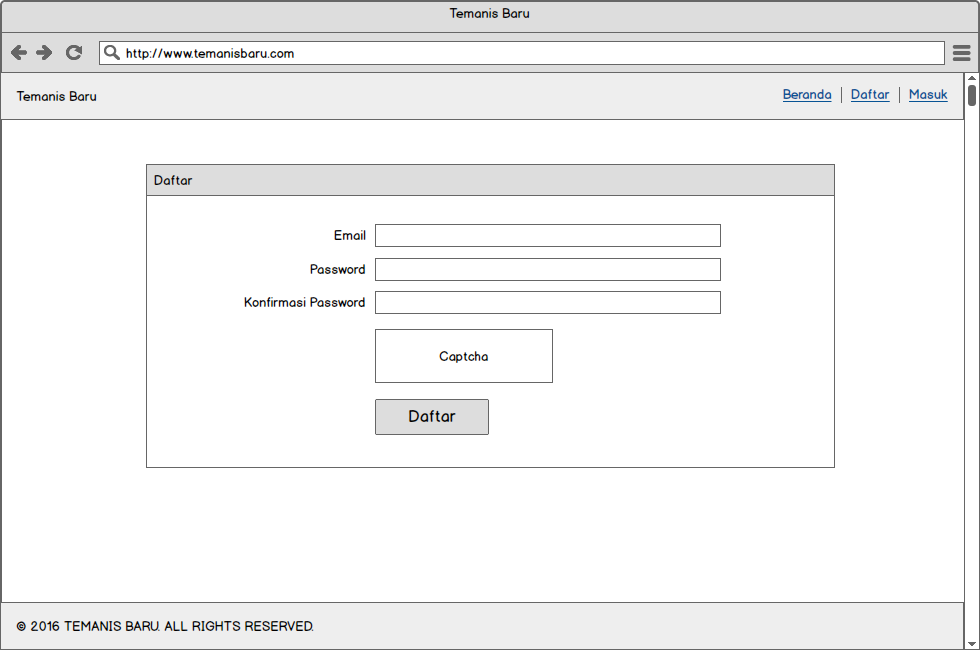
Gambar 3.14a Perancangan Tampilan Halaman Depan



Gambar 3.14b Perancangan Tampilan Halaman Depan

**3.2.3.2 Perancangan Tampilan Halaman *Register***

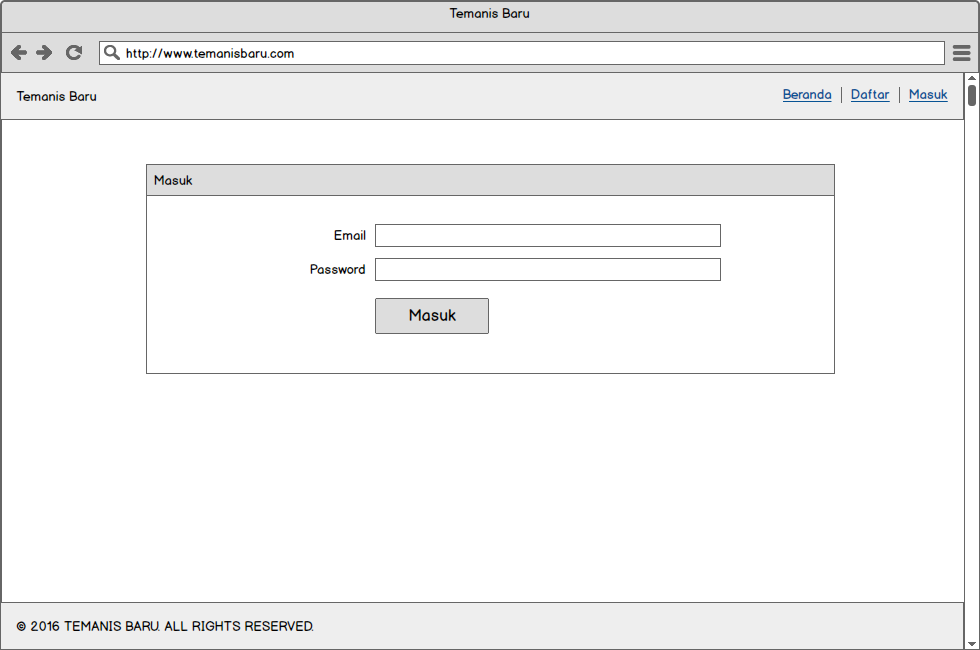
Gambar 3.15 menunjukkan perancangan tampilan halaman *register.* Agar dapat terdaftar dalam sistem, calon siswa hanya perlu mengisi *email*, *password* dan konfirmasi *password* serta menyelesaikan *captcha* untuk membuktikan calon siswa bukanlah robot.



Gambar 3.15 Perancangan Tampilan Halaman *Register*

**3.2.3.3 Perancangan Tampilan Halaman Login**

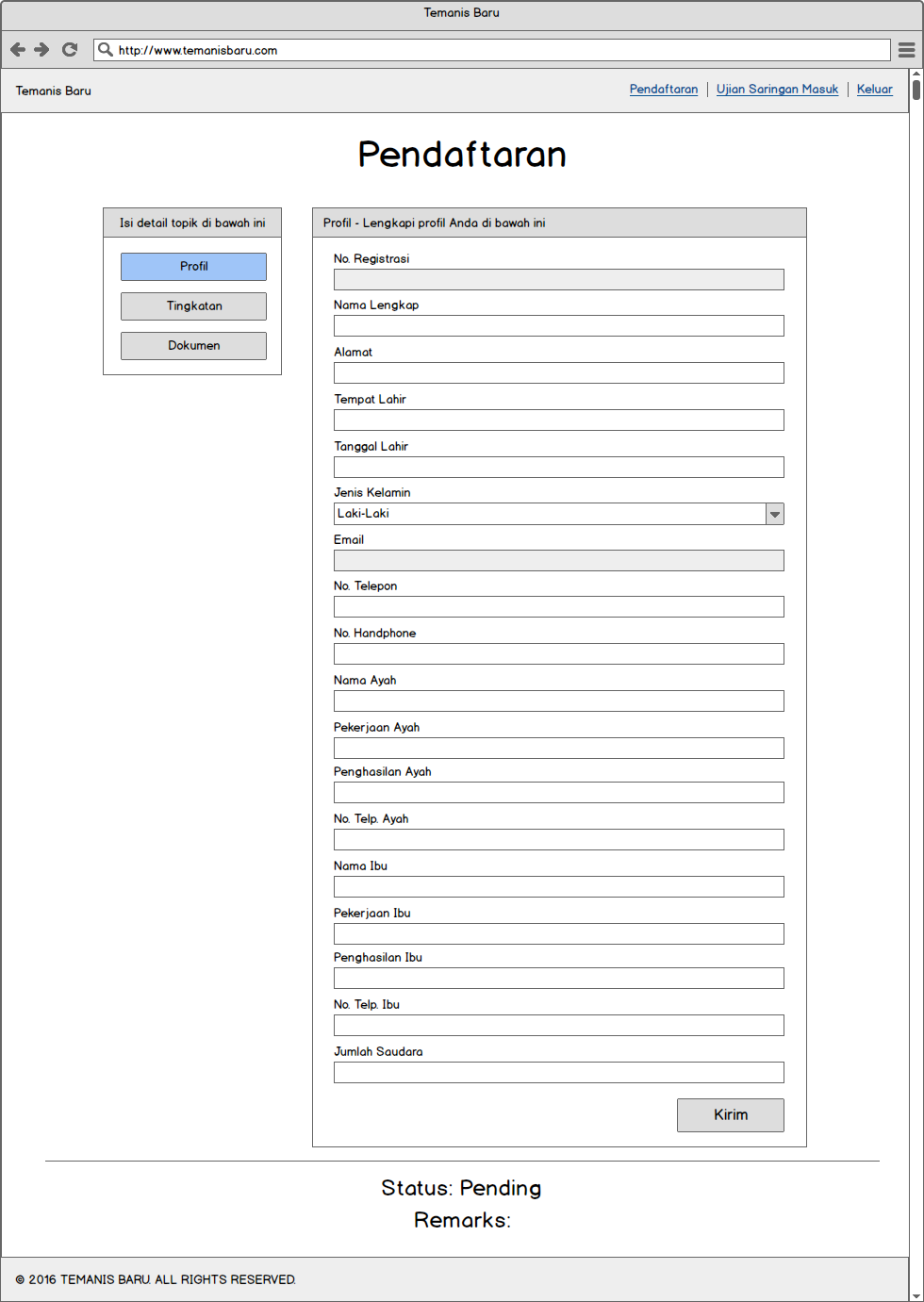
Gambar 3.16 menunjukkan perancangan tampilan halaman *login.* Calon siswa dapat masuk ke dalam sistem dengan mengisi *email* dan *password* secara benar.



Gambar 3.16 Perancangan Tampilan Halaman *Login*

**3.2.3.4 Perancangan Tampilan Halaman Mengisi Data Diri**

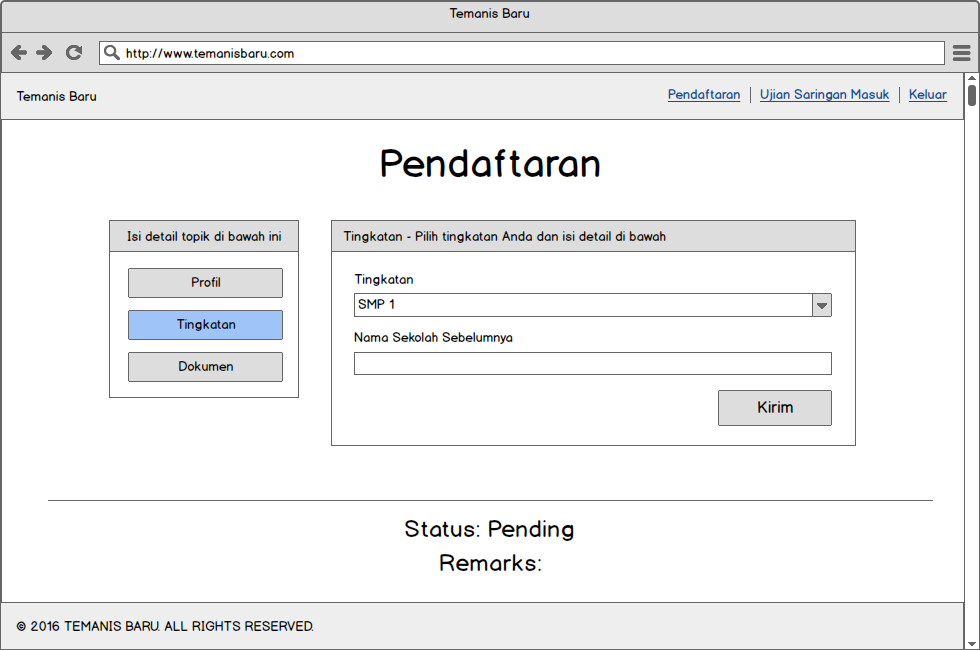
Gambar 3.17 menunjukkan perancangan tampilan halaman mengisi data diri (*tab* profil). *Form* profil merupakan *form* yang berisi data diri calon siswa yang harus diisi secara benar dan lengkap kecuali untuk nomor registrasi (*auto generate* dari sistem) dan *email* (diambil dari *database* ketika registrasi).



Gambar 3.17 Perancangan Tampilan Halaman Mengisi Data Diri

**3.2.3.5 Perancangan Tampilan Halaman Mengisi Data *History* Sekolah**

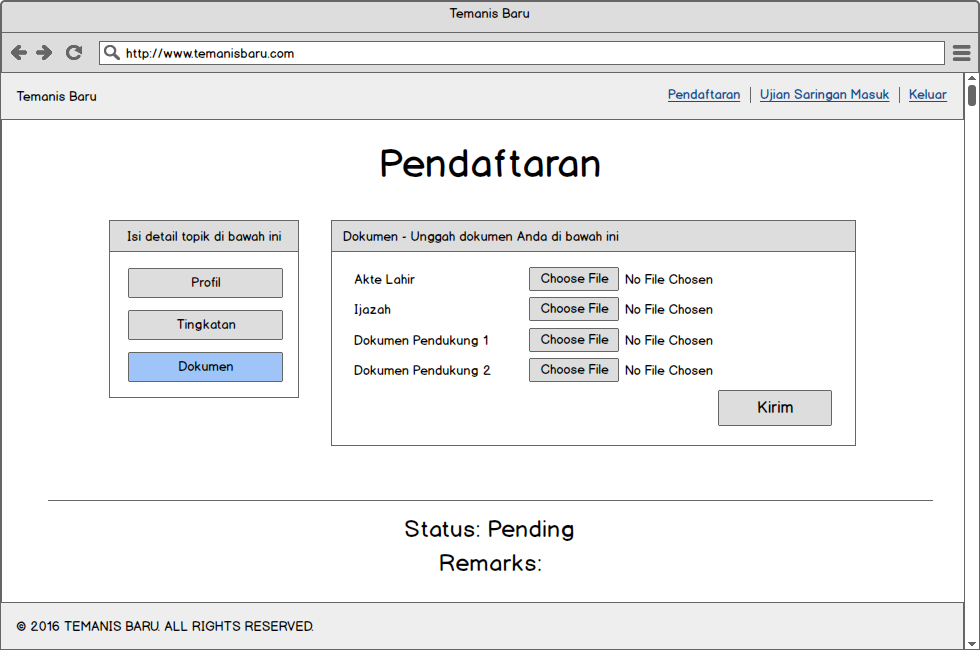
Gambar 3.18 menunjukkan perancangan tampilan halaman mengisi data *history* sekolah. *Form* tingkatan berisi *list* tingkatan (SMP 1 – SMA 3) yang bisa dipilih calon siswa serta informasi nama sekolah sebelumnya.



Gambar 3.18 Perancangan Tampilan Halaman Mengisi Data *History* Sekolah

**3.2.3.6 Perancangan Tampilan Halaman Meng-*upload* Dokumen**

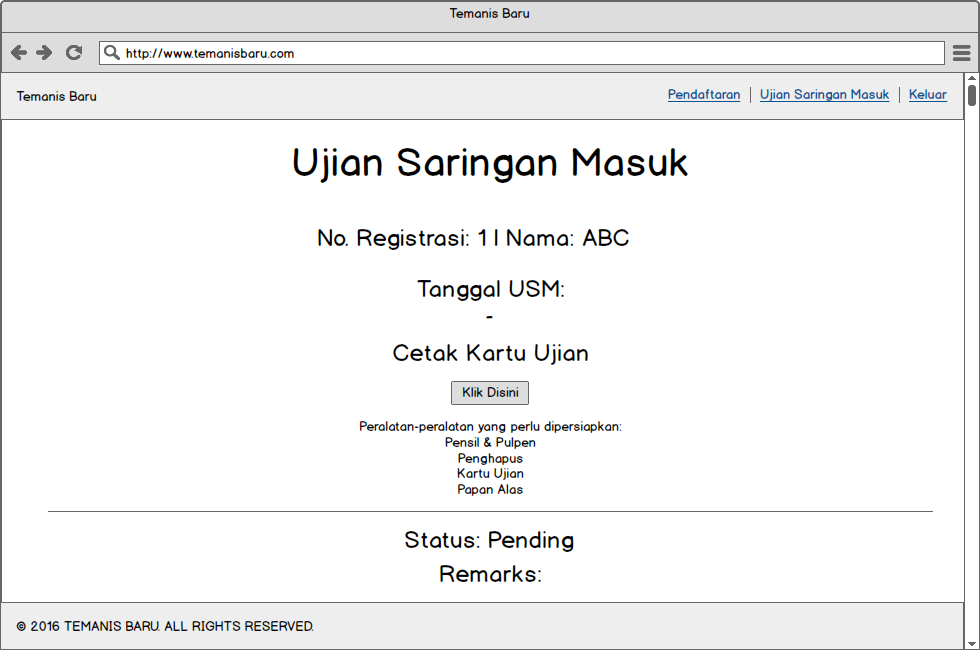
Gambar 3.19 menunjukkan perancangan tampilan halaman meng-upload dokumen. *Form* dokumen berisi dokumen-dokumen yang harus di-*upload* guna kepentingan administrasi pendaftaran siswa baru.



Gambar 3.19 Perancangan Tampilan Halaman Meng-*Upload* Dokumen

**3.2.3.7 Perancangan Tampilan Halaman Ujian Saringan Masuk (Calon Siswa)**

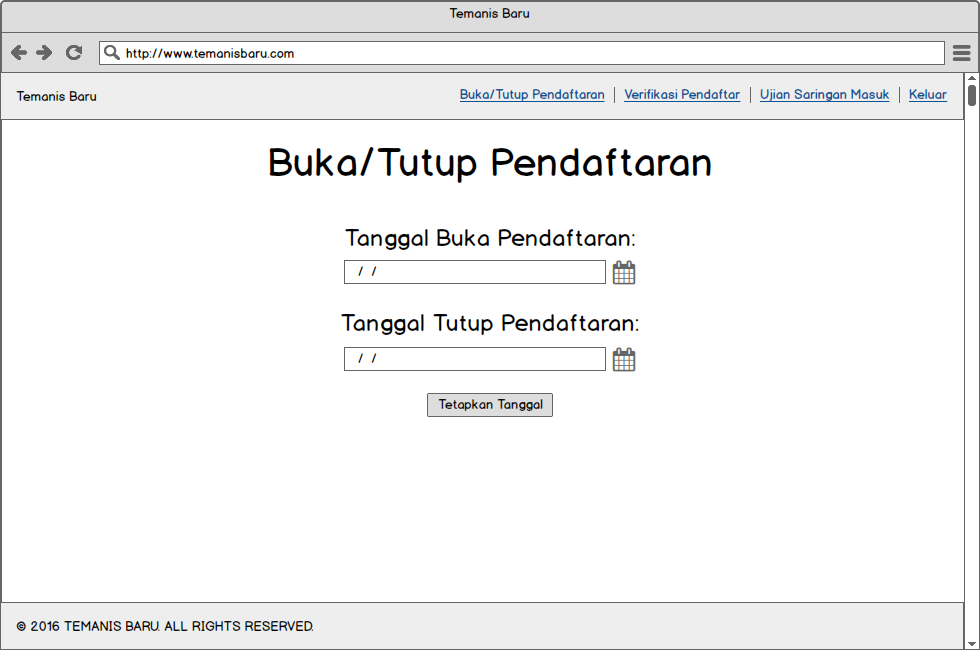
Gambar 3.20 menunjukkan perancangan tampilan halaman ujian saringan masuk bagi calon siswa. *Form* USM yang diakses calon siswa akan menampilkan informasi tentang tanggal USM, peralatan yang perlu disiapkan pada saat USM serta calon siswa dapat mencetak kartu ujian.



Gambar 3.20 Perancangan Tampilan Halaman Ujian Saringan Masuk (Calon Siswa)

**3.2.3.8 Perancangan Tampilan Halaman Buka/Tutup Pendaftaran**

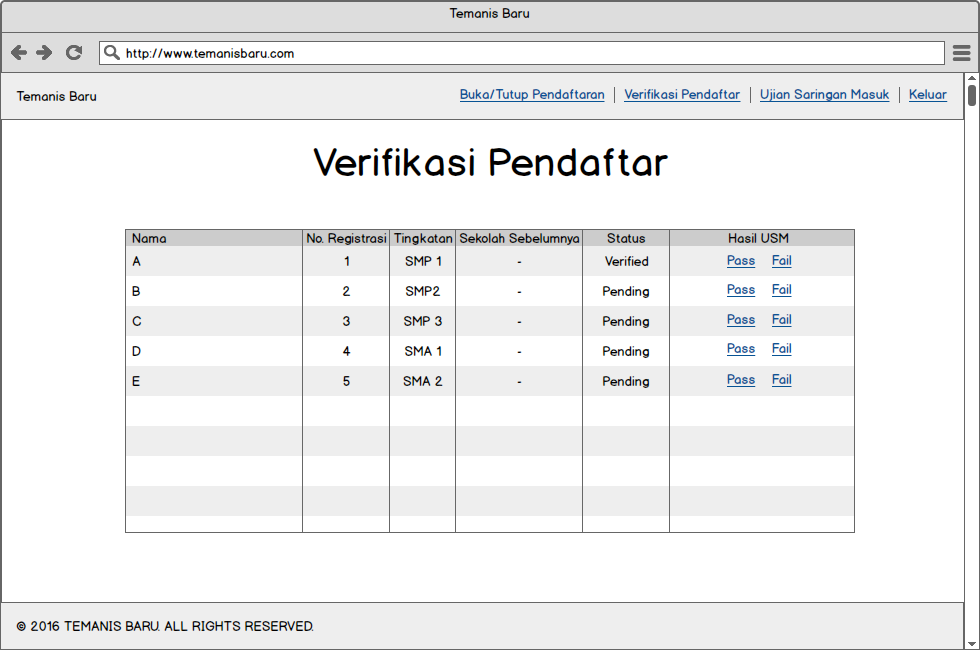
Gambar 3.21 menunjukkan perancangan tampilan halaman buka/tutup pendaftaran. Admin dapat menetapkan tanggal buka maupun tanggal tutup pendaftaran dalam form ini.



Gambar 3.21 Perancangan Tampilan Halaman Buka/Tutup Pendaftaran

**3.2.3.9 Perancangan Tampilan Halaman Verifikasi Pendaftar**

Gambar 3.22 menunjukkan perancangan tampilan halaman verifikasi pendaftar. Dalam *form* verifikasi pendaftar, admin dapat melihat daftar calon siswa yang telah mendaftar dan melakukan pengisian aplikasi. Admin dapat memverifikasi profiil dan dokumen calon siswa serta memasukkan hasil ujian masuk.



Gambar 3.22 Perancangan Tampilan Halaman Verifikasi Pendaftar

**3.2.3.10 Perancangan Tampilan Halaman Ujian Saringan Masuk (Admin)**

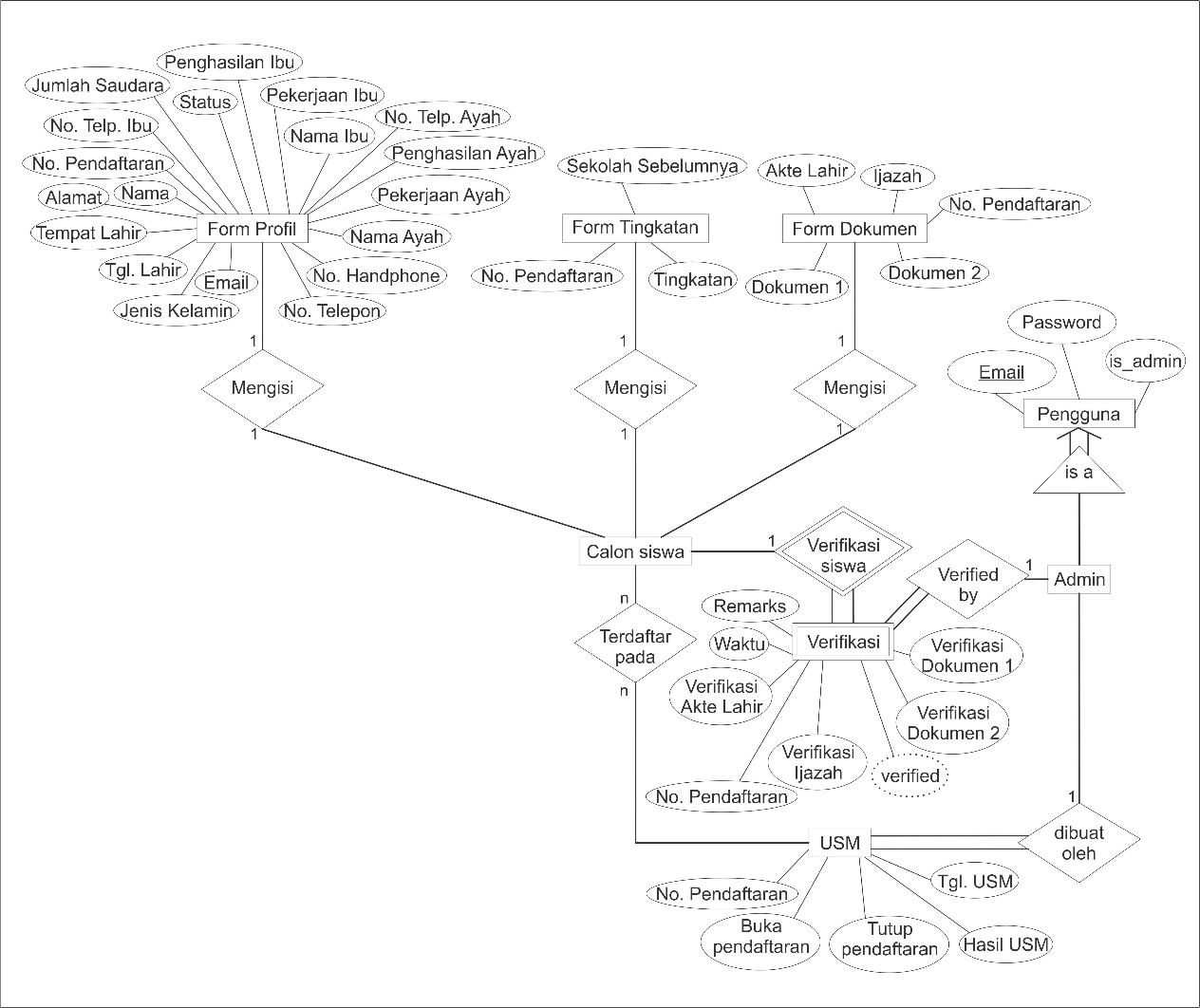
Gambar 3.23 menunjukkan perancangan tampilan halaman ujian saringan masuk bagi admin. Admin dapat menetapkan tanggal ujian saringan masuk dalam *form* ini.



Gambar 3.23 Perancangan Tampilan Halaman Ujian Saringan Masuk (Admin)

**3.2.4 Perancangan Basis Data**

TEMANIS BARU dibuat dengan menggunakan bahasa pemrograman NodeJs serta menggunakan basis data MongoDB. Dalam perancangan basis data, dapat dilihat berdasarkan ERD (Entity Relationship Diagram) pada gambar 3.24.



Gambar 3.24 Entity Relationship Diagram (ERD)