**LAPORAN AKHIR**

**PENGEMBANGAN SISTEM BERORIENTASI OBJEK**

**DEPARTEMEN ILMU KOMPUTER IPB**

**SEMESTER GENAP 2018/2019**

**NAMA APLIKASI** : SBM-Buddy

**NAMA DEVELOPER** : Kelompok 5

**ANGGOTA DEVELOPER** : Rafiqah Hasna Sukma G64160039

: Restu Triadi G64160056

: Alvin Reinaldo G64160067

1. **Deskripsi Singkat Aplikasi**

SBM-Buddy adalah aplikasi pembelajaran berbasis *web* untuk membantu siswa-siswi kelas 3 SMA dalam menghadapi ujian SBMPTN. SBM-Buddy menyediakan soal-soal SBMPTN serta memprediksi nilai *passing grade*, dan perkembangan *passing grade* siswa tiap waktu. SBM-Buddy juga menyediakan *list* *passing grade* beberapa program studi di universitas-universitas yang ada di Indonesia.

1. **User Analysis**

**Analisis User**

Target pengguna utama aplikasi SBM-Buddy adalah siswa-siswi kelas 3 SMA yang akan mengikuti SBMPTN. Pengguna termasuk dalam kategori remaja dengan rentang umur 17-19 tahun. Kebanyakan pengguna belum berpenghasilan sehingga aplikasi haruslah tersedia secara gratis. Pengguna diasumsikan sudah mahir menggunakan teknologi komputer dan internet, karena mulai tahun 2019 SBMPTN dilaksanakan berbasis komputer.

Target pengguna selanjutnya adalah admin untuk mengelola semua kebutuhan pengerjaan soal serta *passing grade*. Pengguna harus merupakan orang yang selalu mengetahui informasi seputar SBMPTN dan perguruan tinggi secara aktual. Pengguna juga sebaiknya merupakan orang yang memiliki akses ke soal-soal SBMPTN terdahulu.

**User Story**

Berikut adalah *user story* dari kedua jenis pengguna pada aplikasi SBM Buddy:

|  |  |
| --- | --- |
| **User Story #1** | Siswa dapat mengerjakan soal sehingga siswa dapat menerima hasilnya |
| **Priority** | 1 |
| **Estimate:** 6 | (2 pts) Soal bisa terisi oleh siswa  (2 pts) Siswa mendapatkan nilai hasil pengerjaan soal  (2 pts) Pembahasan ditampilkan setiap selesai mengerjakan soal |

|  |  |
| --- | --- |
| **User Story #2** | Siswa dapat melihat perkembangan agar siswa termotivasi untuk terus belajar lagi |
| **Priority** | 3 |
| **Estimate:** 2 | (2 pts) Siswa melihat dan mengakses riwayat pengerjaannya |

|  |  |
| --- | --- |
| **User Story #3** | Siswa dapat melihat *passing grade* agar siswa dapat melakukan perkiraan apakah dia diterima di suatu jurusan atau tidak |
| **Priority** | 5 |
| **Estimate:** 1 | (1 pts) Siswa melihat *passing grade* seluruh jurusan |

|  |  |
| --- | --- |
| **User Story #4** | Admin dapat melakukan kustomisasi jurusan karena *passing grade* selalu berubah dan jurusan dapat bertambah atau berkurang |
| **Priority** | 4 |
| **Estimate:** 3 | (1 pts) Admin dapat menambah jurusan baru  (1 pts) Admin dapat menghapus jurusan baru  (1 pts) Admin dapat mengubah jurusan baru |

|  |  |
| --- | --- |
| **User Story #5** | Admin dapat melakukan kustomisasi soal agar siswa dapat mengerjakan soal yang dibuat oleh admin |
| **Priority** | 2 |
| **Estimate:** 6 | (2 pts) Admin dapat menambah soal  (2 pts) Admin dapat menghapus soal  (2 pts) Admin dapat mengubah soal |

1. **Spesifikasi Teknis Lingkungan Pengembangan**

**Teknologi Pengembangan**

Bahasa pemrograman yang digunakan adalah php 7.2, JavaScript , serta HTML. Framework yang digunakan pada pengembangan aplikasi ini adalah Laravel 5.6 untuk *back-end*, Vue.js v2.9.6 dan Bulma v0.7.4 untuk *front-end*.

**Lingkungan Pengembangan**

Perangkat keras yang digunakan pada pengembangan aplikasi ini adalah:

1. Laptop : ACER Swift SF314-54G

Processor : Intel Core(TM) i7-8550U 1.80 GHz

RAM : 8GB

VGA : NVIDIA GeForce MX150

1. Laptop: Lenovo V330 14IKB-81B0

Processor : Intel Core(TM) i5-8120U 1.60GHz

RAM : 4GB

VGA : Intel HD Graphic 560

1. Laptop: ASUS VivoBook A442U

Processor : Intel(R) Core(TM) i5-8250U 1.60 GHz

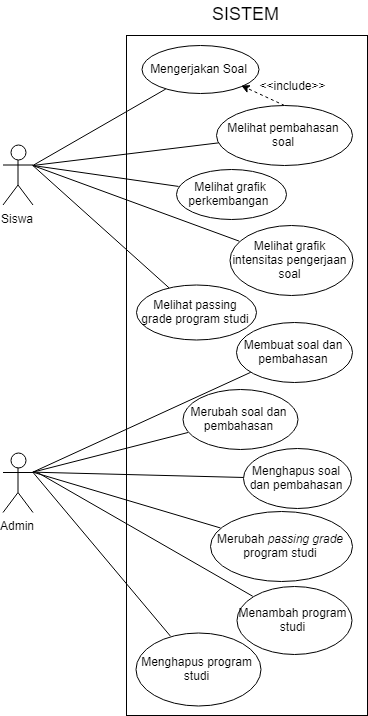
RAM : 4GB

VGA : NVIDIA GeForce 930MX

Sementara perangkat lunak yang digunakan pada pengembangan aplikasi ini adalah:

1. Visual Studio Code versi 1.33.1
2. XAMPP v3.2.2
3. Google Chrome v74.0.3
4. phpMyAdmin 4.8.3
5. **Hasil dan Pembahasan**

**Use Case Diagram**

****

Deskripsi *use case* di atas adalah sebagai berikut:

UC1 : Mengerjakan Soal

* Importance Level : High
* Use Case Type : Essential
* Primary Actor : Siswa
* Stakeholder and Interest :
  1. Siswa : Mengerjakan soalnya
  2. Admin : Membuat soal, menentukan penilaian
* Brief Description : Di dalam use case ini dijelaskan bagaimana siswa mengerjakan soal
* Pre-Condition : Siswa harus register/login
* Trigger : Siswa memilih soal yang ingin dikerjakan
  1. Type : External
* Relationship :
  1. Association : Admin
* Normal Flow of Events :
  1. Siswa Login
  2. Siswa memilih jenis soal yang ingin dikerjakan (Latihan/Try Out)
  3. Siswa mengerjakan soal
  4. Siswa Submit Soal
* Alternate/Exceptional Flows : 1E. Siswa gagal login
* Post-Condition : Siswa dapat mengakses pembahasan soal dan grafik perkembangan

UC2 : Melihat Pembahasan Soal

* Importance Level : High
* Use Case Type : Essential
* Primary Actor : Siswa
* Stakeholder and Interest :
  1. Siswa : Melihat Pembahasan Soal
  2. Admin : Menambahkan Pembahasan Soal
* Brief Description : Di dalam use case ini dijelaskan bagaimana siswa dapat mengakses pembahasan soal
* Pre-Condition : Siswa mengerjakan soal tersebut
* Trigger : Siswa selesai mengerjakan soal
  1. Type : External
* Relationship :
  1. Association : Admin
  2. Include : Mengerjakan Soal
* Normal Flow of Events :
  1. Siswa selesai mengerjakan soal
  2. Pembahasan soal ditampilkan
* Alternate/Exceptional Flows : 2E . Pembahasan belum tersedia

UC3 : Melihat grafik perkembangan

* Importance Level : Medium
* Use Case Type : Essential, System
* Primary Actor : Siswa
* Stakeholder and Interest :
  1. Siswa : Melihat grafik
* Brief Description : Di dalam use case ini dijelaskan bagaimana siswa dapat melihat grafik perkembangan nilai
* Pre-Condition : Siswa harus menyelesaikan minimal 1 set soal
* Trigger : Siswa mengakses laman perkembangan
  1. Type : External
* Normal Flow of Events :
  1. Siswa membuka profil
  2. Siswa melihat grafik perkembangan nilai Try Out

UC4 : Melihat grafik intensitas pengerjaan soal

* Importance Level : Medium
* Use Case Type : Essential, System
* Primary Actor : Siswa
* Stakeholder and Interest :
  1. Siswa : Melihat grafik
* Brief Description : Di dalam use case ini dijelaskan bagaimana siswa dapat melihat grafik intensitas pengerjaan soal dalam satuan waktu.
* Pre-Condition : Siswa harus menyelesaikan minimal 1 set soal
* Trigger : Siswa ingin melihat sebanyak apa soal yang telah dikerjakan
  1. Type : External
* Normal Flow of Events :
  1. Siswa membuka profil
  2. Siswa melihat grafik grafik intensitas pengerjaan soal dalam satuan waktu

UC5 : Melihat passing grade program studi

* Importance Level : Medium
* Use Case Type : System, Essential
* Primary Actor : Siswa
* Stakeholder and Interest :
  1. Siswa : Melihat passing grade
  2. Admin : Menambah/mengubah passsing grade
* Brief Description : Di dalam use case ini dijelaskan bagaimana siswa melihat passing grade program studi
* Trigger : Siswa ingin melihat universitas / prodi yang direkomendasikan
  1. Type : External
* Relationship :
  1. Association : Admin
* Normal Flow of Events :
  1. Siswa membuka profil
  2. Siswa melihat bagian passing grade

UC6 : Membuat Soal dan Pembahasan

* Importance Level : High
* Use Case Type : Essential
* Primary Actor : Admin
* Stakeholder and Interest :
  1. Admin : Membuat paket soal baru
* Brief Description : Di dalam use case ini dijelaskan bagaimana admin soal dapat membuat soal dan pembahasannya
* Pre-Condition : User harus merupakan seorang admin
* Trigger : Admin membuka laman penambahan soal
  1. Type : External
* Normal Flow of Events :
  1. Admin login
  2. Admin menambah paket soal baru
  3. Admin memasukkan setiap butir soal
  4. Admin memasukkan jawaban soal tersebut
  5. Admin memasukkan pembahasan tiap soal
  6. Admin menyimpan soal tersebut

UC7 : Merubah soal dan pembahasan

* Importance Level : Low
* Use Case Type : Essential
* Primary Actor : Admin
* Stakeholder and Interest :
  1. Admin : Ingin merubah soal dan pembahasan dari mata pelajaran yang diujikan.
* Brief Description : Di dalam use case ini dijelaskan bagaimana Admin merubah soal dan pembahasan dari mata pelajaran yang diujikan.
* Pre-Condition : User harus merupakan seorang admin
* Trigger : Terdapat kesalahan saat menginput soal
  1. Type : External
* Normal Flow of Events :
  1. Admin membuka list soal ujian.
  2. Admin merubah soal dan pembahasan dari ujian yang dipilih.

UC8 : Menghapus soal dan pembahasan

* Importance Level : Low
* Use Case Type : Essential
* Primary Actor : Admin
* Stakeholder and Interest :
  1. Admin : Menghapus paket soal
* Brief Description : Di dalam use case ini dijelaskan bagaimana Admin dapat menghapus paket soal atau pembahasan
* Pre-Condition : -
* Trigger : Soal sudah tidak relevan lagi
  1. Type : External
* Normal Flow of Events :
  1. Admin membuka paket-paket soal
  2. Admin memilih paket soal yang akan dihapus
  3. Admin menghapus soal tersebut
  4. Admin melakukan konfirmasi penghapusan

UC9 : Merubah passing grade program studi

* Importance Level : Medium
* Use Case Type : Essential
* Primary Actor : Admin
* Stakeholder and Interest :
  1. Admin : Ingin memperbarui passing grade program studi di sebuah universitas.
* Brief Description : Di dalam use case ini dijelaskan bagaimana Admin merubah passing grade program studi di sebuah universitas.
* Pre-Condition : User harus merupakan seorang admin
* Trigger : Perubahan nilai passing grade dari program studi per tahun.
  1. Type : Temporal
* Normal Flow of Events :
  1. Admin membuka list universitas.
  2. Admin memilih program studi.
  3. Admin merubah passing grade program studi.

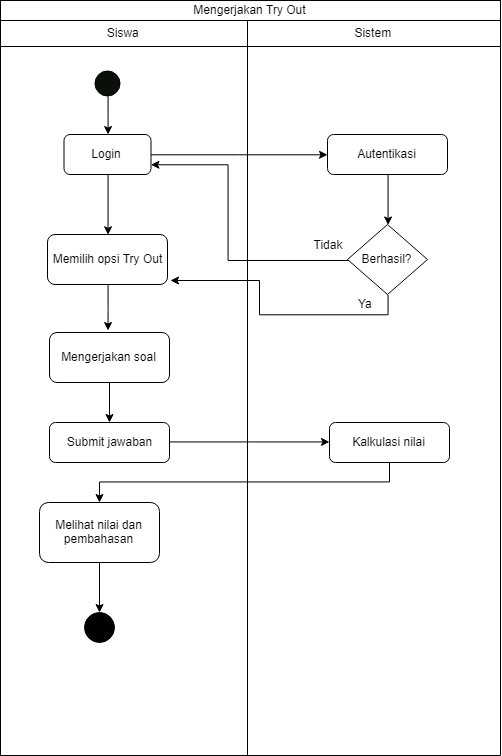
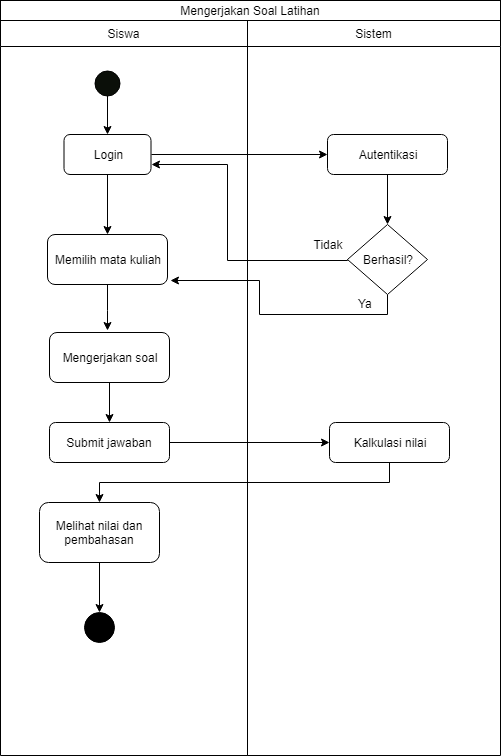
UC10 :Menambah program studi

* Importance Level : Medium
* Use Case Type : Essential, System
* Primary Actor : Admin
* Stakeholder and Interest :
  1. Admin : Ingin menambah program studi di sebuah universitas.
* Brief Description : Di dalam use case ini dijelaskan bagaimana Admin menambah program studi di sebuah universitas.
* Pre-Condition : User harus merupakan seorang admin
* Trigger : Universitas membuka program studi baru.
  1. Type : External
* Normal Flow of Events :
  1. Admin membuka list universitas.
  2. Admin menambah program studi.

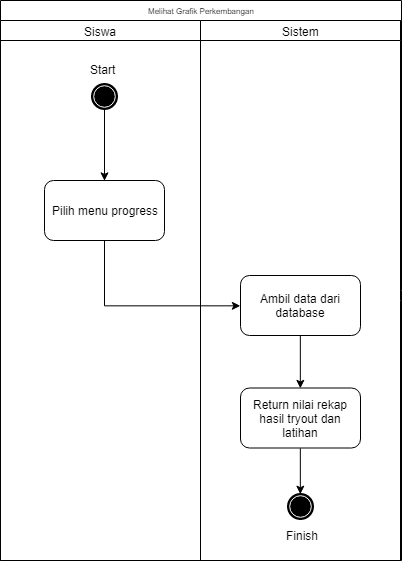
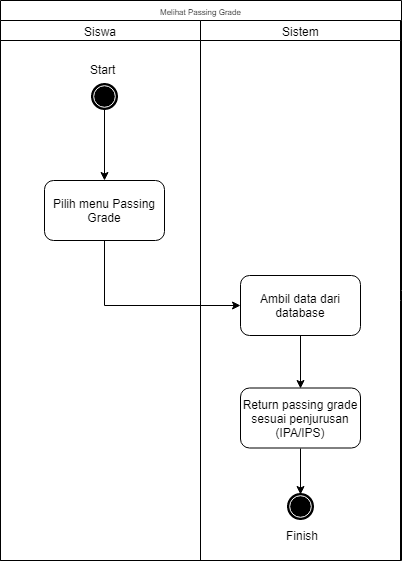
UC11 : Menghapus program studi

* Importance Level : Low
* Use Case Type : Essential, System
* Primary Actor : Admin
* Stakeholder and Interest :
  1. Admin : Ingin menghapus program studi di sebuah universitas.
* Brief Description : Di dalam use case ini dijelaskan bagaimana Admin menghapus program studi di sebuah universitas.
* Pre-Condition : User harus merupakan seorang admin
* Trigger : Universitas menutup program studi yang ada karena alasan tertentu.
  1. Type : External
* Normal Flow of Events :
  1. Admin membuka list universitas.
  2. Admin menghapus program studi

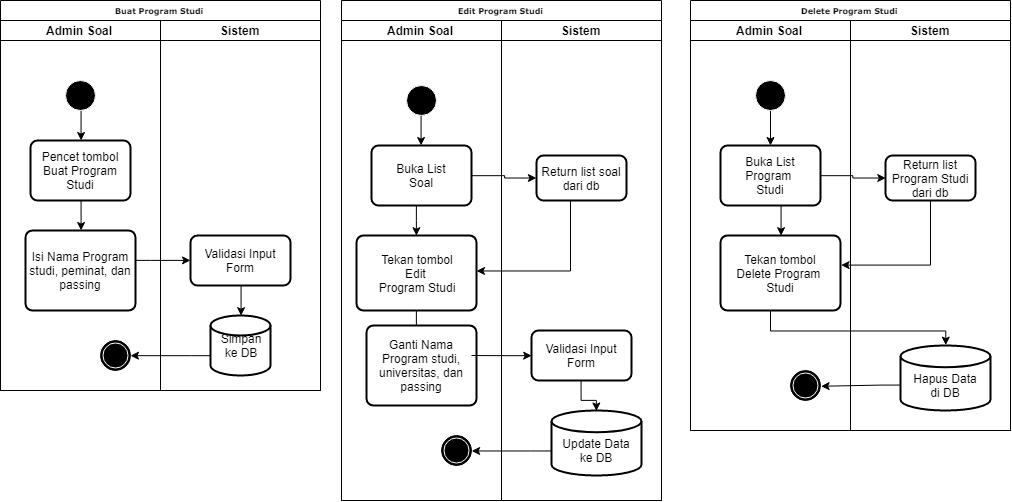
**Activity Diagram**

****

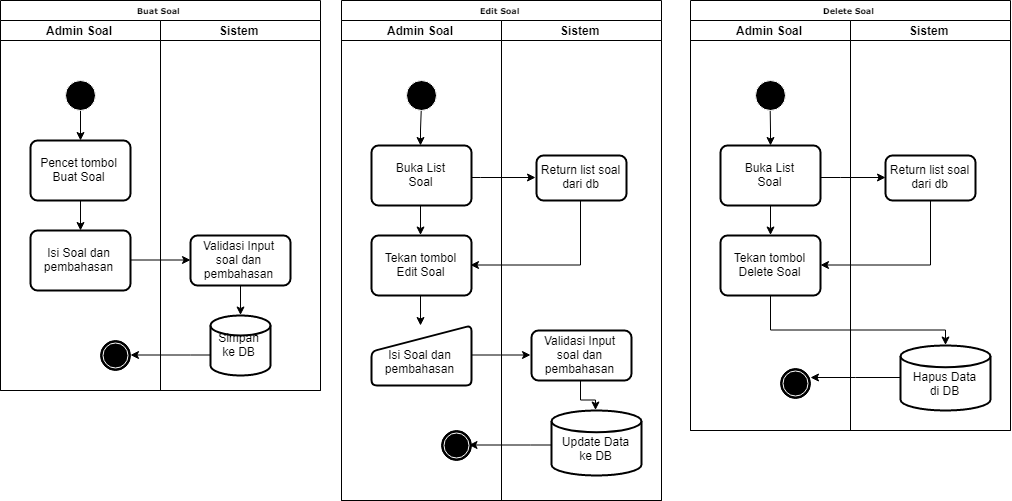
Fitur utama pada siswa di SBM-Buddy yaitu mengerjakan soal SBMPTN yang sudah tersedia. Pengerjaan soal terbagi menjadi dua mode, yaitu mengerjakan soal latihan dimana siswa dapat memilih satu mata pelajaran saja untuk berlatih, sementara pada *try out,* siswa harus mengerjakan seluruh mara pelajaran satu paket, mirip seperti SBMPTN sesungguhnya.

****

Fitur tambahan pada siswa adalah melihat *passing grade* serta melihat perkembangan dalam pengerjaan soal. Alur pada kedua fitur ini sederhana dan mudah untuk diakses.

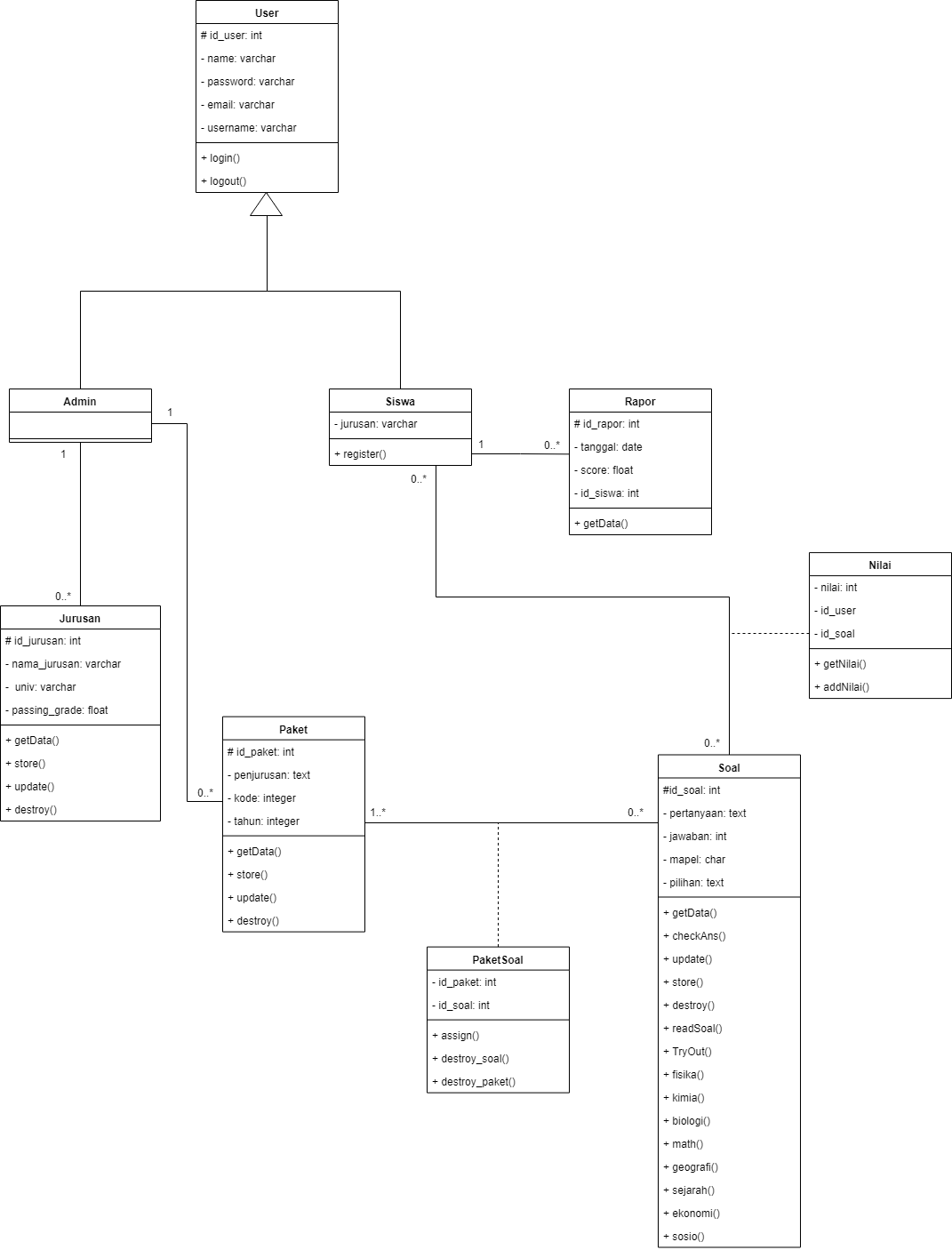
****

Fitur *create, read, update,* serta *delete* program studi sekaligus passing gradenya hanya dapat dilakukan oleh admin. Proses pengerjaannya sederhana, tidak terdapat *requirement* khusus sebelum melakukannya.

****

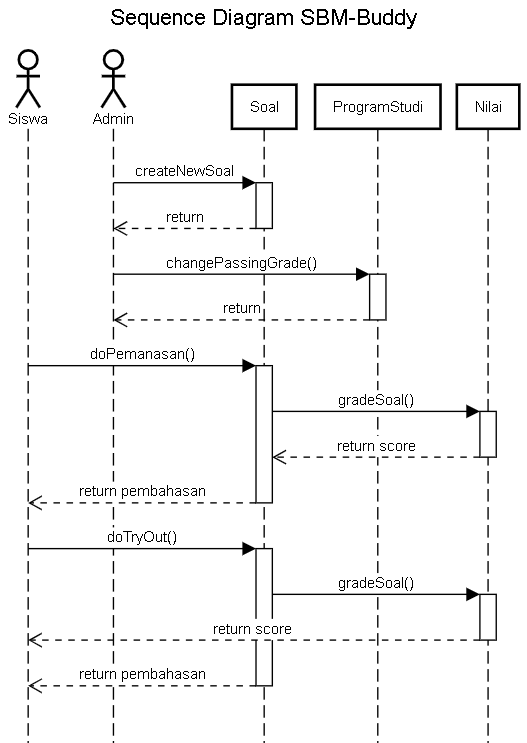
Fitur *create, read, update,* serta *delete* soal merupakan fitur utama dari admin. Mirip seperti CRUD program studi, fitur-fitur ini dirancang sederhana dan mudah digunakan. Satu-satunya syarat untuk menggunakan fitur ini adalah *login* sebagai admin.

**Class Diagram**

****

Terdapat 7 kelas dan 2 kelas asosiasi pada aplikasi ini. Kelas *user* merupakan kelas untuk mengimplementasikan pengguna pada aplikasi. Kelas memiliki dua keturunan, yaitu kelas admin dan siswa. Kelas siswa memiliki satu atribut tambahan, yaitu jurusan karena pengguna siswa terbagi menjadi dua, yaitu siswa IPA dan siswa IPS. Siswa dapat melakukan register sementara admin tidak bisa karena admin hanya ada satu. Kelas jurusan merupakan kelas untuk mengimplementasikan program studi serta *passing grade*nya. Program studi atau jurusan dapat ditambahkan, dihapus, dilihat, serta dimodifikasi oleh admin. Kelas paket adalah kelas untuk mengimplementasikan paket soal. Satu paket soal mempunyai soal-soal matematika, fisika, kimia, biologi (untuk IPA) atau geografi, sejarah, ekonomi, sosiologi yang masing-masing berjumlah 15 soal. Soal dibuat oleh admin dan dikerjakan oleh siswa. Soal dan paket dihubungkan dengan kelas asosiasi paket soal. Penilaian pengerjaan soal dilakukan oleh kelas nilai yang hasilnya akan disimpan pada rapor. Kelas rapor adalah kelas yang mengimplementasikan penyimpanan dan penampilan riwayat pengerjaan soal siswa.

**Sequence Diagram**

****

Sequence diagram di atas menunjukan bahwa terdapat 2 aktor pada sistem yakni siswa dan admin, dan terdapat 3 objek pada sistem yakni soal, program studi, dan nilai. Jika siswa memulai aktivitas dengan tanda panah *(→)* *doPemanasan* maka akan diberikan soal soal yang akan dikerjakan oleh siswa sesuai dengan jurusannya dan kemudian akan dijalankan modul gradeSoal yang akan menilai soal sesuai dengan tingkat kebenarannya dan mengembalikan nilai nya kepada user.

**Implementasi Program**

|  |
| --- |
|  |

|  |
| --- |
|  |

Salah satu fungsi penerapan OOP dari projek kami adalah untuk melakukan grading pada soal yang dikerjakan oleh Siswa yang menerapkan Prinsip Encapsulation dimana semua method dari *NilaiTrait* dapat dijalankan pada fungsi *checkAns* dan *variable* nilai hanya dapat diakses oleh fungsi *getNilai* dan *addNilai*.

|  |
| --- |
|  |

Penerapan OOP lainnya adalah prinsip Polymorphism pada AuthController , yang dimana untuk fungsi username() pada foto pertama merupakan penerapan awal yang mendefinisikan bahwa username yang digunakan untuk melakukan login adalah tabel dengan nama kolom ‘username’, kemudian dilakukan function overriding sehingga fungsi yang digunakan untuk melakukan login adalah tabel dengan nama kolom ‘email’

**Proses Pengujian:**

Proses pengujian aplikasi dilakukan dengan metode Black Box Testing, dimana beberapa fitur dicoba dan dilihat outputnya dan dibandingkan dengan output harapan.

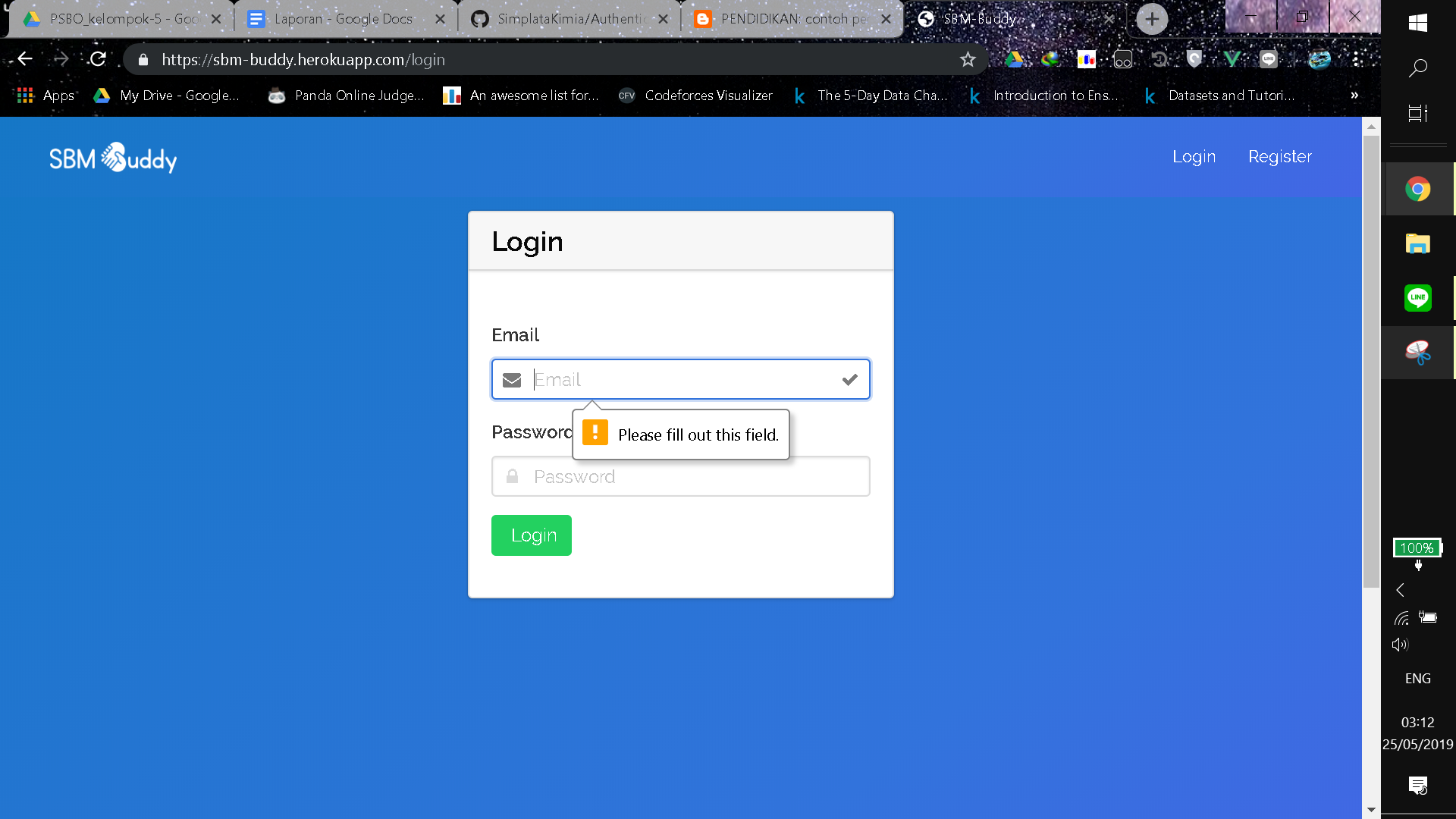
1. Login

* Mengosongkan email dan password

Ekspektasi: Tidak dapat login dan diharuskan mengisi keduanya

Status pengujian: Berhasil

Hasil:

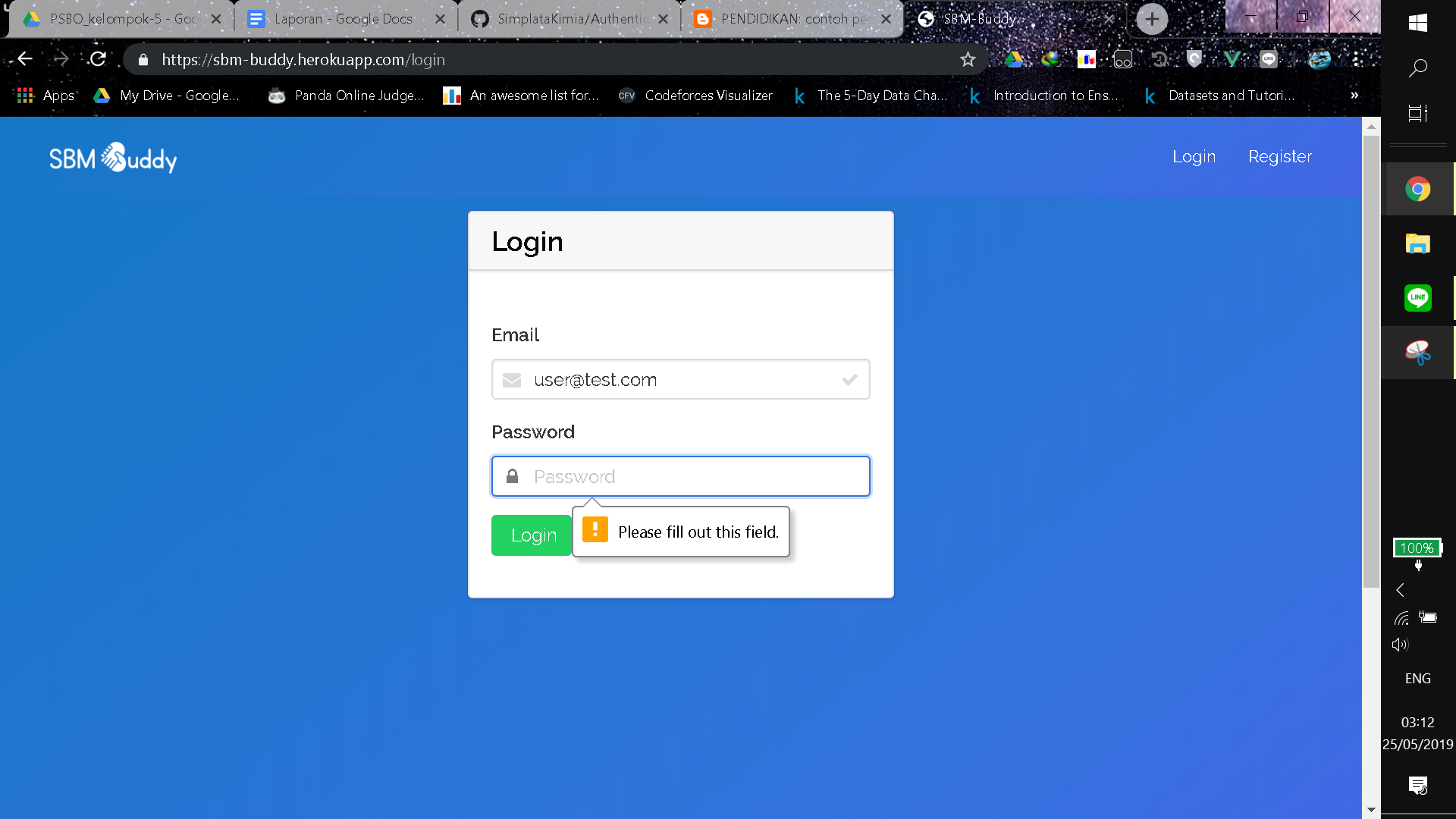


* Mengosongkan password

Ekspektasi: Tidak bisa login dan harus mengisi password terlebih dahulu

Status pengujian: Berhasil

Hasil:

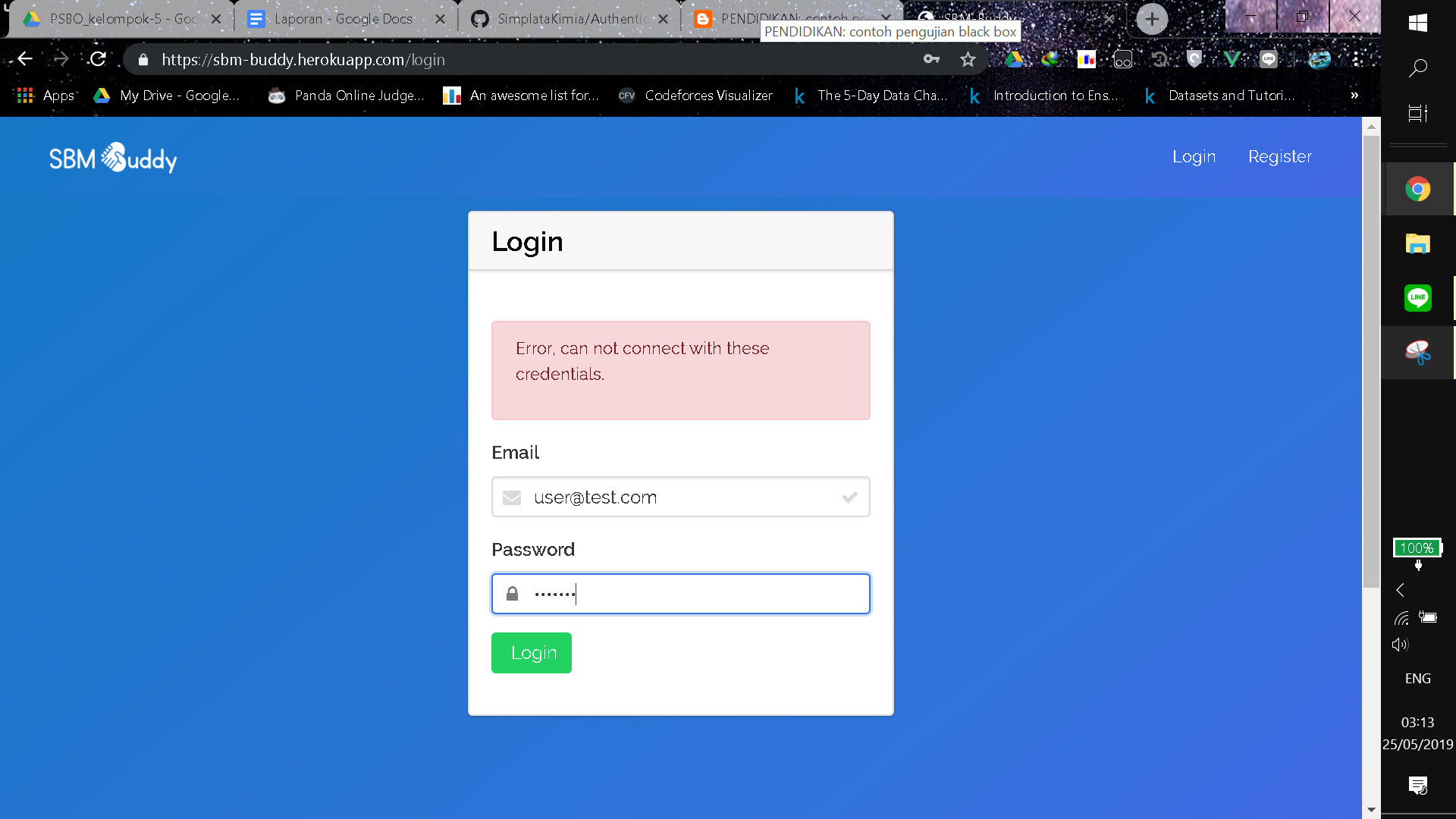


* Memasukan email dan password yang salah

Ekspektasi: Mengeluarkan pesan *error* dan tidak dapat login

Status pengujian: Berhasil

Hasil:

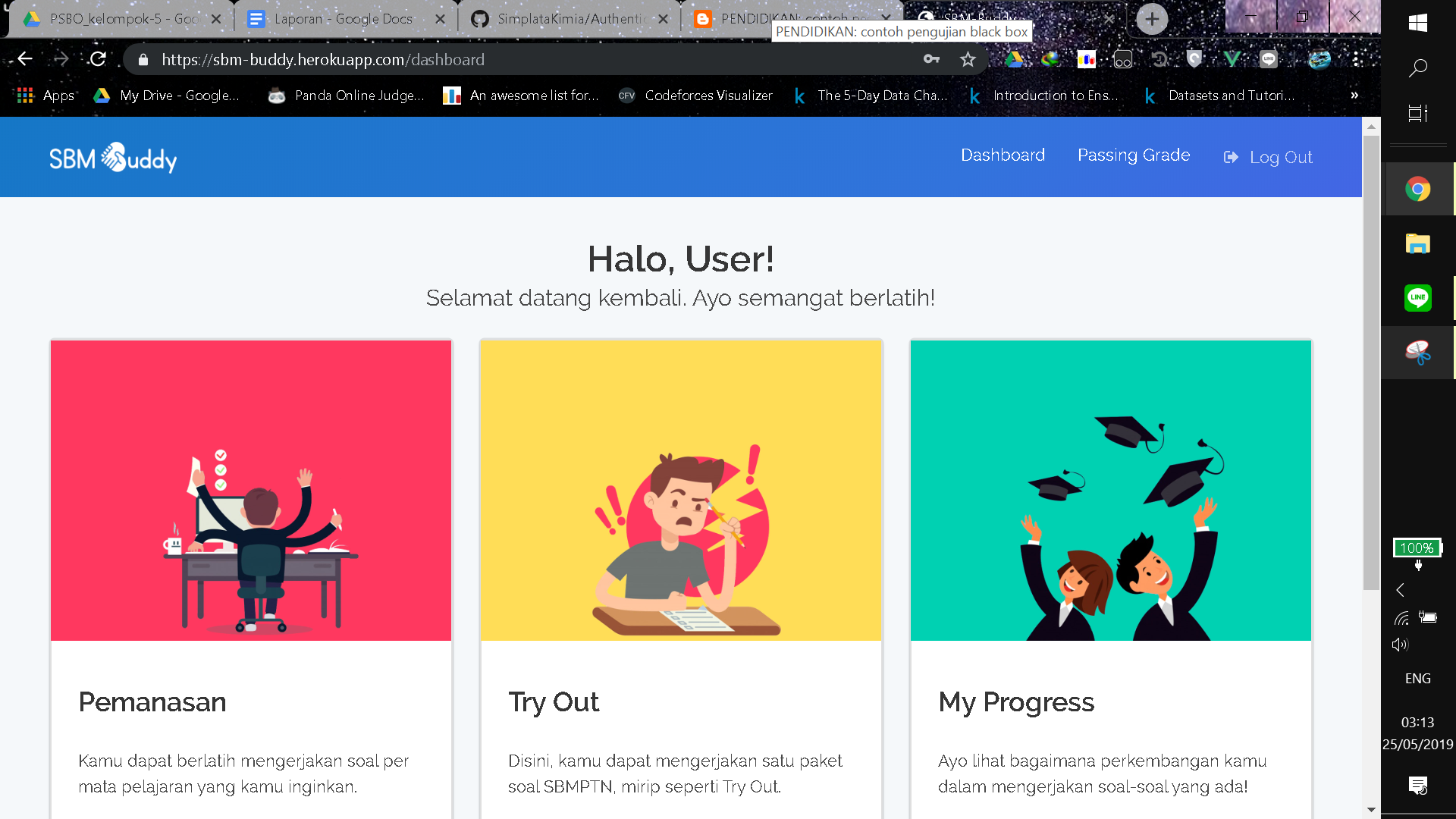


* Memasukan email dan password yang benar

Ekspektasi: Berhasil masuk ke dalam *dashboard*

Status pengujian: Berhasil

Hasil:



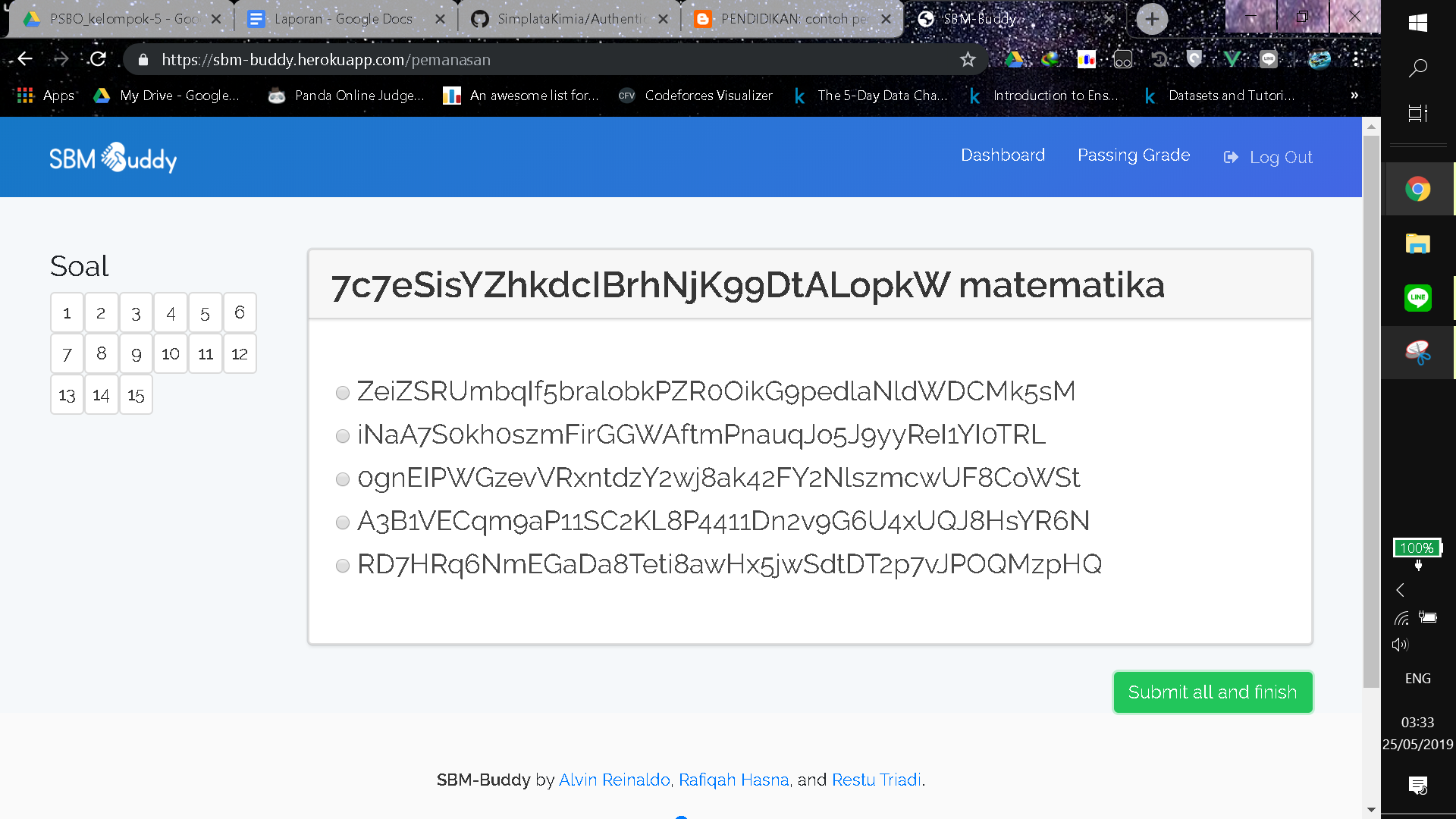
1. Soal

* Mengosongkan semua soal

Ekspektasi: tidak terjadi apa-apa.

Status pengujian: Berhasil

Hasil:



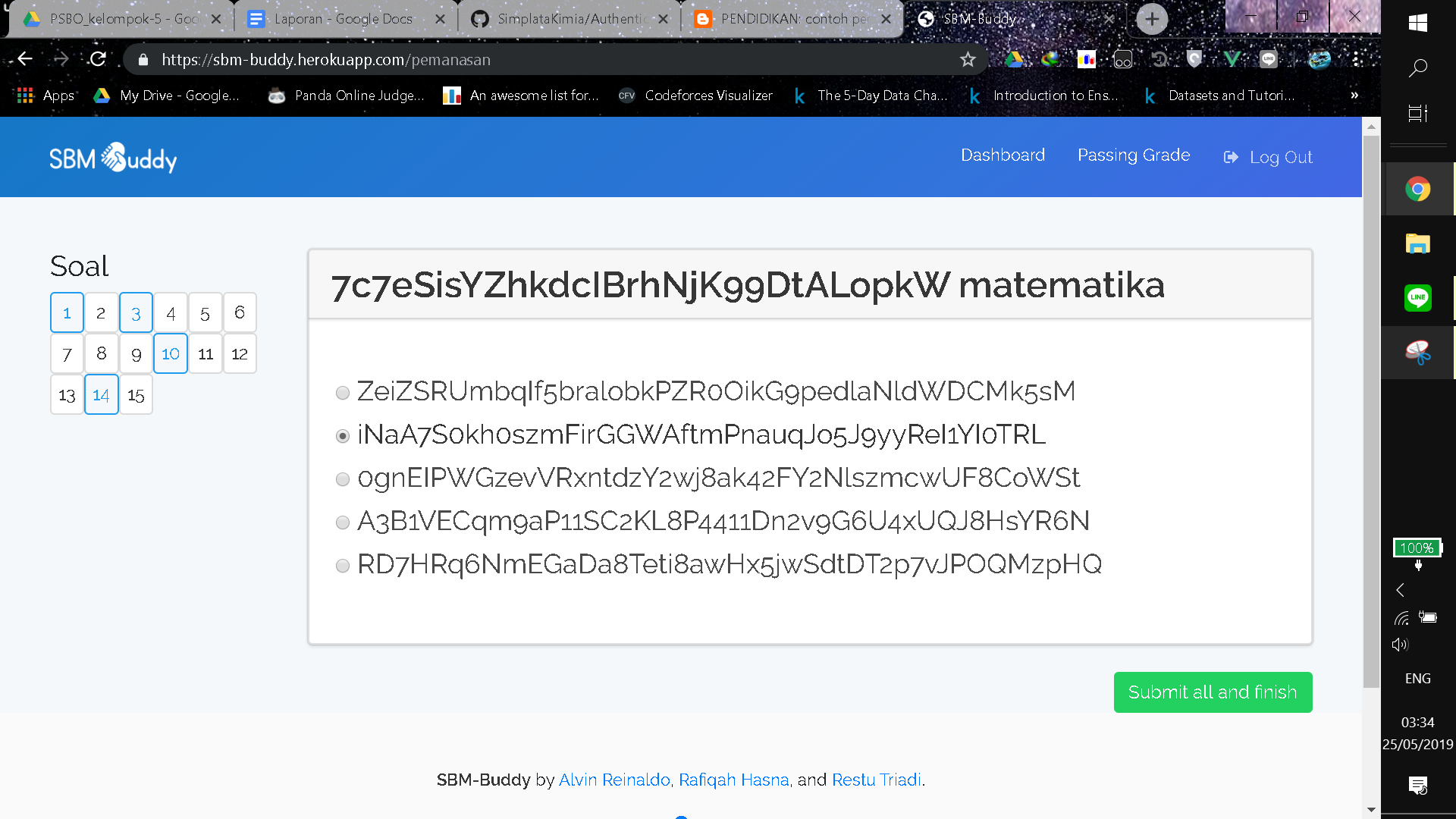
* Mengisi 1-15 soal

Ekspektasi: Berhasil dikumpulkan dan keluar *review* dari pengerjaan soalnya

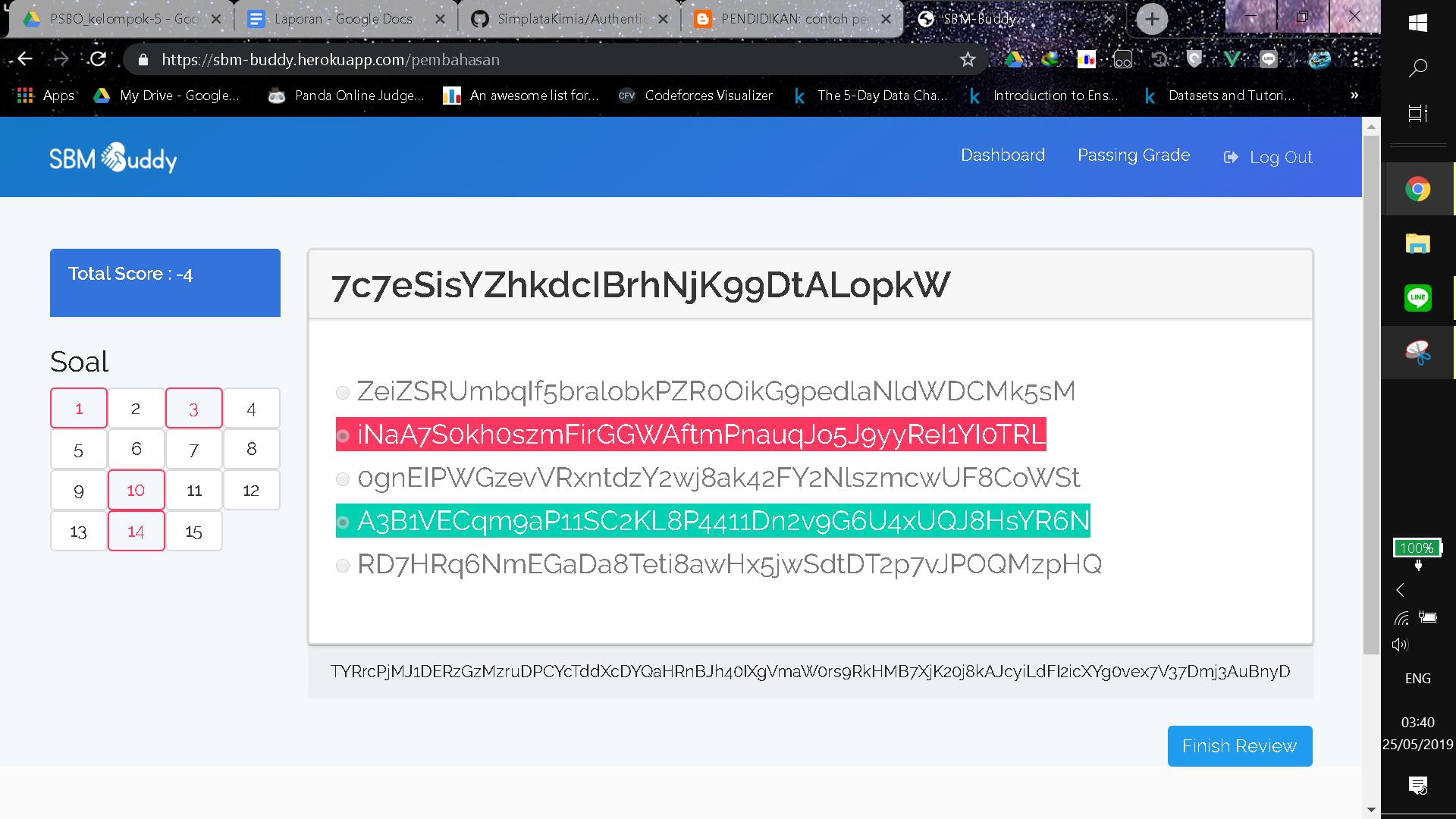
Status pengujian: Berhasil

Hasil:

Input :



Output :



1. **Saran**

Pemberian batas waktu pada setiap pengerjaan mata pelajaran dapat ditambahkan untuk pengembangan selanjutnya, sebab SBMPTN tahun 2019 menggunakan aturan satu mata pelajaran hanya dapat dikerjakan dalam waktu 23 menit. Selain itu, *filter* pada jurusan dan *passing grade* dapat ditambahkan untuk kemudahan akses *passing grade*.

1. **Lampiran**

**Design Pattern**

1. Private Class Data

Design pattern ini digunakan mengurangi keterbukaan atribut dengan membatasi visibilitasnya. Private Class Data mengatasi masalah di atas dengan mengekstraksi kelas data untuk kelas target dan memberikan *instance* kelas target *instance* kelas data yang di ekstraksi. Terdapat fungsi guard() untuk memeriksa apakah akun tersebut terverifikasi sebagai user atau admin. Guard() bersifat rahasia, sehingga hanya bisa diakses dengan AuthController saja.

1. Singleton

Singleton adalah kelas yang anggota *instance*nya hanya satu. Terdapat kelas admin pada sistem SBM Buddy. Admin yang dibutuhkan untuk mengatur keseluruhan sistem hanya satu, sehingga kelas admin hanya terdiri dari satu user, yaitu admin itu sendiri.

**AntiPattern**

1. Functional Decomposition

AntiPattern ini merupakan hasil dari developer *nonobject-oriented* yang mengimplementasikan aplikasi di bahasa pemrograman yang *object oriented*. AntiPattern berakibat pada developer yang mengabaikan hirarki kelas atau bahkan konsep *object-oriented* pada produk. Antipattern ini terjadi pada saat membuat kelas User. User terbagi menjadi dua subclass, admin dan siswa. Namun, pada tahap implementasinya, tidak ada perbedaan yang signifikan terhadap admin dan siswa.

1. Smoke and Mirrors

Smoke and Mirrors adalah salah satu antipattern dimana developer mempromosikan dan menjanjikan fitur-fitur di luar kemampuan developer tersebut. Smoke and Mirror terjadi karena demonstrasi sistem adalah strategi pemasaran yang penting. Demonstrasi sering diinterpretasikan oleh *end user* sebagai representasi kualitas produksinya, sehingga developer berani menjanjikan sesuatu agar dapat meyakinkan *end user* untuk memakai atau membeli produknya.

Antipattern yang terjadi pada projek SBM Buddy adalah ketika developer menjanjikan fitur rekomendasi program studi berdasarkan nilai per mata kuliah dan *passing grade*. Awalnya, developer mencoba untuk mengembangkan fitur ini, namun terdapat banyak *bug* pada fitur dan seiring berjalannya waktu, *bug* tidak kunjung dapat diatasi, sehingga fitur dihilangkan.

|  |
| --- |
|  |
|  |
|  |