**SISTEM PAKAR BIMBINGAN KONSELING DALAM KAMPUS UNIVERSITAS ADVENT INDONESIA BERBASIS WEBSITE MENGGUNAKAN FRAMEWORK LARAVEL MENGGUNAKAN METODE *FORWARD CHAINING***

**SKRIPSI**

Karya Ilmiah Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Komputer   
(S.Kom.) Dari Universitas Advent Indonesia

Disusun oleh:

**HASUDUNGAN SITORUS**

**NIM: 1681024**



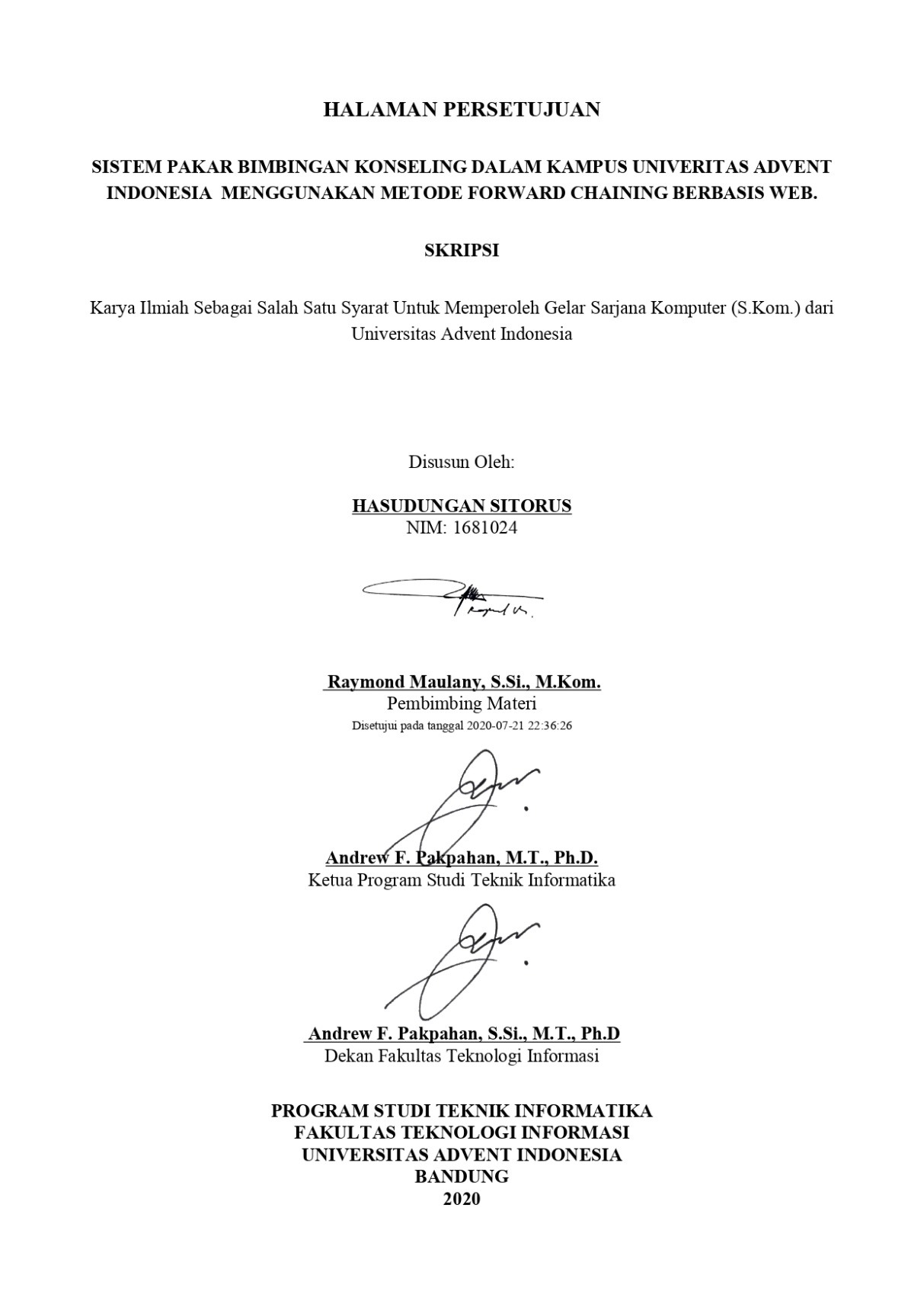
**PROGRAM STUDI TEKNIK INFORMATIKA**

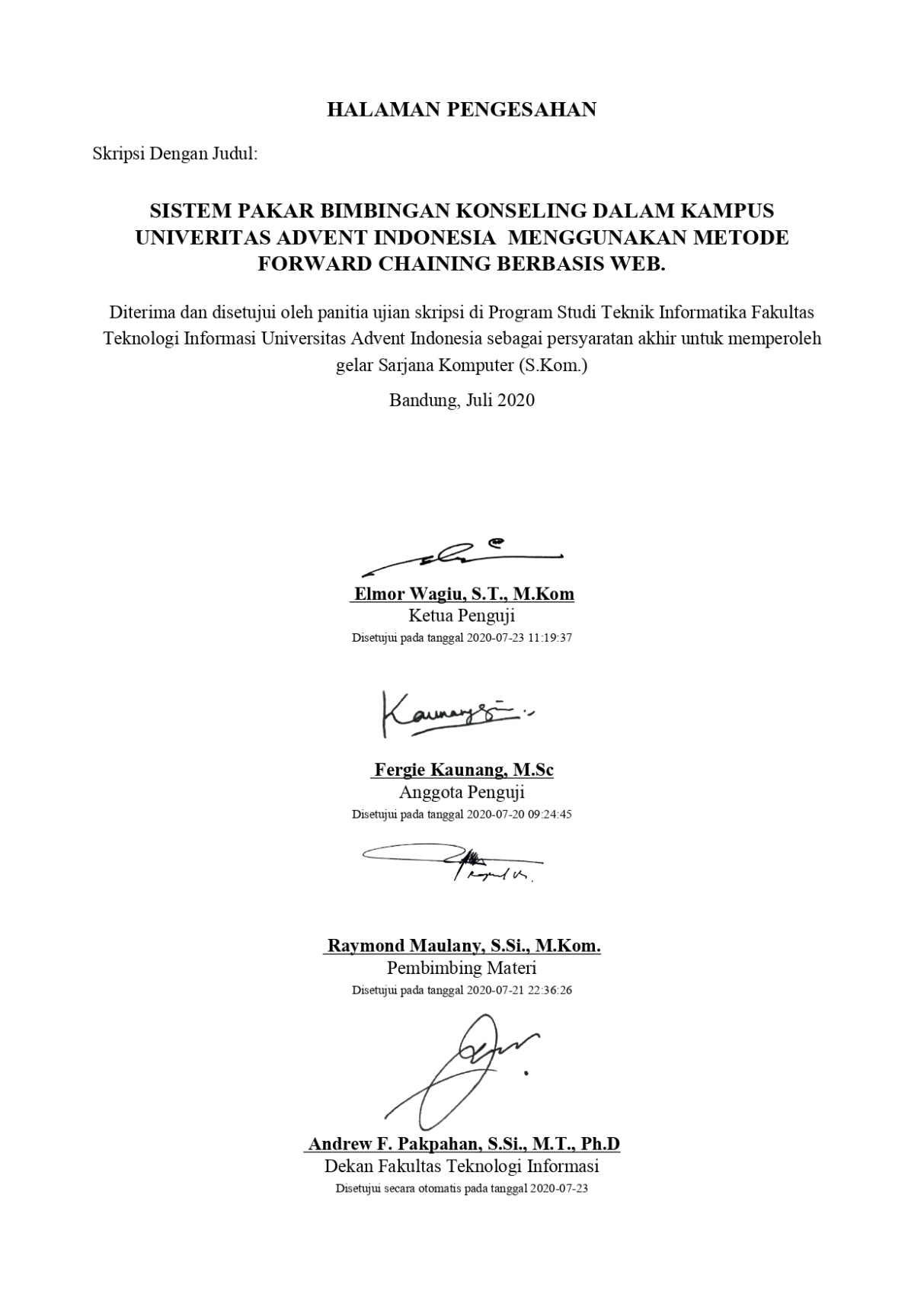
**FAKULTAS TEKNOLOGI INFORMASI**

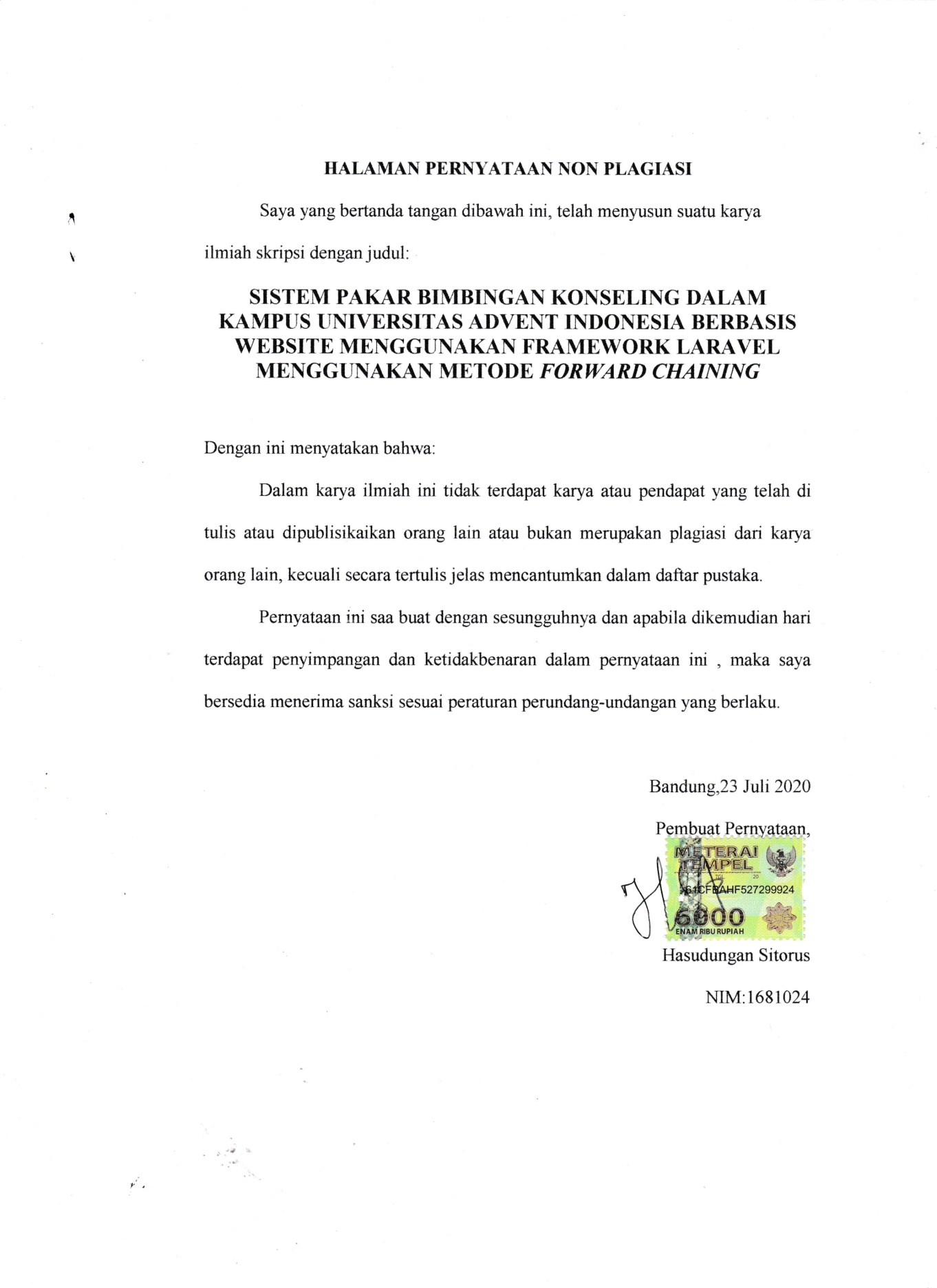
**UNIVERSITAS ADVENT INDONESIA**

**BANDUNG**

**2020**

****





**ABSTRAK**

**SISTEM PAKAR BIMBINGAN KONSELING DALAM KAMPUS UNIVERSITAS ADVENT INDONESIA BERBASIS WEBSITE MENGGUNAKAN FRAMEWORK LARAVEL MENGGUNAKAN METODE *FORWARD CHAINING***

**HASUDUNGAN SITORUS**

**NIM:1681024**

Konsultasi merupakan kegiatan Tanya jawab antara ahli pakar dengan pasien yang bersifat menyelesaikan masalah dengan memberikan jawaban serta solusi terbaik dari seorang ahli pakar sesuai dengan bidangnya. Selain mempunyai manfaat, Konsultasi juga memiliki beberapa kendala yaitu keterbatasan waktu antara *konsellor* maupun pasien. Dengan keadaan seperti ini sistem pakar menjadi jalan alternatif Penerapan Bimbingan Konselling bermanfaat bagi kehidupan sosial khususnya mahasiswa yang ada di kampus Universitas Advent Indonsia. Selain mempunyai manfaat, kegiatan bimbingan konselling juga memiliki keterbatasan salah satunya adalah bimbingan konselling yang sulit di temukan dan juga keterbatasan waktu dalam pertemuan kegiatan bimbingan konselling. Aplikasi sistem pakar bimbingan konseling diharapkan dapat menyelesaikan permasalahan pada mahasiswa dalam waktu pertemuan konseling. Sistem Pakar Bimbingan Konselling menggunakan metode *forward chaining* dengan teknik pencarian penelusuran kesimpulan *best first search*. Hasil dari penelitian ini di harapkan dapat menjadi sistem bantu bimbingan konseling.

Kata Kunci:Konsultasi, Bimbingan Konseling, Kampus Universitas Advent Indonesia, Sistem Pakar, *Forward Chainning*.

***Counseling Guidance Expert System Application at Advent Indonesia University campus Web-based using laravel Framework with Forward Chaining Method***

***Abstract***

*Consultation is a question and answer activity between expert experts and patients who are solving problems by providing answers and the best solutions from an expert in accordance with their fields. In addition to having benefits, consultation also has several obstacles, namely the limited time between counselor and patient. With this condition, the expert system becomes an alternative way for the application of counseling guidance to benefit social life, especially students on the campus of Adventist University of Indonesia. Besides having benefits, counseling guidance activities also have limitations one of which is counseling guidance that is difficult to find and also limited time in counseling guidance meeting activities. Application of counseling guidance expert system is expected to be able to solve problems in students in counseling meetings. Expert Counseling Expert System uses the method of forward chaining with best first search conclusion search techniques. The results of this study are expected to be a guidance counseling assistance system.*

***Keywords:*** *Consultation, Counseling Guidance, Universitas Advent Indonesia Campus, Expert Sistem, Forward Chaining*

**KATA PENGANTAR**

Puji dan syukur penulis panjatkan kepada Tuhan yang kuasa, atas segala kash dan karunianNya sehingga penulisan laporan kerja lapangan dapat diselesaikan dengan baik, Pada kesempatan ini penulis juga hendak memberikan ucapan terima kasih yang sedalam-dalamnya kepada semua pihak yang telah memberikan dukungan penuis berupa ide, motivasi materi dan juga doa, terkhusus kepada:

1. Bapak Andrew F Pakpahan,S.Si, M.T., Ph.D, selaku Ketua Program Studi Teknik Informatika serta Dekan Fakultas Teknologi Informasi yang telah memberikan dukungan, arahan, serta nasehat yang bermanfaat selama berkuliah di Universitas Advent Indonesia.
2. Bapak Raymond Maulany, S.Si., M.Kom., selaku Pembimbing tunggal yang telah meluangkan waktu dan memberikan arahan, bimbingan, dukungan, serta nasehat yang sangat bermanfaat dalam menyusun skripsi ini.
3. Seluruh staff dan dosen Fakultas Teknologi Infomatika yang memberikan ilmu bermaanfaat kepada penulis.
4. Keluarga saya dari ibu R.simanjuntak, ayah saya P sitorus beserta 4 Saudara saya yang tetap membantu dan mendukung secara moral dalam kegiatan yang saya lakukan.
5. Saudara Bliss Jaspis Sitanggang teman dekat yang sudah me dan meluangkan waktu untuk membantu dan mengajari dalam proses perancangan aplikasi sistem.
6. Keluarga Sir R. Hancock beserta anak-anaknya yang memberi motivasi dan bantuan.
7. Teman anak kamar Perry sihombing, Gedion noti, dan yang lainnya yang telah membantu dalam proses penulisan.
8. Teman-Teman dekat yang juga membantu dalam memotivasi, memmbantu dalam penulisan dan juga perancangan.

Penulis Juga menyadari masih banyak kekurangan kelengkapan laporan kerja lapangan yang disusun.Untuk itu penulis tetap menerima kritik maupun saran yang bertujuan mengembangkan penulisan ini, Tuhan memberkati.

Bandung, 29 April 2020

Hasudungan Sitorus

**DAFTAR ISI**

[**BAB 1 PENDAHULUAN** 14](#_Toc45480130)

[1.1. Latar Belakang Masalah 14](#_Toc45480131)

[1.2. Indentifikasi Masalah 15](#_Toc45480132)

[1.3. Maksud dan tujuan Penelitian 15](#_Toc45480133)

[1.4. Batasan Masalah 16](#_Toc45480134)

[1.5. Manfaat Penelitian 16](#_Toc45480135)

[1.6. Metode Pengumpulan Data 16](#_Toc45480136)

[1.7. Kerangka Pemikiran 17](#_Toc45480137)

[1.8. Sistematika Penulisan 17](#_Toc45480138)

[**BAB 2 LANDASAN TEORI** 19](#_Toc45480139)

[2.1. Sistem Pakar 19](#_Toc45480140)

[2.1.1. Pengertian Sistem Pakar 19](#_Toc45480144)

[2.2. Konsep Dasar Sistem Pakar 20](#_Toc45480145)

[2.2.1. Stuktur Sistem Pakar 20](#_Toc45480146)

[2.2.2. Komponen stuktur Sistem Pakar 21](#_Toc45480147)

[2.3. Tujuan Sistem Pakar 23](#_Toc45480148)

[2.4. Kelebihan sistem Pakar 24](#_Toc45480149)

[2.5. Sistem pakar dalam Penalaran 24](#_Toc45480150)

[2.6. Bimbingan-Konseling 25](#_Toc45480151)

[2.7. Perangkat Lunak Pendukung 26](#_Toc45480152)

[2.7.1. *World Wide Web* (www) 26](#_Toc45480153)

[2.7.2. *Hypertext Preprocessor* (PHP) 27](#_Toc45480154)

[2.7.3. HTML 28](#_Toc45480155)

[2.7.4. CSS (*Cascading Style Sheet*) 28](#_Toc45480156)

[2.7.5. Javascript 28](#_Toc45480157)

[2.7.6. *MySQL Database* 29](#_Toc45480158)

[2.7.7. *Apache* 29](#_Toc45480159)

[2.7.8. *UML* (*Unified Modelling Language*) 30](#_Toc45480160)

[**BAB 3**  **ANALISIS DAN PERANCANGAN SISTEM** 34](#_Toc45480161)

[3.1. Identifikasi Kebutuhan Sistem 34](#_Toc45480162)

[3.2. Latar Belakang objek penelitian 34](#_Toc45480163)

[3.3. Proses sistem lama 34](#_Toc45480164)

[3.4. Proses Sistem Baru 35](#_Toc45480165)

[3.5. Basis Pengetahuan 36](#_Toc45480166)

[3.6. Aturan Konsultasi 38](#_Toc45480167)

[3.7. Perancangan Sistem 39](#_Toc45480168)

[3.8. Analisis Kebutuhan Sistem 39](#_Toc45480169)

[3.8.1. Perangkat Lunak 39](#_Toc45480170)

[3.8.2. Perangkat Keras 40](#_Toc45480171)

[*3.9.* *Unifed Modelling Language* (UML) 40](#_Toc45480172)

[*3.6.1.* *Use Case Diagram* 41](#_Toc45480173)

[*3.6.2.* *Activity Diagram* 41](#_Toc45480174)

[*3.6.3.* *Sequence Diagram* 45](#_Toc45480175)

[*3.6.4.* *Class Diagram* 51](#_Toc45480176)

[*3.10.* *Database* 52](#_Toc45480177)

[3.11. Desain Tampilan antar halaman 54](#_Toc45480178)

[3.8.1. Desain *blueprint* Tampilan *form* Login **Error! Bookmark not defined.**](#_Toc45480190)

[3.8.2. Desain *blueprint* Tampilan *form* Register 55](#_Toc45480191)

[3.8.3. Desain *blueprint* Tampilan halaman utama 55](#_Toc45480192)

[3.8.4. Desain *blueprint* Tampilan dashboard admin 56](#_Toc45480193)

[3.8.5. Desain *blueprint* Tampilan Gejala index 56](#_Toc45480194)

[3.8.6. Desain *blueprint* Tampilan Permasalahan index 56](#_Toc45480195)

[3.8.7. Desain *blueprint* Tampilan basis pengetahuan index 57](#_Toc45480196)

[3.8.8. Desain *blueprint* Tampilan Bimbingan index 57](#_Toc45480197)

[3.8.9. Desain *blueprint* Tampilan *form* Permasalahan 58](#_Toc45480198)

[3.8.10. Desain *blueprint* Tampilan *form* Gejala 58](#_Toc45480199)

[3.8.11. Desain *blueprint* Tampilan *form* Bimbingan 59](#_Toc45480200)

[3.8.12. Desain *blueprint* Tampilan *form* Konselling 59](#_Toc45480201)

[**BAB 4 HASIL PEMBAHASAN** 61](#_Toc45480202)

[4.1. Hasil Perancangan antar muka 61](#_Toc45480203)

[4.1.1. Halaman Utama berserta halaman *login* 61](#_Toc45480204)

[4.1.2. Halaman Register *default* laravel 62](#_Toc45480205)

[4.1.3. Halaman *home* 62](#_Toc45480206)

[4.1.4. Halaman Permasalahan index 63](#_Toc45480207)

[4.1.5. Halaman Gejala index 64](#_Toc45480208)

[4.1.6. Halaman basis Pengetahuan index 64](#_Toc45480209)

[4.1.7. Halaman Bimbingan index 65](#_Toc45480210)

[4.1.8. Halaman Konselling index **Error! Bookmark not defined.**](#_Toc45480211)

[4.1.9. Halaman *form* Permasalahan 66](#_Toc45480212)

[4.1.10. Halaman *form* gejala 66](#_Toc45480213)

[4.1.11. Halaman *form* bimbingan 67](#_Toc45480214)

[4.1.12. Halaman *form* konselling 67](#_Toc45480215)

[4.2. Pengujian Komponen Sistem 68](#_Toc45480216)

[4.3. Hasil Pengujian komponen sistem 69](#_Toc45480217)

[**BAB 5 KESIMPULAN DAN SARAN** 73](#_Toc45480218)

[5.1. Kesimpulan 73](#_Toc45480219)

[5.2. Saran 73](#_Toc45480220)

[**DAFTAR PUSTAKA** 74](#_Toc45480221)

**DAFTAR GAMBAR**

**Gambar 2.1** Simbol dan fungsi pada use case diagram 30

**Gambar 3.1** Sistem lama Bimbingan Konselling 33

**Gambar 3.2** *Use Case Diagram* 40

**Gambar 3.3** Activity Diagram *Registrasi* 41

**Gambar 3.4** Diagram Activity *Login* 42

**Gambar 3.5** *Activity diagram form* permasalahan 43

**Gambar 3.6** Activity Diagram  *form* Gejala 43

**Gambar 3.7** Activity Diagram *form* Bimbingan 44

**Gambar 3.8** Activity Diagram *form* Konselling 45

**Gambar 3.9** *Sequence Diagram Register* 46

**Gambar 3.10** *Sequence Diagram Registrasi* 46

**Gambar 3.11** *Sequence Diagram form* Permasalahan 48

**Gambar 3.12** *Sequence Diagram form* Gejala 49

**Gambar 3.13** *Sequence Diagram Form* Bimbingan 50

**Gambar 3.14** *Sequence Diagram form* Konselling 51

**Gambar 3.15** *Class Diagram* Sistem Pakar Bimbingan Konselling 51

**Gambar 3.16** *Desain blueprint* Tampilan *Register default* laravel 55

**Gambar 3.17** *Desain blueprint* Tampilan Utama 55

**Gambar 3.18** *Desain blueprint* Tampilan dashboard admin 56

**Gambar 3.19** *Desain blueprint* Tampilan Gejala index 56

**Gambar 3.20** *Desain blueprint* Tampilan permasalahan index 56

**Gambar 3.21** *Desain blueprint* Tampilan relasi index 57

**Gambar 3.22** *Desain blueprint* Tampilan Bimbingan index 57

**Gambar 3.23** *Desain blueprint* Tampilan *form* Permasalahan 58

**Gambar 3.24** *Desain blueprint* Tampilan *form* Gejala 58

**Gambar 3.25** *Desain blueprint* Tampilan *form* Bimbingan 59

**Gambar 3.26** *Desain blueprint* Tampilan *form* Konselling 59

**Gambar 4.1** Tampilan Halaman Utama 60

**Gambar 4.2** TampilanHalamanRegister default Laravel 61

**Gambar 4.3** Tampilan halaman *home* 61

**Gambar 4.4** Tampilan halaman permasalahan index 62

**Gambar 4.5** Tampilan halaman gejala index 63

**Gambar 4.6** Tampilan halaman relasi index 64

**Gambar 4.7** Tampilan halaman bimbingan index 64

**Gambar 4.8** Tampilan halaman *form* permasalahan 65

**Gambar 4.9** Tampilan halaman *form* gejala 66

**Gambar 4.10** Tampilan halaman *form* bimbingan 66

**Gambar 4.11** Tampilan halaman *form* konselling 67

**DAFTAR TABEL**

**Tabel 2.1** Simbol Serta Penjelasan *Activity Diagram* 30

**Tabel 3.1** Kode Permasalahan Mahasiswa35

**Tabel 3.2** Gejala Permasalahan 36

**Tabel 3.3** Solusi 37

**Tabel 3.4** Aturan Konsultasi 37

**Tabel 3.5** bimbingan 52

**Tabel 3.6** gejala 52

**Table 3.7** permasalahan 53

**Tabel 3. 8** gejalaPermasalahan 53

**Tabel 3. 9** konselling 54

**Table 3.10** tmp\_konselling 54

**Tabel 4. 1** Komponen Sistem Pengujian 67

**Tabel 4. 2** Hasil Pengujian Sistem 68

# **BAB 1 PENDAHULUAN**

## **Latar Belakang Masalah**

Teknologi informasi sudah menjadi bagian penting dalam berbagai bidang kehidupan. Penyebaran informasi termasuk melalui *website* menjadi salah satu pilihan pada pembangunan sistem yang dapat mengintegrasikan kedalam sistem baru guna mempermudah kegiatan bisnis maupun kegiatan lainnya. Pembangunan *website* saat ini dapat juga menggunakan konsep sistem pakar didalamnya.

Sistem Pakar merupakan salah satu bidang dari kecerdasaan buatan yang berkaitan dengan sistem pendukung keputusan yang dirancang dengan memasukkan pengetahuan pakar kedalam sebuah sistem yang bertujuan pengambilan keputusan. pengunaan sistem pakar banyak diterapkan dalam berbagai bidang seperti manufaktur/industri, bidang kedokteran/kesehatan, psikologi, ekonomi, pemodelan, ilmu pengetahuan, eksplorasi alam, Hankam, dan juga bidang sosial budaya.

Aktivitas bimbingan dan konseling pada dasarnya, merupakan interaksi timbal-balik, yang di dalamnya terjadi hubungan saling tanya jawab tatap muka langsung antara konselor sebagai pihak yang membantu dan konseli sebagai pihak yang dibantu.

Bimbingan dan konseling di dalam kampus Universitas Advent Indonesia merupakan salah satu pelayanan kepada mahasiswa dalam memberi bimbingan serta bantuan pada setiap permasalahan yang di alami para mahasiswa. dengan adanya pelayanan kegiatan bimbingan dan konseling memberi dukungan dalam beberapa aspek pada mahasiswa salah satunya di bidang sosial. Pencapaian bimbingan dan konseling yang efektif memiliki beberapa kompetensi salah satu nya yaitu waktu praktek pada konselor, Namum dikesempatan lain terjadi kesulitan melakukan kegiatan konseliing dimana antara konselor ataupun konseli memiliki kesibukan kerja masing-masing atau kegiatan mendesak dan juga sistem pencatatan data konseli masih mengunakan sistem pencatatan manual dalam satu buku tamu.

Dengan adanya penelitian ini, diharapkan dapat membantu pelayanan bimbingan dan konseling yang sering terkendala dan juga pencatatan data koseli kedalam *database* namun tidak menggantikan peran pakar dalam bimbingan dan konseling.

## **Indentifikasi Masalah**

Beberapa hal yang akan menjadi identifikasi masalah dalam perancangan sistem pakar ini adalah sebagai berikut:

1. Bagaimana cara mengintegrasikan ilmu pakar kedalam sistem pada permasalahan sosial serta memberikan solusi yang dialami oleh mahasiswa dengan sistem pakar?
2. Bagaimana cara konselor ataupun kosnseli tetap dapat melakukan konseling dengan kesibukan mendadak atau sulit untuk di temui?
3. Bagaimana cara konselor mencatat, menyimpan, dan melihat data para konseli dengan mengunakan sistem secara komputasi?

## **Maksud dan tujuan Penelitian**

Maksud dari penulis melakukan penelitian ini dengan menerapkan ilmu pakar ke dalam aplikasi website yaitu:

1. Membantu kegiatan bimbingan dan konseling yang bersifat komputasi dan mobalitas.
2. Membuat sistem baru pada setiap bagian kegiatan bimbingan konseling salah satunya pencatatan data konseli.
3. Menghemat waktu yang lebih efesien.

## **Batasan Masalah**

Dalam Batasan masalah pada penelitian ini bertujuan agar memfokuskan kepada permasalahan diantaranya, yaitu:

1. Menampikan hasil bimbingan-konseling yang berisikan keterangan permasalahan, solusi, serta gejala-gejala bila memasukkan gejala-gejala yang tersedia pada sistem.
2. Memberikan notifikasi pemberitahuam error jika tidak memilih gejala-gejala yang tersedia pada halaman konseling didalam sistem.

## **Manfaat Penelitian**

Dalam penulisan penelitian ini, ada beberapa dampak positif yang dapat membantu untuk beberapa pihak, yaitu:

1. Menghemat waktu dalam melakukan kegiatan bimbingan dan konseling.
2. Melatih penulis dalam merancang sistem khususnya sistem pakar.
3. Membantu koselor dalam kegiatan bimbingan dan konseling bila dalam keadaan sibuk.
4. Memberikan kontribusi dalam bentuk karya tulis serta aplikasi sistem pakar bimbingan konseling.

## **Metode Pengumpulan Data**

Dalam Proses memahami dan meng-identifikasi serta menyelesaikan masalah dalam sistem pakar yang di kerjakan, penulis melakukan beberapa hal dalam pencarian sumber penelitian dengan cara:

1. Studi *Literature*
2. Mencari materi ataupun sumber yang berbentuk jurnal melalui internet yang mengarahkan ke dalam sistem pakar konsep bimbingan-konsuling.
3. Mengumpulkan matei dari beberapa buku yang di sediakan oleh kampus Universitas Advent Indonesia yang mengacu kearah sistem pakar.
4. Pengumpulan data dilapangan
5. Melakukan Wawancara kepada pihak Bimbingan-konsuling.

## **Kerangka Pemikiran**

Kegiatan bimbingan dan konseling di kampus Universitas Advent Indoenesia saat ini masih menggunakan sistem lama diantara pencatatan data para konseli masih secara manual dengan buku tamu membuat kegiatan tidak efesien serta konselor yang memiliki kegiatan kesibukan mendadak atau sulit untuk bertemu langsung yang menyulitkan kegiatan bimbingan dan konseling tidak berjalan dengan baik. Diharapakan penelitian ini memberikan solusi serta sistem baru yang memudahkan kegiatan bimbingan dan konseling menjadi mobalitas antara konselor dan juga konseli.

## **Sistematika Penulisan**

Sistematika Penulisan penelitian tugas akhir yang dilakukan adalah sebagai berikut:

BAB Ⅰ adalah Pendahuluan, membahas latar belakang permasalahan, identifikasi masalah, maksud penelitian, batasan masalah, manfaat penelitian, metode pengumpulan data, kerangka pemikiran,sistematika penulisan.

BAB Ⅱ adalah Landasan Teori, yang membahas teori-teori sistem pakar .

BAB Ⅲ Membangun tampilan-tampilan *user interface* pada sistem yang akan di rancang dalam bentuk tampilan abstrak serta menerapkan aturan-aturan ataupun fungsi dalam metode yang di tetapkan sebelumnya.

BAB Ⅳ Penjelasan tampilan setiap halaman sistem aplikasi sitisu websistem pakar sesungguhnya beserta fungsi-fungsi yang di rancang berbentuk gambar tangkapan.

BAB Ⅴ Berisikan kesimpulan dan saran yang bertujuan mengembangkan aplikasi sistem pakar yang telah di rancang oleh penulis pertama pada penilitian yang selanjutnya.

# **BAB 2 LANDASAN TEORI**

## **Sistem Pakar**



### **Pengertian Sistem Pakar**

Sistem pakar termasuk kedalam pengelompokan kecerdasan buatan yang mempunyai kemampuan khusus untuk menyelesaikan kondisi permasalahan yang ada (Gede & Divayana,2014)

Sistem pakar adalah bagian yang terdapat pada kecerdasan buatan yang diperuntukkan dalam pendiagnosaan kerusakan sistem dan sebagai solusi permasalahan (Divya & Sreekumar,2014).

Sistem Pakar adalah *System* yang kinerjanya mengadopsi keahlian yang dimiliki seorang pakar dalam bidang tertentu ke dalam sistem atau program computer yang disajikan dengan tampikan yang dapat digunakan oleh pengguna dapat membuat sebuah keputusan atau menentukan layaknya seorang pakar (Anik Andriani,2016).

Sistem Pakar adalah sistem yang menghasilkan keputusan atas dasar basis pengetahuan yang dimiliki para ahli. (Tan, Wahidin,Tamaldin, 2016)

sistem pakar adalah program computer yang mensimulasi penilaian dan perilaku manusia atau organisasi yang memiliki pengetahuan dan pengalaman ahli dalam bidan tertentu, (Budiharto dan Suhartono, 2014).

Sistem pakar adalah suatu program komputer berbasis pengetahuan yang berusaha seorang pakar ke komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh seorang pakar. Seperti hal nya seorang pakar, sistem pakar terfokus pada suatu dominan masalah yang spesifik, (Minarni dan hidayat ,2013:27).

Menurut Sembiring (2013:7), sistem pakar (*expert system*) adalah sistem berbasis komputer yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli.

Berdasarkan pengertian para peneliti maka Dapat di simpulkan bahwa sistem pakar adalah sebuah sistem yang dapat memecahkan permasalah dan memberikan solusi dengan menggunakan penalaran berdasarkan ilmu pakar.

## **Konsep Dasar Sistem Pakar**

### **Stuktur Sistem Pakar**

Menurut Nita Merlina dan Rahmat Hidayat dalam dalam bukunya perancagan sistem pakar (2012:3), Sistem pakar terdiri atas 2 bagian pokok yaitu lingkungan pengembangan(*development*), dan lingkungan konsultasi (*consultation environment*). berikut pengertiannya:

1. Lingkungan Pengembangan (*development environment*)

Lingkungan Pengembangan adalah ruang lingkup yang berisikan komponen-komponen tertentu dan juga bertujuan mengembangkan basis pengetahuan yang sudah terkumpul serta menggabungkan format kode tertentu di dalam sistem namun tidak untuk menampilkan hasil kepada pengguna dari proses tersebut.

1. Lingkungan Konsultasi (*consultation environment*)

Lingkungan Konsultasi adalah ruang lingkup yang memiliki beberapa komponen-komponen tertentu yang bekerja di dalamnya serta menampilkan hasil dari proses tersebut kepada pengguna sistem.

### **Komponen stuktur Sistem Pakar**

Ada beberapa komponen penting yang terdapat pada sistem pakar menurut (Ramanda,2015), yaitu sebagai berikut:

1. Antar Muka pengguna (*User Interface*)

Antarmuka pengguna (*user interface*) merupakan mekanisme yang digunakan oleh pengguna dan sistem pakar untuk berkomunikasi. Antar muka menerima informasi dari pemakai dan mengubahnya kedalam bentuk yang dapat diterima oleh sistem dan sebaliknya antarmuka juga menerima informasi dari sistem dan menyajikannya kedalam bentuk yang dimengerti oleh pemakai. Jadi pada bagian ini terjadi dialog antara program dan pemakai.

1. Basis Pengetahuan

Basis pengetahuan mengandung pengetahuan untuk pemahaman, formulasi, dan penyelesaian masalah. Komponen sistem pakar ini disusun atas dua elemen dasar yaitu fakta dan aturan. Fakta merupakan informasi tentang obyek dalam area permasalahan tertentu, Aturan merupakan informasi tentang cara memperoleh fakta baru dari fakta yang telah diketahui.

1. Akusisi Pengetahuan

Akuisisi pengetahuan (*knowledge acquisition*) adalah akumulasi, transfer dan transformasi keahlian dalam menyelesaikan masalah dari sumber pengetahuan kedalam program komputer. Knowledge engineer akan menyerap pengetahuan yang selanjutnya akan ditransfer kedalam basis pengetahuan. Pengetahuannya diperoleh dari pakar yang dilengkapi dari buku, basis data, laporan penelitian dan pengalaman si pemakai.

1. Mesin Inferensi

Mesin inferensi merupakan otak dari sebuah sistem pakar dan dikenal juga dengan sebutan struktur kontrol (control *structure*) atau *rule interpreter* (dalam sistem pakar berbasis kaidah)

1. Tempat Kerja (*work place*)

*workplace* merupakan area dari sekumpulan memori kerja (*working memory*) yang digunakan untuk merekam hasil dan kesimpulan yang dicapai.

Ada tiga tipe keputusan yang dapat direkam yaitu:

1. Rencana, bagaimana menghadapi masalah.
2. Agenda, aksi yang potensial yang sedang menunggu untuk dieksekusi.
3. Solusi, calon aksi yang akan dibangkitkan.
4. Fasilitas Penjelasan

Fasilitas penjelasan adalah komponen tambahan yang akan meningkatkan kemampuan sistem pakar. Fasilitas ini juga berguna dalam memberikan penjelasan kepada pemakai mengapa komputer meminta suatu informasi tertentu dari pemakai dan dasar apa yang digunakan komputer sehingga dapat menyimpulkan suatu kondisi.

1. Perbaikan Pengetahuan

Pakar memiliki pengetahuan untuk menganalisis dan meningkatkan kinerjanya serta kemampuan untuk belajar dari kinerjanya. Kemampuan tersebut penting dalam pembelajaran terkomputerisasi sehingga program akan mampu menganalisis penyebab kesuksesan dan kegagalan yang dialaminya.

## **Tujuan Sistem Pakar**

Pada dasarnya sistem pakar diterapkan untuk mendukung aktivitas pemecahan masalah. Beberapa aktivitas pemecahan masalah yang dimaksud seperti (Lestari, 2012):

1. **Interpretasi**. Membuat kesimpulan atau deskripsi dari sekumpulan data mentah. Pengambilan keputusan dari hasil observasi, termasuk pengenalan ucapan, analisis citra, interpretasi sinyal, dll.
2. **Prediksi**. Memproyeksikan akibat-akibat yang dimungkinkan dari situasi-situasi tertentu. Contoh: prediksi demografi, prediksi ekonomi, dll.
3. **Diagnosis**. Menentukan sebab malfungsi dalam situasi kompleks yang didasarkan pada gejala-gejala yang teramati diagnosis medis, elektronis, mekanis, dll.
4. **Perancangan(***design***)**. Menentukan konfigurasi komponen-komponen sistem yang cocok dengan tujuan-tujuan kinerja tertentu yang memenuhi kendala-kendala tertentu. Contoh: perancangan layout sirkuit, bangunan.
5. **Perencanaan**. Merencanakan serangkaian tindakan yang akan dapat mencapai sejumlah tujuan dengan kondisi awal tertentu. Contoh: perencanaan keuangan, militer, dll.
6. **Monitoring**. Membandingkan hasil pengamatan dengan kondisi yang diharapkan. Contoh: *computer aided monitoring system*.
7. **Debugging**. Menentukan dan menginterpretasikan cara-cara untuk mengatasi malfungsi. Contoh: memberikan resep obat terhadap kegagalan.
8. **Instruksi**. Mendeteksi dan mengoreksi defisiensi dalam pemahaman domain subjek. Contoh: melakukan instruksi untuk diagnosis dan debugging.
9. **Kontrol**. Mengatur tingkah laku suatu environment yang kompleks. Contoh: melakukan kontrol terhadap interpretasi, prediksi, perbaikan dan monitoring kelakukan sistem.

## **Kelebihan sistem Pakar**

Ada beberapa kelebihan ataupun keuntungan menggunakan sistem pakar dalam memecahkan permasalahan (Olanlaye, 2014), yaitu:

1. Dengan adanya sistem pakar di sebuah tempat ataupun perusahaan probabilitas dan frekuensi pengambilan keputusan yang baik menjadi tinggi.
2. Sistem pakar memungkinkan pemeriksaan secara objektivitas dengan cara mempertimbangkan keterangan dara dari pengguna tanpa memperhatikan kepribadian pengguna ataupun reaksi emosinal pengguna.
3. Sistem pakar memungkinkan seorang pakar untuk memiliki waktu luang dan lebih berkonsentrasi pada beberapa kegiatan yang bermakna lainnya.
4. Dalam kebanyakan kasus, Pengembangan sistem pakar akan mengurangi biaya pengambilan keputusan karena ketersedian sistem pakar membuat penggunaan yang tepat dan efektif dari data yang tersedia.

## **Sistem pakar dalam Penalaran**

Dalam sistem pakar, ada 2 jenis penalaran menurut Puji Sari Ramadhan & Usti Fatimah (2018:2) yaitu:

1. *Rule Base Reasoning*

Untuk *Rule Base Reasoning* merupakan bentuk penalaran yang menggunakan konsep aturan-aturan dan metode-metode diantaranya adalah:

1. *Certainly Faktor*
2. *Teorema Bayes*
3. *Demster Shafer*
4. *Euclidean Probability*
5. *Case Base Reasoning*

Dalam penalaran *Case Base Reasoning* adaalah bentuk penalaran yang menggunakan Teknik kemiripan antara kasus baru dengan kasus sebelumnya, metode-metode yang digunakan antara lain:

1. *K-Nearest Neighbor*
2. *Manhatan Distance*
3. *Minkowski Distance*

Pada pengembangan sistem pakar juga mengadopsi logika *fuzzy* yang di kenal sebagai metode yang digunakan dalam masalah perdiksi atau ketidakpasitan (Puji Sari Ramdhan & Usti Fatimah, 2018:2)

## **Bimbingan Konseling**

Menurut Tohirin (2013:25), Bimbingan dan Konseling Merupakan proses bantuan yang diberikan oleh pembimbing (*konselor*) kepada individu (*konseli*) melalui pertemuan tatap mika atau hubungan timbal balik antara keduanya, supaya *konseling* mempunyai kemampuan atau kecapakan melihat dan menemukan masalahnya serta mempunyai kemampuan memecahkan masalahnya sendiri.

Menurut Hikmawati (2011:1), Bimbingan dan Konseling adalah Pelayanan bantuan untuk peserta didik, baik secara perorangan maupun kelompok, agar mampu mandiri dan berkembang secara optimal, dalam bidang pengembangan kehidupan pribadi, kehidupan sosial, kemampuan belajar, dan perencanaan karir, melalui berbagai jenis layanan dan kegiatan pendukung berdasarkan norma-norma yang berlaku.

Berdasar uraian diatas maka penulis menyimpulkan bahwa layanan bimbingan dan konseling adalah proses pemberian bantuan kepada peserta didik yang dilakukan oleh konselor yang memgalami permasalahan pribadi, sosial, karir dan belajar melalui layanan-layanan yang terdapat dalam bimbingan dan konseling sehingga konselor mampu secara mandiri dalam menyelesaikan permasalahannya.

## **Perangkat Lunak Pendukung**

Penulis menggunakan beberapa perangkat lunak pendukung yang di gunakan dalam merancang sistem di antaranya

### ***World Wide Web* (www)**

Merupakan sekumpulan halaman yang berisikan informasi yang mengandung unsur suara, video, gambar, maupun teks. ruang informasi. Ada juga ahli yang menyatakan.

Menurut Ginanjar (2014:5), menyatakan bahwa “website adalah rangkaian atau sejumlah halaman di internet yang memiliki topik saling terkait untuk mempresentasikan suatu informasi”. Sebuah website adalah satu set halaman web terdiri dari teks, audio dan video

Menurut Edy Irwansyah dan Jurike V moniaga (2014:34) *World Wide Web* atau *web* adalah sekumpulan halaman yang menyediakan beragam infromasi bagi pengguna baik itu tidak serius hingga serius maupun informasi gratis hingga berbayar.

Menurut Jonathan Sarwono dan K Prihartono (2012:23) *World Wide Web* atau www yang berisi “halaman-halaman” yang dapat menampilkan teks, gambar, suara, animasi, serta elemen-elemen multimedia lainnya dan element-elemen yang ditampilkan bersifat interaktif.

Menurut Shelly dan Velmart (2011:80) *World Wide Web* (WWW) atau web, terdiri dari kumpulan dokumen elektronik di seluruh dunia. Setiap dokumen elektronik di *web* disebut *webpage*, yang dapat berisi teks, grafik, animasi, audio dan video.

Berdasarkan pendapat para ahli diatas, penulis dapat menyimpulkan bahwa *www* atau *world wide web* Sekumpulan halaman elektronik berisikan informasi yang mengandung teks, audio, video, maupun gambar yang dapat di akses oleh pengguna dimana pun dan kapan pun serta saling setiap halaman yang terhubung.

### ***Hypertext Preprocessor* (PHP)**

Menurut Suparno dan putratama(2016:13), php(*Hypertext Preprocessor*)adalah suatau Bahasa pemograman yang digunakan untuk menerjemahkan baris kode program menjadi kode mesin yang dapat dimengerti oleh komputer yang berbasis server-side yang dapat ditambahkan ke dalam html.

Menurut Solichin (2016:11), mengatakan bahwa PHP merupakan salah satu bahasa pemrograman berbasis web yang ditulis oleh dan untuk pengembang web.

Menurut Betha Sidik(2012:4), phpmerupakan secara umum dikenal dengan sebagai Bahasa pemograman script-script yang membuat document HTML secara *on the fly* yang dieksekusi di server webm document HTML yang dihasilkan dari sebuah aplikasi bukan document HTML yang dibuat dengan menggunakan editor teks atau editor HTML, dikenal juga sebagai Bahasa pemograman server side.

Menurut Syafii(2011d:49), *PHP* adalah pemograman interpreter yaitu proses penerjemah baris kode sumber menjadi kode mesin yang dimengerti Komputer secara langsung pada saat baris kode dijalankan.

### **HTML**

Pada Tampilan *website*, dibutuhkan nya tampilan halaman yang dapat berinteraksi dengan halaman lainnya atau sering sebut *Hypertext Markup Language* (*HTML*). Ada juga penulis menyatakan sebagai berikut:

Menurut Salichin (2016:10), Mengatakan bahwa *HTML* merupakan bahasa pemrograman web yang memberitahukan peramban web (*web browser*) bagaimana menyusun dan menyajikan konten di halaman web.

### **CSS (*Cascading Style Sheet*)**

Pada Perancangan website, dibutuhkannya Bahasa script dimana dapat membantu mempercantik website yang akan di rancang. Ada juga para penulis penulis menjelaskan pengertian dari CSS tersebut, yaitu:

Menurut Solichin (2016:10), CSS atau *cascading style sheet* bahasa pemrograman yang diusulkan oleh Hakon Wilum Lie pada tahun 1994 dan distandarisasi oleh W3C yang berfungsi untuk mempercantik tampilan web

Menurut Prasetio (2014:252) menyatakan bahwa CSS adalah suatu teknologi yang digunakan untuk memperindah tampilan halaman website (situs).

### **Javascript**

Javascript adalah bahasa pemograman berbasis *prototipe* yang berjalan di sisi klien. Ada beberapa ahli mengatakan tentang javascript, yaitu:

Menurut Prasetio (2012:300), Javascript adalah Bahasa pemograman yang digunakan untuk sebuah *website* lebih dinamis dan interaktif.

Menurut Sunyato (2007:17), *javascript* adalah Bahasa *scripting* yang popular di internet dan dapat bekerja di sebagian besar browser popular seperti *internet Exploler* (EI), *Mozilla Firefox. Netscape,* dan *Opera.*

### ***MySQL Database***

Menurut Arief (2011e:151), *MySql* (*My Stucture Query Languange*)adalah salah satu jenis database server yang sangat terkenal dan banyak di gunakan untuk membangun aplikasi web yang menggunakan database sebagai sumber dan pengelolaan datanya.

Menurut Aditnya(2011c:61), *MySql* adalah sebuah perangkat lunak sistem manajemen basisi data *SQL* (database management system) atau DBMS yang multi thread, multiuser, dengan sekitar 6 juta instalasi di seluruh dunia.

Menurut Sulhan (2007:118), MySQL merupakan perangkat lunak yang digunakan untuk membangun *database* yang sering digunakan di lingkungan linux.

*MySQL* merupakan salah satu perangkat lunak sistem pengolahan data DBMS (*Data Base Management System.*)

### ***Apache***

Wahana (2014:72),” tugas utama *Apache* adalah menghasilkan halaman web yang benar kepada *browser* klien berdasarkan *PHP* yang di tuliskan oleh pembuat halaman *website.*

### ***UML* (*Unified Modelling Language*)**

Menurut Booch (2005:7) UML adalah Bahasa standar untuk membuat rancangan *software*. UML biasanya digunakan untuk menggambarkan dan membangun, dokumen artifak dari *software* *intensive system*.

Menurut Nugroho (2009:4), UML (*Unified Modeling Language*) adalah Metodologi kolaborasi antara metoda-metoda Booch, OMT (*Object Modeling Technique*), serta OOSE (*Object Oriented Software Enggineering*) dan beberapa metoda lainnya, merupakan metodologi yang paling sering digunakan saat ini untuk analisa dan perancangan sistem dengan metodologi berorientasi objek mengadaptasi maraknya penggunaan bahasa “pemrograman berorientasi objek” (OOP).

Menurut Herlawati (2011:10), bahwa beberapa *literature* menyebutkan bahwa UML menyediakan sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa diagram yang digabung, misanya diagram komunikasi, diagram urutan dan diagram pewaktuan digabung menjadi diagram interaksi.

Berdasarkan uraian diatas maka penulis dapat menyimpulkan bahwa UML Bahasa yang dapat memberikan rekayasa permodelan visual kepada pengguna dari berbagai macam pemograman.

#### **Use Case Diagram**

Menurut Satzinger (2011: 20), *Use Case Diagram* merupakan rangkaian tindakan yang dilakukan oleh sistem, aktor mewakili *user* atau sistem lain yang berinteraksi dengan sistem yang dimodelkan.

Berdasarkan uraian di atas, penulis dapat disimpulkan bahwa sebuah rangkaian yang terkait dalam sebuah sistem tertentu yang dilakukan atau diawasi oleh sebuah aktor.

****

**Gambar 2.1** Simbol dan fungsi pada use case diagram

#### **Activity Diagram**

Menurut Martin Fowler (2005:163), *activity diagram* adalah Teknik untuk menggambarkan logika prosedular, proses bisnis, dan jalur kerja. *Node* pada sebuah *activity diagram* disebut sebagai *action*, sehingga diagram tersebut menampilkan sebuah *activity* yang tersusun dari *action*.

Menurut Angoro (2010:54), *Activity Diagram* adalah pemodelan alur kerja (*work flow*)sebuah proses bisnis dan urutan aktivitas dalam suatu proses.

*Activity Diagram* merupakan sebuah komponen yang dapat memodelkan alur kerja bersifat logika prosedular pada sebuah proses bisnis dan rangkaian yang terkait dalam sebuah sistem.

**Tabel 2.1** Simbol Serta Penjelasan Activity Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **Penjelasan** |
| 1 |  | *Activity* | Memperlihatkan bagaimana masing-masing kelas antar muka saling berinteraksi satu sama lain |
| 2 |  | *Action* | *State* dari sistem yang mencerminkan eksekusi dari suatu aksi. |
| 3 |  | *Initial Node* | Bagaimana objek dibentuk dan atau diawali. |
| 4 |  | *Activity Final Node* | Bagaimana objek dibentuk dan diakhiri. |
| 5 |  | *Decision* | Digunakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu. |
| 6 |  | *Line* | Digunakan untuk menghubungkan satu simbol dengan simbol lainnya. |

#### **Sequence Diagram**

Menurut Munawar (2005: 187), *Sequence Diagram* adalah grafik dua dimensi dimana obyek di tunjukkan dalam dimensi horizontal, sedangkan *lifeline* di tunjukkan dalam dimensi vertical.

Menurut Sukamto dan Shalahuddin (2014:165), *Sequence Diagram* menggambarkan kelakukan objek pada *use case* dengan mendeskripsikan waktu dan di terima antar objek.

Berdasarkan uraian diatas, penulis dapat memberikan kesimpulan bahwa *Squence Diagram* adalah beberapa komponen dalam *flow chart* yang dapat merepresentasikan perilaku objek pada sebuah kasus pembuatan sistem dan saling berhubungan antar objek lainnya.

#### **Class Diagram**

Menurut Satzinger (2011:28), *Diagram Class* atau Diagram Kelas sangat membantu dalam visualisasi kelas dari suatu sistem. Hal ini disebabkan karena kelas adalah deskripsi kelompok obyek-obyek dengan properti, operasi dan relasi yang sama.

Menurut Sukamto dan Shalauddin (2013:141), Diagram kelas atau *class diagram* adalah diagram yang dapat menggambarkan stuktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.

# **BAB 3 ANALISIS DAN PERANCANGAN SISTEM**

## **Identifikasi Kebutuhan Sistem**

Perancangan sistem pakar bimbingan konseling dibangun bertujuan untuk membantu mahasiswa serta konselor dalam melakukan bimbingan konseling yang sering tidak tersosialisasi karna jam kesibukan masing-masing serta masing menggunakan sistem lama dengan pencatatan data diri konseli dalam buku tamu. Maka dari itu perancangan diharapkan sistem ini dapat membantu dalam kegiatan bimbingan dan konseling yang bersifat komputasi serta mobalitas.

## **Latar Belakang objek penelitian**

Pada proses bimbingan konseling didalam kampus Universitas Advent Indonesia, selama ini dilakukan dengan pertemuan langsung atau *face to face*  yang terkadang sulit terjadi di karenakan waktu konselor yang sibuk dan juga dalam pencatatan data konseli masih secara manual yaitu dengan mencatat dalam buku tamu. Maka dari itu penulis merancang sistem pakar bimbingan konseling.

## **Proses** **sistem lama**

Kegiatan bimbingan dan konseliing saat ini masih menggunakan proses sistem lama dari pencatatan data diri konseli yang dicatat secara manual kedalam buku tamu.



**Gambar 3.1** Sistem lama Bimbingan Konselling

## **Proses** **Sistem Baru**

Perancangan sistem baru dengan memberikan kebutuhan pada setiap konseli maupun konselor dalam melakukan konseling. Perancangan sistem baru bimbingan konseling sebagai berikut:

1. Admin harus melakukan registrasi untuk membuat akun terlebih dahulu dalam sistem.
2. Admin dapat melakukan login agar masuk kedalam sistem dan juga bisa melakukan kegiatan sesuai dengan autentikasi yang dimiliki oleh masing-masing.
3. Admin dapat melakukan pengolahan data permasalahan seperti menambah, mengubah, melihat, dan menghapus.
4. Admin dapat melakukan pengolahan data Gejala Permasalahan seperti menambah, mengubah, melihat, dan menghapus.
5. Admin dapat melakukan pengolahan data Relasi seperti menambah, mengubah, melihat, dan menghapus.
6. Admin dapat melakukan registasi pengolahan data calon konselli seperti menambah, mengubah, melihat, dan menghapus.
7. Admin dapat melakukan pengolahan data Konselling konselli seperti menambah, mengubah, melihat, dan menghapus.
8. Admin dapat melakukan registrasi Bimbingan pada calon konselli.
9. Admin dapat melakukan Konselling dan juga melihat hasil dari bimbingan konselling yang dilakukan.
10. Admin dapat melakukan logout untuk keluar dari sistem.

## **Basis Pengetahuan**

Basis Pengetahuan merupakan penggabungan fakta-fakta serta logika yang menjadi landasan sebuah sistem pakar bimbingan konselling berdasarkan pengetahuan pakar.

**Tabel 3. 1** Kode Permasalahan Mahasiswa

|  |  |
| --- | --- |
| **Permasalahan Mahasiwa** | **Kode Permasalahan Mahasiswa** |
| Keluarga | KP01 |
| Karir dan pekerjaan | KP02 |
| Bullying | KP03 |
| Ekonomi dan Keuangan | KP04 |
| Diri Pribadi | KP05 |

**Tabel 3.2** Gejala Permasalahan

|  |  |
| --- | --- |
| **Kode-Gejala Permasalahan** | **Gejala Permasalahan Mahasiswa** |
| KG01 | Berubah menjadi murung, mulai pendiam serta sering menyendiri. |
| KG02 | Kurang semangat dalam proses belajar |
| KG03 | Jarang masuk kekampus atau kelas |
| KG04 | Berkurang Fokus belajar ataupun bekerja |
| KG05 | Nilai IP atau GPA mulai turun. |
| KG06 | Bingung ,ragu dan sukar dalam memahami dan menyelesaikan pelajaran |
| KG07 | Belum memahami secara keseluruhan bidang kerja yang di minati |
| KG08 | Rasa amarah/benci/takut mudah meluap dan meningkat |
| KG09 | Tidak Semangat lagi untuk melanjutkan perkuliahan |
| KG10 | Ceroboh |
| KG11 | Kurang percaya diri dan serta malu bertanya |
| KG12 | Kurang terbuka pada orang lain |
| KG13 | Takut tidak diterima dalam kelompok |
| KG14 | Takut bicara di kelas |
| KG15 | Cara mengajar dosen yang membosankan. |
| KG16 | Meragukan manfaat masuk perguruan tinggi |
| KG17 | Berkurangnya daya juang mengerjakan Tugas perkuliahan atau pekerjaan |

[http://www.kopertis3.or.id/html/wp/content/uploads/2011/04/permasalahan-mahasiswa-untuk-kopertis-wilayah-iii.pdf](http://www.kopertis3.or.id/html/wp-content/uploads/2011/04/permasalahan-mahasiswa-untuk-kopertis-wilayah-iii.pdf)

**Tabel 3. 3** tabel solusi

|  |  |
| --- | --- |
| **Kode Solusi** | **Solusi** |
| KS01 | Beri dukungan, perhatian serta doa setiap waktu kepada teman/orang tua, |
| KS02 | Certitakan/Bertanya kepada orang dewasa/orang tua/ataupun guru yang ahli di dalam permasalahan anda. |
| KS03 | Lakukan observasi diri pribadi dalam menemukan minat dan potensi anda |
| KS04 | Mulai menyusun/merencanakan pekerjaan yang anda setelah selesai kuliah. |
| KS05 | Melakukan belajar intensif secara belajar tambahan secara otodidak atau bimbingan belajar. |
| KS06 | Pertambah hubungan sosial dengan orang yang dapat membantu masalah anda |
| KS07 | Tingkatkan keberanian,Berbicara tegas,rasa percaya diri dalam kondisi apapun. |
| KS08 | Gunakan uang saku anda dengan membeli kebutuhan yang perlu nya saja Menambah hubungan kepada orang yang dapat membantu anda dalam biaya perkuliahan |
| KS09 | Mendekatkan diri kepada kepercayaan agama untuk memperkuat hati. |
| KS10 | Melakukan *self-talk* atau dapat dikatakan berbicara dengan diri sendiri yang berguna untuk meng observasi kelemahan dan kelebihan dalam diri. |
| KS11 | Tingkatkan fokus belajar dan bekerja. |
| KS12 | Bekerja paruh waktu untuk menambah uang saku dan juga membeli kebutuhan/perlengkapan belajar. |
| KS13 | Batasi dan atur kegiatan diluar kegiatan belajar agar dapat belajar lebih efektif dan fokus. |
| KS14 | Belajar teliti dan berhati-hati dalam melakukan kegiatan |

## **Aturan Konsultasi**

Aturan konsultasi merupakan bagian dari basis pengetahuan yang terdiri dari pengelompokan fakta Permasalahan dengan fakta Gejala dari sebuah kasus yang berdasarkan pengetahuan pakar.

**Table 3.4** Aturan Konsultasi

|  |  |
| --- | --- |
| Nama Permasalahan | Gejala |
| [KP01] Keluarga | [KG01] Berubah menjadi murung, mulai pendiam serta sering menyendiri |
| [KG03] Jarang masuk kekampus atau kelas |
| [KG04] Berkurang Fokus belajar ataupun bekerja |
| [KP02] Karir dan pekerjaan | [KG06] Bingung ,ragu dan sukar dalam memahami dan menyelesaikan pelajaran |
| [KG07] Belum memahami secara keseluruhan bidang kerja yang di minati |
| [KP03] Bullying | [KG01] Berubah menjadi murung, mulai pendiam serta sering menyendiri |
| [KG03] Jarang masuk kekampus atau kelas |
| [KG05] Nilai IP atau GPA mulai turun. |
| [KG04] Ekonomi dan Keuangan | [KG01] Berubah menjadi murung, mulai pendiam serta sering menyendiri |
| [KG03] Jarang masuk kekampus atau kelas |
| [KG09] Tidak Semangat lagi untuk melanjutkan perkuliahan |
| [KG05] Diri Pribadi | [KG04] Berkurang Fokus belajar ataupun bekerja |
| [KG10] Ceroboh |
| [KG11] Kurang percaya diri dan serta malu bertanya |
| [KG12] Kurang terbuka pada orang lain |
| [KG13] Takut tidak diterima dalam kelompok |
| [KG17] Berkurangnya daya juang mengerjakan Tugas perkuliahan atau pekerjaan |

## **Perancangan Sistem**

Perancangan sistem pakar bimbingan konselling yang akan dibangun, menjelaskan bagian alur sistem dengan mengimplementasikan ke dalam pemodelan yang dibutuhkan.

## **Analisis Kebutuhan Sistem**

Dalam Perancangan Sistem bimbingan konseling yang di bangun, ada beberapa kebutuhan sistem dari perangkat keras dan juga perangkat lunak, yaitu:

### **Perangkat Lunak**

Perangkat lunak yang digunakan dalam pembangunan sistem diantara nya sebagai berikut:

1. Sistem Operasi windows 10 pro
2. *Xampp 7.3.9*
3. *Php v7.3.9*
4. *Mysql v3.2.4*
5. *Laravel v 7.13*
6. *Composer v1.10.1*
7. *Sublime text v3*
8. *Git bush v2.26.2*
9. *Google chrome*
10. *Start UML v3.2.2.0*
11. *yEd Graph Editor v3.19.1.1*
12. *Balsamiq mockUp3*

### **Perangkat Keras**

Perangkat keras yang digunakan dalam membangun sistem diantaranya:

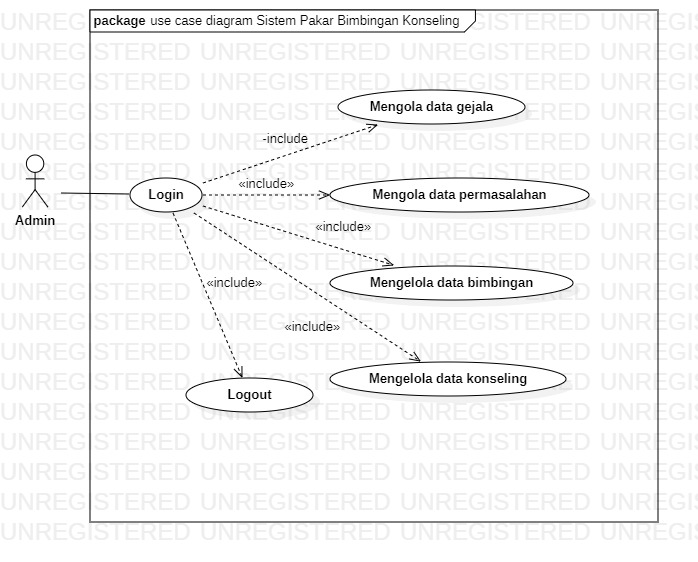
1. Laptop lenovo 80M2.
2. Memori 4096 MB.

## ***Unifed Modelling Language* (UML)**

Peracangan sistem pakar bimbingan konseling dapat dijelaskan dalam pemodelan dalam beberapa diagram-diagram yang sesuai dengan kebutuhan.

### ***Use Case Diagram***

Perancangan sistem yang akan dibangun dengan menggunakan *use case diagram* sesuai dengan kebutuhan dan apa saja yang dapat dilakukan pada aktor.



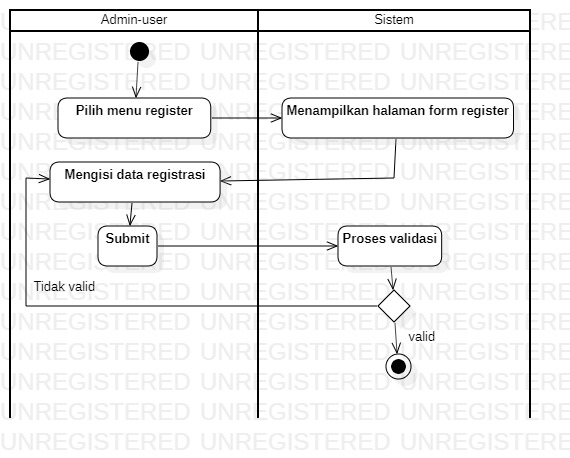
**Gambar 3. 2** Use Case Diagram

### ***Activity Diagram***

Dalam Pemodelan *activity diagram* menjelaskan bagian aktifitas-aktifitas apa saja yang dapat dilakukan.

#### **Activity Diagram Registrasi**

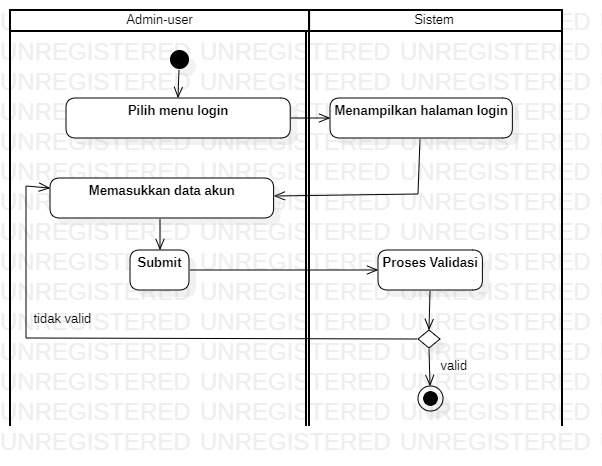
Admin memasukkan *username, Email address, password* dan *confirm* dan akan disimpan kedalam *database*.



**Gambar 3. 3** Activity Diagram Registrasi

#### **Activity Diagram Login**

Admin memasukkan *username* dan *password* pada bagian *login*. Sistem akan mengirimkan ke dalam bagian validasi untuk mencocokkan inputan dengan data di dalam *database*. Jika inputan benar maka admin atau *user* dapat masuk kedalam sistem, sebaliknya jika inputan salah maka sistem akan tetap pada halaman *login*.



**Gambar 3. 4** Diagram Activity Login

#### **Activity Diagram form Permasalahan**

Admin dapat memasukkan fakta permasalahan dan solusi, memilih gejala-gejala berdasarkan pengetahuan pakar dalam *form* permasalahan*.* Jika pada setiap *form* terisi dengan benar sesuai dengan ketentuan yang dibuat dalam sistem, data-data akan tersimpan kedalam *database*. Sebaliknya, jika pada setiap *form* tidak terisi dengan benar sistem akan memberikan peringatan dan juga tetap pada halaman *form* permasalahan serta memberikan notifikasi peringatan bagian kesalahan.



**Gambar 3.5** *Activity diagram form* permasalahan

#### **Activity diagram form Gejala**

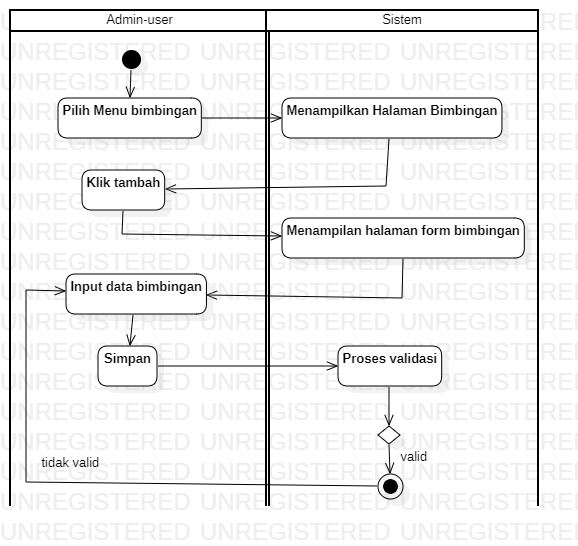
Admin dapat memasukkan fakta gejala berdasarkan pengetahuan pakar dalam *form* gejala*.* Jika pada setiap *form* terisi dengan benar sesuai dengan ketentuan yang dibuat dalam sistem, data-data akan tersimpan kedalam *database*. Sebaliknya, jika pada setiap *form* tidak terisi dengan benar sistem akan memberikan peringatan dan juga tetap pada halaman *forn* gejala.



**Gambar 3.6** Activity Diagram Form Gejala

#### **Activity Diagram form Bimbingan**

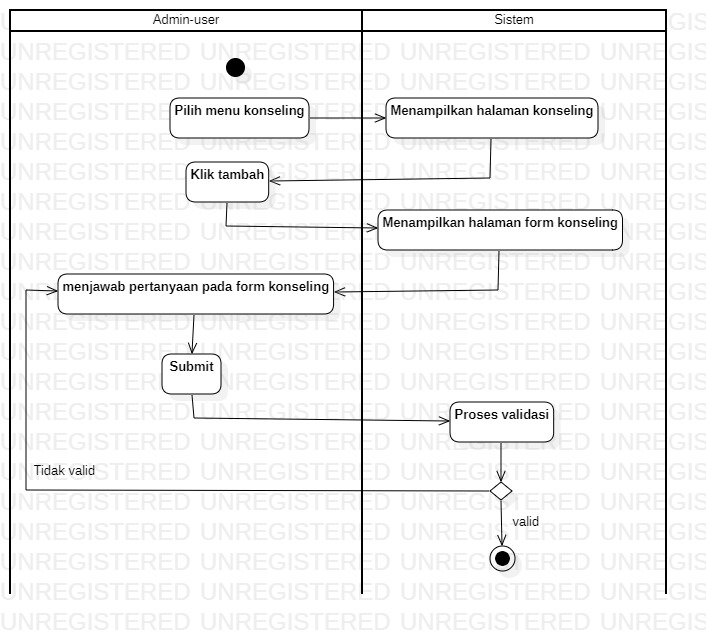
Admin dapat memasukkan data diri konselli*.* Jika pada setiap *form* terisi dengan benar sesuai dengan ketentuan yang dibuat dalam sistem, data-data akan tersimpan kedalam *database*. Sebaliknya, jika pada setiap *form* tidak terisi dengan benar sistem akan memberikan notifikasi peringatan kesalahan dan juga tetap pada halaman *form* bimbingan.



**Gambar 3. 7** Activity Diagram form Bimbingan

#### **Activity Diagram form Konselling**

Admin dapat memillih gejala-gejala pada *form* Konselling*.* Jika Admin memilih beberapa pililihan gejala, dan mengklik tombol check hasil konselling, sistem akan mencocokkan inputan pada *form* pilihan dengan data *class* konselling yang ada pada *database* dan memberikan pernyataan permasalahan, gejala dan solusi sesuai dengan nilai-nilai yang tersimpan sebelumnya. Namun jika admin tidak memilih sama sekali pada pilihan tersebut, sistem akan memberikan notifikasi peringatan kesalahan dan juga tetap pada halaman *form* konselling.

**. **

**Gambar 3. 8** Activity Diagram form Konselling

### ***Sequence Diagram***

*Sequence Diagram* merupakan diagram yang menggambarkan interaksi antara sistem dengan aktor. Berikut bagian-bagian interaksi pada sistem.

#### **Sequence Diagram Register**

Admin memasukkan data *username*, *E-mail address*, *password*, dan *config password* kedalam *form* login. Jika semua *form* inputan terisi dengan benar, sistem akan mengarahkan ke halaman *login* namum jika inputan tidak sesuai dengan persyaratan sistem akan tetap pada halam register serta memberikan notifikasi peringatan kesalahan.



**Gambar 3.9** Sequence Diagram Register

#### **Sequence Diagram form Login**

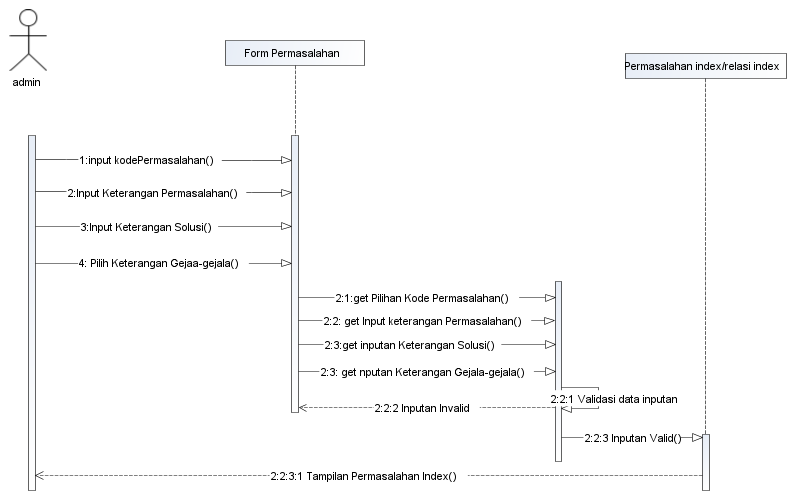
Admin memasukkan *E-mail address* dan *password* kedalam *form* *login*. Jika semua *form* inputan terisi dengan benar, menyimpan nilai serta mengarahkan kehalaman *home* atau *dashboard*. namum jika inputan tidak sesuai dengan persyaratan, sistem akan tetap pada halam *login* serta memberikan notifikasi peringatan kesalahan.



**Gambar 3. 10** Sequence Diagram Login

#### **Sequence Diagram form Permasalahan**

Admin memasukkan nilai kode permasalahan, keterangan permasalahan, solusi, memilih gejala-gejala pada *form* permasalahan. Jika semua *form* inputan terisi dengan benar, sistem akan menyimpan, menampilkan nilai serta mengarahkan kembali ke halaman permasalahan index. namum jika inputan tidak sesuai dengan persyaratan, sistem akan tetap pada halam permasalahan *create* serta memberikan notifikasi peringatan kesalahan.

** Gambar 3. 11** Sequence Diagram form Permasalahan

#### **Sequence Diagram form Gejala**

Admin dapat memilih data kode gejala, mengisi keterangan gejala kedalam *form* gejala. Jika semua *form* inputan terisi dengan benar, sistem akan menyimpan, menampilkan nilai serta mengarahkan kembali ke halaman gejala index. namum jika inputan tidak sesuai dengan persyaratan, sistem akan tetap pada halaman *form* gejala serta memberikan notifikasi peringatan kesalahan.



**Gambar 3. 12** Sequence Diagram form Gejala

#### **Sequence Diagram form Bimbingan**

Admin dapat memasukkan kode bimbingan, mengisi inputan nim, nama lengkap, Jenis kelamin, Status tempat tinggal. yang bertujuan untuk mengisi data calon konselling pada *form* bimbingan. Jika semua *form* pilih terisi dengan benar, sistem menyimpan, dan mengarahkan ke halaman *form* konselling. namum jika *form* tidak terisi sesuai dengan persyaratan, sistem akan tetap pada halaman *form* bimbingan serta memberikan notifikasi peringatan kesalahan.



**Gambar 3.13** Sequence Diagram Form Bimbingan

#### **Sequence Diagram form Konselling**

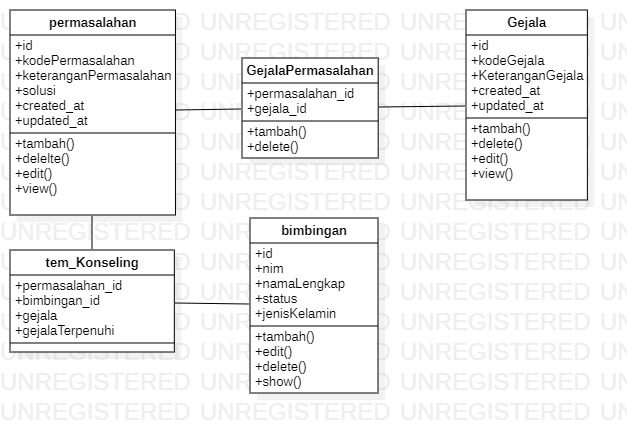
Admin dapat memilih gejala-gejala yang ada pada *form* konselling. Jika pada *form* pilih ada gejala yang terpilih dari beberapa pilihan gejala, sistem akan mencocokkan pillihan-pilahan tersebut dengan nilai yang ada pada *class* relasi di dalam *database* dan mengalihkan ke halaman konselling *result* serta menampilkan hasil dari konselling diantaranya nama permasalahan, gejala permasalahan serta solusi permasalahan. namum jika admin ataupun user tidak memilih pada pilihan gejala-gejala, sistem akan tetap pada halaman *form* konselling serta memberikan notifikasi peringatan kesalahan.



**Gambar 3.14** Sequence Diagram form Konselling

### ***Class Diagram***

*Class diagram* merupakan kumpulan beberapa implementasi dari sebuah *object* pada sebuah sistem yang saling berkaitan berdasarkan *attribute* yang dimiliki oleh setiap *class*nya. Sebagai bentuk *class-class* diagram sistem pakar bimbingan sistem pakar sebagai berikut:



**Gambar 3. 15** Class Diagram Sistem Pakar Bimbingan Konselling

## ***Database***

*Database* atau basis data merupakan tempat penyimpanan nilai-nilai dimana pada setiap *class* memiliki *attribute* yang dibutuhkan dan juga memiliki keterkaitan dengan *class* lainnya. Beberapa *class*/*table* diantaranya:

1. *table* bimbingan

*table* bimbingan berisikan *attribute* dan nilai mengenai data diri konselli

**Tabel 3. 5**  bimbingan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Value** | **Key** | **Keterangan** |
| Id | int | 10 | PK | Increment |
| kodeBimbingan | char | 255 | - | Kode Bimbingan |
| nim | char | 255 | - | Nim mahasiswa |
| namaLengkap | varchar | 255 | - | Nama konselli |
| status | varchar | 255 | - | Status tinggal |
| jenisKelamin | char | 255 | - | Jenis kelamin konselli |
| created\_at | timestamp | - | - | Waktu pembuatan data |
| updated\_at | timestamp | - | - | Waktu pengupdatean data |

1. *table* gejala

*table* gejala berisikan *attribute* dan nilai mengenai data gejala

**Table** **3.6** gejala

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Value** | **Key** | **Keterangan** |
| id | int | 10 | PK | Increment |
| kodeGejala | varchar | 10 | - | Kode gejala |
| namaGejala | varchar | 255 | - | Keterangan Gejala |
| created\_at | timestamp | - | - | Waktu pembuatan nilai |
| updated\_at | timestamp | - | - | Waktu mengupdate nilai |

1. *table* permasalahan

*table* permasalahan berisikan *attribute* dan nilai mengenai data permasalahan.

**Table 3.7**  permasalahan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Value** | **Key** | **Keterangan** |
| id | int | 10 | PK | Increment |
| kodePermasalahan | varchar | 255 | - | Kode permasalahan |
| keteranganPermaslahan | varchar | 255 | - | Keterang Permasalahan |
| solusi | text | - | - | Solusi dari permasalahan |
| created\_at | timestamp | - | - | Waktu pembuatan nilai |
| updated\_at | timestamp | - | - | Waktu mengupdate nilai |

1. *table* gejalaPermasalahan

*table* gejalaPermasalahan berisikan *attribute* dan juga merupakan tempat relasi dimana nilai antara *table* gejala *table* permasalahan saling berhubungan.

**Tabel 3. 8** gejalaPermasalahan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Value** | **Key** | **Keterangan** |
| Permasalahan\_id | int | 10 | foreignkey | Attribute terkait dengan table permasalahan |
| gejala\_id | int | 10 | foreignkey | Attribute terkait dengan table gejala |
| created\_at | timestamp | - | - | Waktu pembuatan nilai |
| update\_at | timestamp | - | - | Waktu mengupdate nilai |

1. *table* tmp\_konselling

*table* tmp\_koselling berisikan attribute dan nilai sementara yang terkait berdasarkan id antara *table* bimbingan, *table* permasalahan, *table* gejala, *table* gejala terpenuhi.

**Table 3.9** tmp\_konselling

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Value** | **Key** | **Keterangan** |
| bimbingan\_id | int | 10 | foreignkey | Attribut terkait dengan table bimbingan |
| permasalahan\_id | int | 10 | foreignkey | Attribut terkait dengan table permasalahan |
| gejala | Int | 10 | - | Atribut Nilai penampung sementara gejala |
| gejalaTerpenuhi | int | 10 | - | Attribut penghitung penampung gejala |

## **Desain *blue print* tampilan antar halaman**

Desain tampilan antar halaman merupakan gambaran *blueprint* pembentukan rancangan sistem aslinya. Berikut gambaran desain tampilan muka untuk perancangan sistem:



### **Desain *blueprint* Tampilan *form* Register**



**Gambar 3. 16** Desain blueprint Tampilan Register default laravel

### **Desain *blueprint* Tampilan halaman utama**

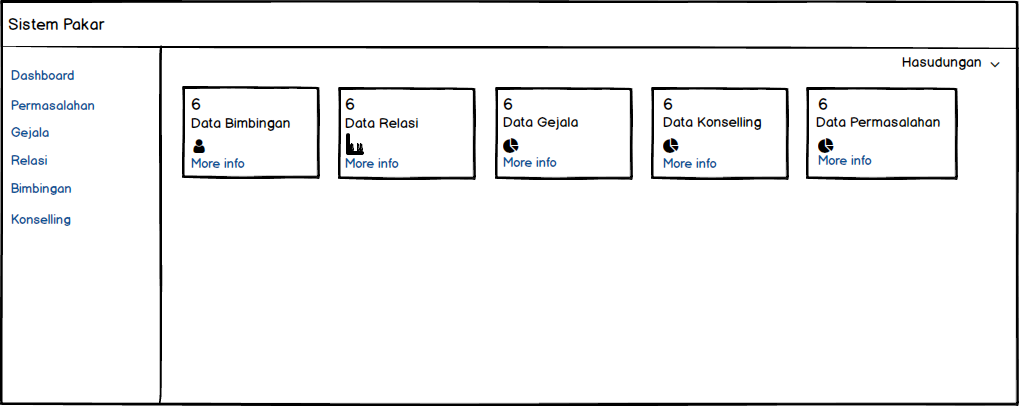
Tampilan halaman utama seperti pada gambar berikut:



**Gambar 3. 17** Desain blueprint Tampilan Utama

### **Desain *blueprint* Tampilan dashboard admin**

Tampilan dashboard admin seperti pada gambar berikut:



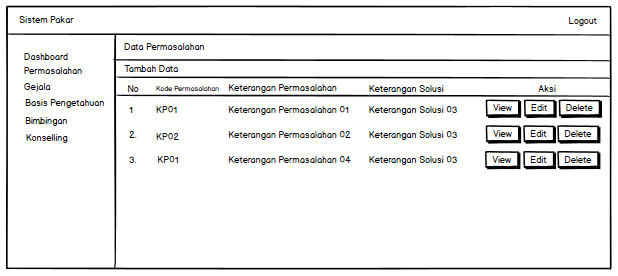
**Gambar 3.18** Desain blueprint Tampilan dashboard admin

### **Desain *blueprint* Tampilan Gejala index**



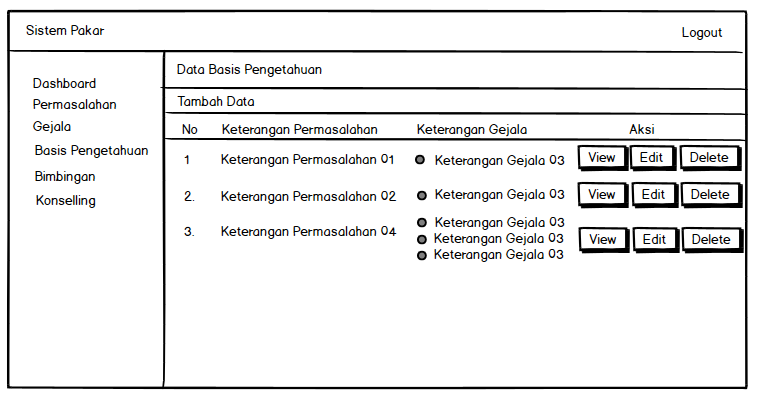
**Gambar 3. 19** Desain blueprint Tampilan Gejala index

### **Desain *blueprint* Tampilan Permasalahan index**



**Gambar 3. 20** Desain blueprint Tampilan permasalahan index

### **Desain *blueprint* Tampilan basis pengetahuan index**



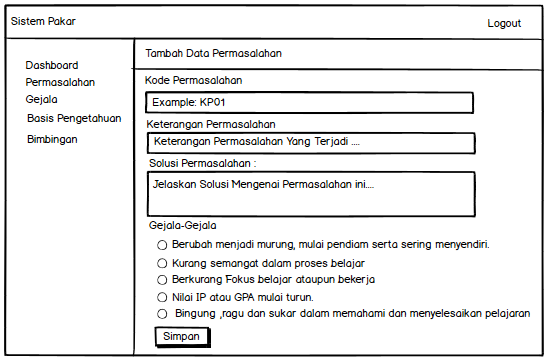
**Gambar 3. 21** Desain blueprint Tampilan basis pengetahuan index

### **Desain *blueprint* Tampilan Bimbingan index**



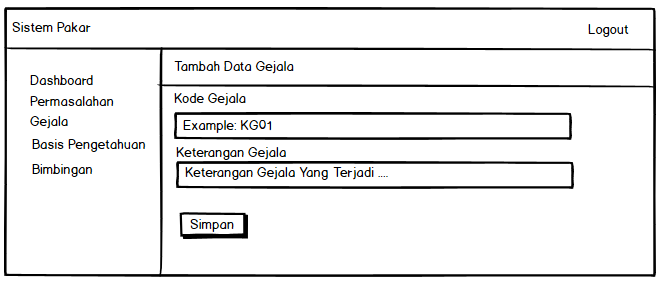
**Gambar** **3.22** *Desain blueprint* Tampilan Bimbingan index

### **Desain *blueprint* Tampilan *form* Permasalahan**



**Gambar 3.23** Desain blueprint Tampilan form Permasalahan

### **Desain *blueprint* Tampilan *form* Gejala**



**Gambar 3.24** Desain blueprint Tampilan form Gejala

### **Desain *blueprint* Tampilan *form* Bimbingan**



**Gambar 3.25** Desain blueprint Tampilan form Bimbingan

### **Desain *blueprint* Tampilan *form* Konselling**



**Gambar 3.26** Desain blueprint Tampilan form Konselling

### **Desain *blueprint* Tampilan Hasil form Konseling**

# *D:\Hasudungan\Skripsi-1\Modal Gambar BluePrint\hasil konseling.PNG*

**Tabel 3. 2** desain blueprint Tampilan hasil form konseling

# **BAB 4 HASIL PEMBAHASAN**

## **Hasil Perancangan antar muka**

Tampilan antar muka merupakan tampilan halaman-halaman *website* yang saling terhubung, tampilan sebagai berikut:

### **Halaman Utama berserta halaman *login***

Tampilan Halaman utama dengan halaman *login* sistem pakar bimbingan konselling.



**Gambar 4.1** Halaman Utama

### **Halaman Register *default* laravel**

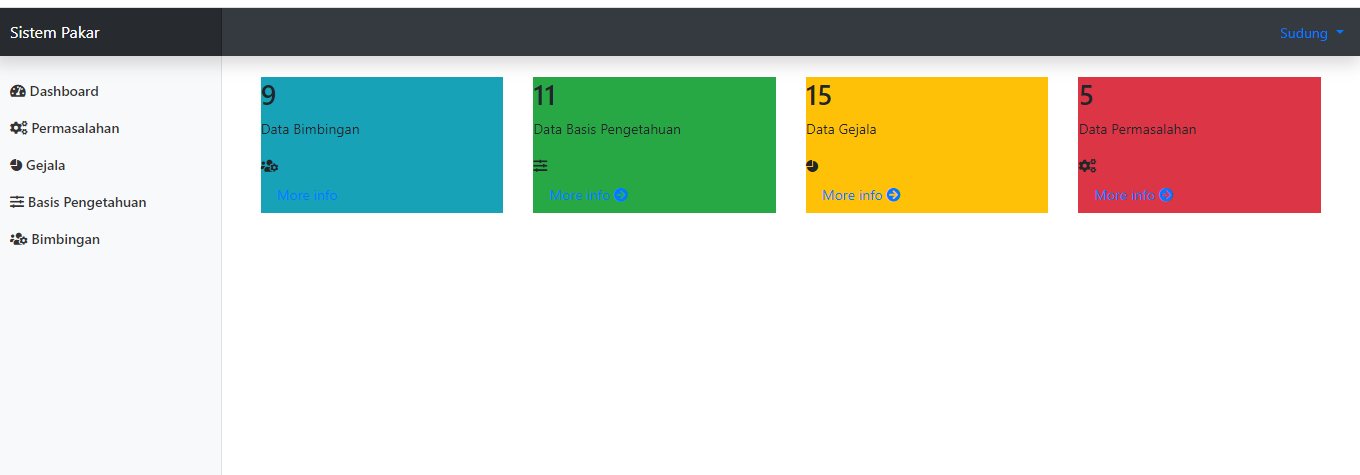
Tampilan halaman *register* *default* dari *framework* laravel ketika ingin melakukan *register* pembuatan akun admin ataupun user.



**Gambar 4.2** Register default Laravel

### **Halaman *home***

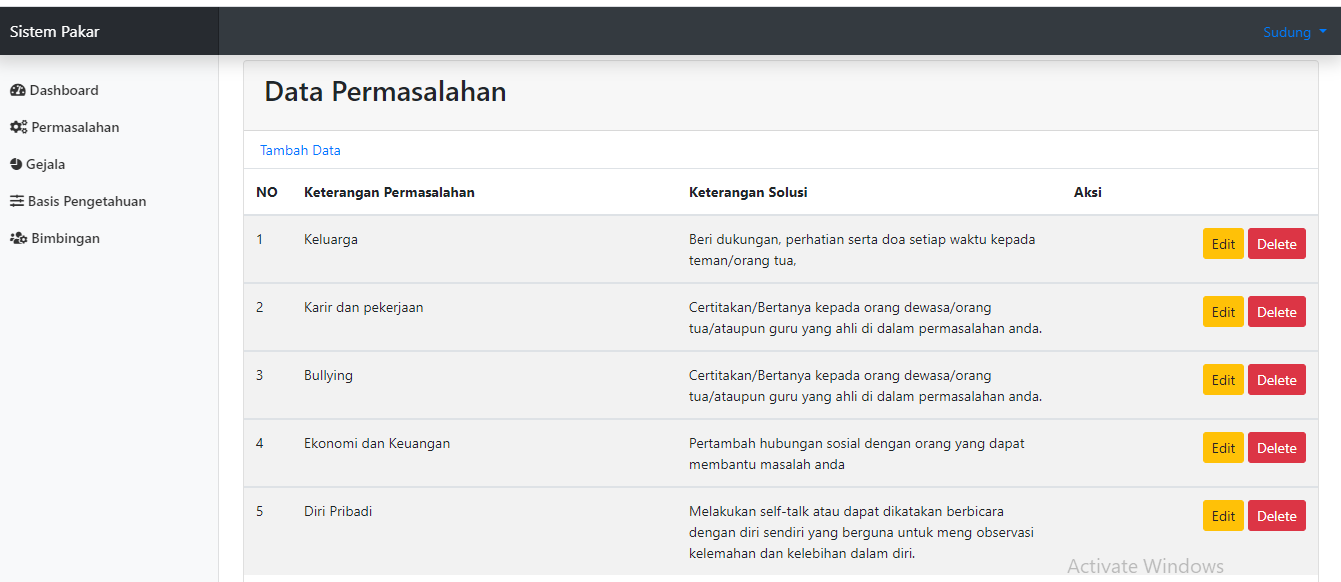
Tampilan halaman home merupakan dashboard admin dimana terdapat beberapa hasil nilai inputan pada setiap halaman autentikasi diantaranya jumlah nilai data bimbingan, relasi, gejala, konselling, dan data nilai permasalahan. admin juga dapat masuk kehalaman setiap halamannya.



**Gambar 4.3** Tampilan halaman home

### **Halaman Permasalahan index**

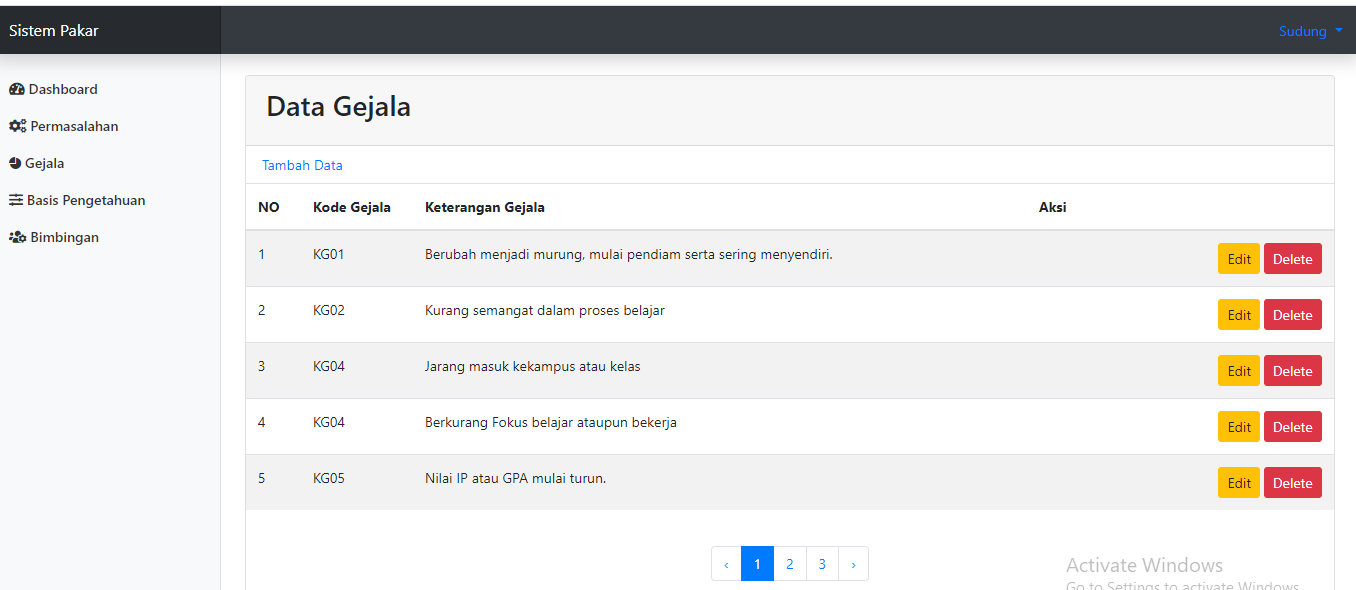
Tampilan halaman permasalahan index ketika admin sudah masuk kedalam halaman permasalahan index dimana terdapat beberapa nilai permasalahan yang telah berhasil di *input* dan juga admin dapat melakukan beberapa aksi diantaranya menghapus, melihat, dan mengedit.



**Gambar 4.4** Tampilan halaman permasalahan index

### **Halaman Gejala index**

Tampilan halaman gejala index ketika admin sudah masuk kedalam halaman gejala index dimana terdapat beberapa nilai gejala yang telah berhasil di *input* serta admin juga dapat melakukan beberapa aksi diantaranya menghapus, melihat, dan mengedit.



**Gambar 4.5** Tampilan halaman gejala index

### **Halaman basis Pengetahuan index**

Tampilan halaman relasi index ketika admin sudah masuk kedalam halaman relasi index dimana terdapat nilai-nilai relasi antara gejala dengan permasalahan yang telah di *input* serta admin dapat melakukan beberapa aksi diantaranya menghapus, melihat, dan mengedit.



**Gambar 4.6** Tampilan halaman basis pengetahuan index

### **Halaman Bimbingan index**

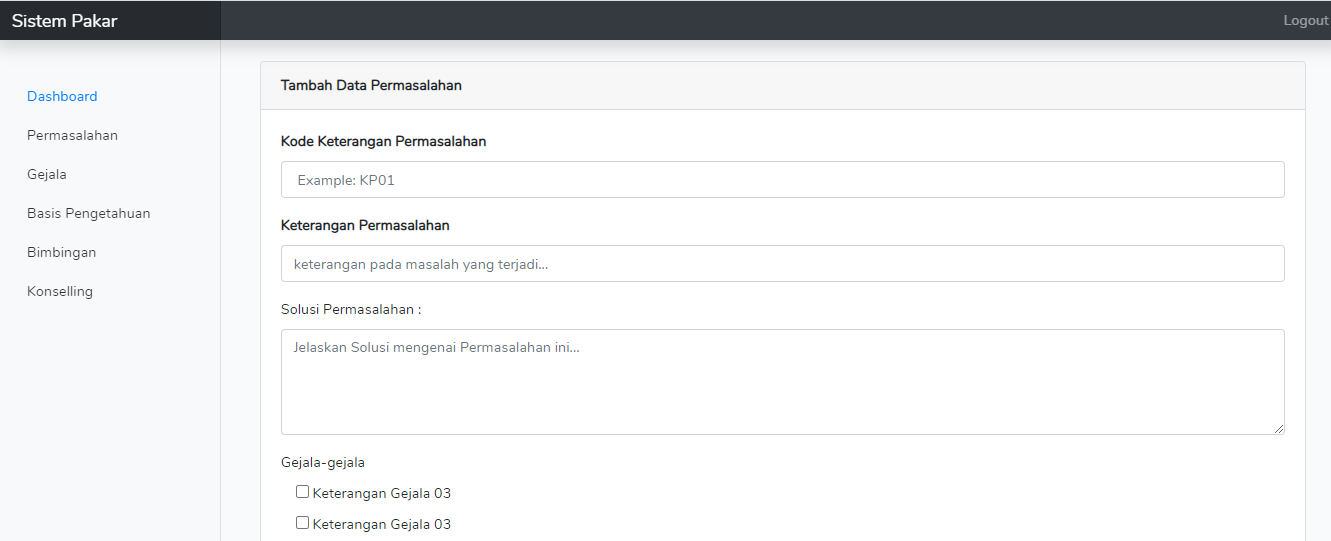
Tampilan halaman bimbingan index ketika admin sudah masuk kedalam halaman bimbingan index dimana nilai-nilai registrasi data konselli yang telah di *input* serta admin dapat melakukan beberapa aksi diantaranya menghapus, melihat, dan mengedit.



**Gambar 4.7** Tampilan halaman bimbingan index

### **Halaman *form* Permasalahan**

Tampilan halaman *form* permasalahan dimana admin memiliki autentikasi untuk memasukkan nilai permasalahan, solusi serta memilih beberapa gejala guna menjadi nilai basis pengetahuan berdasarkan pengetahuan pakar.



**Gambar 4.8** Tampilan halaman form permasalahan

### **Halaman *form* gejala**

Tampilan halaman *form* gejala dimana admin memiliki autentikasi untuk melakukan penginputan data permasalahan berdasarkan pengetahuan pakar.



**Gambar 4.9** Tampilan halaman form gejala

### **Halaman *form* bimbingan**

Tampilan halaman *form* bimbingan dimana admin maupun user dapat melakukan pendaftaran registrasi data diri calon konselli.



**Gambar 4.10** Tampilan halaman form bimbingan

### **Halaman *form* konselling**

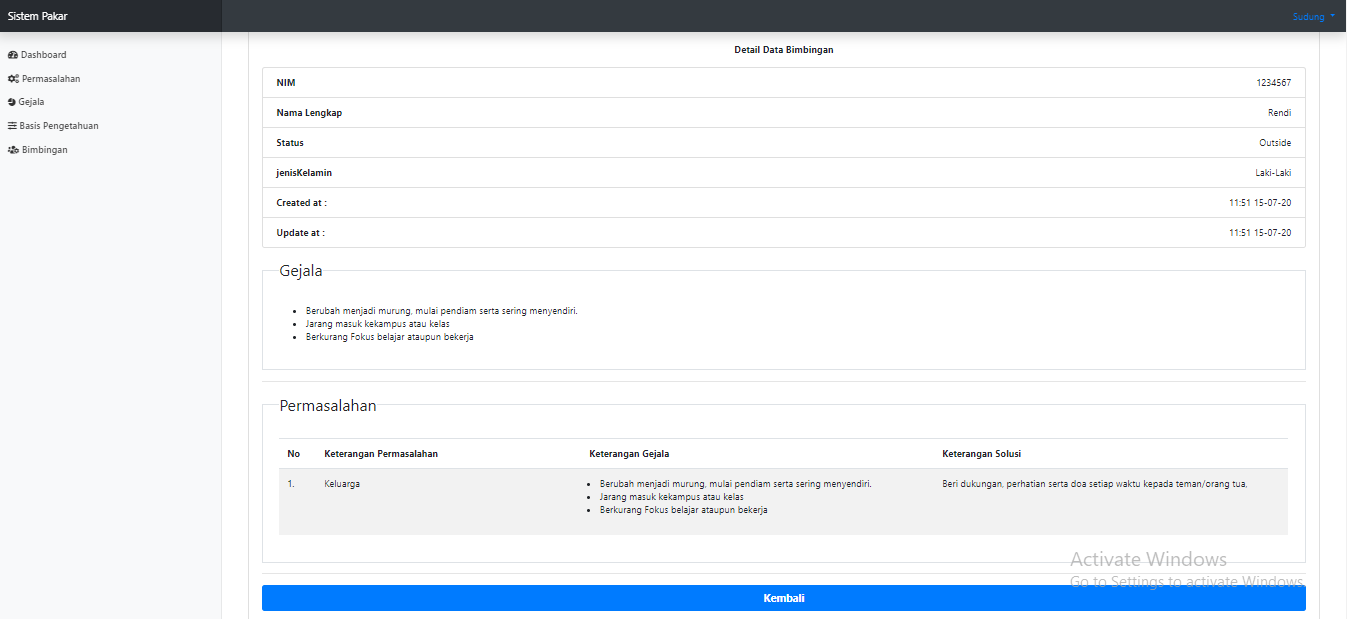
Tampilan halaman *form* konseling dimana admin ataupun user dapat melakukan memilih beberapa gejala-gejala pada *form* konselling.



**Gambar 4.11** Tampilan halaman form konselling

### **Halaman Hasil konseling**

Tampilan halaman hasil konseling menampilkan data diri konseli, permasalahan, gejala serta solusi dari permasalahan yang dialami.



**Tabel 4. 12** Halaman Hasil Konseling

## **Pengujian Komponen Sistem**

Pengujian Sistem pakar bimbingan konselling menggunakan metode *black box* guna menguji bagian sistem berjalan dengan baik. Berikut bagian komponen sistem pengujian serta metode yang dilakukan.

Tabel 4. 1 Komponen Sistem Pengujian

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Sistem yang diuji** | **Komponen Bagian Pengujian** | **Metode Pengujian** |
| 1 | Register | Tombol Register | Black Box |
| 2 | Login | Tombol Login | Black Box |
| 3 | Halaman Permasalahan | Tombol edit |  |
| Tombol delete |
| 4 | Halaman Gejala | Tombol edit | Black Box |
| Tombol delete |
| 5 | Halaman Basis Pengetahuan | Tombol view | Black Box |
| Tombol delete |
| 6 | Halaman Bimbingan | Tombol edit | Black Box |
| Tombol delete |
| **7** | Halaman Konselling | Tombol check hasil konseling | Black Box |
| **8** | Halaman Hasil Konseling | Tombol kembali | Black Box |

## **Hasil Pengujian komponen sistem**

Hasil dari pengujian setiap komponen sistem menggunakan metode black box.

**Tabel 4. 2** Hasil Pengujian Sistem

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Sistem yang di uji** | **Proses pengujian** | **Hasil yang diharapkan** | **Hasil Uji** |
| 1 | *Register* | buat nilai pada setiap *field* dengan benar | Menyimpan nilai-niali setiap inputan, beralih ke halaaman *login* | [√] Diterima  [] Ditolak |
| buat nilai pada setiap *field* dengan salah | Tetap pada halaman, beri notifikasi kesalahan | [√] Diterima  [] Ditolak |
| 2 | *Login* | buat nilai pada setiap *field* dengan benar | Mencocokkan nilai-niali setiap inputan, beralih ke halaaman *dashboard* | [√] Diterima  [] Ditolak |
| buat nilai pada setiap *field* dengan salah | Tetap pada halaman *login*, beri notifikasi kesalahan | [√] Diterima  [] Ditolak |
| 3 | Halaman Permasalahan | buat nilai pada setiap *field* dengan benar | Menyimpan nilai-nilai setiap inputan kedalam *database*, beralih ke halaman permasalahan index | [√] Diterima  [] Ditolak |
| Edit data pada nilai pada setiap field | Nilai berhasil di edit berdasarkan id dalam *database*, beralih ke halaman permasalahan index | [√] Diterima  [] Ditolak |
| hapus data pada nilai Permasalahan | Nilai berhasil di hapus | [√] Diterima  [] Ditolak |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4 | Halaman Gejala | Buat nilai pada setiap *field* dengan benar | Menyimpan nilai-nilai setiap inputan kedalam *database*, beralih ke halaman gejala index | [√] Diterima  [] Ditolak |
| Edit data pada nilai pada setiap field | Nilai berhasil di edit berdasarkan id dalam *database*, beralih ke halaman gejala index | [√] Diterima  [] Ditolak |
| Hapus data pada nilai Permasalahan | Nilai berhasil di hapus berdasarkan id dalam *database*, beralih ke halaman gejala index | [√] Diterima  [] Ditolak |
| 5 | Halaman Bimbingan | Buat nilai pada setiap *field* dengan benar | Menyimpan nilai-nilai setiap inputan kedalam *database*, beralih ke halaman gejala index | [√] Diterima  [] Ditolak |
| Edit data pada nilai pada setiap field | Nilai berhasil di edit berdasarkan id dalam *database*, beralih ke halaman gejala index | [√] Diterima  [] Ditolak |
| Hapus data pada nilai Permasalahan | Nilai berhasil di hapus berdasarkan id dalam *database*, beralih ke halaman gejala index | [√] Diterima  [] Ditolak |
| Lihat detail berdasarkan id data pada nilai permasalahan | Berhasil menampikan detail Nilai berdasarkan id pada *database* | [√] Diterima  [] Ditolak |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 6 | Halaman Basis Pengetahuan | Lihat detail nilai setiap penggabungan antaran nilai permasalahan dengan nilai gejala | Berhasil menampikan detail Nilai berdasarkan id pada *database* | [√] Diterima  [] Ditolak |
| 6 | Halaman Konseling | Memilih beberapa pilihan gejala-gejala dengan benar | Mencocokkan nilai-nilai setiap inputan pada *database*, beralih ke halaman konselling result | [√] Diterima  [] Ditolak |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 7 | Halaman hasil bimbingan konseling | Menampilkan data diri konseli, nilai permasalahan,gejala-gejala serta solusi | Berhasil menampikan detail Nilai berdasarkan id pada *database* | [√] Diterima  [] Ditolak |

# **BAB 5 KESIMPULAN DAN SARAN**

## **Kesimpulan**

Setelah melakukan pengujian sistem pakar bimbingan konselling, maka dapat di simpulkan bahwa:

1. Sistem yang dirancang dapat membantu kegiatan bimbingan dan konseling yang bersifat komputasi dan mobalitas.
2. Memberikan sistem baru pada setiap bagian kegiatan bimbingan konseling salah satunya pencatatan data konseli.

## **Saran**

Beberapa saran untuk mengembangkan aplikasi sistem pakar bimbingan konselling di kemudian hari diantaranya:

1. Menambahkan beberapa fitur sepe rti,notifikasi pada akun user, tempat kolom opini.
2. Dapat mengembangkan ke dalam basis aplikasi *smartphone.*

# **DAFTAR PUSTAKA**

Nahriyatun Na’imah Gantina Komalasari Eka Wahyuni.2016.”Gambaran Permasalaha Sosial Mahasiswa Universitas Negeri Jakarta”

Prabowo Pudjo Widodo, Herlawati, 2011, Mengunakan UML,Bandung: Bandung

Rosa A.S, M.Shalahuddin.2014.Rekayasa Perangkat Lunak-Terstuktur dan Berorientasi Objek,Bandung:Informatika

Yasmiyati.2017”Sistem Pakar *Diagnosis* Penyakit Pada Perokok Dengan Metode *Forward Chaining* Berbasis WEB”Jurnal Informatika

Muhammad Arhami, 2005. Konsep Sistem Pakar, Andi, Yogyakarta.

Kusrini,M.Kom. 2008, Aplikasi sistem pakar-Menenetukan faktor kepastian penggunaan dengan Metode Kuantigikkasi Pertanyaan,Andi,Yogyakarta.

Kusrini,S.Kom, 2006. Sistem Pakar-Teori Dan Aplikasi,Andi, Yogyakarta.

T.Sutojo S.Si,M.Kom, Edy Mulyanto S.Si,M.Kom, Dr.Vincent Suhartono.2011 Kecerdasan Buatan, Andi, Yogyakarta.

Sri Kusmadewi, 2003, Artificial Intelligence-(Teknik dan Apkliasinya), Graha Ilmu, Yogyakarta.

K.Barclay,J.Savage, 2004, Object-Oriented Desgin with UML and Java, Lincre House,Jordan Hill, Oxpord

Abdul Kadir dan Terra Ch. Triwahyuni, 2013. Pengenalan Teknologi, Andi. Yogyakarta.

Suyanto, ST,MSc 2007.Artificial Intelligence-searching, reasoning,planning and learning, Informatika, Bandung.

Adi Nugroho, 2009, Rekayasa Perangkat lunak menggunakan UML dan JAVA, ANDI, Yogyakarta.

**LAMPIRAN**

1. Bagian Routes

<?php

use Illuminate\Support\Facades\Route;

/\*

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

| Web Routes

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

|

| Here is where you can register web routes for your application. These

| routes are loaded by the RouteServiceProvider within a group which

| contains the "web" middleware group. Now create something great!

|

\*/

Route::get('/', function () {

return view('welcome');

});

Auth::routes();

Route::get('/home', 'HomeController@index')->name('home');

Route::get('/admin', function(){

return view('admin');

})->name('adminpage');

Route::get('admin-login','Auth\AdminLoginController@showLoginForm');

Route::post('admin-login', ['as' => 'admin-login', 'uses' => 'Auth\AdminLoginController@login']);

Route::get('admin-register','Auth\AdminLoginController@showRegisterPage');

Route::post('admin-register', 'Auth\AdminLoginController@register')->name('admin.register');

Route::resource('/gejala', 'GejalaController')->middleware('auth');

Route::resource('/permasalahan', 'PermasalahanController')->middleware('auth');

Route::resource('relasi', 'relasiController')->middleware('auth');

Route::resource('/bimbingan', 'BimbinganController')->middleware('auth');

// Route::resource('/solusi', 'SolusiController')

// ->middleware('auth');

Route::resource('/konselling', 'KonsellingController')->middleware('auth');

Route::resource('/bimbingan', 'BimbinganController')->middleware('auth');

1. Bagian Controller
2. **Gejala Controller**

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Models\Solusi;

use App\Models\Gejala;

use App\Models\Permasalahan;

use Illuminate\Support\Facades\DB;

class GejalaController extends Controller

{

/\*\*

\* Display a listing of the resource.

\*

\* @return \Illuminate\Http\Response

\*/

public function index()

{

$gejala = DB::table('gejala')->paginate(5);

return view('gejala.index',compact('gejala'));

}

/\*\*

\* Show the form for creating a new resource.

\*

\* @return \Illuminate\Http\Response

\*/

public function create()

{

$gejala = Gejala::all();

return view('gejala.create',compact('gejala'));

}

/\*\*

\* Store a newly created resource in storage.

\*

\* @param \Illuminate\Http\Request $request

\* @return \Illuminate\Http\Response

\*/

public function store(Request $request)

{

$this->validate($request,[

'kodeGejala' => 'required|size:4',

'keteranganGejala' => 'required'

]);

$gejala = Gejala::create($request->all());

return redirect()->route('gejala.index')->with('status', 'Data Gejala Berhasil di Tambah');

}

/\*\*

\* Display the specified resource.

\*

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function show($id)

{

$gejala = Gejala::find($id);

return view('gejala.detail', compact('gejala'));

}

/\*\*

\* Show the form for editing the specified resource.

\*

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function edit($id)

{

$gejala = Gejala::find($id);

return view('gejala.edit', compact('gejala'));

}

/\*\*

\* Update the specified resource in storage.

\*

\* @param \Illuminate\Http\Request $request

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function update(Request $request, $id)

{

$this->validate($request,[

'kodeGejala' => 'required|size:4',

'keteranganGejala' => 'required'

]);

$gejala = Gejala::find($id);

$gejala->update($request->all());

return redirect()->route('gejala.index')->with('status', 'Data Berhasil di Update');

}

/\*\*

\* Remove the specified resource from storage.

\*

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function destroy($id)

{

Gejala::find($id)->delete();

return redirect()->route('gejala.index')->with('status','Data Berhasil di Hapus');

}

}

1. **Permasalahan controller**

<?php

namespace App\Http\Controllers;

use App\Models\Permasalahan;

use App\Models\Gejala;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\DB;

class PermasalahanController extends Controller

{

/\*\*

\* Display a listing of the resource.

\*

\* @return \Illuminate\Http\Response

\*/

public function index()

{

$permasalahan =DB::table('permasalahan')->paginate(5);

return view('permasalahan.index', compact('permasalahan'));

}

/\*\*

\* Show the form for creating a new resource.

\*

\* @return \Illuminate\Http\Response

\*/

public function create()

{

$gejala = Gejala::all();

return view('permasalahan.Create', compact('gejala'));

}

/\*\*

\* Store a newly created resource in storage.

\*

\* @param \Illuminate\Http\Request $request

\* @return \Illuminate\Http\Response

\*/

public function store(Request $request)

{

$permasalahan = new Permasalahan();

$this->validate($request,[

'keteranganPermasalahan' => 'required',

'solusi' => 'required',

'gejala' => 'required'

]);

$permasalahan = Permasalahan::create($request->all());

$permasalahan->save();

foreach ($request->gejala as $gejala\_id) {

$permasalahan->attachGejala($gejala\_id);

}

// dd($permasalahan);

return redirect()->route('permasalahan.index')->with('status','Berhasil Di Tambah Kedalam baseKnowledge');

}

/\*\*

\* Display the specified resource.

\*

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function show($id)

{

$permasalahan = Permasalahan::find($id);

return view('permasalahan.detail', compact('permasalahan'));

}

/\*\*

\* Show the form for editing the specified resource.

\*

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function edit(Permasalahan $permasalahan)

{

$gejala =Gejala::all();

return view('permasalahan.edit', compact('gejala','permasalahan'));

}

/\*\*

\* Update the specified resource in storage.

\*

\* @param \Illuminate\Http\Request $request

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function update(Request $request, $id)

{

$permasalahan = new Permasalahan();

$this->validate($request,[

'keteranganPermasalahan' => 'required',

'solusi' => 'required'

]);

$permasalahan = Permasalahan::create($request->all());

// Menempelkan nilai baru

foreach ($request->gejala as $gejala\_id) {

$permasalahan->attachGejala($gejala\_id);

}

// Memisahkan/Menggantikan nilai

foreach ($request->gejala as $gejala\_id) {

$permasalahan->detachGejala($gejala\_id);

}

// dd($permasalahan);

$permasalahan->save();

return redirect()->route('permasalahan.index')->with('status','Berhasil Di Update Kedalam basePengetahuan');

}

/\*\*

\* Remove the specified resource from storage.

\*

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function destroy(Permasalahan $permasalahan)

{

$permasalahan->delete();

foreach ($permasalahan->gejala as $gejala) {

$permasalahan->detachGejala($gejala->id);

}

return redirect()->route('permasalahan.index')->with('status','Data Berhasil di Hapus dari basePengetahuan');

}

}

1. **Bimbingan controller**

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Models\Bimbingan;

use App\Models\Gejala;

use App\Models\Relasi;

use Illuminate\Support\Facades\DB;

class BimbinganController extends Controller

{

/\*\*

\* Display a listing of the resource.

\*

\* @return \Illuminate\Http\Response

\*/

public function index()

{

$bimbingan = DB::table('bimbingan')->paginate(5);

return view('bimbingan.index',compact('bimbingan'));

}

/\*\*

\* Show the form for creating a new resource.

\*

\* @return \Illuminate\Http\Response

\*/

public function create()

{

$bimbingan = Bimbingan::all();

return view('bimbingan.create',compact('bimbingan'));

}

/\*\*

\* Store a newly created resource in storage.

\*

\* @param \Illuminate\Http\Request $request

\* @return \Illuminate\Http\Response

\*/

public function store(Request $request)

{

$this->validate($request,[

'nim' => 'required|size:7',

'namaLengkap' => 'required',

'status' => 'required',

'jenisKelamin' => 'required'

]);

$bimbingan = Bimbingan::create($request->all());

return redirect()->route('konselling.create', ['id' => $bimbingan->id])->with('status', 'Mulai Konselling');

}

/\*\*

\* Display the specified resource.

\*

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function show($id)

{

$bimbingan = Bimbingan::find($id);

$permasalahan = $bimbingan->permasalahan()->whereRaw('gejala = gejala\_terpenuhi')->get();

// dd($permasalahan);

return view('bimbingan.detail', compact('bimbingan', 'permasalahan'));

}

/\*\*

\* Show the form for editing the specified resource.

\*

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function edit($id)

{

$bimbingan = Bimbingan::find($id);

return view('bimbingan.edit', compact('bimbingan'));

}

/\*\*

\* Update the specified resource in storage.

\*

\* @param \Illuminate\Http\Request $request

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function update(Request $request, $id)

{

$this->validate($request,[

'kodeBimbingan' => 'required|size:4',

'nim' => 'required|size:7',

'namaLengkap' => 'required',

'status' => 'required',

'jenisKelamin' => 'required'

]);

$bimbingan = Bimbingan::find($id);

$bimbingan->update($request->all());

return redirect()->route('konselling.create')->with('status', 'Mulai Konselling');

}

/\*\*

\* Remove the specified resource from storage.

\*

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function destroy($id)

{

//

}

}

1. **KonselingController**

<?php

namespace App\Http\Controllers;

use App\Models\Konselling;

use App\Models\Permasalahan;

use App\Models\Bimbingan;

use App\Models\Gejala;

use App\Models\Relasi;

use App\Models\TempKonselling;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\DB;

class KonsellingController extends Controller

{

/\*\*

\* Display a listing of the resource.

\*

\* @return \Illuminate\Http\Response

\*/

public function index()

{

$konselling =DB::table('konselling')->paginate(5);

$relasi = Relasi::all();

$gejala = Gejala::all();

$permasalahan = Permasalahan::all();

return view('Konselling.index', compact('konselling', 'relasi', 'gejala', 'permasalahan'));

}

/\*\*

\* Show the form for creating a new resource.

\*

\* @return \Illuminate\Http\Response

\*/

private function selectGejala($bimbingan\_id) {

$gejala = Gejala::all();

return view('konselling.create', compact('gejala', 'bimbingan\_id'));

}

public function create()

{

// $bimbingan\_id = DB::table('tmpGejala')->get();

$bimbingan\_id = $\_GET['id'] ?? 0;

$gejala = Gejala::all();

return view('konselling.create', compact('gejala', 'bimbingan\_id'));

}

/\*\*

\* Store a newly created resource in storage.

\*

\* @param \Illuminate\Http\Request $request

\* @return \Illuminate\Http\Response

\*/

public function store(Request $request)

{

$this->validate($request, [

'gejala' => 'required|min:2'

],

[ 'gejala.min' => 'Gejala yang di pilih minimal 2',

'gejala.required' => 'Gejala wajib dipilih']

);

$bimbingan\_id = $request->bimbingan\_id;

$bimbingan = Bimbingan::find($bimbingan\_id);

// dd($bimbingan\_id);

foreach ($request->gejala as $gejala\_id) {

// dd($gejala\_id);

$pasien = $bimbingan->attachGejala($gejala\_id);

// var\_dump($pasien);

$gejala = Gejala::find($gejala\_id);

foreach ($gejala->permasalahan as $permasalahan) {

$temp\_konselling = TempKonselling::where('bimbingan\_id', $bimbingan\_id)->where('permasalahan\_id', $permasalahan->id);

$temp\_diag = $temp\_konselling->first();

if (!$temp\_diag) {

$temp\_diag = new TempKonselling;

$temp\_diag->bimbingan\_id = $bimbingan\_id;

$temp\_diag->permasalahan\_id = $permasalahan->id;

$temp\_diag->gejala = count($permasalahan->gejala);

$temp\_diag->gejala\_terpenuhi = 1;

$temp\_diag->save();

} else {

$temp\_diag = $temp\_konselling->update(['gejala\_terpenuhi' => $temp\_diag->gejala\_terpenuhi + 1]);

}

}

}

return redirect()->route('bimbingan.show', $bimbingan->id)->with('status', 'konseling selesai');

}

/\*\*

\* Display the specified resource.

\*

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function show($id)

{

//

}

/\*\*

\* Show the form for editing the specified resource.

\*

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function edit($id)

{

//

}

/\*\*

\* Update the specified resource in storage.

\*

\* @param \Illuminate\Http\Request $request

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function update(Request $request, $id)

{

//

}

/\*\*

\* Remove the specified resource from storage.

\*

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function destroy($id)

{

//

}

}

1. **Bagian Model / class**
2. **Gejala**

<?php

namespace App\Models;

use App\Models\Relasi;

use App\Models\Solusi;

use App\Models\Bimbingan;

use App\Models\Permasalahan;

use Illuminate\Database\Eloquent\Model;

class Gejala extends Model

{

protected $guarded = [];

protected $table = 'gejala';

protected $fillable = ['kodeGejala','keteranganGejala'];

public function permasalahan()

{

return $this->belongsToMany(Permasalahan::class, 'gejalapermasalahan');

// return $this->belongsToMany(Permasalahan::class, 'basepengetahuan');

}

public function bimbingan() {

return $this->belongsToMany(Bimbingan::class, 'tmpGejala');

}

public function relasi()

{

return $this->belongsToMany(Relasi::class, 'basepengetahuan');

}

}

1. **Permasalahan**

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

use App\Models\Solusi;

use App\Models\Gejala;

class Permasalahan extends Model

{

protected $table = 'permasalahan';

protected $guarded = ['id', 'created\_at', 'updated\_at'];

protected $fillable = ['kodePermasalahan','keteranganPermasalahan', 'solusi'];

public function gejala()

{

return $this->belongsToMany(Gejala::class, 'gejalaPermasalahan')->withTimestamps();

}

public function attachGejala($gejala\_id)

{

$gejala = Gejala::find($gejala\_id);

return $this->gejala()->attach($gejala);

}

public function detachGejala($gejala\_id)

{

$gejala = Gejala::find($gejala\_id);

return $this->gejala()->detach($gejala);

}

public function solusi()

{

return $this->hasMany(Solusi::class);

}

}

1. **gejalaPermasalahan**

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class GejalaPermasalahan extends Model

{

protected $table = 'gejalaPermasalahan';

protected $fillable = ['permasalahan\_id', 'gejala\_id'];

}

1. **TempKonseling**

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class TempKonselling extends Model

{

protected $table = 'tmp\_konselling';

protected $fillable = ['bimbingan\_id', 'permasalahan\_id', 'gejala', 'gejalaTerpenuhi'];

public $timestamps = false;

}

1. **Bimbingan**

<?php

namespace App\Models;

use App\Models\Relasi;

use App\Models\Gejala;

use Illuminate\Database\Eloquent\Model;

class Bimbingan extends Model

{

protected $table = 'bimbingan';

protected $fillable = ['nim', 'namaLengkap', 'status', 'jenisKelamin'];

protected $guarded = [];

// public function rules()

// {

// return $this->belongsTo(Relasi::class);

// }

public function gejala()

{

return $this->belongsToMany(Gejala::class, 'tmpgejala');

}

public function permasalahan()

{

return $this->belongsToMany(Permasalahan::class, 'tmp\_konselling');

}

public function attachGejala($gejala\_id)

{

$gejala = Gejala::find($gejala\_id);

return $this->gejala()->attach($gejala);

}

}

1. **Bagian view**
2. Auth

* Login.blade.php

@extends('layouts.app')

@section('content')

<div class="wrapper wrapper-full-page">

<div class="page-header login-page header-filter">

<div class="container">

<div class="row align-items-center">

<div class="col-lg-4 col-md-6 col-sm-8 ml-auto mr-auto">

<div class="card card-login mb-3">

<div class="card-header card-header-primary text-center">

<h4 class="card-title">

<strong> sistem Pakar</strong> <h5>Bimbingan Konselling</h5>

</h4>

</div>

<div class="card-body login-card-body">

@if(\Session::has('message'))

<p class="alert alert-info">

{{ \Session::get('message') }}

</p>

@endif

<form action="{{ route('login') }}" method="POST">

{{ csrf\_field() }}

<div class="form-group">

<input type="email" class="form-control{{ $errors->has('email') ? ' is-invalid' : '' }}" required autofocus placeholder="email" name="email" value="{{ old('email', null) }}">

@if($errors->has('email'))

<div class="invalid-feedback">

{{ $errors->first('email') }}

</div>

@endif

</div>

<div class="form-group">

<input type="password" class="form-control{{ $errors->has('password') ? ' is-invalid' : '' }}" required placeholder="password" name="password">

@if($errors->has('password'))

<div class="invalid-feedback">

{{ $errors->first('password') }}

</div>

@endif

</div>

<div class="row">

<div class="col-md-12">

<div class="icheck-primary">

<input type="checkbox" name="remember" id="remember">

<label for="remember">remember me</label>

</div>

</div>

</div>

<div class="card-footer justify-content-center">

<button type="submit" class="btn btn-primary btn-lg">login</button>

<a href="{{ url('/register') }}">belum punya akun?</a>

</div>

</form>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

@endsection

* Register.blade.php

@extends('layouts.app')

@section('content')

<div class="container">

<div class="row justify-content-center">

<div class="col-md-8">

<div class="card">

<div class="card-header">{{ \_\_('Register') }}</div>

<div class="card-body">

<form method="POST" action="{{ route('register') }}">

@csrf

<div class="form-group row">

<label for="name" class="col-md-4 col-form-label text-md-right">{{ \_\_('Name') }}</label>

<div class="col-md-6">

<input id="name" type="text" class="form-control @error('name') is-invalid @enderror" name="name" value="{{ old('name') }}" required autocomplete="name" autofocus>

@error('name')

<span class="invalid-feedback" role="alert">

<strong>{{ $message }}</strong>

</span>

@enderror

</div>

</div>

<div class="form-group row">

<label for="email" class="col-md-4 col-form-label text-md-right">{{ \_\_('E-Mail Address') }}</label>

<div class="col-md-6">

<input id="email" type="email" class="form-control @error('email') is-invalid @enderror" name="email" value="{{ old('email') }}" required autocomplete="email">

@error('email')

<span class="invalid-feedback" role="alert">

<strong>{{ $message }}</strong>

</span>

@enderror

</div>

</div>

<div class="form-group row">

<label for="password" class="col-md-4 col-form-label text-md-right">{{ \_\_('Password') }}</label>

<div class="col-md-6">

<input id="password" type="password" class="form-control @error('password') is-invalid @enderror" name="password" required autocomplete="new-password">

@error('password')

<span class="invalid-feedback" role="alert">

<strong>{{ $message }}</strong>

</span>

@enderror

</div>

</div>

<div class="form-group row">

<label for="password-confirm" class="col-md-4 col-form-label text-md-right">{{ \_\_('Confirm Password') }}</label>

<div class="col-md-6">

<input id="password-confirm" type="password" class="form-control" name="password\_confirmation" required autocomplete="new-password">

</div>

</div>

<div class="form-group row mb-0">

<div class="col-md-6 offset-md-4">

<button type="submit" class="btn btn-primary">

{{ \_\_('Register') }}

</button>

</div>

</div>

</form>

</div>

</div>

</div>

</div>

</div>

@endsection

1. Gejala

* Create.blade.php

@extends('layouts.master')

@section('content')

<div class="container">

<div class="card mt-5">

<div class="card-header text-left">

<strong>Tambah Data gejala</strong>

</div>

@if($errors->all())

@include('layouts.error')

@endif

<div class="card-body">

<form method="POST" action="{{route('gejala.store')}}">

@csrf

<div class="form-group">

<label for="kodeGejala"><strong> Kode Keterangan Gejala </strong></label>

<input type="text" name="kodeGejala" value="{{old('kodeGejala')}}" placeholder=" Example: KG01" class="form-control" id="kodeGejala" >

</div>

<div class="form-group">

<label for="keteranganGejala"><strong> Keterangan Gejala</strong></label>

<input type="text" name="keteranganGejala" placeholder="Keterangan pada Permasalahan..." value="{{old('keteranganGejala')}}" class="form-control " id="keteranganGejala" >

</div>

<div class="form-group">

<input type="submit" class="btn btn-success" value="Simpan">

</div>

</form>

</div>

</div>

</div>

@stop

* Index.blade.php

@extends('layouts.master')

@section('content')

<section class="content">

<div class="card">

<div class="card-header">

<h3 class="card-title">Data Gejala</h3>

</div>

<li class="nav-item d-none d-sm-inline-block">

<a href="{{route('gejala.create')}}" class="nav-link">Tambah Data</a>

</li>

@if(session('status'))

<div class="alert alert-success" role="alert">

{{session('status')}}

</div>

@endif

<div class="card-body p-0">

<table class="table table-striped projects">

<thead>

<tr>

<th style="width: 1%"> NO </th>

<th style="width: 5%"> Kode Gejala </th>

<th style="width: 30%"> Keterangan Gejala </th>

<th style="width: 15%" style="text-decoration: center"> Aksi </th>

</tr>

</thead>

<tbody>

@foreach($gejala as $g)

<tr>

<td> {{ $loop->iteration }} </td>

<td> {{$g->kodeGejala}} </td>

<td> {{ $g->keteranganGejala }} </td>

<td class="project-actions text-right">

<a class="btn btn-warning btn-sm" href="{{route('gejala.edit', $g->id)}}"> Edit </a>

<form action="{{route('gejala.destroy',$g->id)}}" method="post" style="display: inline-block;">

@csrf

@method('DELETE')

<input type="submit" name="" value="Delete" class="btn btn-danger btn-sm">

</form>

</td>

</tr>

@endforeach

</tbody>

</table>

</div>

<div class="pagination justify-content-center" style="margin:20px 0">

{{$gejala->links()}}

</div>

</div>

</section>

@endsection

* Edit.blade.php

@extends('layouts.master')

@section('content')

<div class="container">

<div class="card mt-5">

<div class="card-header text-left">

<strong>Edit Data gejala</strong>

</div>

<div class="card-body">

<form method="POST" action="{{route('gejala.update',[$gejala->id])}}">

@csrf

@method('PUT')

<div class="form-group">

<label for="kodeGejala"><strong> Kode Keterangan Gejala </strong></label>

<input type="text" name="kodeGejala" value="{{$gejala->kodeGejala}}" class="form-control" id="kodeGejala" >

</div>

<div class="form-group {{ $errors->has('keteranganGejala') ? 'has-error' : '' }}">

<label for="keteranganGejala"><strong> Keterangan Gejala</strong></label>

<input type="text" id="keteranganGejala" name="keteranganGejala" class="form-control" value="{{$gejala->keteranganGejala }}" require>

@if($errors->has('keteranganGejala'))

<p class="help-block">

{{ $errors->first('keteranganGejala') }}

</p>

@endif

</div>

<div class="form-group">

<input type="submit" class="btn btn-success" value="Simpan">

</div>

</form>

</div>

</div>

</div>

@stop

* Detail.blade.php

@extends('layouts.master')

@section('content')

<section class="content">

<div class="container-fluid">

<div class="row">

<div class="col">

<div class="card card-primary card-outline">

<div class="card-body box-profile">

<div class="text-center"><p><strong>Detail Data Gejala</strong></p>

</div>

<ul class="list-group list-group-unbordered mb-3">

<li class="list-group-item">

<b>Kode Gejala</b> <a class="float-right">{{ $gejala->kodeGejala }}</a>

</li>

<li class="list-group-item">

<b>Keterangan Gejala</b> <a class="float-right">{{ $gejala->keteranganGejala }}</a>

</li>

<li class="list-group-item">

<b>Created at :</b> <a class="float-right">{{$gejala->created\_at->format(' H:i d-m-y ')}}</a>

</li>

<li class="list-group-item">

<b>Update at :</b> <a class="float-right">{{ $gejala->updated\_at->format(' H:i d-m-y ') }}</a>

</li>

</ul>

<a href="{{route('gejala.index')}}" class="btn btn-primary btn-block"><b>Kembali</b></a>

</div>

</div>

</div>

</div>

</div>

</section>

@endsection

1. Permasalahan

* Create.blade.php

@extends('layouts.master')

@section('content')

<div class="container">

<div class="card mt-5">

<div class="card-header text-left"><strong>Tambah Data Permasalahan</strong>

</div>

@if($errors->all())

@include('layouts.error')

@endif

<div class="card-body">

<form method="POST" action="{{route('permasalahan.store')}}">

@csrf

<div class="form-group">

<label for="kodePermasalahan"><strong> Kode Keterangan Permasalahan </strong></label>

<input type="text" name="kodePermasalahan" value="{{old('kodePermasalahan')}}" placeholder=" Example: KP01" class="form-control" id="kodePermasalahan" >

</div>

<div class="form-group">

<label for="keteranganPermasalahan"><strong> Keterangan Permasalahan </strong></label>

<input type="text" name="keteranganPermasalahan" value="{{old('keteranganPermasalahan')}}" placeholder="keterangan pada masalah yang terjadi..." class="form-control" id="keteranganPermasalahan" >

</div>

<div class="form-group">

<label>Solusi Permasalahan :</label>

<textarea name="solusi" class="form-control" value="{{old('solusi')}}" rows="4" placeholder="Jelaskan Solusi mengenai Permasalahan ini...">

</textarea>

</div>

<div class="form-group">

<label>Gejala-gejala </label>

<div class="col-md-12">

@foreach ($gejala as $gejala)

<div class="checkbox">

<label><input class="flat" type="checkbox" name="gejala[]" value="{{ $gejala->id }}"> {{ $gejala->keteranganGejala }} </label>

</div>

@endforeach

</div>

</div>

<div class="card-footer">

<input type="submit" class="btn btn-success" value="Simpan">

</div>

</form>

</div>

</div>

</div>

@stop

* Detail.blade.php

@extends('layouts.master')

@section('content')

<section class="content">

<div class="container-fluid">

<div class="row">

<div class="col">

<div class="card card-primary card-outline">

<div class="card-body box-profile">

<div class="text-center"><p><strong>Detail Data permasalahan</strong></p>

</div>

<ul class="list-group list-group-unbordered mb-3">

<li class="list-group-item">

<b>Keterangan permasalahan</b> <a class="float-right">{{ $permasalahan->keteranganPermasalahan }}</a>

</li>

<li class="list-group-item">

<b>Keterangan Solusi</b> <a class="float-right">{{ $permasalahan->solusi }}</a>

</li>

<li class="list-group-item">

<b>Created at :</b> <a class="float-right">{{$permasalahan->created\_at->format(' H:i d-m-y ')}}</a>

</li>

<li class="list-group-item">

<b>Update at :</b> <a class="float-right">{{ $permasalahan->updated\_at->format(' H:i d-m-y ') }}</a>

</li>

</ul>

<a href="{{route('permasalahan.index')}}" class="btn btn-primary btn-block"><b>Kembali</b></a>

</div>

</div>

</div>

</div>

</div>

</section>

@endsection

* Edit.blade.php

@extends('layouts.master')

@section('content')

<div class="container">

<div class="card mt-5">

<div class="card-header text-left"><strong>Edit Data Permasalahan</strong>

</div>

@if($errors->all())

@include('layouts.error')

@endif

<div class="card-body">

<form method="POST" action="{{route('permasalahan.update',[$permasalahan->id])}}">

@csrf

@method('PUT')

<div class="form-group">

<label for="kodePermasalahan"><strong> Kode Keterangan Permasalahan </strong></label>

<input type="text" name="kodePermasalahan" value="{{$permasalahan->kodePermasalahan}}"

class="form-control" id="kodePermasalahan" >

</div>

<div class="form-group">

<label for="keteranganPermasalahan"><strong> Keterangan Permasalahan </strong></label>

<input type="text" name="keteranganPermasalahan" value="{{$permasalahan->keteranganPermasalahan}}"

class="form-control" id="keteranganPermasalahan" >

</div>

<div class="form-group">

<label>Solusi Permasalahan :</label>

<textarea name="solusi" class="form-control" rows="4" > {{$permasalahan->solusi}}

</textarea>

</div>

<div class="form-group">

<label>Gejala-gejala </label>

<div class="col-md-12">

@foreach ($gejala as $g)

<div class="checkbox">

<label><input class="flat" type="checkbox" name="gejala[]" value="{{ $g->id }}">

{{ $g->keteranganGejala }}

</label>

</div>

@endforeach

<script>

@foreach ($permasalahan->gejala as $gejala)

$('.check[value={{ $gejala->id }}]').attr('checked', true);

@endforeach()

</script>

</div>

</div>

<div class="card-footer">

<input type="submit" class="btn btn-success" value="Simpan">

</div>

</div>

</form>

</div>

</div>

</div>

@stop

* Index.blade.php

@extends('layouts.master')

@section('content')

<section class="content">

<div class="card">

<div class="card-header">

<h3 class="card-title">Data Permasalahan</h3>

</div>

<li class="nav-item d-none d-sm-inline-block">

<a href="{{route('permasalahan.create')}}" class="nav-link">Tambah Data</a>

</li>

@if(session('status'))

<div class="alert alert-success" role="alert">

{{session('status')}}

</div>

@endif

<div class="card-body p-0">

<table class="table table-striped projects">

<thead>

<tr>

<th style="width: 1%"> NO </th>

<th style="width: 30%"> Keterangan Permasalahan </th>

<th style="width: 30%"> Keterangan Solusi </th>

<th style="width: 20%" > Aksi </th>

</tr>

</thead>

@foreach($permasalahan as $p)

<tbody>

<tr>

<td> {{$loop->iteration}} </td>

<td> {{$p->keteranganPermasalahan}} </td>

<td> {{$p->solusi}} </td>

<td class="project-actions text-right">

<a class="btn btn-warning btn-sm" href=" {{route('permasalahan.edit', $p->id)}} ">

Edit

</a>

<form name="{{ $p->id }}" action="{{route('permasalahan.destroy',$p->id)}}" method="post" style="display: inline-block;">

@csrf

@method('DELETE')

<input type="submit" name="" value="Delete" class="btn btn-danger btn-sm">

</form>

</td>

</tr>

@endforeach

</tbody>

</table>

</div>

<div class="pagination justify-content-center " style="margin:20px 0">

{{$permasalahan->links()}}

</div>

</div>

</section>

@endsection

1. Bimbingan

* Create.blade.php

@extends('layouts.master')

@section('content')

<div class="container">

<div class="card mt-5">

<div class="card-header text-left">

<strong>Registrasi Data Diri Bimbingan konseling</strong>

</div>

@if($errors->all())

@include('layouts.error')

@endif

<div class="card-body">

<form method="POST" action="{{route('bimbingan.store')}}">

@csrf

<div class="form-group">

<label for="nim"><strong> NIM </strong></label>

<input type="text" name="nim" value="{{old('nim')}}" placeholder="Masukkan NIM anda " class="form-control" id="nim"> </div>

<div class="form-group">

<label for="namaLengkap"><strong> Nama Lengkap </strong></label>

<input type="text" name="namaLengkap" value="{{old('namaLengkap')}}" placeholder="Masukkan Nama Lengkap anda " class="form-control" id="namaLengkap" >

</div>

<div class="form-group">

<label for="exampleFormControlSelect1"><strong> Jenis Kelamin </strong></label>

<select type="text" name="jenisKelamin" class="form-control" id="exampleFormControlSelect1">

<option value="Laki-Laki" >Laki-Laki</option>

<option value="Perempuan" >Perempuan</option>

</select>

</div>

<div class="form-group">

<label for="exampleFormControlSelect1"><strong> Status Tempat Tinggal </strong></label>

<select type="text" name="status" class="form-control" id="exampleFormControlSelect1">

<option value="Outside" >Outside</option>

<option value="Inside" >Inside</option>

</select>

</div>

<div class="form-group">

<input type="submit" class="btn btn-success" value="Simpan">

</div>

</form>

</div>

</div>

</div>

@stop

* Detail.blade.php

@extends('layouts.master')

@section('content')

<section class="content">

<div class="container-fluid">

<div class="row">

<div class="col">

<!-- Profile Image -->

<div class="card card-primary card-outline">

<div class="card-body box-profile">

<div class="text-center"><p><strong>Detail Data Bimbingan</strong></p>

</div>

<ul class="list-group list-group-unbordered mb-3">

<li class="list-group-item">

<b> NIM </b> <a class="float-right">{{ $bimbingan->nim }}</a>

</li>

<li class="list-group-item">

<b> Nama Lengkap </b> <a class="float-right">{{ $bimbingan->namaLengkap }}</a>

</li>

<li class="list-group-item">

<b> Status </b> <a class="float-right">{{ $bimbingan->status }}</a>

</li>

<li class="list-group-item">

<b> jenisKelamin </b> <a class="float-right">{{ $bimbingan->jenisKelamin }}</a>

</li>

<li class="list-group-item">

<b>Created at :</b> <a class="float-right">{{$bimbingan->created\_at->format(' H:i d-m-y ')}}</a>

</li>

<li class="list-group-item">

<b>Update at :</b> <a class="float-right">{{ $bimbingan->updated\_at->format(' H:i d-m-y ') }}</a>

</li>

</ul>

<fieldset class="border p-4">

<legend class="w-auto">Gejala</legend>

<ul>

@foreach ($bimbingan->gejala as $item)

<li>{{ $item->keteranganGejala }}</li>

@endforeach

</ul>

</fieldset>

<hr>

<fieldset class="border p-4">

<legend class="w-auto">Permasalahan</legend>

@if (count($permasalahan) > 0)

<table class="table table-striped projects">

<thead>

<tr>

<th style="width: 1%"> No </th>

<th style="width: 15%"> Keterangan Permasalahan </th>

<th style="width: 20%"> Keterangan Gejala </th>

<th style="width: 20%"> Keterangan Solusi </th>

</tr>

</thead>

<tbody>

@foreach ($permasalahan as $p)

<tr>

<td>{{ $loop->iteration }}.</td>

<td>{{ $p->keteranganPermasalahan }}</td>

<td>

<ul style="padding-left: 16px;">

@forelse($p->gejala as $g)

<li>{{ $g->keteranganGejala }}</li>

@empty

Maaf, belum ada data gejala untuk Permasalahan ini.

@endforelse

</ul>

</td>

<td>{{ $p->solusi }}</td>

</tr>

@endforeach

</tbody>

</table>

@else

<div class="alert alert-danger">

Gejala Belum sesuai dengan Permasalahan

</div>

@endif

</fieldset>

<hr>

<a href="{{route('bimbingan.index')}}" class="btn btn-primary btn-block"><b>Kembali</b></a>

</div>

<!-- /.card-body -->

</div>

<!-- /.card -->

</div>

<!-- /.col -->

</div>

<!-- /.row -->

</div><!-- /.container-fluid -->

</section>

@endsection

* Edit.blade.php

@extends('layouts.master')

@section('content')

<div class="container">

<div class="card mt-5">

<div class="card-header text-left">

<strong>Tambah Data Bimbingan</strong>

</div>

<div class="card-body">

<form method="POST" action="{{route('bimbingan.update',[$bimbingan->id])}}">

@csrf

@method('PUT')

<div class="form-group">

<label for="nim"><strong> NIM </strong></label>

<input type="text" name="nim" value="{{ $bimbingan->nim}}" placeholder="Masukkan NIM anda " class="form-control @error('nim') is-invalid @enderror" id="nim" >

@error('nim')

<div class="invalid-feedback">

{{$message}}

</div>

@enderror

</div>

<div class="form-group">

<label for="namaLengkap"><strong> Nama Lengkap </strong></label>

<input type="text" name="namaLengkap" value="{{ $bimbingan->namaLengkap}}" placeholder="Masukkan Nama Lengkap anda " class="form-control @error('namaLengkap') is-invalid @enderror" id="namaLengkap" >

@error('namaLengkap')

<div class="invalid-feedback">

{{$message}}

</div>

@enderror

</div>

<div class="form-group">

<label for="exampleFormControlSelect1"><strong> Jenis Kelamin </strong></label>

<select type="text" name="jenisKelamin" class="form-control" id="exampleFormControlSelect1">

<option value="Laki-Laki" >Laki-Laki</option>

<option value="Perempuan" >Perempuan</option>

</select>

</div>

<div class="form-group">

<label for="exampleFormControlSelect1"><strong> Status Tempat Tinggal </strong></label>

<select type="text" name="status" class="form-control" id="exampleFormControlSelect1">

<option value="Outside" @if($bimbingan->status =="Outside") selected @endif>Outside</option>

<option value="Inside" @if($bimbingan->status =="Inside") selected @endif >Inside</option>

</select>

</div>

<div class="form-group">

<input type="submit" class="btn btn-success" value="Simpan">

</div>

</form>

</div>

</div>

</div>

@stop

* Index.blade.php

@extends('layouts.master')

@section('content')

<section class="content">

<div class="card">

<div class="card-header">

<h3 class="card-title">Data Bimbingan</h3>

</div>

<li class="nav-item d-none d-sm-inline-block">

<a href="{{route('bimbingan.create')}}" class="nav-link">Tambah Data</a>

</li>

@section('notifGejala')

@if(session('status'))

<div class="alert alert-success" role="alert">

{{session('status')}}

</div>

@endif

@stop

<div class="card-body p-0">

<table class="table table-striped projects">

<thead>

<tr>

<th style="width: 1%"> NO </th>

<th style="width: 10%"> Nama Lengkap </th>

<th style="width: 5%"> NIM </th>

<th style="width: 5%"> Jenis Kelamin </th>

<th style="width: 7%"> Aksi </th>

</tr>

</thead>

<tbody>

@foreach($bimbingan as $b)

<tr>

<td> {{ $loop->iteration }} </td>

<td> {{ $b->namaLengkap }} </td>

<td> {{ $b->nim }}</td>

<td> {{ $b->jenisKelamin }}</td>

<td class="project-actions text-right">

<a class="btn btn-info btn-sm" href="{{route('bimbingan.show', $b->id)}}"> View </a>

<a class="btn btn-warning btn-sm" href="{{route('bimbingan.edit', $b->id)}}">

Edit

</a>

<a class="btn btn-danger btn-sm" href="#">

Delete

</a>

</td>

</tr>

@endforeach

</tbody>

</table>

</div>

<div class="pagination justify-content-center" style="margin:20px 0">

{{$bimbingan->links()}}

</div>

</div>

</section>

@endsection

1. Konseling

* Create.blade.php

@extends('layouts.master')

@section('content')

<div class="container">

<div class="card mt-5">

@if(session('status'))

<div class="alert alert-success" role="alert">

{{session('status')}}

</div>

@endif

<div class="card-header text-left">

<strong>Pilih Gejala-Permasalahan </strong>

</div>

@if($errors->all())

@include('layouts.error')

@endif

<div class="card-body">

<form method="POST" action="{{ route('konselling.store') }}">

@csrf

<input type="hidden" name="bimbingan\_id" value="{{ $bimbingan\_id }}">

<div class="form-group">

<label>Gejala-gejala yang nampak pada Sosial anda :</label>

<div class="col-md-12">

@foreach ($gejala as $gejala)

<div class="checkbox">

<label><input class="flat" type="checkbox" name="gejala[]" value="{{ $gejala->id }}"> {{ $gejala->keteranganGejala }} </label>

</div>

@endforeach

</div>

</div>

<div class="form-group">

<button type="submit" class="btn btn-primary pull-right">Cek Hasil Konselling <i class="fa fa-fw fa-search"></i></button>

</div>

</form>

</div>

</div>

</div>

@stop

* Index.blade.php

@extends('layouts.master')

@section('content')

<section class="content">

<div class="card">

<div class="card-header">

<h3 class="card-title">Data Konselling</h3>

</div>

<li class="nav-item d-none d-sm-inline-block">

<a href="{{route('konselling.create')}}" class="nav-link">Tambah Data</a>

</li>

<div class="card-body p-0">

<table class="table table-striped projects">

<thead>

<tr>

<th style="width: 1%"> NO </th>

<th style="width: 5%"> NIM </th>

<th style="width: 10%"> Nama Lengkap </th>

<th style="width: 20%"> Permasalahan di alami </th>

<th style="width: 40%"> Gejala di alami </th>

<th style="width: 20%"> Aksi </th>

</tr>

</thead>

{{-- @foreach($konselling as $k) --}}

<tbody>

<tr>

<td> {{-- {{$loop->iteration}} --}} 1</td>

<td> {{-- {{$k->gejala\_id}} --}} 168023</td>

<td> {{-- {{$k->permasalahan\_id}} --}} Rendi</td>

<td> {{-- {{$k->permasalahan\_id}} --}} Karir dan pekerjaan</td>

<td> {{-- {{$k->permasalahan\_id}} --}} Bingung ,ragu dan sukar dalam memahami dan menyelesaikan pelajaran, Belum memahami secara keseluruhan bidang kerja yang di minati</td>

<td class="project-actions text-right">

<a class="btn btn-primary btn-sm" href="{{-- {{route('konselling.show', $k->id)}} --}}">

View

</a>

<a class="btn btn-info btn-sm" href="{{-- {{route('konselling.edit', $k->id)}} --}}">

Edit

</a>

<a class="btn btn-danger btn-sm" href="#">

Delete

</a>

</td>

</tr>

{{-- @endforeach --}}

</table>

</div>

<div class="pagination justify-content-center" style="margin:20px 0">

{{$konselling->links()}}

</div>

</div>

</section>

@endsection

1. Layouts

* App.blade.php

<!doctype html>

<html lang="{{ str\_replace('\_', '-', app()->getLocale()) }}">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- Font Awesome -->

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

<!-- CSRF Token -->

<meta name="csrf-token" content="{{ csrf\_token() }}">

<title>{{ config('app.name', 'Sistem Pakar ') }}</title>

<!-- Scripts -->

<script src="{{ asset('js/app.js') }}" defer></script>

<!-- Fonts -->

<link rel="dns-prefetch" href="//fonts.gstatic.com">

<link href="https://fonts.googleapis.com/css?family=Nunito" rel="stylesheet">

<!-- Styles -->

<link href="{{ asset('css/app.css') }}" rel="stylesheet">

</head>

<body>

<div id="app">

{{-- <nav class="navbar navbar-expand-md navbar-light bg-white shadow-sm">

<div class="container">

<a class="navbar-brand" href="{{ url('/') }}">

{{ config('app.name', 'Laravel') }}

</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="{{ \_\_('Toggle navigation') }}">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<!-- Left Side Of Navbar -->

<ul class="navbar-nav mr-auto">

</ul>

<!-- Right Side Of Navbar -->

<ul class="navbar-nav ml-auto">

<!-- Authentication Links -->

@guest

<li class="nav-item">

<a class="nav-link" href="{{ route('login') }}">{{ \_\_('Login') }}</a>

</li>

@if (Route::has('register'))

<li class="nav-item">

<a class="nav-link" href="{{ route('register') }}">{{ \_\_('Register') }}</a>

</li>

@endif

@else

<li class="nav-item dropdown">

<a id="navbarDropdown" class="nav-link dropdown-toggle" href="#" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false" v-pre>

{{ Auth::user()->name }} <span class="caret"></span>

</a>

<div class="dropdown-menu dropdown-menu-right" aria-labelledby="navbarDropdown">

<a class="dropdown-item" href="{{ route('logout') }}"

onclick="event.preventDefault();

document.getElementById('logout-form').submit();">

{{ \_\_('Logout') }}

</a>

<form id="logout-form" action="{{ route('logout') }}" method="POST" style="display: none;">

@csrf

</form>

</div>

</li>

@endguest

</ul>

</div>

</div>

</nav> --}}

<main class="py-4">

@yield('content')

</main>

</div>

</body>

</html>

* Error.blade.php

<div class="form-group">

<ul class="alert alert-danger">

@foreach($errors->all() as $error)

<li>{{ $error }}</li>

@endforeach

</ul>

</div>

* Master.blade.php

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<!-- CSRF Token -->

<meta name="csrf-token" content="{{ csrf\_token() }}">

<title>{{ config('app.name', 'Laravel') }}</title>

<style>

.bd-placeholder-img {

font-size: 1.125rem;

text-anchor: middle;

-webkit-user-select: none;

-moz-user-select: none;

-ms-user-select: none;

user-select: none;

}

@media (min-width: 768px) {

.bd-placeholder-img-lg {

font-size: 3.5rem;

}

}

</style>

<link href="dashboard.css" rel="stylesheet">

<!-- Select2 -->

<!-- Font Awesome -->

<link rel="stylesheet" href="{{asset('css/fontawesome.css')}}">

<script src="https://kit.fontawesome.com/a076d05399.js"></script>

<!-- icheck bootstrap -->

<link rel="stylesheet" href="{{asset('css/icheck-bootstrap.min.css')}}">

<!-- Theme style -->

<link rel="stylesheet" href="../../dist/css/adminlte.min.css">

<!-- Google Font: Source Sans Pro -->

<link href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700" rel="stylesheet">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/bootstrap-multiselect/0.9.15/js/bootstrap-multiselect.min.js">

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" integrity="sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk" crossorigin="anonymous">

<!-- Ionicons -->

<link rel="icon" href="/docs/4.5/assets/img/favicons/favicon.ico">

<link rel="stylesheet" href="https://code.ionicframework.com/ionicons/2.0.1/css/ionicons.min.css">

<link href="https://getbootstrap.com/docs/4.5/examples/dashboard/dashboard.css" rel="stylesheet">

</head>

<body>

<div id="app">

<nav class="navbar navbar-dark sticky-top bg-dark flex-md-nowrap p-0 shadow">

<a class="navbar-brand col-md-3 col-lg-2 mr-0 px-3" href="javascript:;">Sistem Pakar</a>

<button class="navbar-toggler position-absolute d-md-none collapsed" type="button" data-toggle="collapse" data-target="#sidebarMenu" aria-controls="sidebarMenu" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<ul class="nav">

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" data-toggle="dropdown" href="#" role="button" aria-haspopup="true" aria-expanded="false">

Sudung

</a>

<div class="dropdown-menu dropdown-menu-right">

{{-- <a class="dropdown-item" href="#">Action</a> --}}

{{-- <div class="dropdown-divider"></div> --}}

<a class="dropdown-item" href="{{ route('logout') }}" onclick="event.preventDefault(); document.getElementById('logout-form').submit();">

Logout

</a>

<form id="logout-form" action="{{ route('logout') }}" method="POST" style="display: none;">

@csrf

</form>

</div>

</li>

</ul>

</nav>

<div class="container-fluid">

<div class="row">

<nav id="sidebarMenu" class="col-md-3 col-lg-2 d-md-block bg-light sidebar collapse">

<div class="sidebar-sticky pt-3">

<ul class="nav flex-column">

<li class="nav-item">

<a class="nav-link " href="{{route('home')}}">

<i class="fas fa-tachometer-alt"></i> Dashboard

</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{{route('permasalahan.index')}}">

<i class="fas fa-cogs"></i> Permasalahan

</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{{ route('gejala.index') }}">

<i class="ion ion-pie-graph"></i> Gejala

</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{{route('relasi.index')}}">

<i class="fas fa-sliders-h"></i> Basis Pengetahuan

</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{{route('bimbingan.index')}}">

<i class="fas fa-users-cog"></i> Bimbingan

</a>

</li>

{{-- <li class="nav-item">

<a class="nav-link" href="{{route('konselling.index')}}">

<i class="fas fa-users"></i> Konselling

</a>

</li> --}}

</ul>

</div>

</nav>

<main role="main" class="col-md-9 ml-sm-auto col-lg-10 px-md-4">

<canvas class="my-2 w-100" id="myChart" width="0" height="0"></canvas>

<!-- Page Gejala -->

{{-- @yield('gejalaIndex')

@yield('gejalaCreate')

@yield('gejalaEdit')

@yield('gejalaDetail')

<!-- Page Permasalahan -->

@yield('permasalahancontent')

@yield('permasalahanCreate')

@yield('permasalahanEdit')

@yield('permasalahahDetail')

<!-- Page Relasi -->

@yield('relasiIndex')

@yield('relasiCreate')

@yield('relasiEdit')

@yield('relasiDetail')

Page Solusi

@yield('solusiIndex')

@yield('solusiCreate')

@yield('solusiEdit')

<!-- Page Bimbingan -->

@yield('bimbinganIndex')

@yield('bimbinganCreate')

@yield('bimbinganEdit')

@yield('bimbinganDetail')

<!-- Page Konselling -->

@yield('konsellingIndex')

@yield('konsellingCreate')

@yield('konsellingEdit')

<!-- Dashboard admin -->

@yield('dashboard') --}}

@yield('content')

</main>

</div>

</div>

</div>

<!-- Bootstrap -->

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js" integrity="sha384-OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/kR0JKI" crossorigin="anonymous"></script>

</body>

</html>

**BIODATA**

**DATA PRIBADI**

Nama Lengkap : Hasudungan Sitorus

Jenis Kelamin : Laki-Laki

Tempat Tanggal Lahir : Tanjung-balai 18 maret 1993

Agama : Kristen

Alamat : Binjai, Sumatra-utara

**PENDIDIKAN**

2000-2006 : SD ADVENT BINJAI

2006-2009 : SMP ADVENT BINJAI

2009-2011 : SMN PUTRA ANDA BINJAI

**PENGALAMAN KERJA**

2016-2020 : Ground Departement Universitas Advent Indonesia