**PROTOTYPING SISTEM ADMINISTRASI**

**BERBASIS WEBDAN USABILITY TEST**

**PADA DIVISI DIKLAT RUMAH SAKIT**

**HARAPAN KITA**

**JAKARTA**

**TUGAS AKHIR**



**PRESTENIKO SEPTI RAHADIAN**

**1102001022**

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS TEKNIK DAN ILMU KOMPUTER**

**UNIVERSITAS BAKRIE**

**JAKARTA**

**2016**

**PROTOTYPING SISTEM ADMINISTRASI**

**BERBASIS WEBDAN USABILITY TEST**

**PADA DIVISI DIKLAT RUMAH SAKIT**

**HARAPAN KITA**

**JAKARTA**

**TUGAS AKHIR**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar**

**Sarjana Strata Satu (S1)**

**Program Studi Informatika**

**Universitas Bakrie**

****

**PRESTENIKO SEPTI RAHADIAN**

**1102001022**

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS TEKNIK DAN ILMU KOMPUTER**

**UNIVERSITAS BAKRIE**

**JAKARTA**

**2016**

****HALAMAN PERNYATAAN ORISINALITAS****

Tugas Akhir ini adalah hasil karya saya sendiri,

dan semua sumber baik yang dikutip maupun yang dirujuk

telah saya nyatakan dengan benar

Nama : Presteniko Septi Rahadian

NIM : 1102001022

Tanda Tangan

Tanggal : XX Februari 2016

HALAMAN PENGESAHAN

Tugas Akhir ini diajukan oleh:

|  |  |  |
| --- | --- | --- |
| Nama | : | Presteniko Septi Rahadian |
| NIM | : | 1102001022 |
| Program Studi | : | Informatika |
| Fakultas | : | Teknik dan Ilmu Komputer |
| Judul | : | *Prototyping* Sistem Administrasi Berbasis WebDan *Usability Test* Pada Divisi Diklat Rumah Sakit Harapan Kita Jakarta |

Telah disetujui oleh pembimbing untuk diajukan ke sidang tugas akhir.

**DEWAN PEMBIMBING**

Pembimbing : Yusuf Lestanto, S.T., M.Sc. ( )

Ditetapkan di : Jakarta

Tanggal : Februari 2016

UCAPAN TERIMA KASIH

Puji syukur kehadirat Allah SWT karena atas karunia dan Ridho Nya Penulis dapat menyelesaikan Tugas Akhir ini dengan baik. Tugas Akhir dengan judul “Prototyping Sistem Administrasi Berbasis WebDan Usability Test Pada Divisi Diklat Rumah Sakit Harapan Kita Jakarta” ditulis untuk memenuhi salah satu syarat dalam menyelesaikan perkuliahan pendidikan strata satu (S1) pada Program Studi Informatika, Universitas Bakrie.

Penulis menyadari sepenuhnya bahwa begitu banyak pihak yang telah turut membantu dalam penyelesaian skripsi ini. Melalui kesempatan ini, dengan segala kerendahan hati, penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Kedua orang tua, Ibu Estudiantina dan (Alm) Bapak Bambang Wijanarko, yang telah membesarkan saya dan memberikan ilmu hidup yang luar biasa.
2. (Alm) Bapak Nelis Sulimer, walaupun singkat, telah memberikan saya kebahagiaan dan merubah saya menjadi pribadi yang lebih baik. *Rest In Peace.*
3. Keluarga besar Benny Hasyim; Yangti, Bude Anna, Pakde Kus, Om Iyung, Tante Esi, Tante Oce, Mbak Uno, Mbak Dona, serta Om Tante dan Sepupu lain yang selalu memberikan dorongan dan semangat sehingga penulis mampu menyelesaikan skripsi hingga akhir.
4. Bapak Yusuf Lestanto, S.T., M.Sc. selaku pembimbing terbaik yang senantiasa selalu memberikan inspirasi, ilmu, bimbingan ,dan atas waktu yang telah diluangkan selama ini. Terima kasih sudah memberikan begitu banyak ilmu dalam proses penyelesaian skripsi ini. Sebuah penghormatan untuk memiliki pembimbing seperti ini.
5. Bapak Irwan Prasetya Gunawan, S.T., M.Eng., Ph.D. selaku kaprodi yang selau memberi saran dan masukan sehingga saya menjadi seorang luar biasa dari yang saya harapkan.
6. DR.dr. Ismoyo Sunu Sp.JP (K), selaku Kepala Divisi Diklat Rumah Sakit Harapan Kita Jakarta yang telah mengijinkan penulis untuk melakukan penelitian di tempat tersebut dan memberikan informasi serta ilmu yang banyak sekali.
7. Segenap staff Divisi Diklat Rumah Sakit Harapan Kita Jakarta yang telah memberikan informasi atas penelitain yang penulis lakukan. Berkat informasi yang diberikan, penulis dapat menyelesaikan penelitian dengan baik.
8. Sahabat SMA; Dyan Savithri, Putra Nogroho, Wisnu Prasetyo, Ganjar Tegas Bayuaji, Yohannes Christian, Ayu Nurul Huda, Anneke Meganovia, Himawan Yusuf Wibisono, Dehan Soefriyadi dan Andreas Febrian atas dorongan dan semangat yang selalu diberikan kepada penulis. Penulis juga mengucapkan terima kasih atas ikatan dan persahabatan erat yang sudah terjalin lama.
9. Teman lama; Aisyah Sabrina, Bimo Wahyu Prakoso, Nabila Viera Yovita dan teman lainnya yang tidak bisa saya sebutkan, terima kasih atas persahabatan yang tak pernah berakhir dan dukungan walaupum jarak memisahkan kita.
10. Sahabat Caver; Muhammad Bagus Oka dan Ikrimah Assyarief yang senantiasa menemani dan memberikan canda tawa di *coffee shop* langganan selama penyelesaian skripsi. Ayo kita sama-sama berjuang.
11. Sahabat jurusan; Richa Vennia, Doni Kuncoro, Bagus Dhanist, Yose Firdaus, Kemal Anshari, Nisrina Ulfah, Rendy Mirza dan Taufik Ismail yang telah memberikan saya pengetahuan dan momen yang tak terlupakan.
12. Semua teman tercinta jurusan Informatika 2010 di Universitas Bakrie terima kasih untuk waktu untuk berbagi dan momen indah tak terlupakan yang telah kita lewati.
13. Semua teman angkatan 2010 Universitas Bakrie yang sedang membangun jalan kehidupan. Semoga kebahagiaan selalu menjadi milik kita semua.
14. Terima kasih pada diri ini, yang selalu berhasil bangkit dari rasa depresi dan putus asa sehingga dapat menyelesaikan tanggung jawab dengan kemampuan yang dimiliki. Semoga penulis dapat menjadikan penelitian ini sebagai ilmu untuk menjadi manusia yang berguna dan lebih baik lagi.

Dengan segala keterbatasan yang ada, penulis menyadari bahwa penyusunan Tugas Akhir ini masih jauh dari kesempurnaan. Untuk itu, saran dan kritik akan selalu diterima agar penulis dapat memperbaiki setiap kekurangan untuk kesempurnaan di masa mendatang. Penulis berharap semoga Tugas Akhir ini berguna dan bermanfaat bagi kita semua.

Jakarta, Februari 2016

Presteniko Septi Rahadian

**PROTOTYPING SISTEM ADMINISTRASI BERBASIS WEBDAN USABILITY TEST PADA DIVISI DIKLAT**

**RUMAH SAKIT HARAPAN KITA JAKARTA**

Presteniko Septi Rahadian

ABSTRAK

Penelitian ini membahas proses rancang bangun *prototype* sistem administrasi pada Divisi Diklat rumah sakit Harapan Kita dan melakukan *usability testing* pada sistem tersebut. Dalam penelitian ini, akan dibahas mengenai langkah-langkah yang dilakukan selama proses perancangan sistem dimulai dari *communication, planning, design, prototyping* serta *feedback* yang dilakukan dengan *usability test*. Pada *usability test,* dilakukan analisis hubungan antara aspek *usability* dengan kepuasan calon pengguna. *Prototype* sistem ini dibuat berdasarkan kebutuhan proses administrasi dari Divisi Diklat Rumah Sakit Harapan Kita Jakarta. Sistem berbasis web dipilih karena kebutuhan divisi diklat yang menginginkan agar akses data dapat dilakukan di mana saja dan kapan saja.

Kata kunci: Sistem administrasi, aplikasi berbasis *web, prototyping, usability test.*

**PROTOTYPING SISTEM ADMINISTRASI BERBASIS WEBDAN USABILITY TEST PADA DIVISI DIKLAT**

**RUMAH SAKIT HARAPAN KITA JAKARTA**

Presteniko Septi Rahadian

ABSTRACT

*This research discusses about prototyping process of administration system in Divisi Diklat rumah sakit Harapan Kita Jakarta and conducts usability testing based on the system prototype. In this research, the process of prototyping will be described; Start from communication, planning, design, protoyping and feedback. The feedback process will be conducted with usability test. The objective of the usability test conducted in this research is for analyzing correlations between usability aspects and candidate users’ satisfaction. The development of system prototype itself is based on the administrative needs of Divisi Diklat Rumah Sakit Harapan Kita. The usage of web-based system based on Divisi Diklat’s need for accessing the data from anywhere and anytime.*

*Keywords: System, administration, web-based system, prototyping, usability test.*

DAFTAR ISI

[HALAMAN PERNYATAAN ORISINALITAS ii](#_Toc443472456)

[HALAMAN PENGESAHAN iii](#_Toc443472457)

[UCAPAN TERIMA KASIH iv](#_Toc443472458)

[ABSTRAK vii](#_Toc443472459)

[ABSTRACT viii](#_Toc443472460)

[DAFTAR ISI ix](#_Toc443472461)

[DAFTAR GAMBAR xi](#_Toc443472462)

[DAFTAR TABEL xiv](#_Toc443472463)

[BAB I PENDAHULUAN 1](#_Toc443472464)

[1.1 Latar Belakang Masalah 1](#_Toc443472465)

[1.2 Perumusan Masalah 3](#_Toc443472466)

[1.3 Pembatasan Masalah 3](#_Toc443472467)

[1.4 Tujuan Penelitian 3](#_Toc443472468)

[1.5 Manfaat Penelitian 4](#_Toc443472469)

[BAB II TINJAUAN PUSTAKA 5](#_Toc443472470)

[2.1 Administrasi 5](#_Toc443472471)

[2.2 Keadaan Sebelum Penelitian 6](#_Toc443472472)

[2.3 Penelitian Terdahulu 7](#_Toc443472473)

[2.4 *Prototyping* 9](#_Toc443472474)

[2.5 *Framework* dan Yii *Framework* 11](#_Toc443472475)

[2.6 *Usability test* 13](#_Toc443472476)

[2.7 Uji Normalitas 15](#_Toc443472477)

[2.8 Uji Korelasi 16](#_Toc443472478)

[2.9 Tools 17](#_Toc443472479)

[BAB III METODE PENELITIAN 19](#_Toc443472480)

[3.1 *Communication* 19](#_Toc443472481)

[3.2 *Planning and Design* 20](#_Toc443472482)

[3.2.1 *Login* 38](#_Toc443472483)

[3.2.2 *Register* 39](#_Toc443472484)

[3.2.3 *Create* 40](#_Toc443472485)

[3.2.4 *Read* 41](#_Toc443472486)

[3.2.5 *Update* 42](#_Toc443472487)

[3.2.6 *Delete* 43](#_Toc443472488)

[3.2.7 Pendaftaran Pelatihan 44](#_Toc443472489)

[3.3 *Prototyping* 45](#_Toc443472490)

[3.4 *Feedback* 47](#_Toc443472491)

[BAB IV IMPLEMENTASI 51](#_Toc443472492)

[4.1 Desain 51](#_Toc443472493)

[4.1.1 Desain Database 51](#_Toc443472494)

[4.1.2 *Deployment diagram* 61](#_Toc443472495)

[4.1.3 Sequence Diagram 62](#_Toc443472496)

[4.1.4 *Class Diagram* 78](#_Toc443472497)

[4.2 Fitur Aplikasi 80](#_Toc443472498)

[4.2.1 Fitur general 80](#_Toc443472499)

[4.2.2 Fitur User (administrator) 86](#_Toc443472500)

[4.2.3 Fitur Peserta 99](#_Toc443472501)

[4.2.4 Fitur Pengajar 107](#_Toc443472502)

[4.3 Usability Test 113](#_Toc443472503)

[4.3.1 Langkah usability test 113](#_Toc443472504)

[4.3.2 Hasil Survey Pengujian Aplikasi 118](#_Toc443472505)

[4.3.3 Uji normalitas 126](#_Toc443472506)

[4.3.4 Uji Korelasi Hipotesis 128](#_Toc443472507)

[4.4 Pembahasan usability test 135](#_Toc443472508)

[BAB V KESIMPULAN DAN SARAN 136](#_Toc443472509)

[5.1 Kesimpulan 136](#_Toc443472510)

[5.2 Saran 137](#_Toc443472511)

[DAFTAR PUSTAKA 139](#_Toc443472512)

[LAMPIRAN 143](#_Toc443472513)

DAFTAR GAMBAR

[Gambar 2.1 Tahap proses prototyping 10](#_Toc443408811)

[Gambar 2.2 Arsitektur MVC pada Yii 13](#_Toc443408812)

[Gambar 2.3 Model penelitian Green dan Pearson 15](#_Toc443408813)

[Gambar 2.4 Rumus Shapiro-Wilk 16](#_Toc443408814)

[Gambar 2.5 Rumus uji korelasi Spearman 17](#_Toc443408815)

[Gambar 3.1 Use Case Diagram analisa fitur sistem 24](#_Toc443408816)

[Gambar 3.2 Activity diagram login 38](#_Toc443408817)

[Gambar 3.3 Activity diagram Register 39](#_Toc443408818)

[Gambar 3.4 Activity diagram create 40](#_Toc443408819)

[Gambar 3.5 Activity diagram read 41](#_Toc443408820)

[Gambar 3.6 Activity diagram update 42](#_Toc443408821)

[Gambar 3.7 Activity diagram delete 43](#_Toc443408822)

[Gambar 3.8 Activity diagram pendaftaran pelatihan 44](#_Toc443408823)

[Gambar 3.9 Desain awal *mock-up* sistem 45](#_Toc443408824)

[Gambar 3.10 Desain awal (*mock-up*) *dashboard* pengguna 46](#_Toc443408825)

[Gambar 3.11 Kerangka berpikir konstruk 47](#_Toc443408826)

[Gambar 4.1 *Conceptual database design* dari *prototype* sistem 53](#_Toc443408827)

[Gambar 4.2 *Logical database design* dari prototype sistem 59](#_Toc443408828)

[Gambar 4.3 Diagram *entity relationship database prototype sistem* 60](#_Toc443408829)

[Gambar 4.4 *Deployment diagram* sistem administrasi 61](#_Toc443408830)

[Gambar 4.5 *Sequence diagram login* pengguna 62](#_Toc443408831)

[Gambar 4.6 *Sequence diagram logout* pengguna 63](#_Toc443408832)

[Gambar 4.7 *Sequence diagram* pengaturan 64](#_Toc443408833)

[Gambar 4.8 *Sequence diagram create* akun administrator 65](#_Toc443408834)

[Gambar 4.9 *Sequence diagram read* akun administrator 66](#_Toc443408835)

[Gambar 4.10 *Sequence diagram update* akun administrator 68](#_Toc443408836)

[Gambar 4.11 *Sequence* diagram *delete* akun administrator 69](#_Toc443408837)

[Gambar 4.12 *Sequence diagram* lihat pelatihan yang diikuti 70](#_Toc443408838)

[Gambar 4.13 *Sequence diagram* pendaftaran pelatihan 71](#_Toc443408839)

[Gambar 4.14 *Sequence diagram* lihat absen 72](#_Toc443408840)

[Gambar 4.15 *Sequence diagram* lihat status pembayatan 73](#_Toc443408841)

[Gambar 4.16 *Sequence diagram* konfirmasi pembayaran 74](#_Toc443408842)

[Gambar 4.17 *Class diagram prototype* sistem 79](#_Toc443408843)

[Gambar 4.18 Tampilan laman utama (*home*) dengan bagian *content* 81](#_Toc443408844)

[Gambar 4.19 laman utama dengan kalender pelatihan 82](#_Toc443408845)

[Gambar 4.20 Bagian footer *prototype* sistem 82](#_Toc443408846)

[Gambar 4.21 laman pelatihan pada *prototype* sistem 83](#_Toc443408847)

[Gambar 4.22 Laman jadwal pelatihan *tab* pendaftaran 84](#_Toc443408848)

[Gambar 4.23 Laman jadwal pelatihan *tab* berjalan 84](#_Toc443408849)

[Gambar 4.24 Laman *login* peserta 85](#_Toc443408850)

[Gambar 4.25 *form login* pengajar dan administrator 85](#_Toc443408851)

[Gambar 4.26 *form* registrasi peserta 86](#_Toc443408852)

[Gambar 4.27 Laman *dashboard* administrator 87](#_Toc443408853)

[Gambar 4.28 *link home* administrator 87](#_Toc443408854)

[Gambar 4.29 *link* admin administrator 88](#_Toc443408855)

[Gambar 4.30 Laman pengaturan akun administrator 88](#_Toc443408856)

[Gambar 4.31 *Link* pengaturan akun pengajar 89](#_Toc443408857)

[Gambar 4.32 Laman pengaturan pengajar 89](#_Toc443408858)

[Gambar 4.33 Laman tambah pengajar 90](#_Toc443408859)

[Gambar 4.34 *Link* pengaturan akun peserta 90](#_Toc443408860)

[Gambar 4.35 Laman pengaturan akun peserta 91](#_Toc443408861)

[Gambar 4.36 Laman pembuatan akun peserta baru 91](#_Toc443408862)

[Gambar 4.37 Laman pengaturan entri pelatihan bagi peserta 92](#_Toc443408863)

[Gambar 4.38 Laman pengaturan absen pelatihan peserta 93](#_Toc443408864)

[Gambar 4.39 Laman tambah data absen 93](#_Toc443408865)

[Gambar 4.40 Laman pengaturan nilai peserta 94](#_Toc443408866)

[Gambar 4.41 *Link* pengaturan pelatihan 94](#_Toc443408867)

[Gambar 4.42 Laman pengaturan jenis pelatihan 95](#_Toc443408868)

[Gambar 4.43 Laman pembuatan jenis pelatihan baru 96](#_Toc443408869)

[Gambar 4.44 Laman pengaturan jadwal pelatihan 96](#_Toc443408870)

[Gambar 4.45 Laman pembuatan jadwal pelatihan baru 97](#_Toc443408871)

[Gambar 4.46 *Link* pembayaran pelatihan 98](#_Toc443408872)

[Gambar 4.47 Laman pengaturan pembayaran 98](#_Toc443408873)

[Gambar 4.48 Laman pembuatan pembayaran baru 99](#_Toc443408874)

[Gambar 4.49 *Link home* 100](#_Toc443408875)

[Gambar 4.50 *Link* pengaturan profil peserta 101](#_Toc443408876)

[Gambar 4.51 Laman edit profil peserta 101](#_Toc443408877)

[Gambar 4.52 Laman ganti password peserta 102](#_Toc443408878)

[Gambar 4.53 *Link* pengaturan pelatihan peserta 102](#_Toc443408879)

[Gambar 4.54 Laman list pelatihan yang diikuti 103](#_Toc443408880)

[Gambar 4.55 Laman lihat daftar absen pelatihan peserta 103](#_Toc443408881)

[Gambar 4.56 Laman pendaftaran pelatihan 104](#_Toc443408882)

[Gambar 4.57 Laman sukses pendaftaran 104](#_Toc443408883)

[Gambar 4.58 Notifikasi sudah terdaftar dalam pelatihan 105](#_Toc443408884)

[Gambar 4.59 *Link* pembayaran pelatihan peserta 105](#_Toc443408885)

[Gambar 4.60 Laman *list* pembayaran 106](#_Toc443408886)

[Gambar 4.61 Laman konfirmasi pembayaran 106](#_Toc443408887)

[Gambar 4.62 Laman *dasboard* pengajar 107](#_Toc443408888)

[Gambar 4.63 *Link home* 108](#_Toc443408889)

[Gambar 4.64 *Link* pengaturan akun pengajar 108](#_Toc443408890)

[Gambar 4.65 Laman edit profil pengajar 109](#_Toc443408891)

[Gambar 4.66 Laman ganti password pengajar 109](#_Toc443408892)

[Gambar 4.67 Laman *list* kelas pelatihan 110](#_Toc443408893)

[Gambar 4.68 Laman *list* peserta 110](#_Toc443408894)

[Gambar 4.69 Laman daftar peserta 111](#_Toc443408895)

[Gambar 4.70 Laman pengaturan nilai 111](#_Toc443408896)

[Gambar 4.71 Laman pengaturan nilai 112](#_Toc443408897)

[Gambar 4.72 Laman *list* absen pelatihan 112](#_Toc443408898)

[Gambar 4.73 Laman daftar absen peserta 113](#_Toc443408899)

[Gambar 4.74 *flowchart usability test* 115](#_Toc443408900)

[Gambar 4.75Diagram *scatterplot* relasi monotonik dan non-monotonik 128](#_Toc443408901)

[Gambar 4.76 Diagram *scatterplot* konstruk X dengan konstruk Y. Terlihat adanya relasi monotonik 129](#_Toc443408902)

DAFTAR TABEL

[Tabel 3.1 Analisa fitur yang dibutuhkan 21](#_Toc443408434)

[Tabel 3.2 Use Case Scenario Login 24](#_Toc443408435)

[Tabel 3.3 Use Case Scenario Logout 25](#_Toc443408436)

[Tabel 3.4 Use Case Scenario melihat menu Administrator 26](#_Toc443408437)

[Tabel 3.5 Use Case Scenario melihat menu Peserta 26](#_Toc443408438)

[Tabel 3.6 Use Case Scenario melihat menu Pengajar 27](#_Toc443408439)

[Tabel 3.7 Use Case Scenario kelola data Peserta 27](#_Toc443408440)

[Tabel 3.8 Use Case Scenario kelola data jadwal 28](#_Toc443408441)

[Tabel 3.9 Use Case Scenario kelola data jadwal 29](#_Toc443408442)

[Tabel 3.10 Use Case Scenario kelola data pengajar 29](#_Toc443408443)

[Tabel 3.11 Use Case Scenario kelola data pembayaran 30](#_Toc443408444)

[Tabel 3.12 Use Case Scenario kelola data nilai 31](#_Toc443408445)

[Tabel 3.13 Use Case Scenario kelola data absen 31](#_Toc443408446)

[Tabel 3.14 Use Case Scenario registrasi 32](#_Toc443408447)

[Tabel 3.15 Use Case Scenario melihat jadwal pelatihan 33](#_Toc443408448)

[Tabel 3.16 Use Case Scenario kelola profil 33](#_Toc443408449)

[Tabel 3.17 Use Case Scenario pendaftaran pelatihan 34](#_Toc443408450)

[Tabel 3.18 Use Case Scenario konfirmasi pembayaran 35](#_Toc443408451)

[Tabel 3.19 Use Case Scenario input data absen 35](#_Toc443408452)

[Tabel 3.20 Use Case Scenario melihat data absen 36](#_Toc443408453)

[Tabel 3.21 Use Case Scenario melihat data nilai 37](#_Toc443408454)

[Tabel 3.22 Konstruk *usability test* 48](#_Toc443408455)

[Tabel 3.23 Skala Likert yang dilakukan pada *usability test* 49](#_Toc443408456)

[Tabel 4.1 Daftar *entity* dan keterangan *database* 51](#_Toc443408457)

[Tabel 4.2 Daftar *attribute* masing-masing *entity* 54](#_Toc443408458)

[Tabel 4.3 Tabel model dan action pengaturan 64](#_Toc443408459)

[Tabel 4.4 Tabel model dan action create 66](#_Toc443408460)

[Tabel 4.5 Tabel model dan function read 67](#_Toc443408461)

[Tabel 4.6 Tabel model dan function update 68](#_Toc443408462)

[Tabel 4.8 Pertanyaan kuisioner *usability test* 116](#_Toc443408463)

[Tabel 4.9 karakteristik responden *usability test* 118](#_Toc443408464)

[Tabel 4.10 *mean* kategori/konstruk *ease of use* 120](#_Toc443408465)

[Tabel 4.11 *mean* kategori/konstruk *customization* 121](#_Toc443408466)

[Tabel 4.12 *mean* kategori/konstruk *downlooad delay* 122](#_Toc443408467)

[Tabel 4.13 *mean* kategori/konstruk *content* 123](#_Toc443408468)

[Tabel 4.14 *mean* kategori/konstruk *satisfaction* 124](#_Toc443408469)

[Tabel 4.15 *mean* setiap kategori/konstruk 125](#_Toc443408470)

[Tabel 4.16 uji normalitas atas kategori/konstruk 126](#_Toc443408471)

[Tabel 4.17 Hasil uji korelasi Spearman antara *ease of use dengan satisfaction* 130](#_Toc443408472)

[Tabel 4.18 Hasil uji korelasi Spearman antara *customization* dengan *satisfaction* 132](#_Toc443408473)

[Tabel 4.19 Hasil uji korelasi Spearman antara *download delay* dengan *satisfaction* 133](#_Toc443408474)

[Tabel 4.20 Hasil uji korelasi Spearman antara *content* dengan *satisfaction* 134](#_Toc443408475)

# PENDAHULUAN

## Latar Belakang Masalah

Menurut Peraturan Pemerintah nomor 101 Tahun 2000 [1] dinyatakan bahwa pendidikan dan pelatihan pegawai negeri sipil yang selanjutnya disebut diklat, adalah proses penyelenggaraan belajar mengajar dalam rangka meningkatkan kemampuan Pegawai Negeri Sipil. Tujuan diklat diantaranya adalah meningkatkan pengetahuan, keterampilan dan sikap agar dapat melaksanakan tugas pekerjaan, baik yang bersifat umum pemerintahan maupun pembangunan, yang berorientasi pada pelayanan, pengayoman, dan pengembangan partisipasi masyarakat.

Berdasarkan hasil wawancara (*interview*) dengan Divisi pendidikan dan pelatihan (Diklat) yang terdapat pada rumah sakit Harapan Kita, Divisi Diklat merupakan sebuah divisi yang memberikan informasi serta mengadakan pelatihan yang berguna bagi pesertanya. Tidak hanya memberikan informasi, divisi ini juga mempunyai tugas untuk melakukan administrasi mengenai data-data terkait yang berhubungan dengan pendaftaran calon peserta, pengajar, absensi setiap harinya, dan nilai dari tiap-tiap pelatihan yang diadakan.

Namun, sampai saat ini Divisi Diklat Rumah Sakit Harapan Kita masih menggunakan sistem administrasi konvensional atau d engan cara manual untuk proses pencatatan dan pembuatan laporan mengenai hal-hal yang berhubungan dengan pelatihan. Para pegawai setiap hari harus melakukan pencatatan data-data nilai, absensi peserta dan pengajar, bahkan mengatur jadwal masuk dan pendaftaran serta pembayaran yang diberikan oleh calon peserta pada sebuah kertas atau Microsoft® Excel, satu persatu. Akibat hal ini, terjadi *human error* pada proses input data. Terjadinya *error* dapat menimbulkan kesalahpahaman yang serius dan frustasi [2] bagi pihak yang terkait dengan sistem administrasi konvensional tersebut. Hilangnya data-data penting juga menghasilkan laporan yang tidak tepat bagi pegawai divisi diklat rumah sakit Harapan Kita.

Penelitian pada sistem administrasi sendiri pada dasarnya bukanlah sebuah hal yang baru. Rancang bangun sistem pengolahan administrasi berbasis web pada kemahasiswaan STIKOM Surabaya [3], penerimaan mahasiswa baru universitas Mataram [4], dan sistem informasi pelatihan koperasi Uji Mutu [5] pernah dilakukan untuk menjawab masalah-masalah administrasi yang beruhubungan dengan tempat tersebut. Hal ini menunjukkan bahwa pengembangan sistem berbasis web dewasa ini dibutuhkan bagi tiap organisasi karena permasalahan yang berbeda-beda.

Pada penelitian ini, sistem yang dibutuhkan bagi Divisi Diklat tersebut adalah sistem yang berbasis web. Alasan penggunaan sistem berbasis web ini tidak lepas dari biaya *maintenance* dan *deployment cost* yang murah serta dapat diakses di mana saja dan kapan saja [6]. Hal ini tidak hanya memudahkan pegawai dan pengajar, tetapi juga peserta baik yang belum melakukan pelatihan ataupun yang sudah terdaftar.

Tentunya dalam melakukan rancang bangun diperlukan sebuah kajian mengenai *usability test*. Tujuan diadakannya *usability test* pada penelitian ini adalah untuk menganalisis aspek *usability* yaitu *ease of use, customization, download delay* dan *content* [7] apakah mempengaruhi sikap kepuasan pengguna sistem terhadap aplikasi [8]. Hal ini berhubungan erat dengan penelitian yang dilakukan oleh User Interface Engineering, Inc., yang mengemukakan bahwa 60% waktu terbuang karena orang tidak bisa menemukan informasi dari website dan berdampak pada penurunan produktivitas, meningkatkan frustasi, dan bentuk kerugian lainnya [9].

## Perumusan Masalah

Dalam sistem administrasi yang akan diteliti oleh penulis, ditemukan beberapa masalah berikut:

1. Bagaimana proses membangun sebuah sistem administrasi berbasis web di divisi diklat rumah sakit Harapan Kita?
2. Apa saja fitur yang dibutuhkan untuk membuat sistem administrasi berbasis web di divisi diklat rumah sakit Harapan Kita?
3. Apakah aspek *usability (ease of use, customization, download delay* dan *content)* dari *prototype* sistemyang dikembangkan mempunyai pengaruh terhadap kepuasan calon pengguna?

## Pembatasan Masalah

Penelitian ini hanya akan membahas mengenai proses *prototyping* sistem administrasi Divisi Diklat Harapan Kita Jakarta, dimulai dari pengumpulan data, perencanaan sistem, pembuatan *prototype* sampai *usability testing*. *Usability testing* yang dilakukan hanya akan menganalisis pengaruh aspek *usability* dari *prototype* sistem terhadap kepuasan calon pengguna.

## Tujuan Penelitian

Tujuan mengapa penelitian tersebut dilakukan adalah:

1. Membangun *prototype* sistem admistrasi Divisi Diklat Rumah Sakit Harapan Kita Jakarta berbasis web dengan menggunakan proses *prototyping*.
2. Mengembangkan *prototype* sistem administrasi berdasarkan fitur yang dibutuhkan oleh Divisi Diklat Rumah Sakit Harapan Kita Jakarta.
3. Mendapatkan hasil analisis hubungan antara apek *usability* dari *prototype* sistem terhadap kepuasan calon pengguna.

## Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Bagi pihak divisi diklat rumah sakit Harapan Kita, hasil dari rancang bangun aplikasi ini dapat dipakai sebagai sistem yang mempermudah proses kerja selama ini.
2. Bagi pengembang aplikasi, penelitian ini diharapkan bisa menjadi acuan perancangan aplikasi berbasis web.
3. Bagi Mahasiswa Teknik Informatika dan Sistem Informasi, diharapkan penelitian ini dapat menjadi bahan pelajaran rekayasa perangkat lunak dan pemrograman berbasis web.
4. Bagi Civitas akademika Fakultas Teknik dan Ilmu Komputer, diharapkan penelitian ini dapat menjadi acuan utuk penelitian di masa mendatang.

# TINJAUAN PUSTAKA

## Administrasi

Dalam membangun sistem ini, diperlukan sebuah pengertian dan makna administrasi. Menurut Simon, administrasi dapat didefinisikan sebagai “*activities of groups cooperating to accomplish common goals”* [10]. Kata administrasi berasal dari bahasa Belanda, “*administratie*” yang mempunyai arti seluruh kegiatan yang meliputi tulis menulis, ketik mengetik, komputerisasi, surat menyurat, dan kearsipan. Berkaca dari arti kata adminisitrasi sebelumnya, dapat kita artikan administrasi adalah pelayanan dalam kegiatan usaha kantor [11].

Tidak lepas dari konsep arti sebelumnya, menurut Prof. Prajudi Atmosudirdjo (Lembaga Administrasi Negara), membedakan administrasi dalam 2 pengertian, yaitu administrasi dalam pengertian sempit dan luas [11]. Dalam pengertian sempit ditinjau dari lingkup kerja yang sempit yaitu hanya berkisar kegiatan tata usaha kantor seperti tulis menulis, surat menyurat dan kearsipan. Sedangkan dalam pengertian luas, administrasi dibedakan lagi menjadi tiga sudut, yaitu proses, fungsi dan tugas, dan Kepranataan/institusi.

Apabila dilihat dari sudut proses, administrasi merupakan keseluruhan proses, mulai dari proses pemikiran, perencanaan, pengaturan, penggerakan, pengawasan sampai dengan pencapaian tujuan [11]. Sedangkan sudut fungsi dan tugas memberikan pengertian bahwa administrasi adalah aktivitas yang harus dilakukan oleh seseorang yang berkedudukan “administrator”. Terakhir, apabila ditinjau dari kepranataan (institusi), maka administrasi adalah melihat kegiatan dalam suatu lembaga yang melakukan aktivitas tertentu, misalnya: perbankan, maka ada orang-orang yang melakukan kegiatan perbankan dalam lembaga itu.

Dari uraian di atas, dapat disimpulkan suatu pemahaman bahwa administrasi adalah “Aktivitas-aktivitas untuk mencapai tujuan” atau “proses penyelenggaraan kerja untuk mencapai tujuan”.

## Keadaan Sebelum Penelitian

Selama melakukan proses *interview,* divisi diklat rumah sakit Harapan Kita memberikan informasi mengenai beberapa proses kerja selama ini. Adapun proses kerja awal yang dibahas penulis merupakan beberapa proses yang dibutuhkan oleh divisi diklat rumah sakit Harapan Kita dalam membangun sistem administrasi berbasis *web* ini. Beberapa proses tersebut adalah:

1. Pendaftaran

Dalam proses ini, divisi diklat rumah sakit Harapan Kita Jakarta menetapkan jadwal pelatihan yang diadakan. Setelah melakukan penetapan jadwal, informasi tersebut diberikan ke divisi SDM yang bertugas memberitahukan adanya pelatihan ke pihak luar dalam hal ini calon peserta. Setelah calon peserta terkumpul, divisi SDM mengirimkan daftar calon peserta ke divisi diklat kembali. Divisi diklat mencatat daftar tersebut secara manual, dan ketika hari pertama pelatihan, setiap calon peserta tersebut mengisi biodata yang diberikan. Pengisian biodata dilakukan berulang-ulang setiap diadakan pelatihan.

1. Data Peserta

Pada proses ini, pendataan untuk data peserta dilakukan oleh staff administrasi dengan aplikasi Microsoft® Excel dan hasilnya dicetak di atas kertas. Kertas-kertas tersebut akan diletakkan di ruang arsip dan juga disimpan dalam Salinan *softcopy.* Pendataan calon peserta ini dilakukan untuk setiap pelatihan yang diadakan oleh divisi diklat rumah sakit Harapan Kita, satu folder dan *file* untuk setiap pelatihan. Pendataan yang dilakukan berulang-ulang ini menurut divisi diklat rumah sakit Harapan Kita dapat menimbulkan data ganda. Tidak ada *database* bagi tiap Peserta juga memberikan masalah tersendiri, yaitu kesulitan mencari data yang sudah lama apabila terjadi suatu masalah.

1. Data pengajar

Sama seperti data peserta, pendataan pengajar dilakukan oleh staff administrasi. Pendataan pengajar juga dilakukan untuk setiap pelatihan yang diadakan oleh divisi diklat rumah sakit Harapan Kita. Tidak ada *database* yang dibuat sebagai basis data pengajar dan kemungkinan adanya data ganda.

1. Absensi

Pada proses absensi, divisi diklat akan menyiapkan kertas absen yang akan dibagikan untuk mencatat hadir atau tidaknya peserta pada setiap jadwal pelatihan yang diadakan. Setelah jadwal pelatihan usai, divisi diklat rumah sakit Harapan Kita mengumpulkan kertas tersebut dan melakukan input data absen secara manual pada Microsoft® Excel satu-persatu. Sama seperti data pengajar, data absensi juga tidak mempunyai *database* sebagai basis datanya.

1. Nilai

Dalam menjalankan proses nilai, pengajar akan memberikan nilai setiap peserta pada setiap akhir pelatihan. Dalam hal ini pengajar akan melakukan *input* dengan *form* cetak kertas yang telah diberikan oleh divisi diklat rumah sakit Harapan Kita. Setelah pengajar mengisi semua nilai yang ada, seperti pada proses absensi, administrator juga akan melakukan input data secara manual pada Microsoft® Excel satu-persatu.

## Penelitian Terdahulu

Penelitian sebelumnya yang telah dilakukan dan mempunyai kemiripan serta kerterkaitan dengan penelitian ini adalah:

1. Penelitian yang dilakukan oleh Lemantara [3] tentang Rancung Bangun Sistem Pengolahan Administrasi Berbasis Web Pada Kemahasiswaan STIKOM Surabaya. Pada penelitian ini dilakukan rancang bangun yang menyangkut administrasi pengiriman mahasiswa dalam sebuah *event* menggunakan metode *Analytical Hierarchy Process (AHP).* Rancang bangun juga dilakukan dalam mengolah administrasi kemahasiswaan yang menyangkut perkuliahan 0 SKS, bakat minat, pengalaman organisasi, serta prestasi mahasiswa.
2. Penelitian yang dilakukan oleh Kusdiantoro [12] tentang analisis *usability* dari *website* akademik perguruan tinggi di Indonesia. Penelitian ini menggunakan metode PROMETHEE, VIKOR, dan ELECTREE. Pada penelitian ini analisis *usability* diambil melalui *website webometrics* dan mengaju kepada situs perguruan tinggi negeri di Indonesia.
3. Penelitian yang dilakukan oleh Ismi [4] tentang rancang bangun sistem administrasi penerimaan mahasiswa baru Universitas Mataram. Rancang bangun sistem dalam penelitian ini menggunakan Bahasa pemrograman PHP dan MySQL. Tujuan dari penelitian ini addalah mempermudah proses penerimaan calon mahasiswa baru yang berada diluar daerah maupun didalam, dengan cara memberikan akses melalui situs internet.
4. Penelitian yang dilakukan oleh Napitupulu [5] tentang rancang bangun sistem informasi pelatihan Koperasi Uji Mutu berbasis *web.* Perancangan penelitian ini menghasilkan sebuah *prototype* sistem dari *website* Koperasi Uji Mutu. Penelitian ini menggunakan metode diagram konteks dan *use case* untuk mengidentifikasi entitas, data dan proses-proses yang terjadi dalam sistem.
5. Studi kasus yang dilakukan oleh Battleson, Booth, dan Weintrop [13] tentang pengujian *usability* dari *website* perpustakaan *University at Buffalo.* Studi kasus ini menggunakan pertanyaan-pertanyaan yang berbentuk tugas (*tasks*) sebagai alat untuk melakukan *usability testing*. Hasil dari studi kasus ini menyimpulkan bahwa masih terdapat kelemahan dalam kemudahan penggunan pada *website* perpustakaan tersebut. Battleson, Booth, dan Weintrop juga menyimpulkan bahwa *usability testing* merupakan suatu hal yang tidak terpisahkan bagi pengembangan sebuah *website.*
6. Penelitian yang dilakukan oleh Green dan Pearson [7] yang berjudul *The Examination Of Two Web Site Usability Instruments For Use In B2C E-Commerce Organizations.* Tujuan dari penelitian ini adalah untuk menemukan kelebihan dan kelemahan pada *web site* B2C (*Business-to-Costumer*) pada sebuah organisasi. Metode *usability testing* dilakukan dengan cara memberikan kuisioner dari simulasi pembelian di sebuah *web site.*
7. Penelitian yang dilakukan oleh Prayoga dan Sensuse [8] tentang analisis *usability* pada aplikasi berbasis *web* dengan menggunakan model kepuasan pengguna (*User satisfaction*). Penelitian ini mempunyai tujuan untuk menghasilkan rekomendasi perbaikan atas aplikasi yang diharapkan dapat meningkatkan kemanfaat (*usability*) aplikasi. Metode dilakukan dengan memberikan *form* kuisioner yang berisi skala likert. Setelah *form* berhasil dikumpulkan, dilakukan *measurement model* menggunakan aplikasi LISREL.

## *Prototyping*

*Prototyping* merupakan sebuah metode pendekatan iteratif pada *systems development* dimana *user* dan pengembang bekerja bersama-sama untuk mengembangkan sebagian atau seluruh sistem yang akan dibangun [14]. Roger S. Pressman menjelaskan bahwa di dalam metode *prototyping* terdapat sebuah paradigma *prototyping* [15, 15]yang terdiri dari lima tahap berulang. Kelima tahap tersebut adalah:

1. *Communication*

Merupakan tahapan awal dari metode *prototyping*. Pada tahap ini, pengembang akan bertemu dengan *stakeholder (user)* untuk menentukan definisi dan tujuan dari sebuah *software* dan *requirement* yang dibutuhkan.

1. *Plan dan Modeling Design*

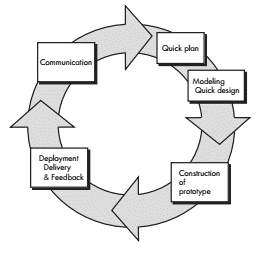
Setelah pengembang dengan *stakeholder (user)* dapat menentukan definisi dan tujuan dari sebuah *software* yang akan dikembangkan, akan berlangsung dua tahapan yang berlangsung pada waktu yang sama atau saling berdekatan. Dalam tahap *plan,* pengembang akan menentukan rencana *prototyping iteration* dan *modeling,* dalam bentuk *design.* Fokus tahapan *modeling design* adalah untuk melakukan representasi dari aspek *software* yang dapat dilihat oleh *stakeholder (user)*. Contoh aspek *software* yang dimaksud adalah *human interface layout* atau *output display formats*.

1. *Construction of Prototype*

Setelah tahapan *modeling design* terjadi tahap selanjutnya, yaitu pembuatan *prototype.* *Prototype* itu sendiri haruslah bertugas sebagai *first system* dan sebagai mekanisme untuk melakukan identifikasi *requirement* dari *stakeholder (user)*.

1. *Deployment Delivery & Feedback*

Setelah pembuatan *prototype* awal selesai, tahap selanjutnya adalah *stakeholder (user)* akan melakukan uji coba. *Stakeholder (user)* akan memberikan *feedback* kepada pengembang untuk melakukan perbaikan berdasarkan *requirement.* Pengembang selanjutnya akan melakukan *tuning* terhadap *software* tersebut sampai *stakeholder (user)* mendapatkan apa yang dibutuhkan.



Gambar 2.II.1 Tahap proses prototyping

*(sumber: Pressman, 2010)*

Penggunaan metode dan model *prototyping* dipilih karena pertimbangan sebagai berikut:

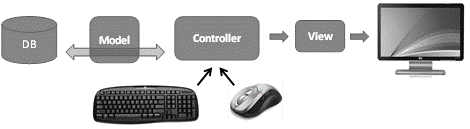
* Pihak Divisi Diklat Rumah Sakit Harapan Kita Jakarta hanya memberikan *general objectives* pada proses *interview* awal yang dilakukan*.*
* Adanya keinginan dari Divisi Diklat Rumah Sakit Harapan Kita Jakarta untuk mengembangkan dan mengimplementasikan sistem lebih lanjut. Sehingga pengembangan akan terus dilakukan setelah pembuatan tugas akhir.
* Cakupan sistem yang cukup luas, yaitu melakukan administrasi untuk administrator, peserta, pengajar, pelatihan, jadwal, absensi, dan nilai.
* Adanya tiga tipe pengguna dalam sistem, yaitu administrator, peserta, dan pengajar. Ketiga tipe tersebut mempunyai jarak umur yang berbeda-beda, sehingga membutuhkan *feedback* dari masing-masing pengguna.

## *Framework* dan Yii *Framework*

*Framework* adalah kumpulan dari *library* kode yang bertujuan untuk membantu pembuatan *software.* Pemakaian *framework*  berdasarkan alasan dimana membuat kode aplikasi dari awal (*scratch*) sangat tidak praktis. Melakukan penggunaan kode yang sudah ada lebih dapat diandalkan dan lebih aman [16]. *Software framework* juga secara signifikan dapat mengurangi waktu, tenaga, dan suber daya yang dipakai untuk membuat aplikasi web. Selain itu,  *framework*  banyak yang memakai arsitektur  *software* terbuka yang sudah diterima secara umum [17]. Beberapa kelebihan lain dari *framework* juga dijelaskan oleh Manger, Trejderowski, dan Paduch [18]. Beberapa kelebihan tersebut yaitu:

* Penggunaan kode yang telah dibuat, diuji, dan digunakan oleh *programmer* lain.
* Tidak perlu mempelajari API yang spesifik atau Bahasa pemrograman tingkat rendah (*low-level*), sehingga memberikan ruang fokus untuk pembuatan *business logic* dan implementasi.
* Akses terhadap kode yang terstandar dan dapat digunakan berulang.
* Akses terhadap *library* dari algoritma dan data struktur yang digunakan untuk memanipulasi hal tersebut.
* Mempunyai semua kelebihan dari *object-oriented programming*.
* *Clean coding* dengan menggunakan konsep MVC (*Model-View-Controller*).
* Terdapat *Database access abstraction layer* yang memungkinkan penulisan *database-independent code* dan merubah sistem penyimpanan data selama pengembangan dengan sedikit perubahan kode.
* Adanya *object oriented access* ke *database* yang memungkinkan pengguna untuk melakukan manipulasi data tanpa perlu memahami Bahasa pemanipulasi *database* (seperti SQL).
* *Code modularity,* yaitu memungkinkan kode yang telah dipakai pada satu aplikasi dapat digunakan pada aplikasi selanjutnya.
* *Library, class,* dan solusi yang siap pakai, seperti *user authorization authentication, security, database connectivity issues,* serta *caching.*
* Tidak terpaut pada platform, sistem, atau *browser* khusus.
* Performa sistem meningkat dengan digunakannya teknik *caching.*
* Pembuatan *front-end* yang lebih mudah.
* Dokumentasi lengkap dan dapat diakses secara bebas melalui internet atau buku-buku pendukung.
* Telah disiapkan secara baik untuk lokalisasi dan internasionalisasi dalam Bahasa.
* *Event-driven programming.*
* Mempunyai berbagai macam *extensions* yang memberikan solusi dan protokol standar untuk tiap area pengembangan.
* Validasi untuk data yang diinput oleh pengguna ataupun yang diambil dari *database.*
* *Error handling* dan *logging.*
* Skrip otomatis untuk pembuatan *basic skeleton* dari aplikasi.
* Konfigurasi yang terpisah, sederhana dan terstandar.

Yii merupakan sebuah *framework* yang menggunakan *pure Object-Oriented Programming* (OOP) [16]. Tidak seperti *framework* yang lain, Yii membutuhkan PHP versi 5. Arsitektur yang digunakan oleh Yii adalah MVC (*Model-View-Controller*) yang terbagi menjadi tiga bagian, yaitu *model, view, dan controller* [19]. Yang dimaksud dengan *model* adalah bagian yang memberikan dasar logika proses serta definisi relasi antara data di database dan aturan yang harus dijalankan ketika melakukan *read/save*. Model pada Yii adalah setiap *instance* dari *class* CModel atau *child class-*nya. Sedangkan pada bagian *view*, bertanggung jawab untuk melakukan *rendering* dari *user interface* berdasarkan data pada model dan merupakan kumpulan semua elemen yang terhubung. Bagian ini biasanya dibuat memakai HTML, dan beberapa PHP *statement* yang simpel untuk melakukan *conditional* dan *looping.* Bagian terakhir, yaitu *controller* berfungsi sebagai penghubung *model* dan *view* serta pengatur proses dan perubahan data. *Controller* berisi aksi-aksi yang dibutuhkan *user* untuk melakukan manipulasi data aplikasi *web*.



Gambar 2.II.2 Arsitektur MVC pada Yii

*(sumber:* [*http://shyjupv.wordpress.com/2013/03/15/mvc-architecture-in-yii-framework/*](http://shyjupv.wordpress.com/2013/03/15/mvc-architecture-in-yii-framework/)*)*

Pemakaian Yii sebagai *framework* didasarkan bahwa Yii satu-satunya diantara *popular framework* yang memiliki semua fitur untuk membuat *modern website* dan pengembangan *web application* [18]*.*  Hanya pemrograman prosedural yang tidak didukung oleh Yii. Hal ini memaksa pengembang untuk menggunakan teknik *object-oriented programming* yang merupakan kelebihan utama bagi *framework* ini.

## *Usability test*

Berdasarkan definisi *usability* dari *International Standards Organization (ISO),* Powell mendefinisikan *web site usability* sebagai “*the extent to which a site can be used by a specified group of users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use”* [20].Berdasarkan pernyataan Powell tersebut, dapat dikatakan bahwa *usability* adalah *user* dan *task dependent* serta berhubungan juga dengan bagaimana *user* dapat menyelesaikan apa yang mereka kerjakan, seberapa efisien *user* dapat mengerjakannya, dan tingkat kepuasan *user* sewaktu dan sesudah proses pengerjaan [21].

Palmer, yang dikenal sebagai pakar dalam *Human-Computer Interaction* [8] mengembangkan konstruk untuk melakukan *website usability test* [22]sebagai berikut:

* 1. Selang Waktu Penerimaan Data (*Download Delay)*.

Parameter yang diukur: Kecepatan awal akses dan kecepatan tampilan halaman.

* 1. Pengelolaan Navigasi Halaman (*Navigation / Organization)*

Parameter yang diukur: Pengaturan, Urutan Halaman, *Links*, *Layout* dan pengelolaan dan penempatan Navigasi.

* 1. Interaktivitas (*Interactivity)*

Parameter yang diukur: Kustomisasi halaman *web* dan Interaktivitas.

* 1. Responsivitas (*Responsiveness)*

Parameter yang diukur: Fasilitas *feedback* dan FAQ.

* 1. Informasi Materi *Website (Information / Content)*

Parameter yang diukur: Jumlah informasi, keragaman informasi, jumlah kata dan kualitas materi *website.*

* 1. *Website* yang berhasil (*User Satisfaction)*

Parameter yang diukur: Kepuasan pengguna, keinginan untuk mengakses kembali *website* dan frekuensi mengakses *website*.

Penelitian mengenai *usability* tidak berhenti pada Palmer saja. Green dan Pearson juga melakukan penelitian mengenai *usability* tersebut dan merumuskan 4 dimensi terbaik untuk mengukur tingkat kepuasan pengguna terhadap *website* [7]. Keempat dimensi variabel tersebut adalah:

1. Kemudahan (*Ease of Use*)
2. Personalisasi (*Customization*)
3. Kecepatan Akses pada Aplikasi (*Download Delay)*
4. Informasi (*Content)*

Selain didapat empat dimensi variabel tersebut, juga didapat suatu model penelitian yang menyimpulkan bahwa kepuasan pengguna (*User Satisfaction)* dalam mengakses *website* menjadi variabel endogen yang dapat mendorong pengunjung untuk mengakses kembali *website* tersebut.

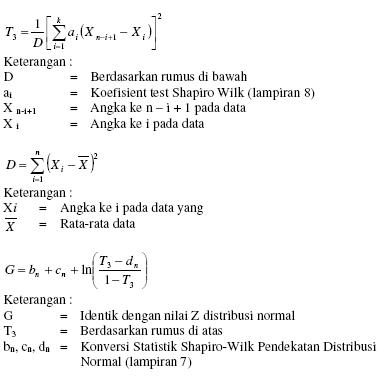
****

Gambar 2.II.3 Model penelitian Green dan Pearson

*(sumber: Prayoga & Sensuse, 2010 )*

## Uji Normalitas

Secara garis besar, normalitas atau *normal* data datang dari populasi yang mempunyai sebaran distribusi normal. Ketika melakukan penelitian statistik, asumsi dari normalitas merupakan hal yang fundamental bagi peneliti. Pentingnya sebasaran data terdistribusi normal tidak dapat dipungkiri, mengingat hal tersebut adalah basis bagi banyak prosedur statistik [23]. Namun dalam kehidupan nyata hampir tidak mungkin (*impossible*) dapat ditemukan data yang terdistribusi normal secara sempurna [24]. Oleh karena itu diperlukan adanya sebuah metode untuk menentukan apakah data yang didapat terdistribusi normal atau tidak. Pada penelitian ini, uji normalitas Shapiro-Wilk dipilih sebagai alat uji normalitas data. Tes normalitas Shapiro dan Wilk [25] merupakan tes yang pada awalnya dibuat sebagai alat uji dengan jumlah sampel kurang dari 50. Tes ini menjadi salah satu tes yang diandalkan berkat *good power properties* yang dimiliki [26]. Rumus dari uji normalitas Shapiro-Wilk [25] adalah:



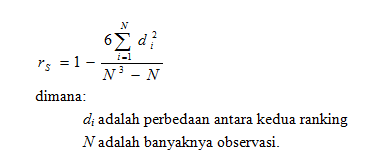
Gambar 2.II.4 Rumus Shapiro-Wilk

*(sumber:* [*http://arini2992.blogspot.co.id/2011/04/normal-0-false-false-false-en-us-x-none.html*](http://arini2992.blogspot.co.id/2011/04/normal-0-false-false-false-en-us-x-none.html)*)*

## Uji Korelasi

Uji korelasi adalah teknik statistik penghitungan dua variabel untuk menemukan ada atau tidaknya relasi diantara keduanya [27]. Pada penelitian ini, uji korelasi akan menggunakan uji korelasi Spearman. Uji korelasi Spearman merupakan uji nonparametrik yang diusulkan oleh Charles Spearman sebagai alat untuk mengukur kekuatan korelasi antar dua variabel [28]. Pengujian ini merupakan pengujian yang mempunyai sifat dapat menerima distribusi data apapun, baik normal maupun tidak normal. Dengan sifat tersebut memberi arti bahwa uji korelasi Spearman tidak melakukan asumsi tentang bentuk ataupun distribusi dari data sampel. Karena alasan tersebut, uji korelasi Spearman banyak dipakai ketika satu atau kedua variabel tidak terdistribusi normal [29].

Rumus dari uji korelasi Spearman adalah:



Gambar 2.II.5 Rumus uji korelasi Spearman

*(sumber: https://digensia.wordpress.com/2013/07/09/uji-korelasi-spearman/)*

## Tools

Dalam perancangan ini, *tools* yang dipakai adalah sebagai berikut:

* HTML, sebagai bahasa pemrograman untuk tampilan *view* dari laman *web.*
* CSS, sebagai bahasa pemrograman untuk mendeskripsikan bentuk tampilan dari laman *web.*
* Javascript, sebagai bahasa pemrograman untuk membuat laman *web* yang interaktif.
* PHP, sebagai pemrograman *server script* yang menyambungkan laman *web* dengan *database* dan membuat komunikasi data dinamis*.*
* *Use Case Diagram,* sebagai representasi bisnis model berdasarkan aktor dan kegunaan.
* *Activity Diagram,* sebagai representasi alur aktifitas pengguna dengan sistem.
* *Sequence Diagram,* sebagai representasi alur proses dari sistem.
* *Deployment Diagram,* sebagai representasi topologi proses ketika dilakukan implementasi.
* Visual Paradigm® Community Edition, sebagai aplikasi pembuat diagram UML.
* Yii *framework,* sebagai kerangka kerja perancangan sistem yang mempunyai metode MVC (*model-view-controller).*
* XAMPP, sebagai aplikasi penyedia Apache HTTP *server* dan *database* MySQL.
* SublimeText, sebagai *coding* editor.
* Google Chrome dan Mozilla Firefox, sebagai aplikasi *browser* yang digunakan untuk menjalankan sistem *web* dan melakukan *testing.*
* IBM SPSS® *Statistics*, sebagai aplikasi penghitung data statistik.

# METODE PENELITIAN

Proses rancang bangun dan *usability test* penelitian ini dibuat berdasarkan metode *protoyping*. Tahap-tahap perancangan aplikasi yang digunakan yaitu *communication, planning and design, prototyping,* dan *feedback.* Karena metode yang digunakan bersifat iteratif, penelitian ini hanya terbatas pada satu kali putaraan siklus model. Penelitian hanya sampai pada proses *feedback* pertama, yaitu dengan menggunakan *usability test.*

## *Communication*

Pada tahap ini dilakukan komunikasi antara penulis dan pihak Divisi Diklat Harapan Kita Jakarta. Tahap *communication* ini mempunyai tujuan untuk mengumpulkan data-data yang dibutuhkan dalam membuat analisis dan rancangan sistem administrasi.

Data-data yang dibutuhkan dan digunakan dalam penelitian ini dikumpulkan dengan metode:

* 1. Wawancara

Metode ini dilakukan dengan cara melakukan tanya jawab secara langsung terhadap nara sumber yang berhubungan dengan keadaan Divisi Diklat Rumah Sakit Harapan Kita Jakarta. Nara sumber yang menjadi acuan dalam metode ini adalah Bapak dr.Dr. Ismoyo Sunu SpJp(K) yang merupakan kepala dari Divisi Diklat Rumah Sakit Harapan Kita Jakarta. Hasil wawancara berupa data kebutuhan sistem dan elitasi yang terdapat pada lampiran 2.

* 1. Dokumen

Pengumpulan data melalui dokumen-dokumen yang didapatkan dari Divisi Diklat Rumah Sakit Harapan Kita Jakarta.

* 1. Observasi

Pengumpulan data melalui observasi secara langsung di tempat Divisi Diklat Rumah Sakit Harapan Kita Jakarta.

Dalam tahap *communication* ini, didapat beberapa objek-objek atau *entity* yang terkait dengan perancangan sistem. Objek-objek tersebut adalah pengguna, yang dalam hal ini terbagi menjadi tiga, Administrator, Pengajar, dan Peserta. Selain pengguna juga terdapat Pelatihan-pelatihan, jadwal pelatihan, absensi, nilai, serta pembayaran.

## *Planning and Design*

Dalam proses sebelumnya, pihak Divisi Diklat Rumah Sakit Harapan Kita Jakarta  memberikan informasi sistem administrsi yang akan dibangun. Informasi ini terdiri dari fitur dan data yang diperlukan untuk membangun sistem informasi tersebut. Detail fitur dan data yang diperlukan untuk setiap proses tersebut adalah:

1. Pendaftaran

Dalam hal ini, Divisi Diklat Rumah Sakit Harapan Kita Jakarta membutuhkan sebuah fitur pendaftaran untuk setiap pelatihan yang diadakan. Setiap calon peserta pelatihan akan diminta untuk mengisi formulir pendaftaran yang terdapat pada sistem. Pada fitur ini, Divisi Diklat Rumah Sakit Harapan Kita Jakarta juga meminta untuk dapat mengetahui dan mencatat semua pendaftaran yang telah dibuat, beserta waktu pembayaran calon peserta.

1. Data Peserta

Untuk fitur data peserta, Divisi Diklat Rumah Sakit Harapan Kita Jakarta meminta untuk dapat melihat siapa saja yang telah mendaftar dan telah menjalani atau sedang melakukan pelatihan. Divisi Diklat Rumah Sakit Harapan Kita Jakarta juga meminta aplikasi dapat memanipulasi data peserta, sepeti membuat, mengedit, dan menghapus. Dalam hal ini, Divisi Diklat Rumah Sakit Harapan Kita Jakarta menginginkan sebuah tampilan data peserta yang mudah dioperasikan.

1. Data Pengajar

Sama seperti data peserta, Divisi Diklat Rumah Sakit Harapan Kita Jakarta meminta untuk dapat melihat siapa saja pengajar yang terdaftar dan melakukan pengajaran. Berbeda dengan data peserta, untuk data pengajar ini Divisi Diklat Rumah Sakit Harapan Kita Jakarta meminta agar dapat menampilkan nilai untuk setiap peserta yang mengikuti pelajaran yang diberikan pengajar.

1. Absensi

Pada absensi, Divisi Diklat Rumah Sakit Harapan Kita Jakarta meminta untuk dapat memantau absensi yang dijalankan oleh pengajar dan peserta untuk setiap pelajaran pelatihan yang diberikan oleh divisi diklat. Absensi dapat dilihat berdasarkan peserta yang dipilih.

1. Nilai

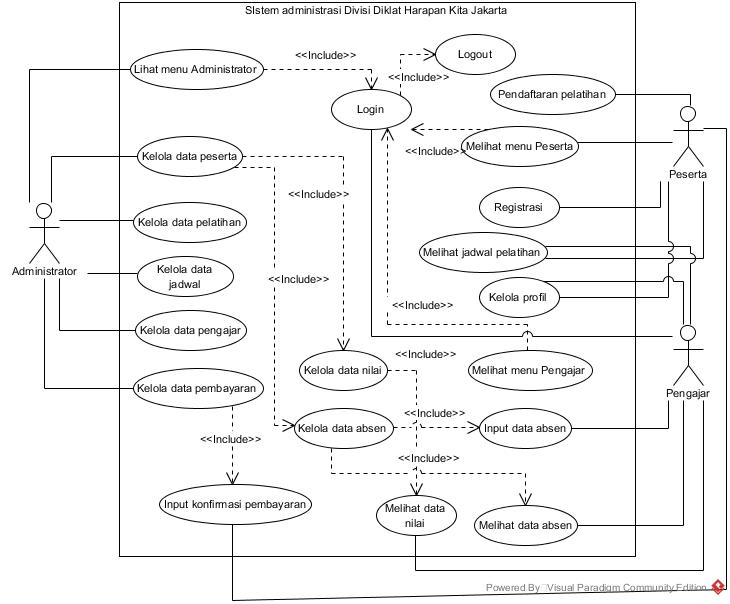
Untuk nilai, Divisi Diklat Rumah Sakit Harapan Kita Jakarta menginginkan agar setiap peserta dan pengajar dapat memantau nilai yang ada melalui sistem. Divisi Diklat Rumah Sakit Harapan Kita Jakarta juga meminta agar dapat memasukkan nilai ke dalam sistem.

Berdasarkan hasil *interview* dan pengamatan awal penulis, didapatkan identifikasi fitur-fitur yang dibutuhkan oleh *user*. Analisa mengenai rancangan sistem awal pada proposal ini hanya akan menggunakan *Use case* diagram. Adapun analisa dari fitur-fitur yang dibutuhkan dapat dilihat pada tabel III.1.

Tabel 3.1 Analisa fitur yang dibutuhkan

|  |  |  |
| --- | --- | --- |
| No. | Nama Fitur | Deskripsi Fitur |
| 1. 2. | Registrasi | Proses ini dibutuhkan untuk melakukan pendaftaran bagi calon peserta baru.  Data Input: *Username, Password*, Nama, NIP/NOPEG, Jenis Kelamin, Tempat & Tanggal lahir, Alamat Tempat Tinggal, Nomor Telepon/HP, Agama, Pangkat/Golongan, Pendidikan Terakhir, Alamat Institusi/Universitas. Instansi, Alamat Instansi, Nomor Telepon Instansi. |
| 1. 3. | Login | Proses ini dibutuhkan untuk melakukan akses kedalam sistem sesuai dengan hak login yang diterima.  Data Input: *Username, Password* |
| 1. 4. | Logout | Proses ini dibutuhkan untuk menghentikan akses kedalam sistem. |
| 1. 5. | Input dan Mengelola data Peserta/Pengajar | Proses ini dibutuhkan untuk melakukan pengelolaan pada data Peserta/Pengajar. Pengelolaan tidak terbatas pada data lama, tetapi juga data baru. Termasuk dalam proses ini adalah input dan mengelola absensi pengajar, input pembayaran peserta, input nilai peserta, dan input absensi Peserta. |
| 1. 6. | Input Pembayaran Peserta | Proses ini dibutuhkan untuk mendapatkan data pembayaran setelah calon peserta membayar biaya pelatihan yang diinginkan. Dalam proses ini juga dibutuhkan waktu transaksi dan rekening yang diberikan oleh peserta.  Data Input: Nomor Rekening, Waktu Pembayaran, Nama Pembayar |
| 1. 7. | Input dan Mengelola Jadwal Pelatihan | Proses ini dibutuhkan sebagai pengatur jadwal pelatihan yang diadakan divisi diklat rumah sakit Harapan Kita. Jadwal pelatihan akan ditampilkan pada sistem agar *user* selain administrator dapat mengetahui informasi tersebut.  Data Input: Nama pelatihan, tanggal pelatihan, tahun pelatihan |
| 1. 8. | Input dan Mengelola Absensi Pengajar | Proses ini dibutuhkan untuk membuat, melihat, menambah, dan menghapus absensi pengajar pada setiap jadwal pelatihan yang diadakan.  Data Input: Tanggal absen, status absen, nama pelatihan. |
| 1. 9. | Input Nilai Peserta | Proses ini dibutuhkan untuk melakukan input data nilai peserta pada setiap pelatihan yang diadakan oleh *user*.  Data Input: Nama peserta, nama pelatihan, nilai |
| 1. 10. | Input Absensi Peserta | Proses ini dibutuhkan untuk melakukan input data absensi peserta pada setiap pelatihan yang diadakan oleh *user.*  Data Input : Nama peserta, nama pelatihan, tanggal pelatihan status absen |
| 1. 11. | Melihat Data Pengajar | Proses ini dibutuhkan agar dapat melihat data yang berhubungan dengan pengajar dan proses pengajaran. |
| 1. 12. | Melihat Jadwal Pelatihan | Proses ini dibutuhkan agar peserta dan pengajar dapat melihat jadwal pelatihan yang diadakan oleh *user.* |
| 1. 13. | Registrasi Pelatihan | Proses ini dibutuhkan agar peserta dapat melakukan registrasi pelatihan baru yang akan dilakukan.  Data Input: Nama Pelatihan, Tanggal Pelatihan. |
| 1. 14. | Melihat Data Peserta | Proses ini dibutuhkan agar dapat melihat data yang berhubungan dengan peserta dan kepesertaannya. |

*Use Case diagram* rancangan awal sistem administrasi berdasarkan hasil *interview* divisi diklat rumah sakit Harapan Kita dapat dilihat pada gambar III.1.



Gambar 3.III.1 Use Case Diagram analisa fitur sistem

Penjelasan dari gambar *use* diatas dijelaskan melalui *use case scenario table*. *Use case scenario table* dapat menjelaskan perilaku setiap *use case* dan reaksi aktor serta sistem.

Tabel 3.2 Use Case Scenario Login

|  |  |  |
| --- | --- | --- |
| ***Use Case Name*** | Login | |
| ***Description*** | *Use case* ini menjelaskan bagaimana pengguna dapat masuk ke dalam sistem. | |
| ***Actor*** | Administrator, Peserta, Pengajar | |
| ***Pre-Condition*** | Pengguna mengakses alamat *website* sistem administrasi Divisi Diklat Harapan Kita Jakarta | |
| ***Typical of events*** | ***Actor Action*** | ***System Response*** |
| 1. Membuka *website* | 1. Menampilkan form |
| 1. Pengguna memasukkan *ID* dan *password* | 1. Menampilkan menu utama |
| ***Alternate Course*** | Jika *ID* dan *password* salah, maka sistem akan menampilkan notifikasi kesalahan dan pengguna harus memasukkan kembali *ID* dan *password* kembali. | |
| ***Post-Condition*** | Sistem menampilkan menu utama sesuai dengan tipe pengguna. | |

Tabel 3.3 Use Case Scenario Logout

|  |  |  |
| --- | --- | --- |
| ***Use Case Name*** | Logout | |
| ***Description*** | *Use case* ini menjelaskan bagaimana pengguna dapat keluar dari sistem. | |
| ***Actor*** | Administrator, Peserta, Pengajar | |
| ***Pre-Condition*** | Pengguna berhasil masuk ke dalam sistem. | |
| ***Typical of events*** | ***Actor Action*** | ***System Response*** |
| 1. Memilih menu “Logout”. | 1. Menghapus *session* dan keluar dari sistem. |
|  | 1. Menampilkan menu *Home* sistem. |
| ***Alternate Course*** | Jika *ID* dan *password* salah, maka sistem akan menampilkan notifikasi kesalahan dan pengguna harus memasukkan kembali *ID* dan *password* kembali. | |
| ***Post-Condition*** | Sistem menampilkan menu utama sesuai dengan tipe pengguna. | |

Tabel 3.4 Use Case Scenario melihat menu Administrator

|  |  |  |
| --- | --- | --- |
| ***Use Case Name*** | Melihat menu Administrator | |
| ***Description*** | *Use case* ini menjelaskan bagaimana pengguna bertipe administrator melihat menu *dashboard* admin. | |
| ***Actor*** | Administrator | |
| ***Pre-Condition*** | Pengguna tipe administrator berhasil masuk ke dalam sistem. | |
| ***Typical of events*** | ***Actor Action*** | ***System Response*** |
|  | 1. Menampilkan menu *dashboard* tipe administrator. |
| 1. Melihat dan memilih menu. |  |
| ***Alternate Course*** | - | |
| ***Post-Condition*** | Sistem menampilkan menu utama *dashboard* admin. | |

Tabel 3.5 Use Case Scenario melihat menu Peserta

|  |  |  |
| --- | --- | --- |
| ***Use Case Name*** | Melihat menu Peserta | |
| ***Description*** | *Use case* ini menjelaskan bagaimana pengguna bertipe peserta melihat menu *dashboard* peserta. | |
| ***Actor*** | Peserta | |
| ***Pre-Condition*** | Pengguna tipe peserta berhasil masuk ke dalam sistem. | |
| ***Typical of events*** | ***Actor Action*** | ***System Response*** |
|  | 1. Menampilkan menu *dashboard* tipe peserta. |
| 1. Melihat dan memilih menu. |  |
| ***Alternate Course*** | - | |
| ***Post-Condition*** | Sistem menampilkan menu utama *dashboard* peserta. | |

Tabel 3.6 Use Case Scenario melihat menu Pengajar

|  |  |  |
| --- | --- | --- |
| ***Use Case Name*** | Melihat menu Pengajar | |
| ***Description*** | *Use case* ini menjelaskan bagaimana pengguna bertipe pengajar melihat menu *dashboard* pengajar. | |
| ***Actor*** | Pengajar | |
| ***Pre-Condition*** | Pengguna tipe pengajar berhasil masuk ke dalam sistem. | |
| ***Typical of events*** | ***Actor Action*** | ***System Response*** |
|  | 1. Menampilkan menu *dashboard* tipe peserta. |
| 1. Melihat dan memilih menu. |  |
| ***Alternate Course*** | - | |
| ***Post-Condition*** | Sistem menampilkan menu utama *dashboard* pengajar. | |

Tabel 3.7 Use Case Scenario kelola data Peserta

|  |  |  |
| --- | --- | --- |
| ***Use Case Name*** | Kelola data Peserta | |
| ***Description*** | *Use case* ini menjelaskan bagaimana pengguna bertipe administrator mengelola data peserta. | |
| ***Actor*** | Administrator | |
| ***Pre-Condition*** | Pengguna bertipe administrator berhasil masuk ke dalam sistem dan melihat menu administrator. | |
| ***Typical of events*** | ***Actor Action*** | ***System Response*** |
| 1. Memilih menu “Peserta”. | 1. Menampilkan semua data peserta. |
| 1. Melakukan manipulasi (CRUD) data peserta yang dipilih. | 1. Menjalankan fungsi manipulasi data sesuai permintaan pengguna. |
| ***Alternate Course*** | - | |
| ***Post-Condition*** | Sistem menampilkan semua data peserta. Pengguna bertipe administrator dapat memilih fungsi manipulasi data yang tersedia. | |

Tabel 3.8 Use Case Scenario kelola data jadwal

|  |  |  |
| --- | --- | --- |
| ***Use Case Name*** | Kelola data Pelatihan | |
| ***Description*** | *Use case* ini menjelaskan bagaimana pengguna bertipe administrator mengelola data pelatihan. | |
| ***Actor*** | Administrator | |
| ***Pre-Condition*** | Pengguna bertipe administrator berhasil masuk ke dalam sistem dan melihat menu administrator. | |
| ***Typical of events*** | ***Actor Action*** | ***System Response*** |
| 1. Memilih menu “Pelatihan”. | 1. Menampilkan semua data pelatihan. |
| 1. Melakukan manipulasi (CRUD) data pelatihan yang dipilih | 1. Menjalankan fungsi manipulasi data sesuai permintaan pengguna. |
| ***Alternate Course*** | - | |
| ***Post-Condition*** | Sistem menampilkan semua data pelatihan. Pengguna bertipe administrator dapat memilih fungsi manipulasi data yang tersedia. | |

Tabel 3.9 Use Case Scenario kelola data jadwal

|  |  |  |
| --- | --- | --- |
| ***Use Case Name*** | Kelola data Jadwal | |
| ***Description*** | *Use case* ini menjelaskan bagaimana pengguna bertipe administrator mengelola data jadwal dari pelatihan. | |
| ***Actor*** | Administrator | |
| ***Pre-Condition*** | Pengguna bertipe administrator berhasil masuk ke dalam sistem dan melihat menu administrator. | |
| ***Typical of events*** | ***Actor Action*** | ***System Response*** |
| 1. Memilih menu “Jadwal”. | 1. Menampilkan semua data jadwal pelatihan. |
| 1. Melakukan manipulasi (CRUD) data jadwal yang dipilih | 1. Menjalankan fungsi manipulasi data sesuai permintaan pengguna. |
| ***Alternate Course*** | - | |
| ***Post-Condition*** | Sistem menampilkan semua data jadwal. Pengguna bertipe administrator dapat memilih fungsi manipulasi data yang tersedia. | |

Tabel 3.10 Use Case Scenario kelola data pengajar

|  |  |  |
| --- | --- | --- |
| ***Use Case Name*** | Kelola data Pengajar | |
| ***Description*** | *Use case* ini menjelaskan bagaimana pengguna bertipe administrator mengelola data pengajar. | |
| ***Actor*** | Administrator | |
| ***Pre-Condition*** | Pengguna bertipe administrator berhasil masuk ke dalam sistem dan melihat menu administrator. | |
| ***Typical of events*** | ***Actor Action*** | ***System Response*** |
| 1. Memilih menu “Pengajar”. | 1. Menampilkan semua data pengajar. |
| 1. Melakukan manipulasi (CRUD) data pengajar yang dipilih | 1. Menjalankan fungsi manipulasi data sesuai permintaan pengguna. |
| ***Alternate Course*** | - | |
| ***Post-Condition*** | Sistem menampilkan semua data pengajar. Pengguna bertipe administrator dapat memilih fungsi manipulasi data yang tersedia. | |

Tabel 3.11 Use Case Scenario kelola data pembayaran

|  |  |  |
| --- | --- | --- |
| ***Use Case Name*** | Kelola data Pembayaran | |
| ***Description*** | *Use case* ini menjelaskan bagaimana pengguna bertipe administrator mengelola data pembayaran. | |
| ***Actor*** | Administrator | |
| ***Pre-Condition*** | Pengguna bertipe administrator berhasil masuk ke dalam sistem dan melihat menu administrator. | |
| ***Typical of events*** | ***Actor Action*** | ***System Response*** |
| 1. Memilih menu “Pembayaran”. | 1. Menampilkan semua data pembayaran. |
| 1. Melakukan manipulasi (CRUD) data pembayaran yang dipilih | 1. Menjalankan fungsi manipulasi data sesuai permintaan pengguna. |
| ***Alternate Course*** | - | |
| ***Post-Condition*** | Sistem menampilkan semua data pembayaran. Pengguna bertipe administrator dapat memilih fungsi manipulasi data yang tersedia. | |

Tabel 3.12 Use Case Scenario kelola data nilai

|  |  |  |
| --- | --- | --- |
| ***Use Case Name*** | Kelola data Nilai | |
| ***Description*** | *Use case* ini menjelaskan bagaimana pengguna bertipe administrator mengelola data nilai. | |
| ***Actor*** | Administrator | |
| ***Pre-Condition*** | Pengguna bertipe administrator berhasil masuk ke dalam sistem dan melihat menu administrator. | |
| ***Typical of events*** | ***Actor Action*** | ***System Response*** |
| 1. Memilih menu “Nilai”. | 1. Menampilkan semua data nilai. |
| 1. Melakukan manipulasi (CRUD) data nilai yang dipilih. | 1. Menjalankan fungsi manipulasi data sesuai permintaan pengguna. |
| ***Alternate Course*** | - | |
| ***Post-Condition*** | Sistem menampilkan semua data nilai. Pengguna bertipe administrator dapat memilih fungsi manipulasi data yang tersedia. | |

Tabel 3.13 Use Case Scenario kelola data absen

|  |  |  |
| --- | --- | --- |
| ***Use Case Name*** | Kelola data absen | |
| ***Description*** | *Use case* ini menjelaskan bagaimana pengguna bertipe administrator mengelola data absen. | |
| ***Actor*** | Administrator | |
| ***Pre-Condition*** | Pengguna bertipe administrator berhasil masuk ke dalam sistem dan melihat menu administrator. | |
| ***Typical of events*** | ***Actor Action*** | ***System Response*** |
| 1. Memilih menu “Absen”. | 1. Menampilkan semua data absen. |
| 1. Melakukan manipulasi (CRUD) data absen yang dipilih | 1. Menjalankan fungsi manipulasi data sesuai permintaan pengguna. |
| ***Alternate Course*** | - | |
| ***Post-Condition*** | Sistem menampilkan semua data absen. Pengguna bertipe administrator dapat memilih fungsi manipulasi data yang tersedia. | |

Tabel 3.14 Use Case Scenario registrasi

|  |  |  |
| --- | --- | --- |
| ***Use Case Name*** | Registrasi | |
| ***Description*** | *Use case* ini menjelaskan bagaimana pengguna bertipe peserta melakukan pendaftaran akun pada sistem. | |
| ***Actor*** | Peserta | |
| ***Pre-Condition*** | Pengguna bertipe peserta mengakses alamat *website* sistem administrasi Divisi Diklat Harapan Kita Jakarta | |
| ***Typical of events*** | ***Actor Action*** | ***System Response*** |
| 1. Memilih menu “Registrasi”. | 1. Menampilkan form registrasi. |
| 1. Mengisi form registrasi dan menekan tombol “register” | 1. Menampilkan laman “sukses registrasi” dan menampilkan laman utama. |
| ***Alternate Course*** | Apabila terjadi kesalahan saat mengisi form, sistem akan memberikan peringatan adanya kesalahan. Sistem akan menampilkan form kembali. | |
| ***Post-Condition*** | Sistem menampilkan laman utama *website*. | |

Tabel 3.15 Use Case Scenario melihat jadwal pelatihan

|  |  |  |
| --- | --- | --- |
| ***Use Case Name*** | Melihat jadwal pelatihan | |
| ***Description*** | *Use case* ini menjelaskan bagaimana pengguna bertipe peserta melihat jadwal pelatihan yang terdapat pada sistem administrasi. | |
| ***Actor*** | Peserta | |
| ***Pre-Condition*** | Pengguna bertipe peserta mengakses alamat *website* sistem administrasi Divisi Diklat Harapan Kita Jakarta | |
| ***Typical of events*** | ***Actor Action*** | ***System Response*** |
| 1. Memilih menu “Jadwal”. | 1. Menampilkan semua data jadwal pelatihan. |
| ***Alternate Course*** | - | |
| ***Post-Condition*** | Sistem menampilkan laman utama *website*. | |

Tabel 3.16 Use Case Scenario kelola profil

|  |  |  |
| --- | --- | --- |
| ***Use Case Name*** | Kelola profil | |
| ***Description*** | *Use case* ini menjelaskan bagaimana pengguna bertipe peserta atau pengajar mengelola data profil. | |
| ***Actor*** | Peserta, Pengajar | |
| ***Pre-Condition*** | Pengguna bertipe peserta atau pengajar berhasil masuk ke dalam sistem dan melihat melihat *dashboard* menu. | |
| ***Typical of events*** | ***Actor Action*** | ***System Response*** |
| 1. Memilih menu “Profil”. | 1. Menampilkan semua data profil. |
| 1. Melakukan manipulasi data profil yang dipilih | 1. Menjalankan fungsi manipulasi data sesuai permintaan pengguna. |
| ***Alternate Course*** | - | |
| ***Post-Condition*** | Sistem menampilkan semua data profil yang berhubungan dengan pengguna. Pengguna bertipe peserta dan pengajar dapat memilih fungsi manipulasi data yang tersedia. | |

Tabel 3.17 Use Case Scenario pendaftaran pelatihan

|  |  |  |
| --- | --- | --- |
| ***Use Case Name*** | Pendaftaran pelatihan | |
| ***Description*** | *Use case* ini menjelaskan bagaimana pengguna bertipe peserta mengelola data profil. | |
| ***Actor*** | Peserta | |
| ***Pre-Condition*** | Pengguna bertipe peserta berhasil masuk ke dalam sistem dan melihat *dashboard* menu. | |
| ***Typical of events*** | ***Actor Action*** | ***System Response*** |
| 1. Memilih menu “Pendaftaran”. | 1. Menampilkan form pendaftaran pelatihan. |
| 1. Melakukan input pelatihan yang dipilih dan menekan tombol “daftar”. | 1. Melakukan pembuatan entri peserta untuk pelatihan yang dipilih. |
| ***Alternate Course*** | Apabila pengguna bertipe peserta sudah melakukan pendaftaran pada pelatihan yang dipilih, sistem akan menampilkan peringatan bahwa pelatihan sudah didaftarkan. Pengguna bertipe peserta melakukan input pelatihan yang lain. | |
| ***Post-Condition*** | Sistem menampilkan laman “sukses daftar”. | |

Tabel 3.18 Use Case Scenario konfirmasi pembayaran

|  |  |  |
| --- | --- | --- |
| ***Use Case Name*** | Input konfirmasi pembayaran | |
| ***Description*** | *Use case* ini menjelaskan bagaimana pengguna bertipe peserta melakukan input konfirmasi pembayaran. | |
| ***Actor*** | Peserta | |
| ***Pre-Condition*** | Pengguna bertipe peserta berhasil masuk ke dalam sistem dan melihat melihat *dashboard* menu. | |
| ***Typical of events*** | ***Actor Action*** | ***System Response*** |
| 1. Memilih menu “Pembayaran”. | 1. Menampilkan form pembayaran pelatihan. |
| 1. Melakukan input konfirmasi pembayaran dan menekan tombol “konfirmasi”. | 1. Melakukan pembuatan entri konfirmasi pembayaran peserta untuk pelatihan yang dipilih. |
| ***Alternate Course*** | - | |
| ***Post-Condition*** | Sistem menampilkan laman “sukses bayar”. | |

Tabel 3.19 Use Case Scenario input data absen

|  |  |  |
| --- | --- | --- |
| ***Use Case Name*** | Input data absen | |
| ***Description*** | *Use case* ini menjelaskan bagaimana pengguna bertipe pengajar melakukan input data absen. | |
| ***Actor*** | Pengajar | |
| ***Pre-Condition*** | Pengguna bertipe pengajar berhasil masuk ke dalam sistem dan melihat melihat *dashboard* menu. | |
| ***Typical of events*** | ***Actor Action*** | ***System Response*** |
| 1. Memilih menu “input absen”. | 1. Menampilkan form input absen. |
| 1. Melakukan input absen berdasarkan pelatihan dan peserta serta menekan tombol “simpan”. | 1. Melakukan pembuatan entri konfirmasi pembayaran peserta untuk pelatihan yang dipilih. |
| ***Alternate Course*** | Apabila terjadi kesalahan saat mengisi form, sistem akan memberikan peringatan adanya kesalahan. Sistem akan menampilkan form kembali. | |
| ***Post-Condition*** | Sistem menampilkan laman “absen”. Dimana laman tersebut akan menampilkan semua data absen beserta data absen yang baru. | |

Tabel 3.20 Use Case Scenario melihat data absen

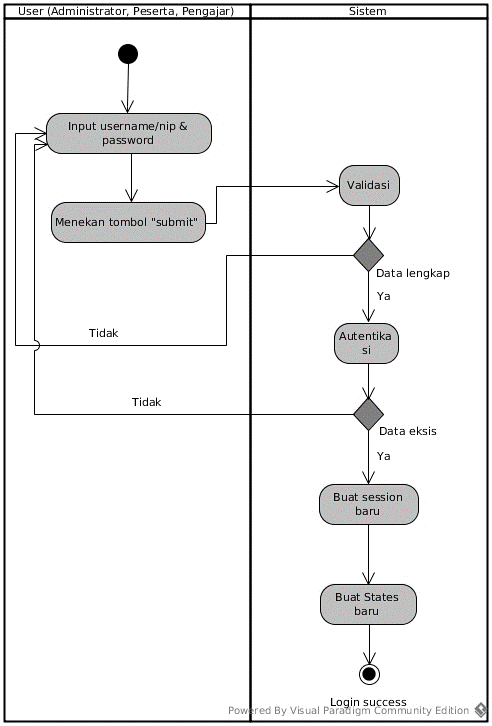
|  |  |  |
| --- | --- | --- |
| ***Use Case Name*** | Melihat data absen | |
| ***Description*** | *Use case* ini menjelaskan bagaimana pengguna bertipe pengajar melihat data absen. | |
| ***Actor*** | Pengajar | |
| ***Pre-Condition*** | Pengguna bertipe pengajar berhasil masuk ke dalam sistem dan melihat melihat *dashboard* menu. | |
| ***Typical of events*** | ***Actor Action*** | ***System Response*** |
| 1. Memilih menu “absen”. | 1. Menampilkan form semua data absen berdasarkan pengguna bertipe pengajar. |
| ***Alternate Course*** | - | |
| ***Post-Condition*** | Sistem menampilkan semua data absen berdasarkan pelatihan yang diajarkan oleh pengguna bertipe pengajar. | |

Tabel 3.21 Use Case Scenario melihat data nilai

|  |  |  |
| --- | --- | --- |
| ***Use Case Name*** | Melihat data nilai | |
| ***Description*** | *Use case* ini menjelaskan bagaimana pengguna bertipe pengajar melihat data nilai. | |
| ***Actor*** | Pengajar | |
| ***Pre-Condition*** | Pengguna bertipe pengajar berhasil masuk ke dalam sistem dan melihat melihat *dashboard* menu. | |
| ***Typical of events*** | ***Actor Action*** | ***System Response*** |
| 1. Memilih menu “nilai”. | 1. Menampilkan form semua data nilai berdasarkan pengguna bertipe pengajar. |
| ***Alternate Course*** | - | |
| ***Post-Condition*** | Sistem menampilkan semua data nilai berdasarkan pelatihan yang diajarkan oleh pengguna bertipe pengajar. | |

Agar dapat lebih memahami proses yang terjadi di dalam sistem administrasi Rumah Sakit Harapan Kita Jakarta, diperlukan sebuah penggambaran dalam bentuk alur kerja yang lebih detail untuk masing-masing proses yang terjadi di dalamnya. Penggambaran alur kerja rancangan sistem menggunakan *Activity Diagram* yang terbagi menjadi proses Login, Register, Create, Read, Update, Delete, Pendaftaran pelatihan. Untuk proses yang terdapat pada diagram *Use case* sebelumnya, pada dasarnya terdiri dari proses Create, Read, Update dan Delete. Hak akses Keempat proses tersebut akan berbeda-beda bagi setiap level pengguna.

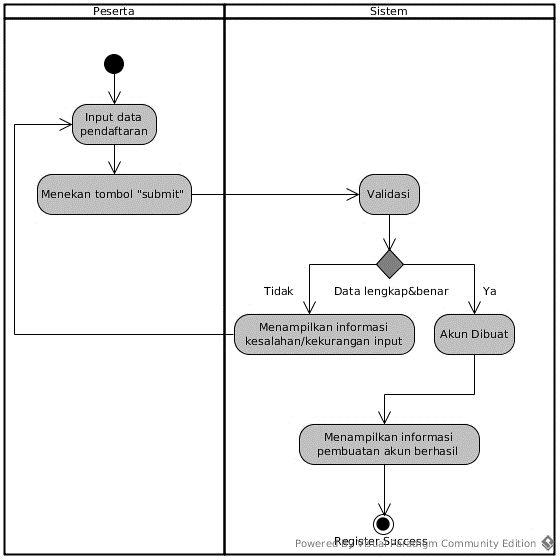
### *Login*



Gambar 3.III.2 Activity diagram login

Dalam proses ini pertama-tama pengguna, yang terdiri dari admin, pengajar, dan peserta, akan melakukan input username/nip dan password. Input data dilakukan dengan form yang telah disediakan. Apabila pengguna telah selesai memasukkan input data, sistem akan melakukan validasi, apakah data yang dimasukkan oleh pengguna sesuai atau lengkap. Apabila terdapat data yang tidak sesuai atau kurang lengkap, sistem akan meminta pengguna untuk melakukan input kembali. Apabila data telah lengkap, maka sistem akan melakukan autentikasi, yang merupakan pengecekan apakah data yang dimasukkan ada atau tidak di database. Apabila tidak ada, seperti proses validasi, sistem akan meminta pengguna untuk melakukan input kembali. Setelah proses autentikasi selesai, maka sistem akan membuat session baru yang sesuai dengan pengguna, dan membuat states yang berisi level akses.

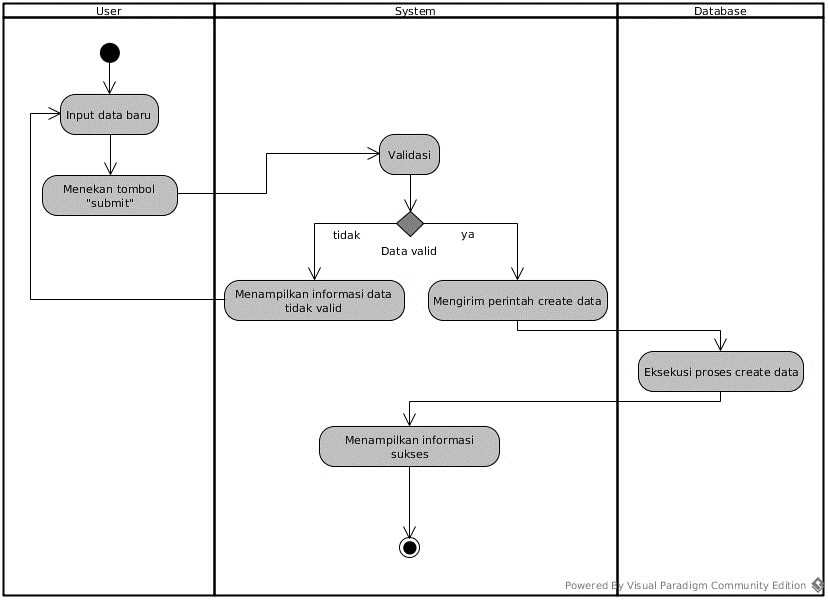
### *Register*

****

Gambar 3.III.3 Activity diagram Register

Proses registrasi tidak melibatkan pengguna lain kecuali peserta. Dalam proses ini peserta pertama-tama akan memasukkan data-data yang dibutuhkan untuk melakukan pendaftaran seperti nip, nama, dan alamat. Input data dimasukkan melalui form yang telah disediakan sistem. Setelah data dimasukkan, peserta akan menekan tombol “submit” agar data diproses oleh sistem. Validasi akan dilakukan oleh sistem. Apabila data valid, lengkap dan benar, maka sistem akan membuat data akun baru yang disimpan di database lalu sistem akan menampilkan informasi sukses kepada peserta. Apabila data tidak valid, kurang lengkap dan tidak benar, maka sistem hanya akan memberikan informasi kesalahan atau kekurangan data dan meminta peserta untuk melakukan input kembali.

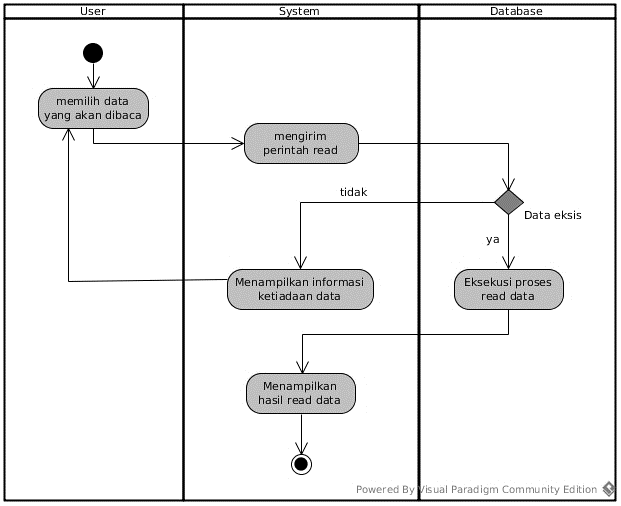
### *Create*



Gambar 3.III.4 Activity diagram create

Proses *create* dimulai dengan pengguna melakukan input data baru. Setelah pengguna melakukan input dan melakukan submit, sistem akan melakukan validasi. Pada proses validasi, sistem akan mengecek apakah data valid (lengkap dan benar). Apabila data tidak valid, maka sistem akan menampilkan informasi data tidak valid dan meminta pengguna untuk melakukan input ulang. Sistem akan mengirim perintah create data apabila data yang di input telah valid dan benar ke database. Setelah itu, database akan melakukan eksekusi proses create data dan sistem akan menampilkan informasi sukses ke pengguna.

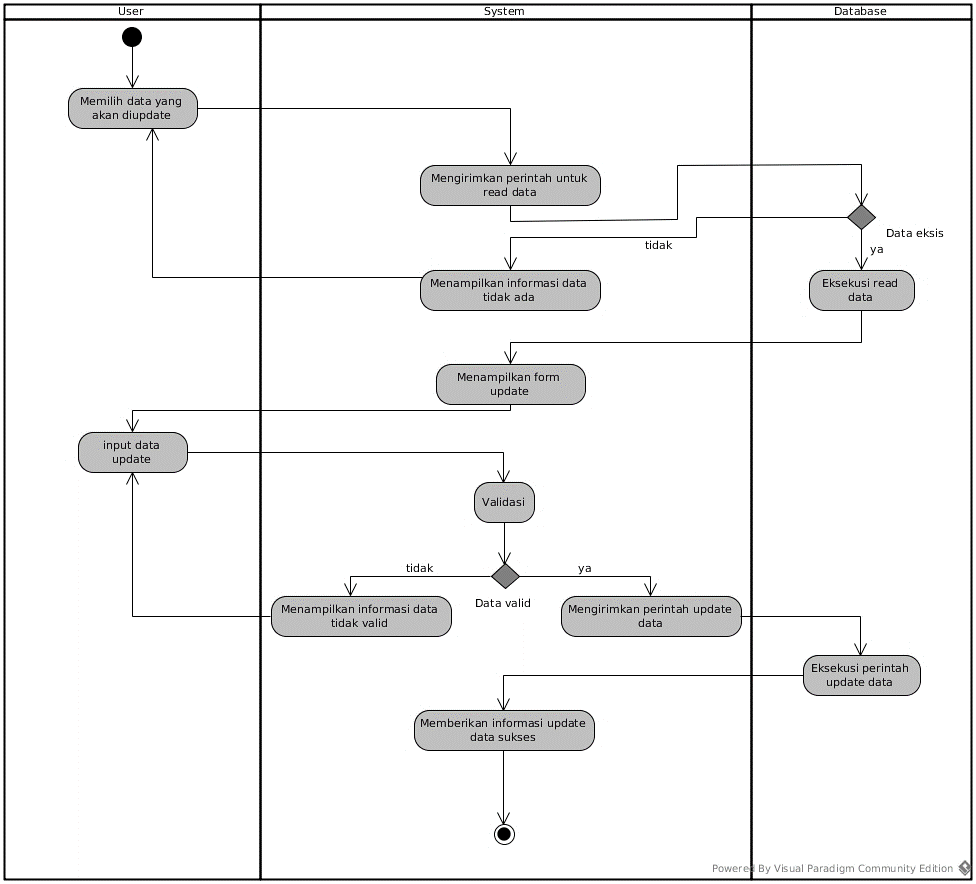
### *Read*



Gambar 3.III.5 Activity diagram read

Dalam proses *read*, pengguna akan memilih data yang akan dibaca. Setelah pemilihan selesai, sistem akan mengirim perintah read ke database. Apabila data tidak eksis, sistem akan menampilkan informasi ketiadaan data ke pengguna dan meminta untuk memilih data kembali. Sedangkan apabila data eksis maka database akan melakukan eksekusi read data dan sistem akan menampilkan hasil read ke pengguna.

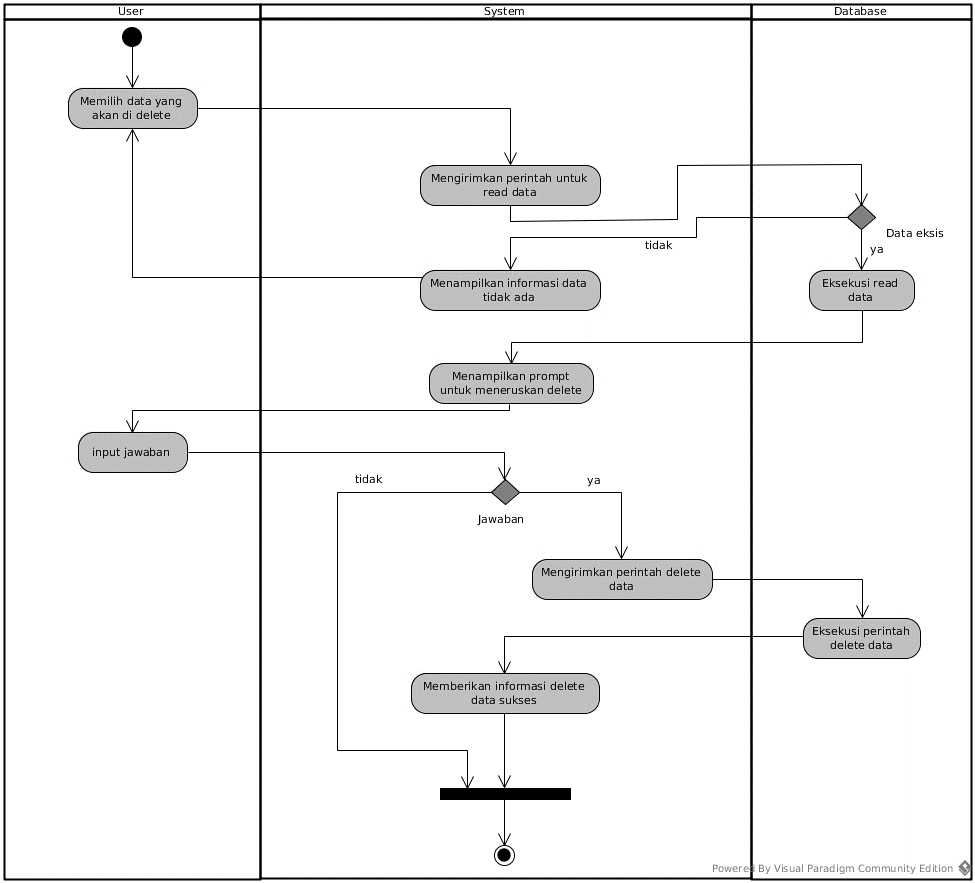
### *Update*

****

Gambar 3.III.6 Activity diagram update

Proses update dimulai dengan pengguna yang memilih data yang akan diupdate. Setelah data dipilih, sistem akan mengirimkan perintah read ke database. Apabila data yang dipilih tidak eksis, maka sistem akan menampilkan informasi data tidak ada kepada pengguna. Sebaliknya, apabila data eksis, maka sistem akan menampilkan form update. Form update ini merupakan form yang sesuai dengan data yang akan di update. Selanjutnya, pengguna akan melakukan input data update dan sistem akan melakukan validasi data tersebut. Apabila data input tersebut tidak valid, maka sistem akan menampilkan informasi data tidak valid. Lalu, pengguna diminta untuk melakukan input kembali. Apabila data valid, maka sistem akan melakukan perintah update data ke database. Sistem akan memberikan informasi update data sukses setelah database berhasil melakukan eksekusi perintah update data.

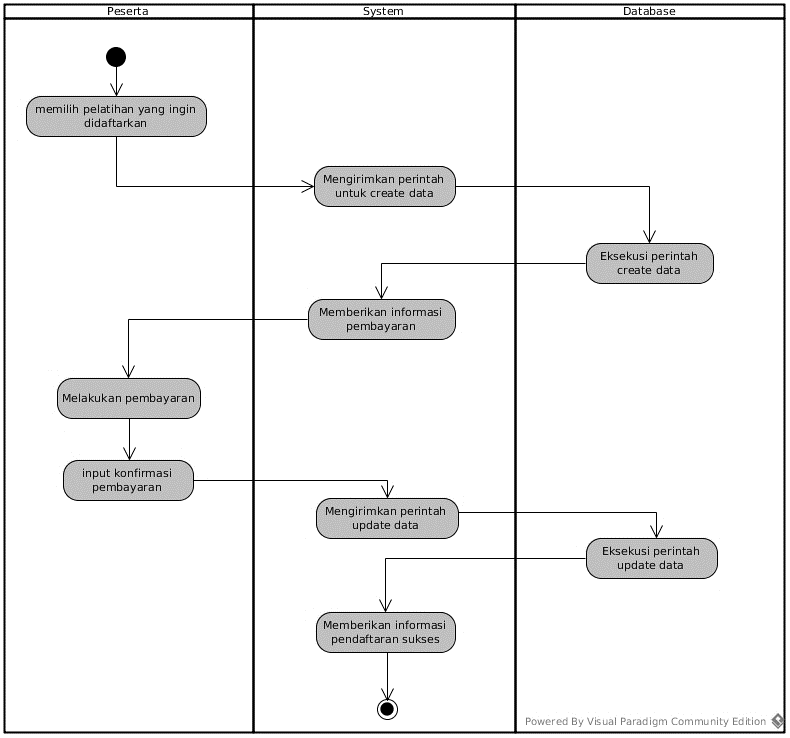
### *Delete*

****

Gambar 3.III.7 Activity diagram delete

Seperti proses update, proses *delete* dimulai dengan pengguna memilih data yang akan di *delete*. Setelah itu, sistem akan mengirimkan perintah untuk melakukan *read* data. Apabila data eksis, maka *database* akan melakukan eksekusi *read* data dan sistem akan menampilkan prompt pertanyaan untuk meneruskan proses *delete*. Sebaliknya, apabila data tidak eksis, maka sistem akan menampilkan informasi bahwa data tidak ada dan pengguna diminta untuk memilih data kembali. Dalam penampilan *prompt*, pengguna akan memilih jawaban berupa “ya” atau “tidak”. Apabila pengguna memilih “ya”, sistem akan mengirimkan perintah *delete* data ke *database*. *Database* akan mengeksekusi perintah tersebut dan memberikan informasi *delete* data sukses kepada pengguna. Sebaliknya apabila pilihan dari pengguna adalah “tidak”, maka sistem akan menghilangkan *prompt* dan proses *delete* selesai. Tidak terjadi *delete* data yang terdapat di *database*.

### Pendaftaran Pelatihan

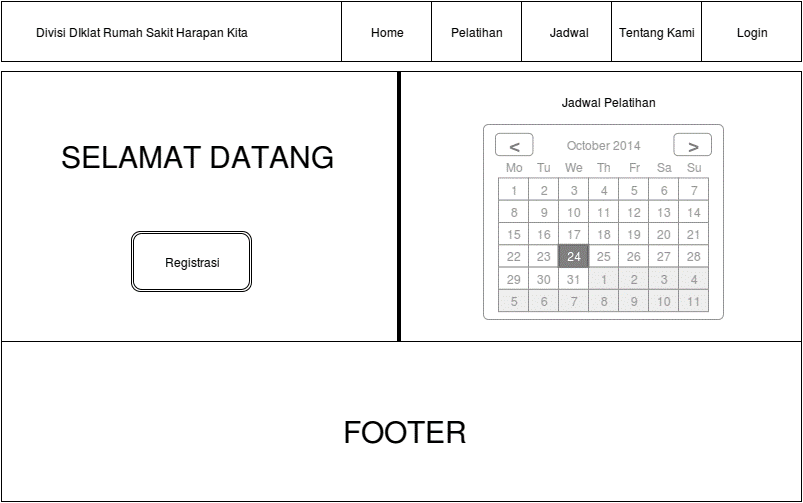


Gambar 3.III.8 Activity diagram pendaftaran pelatihan

Dalam proses pendaftaran pelatihan ini, peserta akan memilih pelatihan yang ingin didaftarkan sesuai dengan jadwal yang ada. Setelah peserta memilih pelatihan yang didaftarkan, sistem akan mengirimkan perintah untuk melakukan *create* data. Data yang akan dibuat ini merupakan data yang terdiri dari id peserta dan id pelatihan yang dipilih. Setelah eksekusi perintah untuk melakukan *create* data berhasil dijalankan, sistem akan memberikan informasi mengenai pembayaran yang harus dilakukan oleh user agar dapat terdaftar pada pelatihan. Informasi yang diberikan akan berisi mengenai jumlah uang yang harus dibayar dan dengan metode apa saja pembayaran bisa dilakukan. Setelah informasi berhasil didapat, peserta akan melakukan pembayaran melalui metode yang diberikan. Untuk melakukan konfirmasi, peserta diharuskan memberikan input data ke sistem. Input data tersebut berisi bukti pembayaran dan metode apa yang digunakan. Selanjutnya sistem akan perintah update data tentang konfirmasi yang dilakukan oleh peserta. Eksekusi perintah *update* data akan berjalan di *database* dan sistem akan memberikan informasi pendaftaran telah sukses dan lunas.

## *Prototyping*

Dalam perancangan, diperlukan sebuah desain awal dari *prototype* sebagai acuan. Berikut adalah rancangan awal interface (*mock-*up) yang dijadikan acuan pembuatan *prototype* sistem.



Gambar 3.III.9 Desain awal *mock-up* sistem

Gambar 3.9 menunjukkan desain awal dari halaman utama sistem. Pada gambar tersebut terdapat bar navigasi yang berisi menu-menu yang berhubungan dengan pelatihan dan jadwal. Menu *login* juga berada pada bar navigasi. Menu tersebut digunakan untuk melakukan *login* oleh pengguna. Untuk menu *Home* sendiri digunakan untuk kembali ke menu utama. Pada bagian *content,* yaitu bagian tengah dari *website*, terbagi menjadi dua. Pada bagian kiri terdapat ucapan selamat datang dan tombol registrasi, sedangkan bagian kanan sistem terdapat jadwal pelatihan yang berbentuk kalendar.Untuk bagian bawah dari *website* terdapat *footer* yang berguna sebagai halaman navigasi dan informasi seperti alamat dan nomor telefon dari Divisi Diklat Harapan Kita.



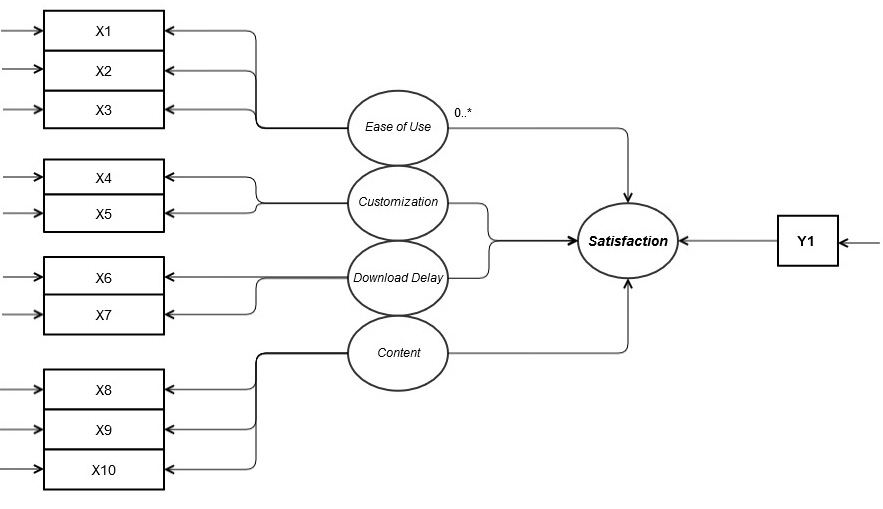
Gambar 3.III.10 Desain awal (*mock-up*) *dashboard* pengguna

Gambar 3.10 menunjukkan rancangan awal dari *dashboard* sistem administrasi. Pada rancangan awal ini *sidebar* yang berisi *link* menu terdapat pada sebelah kiri. Menu yang terdapat pada *sidebar* berbeda-beda sesuai dengan fitur dan hak akses bagi setiap tingkat pengguna. Pada bagian kanan *dashboard* terdapat ruang *content*. Ruang ini akan menampilkan *view* dari *link* yang dipilih pengguna, seperti pendaftaran atau pembuatan, perubahan akun.

Setelah menentukan *planning* dan desain awal dari sistem, dilakukan proses *prototyping.* Proses *protoyping* akan dijelaskan lebih lanjut dalam Bab IV, Implementasi.

## *Feedback*

Pada proses *feedback* akan dilakukan *usability testing* terhadap calon pengguna sistem, yaitu Administrator, Pengajar, dan Peserta. Metode dalam melakukan *usability test* pada rancang bangun sistem administrasi divisi diklat rumah sakit Harapan Kita akan mengacu terhadap teknik yang digunakan oleh Prayoga dan Sensuse dalam “Analisis *Usability* Pada Aplikasi Berbasis *Web* Dengan Mengadopsi Model Kepuasan Pengguna (*User Satisfaction)*” [8] yaitu membandingkan tingkat kepuasan dengan empat konstruk. Pada penelitian ini, *usability test* akan menggunakan *self-reported* *metric data. Metric* sendiri merupakan jalan atau metode untuk mengukur sesuatu atau fenomena [30], sedangkan *self-reported metric data* memberikan informasi tentang persepsi dan perasaan pengguna pada *emotional level* [30].



Gambar 3.III.11 Kerangka berpikir konstruk

Variabel yang dipilih pada penelitian ini berdasarkan model yang dirumuskan oleh Green and Pearson [7]. Variabel tersebut terdiri dari konstruk-konstruk. Tiap konstruk mempunyai sebuah indikator dengan kode yang berbeda. Tabel 2 memberikan gambaran mengenai definisi konstruk variabel penelitian. Gambar 6 menunjukkan kerangka berpikir dari variabel *usability test* sistem administrasi divisi diklat rumah sakit Harapan Kita.

*(sumber : Prayoga & Sensuse, 2010)*

Tabel 3.22 Konstruk *usability test*

|  |  |  |
| --- | --- | --- |
| Konstruk | Indikator | Kode |
| *Ease Of Use* | Struktur Penyajian | X1 |
| Kemudahan akses | X2 |
| Kejelasan penyajian informasi | X3 |
| *Customization* | Materi yang menarik | X4 |
| Personalisasi | X5 |
| *Download Delay* | Kecepatan menemukan informasi | X6 |
| Kontrol terhadap materi | X7 |
| *Content* | Spesifikasi informasi | X8 |
| Pemenuhan Kebutuhan | X9 |
| Kecukupan Materi | X10 |
| *Satisfaction* | Kepuasan | Y1 |

Untuk mengumpulkan data yang berasal dari definisi konstruk tersebut, dibutuhkan sebuah instrumen utama, yaitu kuisioner. Pengukuran variabel dilakukan dengan menggunakan skala Likert [8], yaitu skala yang digunakan untuk mengukur sikap, pendapat dan persepsi seseorang dengan memberikan skor kepada masing-masing alternatif jawaban. Responden dari sampel diminta untuk menyatakan persetujuannya atas dasar persepsi masing-masing. Jawaban terdiri dari enam pilihan, yaitu: Sangat Sangat Tidak Setuju (SSTS), Sangat Tidak Setuju (STS), Tidak Setuju (TS), Setuju (S), Sangat Setuju (SS), dan Sangat Sangat Setuju (SSS).Pemberian nilai (*scoring)* dilakukan dengan cara menurun. Nilai 6 (enam) untuk Sangat Sangat Setuju (SSS) dan seterusnya menurun sampai pada Sangat Sangat Tidak Setuju (SSTS) yang bernilai 1 (satu).

Tabel 3.23 Skala Likert yang dilakukan pada *usability test*

|  |  |  |
| --- | --- | --- |
| Jawaban | Singkatan | Nilai |
| Sangat Sangat Tidak Setuju | SSTS | 1 |
| Sangat Tidak Setuju | STS | 2 |
| Tidak Setuju | TS | 3 |
| Setuju | S | 4 |
| Sangat Setuju | SS | 5 |
| Sangat Sangat Setuju | SSS | 6 |

Setelah mendapatkan hasil data dari kuisioner, penelitian akan dilanjutkan dengan melakukan pengujian hipotesis. Pengujian hipotesis ini bertujuan untuk menganalisis hubungan dan pengaruh (sebab-akibat) dari dua atau lebih fenomena [31]. Hipotesis yang diajukan adalah:

1. Hipotesis pertama (H1): *Ease of use* pada *prototype* sistem memiliki pengaruh positif langsung terhadap *satisfaction* responden.
2. Hipotesis kedua (H2): *Customization* pada *prototype* sistem memiliki pengaruh positif langsung terhadap *satisfaction* responden.
3. Hipotesis ketiga (H3): *Download delay* pada *prototype* sistem memiliki pengaruh positif langsung terhadap *satisfaction* responden.
4. Hipotesis keempat (H4): *Content* pada *prototype* sistem memberikan pengaruh positif langsung terhadap *satisfaction* responden.

# IMPLEMENTASI

## Desain

Perancangan sangat dibutuhkan dalam pengembangan sebuah sistem. Tujuan dari perancangan sendiri adalah sebagai acuan dalam pembuatan sistem pada saat proses *prototyping*. Hasil perancangan dijelaskan dalam desain database, *class diagram, deployment diagram,* dan *sequence diagram*. Adapun hasil perancangan sistem adalah sebagai berikut.

### Desain Database

Perancangan *database* dilakukan berdasarkan kebutuhan yang telah dianalisis pada proses *communication*. Rancangan *database* ini berupa *entity, attribute,* dan relasi. Berdasarkan hasil dari analisa dan *requirement gathering* yang dilakukan, penulis memperoleh beberapa *entity* yang akan digunakan dalam membangun aplikasi. *Entity* yang diperoleh adalah *user, peserta, pengajar, pelatihan, statuspelatihan, pelatihanentry, absen,* dan *pembayaran.* Semua *entity* yang diperoleh juga mewakili nama tabel *database* yang dibuat penulis.Daftar *entity* beserta keterangan untuk tiap *entity* dapat dilihat pada tabel IV.1.

Tabel 4.1 Daftar *entity* dan keterangan *database*

|  |  |  |
| --- | --- | --- |
| **No.** | ***Entity*** | **Keterangan** |
| 1. | User | *Entity* yang memiliki hak tertinggi untuk mengakses semua informasi dan fitur yang berhubungan dengan fungsi administrasi pada sistem administrasi. |
| 2. | Peserta | *Entity* yang bertindak sebagai peserta dan memiliki hak untuk mengakses informasi yang berhubungan dengan dirinya. |
| 3. | Pengajar | *Entity* yang bertindak sebagai pengajar dan memiliki hak untuk mengakses informasi yang berhubungan dengan dirinya dan kepelatihan. |
| 4. | Pelatihan | *Entity* yang berisi data dari tipe pelatihan yang diadakan oleh Divisi diklat Rumah Sakit Harapan Kita. |
| 5. | Statuspelatihan | *Entity* yang berisi data dari pelatihan yang diselenggarakan dalam kurun waktu tertentu. Data pelatihan diambil dari entity pelatihan. |
| 6. | Pelatihanentry | *Entity* yang berisi data pelatihan yang diikuti oleh peserta. Data entity ini diambil dari statuspelatihan dan peserta. |
| 7. | Absen | *Entity* yang berisi data absen dari peserta yang mengikuti pelatihan. Absen berdasarkan entity pelatihanentry. |
| 8. | Pembayaran | *Entity* yang berisi data pembayaran yang dibayarkan oleh peserta. |
| 9. | Nilai | *Entity* yang berisi data nilai bagi peserta. |
| 10. | Instansi | *Entity* yang berisi data instansi yang terkait oleh peserta/calon peserta. |
| 11. | Institusi | *Entity* yang berisi data institusi yang terkait oleh peserta/calon peserta. |

*Entity* tersebut akan mewakili informasi yang terdapat di sistem administrasi Divisi Diklat Rumah Sakit Harapan Kita. Relasi dari masing-masing *entity* adalah sebagai berikut:

User : Tidak mempunyai relasi dengan tabel lain.

Peserta : *One-to-many* dengan pelatihanentry.

Pengajar : *One-to-many* dengan statuspelatihan.

Pelatihan : *One-to-many* dengan statuspelatihan.

Statuspelatihan : *One-to-many* dengan pelatihanentry;

*Many-to-one* dengan pelatihan.

Pelatihanentry : *Many-to-one* dengan statuspelatihan;

*Many-to-one* dengan peserta;

*One-to-one* dengan pembayaran;

*One-to-one* dengan nilai;

*One-to-many* dengan absen.

Absen : *Many-to-one* dengan pelatihanentry.

Pembayaran : *One-to-one* dengan pelatihanentry.

Nilai : *One-to-many* dengan pelatihanentry.

Institusi : *One-to-many* dengan peserta.

Instansi : *One-to-many* dengan peserta.

Berikut adalah *conceptual database design* dari *database* yang dibangun berdasarkan *relationship* yang telah ditentukan.

C:\Users\niko\Downloads\diagram konseptual.png

Gambar 4.IV.1 *Conceptual database design* dari *prototype* sistem

Setiap *entity* mempunyai beberapa *attribute* yang berguna untuk menyimpan segala macam informasi yang menaunginya. *Attribute* untuk setiap *entity* dapat dilihat pada tabel IV.2

Tabel 4.2 Daftar *attribute* masing-masing *entity*

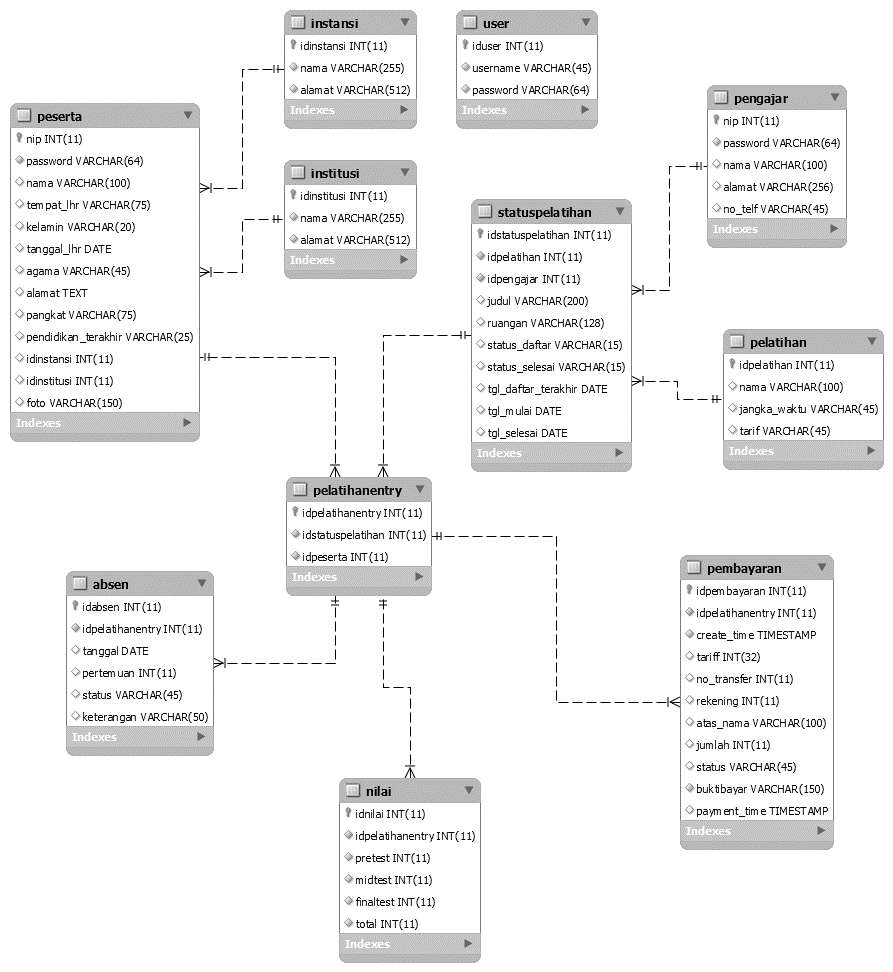
|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Entity** | **Attribute** | **Keterangan** |
| 1. | *User* | *iduser* | *Primary Key* pada *entity user* |
|  |  | *username* | Nama untuk identifikasi dan autentikasi user. |
|  |  | *password* | Kata sandi yang digunakan user untuk melakukan autentikasi pada sistem. |
|  |  | *foto* | *Attribute* yang berisi direktori gambar yang digunakan sebagai identifikasi pengguna. |
| 2. | *Peserta* | *nip* | Nomor induk dari peserta yang juga digunakan sebagai *primary key* |
|  |  | *password* | Kata sandi yang digunakan peserta untuk melakukan autentikasi pada sistem. |
|  |  | *nama* | Nama untuk identifikasi dan autentikasi peserta. |
|  |  | *tempat\_lhr* | Informasi tempat lahir peserta. |
|  |  | *kelamin* | Informasi kelamin (gender) peserta. |
|  |  | *tanggal\_lhr* | Informasi tanggal lahir peserta. |
|  |  | *agama* | Informasi agama peserta. |
|  |  | *alamat* | Informasi alamat peserta. |
|  |  | *pangkat* | Informasi pangkat pekerjaan peserta. |
|  |  | *pendidikan\_terakhir* | Informasi pendidikan terakhir yang dijalankan peserta. |
|  |  | *idinstansi* | Informasi institusi dimana peserta belajar. *Attribute* ini merupakan *foreign key* yang berasal dari tabel instansi. |
|  |  | *idinstitusi* | Informasi institusi dimana peserta belajar. *Attribute* ini merupakan *foreign key* yang berasal dari tabel institusi. |
|  |  | *foto* | *Attribute* yang berisi direktori gambar yang digunakan sebagai identifikasi peserta. |
| 3. | *pengajar* | *nip* | Nomor induk dari pengajar yang juga digunakan sebagai *primary key.* |
|  |  | *password* | Kata sandi yang digunakan pengajar untuk melakukan autentikasi pada sistem. |
|  |  | *nama* | Nama untuk identifikasi dan autentikasi pengajar. |
|  |  | *alamat* | Informasi alamat pengajar. |
|  |  | *no\_telf* | Informasi nomor telefon yang digunakan pengajar. |
|  |  | *foto* | *Attribute* yang berisi direktori gambar yang digunakan sebagai identifikasi pengajar. |
| 4. | *pelatihan* | *idpelatihan* | *Primary key* pada *entity pelatihan.* |
|  |  | *nama* | Nama untuk identifikasi tipe pelatihan yang diadakan oleh Divisi diklat Rumah Sakit Harapan Kita |
|  |  | *jangka\_waktu* | Informasi mengenai berapa lama waktu pengajaran dari tipe pelatihan. |
|  |  | *tarif* | Informasi mengenai berapa biaya yang diperlukan untuk melakukan pendaftaran pelatihan. |
| 5. | *statuspelatihan* | *idstatuspelatihan* | *Primary key* pada *entity statuspelatihan.* |
|  |  | *idpelatihan* | *Foreign key* yang berasal dari *entity pelatihan.* |
|  |  | *idpengajar* | *Foreign key* yang berasal dari *entity pengajar.* |
|  |  | *judul* | Nama untuk identifikasi judul pelatihan yang berjalan. |
|  |  | *ruangan* | Informasi dimana ruangan pelatihan diadakan. |
|  |  | *status\_daftar* | Informasi status pendaftaran dari pelatihan yang berjalan. |
|  |  | *status\_selesai* | Informasi status pengajaran dari pelatihan yang berjalan. |
|  |  | *tgl\_daftar\_terakhir* | Informasi tanggal dimana pendaftaran pelatihan berakhir. |
|  |  | *tgl\_mulai* | Informasi tanggal mulai pelatihan yang berjalan. |
|  |  | *tgl\_selesai* | Informasi tanggal selesai pelatihan yang berjalan. |
| 6. | *pelatihanentry* | *idpelatihanentry* | *Primary key* pada *entity pelatihanentry.* |
|  |  | *idstatuspelatihan* | *Foreign key* yang berasal dari *entity statuspelatihan.* |
|  |  | *idpeserta* | *Foreign key* yang berasal dari *entity peserta* |
| 7. | *absen* | *idabsen* | *Primary key* dari *entity absen* |
|  |  | *idpelatihanentry* | *Foreign key* dari *entity pelatihanentry* |
|  |  | *tanggal* | Informasi tanggal absen dari peserta |
|  |  | *pertemuan* | Informasi pertemuan keberapa dari absen. |
|  |  | *status* | Informasi status absen. |
|  |  | *keterangan* | Informasi keterangan absen. |
| 8. | *pembayaran* | *idpembayaran* | *Primary key* dari *entity pembayaran.* |
|  |  | *idpelatihanentry* | *Foreign key* dari *entity pelatihanentry.* |
|  |  | *create\_time* | Informasi waktu pendaftaran dan pembayaran. |
|  |  | *tariff* | Informasi tarif yang dibutuhkan untuk membayarkan biaya pendaftaran. |
|  |  | *no\_transfer* | Informasi nomor transfer yang peserta bayarkan. |
|  |  | *rekening* | Informasi rekening yang peserta bayarkan. |
|  |  | *atas\_nama* | Informasi nama pemilik rekening yang peserta bayarkan |
|  |  | *jumlah* | Informasi jumlah uang yang dibayarkan peserta. |
|  |  | *status* | Informasi status pembayaran (Lunas/belum lunas). |
|  |  | *buktibayar* | *Attribute* yang berisi direktori gambar yang digunakan sebagai identifikasi bukti pembayaran. |
|  |  | *payment\_time* | Informasi waktu pembayaran. |
| 9. | *nilai* | *idnilai* | *Primary key* dari *entity* nilai. |
|  |  | *idpelatihanentry* | *Foreign key* dari tabel pelatihanentry. |
|  |  | *pretest* | Berisi informasi mengenai nilai *pretest* peserta, yaitu nilai tes yang diadakan sebelum pelatihan berlangsung. |
|  |  | *midtest* | Berisi informasi mengenai nilai *midtest* peserta, yaitu nilai tes yang diadakan di tengah waktu pelatihan berlangsung |
|  |  | *finaltest* | Berisi informasi mengenai nilai *final test* peserta, yaitu nilai tes yang diadakan saat pelatihan akan berakhir. |
|  |  | *total* | Merupakan rata-rata jumlah dari nilai *midtest* dan *final test* peserta. *Attribute* ini menunjukkan nilai total yang didapat oleh peserta. |
| 10. | *Instansi* | *idinstansi* | *Primary key* dari *entity* instansi*.* |
|  |  | *nama* | *Attribute* yang berisi nama dari instansi. |
|  |  | *alamat* | *Attribute* yang berisi alamat dari instansi. |
| 11. | Institusi | *idinstitusi* | *Primary key* dari *entity* institusi*.* |
|  |  | *nama* | *Attribute* yang berisi nama dari institusi. |
|  |  | *alamat* | *Attribute* yang berisi alamat dari institusi. |

Berikut adalah *logical database design* berdasarkan *entity* dan *relationship yang* telah ditentukan sebelumnya. Pada skema ini telah ditentukan *primary, foreign key,* dan atribut pada masing-masing *entity*.



Gambar 4.IV.2 *Logical database design* dari prototype sistem

Berikut adalah gambaran *physical database design* yang telah lengkap dengan *primary key, foreign key,* atribut dan tipe data yang digunakan. Skema ini merupakan desain aktual yang digunakan dalam pembuatan *prototype* sistem.

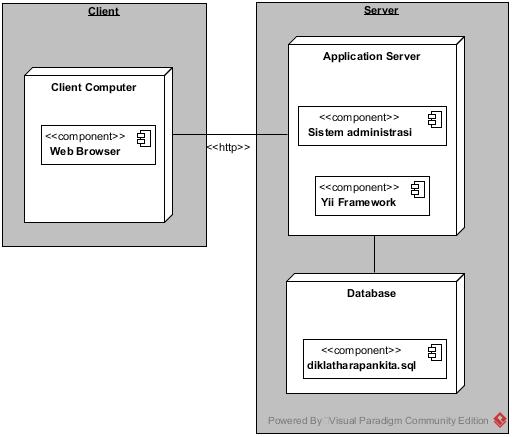


Gambar 4.IV.3 Diagram *entity relationship database prototype sistem*

Dalam ERD sistem administrasi divisi diklat rumah sakit harapan kita terdapat tiga tabel yang mewakili kelas pengguna dalam aplikasi. Ketiga tabel tersebut adalah tabel user, pengajar, dan peserta. Tabel pengajar dan peserta saling terhubung dengan tabel yang lain. Tabel peserta terhubung dengan tabel pelatihanentry, institusi dan instansi. Sedangkan tabel pengajar terhubung dengan tabel statuspelatihan. Tabel pelatihanentry sendiri merupakan sebuah tabel yang berisikan data entri pelatihan yang diikuti peserta, berisikan id dari tabel statuspelatihan dan id dari peserta. Statuspelatihan merupakan tabel yang berisi data pelatihan yang sedang berlangsung. Data pelatihan dari tabel statuspelatihan diambil dari tabel pelatihan. Dalam sistem administrasi ini, diperlukan juga tabel yang berisi data absen dan nilai dari setiap peserta yang mengikuti pelatihan. Oleh karena itu tabel absen dan nilai juga terhubung oleh tabel pelatihanentry. Dalam *database* ini juga terdapat tabel pembayaran Tabel pembayaran juga terhubung dengan tabel pelatihanentry.

### *Deployment diagram*

*Deployment diagram* merupakan sebuah diagram yang menggambarkan topologi komponen sistem ketika dilakukan implementasi.

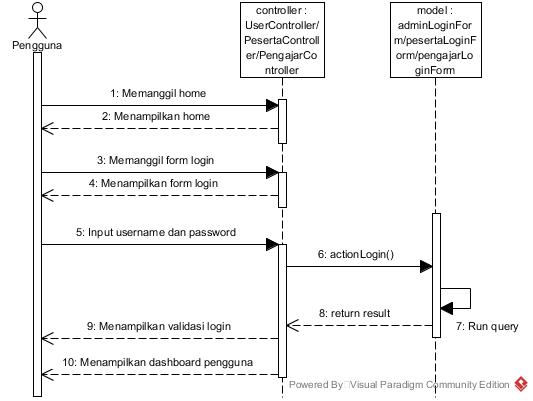


Gambar 4.IV.4 *Deployment diagram* sistem administrasi

Gambar 4.4 menunjukkan bahwa topologi terdiri dari dua bagian, *client* dan *server*. *Client* berisi komponen *web browser* yang terhubung ke *server* melalui http, sedangkan *server* berisi komponen sistem administrasi dan Yii *framework* yang terhubung dengan *database server*.

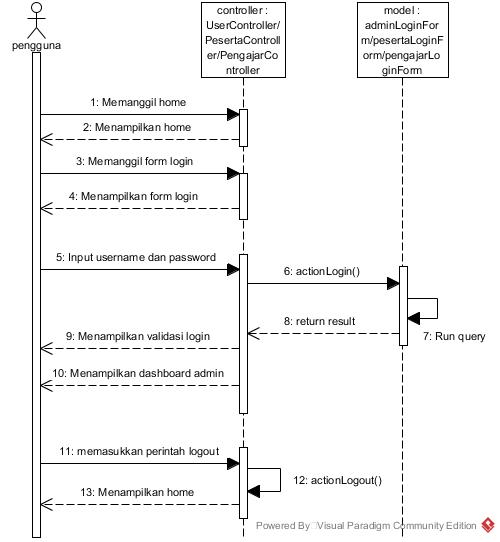
### Sequence Diagram

Berikut adalah *sequence diagram* dari *prototype* sistem yang dibuat.



Gambar 4.IV.5 *Sequence diagram login* pengguna

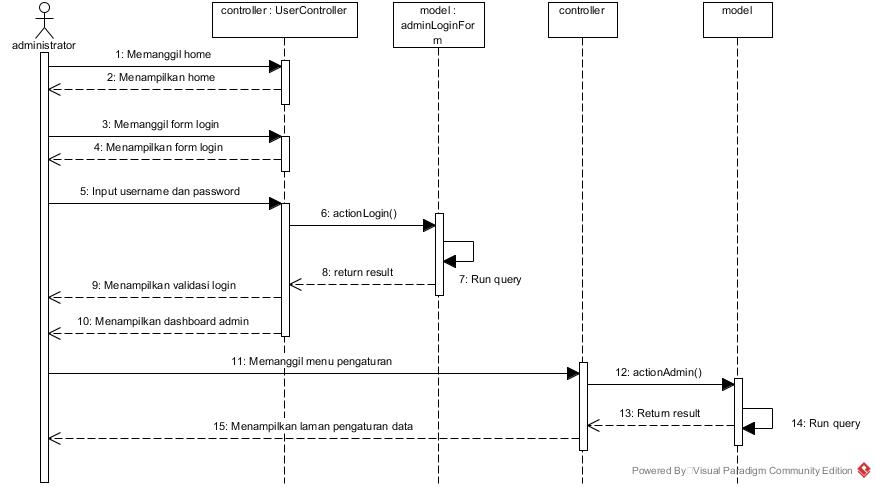
Gambar di atas merupakan *sequence diagram* dari *login* pengguna. Proses ini menggunakan *controller* UserController.php atau PesertaController.php atau PengajarController.php sebagai logika kontrol dan adminLoginForm.php atau pesertaLoginForm atau PengajarLoginForm untuk melakukan *query*.



Gambar 4.IV.6 *Sequence diagram logout* pengguna

Gambar di atas merupakan *sequence diagram* dari *logout* pengguna. Proses ini menggunakan *controller* UserController.php atau PesertaController.php atau PengajarController.php sebagai logika kontrol dan adminLoginForm.php atau pesertaLoginForm atau PengajarLoginForm untuk melakukan *query*.

Pada dasarnya, pengelolaan aku admin, peserta, pengajar, pelatihan, jadwal pelatihan, absen, nilai, dan pembayaran pada dasarnya merupakan proses CRUD (Create, Read, Update,Delete). Jadi, untuk *sequence diagram* pengelolaan oleh administrator hanya akan ditampilkan secara general dengan detail *controller* dan *model* apa saja yang digunakan oleh masing-masing kategori.

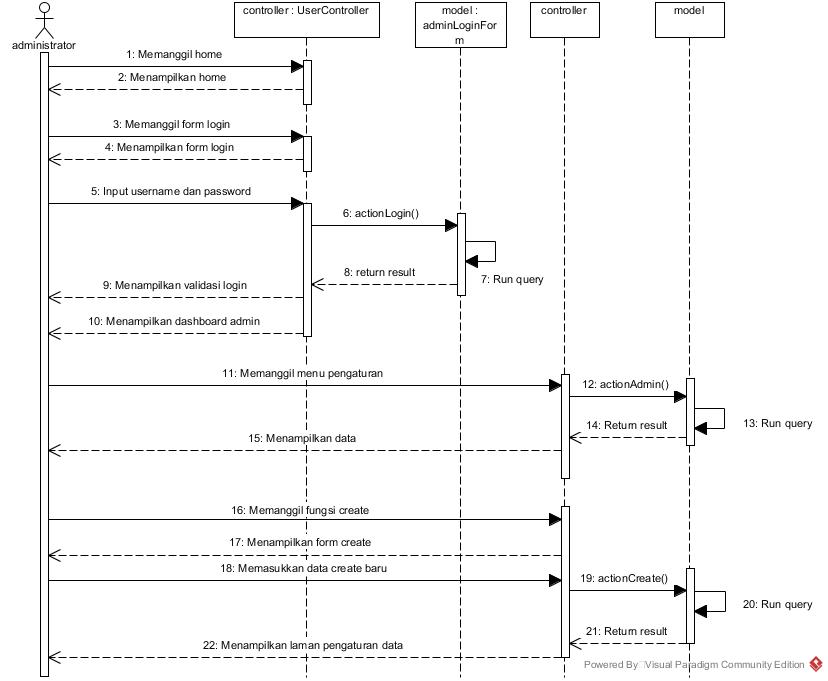


Gambar 4.IV.7 *Sequence diagram* pengaturan

Gambar di atas merupakan *sequence diagram* dari pengaturan akun administrator. Proses ini menggunakan *controller* controllers/UserController.php sebagai logika kontrol dan model/User.php untuk melakukan *query*. *Sequence diagram* ini akan memanggil laman pengaturan data sebagai hasil dari proses. Berikut adalah tabel model dan action untuk kategori yang berbeda.

Tabel 4.3 Tabel model dan action pengaturan

|  |  |  |
| --- | --- | --- |
| Model | Controller | Function |
| Models/User.php | Controllers/UserController.php | actionAdmin() |
| Models/Peserta.php | Controllers/PesertaController.php | actionAdmin() |
| Models/Pengajar.php | Controllers/PengajarController.php | actionAdmin() |
| Models/Pelatihan.php | Controllers/PelatihanController.php | actionAdmin() |
| Models/Pelatihan.php | Controllers/PelatihanController.php | actionAdmin() |
| Models/Pelatihanentry.php | Controllers/PelatihanEntryController.php | actionAdmin() |
| Models/Statuspelatihan.php | Controller/StatusPelatihanController.php | actionAdmin() |
| Models/Absen.php | Controller/AbsenController.php | actionAdmin() |
| Models/Nilai.php | Controller/NilaiController.php | actionAdmin() |
| Models/Pembayaran.php | Controller/Pembayaran.php | actionAdmin() |

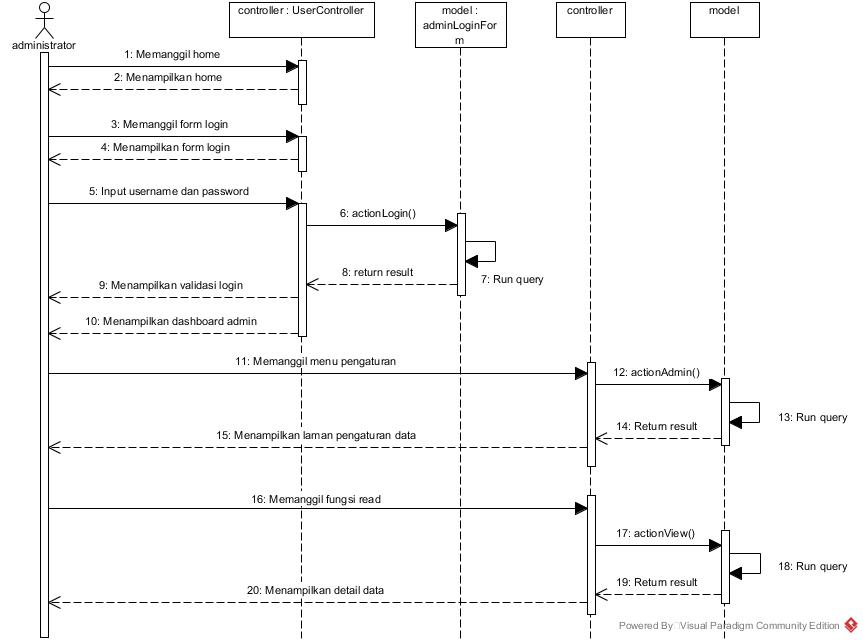


Gambar 4.IV.8 *Sequence diagram create* akun administrator

Gambar di atas merupakan *sequence diagram* dari *create* akun administrator. Proses ini menggunakan *controller* controllers/UserController.php sebagai logika kontrol dan model/User.php untuk melakukan *query*. *Sequence diagram* ini akan memanggil laman pembuatan data baru sebagai hasil dari proses. Berikut adalah tabel model dan action untuk kategori yang berbeda.

Tabel 4.4 Tabel model dan action create

|  |  |  |
| --- | --- | --- |
| Model | Controller | Function |
| Models/User.php | Controllers/UserController.php | actionCreate() |
| Models/Peserta.php | Controllers/PesertaController.php | actionCreate() |
| Models/Pengajar.php | Controllers/PengajarController.php | actionCreate() |
| Models/Pelatihan.php | Controllers/PelatihanController.php | actionCreate() |
| Models/Pelatihan.php | Controllers/PelatihanController.php | actionCreate() |
| Models/Pelatihanentry.php | Controllers/PelatihanEntryController.php | actionCreate() |
| Models/Statuspelatihan.php | Controller/StatusPelatihanController.php | actionCreate() |
| Models/Absen.php | Controller/AbsenController.php | actionCreate() |
| Models/Nilai.php | Controller/NilaiController.php | actionCreate() |
| Models/Pembayaran.php | Controller/Pembayaran.php | actionCreate() |

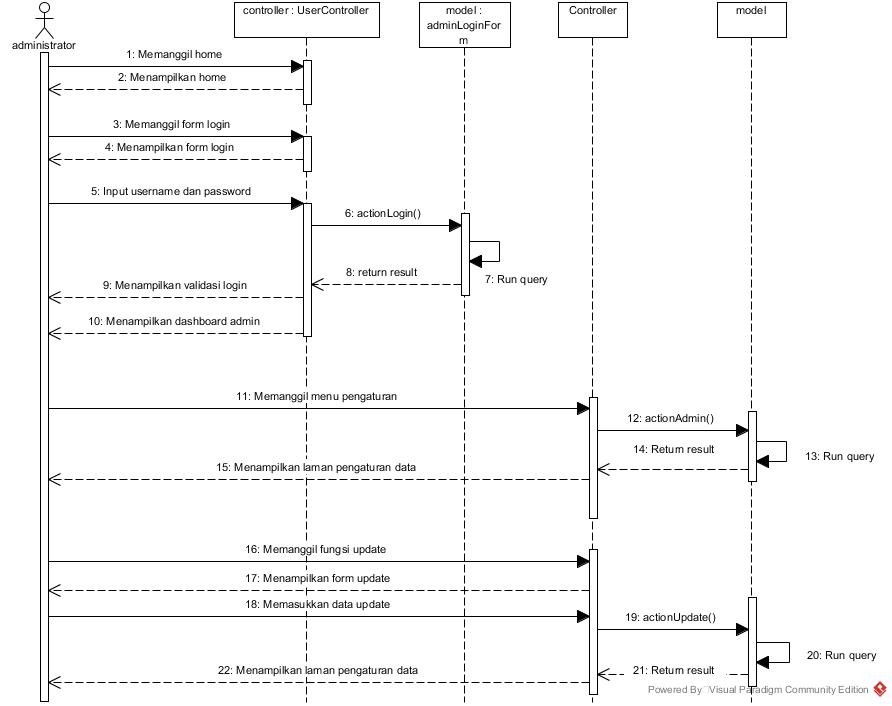


Gambar 4.IV.9 *Sequence diagram read* akun administrator

Gambar di atas merupakan *sequence diagram* dari *read* akun administrator. Proses ini menggunakan *controller* controllers/UserController.php sebagai logika kontrol dan model/User.php untuk melakukan *query*. *Sequence diagram* ini akan memanggil laman detail data sebagai hasil dari proses. Berikut adalah tabel model dan action untuk kategori yang berbeda.

Tabel 4.5 Tabel model dan function read

|  |  |  |
| --- | --- | --- |
| Model | Controller | Function |
| Models/User.php | Controllers/UserController.php | actionView() |
| Models/Peserta.php | Controllers/PesertaController.php | actionView() |
| Models/Pengajar.php | Controllers/PengajarController.php | actionView() |
| Models/Pelatihan.php | Controllers/PelatihanController.php | actionView() |
| Models/Pelatihan.php | Controllers/PelatihanController.php | actionView() |
| Models/Pelatihanentry.php | Controllers/PelatihanEntryController.php | actionView() |
| Models/Statuspelatihan.php | Controller/StatusPelatihanController.php | actionView() |
| Models/Absen.php | Controller/AbsenController.php | actionView() |
| Models/Nilai.php | Controller/NilaiController.php | actionView() |
| Models/Pembayaran.php | Controller/Pembayaran.php | actionView() |

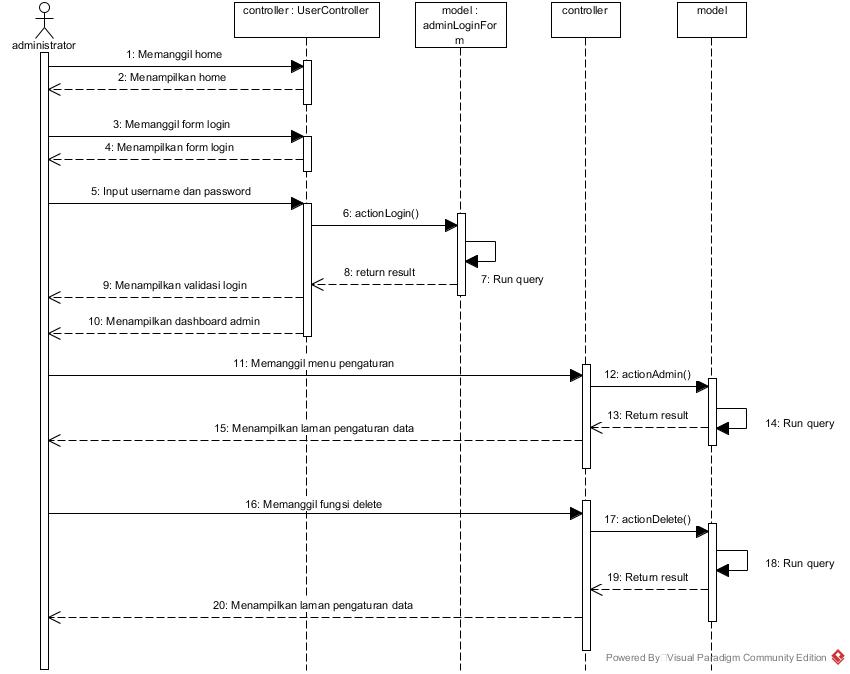


Gambar 4.IV.10 *Sequence diagram update* akun administrator

Gambar di atas merupakan *sequence diagram* dari *update*/pengubahan akun administrator. Proses ini menggunakan *controller* controllers/UserController.php sebagai logika kontrol dan model/User.php untuk melakukan *query*. *Sequence diagram* ini akan memanggil laman pengaturan data sebagai hasil dari proses. Berikut adalah tabel model dan action untuk kategori yang berbeda.

Tabel 4.6 Tabel model dan function update

|  |  |  |
| --- | --- | --- |
| Model | Controller | Function |
| Models/User.php | Controllers/UserController.php | actionUpdate() |
| Models/Peserta.php | Controllers/PesertaController.php | actionUpdate() |
| Models/Pengajar.php | Controllers/PengajarController.php | actionUpdate() |
| Models/Pelatihan.php | Controllers/PelatihanController.php | actionUpdate() |
| Models/Pelatihan.php | Controllers/PelatihanController.php | actionUpdate() |
| Models/Pelatihanentry.php | Controllers/PelatihanEntryController.php | actionUpdate() |
| Models/Statuspelatihan.php | Controller/StatusPelatihanController.php | actionUpdate() |
| Models/Absen.php | Controller/AbsenController.php | actionUpdate() |
| Models/Nilai.php | Controller/NilaiController.php | actionUpdate() |
| Models/Pembayaran.php | Controller/Pembayaran.php | actionUpdate() |

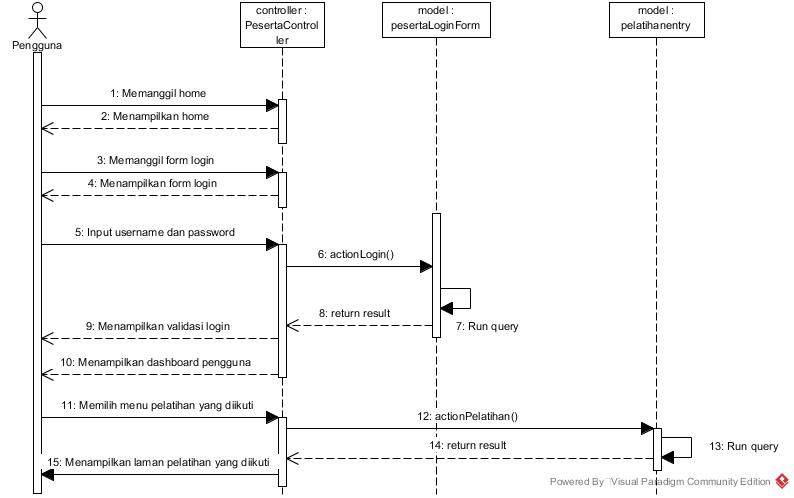


Gambar 4.IV.11 *Sequence* diagram *delete* akun administrator

Gambar di atas merupakan *sequence diagram* dari *delete*  akun administrator. Proses ini menggunakan *controller* controllers/UserController.php sebagai logika kontrol dan model/User.php untuk melakukan *query*. *Sequence diagram* ini akan memanggil laman pengaturan data sebagai hasil dari proses. Berikut adalah tabel model dan action untuk kategori yang berbeda.

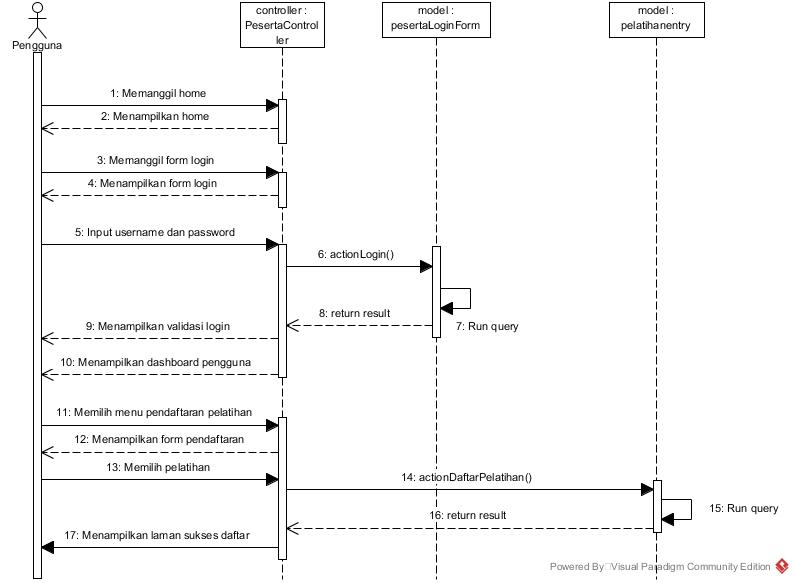
Tabel 47 Tabel model dan function delete

|  |  |  |
| --- | --- | --- |
| Model | Controller | Function |
| Models/User.php | Controllers/UserController.php | actionDelete() |
| Models/Peserta.php | Controllers/PesertaController.php | actionDelete() |
| Models/Pengajar.php | Controllers/PengajarController.php | actionDelete() |
| Models/Pelatihan.php | Controllers/PelatihanController.php | actionDelete() |
| Models/Pelatihan.php | Controllers/PelatihanController.php | actionDelete() |
| Models/Pelatihanentry.php | Controllers/PelatihanEntryController.php | actionDelete() |
| Models/Statuspelatihan.php | Controller/StatusPelatihanController.php | actionDelete() |
| Models/Absen.php | Controller/AbsenController.php | actionDelete() |
| Models/Nilai.php | Controller/NilaiController.php | actionDelete() |
| Models/Pembayaran.php | Controller/Pembayaran.php | actionDelete() |



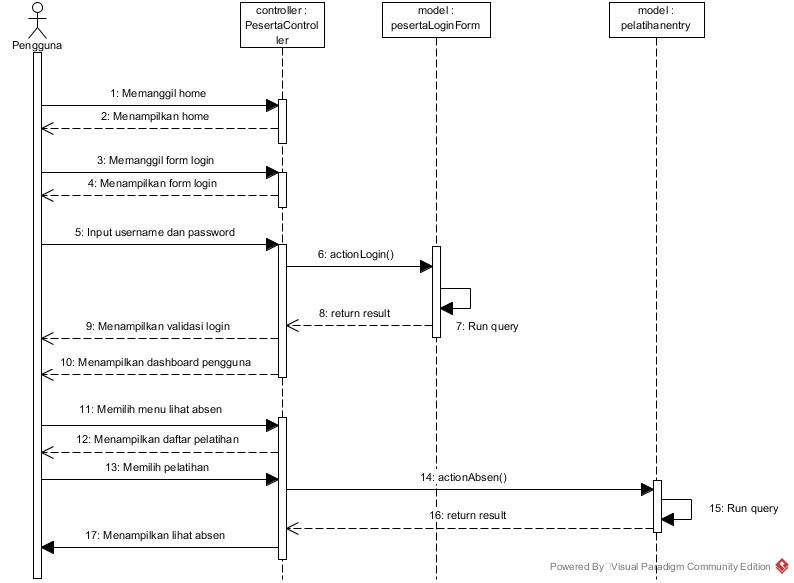
Gambar 4.IV.12 *Sequence diagram* lihat pelatihan yang diikuti

Gambar di atas merupakan *sequence diagram* dari lihat pelatihan yang diikuti. Proses ini menggunakan *controller* controllers/PesertaController.php sebagai logika kontrol dan model/Pelatihanentry.php untuk melakukan *query*. *Sequence diagram* ini akan memanggil laman pelatihan yang diikuti sebagai hasil proses.



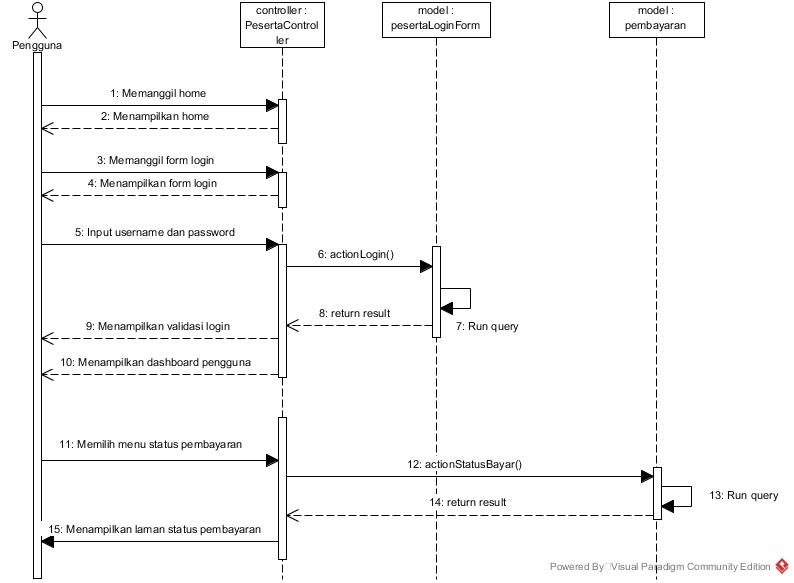
Gambar 4.IV.13 *Sequence diagram* pendaftaran pelatihan

Gambar di atas merupakan *sequence diagram* dari pendaftaran pelatihan yang akan diikuti. Proses ini menggunakan *controller* controllers/PesertaController.php sebagai logika kontrol dan model/Pelatihanentry.php untuk melakukan *query*. *Sequence diagram* ini akan memanggil laman sukses daftar sebagai hasil proses.



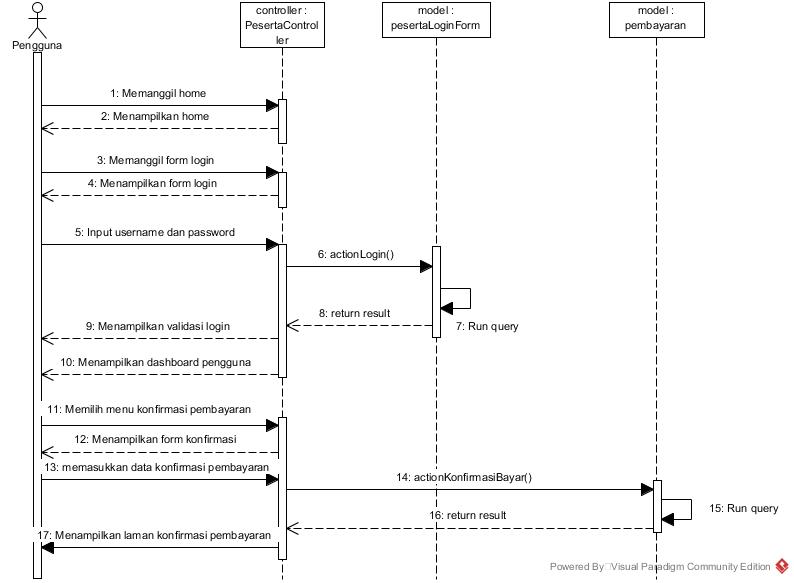
Gambar 4.IV.14 *Sequence diagram* lihat absen

Gambar di atas merupakan *sequence diagram* untuk melihat absensi peserta dari pelatihan yang akan diikuti. Proses ini menggunakan *controller* controllers/PesertaController.php sebagai logika kontrol dan model/Pelatihanentry.php untuk melakukan *query*. *Sequence diagram* ini akan memanggil laman lihat absensi sebagai hasil proses.



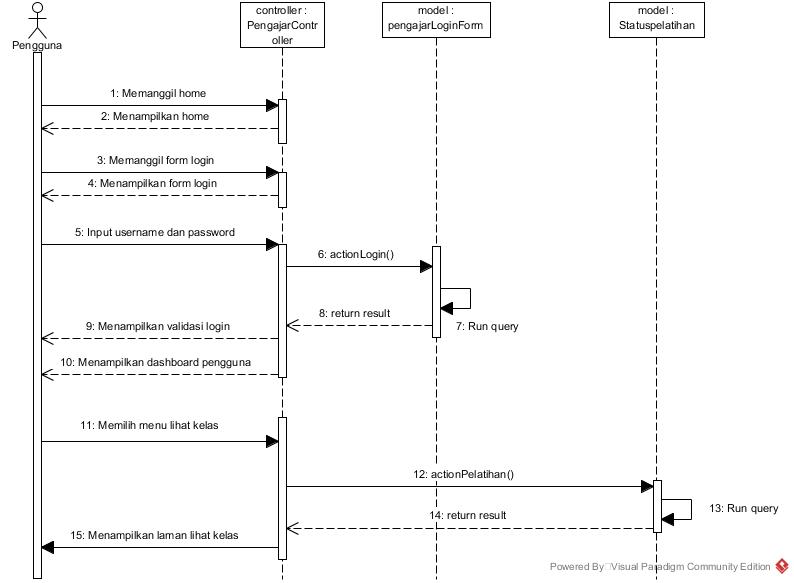
Gambar 4.IV.15 *Sequence diagram* lihat status pembayatan

Gambar di atas merupakan *sequence diagram* untuk melihat absensi peserta dari pelatihan yang akan diikuti. Proses ini menggunakan *controller* controllers/PesertaController.php sebagai logika kontrol dan model/Pembayaran.php untuk melakukan *query*. *Sequence diagram* ini akan memanggil laman lihat absensi sebagai hasil proses.

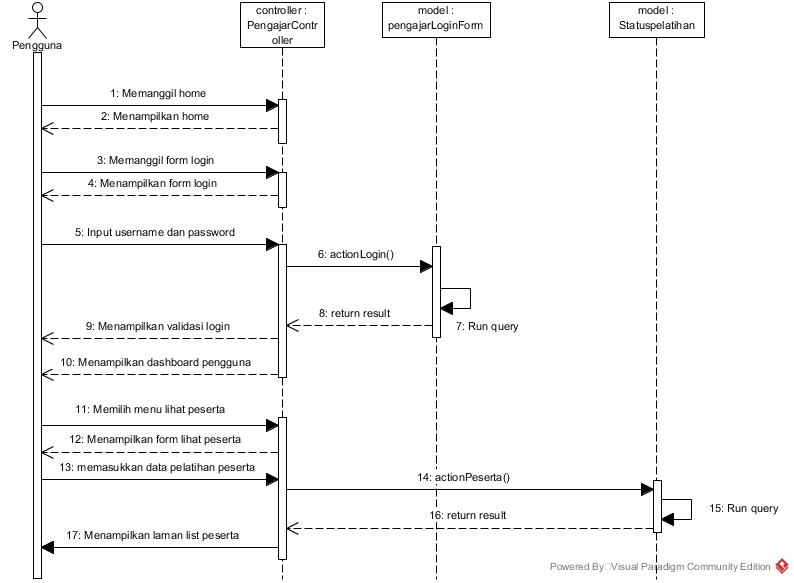


Gambar 4.IV.16 *Sequence diagram* konfirmasi pembayaran

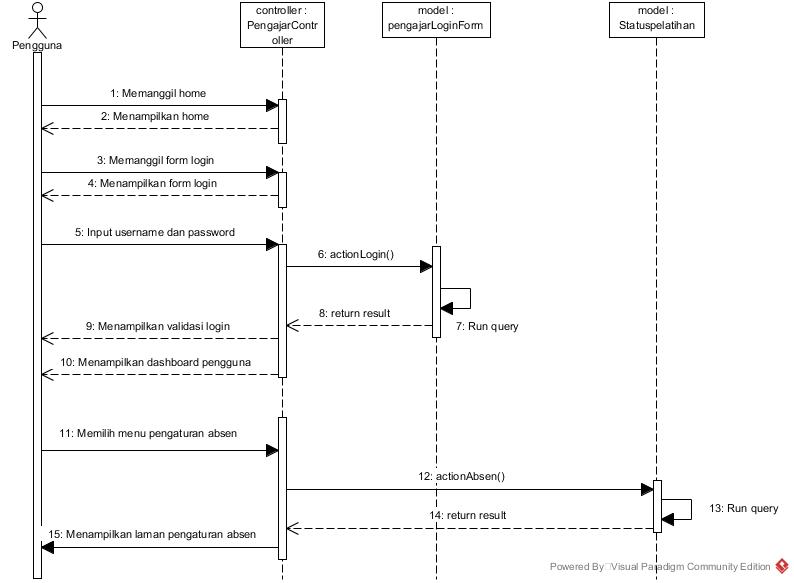
Gambar di atas merupakan *sequence diagram* untuk melihat absensi peserta dari pelatihan yang akan diikuti. Proses ini menggunakan *controller* controllers/PesertaController.php sebagai logika kontrol dan model/Pembayaran.php untuk melakukan *query*. *Sequence diagram* ini akan memanggil laman konfirmasi pembayaran sebagai hasil proses.



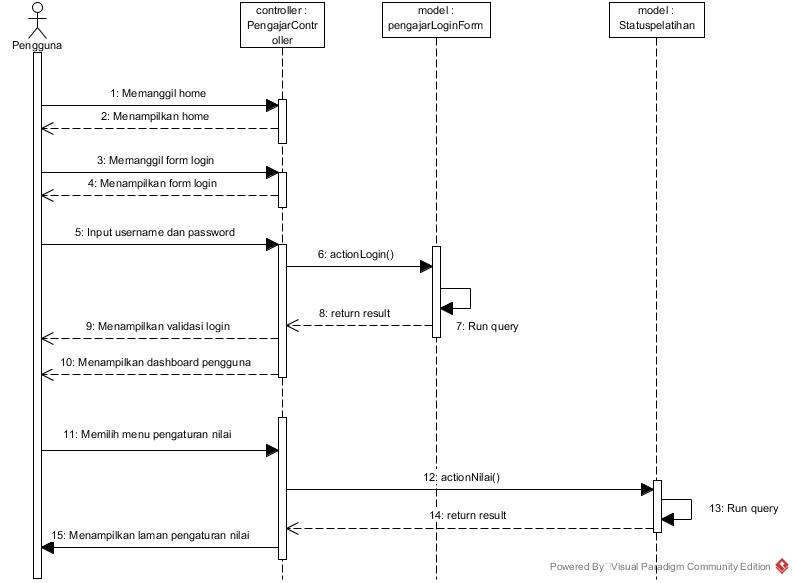
Gambar di atas merupakan *sequence diagram* untuk melihat kelas dari pelatihan yang diajarkan oleh pengajar. Proses ini menggunakan *controller* controllers/PengajarController.php sebagai logika kontrol dan mode/Statuspelatihan.php untuk melakukan *query*. *Sequence diagram* ini akan memanggil laman lihat kelas sebagai hasil proses.



Gambar di atas merupakan *sequence diagram* untuk melihat list peserta dari pelatihan yang diajarkan oleh pengajar. Proses ini menggunakan *controller* controllers/PengajarController.php sebagai logika kontrol dan mode/Statuspelatihan.php untuk melakukan *query*. *Sequence diagram* ini akan memanggil laman lihat peserta sebagai hasil proses.



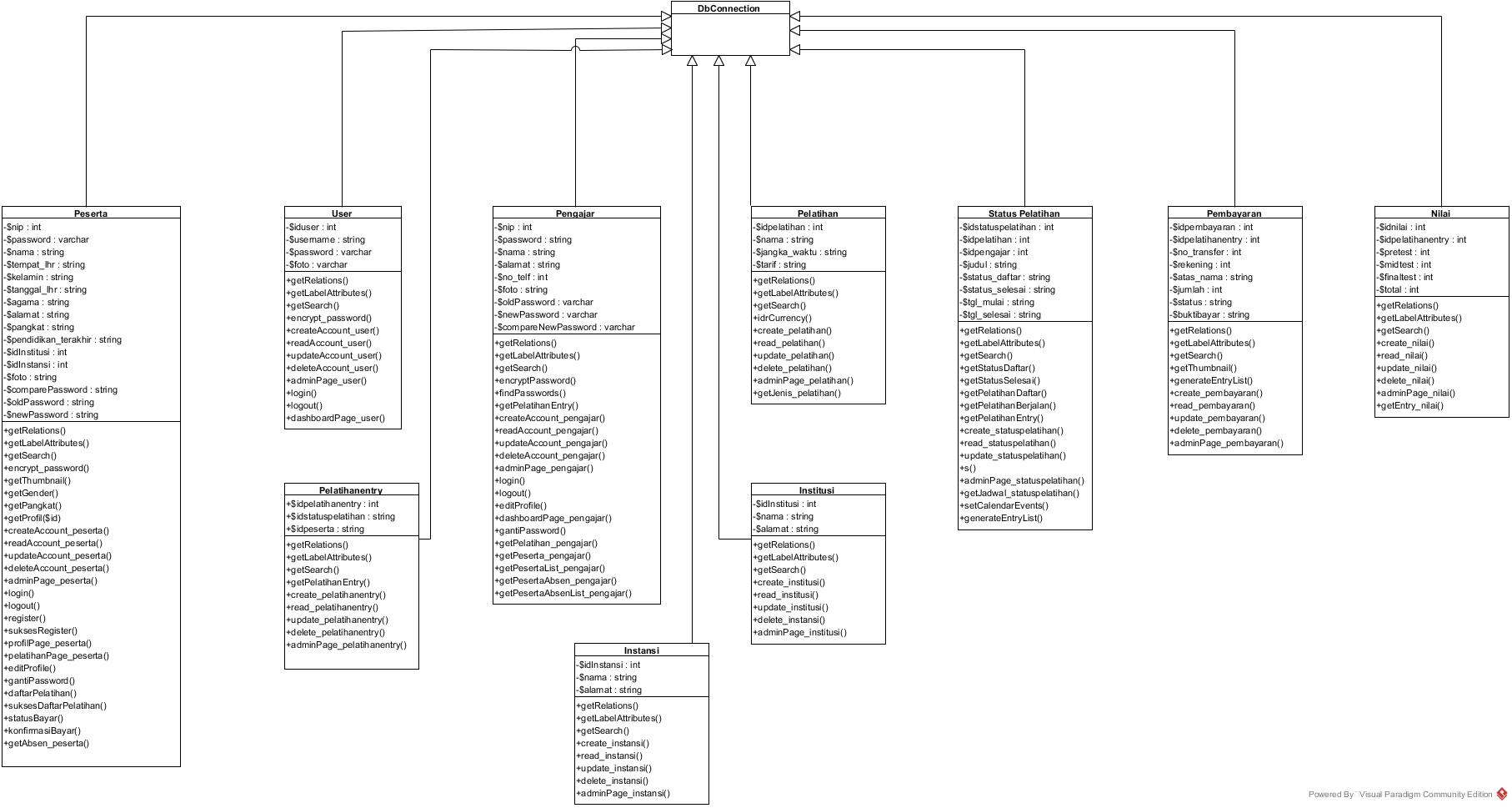
Gambar di atas merupakan *sequence diagram* untuk melihat pengaturan absen dari pelatihan yang diajarkan oleh pengajar. Proses ini menggunakan *controller* controllers/PengajarController.php sebagai logika kontrol dan mode/Statuspelatihan.php untuk melakukan *query*. *Sequence diagram* ini akan memanggil laman pengaturan absen sebagai hasil proses.



Gambar di atas merupakan *sequence diagram* untuk melihat pengaturan nilai dari pelatihan yang diajarkan oleh pengajar. Proses ini menggunakan *controller* controllers/PengajarController.php sebagai logika kontrol dan mode/Statuspelatihan.php untuk melakukan *query*. *Sequence diagram* ini akan memanggil laman pengaturan nilai sebagai hasil proses.

### *Class Diagram*

*Class diagram* menggambarkan hubungan kelas yang dibuat dengan kelas lainnya. Berikut adalah rancangan *class diagram* dari sistem administrasi Divisi Diklat Harapan Kita Jakarta



Gambar . *Class diagram prototype* sistem

Gambar 4.IV.18 *Class diagram prototype* sistem

Pada gambar 4.17, terdapat dua kelas yaitu cActiveRecord dan cController yang merupakan bagian dari Yii *framework*. Kedua kelas tersebut merupakan dasar ekstensi dari kelas-kelas yang terdapat pada *prototype* sistem. *Class diagram* dari Yii *framework* disertakan pada lampiran 7.

## Fitur Aplikasi

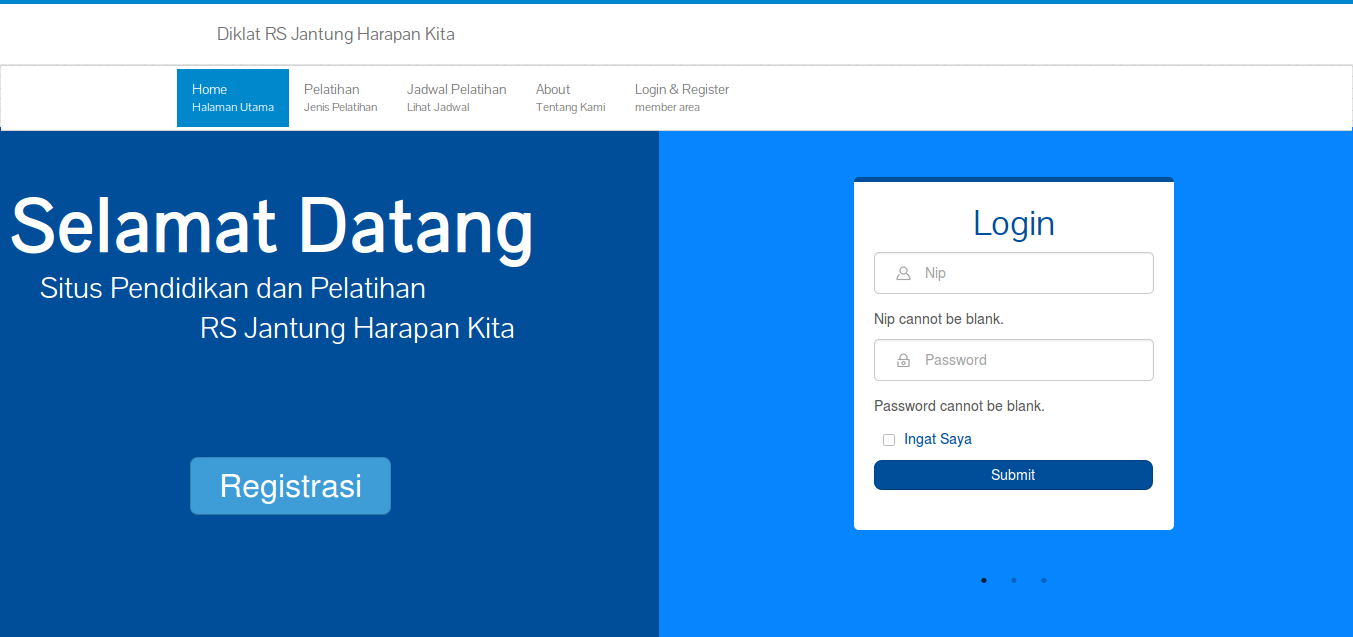
Pada bagian ini akan dijelaskan secara mendalam mengenai fitur-fitur aplikasi yang telah dibuat. Dalam sistem administrasi ini, terdapat tiga level akses yang setiap level mewakili tipe penggunanya. Adapun ketiga level akses tersebut adalah “User” yang mewakili administrator, “Pengajar” sebagai akses bagi pengajar, dan “Peserta” yang merupakan level untuk peserta. Ketiga level tersebut mempunyai fitur yang berbeda-beda. “User” merupakan level tertinggi yang dipakai untuk mengakses berbagai macam informasi yang terdapat di dalam sistem. Penulis membagi fitur-fitur yang terdapat pada sistem menjadi empat bagian, fitur general, user, peserta, dan pengajar.

### Fitur general

Fitur ini dimiliki atau dapat diakses oleh setiap level akses yang terdapat dalam sistem dan mencakup tampilan *user interface* sistem secara keseluruhan. Untuk mengakses fitur general tidak memerlukan autentikasi khusus.

#### Tampilan *user interface* sistem

Berikut merupakan contoh tampilan utama (Home) Sistem administrasi divisi diklat rumah sakit Harapan Kita.

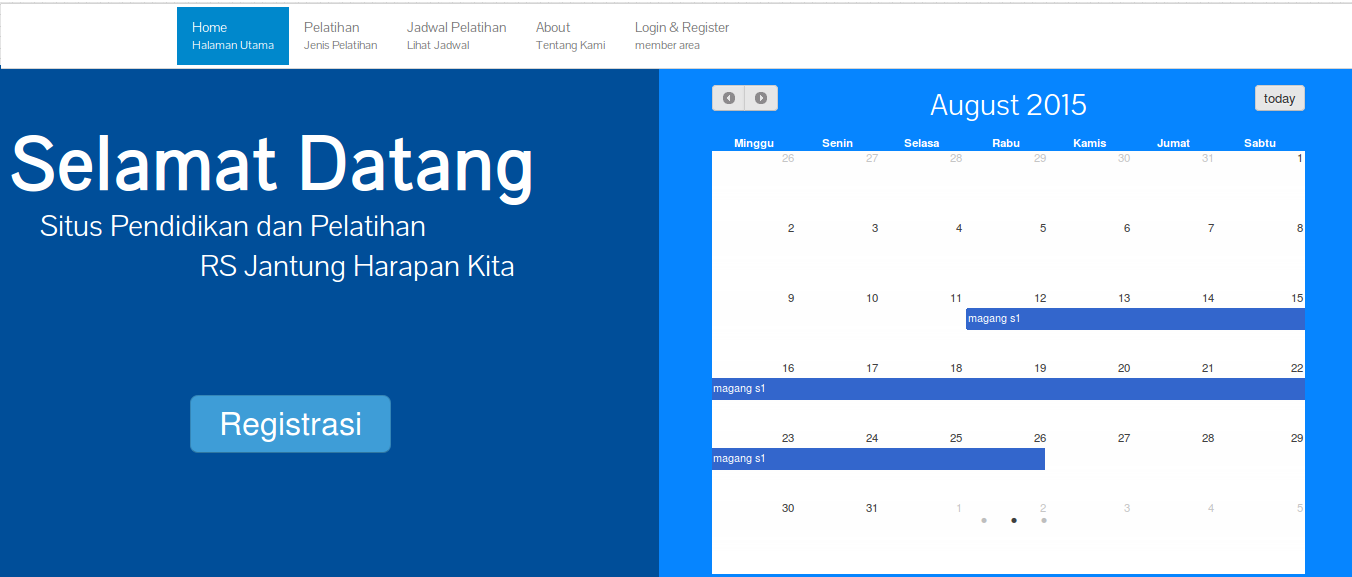


Gambar 4.IV.19 Tampilan laman utama (*home*) dengan bagian *content*

Laman ini akan tampil ketika pengguna membuka sistem untuk pertama kali. Terdapat bar navigasi yang berisi fitur general yang terdapat pada sistem. Dalam sistem ini terdapat bagian *content* yang berguna untuk menampilkan isi dari fungsi atau laman yang diminta pengguna. Pada laman utama ini disematkan tombol registrasi dan form login pada bagian *content* agar dapat memudahkan pengguna untuk segera mengakses akunnya. Sistem administrasi divisi diklat rumah sakit Harapan Kita ini menggunakan aksen biru sebagai basis warnanya. Hal ini karena warna biru secara umum dapat menggambarkan respon positif dan perasaan nyaman serta tenang [32]. Kombinasi warna yang lebih cerah dipakai untuk memudahkan pengguna yang berbeda-beda umur, dimana semakin bertambahnya usia warna akan terlihat lebih gelap. Hal ini membuat warna yang lebih cerah terlihat lebih menarik [33]

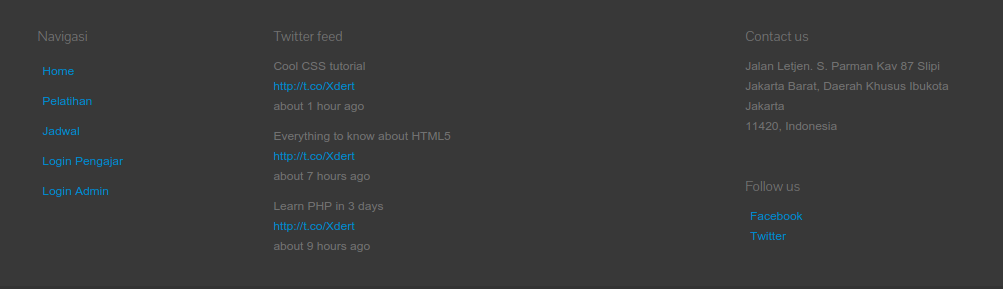
bar navigasi yang terletak pada bagian atas laman depan sistem berisi menu yang dapat digunakan untuk mengakses laman lain yang mempunyai informasi bagi pengguna. menu ini mempunyai 5 tombol yang terdiri dari Home(laman utama), pelatihan, jadwal pelatihan, tentang kami, login&register. Tombol Home akan mengirim pengguna kembali ke halaman utama. Sedangkan tombol pelatihan dan jadwal pelatihan masing-masing akan mengirim pengguna ke laman jenis pelatihan dan jadwal. Tombol tentang kami sendiri akan mengirim pengguna ke laman yang memberikan informasi tentang divisi diklat rumah sakit harapan kita. Terakhir, tombol login & register berisi form dan tombol untuk melakukan registrasi.

Pada laman utama ini, pengguna juga dapat merubah bagian kanan *content* melalui tombol yang terdapat di bawah login sehingga mengganti form tersebut menjadi kalender pelatihan atau informasi yang terkait dengan divisi diklat rumah sakit harapan kita Jakarta.



Gambar 4.IV.20 laman utama dengan kalender pelatihan

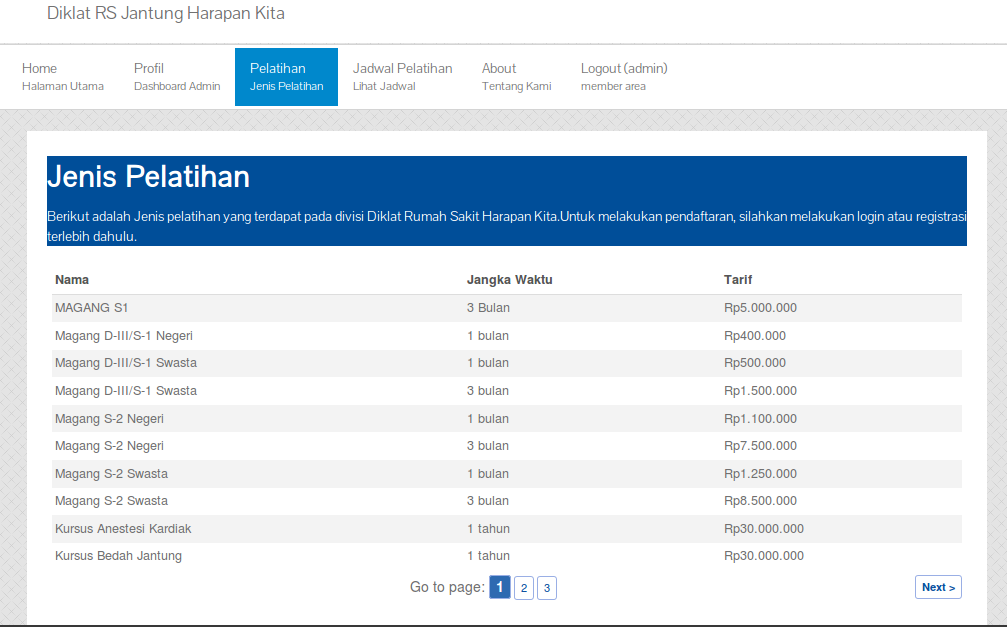
*Footer* juga disematkan dalam sistem administrasi divisi diklat rumah sakit harapan kita Jakarta ini. Bagian ini akan selalu mempunyai tampilan yang sama seperti bar navigasi. Tujuan penggunaan footer adalah untuk memberikan informasi tambahan dan *link login* bagi pengguna selain peserta.



Gambar 4.IV.21 Bagian footer *prototype* sistem

#### Jenis pelatihan

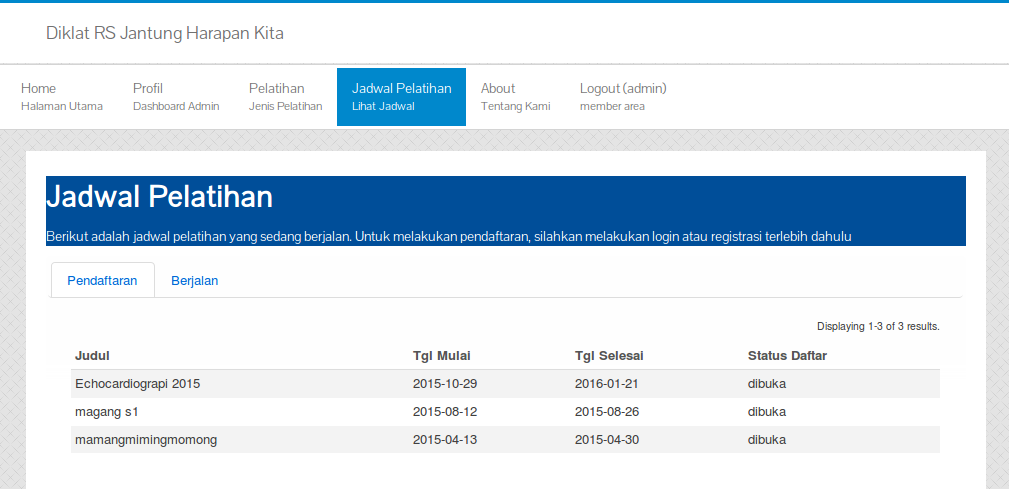
Setiap pengguna dapat melihat jenis pelatihan yang diadakan oleh Divisi Diklat Rumah Sakit Harapan Kita Jakarta. Jenis pelatihan tersebut dikemas dalam bentuk tabel, dan informasi yang disampaikan mencakup nama jenis, tarif, dan jangka waktu pelatihan berlangsung. Laman Jenis pelatihan dapat diakses melalui tombol “Pelatihan” pada bar navigasi.



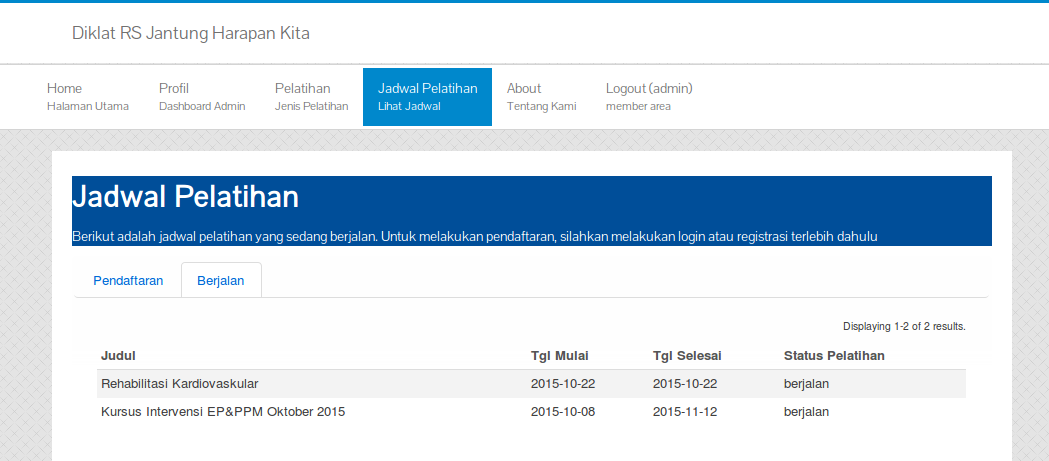
Gambar 4.IV.22 laman pelatihan pada *prototype* sistem

#### Jadwal pelatihan

Selain jenis pelatihan yang diadakan oleh divisi diklat rumah sakit harapan kita Jakarta, pengguna juga dapat melihat jadwal pelatihan yang sedang diadakan. Terdapat dua *tab* yang dibuat pada laman jadwal pelatihan. *Tab* pertama merupakan bagian yang berisi jadwal pelatihan dengan status “pendaftaran”. Hal ini menunjukkan bahwa pelatihan tersebut belum memulai kegiatan pelatihannya dan masih dalam tahap pendaftaran peserta. Sedangkan pada *tab* selanjutnya berisi jadwal pelatihan dengan status “berjalan”. Hal ini menandakan bahwa kegiatan pelatihan sedang berlangsung. Calon peserta tidak dapat melakukan pendaftaran pada jadwal pelatihan yang sedang berlangsung.



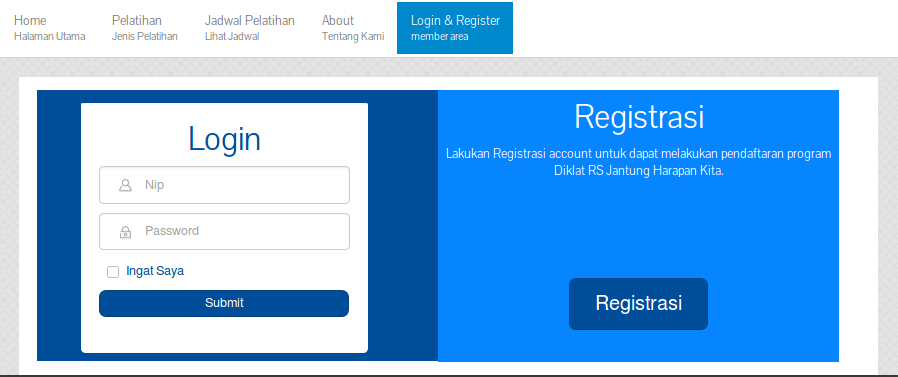
Gambar 4.IV.23 Laman jadwal pelatihan *tab* pendaftaran

******

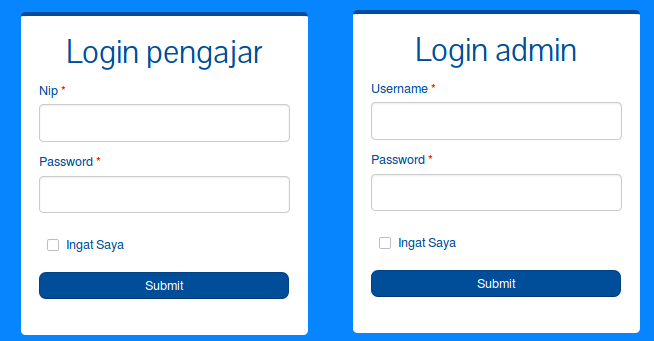
Gambar 4.IV.24 Laman jadwal pelatihan *tab* berjalan

#### *Login*

Dalam sistem administrasi yang telah dibuat oleh penulis, laman login dibagi menjadi tiga yang masing-masing mewakili kelas pengguna yang akan menggunakannya. laman login untuk peserta dapat diakses melalui *home* dan juga bar navigasi. Sedangkan laman login untuk admin dan pengajar dapat diakses melalui *footer* yang terdapat pada bagian bawah sistem. Laman login pada peserta dan pengajar akan meminta pengguna untuk memasukkan nip dan password agar dapat masuk ke dalam sistem. Sedangkan laman login pengguna meminta username admin sebagai pengganti nip.



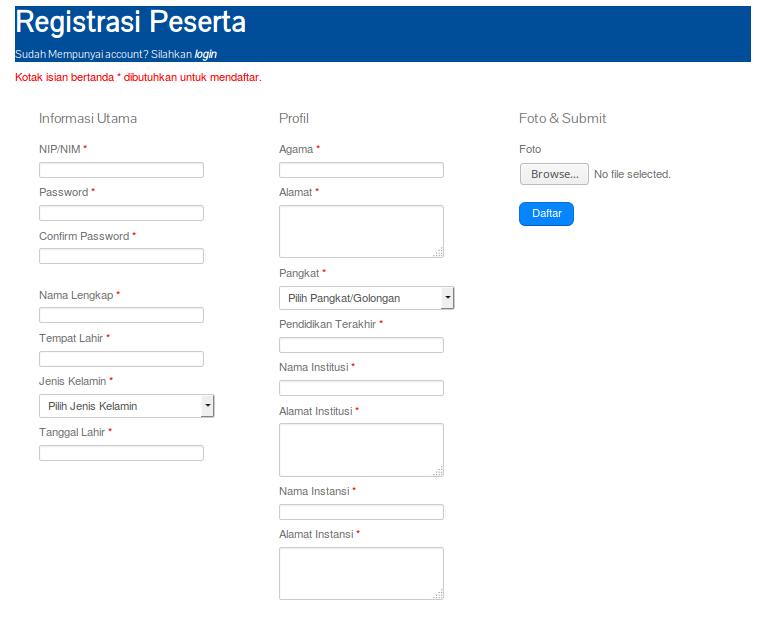
Gambar 4.IV.25 Laman *login* peserta

****

Gambar 4.IV.26 *form login* pengajar dan administrator

#### Registrasi

Laman registrasi berfungsi sebagai form pendaftaran bagi calon peserta. Sebelum bisa melakukan pendaftaran pelatihan, calon peserta diminta untuk mendaftarkan profil mereka terlebih dahulu melalui laman ini.

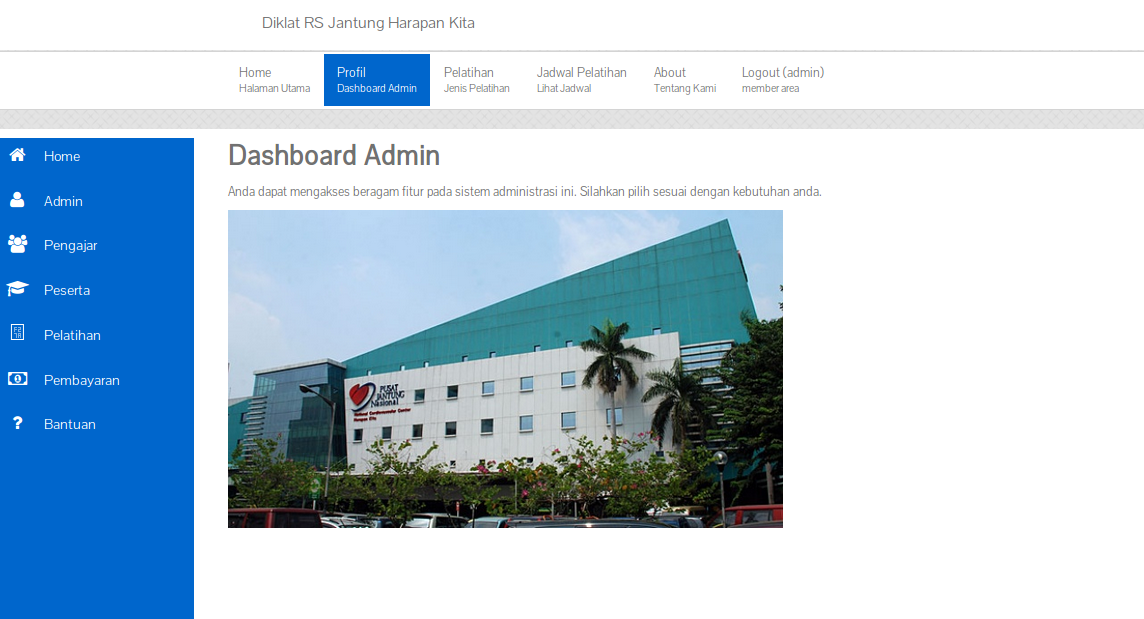


Gambar 4.IV.27 *form* registrasi peserta

### Fitur User (administrator)

Fitur admin adalah kumpulan dari fasilitas sistem yang hanya dapat diakses oleh pengguna bertipe administrator. Fitur ini bertujuan agar dapat mempermudah administrator dalam mengelola data yang terdapat pada sistem. Untuk mengakses fitur ini, administrator harus melakukan login melalui laman login admin.

#### *Dashboard*

****

Gambar 4.IV.28 Laman *dashboard* administrator

Laman dashboard yang terdapat pada sistem administrasi ini menggunakan *sidebar* sebagai menu utama dalam melakukan navigasi. *Sidebar* ini berisi berbagai macam fungsi yang berguna bagi administrator untuk mengolah data yang terdapat pada sistem. Menu yang terdapat pada *sidebar* juga mewakili fitur yang dapat diakses pada akun administrator.

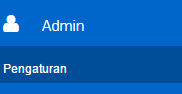
#### Home

Link ini akan membawa pengguna ke halaman utama *dashboard*.



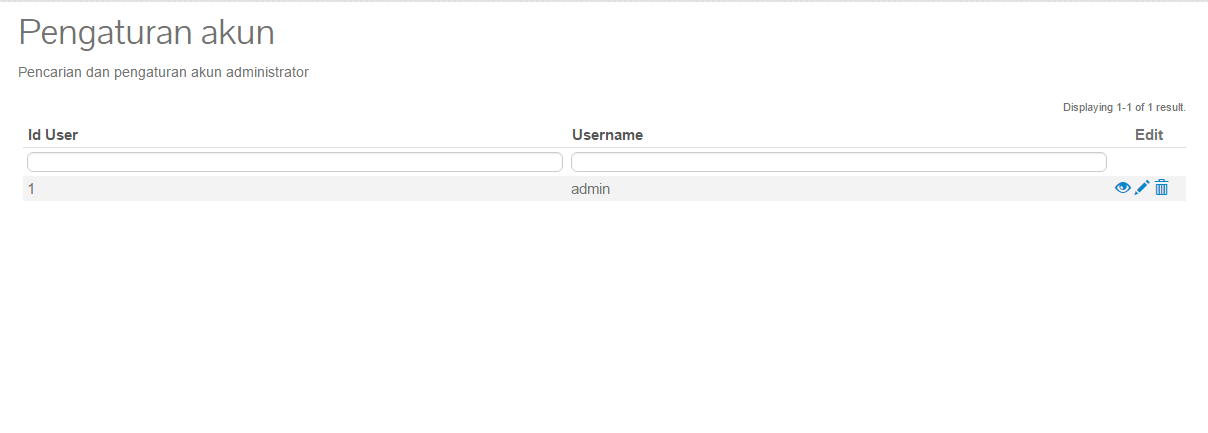
Gambar 4.IV.29 *link home* administrator

#### Admin



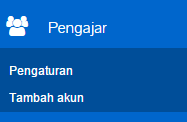
Gambar 4.IV.30 *link* admin administrator

Link ini akan membawa pengguna menuju pengaturan akun bagi administrator. Dalam pengaturan akun administrator, pengguna akan dapat melakukan pengeditan nama dan password untuk setiap akun yang terdapat di database.



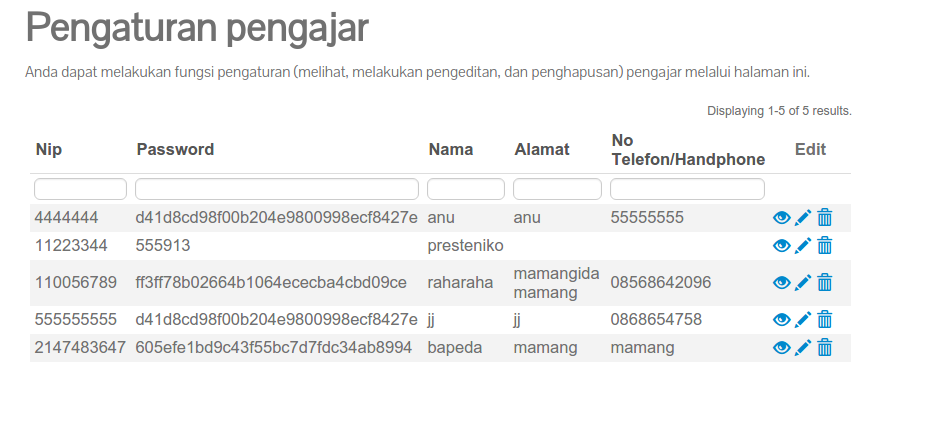
Gambar 4.IV.31 Laman pengaturan akun administrator

#### Pengajar

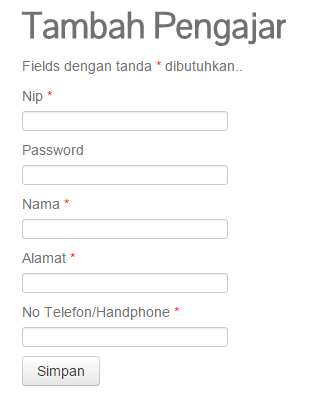


Gambar 4.IV.32 *Link* pengaturan akun pengajar

*Link* ini akan membawa pengguna menuju segala pengaturan sistem yang berhubungan dengan pengajar. Pengaturan pengajar tersebut meliputi pengaturan akun dan penambahan akun. Laman pengaturan pengajar sendiri berguna untuk melakukan perubahan perubahan pada akun pengajar yang terdapat pada *database.* Sedangkan laman penambahan akun berguna untuk pembuatan akun pengajar yang baru.

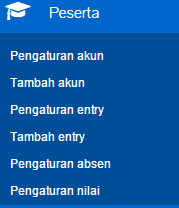


**Gambar 4.IV.33 Laman pengaturan pengajar**



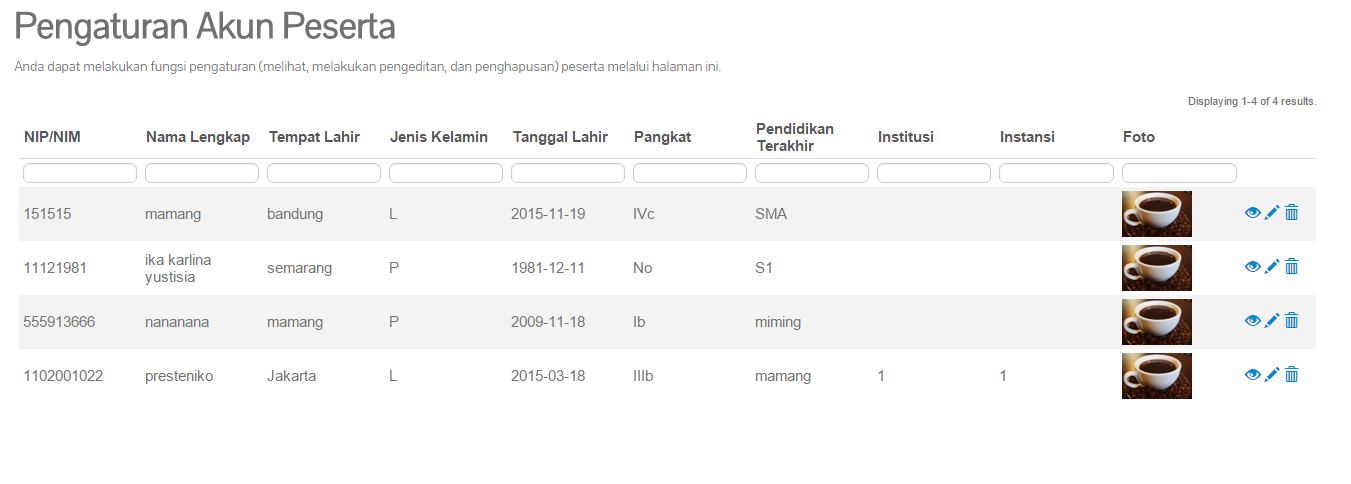
Gambar 4.IV.34 Laman tambah pengajar

#### Peserta



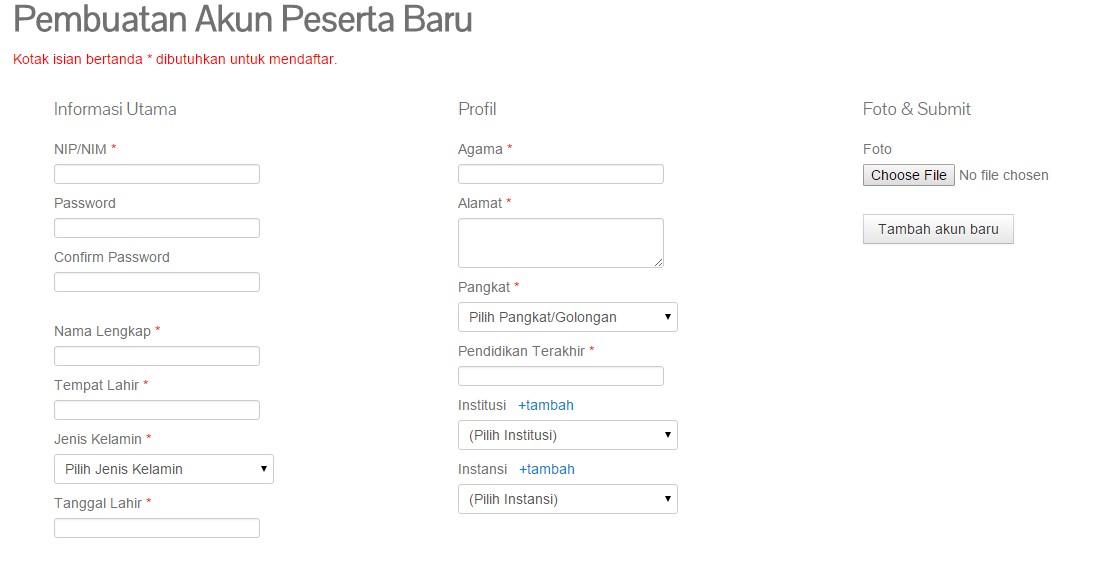
Gambar 4.IV.35 *Link* pengaturan akun peserta

*Link* ini akan membawa pengguna menuju laman pengaturan yang berhubungan dengan peserta, yaitu akun, entry, absen dan nilai. Akun peserta yang telah terdaftar dapat dirubah melalui laman pengaturan akun. Pada laman tersebut terdapat juga fasilitas untuk melakukan penghapusan dan melihat profil peserta secara lebih lengkap.



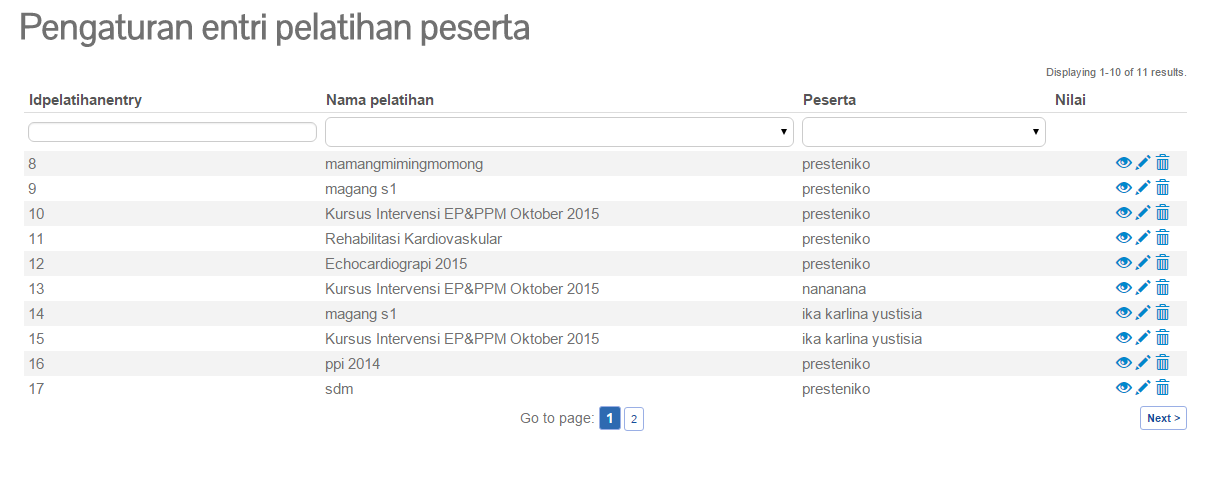
Gambar 4.IV.36 Laman pengaturan akun peserta

Administrator juga dapat melakukan pembuatan akun baru melalui menu tambah akun.



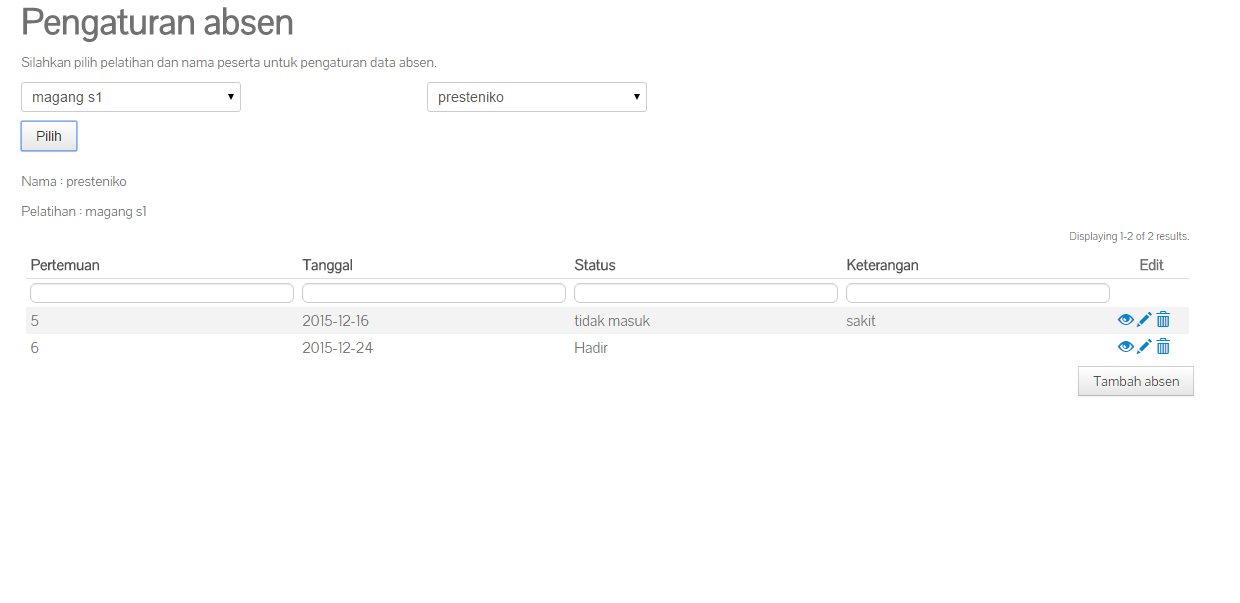
Gambar 4.IV.37 Laman pembuatan akun peserta baru

Selain mengatur akun peserta yang terdapat pada *database*, pengguna administrator juga dapat melakukan pengaturan bagi entry, absen, dan nilai. *Entry* sendiri merupakan pelatihan yang telah didaftarkan oleh peserta. Sama seperti dalam pengaturan akun, pengguna atau administrator dapat melakukan perubahan dan penghapusan *entry* pada laman pengaturan.



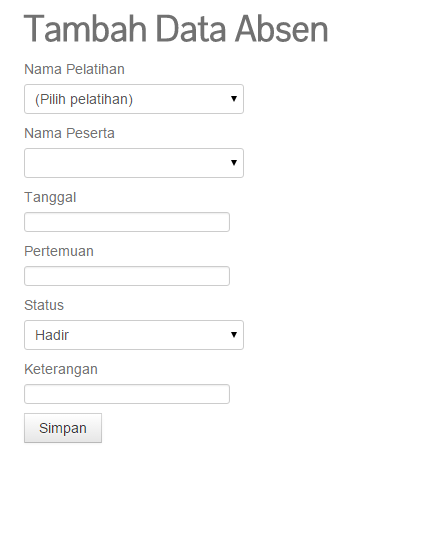
Gambar 4.IV.38 Laman pengaturan entri pelatihan bagi peserta

Untuk laman pengaturan absen, pengguna atau administrator harus terlebih dahulu memilih pelatihan dan nama peserta. Setelah pemilihan dilakukan oleh pengguna, sistem akan menampilkan list absen yang sesuai. Pengguna dapat melakukan perubahan pada list absen tersebut. Pembuatan absen baru dapat dilakukan pada laman pengaturan atau tambah absen.



Gambar 4.IV.39 Laman pengaturan absen pelatihan peserta

Pada penambahan absen baru, pengguna diminta untuk memasukkan data-data yang berkaitan dengan absen peserta, seperti nama pelatihan, peserta, status dan keterangan.



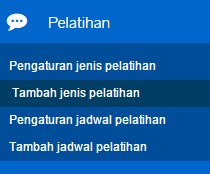
Gambar 4.IV.40 Laman tambah data absen

Seperti pengaturan absen, pengaturan untuk nilai juga dilakukan dengan memilih pelatihan dan nama peserta yang ikut. Pengguna atau administrator tidak dapat melakukan penambahan nilai baru. Hal ini dikarenakan saat seorang peserta melakukan pendaftaran pelatihan, data nilai awal sudah otomatis dibuat oleh sistem. Data nilai awal tersebut bernilai nol (0).



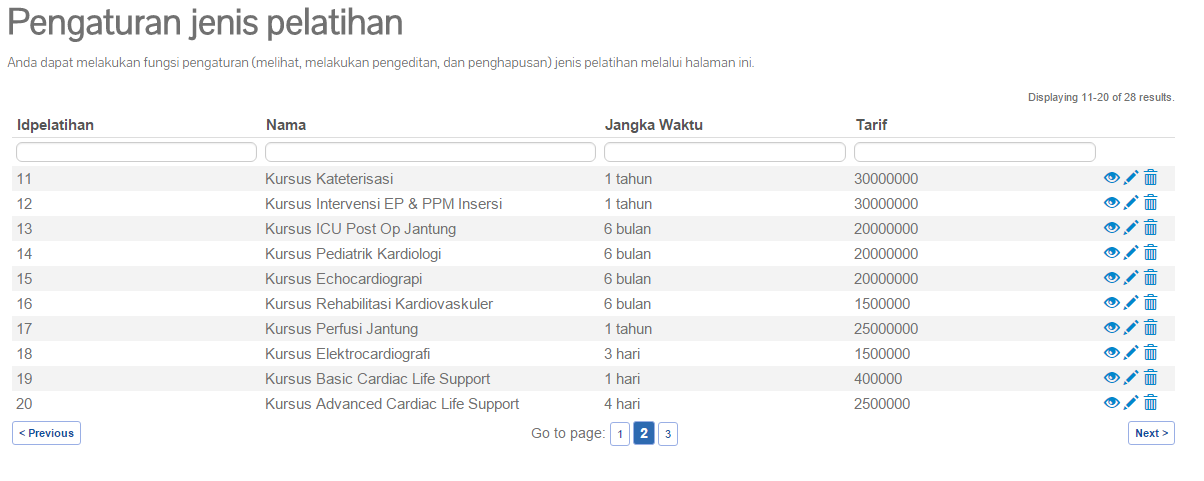
Gambar 4.IV.41 Laman pengaturan nilai peserta

#### Pelatihan



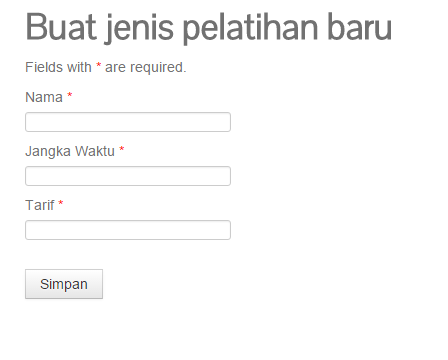
Gambar 4.IV.42 *Link* pengaturan pelatihan

Pada *link* pelatihan, pengguna dapat melakukan pengaturan untuk data-data yang berhubungan dengan jenis dan jadwal pelatihan yang diadakan di Rumah Sakit Harapan Kita Jakarta. Jenis pelatihan merupakan data dasar dari tipe pelatihan yang diadakan Data pada jenis pelatihan mencakup nama, tarif, dan jangka waktu.



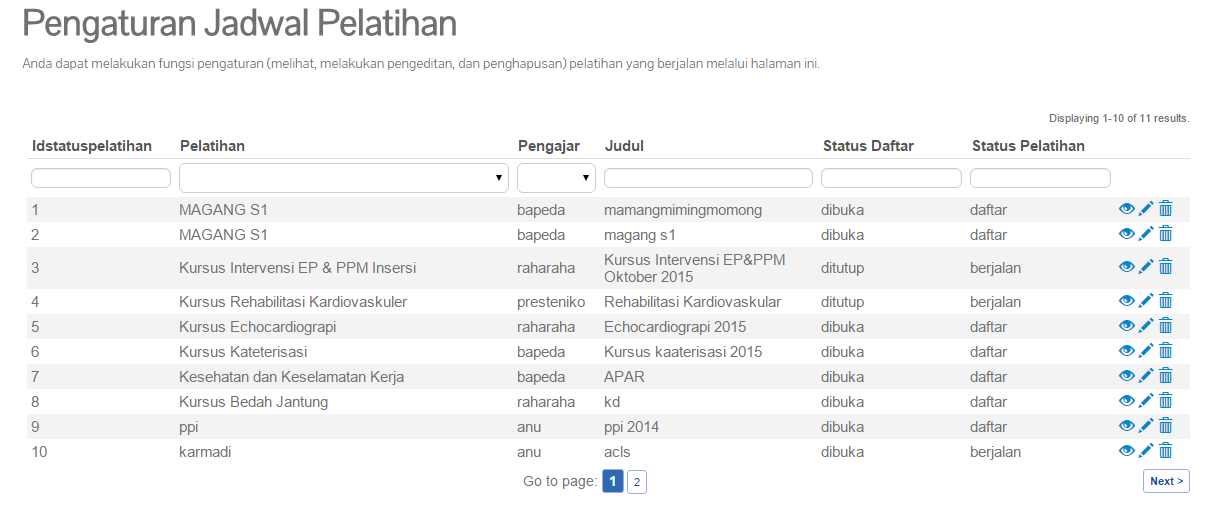
Gambar 4.IV.43 Laman pengaturan jenis pelatihan

Pada laman pengaturan jenis pelatihan ini, pengguna atau administrator dapat melihat jenis pelatihan lebih detail, melakukan pembaruan maupun melakukan penghapusan pada data. Data yang berada pada pengaturan ini akan diteruskan pada menu “pelatihan” yang terdapat pada bar navigasi. Pengguna atau administrator juga dapat melakukan penambahan jenis pelatihan baru yang terdapat pada *link* “tambah jenis pelatihan” yang terdapat pada *sidebar* dari dashboard. Pada laman “tambah jenis pelatihan” ini, pengguna harus memasukkan data nama, jangka waktu, dan tarif dari jenis pelatihan yang akan dibuat.



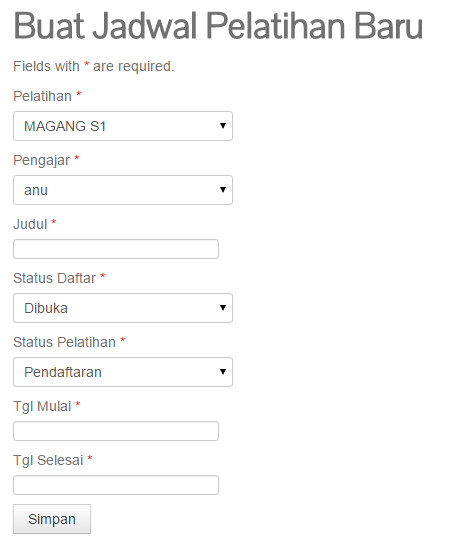
Gambar 4.IV.44 Laman pembuatan jenis pelatihan baru

Berbeda dengan jenis pelatihan, jadwal pelatihan merupakan pelatihan yang sudah mempunyai jadwal. Data pada jadwal pelatihan mencakup judul pelatihan, nama pengajar, status pendaftaran dan tanggal dimana pelatihan berlangsung. Jadwal pelatihan juga memerlukan jenis pelatihan yang akan diadakan. Oleh karena itu, jadwal pelatihan akan mengambil data yang berasal dari jenis pelatihan.



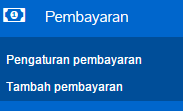
Gambar 4.IV.45 Laman pengaturan jadwal pelatihan

Pada laman pengaturan, Pengguna atau administrator dapat melakukan pengaturan untuk jadwal dari pelatihan. Informasi jadwal pelatihan akan diteruskan pada menu “jadwal” yang terdapat pada bar navigasi. Pengguna atau administrator juga dapat melakukan penambahan jadwal pelatihan baru yang terdapat pada *link* “tambah jadwal pelatihan” yang terdapat pada *sidebar* dari dashboard. Pada laman “tambah jadwal pelatihan” ini, pengguna harus memasukkan data jenis pelatihan, pengajar, judul, status daftar, status pelatihan, tanggal mulai serta tanggal selesai.



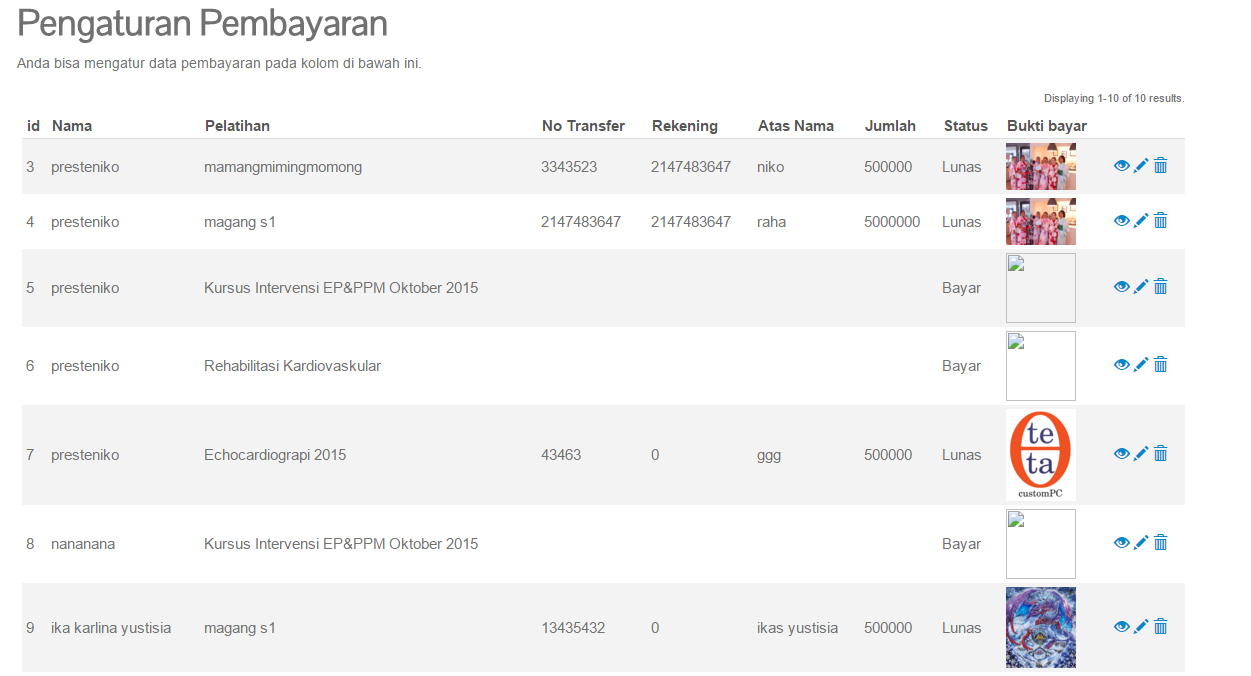
Gambar 4.IV.46 Laman pembuatan jadwal pelatihan baru

#### Pembayaran



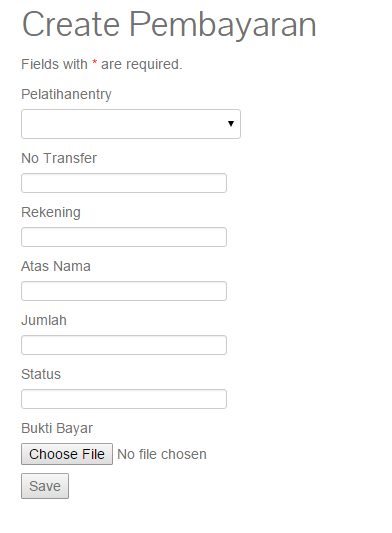
Gambar 4.IV.47 *Link* pembayaran pelatihan

Pada *link* pembayaran, pengguna atau administrator dapat melakukan pengaturan yang berhubungan dengan pembayaran. Pembayaran akan dilakukan sewaktu peserta melakukan pendaftaran dari pelatihan yang diadakan di Rumah Sakit Jantung Harapan Kita Jakarta.



Gambar 4.IV.48 Laman pengaturan pembayaran

Data yang terdapat pada laman pengaturan pembayaran berisi data yang telah dimasukkan oleh peserta pada saat melakukan konfirmasi pembayaran. Pengguna atau administrator dapat melakukan pembaruan atau melihat secara detil dan menghapus data apabila terjadi kesalahan.



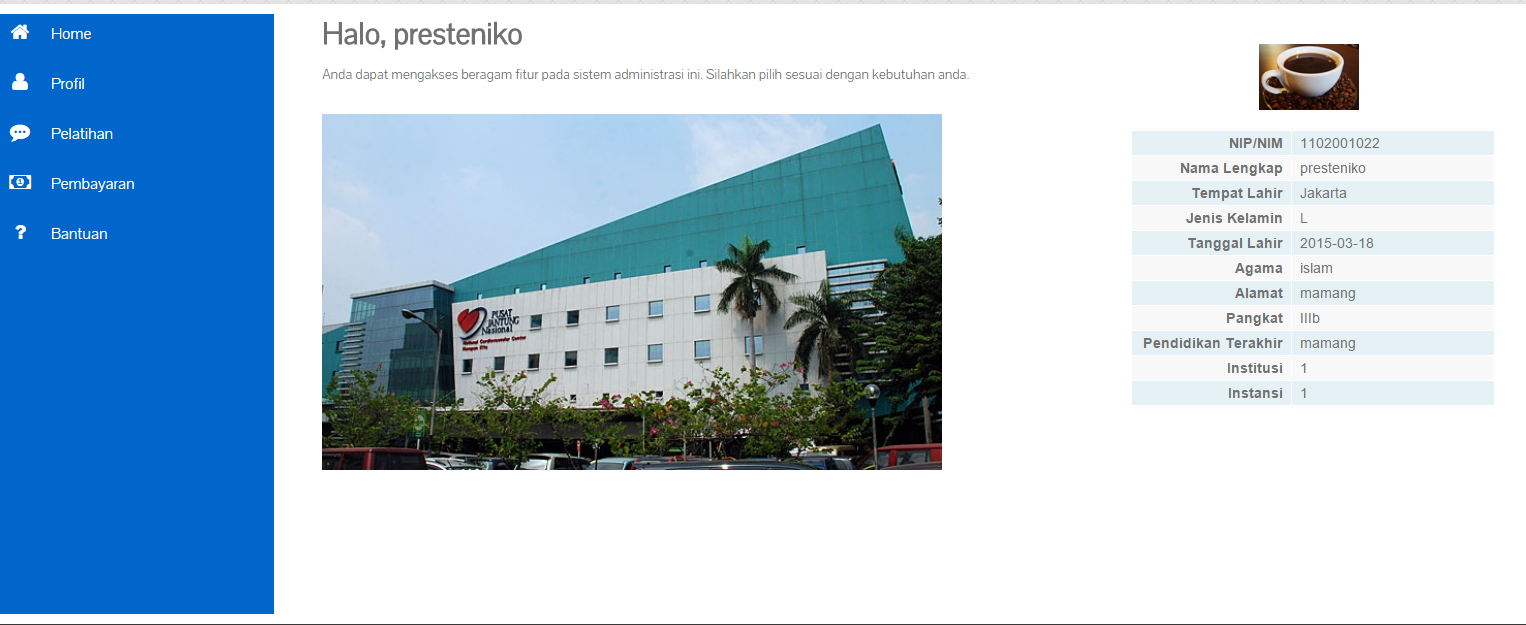
Gambar 4.IV.49 Laman pembuatan pembayaran baru

Pengguna atau administrator juga dapat melakukan penambahan pembayaran melalui laman “tambah pembayaran” yang terdapat pada menu *sidebar* dashboard. Pada laman ini pengguna atau administrator diminta untuk memasukkan data entri, nomor transfer, rekening, atas nama, jumlah, status serta foto bukti bayar.

### Fitur Peserta

Fitur peserta adalah kumpulan dari fasilitas sistem yang hanya dapat diakses oleh pengguna bertipe peserta. Fitur ini bertujuan agar dapat mempermudah peserta dalam melihat data pelatihan dan pembayaran yang diikuti. Untuk mengakses fitur ini, peserta harus mendaftarkan diri terlebih dahulu di laman registrasi dan melakukan login pada laman login peserta.

#### Dashboard profil



Laman dashboard yang terdapat pada sistem administrasi ini menggunakan *sidebar* sebagai menu utama dalam melakukan navigasi. *Sidebar* ini berisi berbagai macam fungsi fitur yang berhubungan dengan pengguna peserta. Pada laman ini juga terdapat profil peserta. Menu yang terdapat pada *sidebar* juga mewakili fitur yang dapat diakses pada akun peserta.

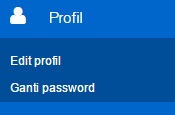
#### Home

Link ini akan membawa peserta ke laman utama dari dashboard.



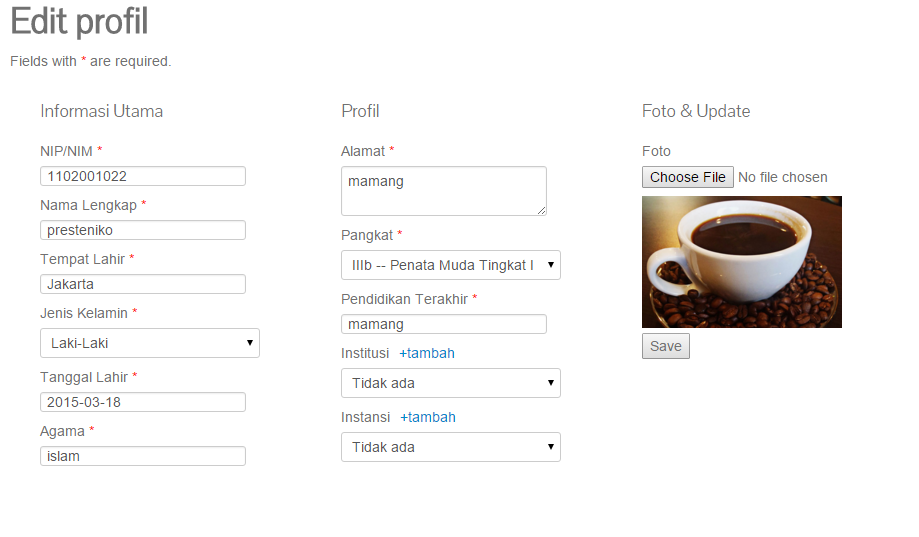
Gambar 4.IV.50 *Link home*

#### Profil



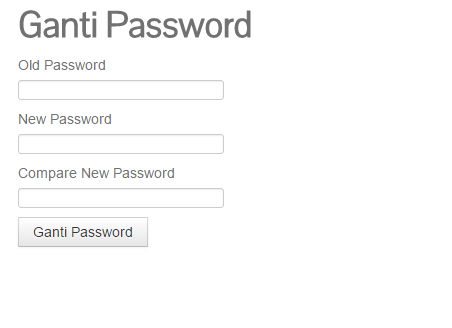
Gambar 4.IV.51 *Link* pengaturan profil peserta

*Link* ini akan membawa peserta ke laman pengaturan yang berhubungan dengan akun. Peserta dapat melakukan perubahan atau pembaruan profil pada laman “edit profil”. Laman “edit profil” ini akan menampilkan *form* yang dapat diperbarui oleh peserta.



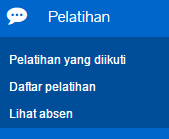
Gambar 4.IV.52 Laman edit profil peserta

Tidak hanya melakukan pembaruan pada profil, peserta juga dapat melakukan perubahan password di laman “ganti password”.



Gambar 4.IV.53 Laman ganti password peserta

#### Pelatihan



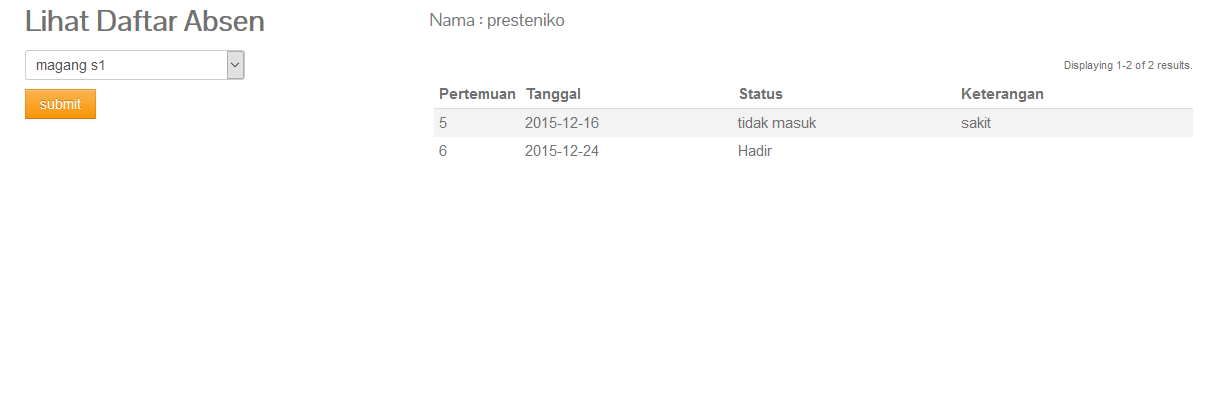
Gambar 4.IV.54 *Link* pengaturan pelatihan peserta

*Link* ini akan membawa peserta ke hal-hal yang berhubungan dengan pelatihan yang akan atau sudah mereka ikuti. Peserta dapat melihat daftar pelatihan yang mereka ikuti melalui laman “pelatihan yang diikuti”. Pada laman ini terdapat kolom nilai yang merupakan nilai rata-rata dari *midtest* dan *finaltest*.



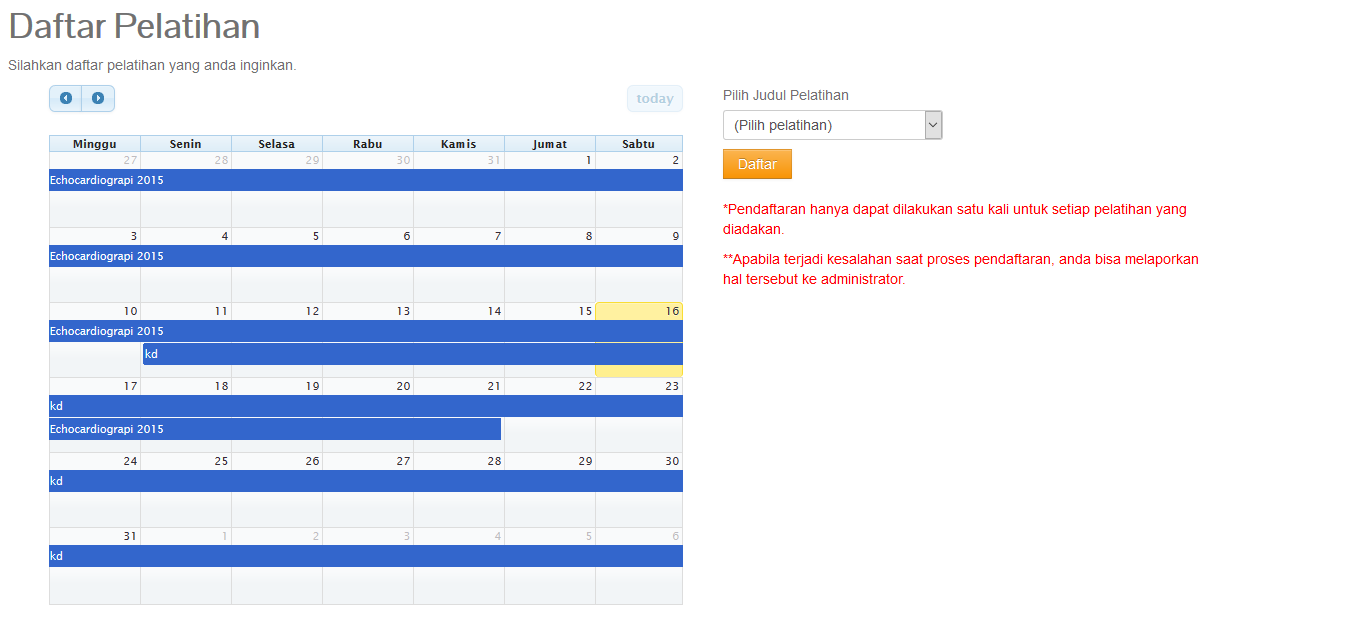
Gambar 4.IV.55 Laman list pelatihan yang diikuti

Peserta juga dapat melihat daftar absen dari pelatihan yang sedang mereka ikuti melalui laman “lihat absen”. Pada laman ini, peserta akan diminta untuk memasukkan nama pelatihan yang mereka ikuti. Setelah itu sistem akan menampilkan list absen yang ada sesuai dengan pelatihan yang dimasukkan.



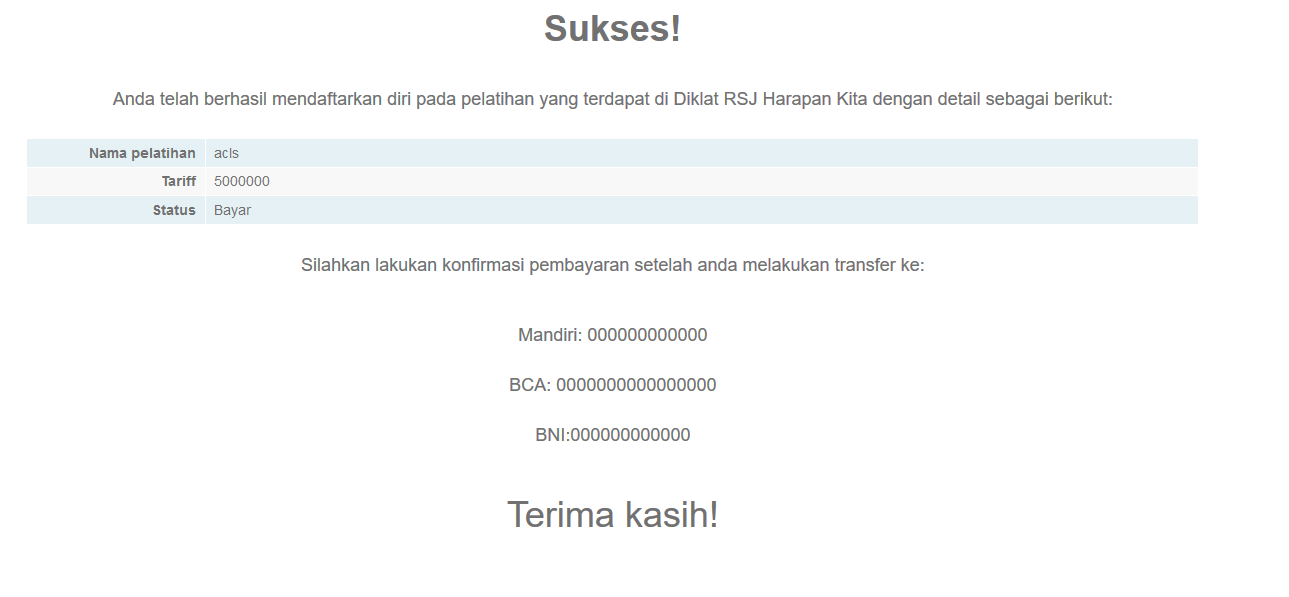
Gambar 4.IV.56 Laman lihat daftar absen pelatihan peserta

Tidak hanya melihat pelatihan yang diikuti dan absen, peserta juga dapat melakukan pendaftaran melalui *link* ini. Laman “daftar pelatihan” akan menampilkan kalender kegiatan yang dapat dilihat peserta. Setelah itu peserta akan melakukan pendaftaran dengan memasukkan nama pelatihan yang dia inginkan.



Gambar 4.IV.57 Laman pendaftaran pelatihan

Apabila peserta berhasil melakukan pendaftaran, maka sistem akan membawa peserta ke laman “sukses daftar”. Pada laman ini sistem akan memberikan informasi kepada peserta tentang tata cara pembayaran dan berapa nominal uang yang harus dibayarkan.



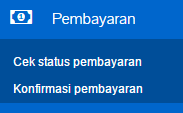
Gambar 4.IV.58 Laman sukses pendaftaran

Sedangkan apabila peserta telah mendaftarkan diri pada pelatihan yang sama sebelumnya, maka sistem akan melarang pendaftaran tersebut dan memberikan pemberitahuan bahwa peserta telah melakukan pendaftaran pada pelatihan yang sama.



Gambar 4.IV.59 Notifikasi sudah terdaftar dalam pelatihan

#### Pembayaran



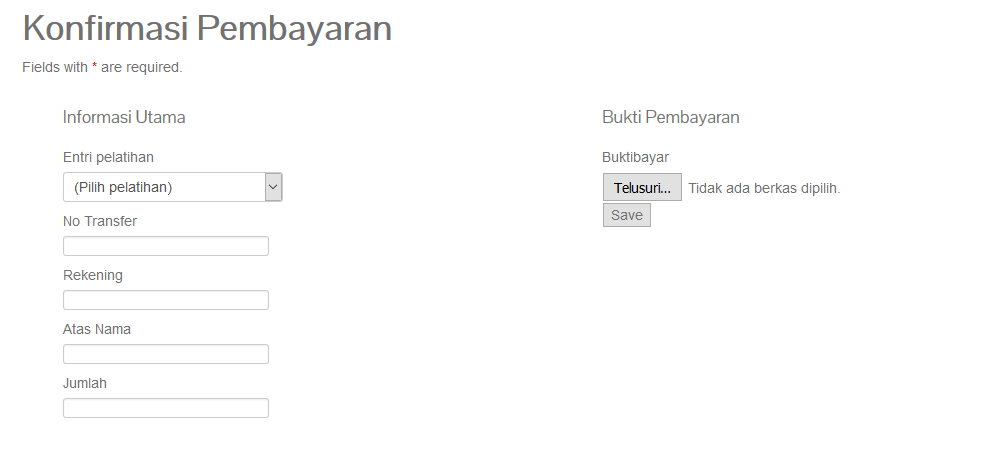
Gambar 4.IV.60 *Link* pembayaran pelatihan peserta

*Link* ini akan membawa peserta ke hal-hal yang berhubungan dengan pembayaran yang telah mereka ikuti. Peserta akan dapat melacak status pembayaran melalui sistem. Pada laman “cek status pembayaran”, peserta dapat melihat status pembayaran dari setiap pelatihan yang mereka ikuti. Apabila pembayaran dari pelatihan tersebut belum lunas, maka kolom status akan menunjukkan kata “bayar” dengan kolom pembayaran berisi “konfirmasi”. Apabila pembayaran peserta sudah lunas, maka isi kolom status akan berubah ke “lunas”.



Gambar 4.IV.61 Laman *list* pembayaran

Pada saat pendaftaran, peserta akan diminta untuk melakukan pendaftaran. Setelah melakukan pembayaran, peserta akan diminta untuk melakukan konfirmasi agar peserta benar-benar telah terdaftar pada pelatihan yang didaftarkan. Peserta dapat melakukan konfirmasi tersebut melalui laman “konfirmasi pembayaran”. Pada laman “konfirmasi pembayaran” ini, peserta harus mengisi data pelatihan yang diikuti, nomor transfer yang telah dilakukan, nama rekening transfer, atas nama, serta jumlah yang diberikan. Tidak hanya itu, peserta juga diminta untuk mengunggah foto bukti transfer pembayaran pelatihan.

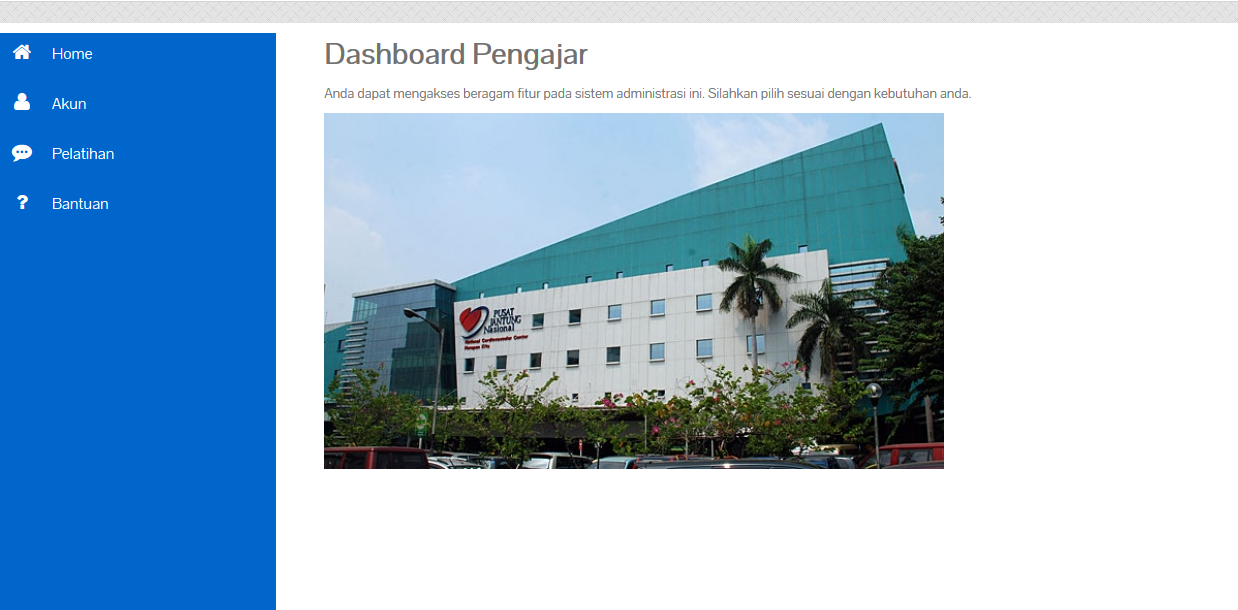


Gambar 4.IV.62 Laman konfirmasi pembayaran

### Fitur Pengajar

Fitur pengajar adalah kumpulan dari fasilitas sistem yang hanya dapat diakses oleh pengguna bertipe pengajar. Fitur ini bertujuan agar pengajar dapat melakukan pengecekan serta melakukan penambahan data berdasarakan pelatihan yang meraka ajarkan. Pengajar harus melakukan login agar dapat masuk ke fitur ini.

#### *Dashboard*



Gambar 4.IV.63 Laman *dasboard* pengajar

Laman *dashboard* yang terdapat pada sistem administrasi ini menggunakan *sidebar* sebagai menu utama dalam melakukan navigasi. *Sidebar* ini berisi berbagai macam fungsi fitur yang berhubungan dengan pengguna pengajar. Menu yang terdapat pada *sidebar* juga mewakili fitur yang dapat diakses pada akun peserta.

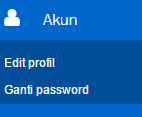
#### Home

Link ini akan membawa peserta ke laman utama dari dashboard.



Gambar 4.IV.64 *Link home*

#### Profil



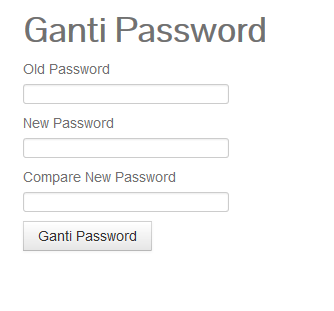
Gambar 4.IV.65 *Link* pengaturan akun pengajar

*Link* ini akan membawa pengajar ke laman pengaturan yang berhubungan dengan akun. Pengajar dapat melakukan perubahan atau pembaruan profil pada laman “edit profil”. Laman “edit profil” ini akan menampilkan *form* yang dapat diperbarui oleh pengajar.



Gambar 4.IV.66 Laman edit profil pengajar

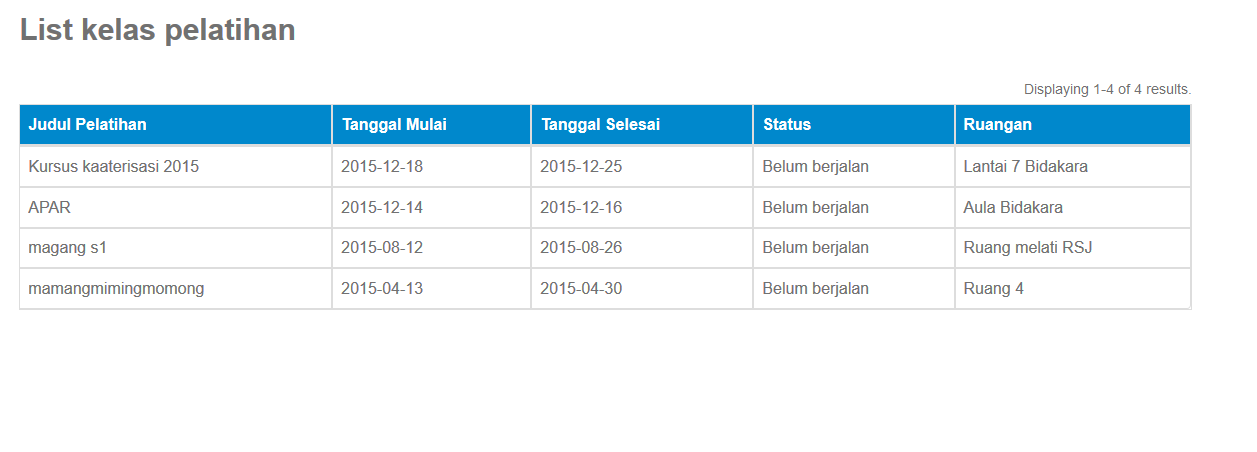
Tidak hanya melakukan pembaruan pada profil, pengajar juga dapat melakukan perubahan password di laman “ganti password”.



Gambar 4.IV.67 Laman ganti password pengajar

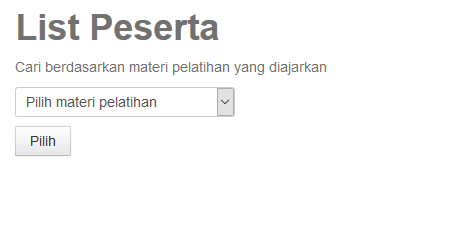
#### Pelatihan

*Link* ini akan membawa pengajar ke laman pengaturan yang berhubungan dengan pelatihan yang diajarkan. Pada laman “lihat kelas”, pengajar dapat melihat pelatihan apa saja yang mereka ajarkan. Pengajar juga dapat melihat status serta ruang kelas untuk pelatihan tersebut.



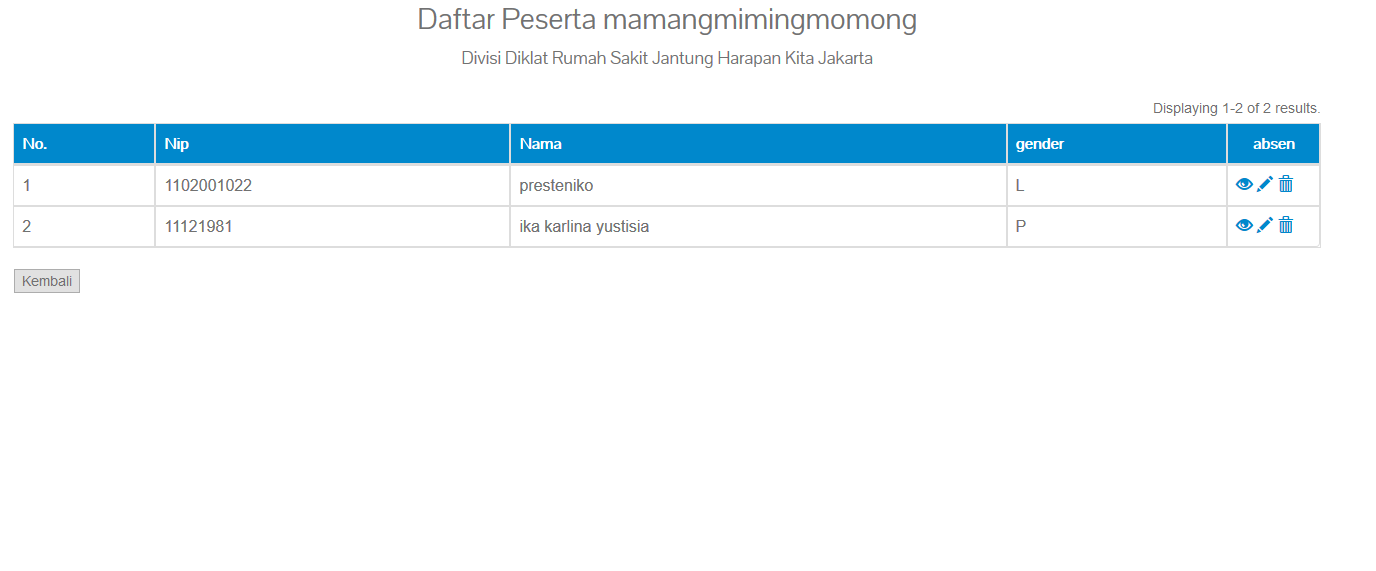
Gambar 4.IV.68 Laman *list* kelas pelatihan

Selain melihat pelatihan apa saja yang diajarkan, pengajar juga dapat melihat list peserta yang ikut dalam pelatihan tersebut. Pada laman “lihat peserta”, pengajar diminta untuk memasukkan nama pelatihan yang diinginkan. Hanya pelatihan yang diajarkan oleh pengajar yang dapat dilihat.



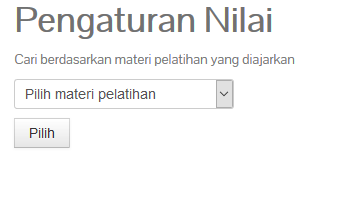
Gambar 4.IV.69 Laman *list* peserta

Setelah nama pelatihan berhasil dimasukkan, sistem akan memberikan list peserta.



Gambar 4.IV.70 Laman daftar peserta

Nilai untuk setiap peserta juga dapat dilihat melalui laman “pengaturan nilai”. sama seperti laman “list peserta”, pengajar diminta untuk memasukkan nama pelatihan yang ingin dilihat nilainya. Hanya pelatihan yang diajarkan oleh pengajar yang dapat dipilih.



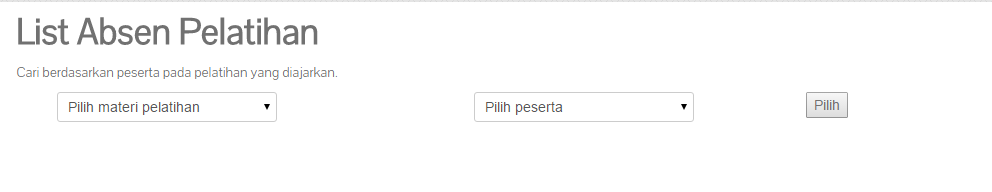
Gambar 4.IV.71 Laman pengaturan nilai

Setelah nama pelatihan berhasil dimasukkan, sistem akan memberikan *list* peserta beserta nilai yang mereka punyai.



Gambar 4.IV.72 Laman pengaturan nilai

Pada laman “pengaturan absen”, pengajar dapat melihat daftar absen dari setiap peserta yang diajarkan. Untuk dapat melihat absen, pengajar diminta untuk memasukkan nama pelatihan dan peserta.



Gambar 4.IV.73 Laman *list* absen pelatihan

Setelah pengajar berhasil memasukkan nama pelatihan dan peserta, sistem akan memberikan list absen dari hasil masukan.



Gambar 4.IV.74 Laman daftar absen peserta

## Usability Test

### Langkah usability test

*Usability testing* dilakukan pada tahap terakhir dari penelitian, yaitu *feedback*. Tujuan dari *testing* adalah untuk mengukur kegunaan dari *prototype* sistem berdasarkan perasaan yang dirasakan oleh calon pengguna. Sebanyak 30 orang calon pengguna yang terdiri dari pegawai, pengajar, dan peserta ikut serta dalam melakukan pengujian ini. Pengambilan sampel sebanyak 30 orang untuk *usability test* ini didasarkan oleh penelitian yang dilakukan oleh Laura Faulkner. Penelitian ini menyimpulkan bahwa *test* yang dilakukan dengan 20 atau lebih sampel dapat meningkatkan penemuan masalah pada *usability* [34]. Faulkner juga menerangkan bahwa dengan 30 sampel, presentase penemuan masalah dapat bernilai sampai dengan 98%. Penghitungan *usability test* berdasarkan *self-reported metric* dengan mengukur tingkat kepuasan pada *prototype* sistem. Langkah-langkah *usability test* yang dilakukan pada penelitian ini adalah:

1. Menemui calon pengguna

Pada tahap ini, Penulis menemui calon pengguna dan memberikan penjelasan tentang *usability test* yang akan dilakukan.

1. Penulisan data diri

Pada tahap ini, Penulis akan memberikan lembaran survey kepada Calon pengguna. Calon pengguna lalu menuliskan data diri yang terdiri dari nama, nomor telefon, jenis kelamin, jarak umur, dan pendidikan terakhir.

1. Penjelasan sistem

Pada tahap ini, Penulis akan memberikan penjelasan tentang *prototype* sistem yang dibangun pada penelitian ini kepada Calon penguna.

1. Penjelasan langkah *usability test*

Pada tahap ini , Penulis akan memberikan penjelasan tentang langkah yang dilakukan oleh Calon pengguna pada *usability test*. Langkah tersebut meliputi:

1. Calon pengguna akan dibimbing oleh Penulis dalam mencoba dan menggunakan *prototype* sistem. Pada lembaran survey terdapat *list* dari beberapa hal yang dapat dilakukan oleh Calon pengguna sesuai dengan tipe masing-masing (administrator, pengajar, peserta). *List* tersebut hanya digunakan sebagai panduan.
2. Setelah Calon pengguna mencoba dan menggunakan *prototype* sistem, mereka akan diminta untuk mengisi kuisioner yang terdapat pada lembaran survey. Kuisoner tersebut berisi dua puluh dua pertanyaan yang terbagi menjadi lima kategori. Kategori tersebut adalah *ease of use, customization, download delay, content, satisfaction.*
3. Percobaan *prototype* sistem

Pata tahap ini, Calon pengguna akan mencoba dan menggunakan *prototype* sistem sesuai dengan *list* yang terdapat pada lembaran survey. Penulis akan membantu dan membimbing Calon pengguna.

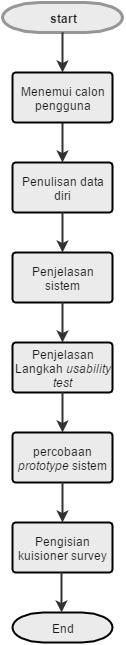
1. Pengisian kuisioner survey

Pada tahap ini, Calon pengguna akan melakukan pengisian lembaran survey sesuai dengan pengalaman yang didapatkan saat melakukan percobaan *prototype* sistem.

1. Selesai

*Usability test*  yang dilakukan selesai.

Gambar 4.74 menunjukkan *flowchart* *usability testing* pada penelitian ini.



Gambar 4.IV.75 *flowchart usability test*

Kuesioner yang dibagikan menggunakan skala Likert sebagai nilai ukur dari jawaban Calon pengguna. Skala likert pada penelitian ini terdiri dari enam tingkatan poin. Nilai terendah adalah satu (1) dan tertinggi bernilai enam (6) poin. Tingkatan nilai jawaban tersebut adalah:

1. Sangat Sangat Setuju (SSS) = 6 poin
2. Sangat Setuju (SS) = 5 poin
3. Setuju (S) = 4 poin
4. Tidak Setuju (TS) = 3 poin
5. Sangat Tidak Setuju (STS) = 2 poin
6. Sangat Sangat Tidak Setuju (SSTS) = 1 poin

Pertanyaan yang diajukan dalam *usability test* ini dapat dilihat pada tabel 4.8.

Tabel 4.8 Pertanyaan kuisioner *usability test*

|  |  |  |  |
| --- | --- | --- | --- |
| No | Kategori | Pertanyaan | X/Y |
| 1. | *Ease of use* | Tampilan layout sistem baik. | X1 |
| 2. | Penggunaan skema warna sistem baik. | X1 |
| 3. | Huruf dan teks dapat dibaca dengan jelas. | X2 |
| 4. | Kata-kata pada menu dapat dipahami. | X2 |
| 5. | Penempatan menu pada sistem mudah dijangkau. | X3 |
| 6. | Kata-kata yang dipakai untuk memberikan informasi sudah jelas. | X3 |
| 7. | *Customization* | Penyajian informasi yang ditampilkan menarik. | X4 |
| 8. | Informasi yang ditampilkan mudah diingat. | X4 |
| 9. | Pembuatan dan pembaruan profil mudah. | X5 |
| 10. | Penggantian *password* mudah. | X5 |
| 11. | *Download Delay* | Pencarian informasi mengenai penelitian diklat cepat. | X6 |
| 12. | Pencarian informasi mengenai absensi dan nilai cepat. | X6 |
| 13. | Dapat mencari informasi pelatihan berdasarkan *keyword* dengan cepat. | X7 |
| 14. | Dapat mencari informasi nilai dan absen berdasarkan pelatihan yang diadakan dengan cepat. | X7 |
| 15. | *Content* | Informasi yang diberikan sistem sesuai dengan apa yang dicari/diinginkan | X8 |
| 16. | Tidak ada informasi ganda pada sistem | X8 |
| 17. | Sistem dapat melakukan *login* atau registrasi dengan mudah | X9 |
| 18. | Informasi dan fitur yang difasilitasi sistem sudah sesuai dengan tujuan sistem. | X9 |
| 19. | Informasi yang diberikan sistem lengkap | X10 |
| 20. | Fitur dan fasilitas yang diberikan sistem sudah dapat memenuhi kebutuhan administrasi pendidikan dan pelatihan. | X10 |
| 21. | *Satisfaction* | Secara keseluruhan anda puas dengan informasi, fitur, dan kemudahan yang diberikan oleh sistem | Y1 |
| 22. | Apabila sistem telah diimplementasi, anda mempunyai keinginan untuk menggunakan sistem ini sebagai pemenuh kebutuhan informasi dan administrasi yang berhubungan dengan pelatihan di Rumah Sakit Harapan Kita Jakarta. | Y1 |

Pertanyaan kuisoner terbagi menjadi dua kelompok, X dan Y. Hubungan antara kelompok X dan Y sudah dijelaskan pada Bab III. Terdapat 10 kelompok X yang masing-masing mempunyai dua pertanyaan. Kelompok Y yang hanya mempunyai satu kelompok juga terbagi menjadi dua pertanyaaan. Untuk mendapatkan nilai X dan Y pada setiap kelompok, digunakan rumus *mean*, yaitu:

Dimana:

*: Nilai Rata-rata kelompok*

*: Jumlah nilai dari kelompok (X atau Y)*

*: Banyaknya nilai pada kelompok (X atau Y)*

Setelah mendapatkan nilai dari setiap X dan Y, dilakukan penghitungan untuk mendapatkan nilai setiap kategori. Perhitungan menggunakan rumus *mean* yang telah dijelaskan diatas. Setelah mendapatkan nilai tersebut, dilakukan pengujian hipotesa yang dijelaskan pada Bab III. Pengujian akan membandingkan satu-satu kelompok X (*ease of use, customization, download delay, content*) dengan kelompok Y (*satisfaction*).

### Hasil Survey Pengujian Aplikasi

Berdasarkan kuesioner yang dibagikan untuk menguji *usability* test terhadap *prototype* sistem, didapat data-data dan fakta-fakta. Data-data karakteristik mengenai responden ditampilkan pada tabel 4.9.

Tabel 4.9 karakteristik responden *usability test*

|  |  |  |
| --- | --- | --- |
| **Karakteristik** | **Jumlah** | **Presentase** |
| **Umur** | | |
| Dibawah 20 | 4 | 13,33% |
| 21-30 | 9 | 30% |
| 31-40 | 7 | 23,33% |
| 41-50 | 2 | 6,66% |
| Diatas 51 | 8 | 26,66% |
| **Jenis Kelamin** | | |
| Laki-laki | 18 | 60% |
| Perempuan | 12 | 40% |
| **Kelas Pengguna** | | |
| Administrator | 9 | 23,07% |
| Pengajar | 7 | 17,94% |
| Peserta | 14 | 35,89% |

Data yang ditampilkan pada Bab ini hanya terbatas pada *mean* dari kelompok X dan Y bagi masing-masing kategori/konstruk. Selain data masing-masing, juga terdapat data *mean* gabungan setiap kategori/konstruk. Untuk tabel penyebaran data dapat dilihat pada lampiran 6.

Berikut adalah tabel *mean* dari kategori/konstruk *ease of use*:

Tabel 4.10 *mean* kategori/konstruk *ease of use*

|  |  |  |  |
| --- | --- | --- | --- |
| *Ease of use* | | | |
| **X1** | **X2** | **X3** | ***Mean*** |
| 4.5 | 4.5 | 5 | 4.66666667 |
| 5 | 5 | 5 | 5 |
| 5.5 | 6 | 6 | 5.83333333 |
| 5 | 5.5 | 5.5 | 5.33333333 |
| 5 | 5 | 4.5 | 4.83333333 |
| 4.5 | 4.5 | 5 | 4.66666667 |
| 4.5 | 4.5 | 4.5 | 4.5 |
| 5.5 | 4.5 | 6 | 5.33333333 |
| 5 | 5 | 4.5 | 4.83333333 |
| 6 | 6 | 6 | 6 |
| 3.5 | 4 | 5 | 4.16666667 |
| 5 | 6 | 5 | 5.33333333 |
| 5 | 5 | 5.5 | 5.16666667 |
| 4.5 | 5 | 4.5 | 4.66666667 |
| 5 | 5 | 5 | 5 |
| 5 | 5 | 4.5 | 4.83333333 |
| 5 | 5.5 | 5.5 | 5.33333333 |
| 4.5 | 5.5 | 4.5 | 4.83333333 |
| 5 | 5 | 5.5 | 5.16666667 |
| 5 | 5 | 5 | 5 |
| 5 | 4 | 5 | 4.66666667 |
| 4.5 | 4.5 | 5 | 4.66666667 |
| 5 | 5 | 5 | 5 |
| 4.5 | 5.5 | 5 | 5 |
| 5.5 | 5 | 4.5 | 5 |
| 4.5 | 5 | 3.5 | 4.33333333 |
| 3.5 | 5 | 4.5 | 4.33333333 |
| 4 | 4.5 | 5 | 4.5 |
| 3.5 | 5 | 4.5 | 4.33333333 |
| 5.5 | 5 | 4.5 | 5 |

Berikut adalah tabel *mean* dari kategori/konstruk *customization*:

Tabel 4.11 *mean* kategori/konstruk *customization*

|  |  |  |
| --- | --- | --- |
| *Customization* | | |
| **X4** | **X5** | ***Mean*** |
| 4 | 4.5 | 4.25 |
| 5 | 5 | 5 |
| 5 | 6 | 5.5 |
| 5 | 5 | 5 |
| 5.5 | 5.5 | 5.5 |
| 5 | 5.5 | 5.25 |
| 4.5 | 4 | 4.25 |
| 6 | 5.5 | 5.75 |
| 4.5 | 5 | 4.75 |
| 6 | 6 | 6 |
| 5 | 4 | 4.5 |
| 5.5 | 5.5 | 5.5 |
| 5 | 5 | 5 |
| 4 | 4.5 | 4.25 |
| 4.5 | 5 | 4.75 |
| 4.5 | 5.5 | 5 |
| 5.5 | 5 | 5.25 |
| 5 | 5 | 5 |
| 5 | 5 | 5 |
| 5 | 5 | 5 |
| 5 | 4.5 | 4.75 |
| 5 | 5 | 5 |
| 5 | 5 | 5 |
| 4.5 | 4.5 | 4.5 |
| 4.5 | 5 | 4.75 |
| 5 | 5 | 5 |
| 4.5 | 4.5 | 4.5 |
| 5 | 4.5 | 4.75 |
| 5 | 4 | 4.5 |
| 5.5 | 5 | 5.25 |

Berikut adalah tabel *mean* dari kategori/konstruk *download delay*:

Tabel 4.12 *mean* kategori/konstruk *downlooad delay*

|  |  |  |
| --- | --- | --- |
| *Download Delay* | | |
| **X4** | **X5** | ***Mean*** |
| 4.25 | 5 | 4.625 |
| 5 | 5 | 5 |
| 5.5 | 5.5 | 5.5 |
| 5 | 5.5 | 5.25 |
| 5.5 | 4.5 | 5 |
| 5.25 | 4.5 | 4.875 |
| 4.25 | 4 | 4.125 |
| 5.75 | 5.5 | 5.625 |
| 4.75 | 5 | 4.875 |
| 6 | 6 | 6 |
| 4.5 | 4 | 4.25 |
| 5.5 | 4.5 | 5 |
| 5 | 5 | 5 |
| 4.25 | 4.5 | 4.375 |
| 4.75 | 5 | 4.875 |
| 5 | 5 | 5 |
| 5.25 | 4.5 | 4.875 |
| 5 | 5 | 5 |
| 5 | 5.5 | 5.25 |
| 5 | 5 | 5 |
| 4.75 | 5.5 | 5.125 |
| 5 | 5 | 5 |
| 5 | 5 | 5 |
| 4.5 | 4.5 | 4.5 |
| 4.75 | 4.5 | 4.625 |
| 5 | 4.5 | 4.75 |
| 4.5 | 4.5 | 4.5 |
| 4.75 | 5 | 4.875 |
| 4.5 | 5.5 | 5 |
| 5.25 | 5 | 5.125 |

Berikut adalah tabel *mean* dari kategori/konstruk *content*:

Tabel 4.13 *mean* kategori/konstruk *content*

|  |  |  |  |
| --- | --- | --- | --- |
| *Content* | | | |
| **X8** | **X9** | **X10** | ***Mean*** |
| 4.5 | 4 | 4.5 | 4.333333 |
| 5 | 4.5 | 5 | 4.833333 |
| 6 | 5.5 | 5.5 | 5.666667 |
| 5 | 5.5 | 5 | 5.166667 |
| 5 | 5.5 | 5.5 | 5.333333 |
| 5 | 5.5 | 5 | 5.166667 |
| 4 | 4 | 4 | 4 |
| 5.5 | 4.5 | 5 | 5 |
| 5 | 5 | 5 | 5 |
| 6 | 6 | 6 | 6 |
| 4.5 | 4.5 | 5 | 4.666667 |
| 5 | 5.5 | 5.5 | 5.333333 |
| 5 | 5.5 | 5 | 5.166667 |
| 5 | 4.5 | 5.5 | 5 |
| 4.5 | 5.5 | 5.5 | 5.166667 |
| 4.5 | 4 | 4.5 | 4.333333 |
| 4 | 4 | 5 | 4.333333 |
| 5 | 4.5 | 5 | 4.833333 |
| 4.5 | 5 | 5 | 4.833333 |
| 5.5 | 5 | 5 | 5.166667 |
| 5 | 5 | 5 | 5 |
| 4.5 | 5.5 | 5 | 5 |
| 5 | 5.5 | 5.5 | 5.333333 |
| 5.5 | 5.5 | 5.5 | 5.5 |
| 5 | 5.5 | 5 | 5.166667 |
| 5 | 6 | 4.5 | 5.166667 |
| 5.5 | 6 | 4 | 5.166667 |
| 4.5 | 5.5 | 4 | 4.666667 |
| 5 | 5 | 4.5 | 4.833333 |
| 5 | 5 | 4 | 4.666667 |

Berikut adalah tabel *mean* dari kategori/konstruk *Satisfaction:*

Tabel 4.14 *mean* kategori/konstruk *satisfaction*

|  |  |  |
| --- | --- | --- |
| *Satisfaction* | | |
| **Y1** | | ***Mean*** |
| 4 | 4 | 4 |
| 5 | 5 | 5 |
| 6 | 6 | 6 |
| 6 | 5 | 5.5 |
| 6 | 5 | 5.5 |
| 5 | 5 | 5 |
| 4 | 5 | 4.5 |
| 5 | 5 | 5 |
| 5 | 5 | 5 |
| 6 | 5 | 5.5 |
| 5 | 5 | 5 |
| 5 | 6 | 5.5 |
| 5 | 5 | 5 |
| 5 | 5 | 5 |
| 6 | 6 | 6 |
| 5 | 5 | 5 |
| 5 | 5 | 5 |
| 5 | 4 | 4.5 |
| 5 | 5 | 5 |
| 5 | 5 | 5 |
| 5 | 4 | 4.5 |
| 5 | 5 | 5 |
| 5 | 5 | 5 |
| 4 | 4 | 4 |
| 6 | 6 | 6 |
| 5 | 5 | 5 |
| 5 | 6 | 5.5 |
| 4 | 5 | 4.5 |
| 4 | 6 | 5 |
| 4 | 4 | 4 |

:

Berikut adalah tabel *mean* dari setiap kategori/konstruk dengan dua desimal*:*

Tabel 4.15 *mean* setiap kategori/konstruk

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | ***Mean* semua Kategori/Konstruk** | | | | | |
| *No.* | *Ease of use* | *Customization* | *Download Delay* | *Content* |  | *Satisfaction* |
| 1 | 4.67 | 4.25 | 4.63 | 4.33 |  | 4.00 |
| 2 | 5.00 | 5.00 | 5.00 | 4.83 |  | 5.00 |
| 3 | 5.83 | 5.50 | 5.50 | 5.67 |  | 6.00 |
| 4 | 5.33 | 5.00 | 5.25 | 5.17 |  | 5.50 |
| 5 | 4.83 | 5.50 | 5.00 | 5.33 |  | 5.50 |
| 6 | 4.67 | 5.25 | 4.88 | 5.17 |  | 5.00 |
| 7 | 4.50 | 4.25 | 4.13 | 4.00 |  | 4.50 |
| 8 | 5.33 | 5.75 | 5.63 | 5.00 |  | 5.00 |
| 9 | 4.83 | 4.75 | 4.88 | 5.00 |  | 5.00 |
| 10 | 6.00 | 6.00 | 6.00 | 6.00 |  | 5.50 |
| 11 | 4.17 | 4.50 | 4.25 | 4.67 |  | 5.00 |
| 12 | 5.33 | 5.50 | 5.00 | 5.33 |  | 5.50 |
| 13 | 5.17 | 5.00 | 5.00 | 5.17 |  | 5.00 |
| 14 | 4.67 | 4.25 | 4.38 | 5.00 |  | 5.00 |
| 15 | 5.00 | 4.75 | 4.88 | 5.17 |  | 6.00 |
| 16 | 4.83 | 5.00 | 5.00 | 4.33 |  | 5.00 |
| 17 | 5.33 | 5.25 | 4.88 | 4.33 |  | 5.00 |
| 18 | 4.83 | 5.00 | 5.00 | 4.83 |  | 4.50 |
| 19 | 5.17 | 5.00 | 5.25 | 4.83 |  | 5.00 |
| 20 | 5.00 | 5.00 | 5.00 | 5.17 |  | 5.00 |
| 21 | 4.67 | 4.75 | 5.13 | 5.00 |  | 4.50 |
| 22 | 4.67 | 5.00 | 5.00 | 5.00 |  | 5.00 |
| 23 | 5.00 | 5.00 | 5.00 | 5.33 |  | 5.00 |
| 24 | 5.00 | 4.50 | 4.50 | 5.50 |  | 4.00 |
| 25 | 5.00 | 4.75 | 4.63 | 5.17 |  | 6.00 |
| 26 | 4.33 | 5.00 | 4.75 | 5.17 |  | 5.00 |
| 27 | 4.33 | 4.50 | 4.50 | 5.17 |  | 5.50 |
| 28 | 4.50 | 4.75 | 4.88 | 4.67 |  | 4.50 |
| 29 | 4.33 | 4.50 | 5.00 | 4.83 |  | 5.00 |
| 30 | 5.00 | 5.25 | 5.13 | 4.67 |  | 4.00 |

Setelah mendapatkan *mean* dari setiap konstruk, dilakukan tes pengujian hipotesa. Data *mean* pada tabel diatas dianalisis berdasarkan hipotesa hubungan antara kelompok X (*ease of use, customization, download delay,* dan *content*) dengan kelompok Y (*satisfaction*) sesuai dengan kerangka berfikir Bab III. Hipotesis yang diuji adalah:

1. Hipotesis pertama (H1): *Ease of use* pada *prototype* sistem memiliki pengaruh positif langsung terhadap *satisfaction* responden.
2. Hipotesis kedua (H2): *Customization* pada *prototype* sistem memiliki pengaruh positif langsung terhadap *satisfaction* responden.
3. Hipotesis ketiga (H3): *Download delay* pada *prototype* sistem memiliki pengaruh positif langsung terhadap *satisfaction* responden.
4. Hipotesis keempat (H4): *Content* pada *prototype* sistem memberikan pengaruh positif langsung terhadap *satisfaction* responden.

Sebelum melakukan pengujian, dilakukan uji normalitas. Uji normalitas berguna untuk mengetahui apakah data yang didapat terdistribusi normal atau tidak. Hal ini berpengaruh pada metode pengujian hipotesa yang digunakan.

### Uji normalitas

Uji normalitas pada data yang didapatkan dari survei kuisioner dilakukan dengan melakukan tes Shapiro-Wilk [25]. Tes tersebut dipilih karena jumlah sampel yang lebih kecil dari 50 (*n<*50). Pengujian dilakukan menggunakan *tools* IBM SPSS Statistics. Berikut adalah tabel hasil uji normalitas setiap kategori/konstruk.

Tabel 4.16 uji normalitas atas kategori/konstruk

|  |  |  |  |
| --- | --- | --- | --- |
| **Tests of Normality** | | | |
|  | Shapiro-Wilk | | |
| Statistics | df | Sig. |
| easeofuse | .953 | 30 | .209 |
| customization | .949 | 30 | .157 |
| downloaddelay | .937 | 30 | .077 |
| content | .960 | 30 | .311 |
| satisfaction | .890 | 30 | .005 |

Ditentukan nilai *α* = 0.05 dan *null hypothesis (h0)* yang menyebutkan bahwa data terdistribusi normal sebagai dasar pengambil keputusan dalam uji normalitas ini. Berdasarkan tabel 4.16, diambil fakta-fakta yang berhubungan dengan kategori/konstruk sebagai berikut:

1. Nilai “*Sig.*” yang terdapat pada tabel menunjukkan *p-value* [35]*.*
2. Data dari kategori/konstruk *satisfaction* tidak terdistribusi dengan normal. Berdasarkan perbandingan nilai  *α* dengan *p-value,* dimana :

Memberikan keluaran (*output*) bahwa *null hypothesis (h0)* ditolak. QQ-plot

1. Data dari kategori/konstruk *ease of use* terdistribusi dengan normal. Berdasarkan perbandingan nilai  *α* dengan *p-value,* dimana :

Memberikan keluaran (*output*) bahwa *null hypothesis (h0)* diterima.

1. Data dari kategori/konstruk *customization* terdistribusi dengan normal. Berdasarkan perbandingan nilai  *α* dengan *p-value,* dimana :

Memberikan keluaran (*output*) bahwa *null hypothesis (h0)* diterima.

1. Data dari kategori/konstruk *download delay* terdistribusi dengan normal. Berdasarkan perbandingan nilai  *α* dengan *p-value,* dimana :

Memberikan keluaran (*output*) bahwa *null hypothesis (h0)* diterima.

1. Data dari kategori/konstruk *content* terdistribusi dengan normal. Berdasarkan perbandingan nilai  *α* dengan *p-value,* dimana :

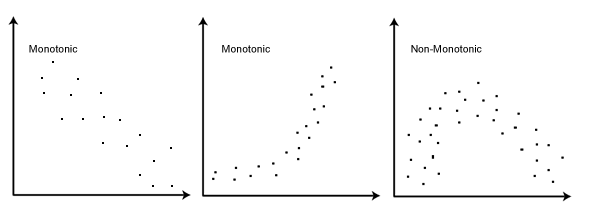
Memberikan keluaran (*output*) bahwa *null hypothesis (h0)* diterima.

Berdasarkan fakta-fakta diatas, dapat disimpulkan bahwa selain *satisfaction*, semua data dari kategori/konstruk terdistribusi normal. Dikarenakan terdapat data yang tidak terdistribusi normal, pengujian korelasi dilakukan dengan menggunakan tes Spearman, yang merupakan uji korelasi hipotesa.

### Uji Korelasi Hipotesis

Dalam melakukan uji korelasi dengan menggunakan tes Spearman, dibutuhkan pengecekan asumsi-asumsi agar dapat memberikan hasil yang valid [36]. Asumsi-asumsi tersebut adalah:

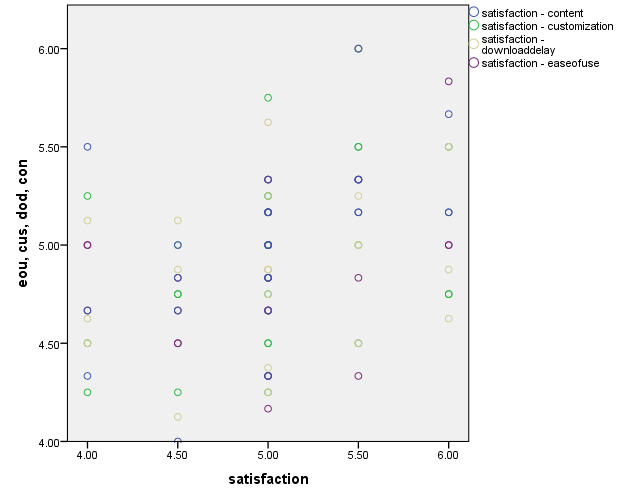
1. Variabel yang diukur harus berbentuk ordinal, interval, atau rasio.
2. Variabel harus mempunyai relasi monotonik satu sama lain. Relasi monotonik eksis apabila nilai variabel naik bersama-sama atau ketika salah satu nilai variabel naik, nilai variabel yang lain turun. Penggambaran dengan grafik *scatterplot* dapat melihat ada tidaknya relasi monotonik antar variabel.



Gambar 4.IV.76Diagram *scatterplot* relasi monotonik dan non-monotonik

*(sumber: https://statistics.laerd.com/spss-tutorials/spearmans-rank-order-correlation-using-spss-statistics.php )*

Pada pengecekan asumsi tersebut, data yang didapat valid untuk kedua asumsi. Untuk asumsi pertama, data valid karena berupa nilai yang berasal dari skala Likert yang berbentuk ordinal. Berdasarkan *scatterplot* yang dibuat untuk pengecekan asumsi kedua, Ditemukan bahwa kelima kategori/konstruk mempunyai nilai yang saling naik. Hal ini membuktikan bahwa asumsi kedua, yaitu adanya relasi monotik valid. Berikut adalah grafik *scatterplot* perbandingan antara kelompok konstruk X dengan Y. terlihat pada grafik bahwa nilai *scatterplot* semuanya merangkak naik. Grafik dibuat menggunakan IBM SPSS Statistics.



Gambar 4.IV.77 Diagram *scatterplot* konstruk X dengan konstruk Y. Terlihat adanya relasi monotonik

Setelah mendapatkan kesimpulan data valid pada uji asumsi, pengujian korelasi hipotesa dilakukan. Pengujian dilakukan dengan membandingkan satu-satu antara kelompok X dengan Y. uji korelasi yang dilakukan adalah:

1. Korelasi antara *ease of use* dengan *satisfaction.*
2. Korelasi antara *customization* dengan *satisfaction*.
3. Korelasi antara *download delay* dengan *satisfaction.*
4. Korelasi antara *content* dengan *satisfaction.*

#### Hubungan antara Ease of Use dengan Satisfaction

Hubungan antara *ease of use* dengan *satisfaction* pada *prototype* sistem yang dianalisis dengan uji korelasi Spearman mempunyai tahap-tahap sebagai berikut:

1. Menentukan hipotesis, dimana:

, Tidak ada pengaruh positif antara *ease of use* dengan *satisfaction* respondenpada *prototype* sistem.

, *Ease of use* pada *prototype* sistem memiliki pengaruh positif langsung terhadap *satisfaction* responden.

1. Menentukan kriteria pengambilan keputusan dengan membandingkan nilai *rs* hitung dengan *rs* tabel, yaitu:

* Nilai
* Nilai

Dalam menentukan *rs* digunakan uji satu sisi:



1. Uji korelasi peringkat Spearman.

Pengujian korelasi peringkat Spearman diolah dengan menggunakan IBM SPSS *Statistics* terhadap data yang diperoleh menghasilkan hasil pada tabel IV.12 sebagai berikut:

Tabel 4.17 Hasil uji korelasi Spearman antara *ease of use dengan satisfaction*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Correlations** | | | | |
|  | | | easeofuse | satisfaction |
| Spearman's rho | easeofuse | Correlation Coefficient | 1.000 | .373\* |
| Sig. (2-tailed) | . | .042 |
| N | 30 | 30 |
| satisfaction | Correlation Coefficient | .373\* | 1.000 |
| Sig. (2-tailed) | .042 | . |
| N | 30 | 30 |
| \*. Correlation is significant at the 0.05 level (2-tailed). | | | | |

Berdasarkan tabel 4.17 diperoleh nilai korelasi Spearman (*rs* hitung) antara *ease of use* dengan *satisfaction* respondenpada *prototype* sistem sebesar 0,373. Dapat diketahui bahwa maka *H0*ditolak. Hal ini menunjukkan bahwa terdapat hubungan positif antara *ease of use* dengan *satisfaction* pada *prototype* sistem. Tingkat Signifikasi sebesar 0,042 < α maka *Ho*ditolak. Hal ini menunjukkan ada hubungan yang signifikan pada taraf nyata 0,05.

#### Hubungan antara Customization dengan Satisfaction

Hubungan antara *customization* dengan *satisfaction* pada *prototype* sistem yang dianalisis dengan uji korelasi Spearman mempunyai tahap-tahap sebagai berikut:

1. Menentukan hipotesis, dimana:

, Tidak ada pengaruh positif antara *customization* dengan *satisfaction* respondenpada *prototype* sistem.

, *Customization* pada *prototype* sistem memiliki pengaruh positif langsung terhadap *satisfaction* responden.

1. Menentukan kriteria pengambilan keputusan dengan membandingkan nilai *rs* hitung dengan *rs* tabel, yaitu:

* Nilai
* Nilai

Dalam menentukan *rs* digunakan uji satu sisi:



1. Uji korelasi peringkat Spearman.

Pengujian korelasi peringkat Spearman diolah dengan menggunakan IBM SPSS *Statistics* terhadap data yang diperoleh menghasilkan hasil pada tabel IV.13 sebagai berikut:

Tabel 4.18 Hasil uji korelasi Spearman antara *customization* dengan *satisfaction*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Correlations** | | | | |
|  | | | customization | satisfaction |
| Spearman's rho | customization | Correlation Coefficient | 1.000 | .348 |
| Sig. (2-tailed) | . | .060 |
| N | 30 | 30 |
| satisfaction | Correlation Coefficient | .348 | 1.000 |
| Sig. (2-tailed) | .060 | . |
| N | 30 | 30 |

Berdasarkan tabel 4.18 diperoleh nilai korelasi Spearman (*rs* hitung) antara *customization* dengan *satisfaction* pada *prototype* sistem sebesar 0,348. Dapat diketahui bahwa maka *H0*diterima. Hal ini menunjukkan bahwa tidak terdapat hubungan positif antara *ease of use* dengan *satisfaction* pada *prototype* sistem. Tingkat Signifikasi sebesar 0,060 > α maka *Ho*diterima. Hal ini menunjukkan tidak ada hubungan yang signifikan secara statistik.

#### Hubungan antara Download Delay dengan Satisfaction

Hubungan antara *download delay* dengan *satisfaction* pada *prototype* sistem yang dianalisis dengan uji korelasi Spearman mempunyai tahap-tahap sebagai berikut:

1. Menentukan hipotesis, dimana:

, Tidak ada pengaruh positif antara *download delay* dengan *satisfaction* respondenpada *prototype* sistem.

, *download delay* pada *prototype* sistem memiliki pengaruh positif langsung terhadap *satisfaction* responden.

1. Menentukan kriteria pengambilan keputusan dengan membandingkan nilai *rs* hitung dengan *rs* tabel, yaitu:

* Nilai
* Nilai

Dalam menentukan *rs* digunakan uji satu sisi:



1. Uji korelasi peringkat Spearman.

Pengujian korelasi peringkat Spearman diolah dengan menggunakan IBM SPSS *Statistics* terhadap data yang diperoleh menghasilkan hasil pada tabel IV.14 sebagai berikut:

Tabel 4.19 Hasil uji korelasi Spearman antara *download delay* dengan *satisfaction*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Correlations** | | | | |
|  | | | downloaddelay | satisfaction |
| Spearman's rho | downloaddelay | Correlation Coefficient | 1.000 | .196 |
| Sig. (2-tailed) | . | .300 |
| N | 30 | 30 |
| satisfaction | Correlation Coefficient | .196 | 1.000 |
| Sig. (2-tailed) | .300 | . |
| N | 30 | 30 |

Berdasarkan tabel 4.19 diperoleh nilai korelasi Spearman (*rs* hitung) antara *download delay* dengan *satisfaction* pada *prototype* sistem sebesar 0,196. Dapat diketahui bahwa maka *H0*diterima. Hal ini menunjukkan bahwa tidak terdapat hubungan positif antara *download delay* dengan *satisfaction* pada *prototype* sistem. Tingkat Signifikasi sebesar 0,300 > α maka *Ho*diterima. Hal ini menunjukkan tidak ada hubungan yang signifikan secara statistik.

#### Hubungan antara *Content* dengan *Satisfaction*

Hubungan antara *content* dengan *satisfaction* pada *prototype* sistem yang dianalisis dengan uji korelasi Spearman mempunyai tahap-tahap sebagai berikut:

1. Menentukan hipotesis, dimana:

, Tidak ada pengaruh positif antara *content* dengan *satisfaction* respondenpada *prototype* sistem.

, *Content* pada *prototype* sistem memiliki pengaruh positif langsung terhadap *satisfaction* responden.

1. Menentukan kriteria pengambilan keputusan dengan membandingkan nilai *rs* hitung dengan *rs* tabel, yaitu:\

* Nilai
* Nilai

Dalam menentukan *rs* digunakan uji satu sisi:



1. Uji korelasi peringkat Spearman.

Pengujian korelasi peringkat Spearman diolah dengan menggunakan IBM SPSS *Statistics* terhadap data yang diperoleh menghasilkan hasil pada tabel IV.15 sebagai berikut:

Tabel 4.20 Hasil uji korelasi Spearman antara *content* dengan *satisfaction*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Correlations** | | | | |
|  | | | content | satisfaction |
| Spearman's rho | content | Correlation Coefficient | 1.000 | .585\*\* |
| Sig. (2-tailed) | . | .001 |
| N | 30 | 30 |
| satisfaction | Correlation Coefficient | .585\*\* | 1.000 |
| Sig. (2-tailed) | .001 | . |
| N | 30 | 30 |
| \*\*. Correlation is significant at the 0.01 level (2-tailed). | | | | |

Berdasarkan tabel 4.20 diperoleh nilai korelasi Spearman (*rs* hitung) antara *ease of use* dengan *satisfaction* pada *prototype* sistem sebesar 0,585. Dapat diketahui bahwa maka *H0*ditolak. Hal ini menunjukkan bahwa terdapat hubungan positif antara *content* dengan *satisfaction* pada *prototype* sistem. Tingkat Signifikasi sebesar 0,001 < α maka *Ho*ditolak. Hal ini menunjukkan ada hubungan yang signifikan pada taraf nyata 0,01.

## Pembahasan usability test

Berdasarkan hasil uji korelasi yang telah dilakukan, maka dapat diketahui bahwa terdapat pengaruh positif antara *ease of use* dan *content* terhadap *satisfaction* responden. Sedangkan *customization* dan *download delay* tidak mempunyai pengaruh positif terhadap *satisfaction* dari responden.

Pengaruh positif pada *ease of use* menunjukkan bahwa semakin tinggi kemudahan dalam penggunaan sistem, maka semakin tinggi juga kepuasan. Apabila dihubungkan dengan data skoring hasil kuisioner, Hal ini terjadi karena responden menganggap *prototype* sistem mempunyai *layout* dan menu yang mudah dipahami.Variabel *content* yang juga memberikan hasil positif menunjukkan bahwa semakin banyak informasi yang diberikan atau didapatkan pada sistem, maka semakin tinggi kepuasan. Apabila dihubungkan dengan data skoring hasil kuisioner, Hal ini terjadi karena responden menganggap *prototype* sistem mempunyai informasi yang sesuai. Hal ini sejalan dengan tujuan rancang bangun sistem yaitu memberikan informasi administrasi terkait dengan Divisi Diklat Rumah Sakit Harapan Kita Jakarta.

Tidak semua variabel memberikan pengaruh positif terhadap *satisfaction* responden pada *prototype* sistem. Variabel *customization* tidak memberikan pengaruh positif menunjukkan bahwa semakin tinggi atau banyak fitur *customization* tidak akan berpengaruh pada kepuasan pengguna. Hal ini juga sejalan dengan variabel *download delay* yang menunjukkan tidak ada pengaruh terhadap kepuasan apabila kecepatan dari pencarian informasi bertambah.

# KESIMPULAN DAN SARAN

## Kesimpulan

Berdasarkan hasil penelitian rancang bangun dan *usability test* sistem administrasi berbasis web pada Divisi Diklat Rumah Sakit Harapan Kita Jakarta maka dapat disimpulkan beberapa hal sebagai berikut:

1. Proses perancangan sistem dapat dilakukan dengan menerapkan metode *prototyping.* Dengan menerapkan metode *prototoyping* aplikasi yang mempunyai cakupan yang cukup luas dapat dirancang berdasarkan *requirement* yang didapatkan serta *feedback* dari calon pengguna. Juga dengan menerapkan metode *prototyping* pengembangan *prototype* sistem dapat terus berjalan mengikuti perkembangan kebutuhan pengguna.
2. Pengembangan sistem administrasi Divisi Diklat Harapan Kita Jakarta yang dilakukan dengan metode *prototyping* mengikuti satu alur tahap *(cycle)* metode yaitu *communication, planning & design, prototyping* dan *feedback.* Pertama-tama *communication* dilakukan untuk mendapatkan kebutuhan yang diharapkan calon pengguna. Pada tahap ini digunakan wawancara, dokumen, dan observasi sebagai alat untuk mendapatkan informasi. Selanjutnya, dilakukan dengan tahap *planning & design* yaitu menganalisa kebutuhan calon pengguna menjadi sebuah rencana dan desain awal dari *prototype system*. Tahap ini menghasilkan analisa rancangan yang dijelaskan dengan tabel, *use case diagram, activity diagram,* serta gambar rancangan awal *layout* sistem. Setelah analisa rancangan dibuat, dimulailah tahapan selanjutnya yaitu *prototyping*. Pada tahapan *prototyping* dilakukan pengembangan *prototype* sistem berdasarkan analisa rancangan. *Prototype* sistem dibuat menggunakan Bahasa pemrograman PHP dan *database* MySQL. Yii juga digunakan sebagai *framework* yang mengaplikasikan pengembangan sistem berbasis MVC (*Model-View-Controller*). Pada tahap terakhir, yaitu *feedback*, dilakukan proses *usability test* pada *prototype* sistem yang dikembangkan. Pada tahap ini, *usability test* menggunakan *self-reported metric data*. Pengukuran dilakukan dengan menggunakan survey kuisioner berskala Likert. Kategori/konstruk yang digunakan pada *usability test* adalah *ease of use, customization, download delay, content,*dan *satisfaction.* Tujuan *usability test* pada tahap ini adalah untuk mengetahui apakah terdapat pengaruh positif antara *satisfaction* dengan kategori/konstruk lainnya.
3. *Usability test* yang dilakukan pada proses *feedback* dilakukan dengan tahapan mendatangi responden, pengisian data diri, penjelasan sistem, pengujian sistem, dan pengisian kuisioner. Setelah mendapatkan data dari *usability test,* dilakukan uji normalitas dan korelasi untuk mengetahui pengaruh positif antara *satisfaction* dengan kategori/konstruk lainnya. Pengujian pertama dilakukan dengan uji normalitas Saphiro-Wilk. Pengujian ini menunjukkan data yang tidak terdistribusi normal. Karena data tidak terdistribusi normal, pengujian korelasi dilakukan dengan uji korelasi Spearman. Uji korelasi tersebut dipilih karena data pada variabel tidak terdistribusi normal.
4. Hasil uji korelasi yang dilakukan dengan menggunakan uji korelasi Spearman menghasilkan kesimpulan bahwa tidak semua variabel mempunyai pengaruh positif terhadap kategori/konstruk *satisfaction.* Hanya kategori/konstruk *ease of use* dan *content* yang mempunyai pengaruh positif pada kepuasan responden. Sedangkan *customization* dan *download delay* tidak mempunyai pengaruh pada kepuasan responden.

## Saran

Berdasarkan pembahasan dan kesimpulan yang telah dikemukakan, dapat diajukan beberapa saran untuk pengembangan penelitian sistem administrasi berbasis *web* lebih lanjut di masa depan, saran-saran ini antara lain:

1. Perancangan *prototype* sistem administrasi dapat dikembangkan lebih lanjut sampai pada versi final dari sistem tersebut.
2. Penambahan fitur sistem untuk pembayaran otomatis dengan menggunakan kerja sama *e-banking*. Hal ini bertujuan untuk memudahkan pembayaran oleh pengguna.
3. Pengembangan desain aplikasi agar lebih dititik beratkan pada *platform mobile*. Hal ini bertujuan untuk memudahkan akses kepada sistem administrasi oleh pengguna.
4. Pengembangan sistem administrasi disarankan lebih mempertimbangkan kostumisasi dan kecepatan mendapatkan informasi. Hal ini bertujuan agar memberikan pengaruh positif terhadap kepuasan pengguna.

DAFTAR PUSTAKA

|  |  |  |
| --- | --- | --- |
| [1] |  | REPUBLIK INDONESIA, “Peraturan Pemerintah Republik Indonesia (PP) Nomor 101 Tahun 2000 (101/2000) Tentang Pendidikan dan Pelatihan Jabatan Pegawai Negeri Sipil,” 10 November 2000. [Online]. Available: http://psdg.bgl.esdm.go.id/kepmen\_pp\_uu/pp\_101\_2000.pdf. |
| [2] |  | D. A. Norman, “Design Rules Based on Analyses of Human Error,” *Communications of the ACM,* pp. 254-258, 1983. |
| [3] |  | J. Lemantara, “Rancang Bangun Sistem Pengolahan Administrasi Berbasis Web Pada Kemahasiswaan STIKOM Surabaya,” Sekolah Tinggi Manajemen Informatika & Teknik Komputer, Surabaya, 2009. |
| [4] |  | N. Ismi, “Rancang Bangun Sistem Informasi Administrasi Penerimaan Mahasiswa Baru Berbasis Web Universitas Mataram Menggunakan PHP dan MySQL,” Universitas Islam Negeri Maulana Malik Ibrahim, Malang, 2010. |
| [5] |  | D. B. Napitupulu, “Perancangan Sistem Informasi Pelatihan Koperasi Uji Mutu Berbasis Web,” *Jurnal Sistem Informasi MTI-UI, Volume 4, Nomor 1,* pp. 67-71, 2008. |
| [6] |  | S. Ziemer, “An Architecture for Web Applications Essay in DIF 8914 Distributed Information Systems,” 2002. |
| [7] |  | D. T. Green dan J. M. Pearson, “The Examination of two web site usability instruments for use in B2C e-Commerce Organizations,” *Journal of Computer Information Systems,* 2009. |
| [8] |  | S. H. Prayoga dan D. I. Sensuse, “Analisis Usability Pada Aplikasi Berbasis Web Dengan Mengadopsi Model Kepuasan Pengguna,” *Jurnal Sistem Informasi MTI UI, Volume 6, Nomor 1,* pp. 64-73, 2010. |
| [9] |  | U. D. o. H. &. H. Services, “Usability Basics,” 10 October 2014. [Online]. Available: http://www.usability.gov. |
| [10] |  | H. A. Simon, V. A. Thompson dan D. W. Smithburg, Public Administration, New Jersey: Transaction Publishers, 1991. |
| [11] |  | BMKG, “Pengertian Administrasi,” 21 April 2012. [Online]. Available: http://data.bmkg.go.id/share/Dokumen/modul%201-4.pdf. |
| [12] |  | Kusdiantoro, “Analisis Usability Website Akademik Perguruan Tinggi Di Indonesia Menggunakan Metode Promethee, Vikor, Dan Electree,” *Fakultas Teknik Universitas Negeri Yogyakarta,* 2012. |
| [13] |  | B. Battleson, A. Booth dan J. Weintrop, “Usability Testing of an Academic Library Website: A Case Study,” *The Journal of Academic Librarianship,* pp. 188-198, 2001. |
| [14] |  | J. T. Marchewka, Information Technology Project Management, New Jersey: Wiley, 2010. |
| [15] |  | R. S. Pressman, Software Engineering : A Practitioner's Approach, New York: McGraw-Hill, 2010. |
| [16] |  | L. Ulman, The Yii Book, Self-Published, 2014. |
| [17] |  | T. C. Shan dan W. W. Hua, “Taxonomy of Java Web Application Frameworks,” *IEEE International Conference on e-Business Engineering,* 2006. |
| [18] |  | C. Manger, T. Trejderowski dan J. Paduch, “Advantages and disadvantages of framework programming with reference to Yii php framework, gideon .net framework and other modern frameworks,” *Studia Informatica,* pp. 119-137, 2010. |
| [19] |  | Badiyanto, Buku Pintar Framework Yii, Yogyakarta: Mediakom, 2013. |
| [20] |  | T. A. Powell, Web Design: The Complete Reference, Berkeley, CA: Osborne/McGraw-Hill, 2000. |
| [21] |  | C. C. Whitehead, “Evaluating Web Page and Web Site Usability,” dalam *ACM-SE 44 Proceedings of the 44th annual Southeast regional conference*, New York, 2006. |
| [22] |  | J. W. Palmer, “Web Site Usability, Design, and Performance Metrics,” *Information Systems Research,* p. 151, 2002. |
| [23] |  | N. Mohd Razali, “Test of Normality: A Power Comparison of Kolmogorov-Smirnov, Anderson-Darling, Shapiro-Wilk And Lilliefors Tests,” *Universiti Teknologi MARA,* 2009. |
| [24] |  | R. Geary, “Testing for Normality,” *Biometrika,* pp. 209-242, 1947. |
| [25] |  | S. S. Shapiro dan M. B. Wilk, “An Analysis of Variance Test for Normality (Complete Samples),” *Biometrika,* pp. 591-611, 1965. |
| [26] |  | M. Mendes dan A. Pala, “Type I Error Rate and Power of Three Normality Tests,” *Pakistan Journal of Information and Technology 2 ,* pp. 135-139, 2003. |
| [27] |  | Biddle, “The Correlation Coefficient,” 2015. [Online]. Available: http://www.biddle.com/documents/bcg\_comp\_chapter2.pdf. |
| [28] |  | J. Hauke dan T. Kossowski , “Comparison of Values of Pearson's And Spearman's Correlation Coefficients On The Same Sets of Data,” *QUAESTIONES GEOGRAPHICAE,* pp. 87-93, 2011. |
| [29] |  | A. Lehman, N. O'Rourke, L. Hatcher dan E. J. Stepanski, JMP for Basic Univariate and Multivariate Statistics, North Carolina: SAS, 2005. |
| [30] |  | T. Tullis dan B. Albert, Measuring The User Experience : Collecting, Analyzing, and Presenting Usability Metrics (2nd edition), Massachusetts: Morgan Kauffmann, 2013. |
| [31] |  | U. Sekaran, Research Methods for Business, a Skill-Building Approach. Fourth Edition., Canada: Wiley & Sons, 2006. |
| [32] |  | N. Kaya, “Relationship between color and emotion: a study of college student,” vol. 38, no. 3. 396-405, 2004. |
| [33] |  | R. M. Rider, “Color Psychology and Graphic Design Applications,” *Senior Honors Theses,* 2010. |
| [34] |  | L. Faulkner, “Beyond the Five-user Assumption: Benefits of Increased Sample Sizes in Usability Testing,” *Behavior Research Methods, Instruments & Computers,* pp. 379-383, 2003. |
| [35] |  | A. Field, Discovering Statistics Using IBM SPSS Statistics (4th. Edition), London: SAGE, 2013. |
| [36] |  | Laerd, “Spearman's Rank-Order Correlation using SPSS Statistics,” 2013. [Online]. Available: https://statistics.laerd.com/spss-tutorials/spearmans-rank-order-correlation-using-spss-statistics.php. |
| [37] |  | J. M. Wilson, “Gantt charts: A centenary aprreciation,” *European Journal of Operational Research,* pp. 430-437, 2003. |
| [38] |  | S. Vivekanand dan M. Mistry, “Impact of AJAX In Web Applications,” *International Journal of Advanced Engineering Technology,* p. 30, 2012. |
| [39] |  | R. S. Pressman, Software Engineering: A Practitioner's Approach, New York: McGraw-Hill, 2010. |
| [40] |  | K. C. Laudon dan J. P. Laudon, Management Information Systems, New Jersey: Pearson, 2010. |
| [41] |  | L. Kholisoh, Teori dan Soal Statistika dan Probabilitas, Depok: Gunadarma, 1994. |
| [42] |  | D. D. Dvorski, “Installing, Configuring, and Developing With XAMPP,” March 2007. [Online]. Available: http://dalibor.dvorski.net/downloads/docs/InstallingConfiguringDevelopingWithXAMPP.pdf. |
| [43] |  | R. Dickstein dan V. Mills , “Usability testing at the University of Arizona Library: How to let the users in on the design,” *Information Technology and Libraries, Volume 19, Issue 3,* pp. 141-151, 2000. |
| [44] |  | J. Bacon, Practical PHP and MySQL, Boston: Prentince Hall, 2007. |
| [45] |  | L. Babin, Beginning Ajax With PHP : From Novice to Professional, New York: Apress, 2007. |
| [46] |  | w3.org, “HTML & CSS,” [Online]. Available: http://www.w3.org/standards/webdesign/htmlcss. |

LAMPIRAN