

BAB II TINJAUAN PUSTAKA

2.1 Pengertian Bimbingan Konseling

Menurut [4], bimbingan konseling adalah salah satu komponen yang penting dalam proses pendidikan sebagai suatu sistem. Bimbingan merupakan bantuan kepada individu dalam menghadapi persoalan-persoalan yang dapat timbul dalam hidupnya. Sedangkan konsep konseling pengertian konseling sebagai upaya membantu individu melalui proses interaksi yang bersifat pribadi antara konselor dan konseling agar konseli mampu memahami diri dan lingkungannya, mampu membuat keputusan dan menentukan tujuan berdasarkan nilai yang diyakininya, sehingga konseli merasa bahagia dan efektif perilakunya. Bimbingan membantu individu mencapai perkembangan diri secara optimal sebagai makhluk sosial.

2.2 Kesehatan Mental

Dalam buku *Mental Hygiene*, Kesehatan mental berkaitan dengan beberapa hal. Pertama, bagaimana seseorang memikirkan, merasakan dan menjalani keseharian dalam kehidupan; Kedua, bagaimana seseorang memandang diri sendiri dan orang lain; dan Ketiga, bagaimana seseorang mengevaluasi berbagai alternatif solusi dan bagaimana mengambil keputusan terhadap keadaan yang dihadapi [5].

Kesehatan mental merujuk pada kesehatan seluruh aspek perkembangan seseorang, baik fisik maupun psikis. Kesehatan mental juga meliputi upaya-upaya dalam mengatasi stress, ketidakmampuan dalam menyesuaikan diri, bagaimana berhubungan dengan orang lain, serta berkaitan dengan pengambilan keputusan. Kesehatan mental tiap individu berbeda dan mengalami dinamisasi dalam perkembangannya. Karena pada hakikatnya manusia dihadapkan pada kondisi dimana ia harus menyelesaikannya dengan beragam alternatif pemecahannya. Adakalanya, tidak sedikit orang yang pada waktu tertentu mengalami masalah-masalah kesehatan mental dalam kehidupannya.

Menurut APA [6], *Mental illness* adalah kondisi kesehatan yang melibatkan perubahan emosi, pemikiran, atau perilaku (atau kombinasi dari ini). *Mental illness*

dikaitkan dengan kesulitan dan/atau masalah yang mengganggu keberfungsian dalam kegiatan sosial, pekerjaan, atau aktivitas keluarga. Artinya, orang yang memiliki *mental illness* secara signifikan memengaruhi cara mereka merasa, berpikir, berperilaku, dan berinteraksi dengan orang lain. Mental illness bukanlah hal yang memalukan melainkan sebuah kondisi medis, sama seperti penyakit jantung atau diabetes. Berikut gangguan kesehatan mental diantaranya adalah :

a. Gangguan Depresi

Depresi adalah jenis gangguan mental yang paling sering terjadi. Jenis gangguan mental ini ditandai dengan kesedihan yang terlalu lama hingga penderitanya bisa merasa putus asa, bersalah, tidak berharga, tidak termotivasi, hingga berbagai keluhan fisik yang tidak jelas penyebabnya.

b. Gangguan Bipolar

Gangguan bipolar adalah penyakit mental yang ditandai dengan perubahan suasana hati yang tidak biasa. Perubahan ini bisa terjadi dari sangat bahagia menjadi sangat sedih dan putus asa.

c. Gangguan PTSD

Gangguan stres pascatrauma (PTSD) adalah gangguan kesehatan mental yang terjadi setelah seseorang mengalami atau melihat peristiwa traumatis. Peristiwa ini bisa berupa kejadian yang mengancam jiwa, seperti bencana alam, kecelakaan mobil, atau kekerasan seksual, atau pengalaman lain yang membuatnya trauma.

d. Gangguan Kecemasan

Gangguan kecemasan adalah perasaan cemas yang sangat kuat, berlebihan, serta berlangsung lama dan bisa memburuk seiring waktu, hingga terasa sangat membebani. Jenis gangguan ini meliputi serangan panik, gangguan obsesif kompulsif (OCD), dan fobia.

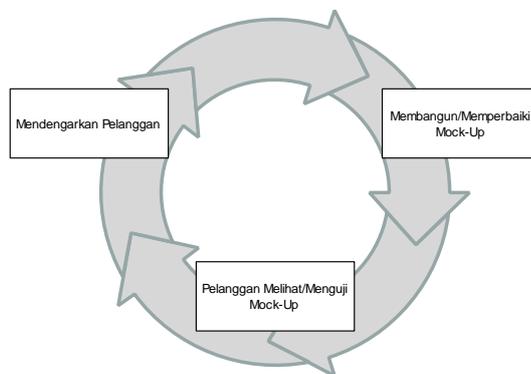
e. Gangguan Skizofrenia

Skizofrenia adalah jenis *mental illness* yang terjadi saat penderitanya tidak mampu membedakan kenyataan dan pikirannya sendiri. Gangguan ini bisa menyebabkan penderitanya mengalami pemikiran yang tidak realistis, halusinasi, dan perubahan perilaku.

2.3 Metode Pengembangan Sistem Prototipe

Prototipe merupakan metode yang efektif dalam merancang perangkat lunak. Prototipe dimulai dengan mengumpulkan kebutuhan yang akan di rancang. Pengembang mendefinisikan objek keseluruhan dari perangkat lunak, mengidentifikasi segala aktifitas yang diketahui dan kemudian melakukan “perancangan kilat” [7]. Perancangan kilat berfokus pada penyajian dari aspek-aspek perangkat lunak tersebut yang akan nampak bagi pelanggan atau pemakai (contohnya pendekatan *input* dan format *output*).

Metode prototipe merupakan metode pengembangan sistem dimana hasil analisa perbagian sistem langsung diterapkan kedalam sebuah model tanpa menunggu seluruh sistem selesai. *Mock-up* adalah sesuatu yang digunakan sebagai model desain yang digunakan untuk mengajar, demonstrasi, evaluasi desain, promosi, atau keperluan lain. Sebuah *mock-up* disebut sebagai prototipe perangkat lunak jika menyediakan atau mampu mendemonstrasikan sebagian besar fungsi sistem perangkat lunak dan memungkinkan pengujian desain sistem perangkat lunak. Iterasi terjadi pada pembuatan prototipe sampai sesuai dengan keinginan pelanggan atau pengguna [8].



Gambar 2.1 Pemodelan Prototipe

(Sumber : [8])

Adapun langkah atau tahapan penggunaan metode prototipe dalam penelitian ini adalah :

a. Tahap Mendengarkan Pelanggan

Pada tahapan ini, peneliti berkomunikasi dengan pelanggan untuk mengetahui kebutuhan sistem yang akan dibuat. Peneliti mendengarkan keluhan pelanggan

tentang bimbingan konseling kesehatan mental di Rumah Sakit Jiwa Darah Provinsi Lampung. Hasil komunikasi berupa pengumpulan data terdiri dari observasi dan wawancara.

b. Tahap Membangun/Memperbaiki *Mock-Up*

Setelah pengumpulan data yang dibutuhkan didapatkan, maka selanjutnya merancang sistem atau aplikasi perpustakaan elektronik. Perancangan sistem terdiri dari *use case diagram*, *activity diagram*, *class diagram*, dan antarmuka aplikasi berbasis *Android*. Setelah dilakukan perancangan, maka rancangan tersebut diimplementasikan ke dalam pemrograman aplikasi *Android*. Bahasa pemrograman yang digunakan adalah *Kotlin* dan penyimpanan data menggunakan *MySQL*. Aplikasi yang digunakan dalam media pembuatan aplikasi *Android* adalah *Android Studio*.

c. Tahap Pelanggan Melihat/Menguji *Mock-Up*

Tahap ini dilakukan pengujian aplikasi dengan menguji pemasangan aplikasi dengan tujuan apakah aplikasi mampu berjalan pada *platform smartphone* *Android* dengan sebagaimana semestinya yang mengacu pada fungsional sistem.

2.4 Pengertian *Android*

Android adalah sistem operasi berbasis *Linux* yang dirancang untuk perangkat seluler layar sentuh seperti *smartphone* dan komputer tablet. *Android* pada awalnya dikembangkan oleh *Android, Inc.*, dengan dukungan keuangan dari *Google*, yang kemudian membelinya pada tahun 2005. Antarmuka pengguna *Android* umumnya merupakan manipulasi langsung, menggunakan gerakan sentuh yang mirip dengan tindakan nyata, seperti menggesek, mengetuk, dan mencubit untuk memanipulasi objek di layar, seperti serta *keyboard* virtual untuk menulis teks. *Android* adalah sistem operasi sumber terbuka, dan *Google* merilis kodenya di bawah lisensi *Apache*. Kode dengan *open source* dan lisensi-lisensi di *Android* memungkinkan perangkat lunak untuk menjadi dimodifikasi dan didistribusikan secara bebas oleh pembuat perangkat, operator nirkabel, dan aplikasi pengembang. Selain itu, *Android* memiliki sejumlah besar Komunitas pengembang aplikasi (*apps*) yang memperluas fungsionalitas perangkat, umumnya ditulis dalam versi yang disesuaikan dari bahasa pemrograman *Java* [9].

Android adalah sebuah sistem operasi perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware*, dan aplikasi. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. *Android* bisa berjalan di beberapa macam perangkat dari banyak produsen yang berbeda. *Android* menyertakan *kit development* perangkat lunak untuk penulisan kode asli dan perakitan modul perangkat lunak untuk membuat aplikasi bagi pengguna *Android*. *Android* Juga menyediakan pasar untuk mendistribusikan aplikasi. secara keseluruhan, *Android* menyatakan ekosistem untuk aplikasi seluler [10].

2.5 Bahasa Pemrograman dan *Database*

2.5.1 *Kotlin*

Kotlin adalah sebuah bahasa pemrograman dengan *Statically typed* (tipe statis) yang berjalan pada platform *Java Virtual Machine* (JVM). *Kotlin* menggunakan *compiler* LLVM yang artinya, dapat dikompilasi ke dalam kode *JavaScript*. Pengembang utamanya berasal dari tim programmer *JetBrains* yang bermarkas di Rusia. Bahasa pemrograman yang satu ini banyak diminati oleh para *developer*. *Kotlin* merupakan bahasa yang *powerfull* tentu cocok bagi *developer* dalam membuat aplikasi *Android*. Karena *Kotlin* adalah bahasa nomor satu untuk pengembangan aplikasi *Android*. Adapun kelebihan *Kotlin* adalah [11]:

- a. *Concise* : *Kotlin* mampu mengurangi *boilerplate of code* atau tingkat kerumitan dari kode yang biasa kita tulis, ketika menggunakan bahasa *Java*.
- b. *Safe* : *Kotlin* mampu menjamin bahwa setiap *syntax* yang kita tulis secara proses kompilasi dapat mencegah kemungkinan terjadinya *error*, misalnya mampu mencegah terjadinya *NullPointerExceptions* ketika kita *coding* menggunakan bahasa *Java*.
- c. *Versatile* : *Kotlin* sejatinya sama seperti *Java*, karena memang *kotlin* itu sendiri di turunkan dari bahasa induknya, yaitu *Java*. Sehingga *kotlin* juga dapat di pakai dalam pengembangan aplikasi di *web* maupun *mobile*.
- d. *Interoperable* : *Kotlin* tidak sama seperti bahasa *Java* turunan lainnya (misal; *Scala* ataupun *Clojure*) yang tidak dapat dijalankan bersamaan dengan kode yang kita tulis menggunakan *Java*. *Kotlin* mampu membaca kode lama atau

library yang kita gunakan atau kita tulis dengan bahasa *Java* dan begitupun sebaliknya.

2.5.2 MySQL

MySQL adalah pengembangan lanjutan dari proyek UNIREG yang dikerjakan oleh Michael Monty Widenius dan TcX (perusahaan perangkat lunak asal Swedia). *MySQL* adalah DBMS yang *open source* dengan dua bentuk lisensi, yaitu *free software* (perangkat lunak bebas) dan *shareware* (perangkat lunak berpemilik yang penggunaannya terbatas). Jadi, *MySQL* adalah *database server* yang gratis dengan lisensi GNU *General Public License* (GPL) sehingga dapat dipakai untuk keperluan pribadi atau komersial tanpa harus membayar lisensi yang ada [12].

Seperti yang sudah disebutkan sebelumnya, *MySQL* masuk ke dalam jenis RDBMS (*Relational Database Management Sistem*). Maka dari itu, istilah semacam baris, kolom, tabel dipakai pada *MySQL*. *MySQL* merupakan *server* yang melayani *database*, kita dapat mempelajari pemrograman khusus yang disebut *query* (perintah) *SQL*. *MySQL* merupakan *database engine* atau *server database* yang mendukung bahasa *SQL* sebagai bahasa interaktif dalam mengelola data. *MySQL* adalah sebuah perangkat lunak sistem manajemen basis data *SQL* atau DBMS yang *multithread*, *multi-user* [13].

2.6 Pemodelan Sistem UML (*Unified Modelling Language*)

UML (*Unified Modelling Language*) adalah suatu metode dalam pemodelan secara visual yang digunakan sebagai sarana perancangan sistem berorientasi objek. *UML* juga dapat didefinisikan sebagai suatu bahasa standar visualisasi, perancangan, dan pendokumentasian sistem, atau dikenal juga sebagai bahasa standar penulisan *blueprint* sebuah *software* (Rosa & Shalahuddin, 2018). *UML* adalah bahasa pemodelan visual yang digunakan untuk merancang dan mendokumentasikan struktur dan perilaku sistem perangkat lunak. Ini adalah standar industri yang digunakan oleh para pengembang perangkat lunak di seluruh dunia untuk menyatukan pemahaman tentang sistem yang kompleks. Ada beberapa jenis diagram dalam *UML* yang mewakili berbagai aspek dari sistem, seperti:

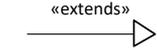
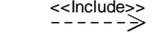
- a. Diagram Use Case: Menggambarkan fungsionalitas sistem dari sudut pandang pengguna.
- b. Diagram Kelas: Menunjukkan struktur kelas dalam sistem dan hubungan antara mereka.
- c. Diagram Sekuensi: Menggambarkan interaksi antara objek dalam urutan waktu tertentu.
- d. Diagram Aktivitas: Menunjukkan alur kerja atau proses bisnis dalam sistem.
- e. Diagram Komponen: Menggambarkan komponen fisik sistem dan hubungan antara mereka.
- f. Diagram Penyekalaan: Menunjukkan bagaimana sistem didekomposisi menjadi bagian-bagian yang lebih kecil.

Penggunaan UML membantu dalam memahami, merancang, dan mengkomunikasikan struktur dan perilaku sistem dengan jelas dan konsisten di antara pengembang perangkat lunak, analis, dan pemangku kepentingan lainnya. Jenis pemodelan UML yang digunakan dalam pembuatan aplikasi pada penelitian ini adalah *use case diagram*, *activity diagram*, dan *class diagram*.

2.6.1 Use Case Diagram

Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang dibuat aktor dan *use case*. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi, walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.

Tabel 2.1 Simbol *Use Case Diagram*

Simbol	Deskripsi
<p><i>Use case</i></p> 	Fungsionalitas yang disediakan sistem sebagai unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i>
<p>Aktor</p> 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar itu sendiri.
<p>Asosiasi</p> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
<p>Ekstensi</p> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> , dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan.
<p>Generalisasi</p> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
<p><i>Include</i></p> 	<i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan

2.6.2 Activity Diagram

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

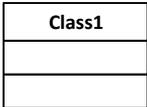
Tabel 2.2 Simbol *Activity Diagram*

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

2.6.3 Class Diagram

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki suatu kelas, sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Tabel 2.3 Simbol *Class Diagram*

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem.

Simbol	Deskripsi
Natarmuka/ <i>interface</i> Interface2 ○—	Sama dengan konsep <i>interface</i> dalam pemograman berorientasi objek.
Asosiasi —————	Relasi antar kelas dalam makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Asosiasi berarah —————>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Generalisasi —————>	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
Kebergantungan ----->	Relasi antar kelas dengan makna kebergantungan antar kelas.
Agregasi —————◇	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>).

2.7 Kamus Data

Kamus data adalah kumpulan atau struktur data yang menyimpan informasi dalam bentuk yang terstruktur, biasanya dalam format tabel atau kumpulan nilai kunci dan nilai-nilai terkaitnya. Kamus data memberikan deskripsi tentang struktur, tipe data, hubungan, dan metadata lainnya yang terkait dengan setiap elemen data dalam suatu sistem atau *database*. Kamus data berfungsi sebagai panduan atau referensi untuk memahami bagaimana data disimpan, diakses, dan dikelola dalam suatu sistem. Kamus data menyediakan informasi penting tentang elemen-elemen data seperti nama kolom, tipe data, panjang maksimum, batasan, dan deskripsi singkat tentang setiap kolom atau entitas dalam *database*.

Dengan kamus data, pengguna dapat memahami struktur data, mengerti relasi antara entitas-entitas, dan memastikan konsistensi dan integritas data. Ini juga membantu dalam pengembangan aplikasi, pemeliharaan sistem, dan analisis data. Dengan kata lain, kamus data berperan penting dalam manajemen dan pengelolaan data secara efisien dalam suatu organisasi [14].

Struktur kamus data dapat bervariasi tergantung pada kebutuhan dan kompleksitas sistem atau database yang digunakan. Namun, secara umum, kamus data mencakup beberapa komponen utama, termasuk:

- a. Nama Kolom/Entitas: Menunjukkan nama dari setiap kolom dalam tabel atau entitas dalam struktur data. Nama ini harus jelas dan deskriptif agar pengguna dapat dengan mudah memahaminya.
- b. Tipe Data: Menentukan jenis data yang dapat disimpan dalam setiap kolom, seperti string, angka, tanggal, atau jenis data khusus lainnya. Penetapan tipe data membantu memastikan konsistensi dan integritas data.
- c. Panjang Maksimum: Untuk kolom data yang memiliki tipe data terbatas, seperti string atau angka dengan panjang tertentu, kamus data harus mencantumkan panjang maksimum yang diizinkan untuk setiap kolom.
- d. Keterangan/Deskripsi: Memberikan penjelasan atau deskripsi singkat tentang setiap kolom atau entitas, termasuk informasi tentang penggunaan, makna, atau aturan khusus yang terkait dengan kolom tersebut.
- e. Batasan dan Kendala: Mendefinisikan batasan atau kendala yang berlaku untuk setiap kolom atau entitas, seperti kunci unik, kunci asing, batasan nilai, atau aturan validasi lainnya.
- f. Indeks: Jika diperlukan, kamus data dapat mencakup informasi tentang indeks yang diterapkan pada kolom-kolom tertentu untuk meningkatkan kinerja pencarian dan pengurangan data.
- g. Referensi dan Relasi: Jika ada, kamus data juga dapat mencakup informasi tentang hubungan atau relasi antara entitas-entitas dalam struktur data, termasuk kunci asing yang menghubungkan kolom-kolom antara tabel.
- h. Metadata Tambahan: Kamus data juga dapat mencakup metadata tambahan seperti pembuat data, tanggal pembuatan, versi kamus data, atau informasi lain yang mungkin relevan untuk manajemen dan pemahaman data.

2.8 Penelitian Terkait

Penelitian terkait dengan penelitian yang dilakukan saat ini adalah seperti pada tabel 2.4.

Tabel 2.4 Penelitian Terkait

Nama	Judul	Metode Pengembangan Sistem	Kesimpulan
[15]	Sistem Informasi Layanan E-Konseling Psikologi untuk Mahasiswa Berbasis Website dengan Metode Prototype	Prototype	Sistem dapat digunakan untuk layanan e-konseling secara online mulai dari penjadwalan, proses konseling, dan memberikan rating kepuasan kepada konselor
[16]	Pengembangan Sistem Informasi Manajemen Layanan Konseling FILKOM UB berbasis Mobile	Waterfall	Pengguna memiliki tingkat kepuasan yang lebih tinggi dalam menggunakan sistem informasi manajemen layanan konseling berbasis mobile dibandingkan dengan sistem yang sudah ada sebelumnya
[1]	Pembangunan Aplikasi Perangkat Bergerak Konseling Online pada Anxiety Disorder berbasis Android menggunakan Metode Personal	Personal Extreme Programming	Membantu memberikantindakan preventif dan penanganan terhadap gejala gangguan kecemasan, mengatasi kesulitan dalam mencari informasi layanan kesehatan mental, memberi

Nama	Judul	Metode Pengembangan Sistem	Kesimpulan
	Extreme Programming		kemudahan dalam melakukan konseling, mengedukasi pentingnya kesehatan mental, hingga membantu mencari pertolongan ketika keadaan darurat
[17]	Perancangan Aplikasi Kesehatan Mental “NAFS” (Islamic Psycho Spiritual Therapy) Berbasis Android Menggunakan Metode PDCA (Plan-Do - CheckAction)	PDCA	Aplikasi NAFS bisa digunakan terutama bagi individu dengan mental illness, sebagai bentuk pertolongan pertama kesehatan mental (mental health first aid) dan media edukasi kesehatan mental
[18]	Pengembangan Prototype Cyber Counseling dalam Setting Pendidikan dan Kesehatan Mental Bagi Mahasiswa	Tidak disebutkan	Membantu mahasiswa mencurahkan permasalahannya baik akademik maupun non akademik kepada konselor
[19]	Perancangan dan Implementasi E-Service BAAK IIB Darmajaya Untuk Meningkatkan Layanan Kepada Mahasiswa Di Era	Tidak disebutkan	Layanan yang dihasilkan pada penelitian ini diberi nama E- Service. E-Service mampu memberikan layanan kepada mahasiswa

Nama	Judul	Metode Pengembangan Sistem	Kesimpulan
	Pandemi Berbasis Android		tanpa mereka datang ke kampus
[20]	Implementasi Sistem Pelaporan Sarana dan Prasarana Kegiatan Belajar Mengajar Berbasis Android	<i>Prototype</i>	<p>Penggunaan aplikasi ini dapat memberi kemudahan untuk menangani kerusakan serta informasi aduan kerusakan.</p> <p>Bedasarkan hal tersebut, dapat disimpulkan bahwa dibutuhkan sistem yang dapat menyampaikan pelaporan secara cepat dan tepat, <i>real time</i></p>
[21]	<i>Prototype</i> Pengadaan Dan Distribusi Barang Pada Waralaba Fried Chicken dan Burger Lampung	<i>Prototype</i>	Sistem yang dibuat dapat menghasilkan laporan-laporan yang sebelumnya belum ada, seperti laporan data barang, data kios, data petugas, dan data supplier