

BAB II KAJIAN LITERATUR

2.1 Sistem Informasi

Sistem adalah sekumpulan unsur atau elemen yang saling berkaitan dan saling mempengaruhi dalam menjalankan suatu fungsi untuk mencapai tujuan yang telah direncanakan [6]. Informasi merupakan kumpulan data yang berhasil diolah dan disusun sehingga memiliki makna serta nilai guna bagi orang lain [7]. Sistem informasi berbasis web merupakan sistem yang memanfaatkan teknologi web untuk menyediakan akses informasi melalui jaringan internet [8]. Sistem informasi tersusun oleh beberapa komponen utama yang saling berkaitan dan bekerja sama, komponen ini biasa disebut *building block* atau blok bangunan sistem informasi. Terdapat 6 blok yang saling bekerja sama untuk mencapai tujuan yang telah direncanakan [9].

1. Blok Masukan (*Input Block*)

Komponen sistem informasi yang berfungsi sebagai sarana untuk mengumpulkan data dari berbagai sumber. Data yang dikumpulkan melalui blok ini tidak hanya berupa dokumen-dokumen, formulir, transaksi atau data digital yang diolah oleh sistem, tetapi lebih ke bahan awal yang akan diproses sampai menjadi sebuah informasi.

2. Blok Model (*Model Block*)

Tempatnya prosedur, aturan logika, serta metode perhitungan atau algoritma yang digunakan untuk mengolah data masukan. Blok ini merupakan otak dalam sistem dalam arti prosedur-prosedur yang digunakan untuk menjalankan logika dalam memproses data. Sistem melakukan proses di mana data yang telah diambil dari blok masukan dan kemudian diproses yang akhirnya menghasilkan informasi sesuai yang dibutuhkan.

3. Blok Keluaran (*Output Block*)

Hasil akhir dari proses pengolahan sistem dalam bentuk laporan. Laporan keluaran ini yang digunakan sebagai dasar *input* bagi sebuah keputusan, perencanaan, dan pengendalian aktivitas organisasi.

4. Blok Teknologi (*Technology Block*)

Komponen dalam sistem yang mendukung hingga proses *output* data. Teknologi ini mencakup semua peralatan dalam proses penerimaan data masukan, pemrosesan data, penyimpanan data dan yang menjadi *input* berupa keluaran. Sistem teknologi melibatkan perangkat keras dan perangkat lunak serta sumber daya manusia yang terintegrasi dalam proses tersebut.

5. Blok Basis Data (*Database Block*)

Koleksi data yang saling berhubungan dan disimpan dalam media penyimpanan terstruktur. Fungsi blok basis data ini adalah sebagai tempat penyimpanan data agar dapat diakses, diolah dan diperbaharui oleh sistem secara terpisah.

6. Blok Kendali (*Control Block*)

Blok ini merupakan mekanisme pengawasan dan pengendalian dalam sistem untuk memastikan bahwa sistem dijalankan sesuai dengan yang diharapkan. Ini meliputi terjadinya kerusakan pada sistem dan penyalahgunaan data, kecurangan dan gangguan yang dapat mempengaruhi kinerja sistem.

2.2 Waterfall

Model *Waterfall* merupakan salah satu model *Software Development Life Cycle* (SDLC) yang sering digunakan dalam pengembangan sistem informasi atau perangkat lunak. Model ini menggunakan pendekatan sistematis dan berurutan. Tahapan dalam model ini dimulai dari tahap analisis kebutuhan hingga tahap pengelolaan (*maintenance*) dan dilakukan secara bertahap [10].



Gambar 2. 1 Model Waterfall [11]

Gambar 2.1 di atas merupakan tahapan dari metode *waterfall*, berikut penjelasan singkat dari setiap tahapan-tahapannya [11]:

1. *Requirements Analysis* (Analisis Kebutuhan)

Tahapan awal dari model *waterfall* yang bertujuan untuk menganalisis kebutuhan sistem dengan cara mengumpulkan data melalui observasi, wawancara, diskusi dan lainnya untuk mengetahui kebutuhan dan permasalahan yang ada.

2. *Design* (Perancangan)

Tahapan ini adalah membuat desain aplikasi sebelum masuk ke tahap pengembangan untuk mengetahui atau gambaran untuk merancang tampilan antarmuka pengguna sesuai yang diinginkan.

3. *Development* (Pengembangan)

Rancangan desain di implementasikan ke dalam bentuk kode program oleh tim pengembang. Proses pengembangan dilakukan sesuai dengan rancangan desain yang telah ditentukan sebelumnya.

4. *Testing* (Pengujian)

Tahapan ini dilakukan setelah rancangan desain sudah selesai di implementasikan ke dalam bentuk kode program dan menghasilkan sebuah sistem. Pengujian ini bertujuan untuk memastikan sistem berjalan dengan baik dan mencegah terjadinya masalah sebelum sistem benar-benar digunakan.

5. *Maintenance* (Pemeliharaan)

Pemeliharaan adalah tahapan terakhir dari model *waterfall* setelah sistem sudah digunakan. Tahapan ini melibatkan *update* sistem atau pembaruan untuk menjaga sistem tetap berjalan dengan baik sesuai kebutuhan di masa yang akan datang.

Adapun kelebihan dari metode *waterfall* adalah sebagai berikut [10]:

1. Kualitas dari sistem yang dihasilkan akan baik, karena pelaksanaannya dilakukan secara bertahap.
2. Proses pengembangan model fase *one by one*, sehingga meminimalkan kesalahan yang mungkin akan terjadi.
3. Dokumen pengembangan sistem sangat terorganisir, karena setiap fase harus terselesaikan dengan lengkap sebelum melangkah ke fase berikutnya.

Adapun kekurangan dari metode *waterfall* adalah sebagai berikut [10]:

1. Waktu pengembangan lama dan biayanya mahal.
2. Diperlukan manajemen yang baik, karena proses pengembangan tidak dapat dilakukan secara berulang sebelum terjadinya suatu produk.
3. Kesalahan kecil akan menjadi masalah besar jika tidak diketahui sejak awal pengembangan yang berakibat pada tahapan selanjutnya.
4. Pada kenyataannya, jarang mengikuti urutan sekuensial (runtutan) seperti pada teori. Iterasi (perulangan) sering terjadi menyebabkan masalah baru.

2.3 Teknologi Pengembangan Sistem

Teknologi sangat berperan penting dalam di era saat ini, di mana dengan teknologi yang tepat sebuah pekerjaan dapat diselesaikan dengan efektif dan efisien. Pemilihan teknologi yang tepat sangat berpengaruh terhadap efisiensi kinerja sebuah sistem [12].

2.3.1 Website

Website merupakan kumpulan halaman pada suatu situs web yang berisi informasi-informasi berupa teks, gambar, video dan sebagainya yang dapat diakses oleh siapa pun, di mana pun dan kapan pun melalui internet [13]. *Website* juga dapat disebut sebagai media berbasis internet yang menyajikan informasi maupun layanan yang disediakan untuk pengguna. Halaman *website* dibuat dengan menggunakan bahasa standar yang biasa disebut HTML yang di mana kode HTML ini dibaca dan diproses oleh web browser sehingga menampilkan informasi yang dapat dilihat atau dibaca pengguna [14].

Fungsi *website* adalah sebagai penyedia layanan informasi yang dapat diakses oleh pengguna melalui perangkat yang terhubung ke internet seperti *Smartphone*, laptop, tablet dan komputer. Selain sebagai media informasi, *website* juga dimanfaatkan untuk berbagai keperluan, seperti promosi bisnis, penjualan layanan atau jasa, serta penyedia layanan lainnya yang berbasis *online*. Oleh karena itu, *website* menjadi salah satu layanan penting dalam dunia internet dan banyak digunakan oleh berbagai organisasi untuk mendukung layanan informasi kepada pengguna [15].

Dalam pelayanan gereja saat ini, *website* menjadi kebutuhan yang sangat penting dalam mendukung setiap proses pelayanan. *Website* memberikan kemudahan bagi gereja dalam mengelola pelayanan, khususnya dalam penyampaian informasi kepada jemaat dalam sebuah gereja. Pemanfaatan *website* yang baik akan sangat membantu baik dalam mengelola data jemaat, penyampaian informasi maupun proses administrasi yang ada. Dengan adanya *website* juga akan memperkuat hubungan antar komunitas pada lembaga keagamaan dan meningkatkan keterlibatan jemaat dan menyebarkan pesan kepada lebih banyak orang [16].

2.3.2 Bahasa Pemrograman PHP

PHP singkatan dari *Hypertext Preprocessor* merupakan bahasa berbentuk *script* yang ditempatkan dalam server dan diproses di server. Hasilnya akan dikirimkan ke *client*, tempat pemakai menggunakan browser. PHP dikenal sebagai bahasa *scripting*, yang menyatu dengan tag-tag HTML, di eksekusi di server, dan digunakan untuk membuat halaman web yang dinamis seperti halnya *Active Server Pages* (ASP) atau *Jawa Server Pages* (JSP). PHP merupakan sebuah *software open source* [17].

PHP mendukung interaksi langsung dengan sistem basis data, sebagai program berbasis server, PHP mampu mengelola permintaan pengguna dengan cara manipulasi data serta menampilkan secara *real-time* ke halaman web. Salah satu fitur PHP adalah kemampuannya untuk bekerja dengan berbagai sistem manajemen basis data, khususnya adalah MySQL. Dengan hal ini proses penyimpanan dan pengelolaan data akan lebih efisien. Hal ini menjadikan PHP sebagai pilihan yang populer dan fleksibel dalam pengembangan aplikasi web berbasis data [18].

2.3.3 Framework Laravel

Framework laravel adalah salah satu *framework* yang populer digunakan oleh para *programmer* saat ini dalam membantu proses pengembangan suatu sistem [19]. Laravel juga membantu *developer* untuk memaksimalkan penggunaan PHP di dalam proses pengembangan *website* dengan fitur unggulan yang dimiliki seperti *template engine*, *routing*, dan *modularity* [20].

Sebagai sebuah *framework* PHP, laravel hadir sebagai platform pengembangan web yang bersifat *open source*. Laravel dikenal memiliki sintaks yang elegan dan ekspresif sehingga menarik bagi para pengembang [21]. Dengan sintaks yang jelas dan ekspresif, dokumentasi yang lengkap, serta dukungan komunitas yang luas dan aktif, laravel menjadi pilihan utama bagi banyak pengembang dalam mengoptimalkan proses pengembangan aplikasi web mereka [22].

Salah satu fitur unggulan laravel adalah *Eloquent*, ORM bawaan yang memungkinkan pengembangan berinteraksi dengan basis data menggunakan sintaks PHP yang bersih dan intuitif. Selain itu, *Blade*, mesin *templating laravel*, mempermudah pengembangan tampilan dengan sintaksis yang mudah dipahami. *Framework* ini juga menawarkan alat-alat untuk pengelolaan antrian, sistem notifikasi, dan *caching*, semuanya dirancang untuk mempercepat dan mengoptimalkan kinerja aplikasi web. Dengan ekosistem paket tambahan yang kuat dan dukungan terus-menerus dari komunitas pengembang yang aktif, Laravel menjadi pilihan yang sangat baik bagi siapa pun yang ingin mengembangkan aplikasi web dengan cepat dan efisien [22].

2.3.4 Laravel Filament

Laravel filament merupakan *package* pada *framework* laravel yang dirancang untuk mempermudah pembuatan *dashboard* admin, terutama untuk aplikasi CRUD yang kompleks dengan antarmuka yang sederhana dan mudah diatur [23]. Salah satu fitur utama laravel

filament adalah tampilan panel administrasinya yang dirancang intuitif dan mudah digunakan. Antarmuka yang bersifat *user-friendly* ini mendukung peningkatan produktivitas pengembang, karena proses pengelolaan data dapat dilakukan dengan lebih praktis tanpa mengurangi fleksibilitas dalam pengembangan sistem. Laravel filament dibangun dengan pendekatan TALL Stack, yaitu kombinasi Tailwind CSS, Alpine.js, Laravel, dan Livewire. Integrasi teknologi tersebut memungkinkan pengembangan panel admin yang responsif, interaktif, serta terstruktur dengan baik [24].

Laravel filament menyediakan berbagai fitur yang mendukung kebutuhan pengelolaan aplikasi seperti kemudahan dalam pembuatan panel admin, fasilitas Create, Read, Update, Delete (CRUD) yang dapat dihasilkan secara otomatis, pengelolaan data yang lebih terorganisir, desain antarmuka yang modern, serta formulir yang dinamis dan fleksibel [24].

2.3.5 MySql

Basis data paling populer digunakan di situs web adalah MySQL. MySQL adalah Sistem Manajemen Database Relasional (RDBMS). Jantung dan pengelola sistem *database* adalah server MySQL. Untuk mendukung pengelolaan data menggunakan Mysql pengembangan sistem, diperlukan sebuah *software* yang mampu menjalankan server secara lokal seperti XAMPP [25].

XAMPP (Cross Platform, Apache, MySQL, PHP, dan Perl) adalah aplikasi *open source* yang dikembangkan oleh *Apache Friends*. XAMPP memberikan solusi sederhana dan ringan untuk digunakan untuk mencoba menjalankan sebuah *website*, XAMPP juga dapat digunakan secara gratis. Selain sistem operasi Windows, XAMPP juga dapat dijalankan pada Mac dan Linux [26]. Di bawah ini merupakan tampilan aplikasi XAMPP yang digunakan sebagai server lokal.

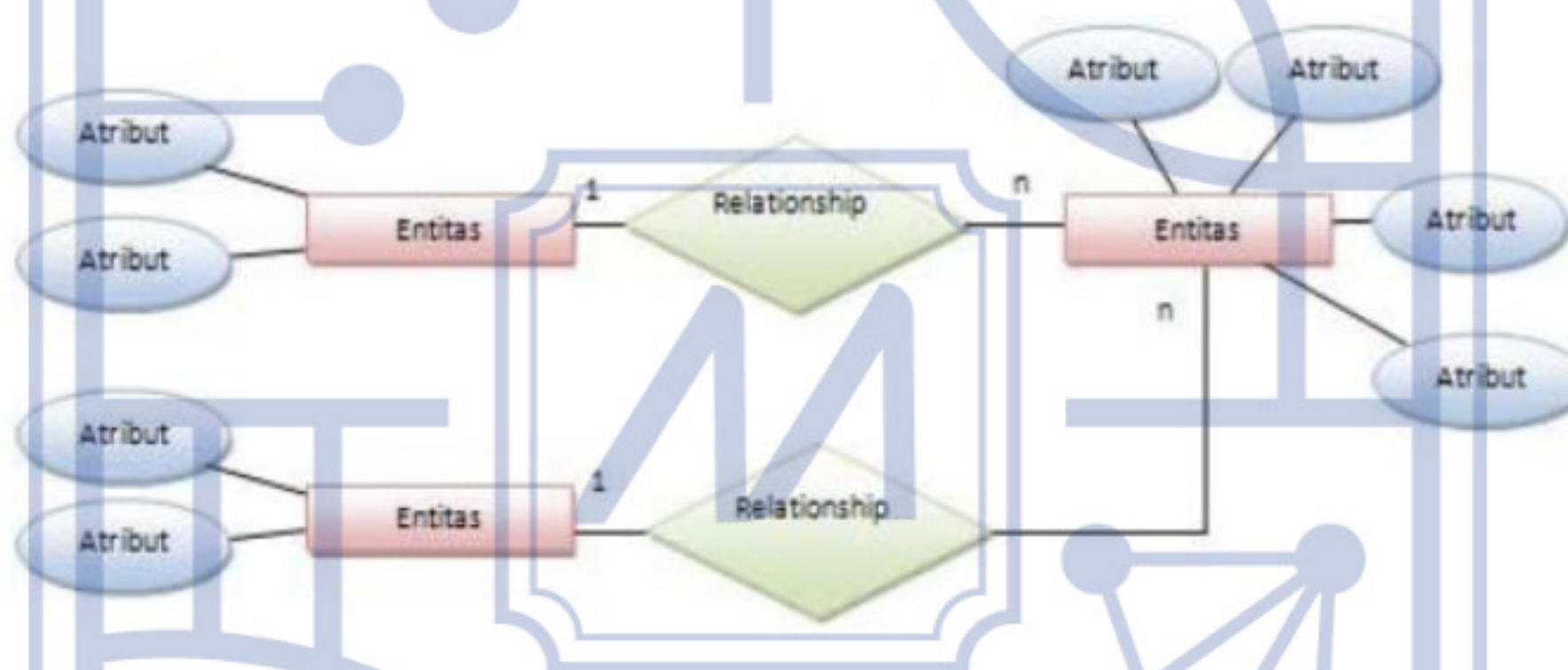


Gambar 2. 2 Tampilan Aplikasi XAMPP [25]

2.3.6 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah suatu diagram yang digunakan untuk merancang suatu basis data, dipergunakan untuk memperlihatkan hubungan atau relasi antar entitas atau objek yang terlihat beserta atributnya. Dengan kata lain, ERD menjadi suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi [27].

ERD merupakan model pada basis data untuk menggambarkan skema konseptual suatu sistem berdasarkan hubungan data yang ada yang umumnya diterapkan pada basis data relasional dan bersifat *top-down*, yaitu dimulai dari gambaran umum sistem kemudian diturunkan ke detail data [28]. Di bawah ini merupakan contoh penerapan ERD sebagai berikut.



Gambar 2. 3 Contoh Entity Relationship Diagram [27]

ERD berfungsi sebagai alat bantu dalam pembuatan basis data dan memberikan gambaran bagaimana kerja basis data yang akan dibuat [28]. Terdapat 3 elemen dasar pada ERD sebagai berikut [28]:

1. Entitas: Objek dalam suatu *database* yang dapat diidentifikasi baik berupa orang, tempat, atau benda.
2. Atribut: Informasi atau karakteristik dalam entitas di mana yang entitas tersebut harus memiliki *primary key*. Atribut ini akan menjelaskan secara detail data dalam entitas.
3. Relasi: Hubungan antara satu entitas dengan entitas lain. Relasi menunjukkan hubungan antara entitas. Relasi dalam ERD ada beberapa macam yaitu :
 - a. *One to One*
Hubungan satu ke satu (1:1) yaitu di mana satu entitas hanya memiliki hubungan dengan satu entitas lain.
 - b. *One to Many*

Hubungan satu ke banyak (1:N) yaitu di mana satu entitas dapat memiliki hubungan dengan banyak entitas lainnya.

c. *Many to Many*

Hubungan banyak ke banyak (M:N) yaitu di mana banyak entitas dapat saling berhubungan satu dengan entitas lain.

2.4 GSJA PCC Brayan

Gereja Sidang Jemaat Allah, Praise Community Church Brayan adalah sebuah rumah ibadah yang terletak di Pulo Brayan Bengkel, Kecamatan Medan Timur, Kota Medan, Sumatera Utara. GSJA PCC Brayan merupakan salah satu gereja yang bertumbuh oleh kasih karunia Tuhan melalui kesetiaan jemaat dan pertolongan Tuhan dari tahun ke tahun. Gereja ini diresmikan pada tanggal 4 Februari 2007 di rumah kecil tempat tinggal Gembala dan keluarga (Pdt Salomo Sinaga), dengan jumlah jemaat awal sebanyak 9 orang dengan 2 Komunitas Sel (komsel). Meskipun dimulai dengan jumlah yang kecil, namun gereja ini memiliki kerinduan besar untuk menjadi tempat pertumbuhan iman dan pelayanan bagi masyarakat sekitar.

Pada tahun 2008, gereja mulai mengalami perkembangan yang signifikan. Jumlah jemaat bertambah menjadi sekitar 40 orang dan ada 4 komsel, sehingga dimulailah pembelian tanah dan pembangunan gedung gereja pertama sebagai sarana ibadah yang lebih memadai. Memasuki tahun 2009, GSJA PCC Brayan memperluas pelayanannya dengan menjalin kemitraan bersama Yayasan Compassion. Pada tahun itu juga gereja mulai menjalankan pelayanan pendidikan anak melalui pembukaan kegiatan Taman Kanak-Kanak (TK) sebagai bentuk perhatian gereja terhadap generasi anak. Pada tahun 2011, seiring pertumbuhan jemaat yang meningkat hingga mencapai 90 orang dan ada 9 komsel, pembangunan gedung gereja dilanjutkan dengan perluasan lantai dua dan bagian bawah untuk menampung kebutuhan ibadah dan kegiatan pelayanan. Pada tahun 2012 menjadi masa perkembangan lanjutan, di mana gereja membeli sebidang tanah di depan dan membangun ruangan Pelayanan Anak dan Remaja (PPA) serta kantor gereja guna mendukung pelayanan yang semakin luas dan terorganisir. Pada tahun 2014, gereja mengalami terobosan besar dengan melakukan perluasan melalui pembelian tanah di samping gereja. Pembangunan gedung baru dilakukan dengan ukuran 9,5 x 25 meter, sekaligus pembangunan lantai dua. Gedung gereja ini menghadap langsung ke Jalan Bengkel, menjadi lokasi strategis dan lebih representatif. Pada tahun itu jumlah jemaat telah bertumbuh pesat hingga mencapai sekitar 250 orang dan 14 komsel. Pada tahun 2015, gereja

terus melengkapi fasilitas dengan pemasangan AC di dalam ruang ibadah, pembangunan kamar mandi, serta pembuatan pagar gereja untuk kenyamanan dan keamanan jemaat. Kemudian pada tahun 2017, gereja kembali mengambil langkah besar dengan melakukan pembelian rumah yang digunakan sebagai tempat gedung Sekolah Minggu, TK, dan kelas PPA, sehingga pelayanan anak-anak semakin berkembang. Memasuki tahun 2024, gereja terus bertumbuh dan jemaat semakin diberkati Tuhan. Jumlah jemaat meningkat menjadi sebanyak 360 orang dan 26 komsel. Pertumbuhan ini membawa tantangan baru, salah satunya keterbatasan lahan parkir kendaraan. Oleh sebab itu, gereja merencanakan pembelian tanah tambahan di samping gereja. Pada bulan Februari 2026, gereja berhasil membeli tanah lahan parkir dari janji iman jemaat dan bantuan lainnya, sebagai bentuk nyata pertolongan Tuhan untuk menopang kebutuhan pelayanan gereja yang terus berkembang.

Saat ini GSJA PCC Brayan memiliki kurang lebih 400 orang jemaat, memiliki 3 cabang gereja yaitu PCC Tanjung Mulia, PCC Helvetia, PCC Mabar Hilir, dan satu pos pelayanan Pasar IX. Ibadah umum di gereja ini ada 3 sesi, yaitu sesi pertama di pagi hari pukul 08.15, sesi 2 di siang hari pukul 10.30 dan sesi 3 di sore hari pukul 16.30. Nilai-nilai dari GSJA yaitu, cinta Tuhan, rendah hati, jujur, rajin & bertanggung jawab. Visi dari gereja ini adalah Bertumbuh dalam iman, berbuah dalam kehidupan (Efesus 4:13) dan Misinya adalah menjadikan jemaat yang memberitakan injil ke semua suku, kaum dan bahasa.



Gambar 2. 4 GSJA PCC Brayan

Gambar 2.4 ini adalah suasana kegiatan ibadah raya yang dilakukan di GSJA PCC brayan setiap minggunya. Ibadah ini dibawakan oleh tim *praise & worship* yang didominasi oleh anak-anak muda gereja. Banyak kegiatan yang dilaksanakan di setiap minggu di gereja ini, seperti ibadah raya atau umum dan ibadah lainnya. Jadwal kegiatan ibadah yang akan

dilaksanakan biasanya akan di informasikan setiap minggu pada sesi pengumuman di akhir ibadah raya. Penyampaian informasi dilakukan secara langsung melalui pengumuman lisan yang ditampilkan di melalui *Videotron* gereja. Untuk jemaat yang tidak hadir pada saat ibadah raya, kemungkinan tidak akan mendapatkan informasi pengumuman yang telah disampaikan. Adapun pengumuman lainnya akan di sampaikan oleh pengurus gereja melalui grup jemaat. Berikut ini adalah ibadah rutin yang dilaksanakan setiap minggu di GSJA PCC Brayan :

1. Ibadah Raya

Ibadah raya atau ibadah umum adalah ibadah yang dilakukan setiap hari minggu di gereja. Ibadah umum di GSJA PCC Brayan ada 3 sesi, yaitu sesi pertama pukul 08:15 WIB, sesi kedua pukul 10:30 WIB dan sesi ketiga pukul 16:30 WIB.



Gambar 2. 5 Flyer Ibadah

2. Ibadah Menara Doa

Ibadah menara doa adalah ibadah khusus untuk menaikkan doa-doa atau biasa disebut doa syafaat yang dilakukan oleh para pelayan-pelayan dan jemaat dapat hadir mengikutinya. Ibadah menara doa biasa dilaksanakan pada hari rabu pukul 19:30 WIB sampai selesai.

3. Ibadah Kelompok Sel (Komsel)

Ibadah komsel adalah ibadah yang dilakukan oleh jemaat-jemaat di kelompok masing-masing. Ibadah ini di tangani oleh setiap pengurus kelompok sel (PKS) di komsel yang mereka tanggung jawab i. Ibadah komsel biasanya dilakukan di rumah jemaat sesuai dengan jadwal yang ditentukan di setiap komsel.

4. Ibadah Kaum Pria

Ibadah kaum pria adalah ibadah yang dilaksanakan setiap hari Selasa khusus kaum pria atau kaum bapak. Biasanya diadakan pada pukul 19:30 WIB di gereja maupun di rumah kaum pria secara bergantian.

5. Ibadah Kaum Wanita

Ibadah kaum wanita adalah ibadah yang dilaksanakan setiap hari Senin khusus kaum ibu. Biasanya diadakan pukul 19:30 WIB di gereja maupun di rumah kaum ibu secara bergantian.

6. Ibadah Geolinch (*Generation of Light in Christ*)

Ibadah *geolinch* adalah ibadah yang dilaksanakan setiap hari Sabtu pukul 19:30 WIB di gereja. Ibadah ini khusus dilakukan oleh kaum muda-mudi.

7. Ibadah Glow (*Generation Love of Worship*)

Ibadah ini adalah ibadah khusus anak remaja maupun PPA, ibadah ini dilaksanakan setiap hari Jumat di gereja pukul 16:30 WIB. Di bawah ini merupakan gambar kegiatan ibadah yang dilakukan di GSJA PCC Brayan.

2.5 Basis Data

Basis data merupakan sekumpulan data yang disusun secara terstruktur dan terintegrasi di dalam sistem pengelolaan basis data. Basis data tersimpan dalam bentuk beberapa tabel yang terdiri dari kolom dan baris. Basis data bertujuan untuk pengambilan keputusan atau pemrosesan data [29].

Basis data adalah kumpulan data terkait yang disimpan bersama dengan redundansi terkontrol untuk melayani satu atau lebih aplikasi secara optimal. Data disimpan sedemikian rupa sehingga terlepas dari program yang digunakan orang untuk mengakses data. Akses untuk menambahkan data baru, mengedit, dan mengambil data yang ada dari *database* adalah umum dan terkontrol. Basis data juga didefinisikan sebagai kumpulan informasi yang terkait secara logis, disimpan bersama, dirancang untuk memenuhi kebutuhan informasi suatu organisasi [30].

Contoh *database* adalah buku telepon yang berisi nama, alamat, dan nomor telepon orang yang disimpan di memori komputer. Basis data diatur oleh *field*, *record*, dan file sebagai berikut [30].

1. *Field*

Unit data terkecil yang memiliki arti bagi penggunaannya dan juga disebut data item atau elemen data. Nama, Alamat, dan Nomor telepon adalah contoh *field*. Ini dipresentasikan dalam basis data oleh sebuah nilai.

2. *Record*

Kumpulan *field* yang secara logis terkait dan setiap *field* memiliki jumlah *byte* yang tetap dan merupakan jenis data yang tetap atau dapat dikatakan bahwa sebuah *record* adalah satu set lengkap *field* dan setiap *field* memiliki beberapa nilai. Informasi lengkap tentang nomor telepon tertentu dalam basis data mewakili sebuah *record*. *Record* terdiri dari dua jenis yaitu *record* dengan panjang tetap dan *record* dengan panjang variabel.

3. *File*

Sebuah file adalah kumpulan *record* terkait. Umumnya, semua *record* dalam sebuah file memiliki ukuran dan jenis *record* yang sama, tetapi hal ini tidak selalu benar. *Record* dalam sebuah file dapat memiliki panjang tetap atau variabel tergantung pada ukuran *record* yang terkandung dalam file tersebut. Buku telepon yang berisi catatan tentang pemegang telepon yang berbeda adalah contoh file.

2.6 Unified Modeling Language (UML)

Unified Modeling Language atau biasa dikenal dengan singkatan UML, adalah bahasa standar yang digunakan untuk menggambarkan, menjelaskan, dan membangun perangkat lunak. Ini merupakan metode dalam pengembangan sistem berbasis objek yang juga berfungsi sebagai alat bantu dalam proses pengembangan sistem. Beberapa alat bantu yang digunakan dalam perancangan berbasis objek dengan menggunakan UML meliputi *use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram* [31].

UML merupakan alat yang sangat berharga dalam pengembangan perangkat lunak modern. Dengan kemampuannya untuk mendukung berbagai jenis pemodelan dan integrasi dengan metodologi lain, UML tetap relevan dalam menghadapi tantangan desain sistem yang semakin kompleks. Penggunaan UML yang tepat dapat meningkatkan komunikasi antara tim pengembang dan pemangku kepentingan, serta menghasilkan sistem yang lebih berkualitas dan sesuai dengan kebutuhan pengguna [32].

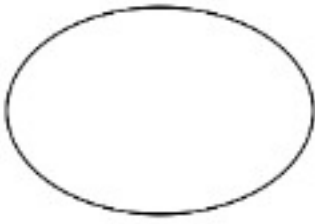
2.6.1 Use Case Diagram

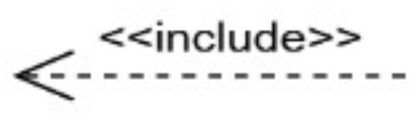
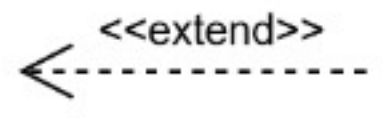
Use case diagram merupakan salah satu bentuk diagram dalam pemodelan sistem yang digunakan untuk menggambarkan hubungan serta interaksi antara sistem informasi dan pihak yang berperan sebagai pengguna (actor). Diagram ini memperlihatkan bagaimana suatu sistem merespons tindakan pengguna dalam menjalankan proses tertentu untuk mencapai tujuan yang telah ditetapkan. *Use case diagram* sangat membantu dalam mengetahui kebutuhan sistem serta menjadi acuan dalam tahap perancangan dan pengembangan sebuah sistem [15].

Dalam *use case diagram*, penamaan setiap *use case* sebaiknya dibuat sesederhana mungkin agar mudah dipahami. Terdapat dua elemen utama pada *use case* yaitu aktor dan *use case*. Aktor merupakan pihak yang berinteraksi dengan sistem yang dikembangkan. Aktor tidak selalu merujuk kepada manusia meskipun simbol aktor digambarkan sebagai manusia. Aktor bisa berupa individu, proses, maupun sistem lain. *Use case* menggambarkan layanan yang disediakan oleh sistem kepada aktor [33].

Dalam penerapan *use case*, terdapat simbol-simbol yang digunakan untuk menjelaskan sebuah proses. Di bawah ini adalah simbol-simbol *use case diagram* beserta penjelasan singkatnya sebagai berikut.

Tabel 2. 1 Simbol-Simbol Use Case Diagram

NO	Simbol	Nama	Keterangan
1		Aktor	Aktor menggambarkan pihak yang berinteraksi dengan sistem, baik berupa individu, kelompok pengguna, maupun sistem lain yang berada di luar batas sistem yang dikembangkan.
2		Use case	Menunjukkan fungsi atau layanan yang disediakan oleh sistem kepada aktor.
3		Asosiation	Garis ini menunjukkan adanya hubungan atau interaksi antara aktor dan sistem.

4		Include	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.
5		Extend	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi.
6		Generalization	Menunjukkan hubungan umum ke khusus baik antar aktor maupun antar <i>use case</i> , di mana entitas yang lebih spesifik mewarisi karakteristik dari entitas yang lebih umum.

Berdasarkan table di atas, simbol-simbol tersebut memiliki peran masing-masing yang akan dijelaskan sebagai berikut [34].

1. *Actor*

Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Aktor merupakan peran yang dimainkan oleh pemakai ketika berinteraksi dengan sistem.

2. *Use case*

Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama *use case*.

3. *Asosiation*

Komunikasi antar aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor.

4. *Include*

Include berarti *use case* yang ditambahkan akan selalu dipanggil saat *use case* tambahan dijalankan. *Include* berarti *use case* yang tambahan akan selalu melakukan pengecekan apakah *use case* yang ditambahkan telah dijalankan sebelum *use case* tambahan dijalankan.

5. *Extend*

Relasi *use case* tambahan ke sebuah *use case* dan *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu. Biasanya *use case* tambahan memiliki nama depan yang sama dengan *use case* yang ditambahkan,

6. Generalization

Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah *use case* di mana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.

2.6.2 Activity Diagram

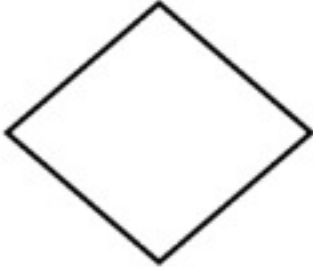
Activity diagram sering kali dianggap serupa dengan flowchart karena sama-sama menggambarkan alur proses. *Activity diagram* tidak hanya menampilkan alur kegiatan secara berurutan, tetapi juga mampu menggambarkan percabangan, proses paralel, aktivitas yang berjalan bersamaan, maupun alur tunggal dalam suatu sistem. *Activity diagram* dalam UML digunakan untuk melengkapi *use case* yang telah dirancang sebelumnya dengan menyajikan representasi grafis mengenai alur interaksi dalam suatu skenario yang sifatnya lebih spesifik [33].

Activity diagram digunakan untuk memodelkan perilaku objek atau sistem dalam menjalankan suatu proses bisnis. Diagram ini dapat menggambarkan berbagai tingkat proses, mulai dari alur kerja secara umum yang melibatkan beberapa *use case*, hingga detail aktivitas pada satu *use case* tertentu secara lebih rinci [33].

Dalam penerapan *activity diagram*, terdapat simbol-simbol yang digunakan untuk menjelaskan proses aktivitas yang terjadi seperti pada *use case diagram*. Berikut ini adalah simbol-simbol yang biasa digunakan pada *activity diagram*.

Tabel 2. 2 Simbol-Simbol Activity Diagram

NO	Simbol	Nama	Keterangan
1		Start Point	Simbol ini menunjukkan titik awal dimulainya suatu proses atau aktivitas
2		Aktivitiy	Simbol ini mempresentasikan suatu kegiatan atau proses yang dilakukan dalam sistem

3		Percabangan (Decision)	Simbol ini digunakan ketika terdapat suatu kondisi yang menyebabkan alur proses bercabang berdasarkan keputusan tertentu.
4		Penggabungan (Join)	Join node digunakan untuk menggabungkan kembali aktivitas paralel menjadi satu proses.
5		Control Flow	Alur urutan aktivitas.
6		Object flow	Menunjukkan aliran data atau objek yang digunakan.
7		Percabangan Pararel (Fork Node)	Simbol ini menunjukkan bahwa satu proses dapat berjalan menjadi beberapa aktivitas secara bersamaan.
8		End Point	Simbol ini menunjukkan akhir dari sebuah proses.

Berdasarkan table di atas, simbol-simbol tersebut memiliki peran masing-masing yang akan dijelaskan sebagai berikut [34].

1. *Start Point*

Simbol ini menyatakan bahwa sebuah objek dibentuk atau diawali.

2. *Activity*

Simbol ini menyatakan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.

3. *Decision*

Simbol ini menunjukkan penggambaran suatu keputusan/tindakan yang harus diambil pada kondisi tertentu

4. *Join*

Simbol ini menyatakan untuk menggabungkan kembali aktivitas dan aksi yang paralel.

5. *Control Flow*

Simbol ini menunjukkan urutan eksekusi

6. *Object Flow*

Simbol ini menunjukkan aliran objek dari sebuah aksi atau *action* ke *activity*.

7. *Fork Node*

Simbol ini menyatakan untuk memecah *behavior* menjadi *activity* atau *action* yang paralel

8. *End Point*

Simbol ini menyatakan bahwa sebuah objek dibentuk atau diakhiri.

2.7 Black Box Testing

Black box merupakan salah satu pendekatan dalam pengujian perangkat lunak yang digunakan tanpa meninjau maupun menganalisis kode sumber program. Dalam pengujian *black box*, penguji tidak memperhatikan struktur internal dari suatu sistem, tetapi hanya memperhatikan aspek eksternal suatu sistem, yaitu berupa antarmuka dan kecocokan data *input-process-output* [35].

Melalui pendekatan *black box testing*, pengembang maupun penguji perangkat lunak dapat membuat beberapa skenario pengujian dengan mengacu pada kebutuhan fungsional yang telah ditetapkan. Dengan demikian seluruh persyaratan fungsional program dapat diuji semua, sehingga dapat memastikan bahwa sistem dapat bekerja sesuai dengan yang telah dirancang sebelumnya [35]. Adapun kelebihan dan kekurangan dari *black box testing* adalah sebagai berikut [36]:

1. Penguji tidak perlu memerlukan pengetahuan teknis kode
2. Mengutamakan perspektif pengguna
3. Efektif dalam mendeteksi cacat fungsional
4. Dapat digunakan untuk pengujian otomatis
5. Tidak dapat mengidentifikasi masalah kode internal
6. Pengujian terbatas pada fungsionalitas yang terlihat