



BAB II

TINJAUAN PUSTAKA DAN

LANDASAN TEORI

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian dilakukan oleh Yosifa Gianinda dkk pada tahun 2019. Sistem ini bertujuan untuk memudahkan siswa dalam melakukan pendaftaran kegiatan ekstrakurikuler secara online, dan juga dapat membantu pembina dalam merekap data anggota serta mengelola data anggota dengan mudah dan memudahkan pembina agar dapat melakukan pengajuan proposal dan laporan kepada kesiswaan. Pengembangan sistem ini menggunakan metode prototype dan pengujian menggunakan black box testing. [3].

Penelitian ditulis oleh Syahid Abdullah dkk pada tahun 2020. Sistem bertujuan untuk membuat pendaftaran kegiatan ekstrakurikuler dan mengelola data anggota serta memudahkan kesiswaan untuk dapat memeriksa dan memverifikasi hasil proposal kegiatan. Hasil pada penelitian ini yaitu membantu siswa dalam melakukan pendaftaran kegiatan ekstrakurikuler, setelah siswa mendaftar maka sistem tersebut secara langsung akan memunculkan data anggota yang telah mendaftar pada masing-masing kegiatan ekstrakurikuler dan kesiswaan dapat juga menerima atau menolak usulan proposal tersebut. Pengembangan sistem ini menggunakan metode UA (Unfied Approach) dan pengujian menggunakan black box testing yang mengacu pada kinerja fungsional [4].

Penelitian ditulis oleh Muhammad Latif Mubarak pada tahun 2020. Bertujuan untuk memudahkan organisasi mahasiswa agar bisa mengajukan proposal kegiatan dan juga memudahkan LPMA untuk mengecek kembali dan menyetujui hasil laporan tersebut. Hasil dari penelitian ini adalah memudahkan organisasi kesiswaan dan LPMA dalam mengajukan proposal dan mengecek serta menyetujui hasil laporan di masing-masing kegiatan.. Pengembangan sistem ini menggunakan metode waterfall dan pengujian menggunakan black box testing [5].

Penelitian ditulis oleh Anita Muliawati dan Bambang Tri Wahyono pada tahun 2020. Hasil pada peneliti ini siswa dapat melakukan pendaftaran secara online sehingga memudahkan siswa untuk mengikuti kegiatan ekstrakurikuler yang diminati dan sistem dapat merekap dan mengelola data anggota, sistem dapat memberitahukan absensi per tiap tanggal, menampilkan nilai siswa pada masing-masing kegiatan ekstrakurikuler agar pembina tidak mengalami Pengembangan sistem ini menggunakan metode waterfall, pengujian

black box testing dengan bahasa pemrograman PHP dan MYSQL sebagai database [6].

Penelitian lainnya ditulis oleh Rizki Rafdiansyah dkk, pada tahun 2017. Sistem ini bertujuan untuk membantu pengurus ekstrakurikuler dalam memonitoring hasil kegiatan serta usulan proposal kegiatan yang memudahkan pembina dalam mengunggah hasil proposal kepada kesiswaan sehingga nantinya akan di cek oleh kesiswaan apakah usulan proposal kegiatan dapat disetujui atau masih dalam revisi. Pengembangan sistem ini menggunakan metode waterfall dengan bahasa pemrograman PHP dan MYSQL sebagai database [7].

2.2 Landasan Teori

Landasan teori berisi hal-hal atau teori-teori yang berkaitan dengan permasalahan dan ruang lingkup permasalahan sebagai landasan dalam pembuatan laporan ini.

2.2.1 Sistem Informasi

Sistem adalah kumpulan orang yang saling bekerja sama dengan ketentuan-ketentuan aturan yang sistematis dan terstruktur untuk membentuk satu kesatuan melaksanakan suatu fungsi untuk mencapai tujuan. Sistem memiliki beberapa karakteristik atau sifat yang terdiri dari komponen sistem, batasan sistem, lingkungan luar sistem, penghubung sistem, masukan sistem, keluaran sistem, pengolahan sistem dan sasaran sistem [8].

Informasi adalah data yang diolah menjadi lebih berguna dan berarti bagi penerimanya dan untuk mengurangi ketidakpastian dalam proses pengambilan keputusan mengenai suatu keadaan [8].

Sistem informasi merupakan suatu cara tertentu untuk menyediakan informasi yang dibutuhkan oleh organisasi untuk beroperasi dengan cara yang sukses dan untuk organisasi bisnis dengan cara yang menguntungkan. Pada dasarnya sistem informasi merupakan suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi. Sistem informasi di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan [9]

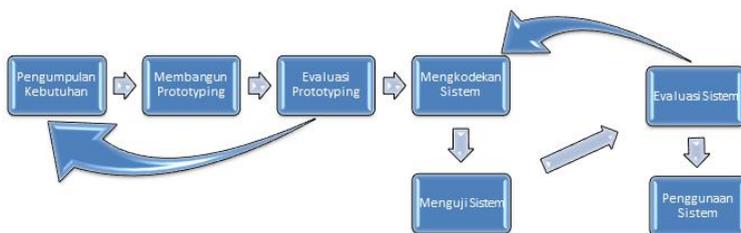
2.2.2 Rekayasa Web

Rekayasa Web merupakan pembangunan dengan menggunakan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi yang dipercaya dan bekerja secara efisien menggunakan mesin.

Berikut adalah metode dan tools yang digunakan :

1. Metode

Metode dapat diartikan sebagai menyediakan cara bagaimana secara teknis membangun perangkat lunak yang harus berada pada sebuah komitmen dasar menuju kualitas. Metode yang digunakan pada penelitian ini adalah metode *prototype*. Metode prototype dapat dilihat pada Gambar 2.1



Gambar 2.1 Pemodelan Prototype

Adapun tahapan-tahapannya yaitu :

a. Analisis Kebutuhan

Tahap ini merupakan tahap awal dalam metode prototype dimana klien dan developer saling bertemu untuk menentukan kebutuhan dan tujuan awal.

b. Membangun Prototyping

Pada tahap kedua ini, proses membangun prototype dilakukan jika data dan kebutuhan telah terkumpul.

c. Evaluasi Prototyping

Pada tahap selanjutnya, evaluasi ini dilakukan untuk mengetahui apakah prototyping sudah sesuai dengan kebutuhan yang diinginkan. Ketika sistem sudah sesuai dengan yang

diinginkan maka dapat dilanjutkan ke tahap selanjutnya, jika belum maka harus mengulangi ke tahap sebelumnya.

d. Pengkodean Sistem

Tahap selanjutnya pengkodean, hasil pembuatan prototype yang sudah sesuai dengan yang diinginkan akan diterjemahkan ke dalam Bahasa pemrograman

e. Pengujian Sistem

Setelah dilakukan pengkodean sistem, maka akan dilakukan pengujian. Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, harus dites dahulu sebelum digunakan. Pengujian ini dilakukan dengan Black Box.

f. Evaluasi Sistem

Setelah itu, pelanggan mengevaluasi apakah sistem yang sudah jadi sesuai dengan yang diharapkan. Jika sudah, maka sistem dapat dilanjutkan ke tahap terakhir. Jika belum maka diulang ke tahap pengkodean dan pengujian sistem.

g. Menggunakan Sistem

Tahap terakhir ini, jika perangkat lunak sudah diuji dan disetujui maka sistem siap dan layak untuk digunakan.

Terkait dengan pengujian program, pengujian yang akan digunakan adalah pengujian *black-box*. Pengujian *black-box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black-box* memungkinkan rekayasa perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian *black-box* bukan merupakan alternatif dari teknik *white-box*, tetapi merupakan pendekatan komplementer yang memungkinkan besar mampu mengungkap kelas kesalahan daripada metode *white-box*. Pengujian *black-box* berusaha menemukan kesalahan dalam kategori sebagai berikut [10] :

1. Fungsi-fungsi yang tidak benar atau hilang,
2. Kesalahan *interface*,
3. Kesalahan dalam struktur data atau akses *database* eksternal,
4. Kesalahan kinerja,
5. Inisialisasi dan kesalahan terminal.

Tidak seperti pengujian *white-box*, yang dilakukan pada saat awal proses pengujian, pengujian *black-box* cenderung diaplikasikan selama tahap akhir pengujian. Karena pengujian *black-box* memperhatikan

struktur struktur kontrol, maka perhatian berfokus pada domain informasi.

Pengujian di desain untuk menjawab pertanyaan-pertanyaan berikut [11].

1. Bagaimana validitas fungsional diuji?
 2. Kelas *input* apa yang akan *test case* menjadi baik?
 3. Apakah sistem sangat sensitif terhadap harga *input* tertentu?
 4. Bagaimana batasan dari suatu data diisolasi?
 5. Kecepatan data apa dan volume data apa yang dapat ditolehir oleh sistem?
 6. Apa pengaruh kombinasi tertentu dari data terhadap operasi sistem
- Pengujian *black-box* ini terdapat beberapa proses. Proses-proses yang ada dalam pengujian ini diantaranya :
- a. Menganalisa kebutuhan dan spesifikasi dari perangkat lunak.
 - b. Pemilihan jenis *input* yang memungkinkan menghasilkan *output* dengan benar serta jenis *input* yang memungkinkan *output* salah pada perangkat lunak yang sedang diuji.
 - c. Menentukan *output* untuk satu jenis *input*.
 - d. Pengujian dilakukan dengan *input-input* yang telah benar-benar diseleksi.
 - e. Melakukan pengujian
 - f. Perbandingan *output* yang dihasilkan dengan *output* yang diharapkan.
5. Pemeliharaan (*maintenance*)
- Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Pada tahap ini dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

B. Tools/ Alat Bantu Penelitian

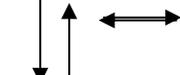
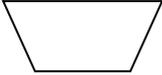
1. Flowchart

Flowchart adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir (*flowchart*) digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi[12]. Ada beberapa jenis-jenis *flowchart* diantaranya :

- A. Bagan alir sistem (*system flowchart*)
- B. Bagan alir dokumen (*document flowchart*)
- C. Bagan alir skematik (*schematic flowchart*)
- D. Bagan alir program (*program flowchart*)
- E. Bagan alir proses (*process flowchart*)

Simbol-simbol dalam *flowchart* dapat dilihat pada Table 2.1.

Table 2. 1 Simbol Flowchart

No	Simbol	Keterangan
1	 <p>Flow Direction Symbol</p>	Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga connecting line.
2	 <p>Terminator Symbol</p>	Yaitu simbol untuk permulaan (<i>start</i>) atau akhir (<i>end</i>) dari suatu kegiatan.
3	 <p>Processing Symbol</p>	Yaitu simbol yang menunjukkan pengolahan yang dilakukan oleh komputer.
4	 <p>Manual Operation Symbol</p>	Yaitu simbol yang menunjukkan pengolahan data yang tidak dilakukan oleh komputer.
5	 <p>Decision Symbol</p>	Yaitu simbol untuk pemilihan proses berdasarkan kondisi yang ada.
5	 <p>Input-Output Symbol</p>	Yaitu simbol yang menyatakan proses input dan output tanpa tergantung dari jenis peralatannya.

6	 <p style="text-align: center;">Document Symbol</p>	Yaitu simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.
---	---	--

2. Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah hasil kerja dari konsorium berbagai organisasi yang berhasil dijadikan sebagai standar baku dalam *Object Oriented Analysis* dan *Design* (OOAD). Macam-macam dari *Unified Modeling Language* (UML) antara lain [11] : *use case* diagram, *sequence* diagram dan *class* diagram.

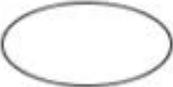
a. Use Case

Use Case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use Case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem tersebut *scenario* sedangkan pengguna disebut *actor*. *Actor* adalah sebuah peran yang biasa dimainkan oleh pengguna dalam interaksinya dengan sistem. Model *use case* adalah bagian dari model *requirement*. Definisi lain *use case* adalah abstraksi dari interaksi antara sistem dan *actor*. *Use case* dibuat berdasarkan keperluan *actor*.

Berdasarkan definisi diatas maka dapat disimpulkan bahwa *Use Case* adalah kontruks untuk mendeskripsikan bagaimana sistem akan terlihat dimata pengguna potensial yang terdiri dari sekumpulan *scenario* dan *actor*. Sedangkan *use case* diagram memfasilitasi komunikasi diantara analis dan pengguna serta analis dan klien. Simbol *Use Case* dapat dilihat pada Table 2.2.

Table 2. 2 Simbol *Use Case Diagram*

No.	Simbol	Nama Simbol dan Keterangan
1.		<i>Actor</i> , Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2.		<i>Dependency</i> , Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri(independent).

3.		<i>Generalization</i> , Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (ancestor).
4.		<i>Include</i> , Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
5.		<i>Extend</i> , Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6.		<i>Association</i> , yang menghubungkan antara objek satu dengan objek lainnya
7.		<i>System</i> , Menspesifikasikan paket yang menampilkan sistem secara terbatas
8.		<i>Use Case</i> , Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor.
9.		<i>Collaboration</i> , Interaksi aturan-aturan dan elemen lain yang bekerjasama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10.		<i>Note</i> , Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

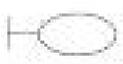
b. *Sequence Diagram*

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah *scenario*. Diagram ini menunjukkan sejumlah obyek dan *message* yang diletakan antara obyek-obyek didalam *use case*.

Komponen utama *sequence diagram* terdiri atas objek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress*

vertical. Sequence Diagram menambahkan dimensi waktu pada interaksi diantara obyek. Simbol-simbol yang dipakai dalam pembuatan *sequence diagram* dapat dilihat pada Table 2.3.

Table 2. 3 Simbol *Sequence Diagram*

No	Simbol	Keterangan
1		ACTOR Menggambarakan orang yang sedang berinteraksi dengan sistem
2		ENTITY CLASS Menggambarakan beberapa kegiatan yang akan dilakukan
3		BOUNDARY CLASS Menggambarakan sebuah programisasi dari form
4		CONTROL CLASS Menggambarakan penghubung antara boundary dengan tabel
5		A FOCUS OF CONTROL & A LIFE LINE Menggambarakan tempat mulai dan berakhirnya sebuah message
6		A MESSAGE Menggambarakan pengiriman pesan

c. *Pemrograman Berorientasi Objek*

Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan proyek yang berisi data dan operasi yang diperlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. [14]

Sistem berorientasi objek merupakan sebuah sistem yang dibangun dengan berdasarkan metode berorientasi objek adalah sebuah sistem yang komponennya dibungkus menjadi kelompok data dan fungsi.

Setiap komponen dalam sistem tersebut dapat mewarisi atribut, sifat, komponen lainnya, dan dapat berinteraksi satu sama lain.

2.2.3 Basis Data

Basis data adalah kumpulan file-file yang mempunyai kaitan antara satu file dengan file lain sehingga membentuk satu bangun data untuk menginformasikan suatu perusahaan instansi, dalam bahasan tertentu [13].

MySQL adalah sebuah program database server yang mampu menerima dan mengirimkan datanya dengan sangat cepat, multi *user* serta menggunakan perintah standar *Structured Query Language* (SQL). MySQL memiliki dua bentuk lisensi, yaitu *Free Software* dan *Shareware*. MySQL yang biasa kita gunakan adalah *MySQL Free Software I* yang berada dibawah Lisensi *General Public License* (GPL)^[15].

MySQL juga dapat didefinisikan sebagai sebuah database server, dapat juga berpersan sebagai *client* sehingga sering disebut *database client / server* yang *open source* dengan kemampuan dapat berjalan baik di Operasi Sistem maupun dengan *Platform Windows* maupun *Linux*.

MySQL dikembangkan oleh sebuah perusahaan Swedia bernama MySQL AB, yang pada saat itu bernama TcX Data Konsult AB sekitar tahun 1994-1995. MySQL sudah ada sejak 1979. MySQL termasuk jenis *Relation Database Management System* (RDBMS) digunakan oleh banyak portal-portal internet sebagai basis data dari informasi yang ditampilkan pada situs *web*. Kepopuleran MySQL dimungkinkan karena kemudahannya untuk digunakan, cepat secara kinerja *query*, dan mencukupi untuk kebutuhan basis data perusahaan-perusahaan skala menengah dan kecil. Istilah seperti tabel, baris, dan kolom tetap digunakan dalam MySQL. Sebuah basis data yang terdapat pada MySQL mengandung satu atau beberapa tabel yang terdiri dari sejumlah baris dan kolom.

MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah suatu bahasa (*language*) yang digunakan untuk mengakses data di dalam sebuah database relasional. *SQL* sering juga disebut dengan istilah *query*, dan bahasa *SQL* secara praktiknya digunakan sebagai bahasa standar untuk manajemen database relasional. Hingga saat ini hampir seluruh server database atau software database mengenal dan mengerti bahasa *SQL*. Dalam penggunaan *SQL* terdapat beberapa perintah yang berguna untuk mengakses dan memanajemen data yang

terdapat dalam database. Secara garis besar, *SQL Server* mempunyai 3 (Tiga) jenis perintah *SQL* yaitu :

1.) *Data Definition Language (DDL)*

DDL adalah sub perintah dari bahasa SQL yang digunakan untuk membangun kerangka sebuah database, dalam hal ini database dan table. Terdapat tiga perintah penting dalam DDL, yaitu :

- a) **CREATE**: perintah ini digunakan untuk membuat, termasuk di dalamnya membuat database baru, tabel baru view baru, dan kolom baru. Contoh: `CREATE DATABASE nama_database.`
- b) **ALTER**: perintah ALTER berfungsi untuk mengubah struktur tabel yang telah dibuat. Mencakup di dalamnya mengubah nama tabel, menambah kolom, mengubah kolom, menghapus kolom, dan memberikan atribut pada kolom. Contoh: `ALTER TABLE nama_tabel ADD nama_kolom datatype.`
- c) **DROP**: perintah DROP berfungsi untuk menghapus database atau tabel. Contoh: `DROP DATABASE nama_database.`

2) *Data Manipulation Language (DML)*

DML adalah sub perintah dari bahasa SQL yang digunakan untuk memanipulasi data dalam database yang telah dibuat. Terdapat 4 (Empat) perintah penting dalam DML, yaitu :

- a) **INSERT**: perintah ini digunakan untuk memasukkan data baru ke dalam sebuah tabel. Perintah ini tentu saja bisa dijalankan ketika database dan tabel sudah dibuat. Contoh: `INSERT INTO nama_tabel VALUES (data1, data2, dst...);`
- b) **SELECT**: perintah ini digunakan untuk mengambil dan menampilkan data dari tabel atau bahkan dari beberapa tabel dengan penggunaan relasi. Contoh: `SELECT nama_kolom1, nama_kolom2 FROM nama_tabel;`
- c) **UPDATE**: perintah update digunakan untuk memperbaharui data pada sebuah tabel. Contoh: `UPDATE nama_tabel SET kolom1=data1, kolom2=data2,... WHERE kolom=data;`
- d) **DELETE**: perintah delete digunakan untuk menghapus data dari sebuah tabel. Contoh: `DELETE FROM nama_tabel WHERE kolom=data;`

3.) *Data Control Language (DCL)*

DCL adalah sub bahasa SQL yang berfungsi untuk melakukan pengontrolan data dan server databasenya, seperti manipulasi user dan

hak akses (priviledges). Yang termasuk perintah dalam DCL ada 2 (Dua), yaitu :

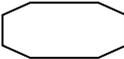
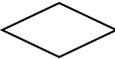
- a) GRANT: perintah ini digunakan untuk memberikan hak akses oleh admin ke salah satu user atau pengguna. Hak akses tersebut bisa berupa hak membuat (CREATE), mengambil data (SELECT), menghapus data (DELETE), mengubah data (UPDATE), dan hak khusus lainnya yang berhubungan dengan sistem database.
- b) REVOKE: perintah ini digunakan untuk mencabut hak akses yang telah diberikan kepada user. Dalam ini merupakan kebalikan dari perintah GRANT[14].

4.) *Entity Relationship Diagram (ERD)*

ERD adalah suatu rancangan atau bentuk hubungan suatu kegiatan di dalam sistem yang berkaitan langsung dan mempunyai fungsi di dalam proses tersebut. ERD adalah suatu pemodelan dari basis data relasional yang didasarkan atas persepsi di dalam dunia nyata, dunia ini senantiasa terdiri dari sekumpulan objek yang saling berhubungan antara satu dengan yang lainnya. Suatu objek disebut *entity* dan hubpeungan yang dimilikinya disebut *relationship*. Suatu *entity* bersifat unik dan memiliki atribut sebagai pembeda dengan entity lainnya [17].

Berikut merupakan simbol-simbol dari ERD pada Table 2.4 :

Table 2. 4 Simbol ERD

No.	Nama	Simbol	Keterangan Fungsi
1.	Entitas		Persegi panjang menyatakan himpunan entitas adalah orang, kejadian, atau berada.
2.	Atribut		Atribut merupakan informasi yang diambil tentang sebuah entitas.
3.	Relasi		Belah ketupat menyatakan himpunan relasi merupakan hubungan antar entitas.

4.	Link		Garis sebagai penghubung antara himpunan, relasi, dan himpunan entitas dengan atributnya
----	------	---	--

ERD memiliki derajat relasi atau biasa disebut kardinalitas. Kardinalitas menjelaskan batasan jumlah keterhubungan satu entity dengan entity lainnya.

Table 2. 5 Macam-Macam *Kardinalitas*

No.	Simbol	Nama	Keterangan
1.		Relasi Satu ke Satu (<i>One to One</i>)	Relasi yang menunjukkan bahwa setiap himpunan entitas berhubungan dengan tepat satu himpunan entitas lainnya
2.		Relasi Satu ke Banyak (<i>One to Many</i>)	Relasi yang menunjukkan bahwa hubungan antara entitas pertama dengan entitas kedua adalah satu banding banyak, begitu pula sebaliknya
3.		Relasi Banyak ke Banyak (<i>Banyak to Many</i>)	Relasi yang menunjukkan bahwa setiap himpunan entitas boleh berhubungan dengan banyak himpunan entitas lainnya dan sebaliknya

2.2.4 Pendidikan

Pendidikan adalah suatu proses belajar mengajar dari sekelompok orang untuk dapat memperoleh suatu pengetahuan, keterampilan serta proses perubahan sikap dan tata laku seseorang dalam usaha mendewasakan manusia melalui upaya pengajaran dan pelatihan. Didalam lingkungan sekolah, untuk dapat mengembangkan

keterampilan serta wawasan yang luas maka terdapat suatu organisasi yang dapat memunculkan ide atau karya-karyanya melalui ekstrakurikuler. Ekstrakurikuler merupakan suatu kegiatan yang dilakukan oleh para siswa sekolah diluar jam belajar kurikulum standar. Ekstrakurikuler sebagai wadah pengembangan potensi peserta didik, dapat memberikan dampak positif dalam penguatan pendidikan karakter.

Peserta didik dapat mengembangkan karakter dengan bersikap mandiri kreatif, bergotong royong, bernalar kritis, dan berakhlak mulia. Salah satu pendidikan yang memiliki kewajiban untuk menyelenggarakan kegiatan ekstrakurikuler sebagai wahana memfasilitasi pengembangan bakat dan minat peserta didik. Oleh sebab itu, kegiatan ekstrakurikuler harus dikelola secara sistematis dan terpolo agar bermuara pada pencapaian tujuan yang dimaksud. Agar dapat menyusun dan mengembangkan kegiatan ekstrakurikuler yang tersistem dan terpolo, sekolah perlu memahami cara dan tahapan diperlukan panduan yang dapat membimbing satuan pendidikan dalam menyelenggarakannya.

Ekstrakurikuler terbagi menjadi 2 yaitu ekstrakurikuler wajib dan tidak wajib. Ekstrakurikuler wajib adalah kegiatan ekstrakurikuler yang wajib diselenggarakan oleh satuan pendidikan dan wajib diikuti oleh seluruh peserta didik, contoh ekstra yang wajib di SMA Muhammadiyah adalah HW dan IPM. Sedangkan ada pula ekstrakurikuler pilihan adalah kegiatan ekstrakurikuler yang dapat dikembangkan dan diselenggarakan oleh satuan pendidikan dan dapat diikuti oleh peserta didik sesuai bakat dan minatnya masing-masing.

Ekstrakurikuler di SMA Muhammadiyah memiliki fungsi :

1. Pengembangan, yaitu sebagai wahana pengembangan minat dan bakat peserta didik.
2. Sosial, yaitu sebagai wahana untuk memperluas pengalaman bersosialisasi, praktik keterampilan berkomunikasi dan internalisasi nilai-nilai karakter.
3. Rekreatif, yaitu dilakukan dalam suasana gembira dan menyenangkan, sehingga suasana ini menunjang proses perkembangan potensi/kemampuan personal peserta didik.
4. Persiapan Karir, yaitu sebagai wahana memfasilitasi persiapan peserta didik melalui pengembangan bakat dan minat dalam bidang ekstrakurikuler yang diminati.