Laporan Tugas Besar



Anggota :

Reggie M. Gunawan 2016730001

Michael N. Christian 2016730009

Billy Setiadi 2016730093

Teknik Informatika

Universitas Katolik Parahyangan

Bandung

**Kata Pengantar**

Terimakasih kepada Tuhan Yang Maha Esa karena telah memberikan kami kekuatan dan kemampuan untuk menyelesaikan tugas akhir ini dengan baik. Tanpa pertolongannya, kami tidak akan bias menyelesaikan tugas ini dengan maksimal.

Laporan ini disusun demi penyelesaian tugas akhir mata kuliah Pemrograman Pada Perangkat Bergerak. Kami menyadari bahwa hasil karya ini masih jauh dari kata sempurna maka dari itu kritik dan saran sangat diharapkan demi kebaikan karya kami ke depannya.

Akhir kata, kami ucapkan terimakasih atas perhatiannya dan semoga karya ini berguna bagi pembelajaran banyak pihak.

**Deskripsi Aplikasi**

Nama aplikasi : Gyro Ball

Bahasa Pemrograman : Java

Aplikasi pengembangan : Android Studio

Penjelasan singkat :

Aplikasi Gyro Ball ini adalah sebuah game di platform android yang dimainkan dengan menggunakan sensor kemiringan. Pemain harus memiringkan bola agar bola mengarah ke sebuah lubang berwarna hitam dan mendapatkan poin sebanyak-banyaknya dalam rentan waktu tertentu.

Fitur dasar :

1. Gerakan bola diatur oleh kemiringan handphone
2. Penempatan bola sasaran dan bola yang bergerak diacak di awal permainan
3. Mencatat waktu dari mulai sampai permainan selesai.
4. Menampilkan score di akhir permainan yang dihitung dari waktu dan ketentuan lain
5. Perpindahan halaman menggunakan fragment
6. Dibuat pada struktur kelas yang relatif benar

Langkah Permainan :

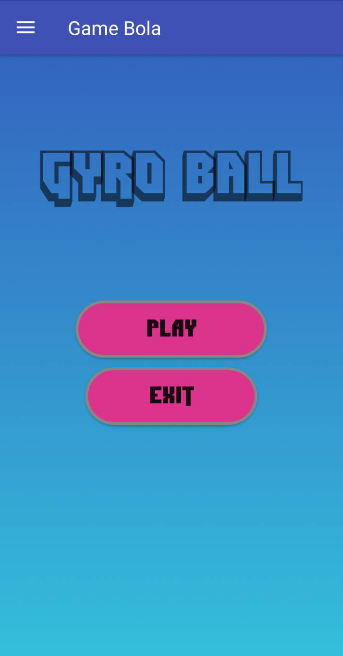
1. Pada home game, terdapat 2 tombol yaitu New Game dan Exit, dan navigation drawer yang berisi Game, Setting, dan Exit.
2. Pemain menekan tombol New Game untuk memulai permainan yang akan diarahkan ke halaman game.
3. Pada halaman game terdapat 2 tombol yaitu PLAY dan EXIT.
4. Pemain menekan tombol PLAY dan akan muncul 2 bola di area kanvas permainan, bola hitam sebagai target dan bola biru sebagai objek yang bergerak.
5. Pemain harus memiringkan handphone untuk mengarahkan bola biru ke target untuk mendapatkan point sebanyak-banyaknya sebelum waktu habis.
6. Setelah waktu habis, akan muncul pesan yang menunjukkan waktu telah habis dan skor terakhir pemain.
7. Jika ingin melihat papan high score, pemain tinggal meng-swipe atau menekan tombol navigation lalu memilih opsi High Score.
8. Pemain menekan tombol EXIT untuk mengakhiri permainan dan keluar dari aplikasi.

Fitur Tambahan (berhasil) :

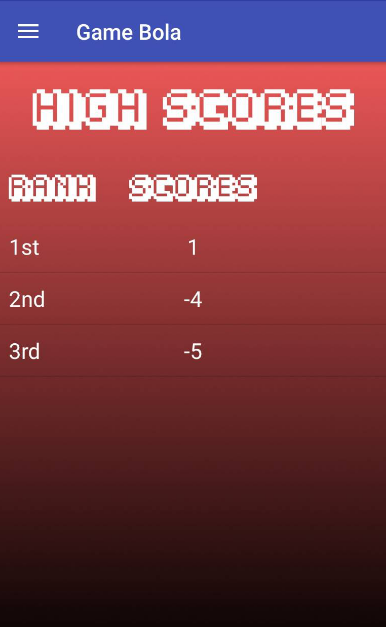
1. Membuat navigation drawer.
2. Percepatan bola diatur oleh kemiringan handphone.
3. Bola memantul jika mengenai ujung layar.
4. Ada objek lain dalam kanvas yang jika dilewati dapat mengurangi skor pemain.
5. Ada lebih dari satu bola yang dapat dikontrol pemain.

**Desain Tampilan Aplikasi**

Menu Home



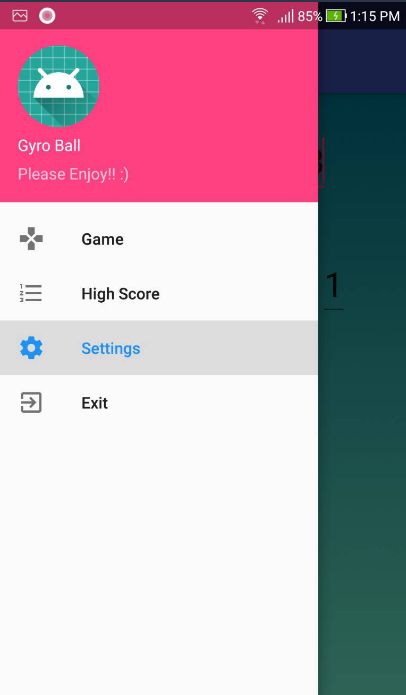
Menu High Score



Menu Setting

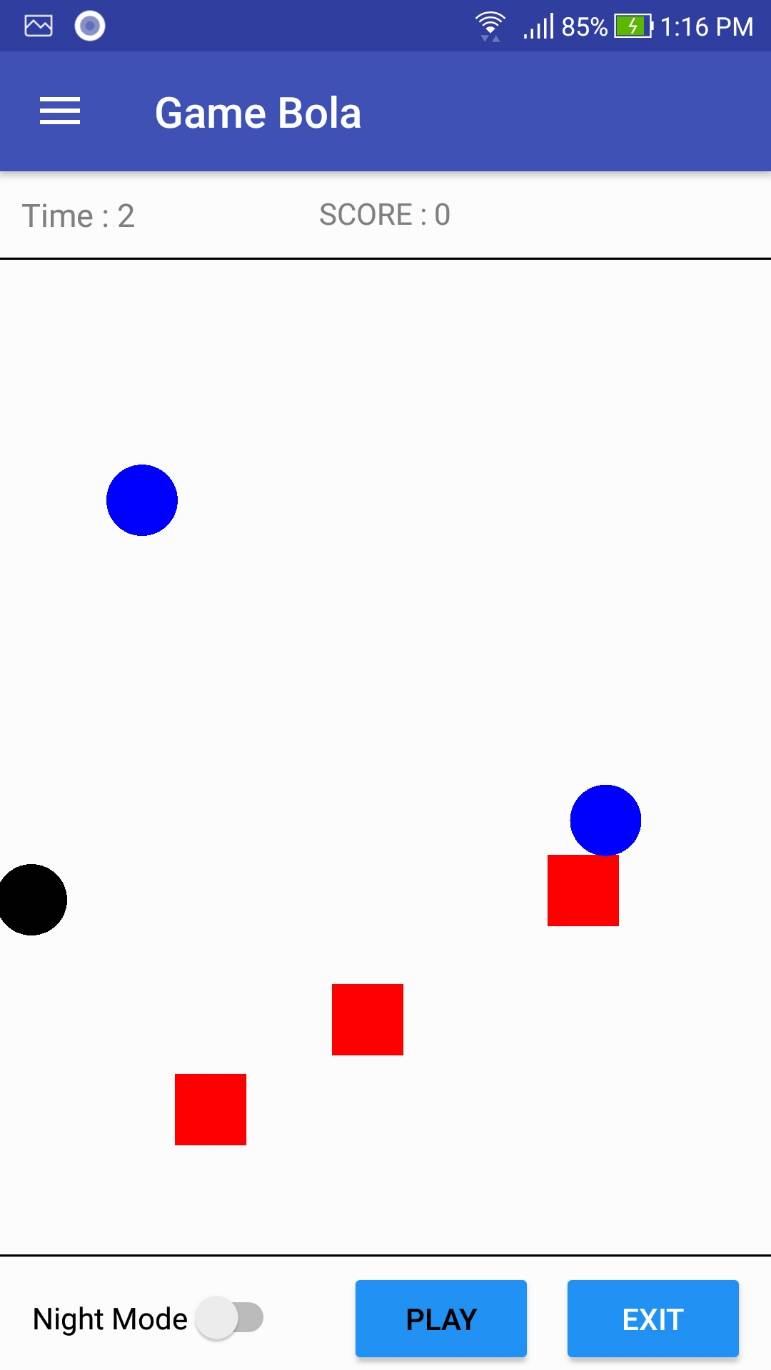


Navigation Drawer



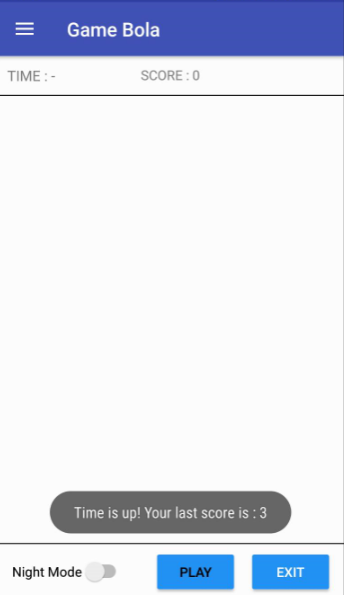
Menu awal permainan

Permainan setelah tombol PLAY ditekan



Ket : Bola biru adalah bola yang dapat bergerak sedangkan bola hitam adalah bola tujuan. Kotak merah adalah obstacle yang dapat mengurangi score.

Setelah permainan selesai



**Desain Struktur Kelas**

Pada tugas besar ini, kami mengimplementasikan kelas-kelas dengan menggunakan arsitektur Model-View-Presenter (MVP).

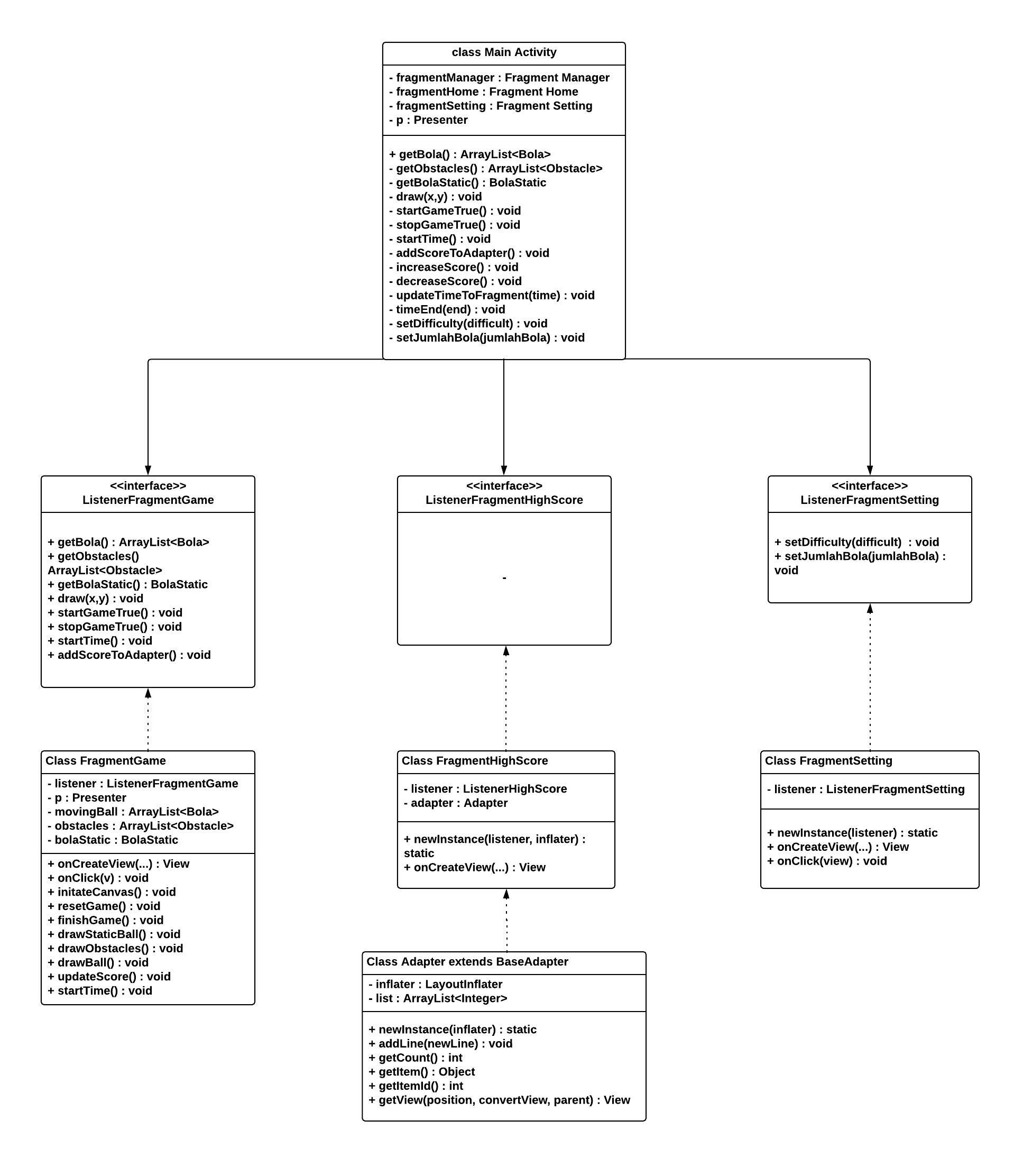
Tujuan : Memudahkan kelas-kelas untuk bertukar data agar aplikasi berjalan dengan optimal dan mengurangi beban View.

Tabel Kelas

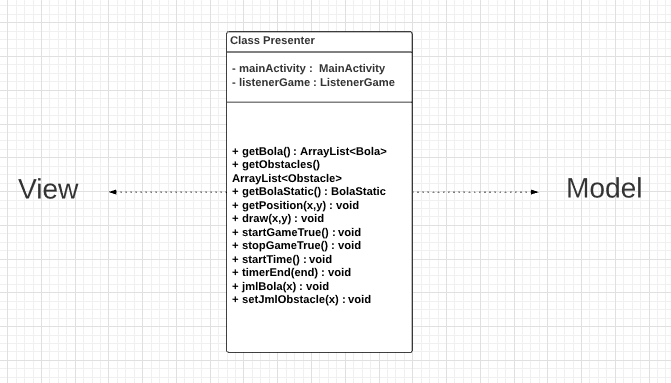
|  |  |  |
| --- | --- | --- |
| View | Presenter | Model |
| -Main Activity  -FragmentGame  -FragmentHighScore  -FragmentSetting  -ListenerFragmentGame  -ListenerFragmentSetting  -ListenerFragmentHighScore  -Adapter | Presenter | -Game  -ListenerGame  -Bola  -BolaStatic  -Obstacle  -SensorReader  -Timer |

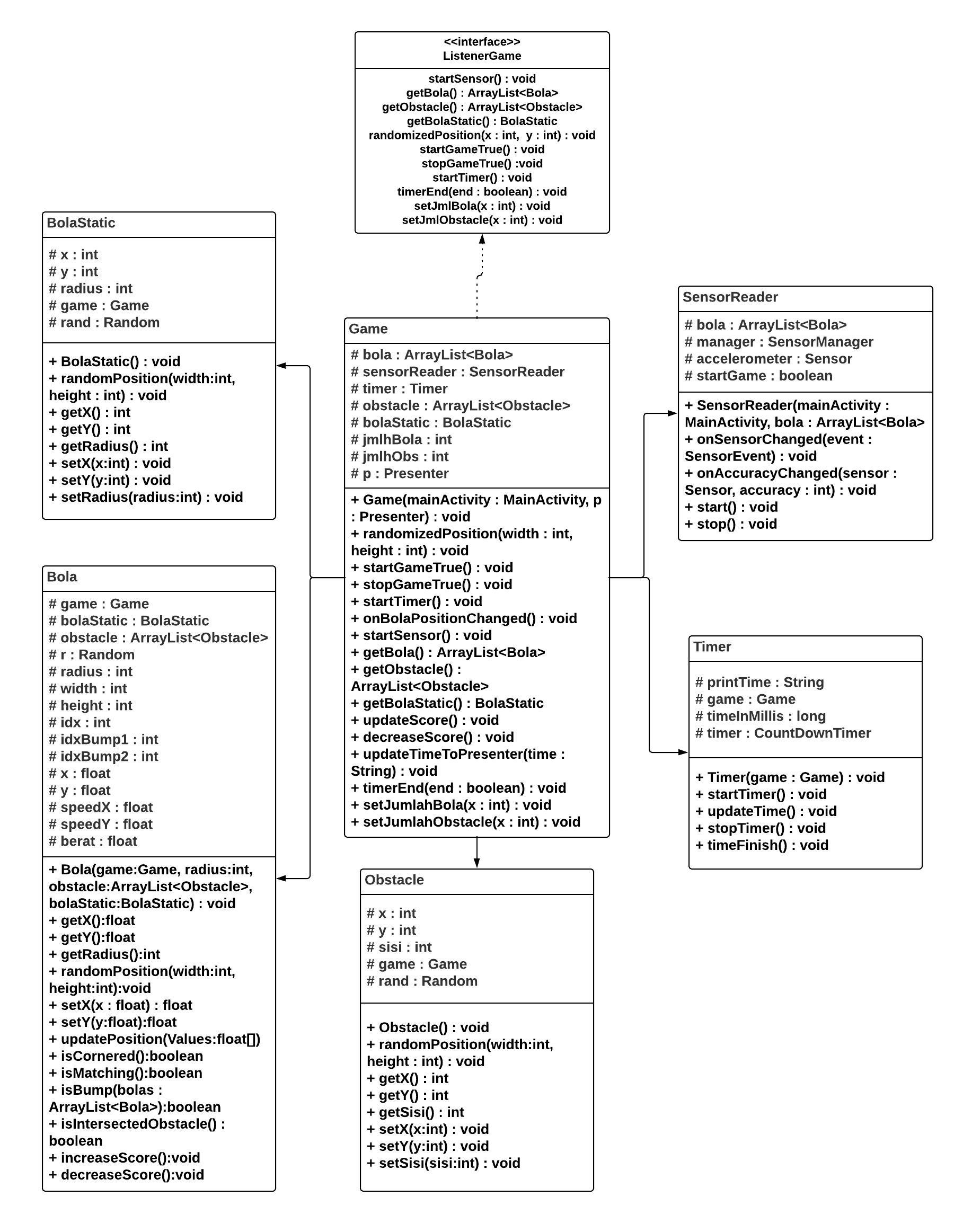
**Diagram Kelas**

View



Presenter



Model

**Penjelasan Kelas & Method**

View

|  |  |
| --- | --- |
| Main Activity | |
| Tujuan : Mengatur segala sesuatu yang akan ditampilkan ke layer | |
| Method | Deskripsi |
| view OnCreate(..) | Membuat activity |
| void OnClick(view) | Menghandle button mana yang di tekan |
| ArrayList<Bola> getBola() | Return bola dari Presenter |
| ArrayList<Obstacle> getObstacle() | Return obstacle dari Presenter |
| BolaStatic getBolaStatic() | Return bola static dari Presenter |
| void draw(x, y) | Menggambar object ke FragmentGame |
| void startGameTrue() | Memulai game ke Presenter |
| void stopGameTrue() | Menghentikan pergerakan bola |
| void startTimer() | Memulai timer waktu ke Presenter |
| void addScoreToAdapter(x) | Membantu pertukaran data dari FragmentGame ke FragmentHighScore |
| void increaseScore() | Menambahkan atribut score di FragmentHighScore |
| void decreaseScore() | Mengurangi atribut score di FragmentHighScore |
| void updateTimeToFragment(time) | Perantara dari Presenter ke FragmentGame untuk mengupdate waktu |
| void timerEnd(end) | Menghentikan waktu ke FragmentGame |
| void setDifficulty(difficulty) | Set jumlah obstacle ke Presenter |
| void setJumlahBola(jumlahBola) | Set jumlah bola yang akan dimainkan ke Presenter |
| void onBackPressed() | Method override dari Navigation Drawer |
| boolean onNavigationItemSelected(items) | Method override dari Navigation Drawer |
| void BuatTabel() | Membuat database untuk high score |
| void changeToGame() | Mengubah fragment ke FragmentGame |

|  |  |
| --- | --- |
| Class FragmentGame | |
| Tujuan : Mengatur gambar permainan | |
| Method | Deskripsi |
| view onCreateView(..) | Membuat fragment |
| static newInstance(listener, p) | Membuat kelas langsung jadi saat program dijalankan |
| void onClick(v) | Menghandle button PLAY dan EXIT saat ditekan |
| void initiateCanvas() | Menyiapkan canvas awal untuk digambar |
| void resetGame() | Mengembalikan keadaan game ke kondisi awal |
| void finishGame() | Menghentikan permainan dan menambah skor terakhir pemain ke database |
| void drawStaticBall() | Menggambar bola tujuan |
| void drawObstacle() | Menggambar banyaknya obstacle di layer |
| void drawBall() | Menggambar bola yang bergerak |
| void updateScore() | Mengupdate score secara berkala di label Score |
| void startTime() | Memulai timer saat button PLAY ditekan |

|  |  |
| --- | --- |
| Class FragmentHighScore | |
| Tujuan : Menampilkan daftar score pemain | |
| Method | Deskripsi |
| view onCreateView(..) | Membuat fragment |
| static newInstance(listener, inflater) | Membuat kelas langsung jadi saat program dijalankan |

|  |  |
| --- | --- |
| Class FragmentSetting | |
| Tujuan : Menampilkan menu pengaturan permainan | |
| Method | Deskripsi |
| view onCreateView(..) | Membuat fragment |
| static newInstance(listener) | Membuat kelas langsung jadi saat program dijalankan |
| void onClick(v) | Mengimplementasi pengaturan permainan saat tombol APPLY ditekan |

|  |  |
| --- | --- |
| interface ListenerFragmentGame | |
| Tujuan : Menyambungkan kelas FragmentGame ke MainActivity | |
| Method | Deskripsi |
| view onCreateView(..) | Membuat fragment |
| static newInstance(listener, inflater) | Membuat kelas langsung jadi saat program dijalankan |
| ArrayList<Bola> getBola() | Mendapatkan bola dari MainActivity |
| ArrayList<Obstacle> getObstacle() | Mendapatkan obstacle dari MainActivity |
| BolaStatic getBolaStatic() | Mendapatkan bola static dari MainActivity |
| void draw(x, y) | Menggambar object ke dari MainActivity ke FragmentGame |
| void startGameTrue() | Memulai game ke MainActivity |
| void stopGameTrue() | Menghentikan pergerakan bola dari MainActivity |
| void startTimer() | Memulai timer waktu ke MainActivity |
| void addScoreToAdapter(x) | Membantu pertukaran data dari FragmentGame ke FragmentHighScore |

|  |
| --- |
| interface ListenerHighScore |
| Tujuan : Menyambungkan kelas FragmentHighScore dengan MainActivity |
| Method & Deskripsi |
| Kelas ini kosong karena tidak terdapat pertukaran data antar kelas. Sebenarnya ada kelas Adapter tetapi Adapter dibuat menggunakan newInstance sehingga sudah terbuat saat MainActivity dimulai. Penambahan score ke Adapter pun dilakukan via MainActivity. |

|  |  |
| --- | --- |
| interface ListenerFragmentSetting | |
| Tujuan : Menyambungkan kelas FragmentHighScore dengan MainActivity | |
| Method | Deskripsi |
| void setDifficulty(difficulty) | Mengantarkan nilai difficult dari FragmentSetting ke FragmentGame |
| void setJumlahBola(jumlahBola) | Mengantarkan nilai jumlahBola dari FragmentSetting ke FragmentGame |

|  |  |
| --- | --- |
| class Adapter | |
| Tujuan : Menampilkan nilai high score ke ListView di kelas FragmentHighScore | |
| Method | Deskripsi |
| static newInstance(inflater) | Membuat kelas saat MainActivity dijalankan dengan LayoutInflater milik MainActivity sbg parameternya. |
| void addLine(newLine) | Menambahkan nilai dari parameter newLine pada list |
| int getCount() | Mengembalikan jumlah nilai yang ada pada atribut list |
| Object getItem(position) | Mengembalikan object pada list di posisi ke-position |
| long getItemId(position) | Mengembalikan nilai 0 |
| view getView(position, convertView, parent) | Mengembalikan sebuah list baru yang sudah diisikan nilai pada atribut list di posisi ke-position |
| void descending() | Mengurutkan nilai pada list secara menurun dari paling besar ke kecil (descending) |

Presenter

|  |  |
| --- | --- |
| class Presenter | |
| Tujuan : Medium pertukaran data antar View dan Model | |
| Method | Deskripsi |
| ArrayList<Bola> getBola() | Return object bola ke MainActivity |
| ArrayList<Obstacle> getObstacle() | Return object obstacle ke MainActivity |
| BolaStatic getBolaStatic() | Return object bolaStatic ke MainActivity |
| void getPosition(x, y) | Merandom posisi bola dari FragmentGame ke kelas Game |
| void draw(x, y) | Menggambar object ke FragmentGame via MainActivity |
| void startGameTrue() | Memulai gambar bola ke SensorReader |
| void stopGameTrue() | Menghentikan gambar bola ke SensorReader |
| void startTimer() | Memulai timer waktu ke Timer |
| void increaseScore() | Menambahkan atribut score di FragmentHighScore dari Bola |
| void decreaseScore() | Mengurangi atribut score di FragmentHighScore dari Bola |
| void updateTimer(time) | Mengupdate timer scr berkala dari kelas Timer ke FragmentGame |
| void timerEnd(end) | Menghentikan waktu ke FragmentGame dari kelas Timer |
| void setJmlObstacle(x) | Set jumlah obstacle dari FragmentSetting ke kelas Obstacle |
| void jmlBola(x) | Set jumlah bola dari FragmentSetting ke kelas Bola |

Model

|  |  |
| --- | --- |
| class Bola | |
| Tujuan: Menyimpan nilai-nilai yang mengatur membuat bola permainan | |
| Method | Deskripsi |
| void randomPosition(width, height) | Mengacak nilai posisi bola yang digambar pada canvas. |
| void updatePosition(values) | Mengupdate posisi bola untuk digambar melalui nilai dari sensor x dan y-nya. |
| boolean isCornered() | Mengembalikan nilai true atau false untuk menyatakan jika bola sudah menabrak dinding atau belum. |
| boolean isMatching() | Mengembalikan nilai true atau false untuk menyatakan jika bola sudah melewati lubang atau belum. |
| boolean isIntersectObstacle() | Mengembalikan nilai true atau false untuk menyatakan jika bola menabrak kotak penghalang atau tidak. |
| boolean isBump(masih belum bekerja) | Mengembalikan nilai true atau false jika bola saling bertabrakan atau tidak. |
| void increaseScored() | Menambahkan skor nilai setiap kali bola melewati lubang. |
| void decreasedScored() | Mengurangi skor nila setiap kali bola menabrak kotak penghalan/obstacle. |

|  |  |
| --- | --- |
| class BolaStatic | |
| Tujuan: Menyimpan nilai-nilai untuk mengambar lubang atau bola yang diam sebagai tujuan | |
| Method | Deskripsi |
| void randomPosition(width,height) | Membuat nilai acak untuk menggambar posisi lubang di canvas. |

|  |  |
| --- | --- |
| class Obstacle | |
| Tujuan: Menyimpan nilai-nilai untuk menggambar bentuk kotak sebagai penghalang bola | |
| Method | Deskripsi |
| void randomPosition(width,height) | Membuat nilai acak untuk menggambar posisi kotak sebagai penghalang di canvas. |

|  |  |
| --- | --- |
| class SensorReader | |
| Tujuan: Menyimpan nilai-nilai sensor hp dimensi x,y dan z untuk mengatur pergerakan bola | |
| Method | Deskripsi |
| void onSensorChanged(values) | Menyimpan nilai x,y dan z dari membaca gerakan smartphone. |
| void start() | Memulai fungsi sensor untuk membaca gerakan smartphone. |
| void stop() | Mengehentikan fungsi sensor untuk membaca gerakan smartphone. |
| void onAccuracyChanged() | Hanya mengoverride method. |

|  |  |
| --- | --- |
| class Timer | |
| Tujuan: Menyimpan dan mengupdate terus waktu mundur permainan | |
| Method | Deskripsi |
| void startTimer() | Mengeset nilia timer menjadi 10 detik. |
| void updateTimer() | Mengupdate waktu agar berkurang dari 10-0 detik. |
| void stopTimer() | Menghentikan penghitungan waktu mundur. |
| void timerFinish() | Memanggil method timerFinish(). |
| void onTick(s) | Memanggil method updateTime() dan mengeset waktu. |
| void onFinish() | Mengehentikan waktu jika sudah mencapai 0 detik. |

|  |  |
| --- | --- |
| class Game | |
| Tujuan: Menggambungkan seluruh komponen class di model untuk membuat game | |
| Method | Deskripsi |
| void randomizedPosition(width,height) | Menggambar secara acak posisi bola pada canvas. |
| void startGameTrue() | Memberitahu jika permainan sudah dimulai maka sensor mulai membaca gerakan. |
| void stopGameTrue() | Memberitahu jika permainan dihentikan atau belum bermain maka sensor tidak dibaca. |
| void startTimer() | Memulia hitung mundur timer. |
| void onBolaPositionChanged() | Mengambar ulang setiap kali posisi bola diupdate. |
| void startSensor() | Memulai menyalakan fungsi sensor. |
| ArrayList<Bola> getBola() | Mengembalikan objek dalam array yang berisi bola. |
| ArrayList<Obstacle> getObstacle() | Mengembalikan objek dalam array yang berisi kotak. |
| BolaStatic getBolaStatic() | Mengembalikan objek bola sebagai lubang |
| void updateScore() | Menambahkan skor di presenter. |
| void decreaseScore() | Mengurangi skor di presenter. |
| void updateTimeToPresenter(time) | Mengupdate waktu ke presenter. |
| void timerEnd(end) | Mengehentikan timer di presenter. |
| void setJumlahBola(x) | Mengassign jumlah bola yang akan digambarkan. |
| void setJmlObstacle(x) | Menggassign jumlah kotak yang akan digambarkan. |

|  |  |
| --- | --- |
| Interface ListenerGame | |
| Tujuan: Menggabungkan kelas Game dengan kelas Presenter | |
| Method | Deskripsi |
| void startSensor() | Mengantarkan tanda memulai sensor dari FragmentGame ke SensorReader |
| ArrayList<Bola> getBola() | Mengantarkan objek array bola dari Game ke Presenter. |
| ArrayList<Obstacle> getObstacle() | Mengantarkan objek array obstacle(kotak) dari Game ke Presenter. |
| BolaStatic getBolaStatic(x,y) | Mengantarkan nilai lubang dari Game ke Presenter. |
| void randomizePosition() | Mengantarkan nilai posisi acak dari Game ke Presenter. |
| void startGameTrue() | Menyatakan jika game dimulai dari Game ke Presenter. |
| void stopGameTrue() | Menyatakan jika game dihentikan dari Game ke Presenter. |
| void startTimer() | Mengantarkan nilai timer dari Game ke Presenter. |
| void timerEnd(end) | Mengantarkan nilai boolean jika waktu berhenti. |
| void setJmlBola(x) | Mengantarkan nilai jumlah bola yang akan digambar dari Game ke Presenter. |
| void setJmlObstacle(x) | Mengentarkan nilai jumlah obstacle(kotak) yang akan digambar dari Game ke Presenter. |

**Referensi Tugas**

**-** Youtube : https://www.youtube.com

- Stack Overflow : https://stackoverflow.com

- Developer Android : https://developer.android.com

- Google : searching via google.com

- Dosen : bertanya via dosen pengajar maupun asisten dosen

- Lucid Chart (diagram kelas)