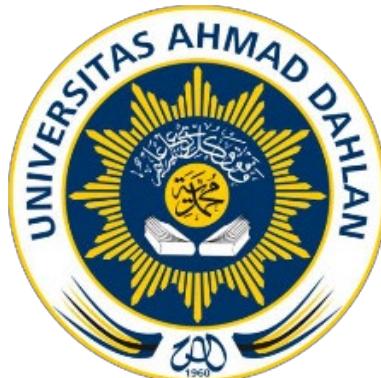


LAPORAN PRAKTIKUM

“Post Test Pertemuan ke-II”

Diajukan untuk memenuhi salah satu praktikum Mata Kuliah Matematika Diskrit yang diampu oleh:

Nur Rochmah Dyah PA, S.T., M.Kom.



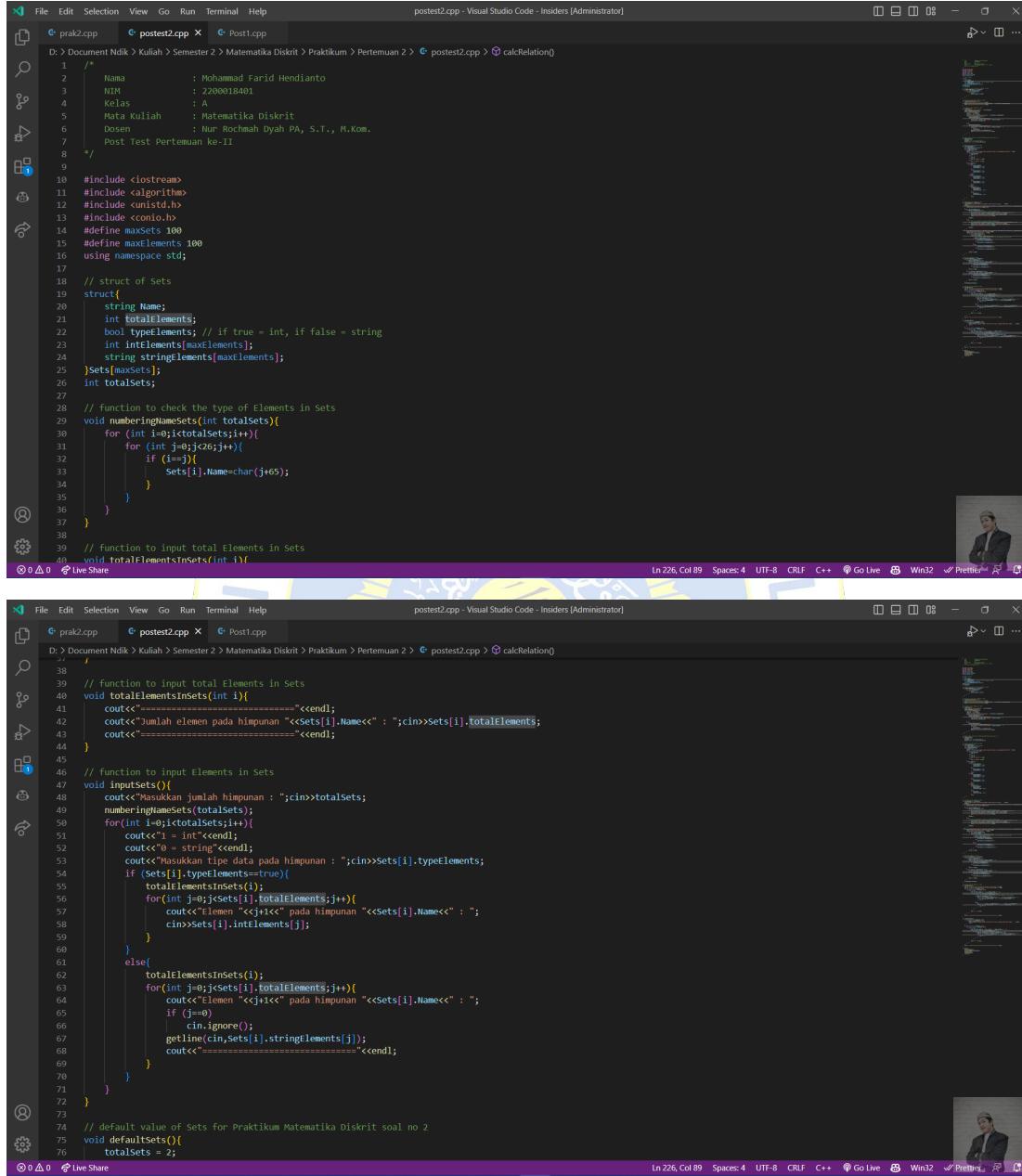
Disusun Oleh:

Mohammad Farid Hendianto 2200018401

UNIVERSITAS AHMAD DAHLAN
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI INFORMATIKA
TAHUN 2023

Modifikasi kodingan materi supaya menjadi inputan dinamis!

Berikut hasil modifikasi kodingan materi di Visual Studio Code



```

File Edit Selection View Go Run Terminal Help
prak2.cpp postest2.cpp Post.cpp
D:\Document Ndk\Kuliah > Semester 2 > Matematika Diskrit > Praktikum > Pertemuan 2 > postest2.cpp > calcRelation()
1 /*
2  * Nama : Mohammad Farid Hendianto
3  * NIM : 2200018401
4  * Kelas : A
5  * Mata Kuliah : Matematika Diskrit
6  * Dosen : Nur Rochmah Dyah PA, S.T., M.Kom.
7  * Post Test Pertemuan ke-II
8 */
9
10 #include <iostream>
11 #include <algorithm>
12 #include <iostream.h>
13 #include <conio.h>
14 #define maxSets 100
15 #define maxElements 100
16 using namespace std;
17
18 // struct of Sets
19 struct{
20     string Name;
21     int totalElements;
22     bool typeElements; // if true = int, if false = string
23     int intElements[maxElements];
24     string stringElements[maxElements];
25 }Sets[maxSets];
26 int totalSets;
27
28 // function to check the type of Elements in Sets
29 void numberingNameSets(int totalSets){
30     for (int i=0;i<totalSets;i++){
31         for (int j=0;j<2;j++){
32             if (j==0){
33                 Sets[i].Name=char(j+65);
34             }
35         }
36     }
37 }
38
39 // function to input total Elements in Sets
40 void totalElementsInSets(int i){
41     cout<<"-----"<<endl;
42     cout<<"Jumlah elemen pada himpunan "<<Sets[i].Name<< " : ";cin>>Sets[i].totalElements;
43     cout<<"-----"<<endl;
44 }
45
46 // function to input Elements in Sets
47 void inputssets(){
48     cout<<"Masukkan jumlah himpunan : ";cin>>totalSets;
49     numberingNameSets(totalSets);
50     for(int i=0;i<totalSets;i++){
51         cout<<"1 = int<<endl;
52         cout<<"0 = string<<endl;
53         cout<<"Masukkan tipe data pada himpunan : ";cin>>Sets[i].typeElements;
54         if (Sets[i].typeElements==true){
55             totalElementsInSets(i);
56             for(int j=0;j<Sets[i].totalElements;j++){
57                 cout<<"Elemen "<<j+1<<" pada himpunan "<<Sets[i].Name<< " : ";
58                 cin>>Sets[i].intElements[j];
59             }
60         }
61         else{
62             totalElementsInSets(i);
63             for(int j=0;j<Sets[i].totalElements;j++){
64                 cout<<"Elemen "<<j+1<<" pada himpunan "<<Sets[i].Name<< " : ";
65                 if (j==0)
66                     cin.ignore();
67                 getline(cin,Sets[i].stringElements[j]);
68                 cout<<"-----"<<endl;
69             }
70         }
71     }
72 }
73
74 // default value of Sets for Praktikum Matematika Diskrit soal no 2
75 void defaultSets(){
76     totalSets = 2;
77 }

```

The screenshot shows a Visual Studio Code window with the following details:

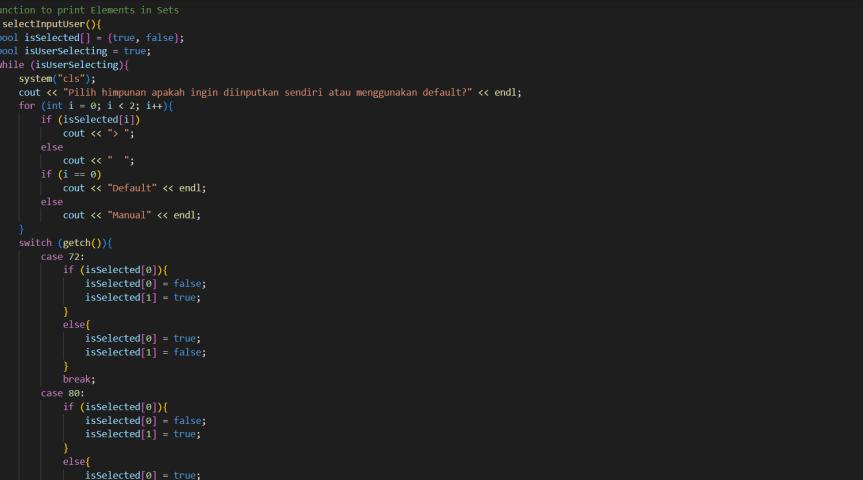
- Title Bar:** postest2.cpp - Visual Studio Code - Insiders [Administrators]
- File Explorer:** Shows files prak2.cpp, postest2.cpp, and Post.cpp.
- Code Editor:** The postest2.cpp file is open, displaying C++ code. The code includes functions for calculating relations between sets and a function to print elements from sets. It also contains a loop for selecting input from the user, either default or manual, and handles user input for sets A and B.
- Status Bar:** In 226, Col 89 | Spaces: 4 | UTF-8 | CR/LF | C++ | Go Live | Win32 | Preferred

```
File Edit Selection View Go Run Terminal Help
postest2.cpp - Visual Studio Code - Insiders [Administrators]

prak2.cpp  postest2.cpp  Post.cpp

D:\Document Ndk\Kuliah\Semester 2 > Matematika Diskrit > Praktikum > Pertemuan 2 > postest2.cpp > calcRelation()

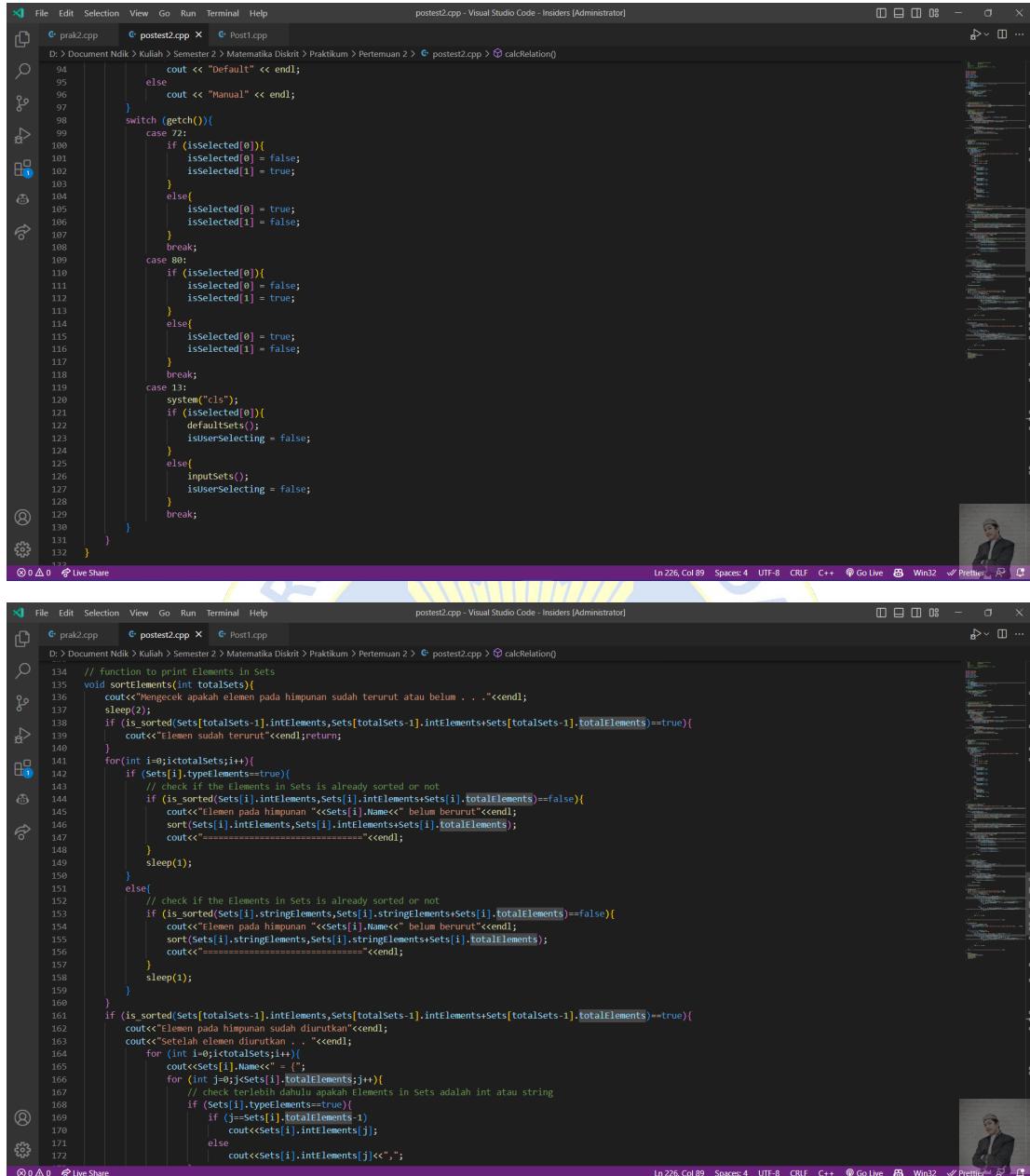
74 // default value of Sets for Praktikum Matematika Diskrit soal no 2
75 void defaultSets(){
76     totalSets = 2;
77     sets[0] = { "c", 3, true, {3,3,4}, {} };
78     sets[1] = { "d", 5, true, {2,4,6,10,12}, {} };
79 }
80
81 // Function to print Elements in Sets
82 void selectInputUser(){
83     bool isSelected[] = {true, false};
84     bool isUserSelecting = true;
85     while (isUserSelecting){
86         system("cls");
87         cout << "Pilih himpunan apakah ingin diinputkan sendiri atau menggunakan default?" << endl;
88         for (int i = 0; i < 2; i++){
89             if (isSelected[i])
90                 cout << "1 ";
91             else
92                 cout << " ";
93             if (i == 0)
94                 cout << "Default" << endl;
95             else
96                 cout << "Manual" << endl;
97         }
98         switch (getch()){
99             case '2':
100                 if (isSelected[0]){
101                     isSelected[0] = false;
102                     isSelected[1] = true;
103                 }
104                 else{
105                     isSelected[0] = true;
106                     isSelected[1] = false;
107                 }
108                 break;
109             case '0':
110                 if (isSelected[0]){
111                     isSelected[0] = false;
112                     isSelected[1] = true;
113                 }
114         }
115     }
116 }
117
118 }
```



The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** File Edit Selection View Go Run Terminal Help Postest2.cpp - Visual Studio Code - Insiders [Administrator]
- Left Sidebar:** Includes icons for file operations like Open, Save, Find, and others.
- Code Editor:** The main area displays C++ code for `Postest2.cpp`. The code implements a function to print elements from sets based on user input (Manual or Default) and handles selection logic for two elements (0 and 1).
- Right Sidebar:** Shows a preview of the code's output or a related document.
- Bottom Status Bar:** Shows the current file path (`D:\.../Postest2.cpp`), line number (Line 226), column (Col 89), and encoding (UTF-8). It also includes icons for Live Share, Go Live, and other developer tools.

```
// function to print Elements in Sets
void selectInputUser(){
    bool isSelected[2] = {true, false};
    bool isUserSelecting = true;
    while (isUserSelecting){
        system("cls");
        cout << "Pilih himpunan apakah ingin diinputkan sendiri atau menggunakan default?" << endl;
        for (int i = 0; i < 2; i++){
            if (isSelected[i])
                cout << "> ";
            else
                cout << " ";
            if (i == 0)
                cout << "Default" << endl;
            else
                cout << "Manual" << endl;
        }
        switch (getch()){
            case 72:
                if (isSelected[0]){
                    isSelected[0] = false;
                    isSelected[1] = true;
                }
                else{
                    isSelected[0] = true;
                    isSelected[1] = false;
                }
                break;
            case 80:
                if (isSelected[0]){
                    isSelected[0] = false;
                    isSelected[1] = true;
                }
                else{
                    isSelected[0] = true;
                    isSelected[1] = false;
                }
                break;
            case 13:
        }
    }
}
```

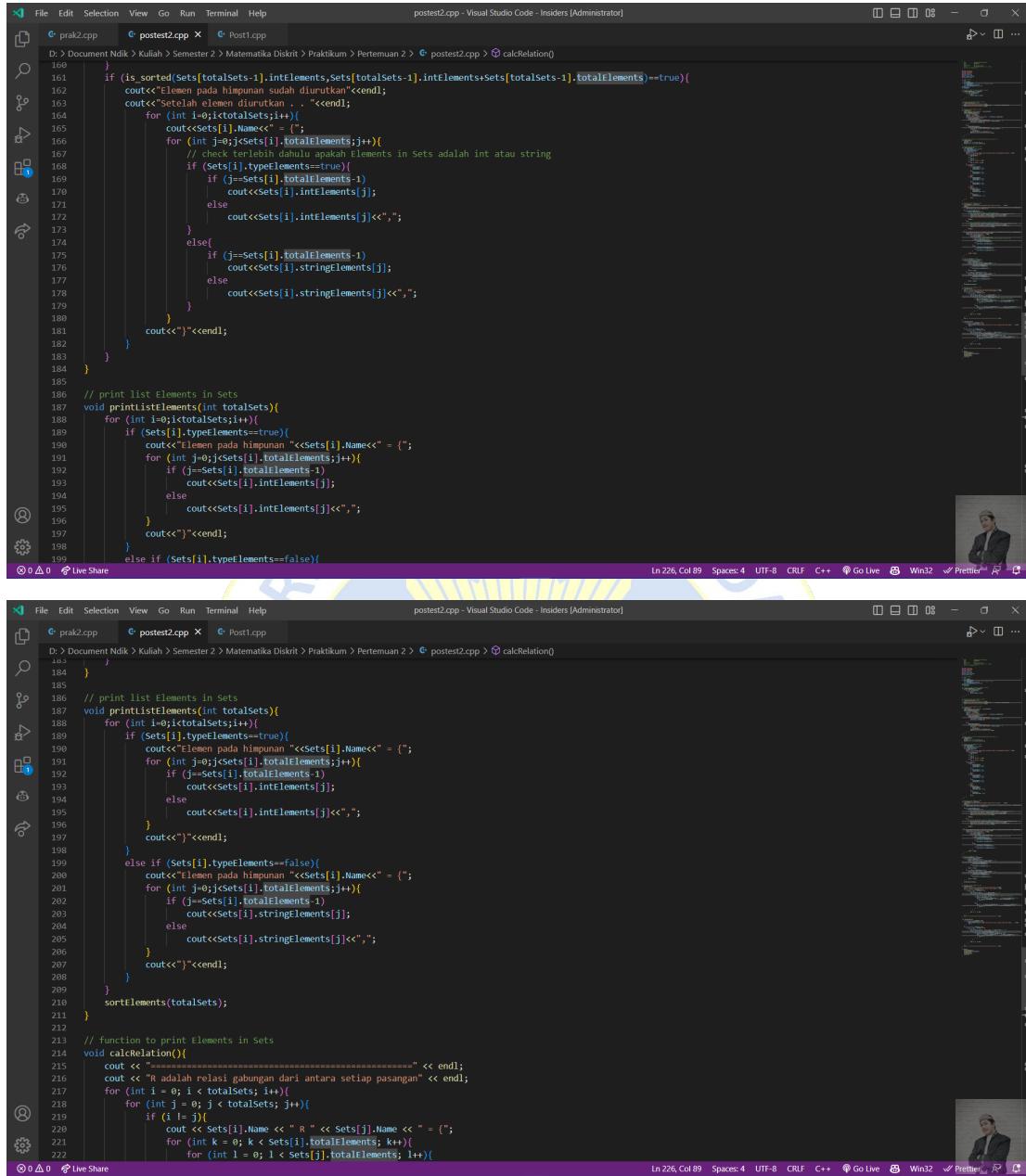


```

File Edit Selection View Go Run Terminal Help
postest2.cpp - Visual Studio Code - Insiders [Administrator]
D: > Document Ndk > Kuliah > Semester 2 > Matematika Diskrit > Praktikum > Pertemuan 2 > postest2.cpp > calcRelation()

94     cout << "Default" << endl;
95     else
96         cout << "Manual" << endl;
97     }
98     switch (getch()){
99         case 72:
100             if (isSelected[0]){
101                 isSelected[0] = false;
102                 isSelected[1] = true;
103             }
104             else{
105                 isSelected[0] = true;
106                 isSelected[1] = false;
107             }
108             break;
109         case 80:
110             if (isSelected[0]){
111                 isSelected[0] = false;
112                 isSelected[1] = true;
113             }
114             else{
115                 isSelected[0] = true;
116                 isSelected[1] = false;
117             }
118             break;
119         case 13:
120             system("cls");
121             if (isSelected[0]){
122                 defaultSets();
123                 isUserSelecting = false;
124             }
125             else{
126                 inputSets();
127                 isUserSelecting = false;
128             }
129             break;
130     }
131 }
132 }

133 // function to print Elements in Sets
134 void sortElements(int totalSets){
135     cout<<"Mengecek apakah elemen pada himpunan sudah terurut atau belum . . ."<<endl;
136     sleep(2);
137     if (is_sorted(Sets[totalSets-1].intElements,Sets[totalSets-1].intElements+Sets[totalSets-1].totalElements)==true){
138         cout<<"Elemen sudah terurut"<<endl;return;
139     }
140     for(int i=0;i<totalSets;i++){
141         if (Sets[i].typeElements==true){
142             // check if the Elements in Sets is already sorted or not
143             if (is_sorted(Sets[i].intElements,Sets[i].intElements+Sets[i].totalElements)==false){
144                 cout<<"Elemen pada himpunan "<<Sets[i].Name<< " belum berurut"<<endl;
145                 sort(Sets[i].intElements,Sets[i].intElements+Sets[i].totalElements);
146                 cout<<=====<<endl;
147             }
148             sleep(1);
149         }
150     }
151     else{
152         // check if the Elements in Sets is already sorted or not
153         if (is_sorted(Sets[i].stringElements,Sets[i].stringElements+Sets[i].totalElements)==false){
154             cout<<"Elemen pada himpunan "<<Sets[i].Name<< " belum berurut"<<endl;
155             sort(Sets[i].stringElements,Sets[i].stringElements+Sets[i].totalElements);
156             cout<<=====<<endl;
157         }
158         sleep(1);
159     }
160 }
161 if (is_sorted(Sets[totalSets-1].intElements,Sets[totalSets-1].intElements+Sets[totalSets-1].totalElements)==true){
162     cout<<"Elemen pada himpunan sudah diurutkan"<<endl;
163     cout<<"Setelah elemen diurutkan . . ."<<endl;
164     for (int i=0;i<totalSets;i++){
165         cout<<Sets[i].Name<< "=" ;
166         for (int j=0;j<Sets[i].totalElements;j++){
167             // check terlebih dahulu apakah elements in Sets adalah int atau string
168             if (Sets[i].typeElements==true){
169                 if (j==Sets[i].totalElements-1)
170                     cout<<Sets[i].intElements[j];
171                 else
172                     cout<<Sets[i].intElements[j]<<",";
```



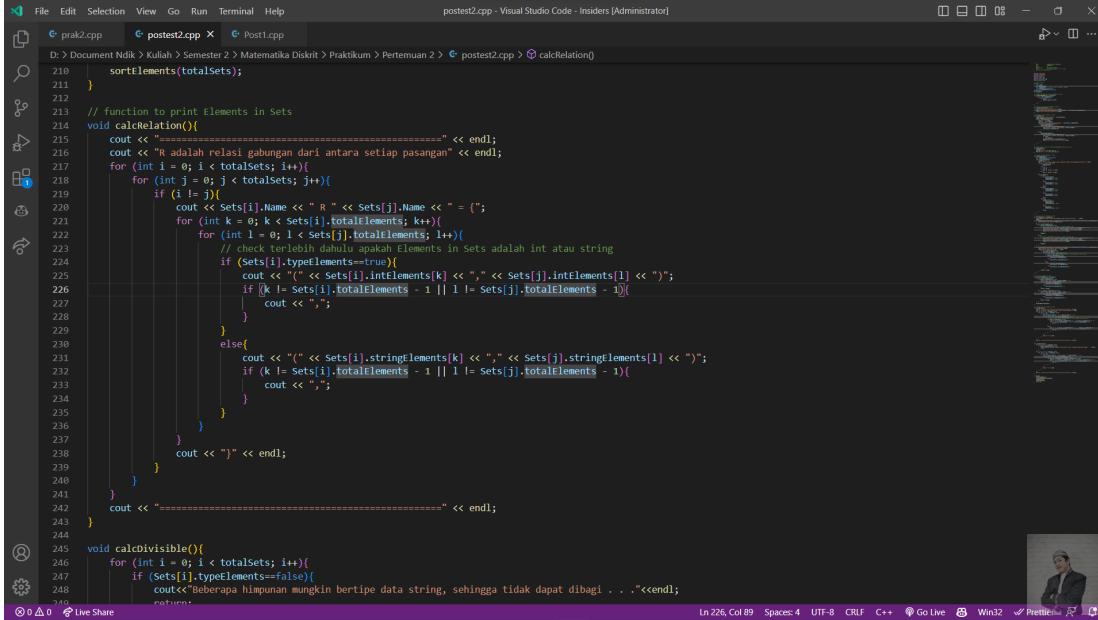
```

160 }
161     if (is_sorted(Sets[totalSets-1].intElements,Sets[totalSets-1].intElements+Sets[totalSets-1].totalElements)==true){
162         cout<<"Elemen pada himpunan sudah urutkan"<<endl;
163         cout<<"Setelah elemen diurutkan . . ."<<endl;
164         for (int i=0;i<totalSets;i++){
165             cout<<Sets[i].Name<<"";
166             for (int j=0;j<Sets[i].totalElements;j++){
167                 if (Sets[i].typeElements==true){
168                     if (j==Sets[i].totalElements-1)
169                         cout<<Sets[i].intElements[j];
170                     else
171                         cout<<Sets[i].intElements[j]<< ",";
172                 }
173                 else{
174                     if (j==Sets[i].totalElements-1)
175                         cout<<Sets[i].stringElements[j];
176                     else
177                         cout<<Sets[i].stringElements[j]<< ",";
178                 }
179             }
180             cout<<")"<<endl;
181         }
182     }
183 }
184 }

185 // print list Elements in Sets
186 void printListElements(int totalSets){
187     for (int i=0;i<totalSets;i++){
188         if (Sets[i].typeElements==true){
189             cout<<"Elemen pada himpunan "<<Sets[i].Name<<" = ";
190             for (int j=0;j<Sets[i].totalElements;j++){
191                 if (j==Sets[i].totalElements-1)
192                     cout<<Sets[i].intElements[j];
193                 else
194                     cout<<Sets[i].intElements[j]<< ",";
195             }
196             cout<<")"<<endl;
197         }
198         else if (Sets[i].typeElements==false){
199             cout<<"Elemen pada himpunan "<<Sets[i].Name<<" = ";
200             for (int j=0;j<Sets[i].totalElements;j++){
201                 if (j==Sets[i].totalElements-1)
202                     cout<<Sets[i].stringElements[j];
203                 else
204                     cout<<Sets[i].stringElements[j]<< ",";
205             }
206             cout<<")"<<endl;
207         }
208     }
209     sortElements(totalSets);
210 }
211 }

212 // function to print Elements in Sets
213 void calcRelation(){
214     cout << "-----" << endl;
215     cout << "R adalah relasi gabungan dari antara setiap pasangan" << endl;
216     for (int i = 0; i < totalSets; i++){
217         for (int j = 0; j < totalSets; j++){
218             if (i != j){
219                 cout << Sets[i].Name << " R " << Sets[j].Name << " = ";
220                 for (int k = 0; k < Sets[i].totalElements; k++){
221                     for (int l = 0; l < Sets[j].totalElements; l++){

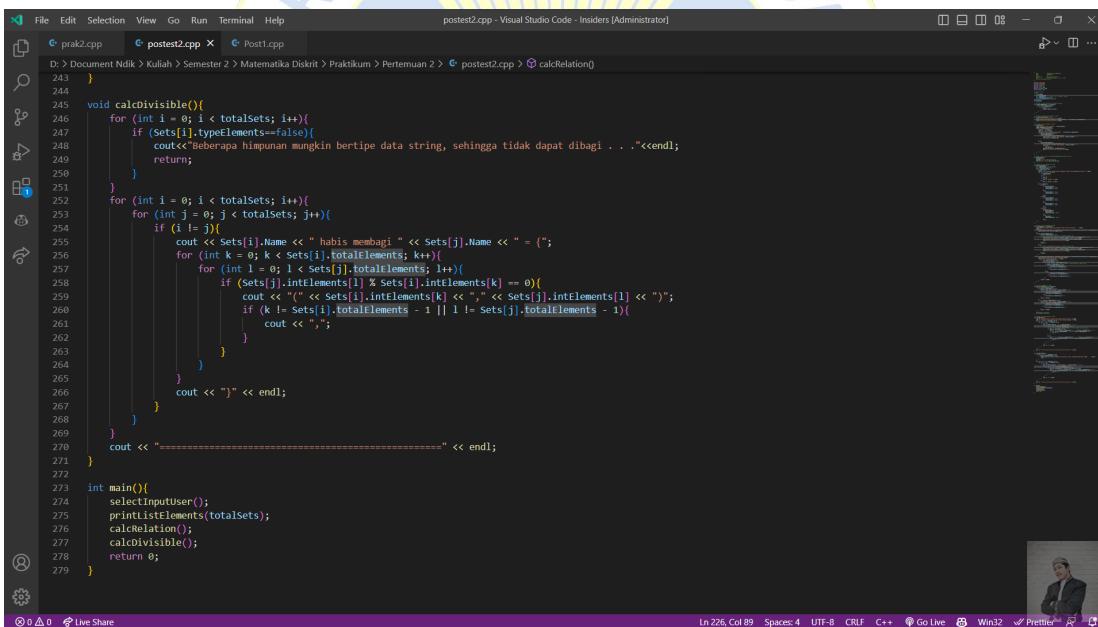
```



```

210     sortElements(totalSets);
211 }
212
213 // function to print Elements in Sets
214 void calcRelation(){
215     cout << "===== R =====" << endl;
216     cout << "R adalah relasi gabungan dari antara setiap pasangan" << endl;
217     for (int i = 0; i < totalSets; i++){
218         for (int j = 0; j < totalSets; j++){
219             if (i != j){
220                 cout << Sets[i].Name << " R " << Sets[j].Name << " = ";
221                 for (int k = 0; k < Sets[i].totalElements; k++){
222                     for (int l = 0; l < Sets[j].totalElements; l++){
223                         // check apakah Elements dalam Sets adalah int atau string
224                         if (Sets[i].typeElements==true){
225                             cout << "(" << Sets[i].intElements[k] << "," << Sets[j].intElements[l] << ")";
226                             if ((k != Sets[i].totalElements - 1 || l != Sets[j].totalElements - 1)){
227                                 cout << ",";
228                             }
229                         }
230                         else{
231                             cout << "(" << Sets[i].stringElements[k] << "," << Sets[j].stringElements[l] << ")";
232                             if ((k != Sets[i].totalElements - 1 || l != Sets[j].totalElements - 1)){
233                                 cout << ",";
234                             }
235                         }
236                     }
237                 }
238                 cout << ")" << endl;
239             }
240         }
241     }
242     cout << "===== R =====" << endl;
243 }
244
245 void calcDivisible(){
246     for (int i = 0; i < totalSets; i++){
247         if (Sets[i].typeElements==false){
248             cout << "Beberapa himpunan mungkin bertipe data string, sehingga tidak dapat dibagi . . ." << endl;
249             return;
250         }
251     }
252     for (int i = 0; i < totalSets; i++){
253         for (int j = 0; j < totalSets; j++){
254             if (i != j){
255                 cout << Sets[i].Name << " habis membagi " << Sets[j].Name << " = ";
256                 for (int k = 0; k < Sets[i].totalElements; k++){
257                     for (int l = 0; l < Sets[j].totalElements; l++){
258                         if (Sets[j].intElements[l] % Sets[i].intElements[k] == 0){
259                             cout << "(" << Sets[i].intElements[k] << "," << Sets[j].intElements[l] << ")";
260                             if ((k != Sets[i].totalElements - 1 || l != Sets[j].totalElements - 1)){
261                                 cout << ",";
262                             }
263                         }
264                     }
265                 }
266                 cout << ")" << endl;
267             }
268         }
269     }
270     cout << "===== R =====" << endl;
271 }
272
273 int main(){
274     selectInputUser();
275     printListSets(totalSets);
276     calcRelation();
277     calcDivisible();
278     return 0;
279 }

```



```

210     sortElements(totalSets);
211 }
212
213 // function to print Elements in Sets
214 void calcRelation(){
215     cout << "===== R =====" << endl;
216     cout << "R adalah relasi gabungan dari antara setiap pasangan" << endl;
217     for (int i = 0; i < totalSets; i++){
218         for (int j = 0; j < totalSets; j++){
219             if (i != j){
220                 cout << Sets[i].Name << " R " << Sets[j].Name << " = ";
221                 for (int k = 0; k < Sets[i].totalElements; k++){
222                     for (int l = 0; l < Sets[j].totalElements; l++){
223                         // check apakah Elements dalam Sets adalah int atau string
224                         if (Sets[i].typeElements==true){
225                             cout << "(" << Sets[i].intElements[k] << "," << Sets[j].intElements[l] << ")";
226                             if ((k != Sets[i].totalElements - 1 || l != Sets[j].totalElements - 1)){
227                                 cout << ",";
228                             }
229                         }
230                         else{
231                             cout << "(" << Sets[i].stringElements[k] << "," << Sets[j].stringElements[l] << ")";
232                             if ((k != Sets[i].totalElements - 1 || l != Sets[j].totalElements - 1)){
233                                 cout << ",";
234                             }
235                         }
236                     }
237                 }
238                 cout << ")" << endl;
239             }
240         }
241     }
242     cout << "===== R =====" << endl;
243 }
244
245 void calcDivisible(){
246     for (int i = 0; i < totalSets; i++){
247         if (Sets[i].typeElements==false){
248             cout << "Beberapa himpunan mungkin bertipe data string, sehingga tidak dapat dibagi . . ." << endl;
249             return;
250         }
251     }
252     for (int i = 0; i < totalSets; i++){
253         for (int j = 0; j < totalSets; j++){
254             if (i != j){
255                 cout << Sets[i].Name << " habis membagi " << Sets[j].Name << " = ";
256                 for (int k = 0; k < Sets[i].totalElements; k++){
257                     for (int l = 0; l < Sets[j].totalElements; l++){
258                         if (Sets[j].intElements[l] % Sets[i].intElements[k] == 0){
259                             cout << "(" << Sets[i].intElements[k] << "," << Sets[j].intElements[l] << ")";
260                             if ((k != Sets[i].totalElements - 1 || l != Sets[j].totalElements - 1)){
261                                 cout << ",";
262                             }
263                         }
264                     }
265                 }
266                 cout << ")" << endl;
267             }
268         }
269     }
270     cout << "===== R =====" << endl;
271 }
272
273 int main(){
274     selectInputUser();
275     printListSets(totalSets);
276     calcRelation();
277     calcDivisible();
278     return 0;
279 }

```

Berikut adalah full kode source kodingan dinamis

```

1 /*
2  * Nama : Mohammad Farid Hendianto
3  * NIM : 2200018401
4  * Kelas : I
5  * Mata Kuliah : Matematika Diskrit
6  * Dosen : Nur Rochmah Dyah PA, S.T., M.Kom,
7  * Post Test Pertemuan ke-II
8 */
9
10 #include <iostream>
11 #include <algorithm>
12 #include <unistd.h>
13 #include <conio.h>
14 #define maxSets 100
15 #define maxElements 1000
16 using namespace std;
17
18 // struct of Sets
19 struct Sets {
20     string Name;
21     int totalElements;
22     bool typeElements; // if true = int, if false = string
23     int intElements[maxElements];
24     string stringElements[maxElements];
25 } Sets[maxSets];
26 int totalSets;
27
28 // Function to check the type of Elements in Sets
29 void numberInSets(int totalSets) {
30     for (int i=0;i<totalSets;i++){
31         for (int j=0;j<26;j++){
32             if (j==0){
33                 Sets[i].Name=char(j+65);
34             }
35         }
36     }
37 }
38
39 // Function to input total elements in Sets
40 void totalElementsInSets(int i){
41     cout<<=====<<endl;
42     cout<<"Jumlah elemen pada himpunan "<<Sets[i].Name<<" : ";cin>>Sets[i].totalElements;
43     cout<<=====<<endl;
44 }
45
46 // Function to input Elements in Sets
47 void InputSets(){
48     cout<<"Masukkan Jumlah himpunan : ";cin>>totalSets;
49     numberInSets(totalSets);
50     for (int i=0;i<totalSets;i++){
51         cout<<"1 = int<<endl;
52         cout<<"0 = string<<endl;
53         cout<<"Masukan tipe data pada himpunan : ";cin>>Sets[i].typeElements;
54         if (Sets[i].typeElements==true){
55             totalElementsInSets(i);
56             for (int j=0;j<Sets[i].totalElements;j++){
57                 cout<<"Elemen "<<j+1<<" pada himpunan "<<Sets[i].Name<<" : ";
58                 cin>>Sets[i].intElements[j];
59             }
60         }
61         else{
62             initialElementsInSets(i);
63             for (int j=0;j<Sets[i].totalElements;j++){
64                 cout<<"Elemen "<<j+1<<" pada himpunan "<<Sets[i].Name<<" : ";
65                 if (j==0)
66                     cin.ignore();
67                 getline(cin,Sets[i].stringElements[j]);
68                 cout<<=====<<endl;
69             }
70         }
71     }
72 }
73
74 // default value of Sets for Praktikum Matematika Diskrit soal no 2
75 void defaultSets(){
76     totalSets = 2;
77     Sets[0].Name = "A", Sets[0].typeElements = true;
78     Sets[0].intElements[0] = 2, Sets[0].intElements[1] = 3, Sets[0].intElements[2] = 4, Sets[0].intElements[3] = 1;
79     Sets[1].Name = "B", Sets[1].typeElements = false;
80     Sets[1].stringElements[0] = "d", Sets[1].stringElements[1] = "e", Sets[1].stringElements[2] = "f", Sets[1].stringElements[3] = "g", Sets[1].stringElements[4] = "h";
81
82     // function to print Elements in Sets
83     void selectInputUser(){
84         bool isUserSelecting = true;
85         bool isUserSelectingD = true;
86         while (isUserSelectingD){
87             system("cls");
88             cout << "Pilih himpunan apakah ingin diinputkan sendiri atau menggunakan default" << endl;
89             for (int i = 0; i < 2; i++){
90                 if (isUserSelected[i])
91                     cout << " ";
92                 else
93                     cout << " ";
94             }
95             cout << "Default" << endl;
96             cout << "Manual" << endl;
97         }
98         switch (getch()){
99             case '2':
100                 if (!isUserSelected[0]){
101                     isUserSelected[0] = !isUserSelected[0];
102                     isUserSelected[1] = !isUserSelected[1];
103                 }
104                 else{
105                     isUserSelected[0] = true;
106                     isUserSelected[1] = false;
107                 }
108                 break;
109             case '0':
110                 if (!isUserSelected[0]){
111                     isUserSelected[0] = !isUserSelected[0];
112                     isUserSelected[1] = !isUserSelected[1];
113                 }
114                 else{
115                     isUserSelected[0] = true;
116                     isUserSelected[1] = false;
117                 }
118                 break;
119             case '1':
120                 system("cls");
121                 if (isUserSelected[0]){
122                     defaultSets();
123                     isUserSelectingD = !isUserSelectingD;
124                 }
125                 else{
126                     inputSets();
127                     isUserSelectingD = !isUserSelectingD;
128                 }
129             break;
130         }
131     }
132 }

```

```

134 // function to print Elements in Sets
135 void sortElements(int totalSets){
136     cout<<"Mengecek apakah elemen pada himpunan sudah terurut atau belum . ."<<endl;
137     sleep(2);
138     if (is_sorted(Sets[totalSets-1].intElements,Sets[totalSets-1].intElements+Sets[totalSets-1].totalElements)==true){
139         cout<<"Elemen Sudah Terurut "<<endl;return;
140     }
141     for (int i=0;i<totalSets;i++){
142         if (Sets[i].typeElements==true){
143             // check if the elements in Sets is already sorted or not
144             if ((is_sorted(Sets[i].stringElements,Sets[i].stringElements+Sets[i].totalElements)==false)|
145                 (cout<<"Elemen pada himpunan "<<Sets[i].Name<<" belum berurut"<<endl));
146             sort(Sets[i].intElements,Sets[i].intElements+Sets[i].totalElements);
147             cout<<"----- "<<endl;
148         }
149         sleep(1);
150     }
151     else{
152         // check if the Elements in Sets is already sorted or not
153         if (is_sorted(Sets[i].stringElements,Sets[i].stringElements+Sets[i].totalElements)==false){
154             cout<<"Elemen pada himpunan "<<Sets[i].Name<<" belum berurut"<<endl;
155             sort(Sets[i].stringElements,Sets[i].stringElements+Sets[i].totalElements);
156             cout<<"----- "<<endl;
157         }
158         sleep(1);
159     }
160 }
161 if (is_sorted(Sets[totalSets-1].intElements,Sets[totalSets-1].intElements+Sets[totalSets-1].totalElements)==true){
162     cout<<"Elemen pada himpunan sudah diurutkan . ."<<endl;
163     cout<<"Setelah elemen diurutkan . ."<<endl;
164     for (int i=0;i<totalSets;i++){
165         cout<<"----- "<<endl;
166         for (int j=0;j<Sets[i].totalElements;j++){
167             // check terlebih dahulu apakah Elements in Sets adalah int atau string
168             if (Sets[i].typeElements==true){
169                 if (j==Sets[i].totalElements-1)
170                     cout<<Sets[i].intElements[j];
171                 else
172                     cout<<Sets[i].intElements[j]<<",";
173             }
174             else{
175                 if (j==Sets[i].totalElements-1)
176                     cout<<Sets[i].stringElements[j];
177                 else
178                     cout<<Sets[i].stringElements[j]<<",";
179             }
180         }
181     }
182     cout<<"----- "<<endl;
183 }
184 }
185
186 // print list Elements in sets
187 void printListElements(int totalSets){
188     for (int i=0;i<totalSets;i++){
189         if (Sets[i].typeElements==true){
190             cout<<"Elemen pada himpunan "<<Sets[i].Name<<" = (";
191             for (int j=0;j<Sets[i].totalElements;j++){
192                 if (j==Sets[i].totalElements-1)
193                     cout<<Sets[i].intElements[j];
194                 else
195                     cout<<Sets[i].intElements[j]<< ",";
196             }
197             cout<<")"<<endl;
198         }
199         else if (Sets[i].typeElements==false){
200             cout<<"Elemen pada himpunan "<<Sets[i].Name<<" = (";
201             for (int j=0;j<Sets[i].totalElements;j++){
202                 if (j==Sets[i].totalElements-1)
203                     cout<<Sets[i].stringElements[j];
204                 else
205                     cout<<Sets[i].stringElements[j]<< ",";
206             }
207             cout<<")"<<endl;
208         }
209     }
210     sortElements(totalSets);
211 }
212
213 // function to print elements in sets
214 void calcRelation(){
215     cout << "=====";
216     cout << "# adalah relasi gabungan dari antara setiap pasangan" << endl;
217     for (int i = 0; i < totalSets; i++)
218         for (int j = 0; j < totalSets; j++){
219             if (i != j){
220                 cout << Sets[i].Name << " # " << Sets[j].Name << " = (";
221                 for (int k = 0; k < Sets[i].totalElements; k++){
222                     for (int l = 0; l < Sets[j].totalElements; l+=2){
223                         // check terlebih dahulu apakah Elements in Sets adalah int atau string
224                         if (Sets[i].typeElements==true){
225                             cout << "(" << Sets[i].intElements[k] << "," << Sets[j].intElements[l] << ")";
226                             if (k == Sets[i].totalElements - 1 || l == Sets[j].totalElements - 1)
227                                 cout << ")";
228                         }
229                         else{
230                             cout << "(" << Sets[i].stringElements[k] << "," << Sets[j].stringElements[l] << ")";
231                             if (k == Sets[i].totalElements - 1 || l == Sets[j].totalElements - 1)
232                                 cout << ")";
233                         }
234                     }
235                 }
236             }
237             cout << ")" << endl;
238         }
239     }
240 }
241 }
242 cout << "===== " << endl;
243
244 void calcDivisible(){
245     for (int i = 0; i < totalSets; i++){
246         if (Sets[i].typeElements==false){
247             cout<<"Berdasarkan himpunan mungkin bertipe data string, sehingga tidak dapat dibagi . ."<<endl;
248             return;
249         }
250     }
251 }
252 for (int i = 0; i < totalSets; i++){
253     for (int j = 0; j < totalSets; j++){
254         if (i != j){
255             cout << Sets[i].Name << " habis membagi " << Sets[j].Name << " = [";
256             for (int k = 0; k < Sets[i].totalElements; k++){
257                 for (int l = 0; l < Sets[j].totalElements; l++){
258                     if (Sets[i].intElements[k] % Sets[j].intElements[l] == 0){
259                         cout << Sets[i].intElements[k] << Sets[j].intElements[l] << ",";
260                         if (k == Sets[i].totalElements - 1 || l == Sets[j].totalElements - 1)
261                             cout << ",";
262                     }
263                 }
264             }
265             cout << "]" << endl;
266         }
267     }
268 }
269 }
270 cout << "----- " << endl;
271 }
272
273 int main()
274 {
275     cin.tie(NULL);
276     printListElements(totalSets);
277     calcRelation();
278     calcDivisible();
279     return 0;
280 }

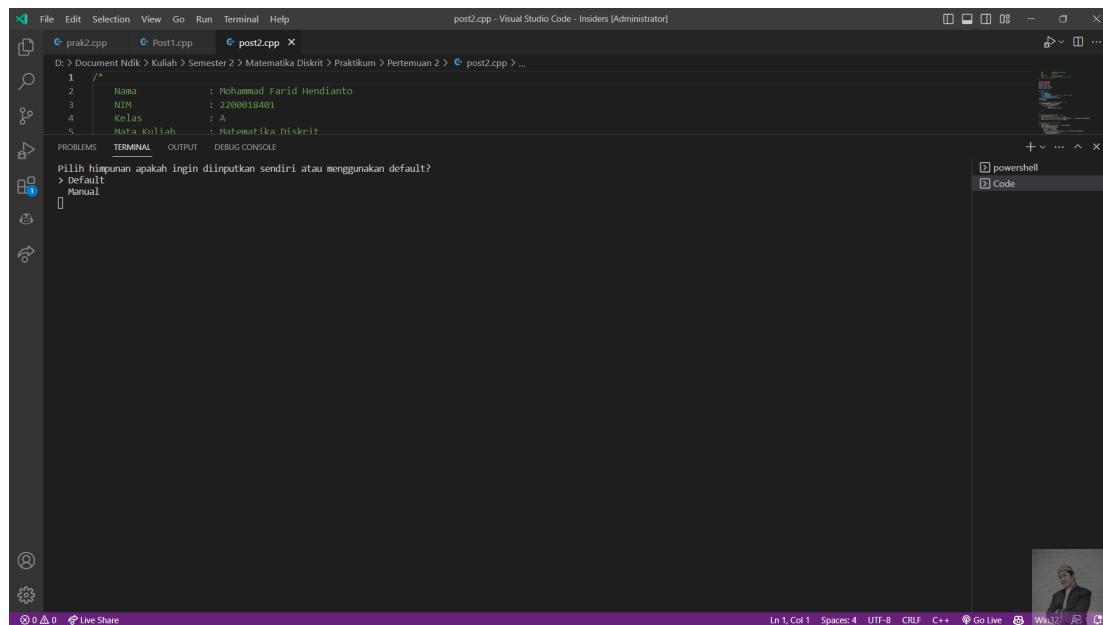
```



Untuk full source code pada program dinamis tentang relasi, dapat mengakses link sebagai berikut

<https://github.com/IRedDragonICY/Matematika-Diskrit/tree/main/Praktikum>

Berikut adalah tampilan program saat dijalankan



The screenshot shows a Visual Studio Code interface with a dark theme. In the center, a terminal window displays the following text:

```
1 //  
2 Nama : Mohammad Farid Hendianto  
3 NIM : 2200018401  
4 Kelas : A  
5 Mata Kuliah : Matematika Diskrit
```

Below the terminal, a message asks: "Pilih himpunan apakah ingin diinputkan sendiri atau menggunakan default?". The user has selected "Manual".

At the bottom of the terminal window, there are status indicators: Line 1, Col 1, Spaces: 4, UTF-8, CR/LF, C++, Go Live, and Webview.

Pertama user akan diminta, apakah ingin menggunakan himpunan yang sudah ada di program (default) atau tidak. Untuk settings default sendiri, untuk menjawab soal pada nomor kedua, yaitu Tampilkan hasil himpunan D habis membagi C. Untuk memilih opsi, bisa diarahkan dengan arah panah atas dan bawah pada keyboard.

```

1 /*
2 2 Nama : Mohammad Farid Hendianto
3 3 NIM : 2200018401
4 4 Kelas : A
5 5 Mata Kuliah : Matematika Diskrit
Element pada himpunan c = {2,3,4}
Element pada himpunan d = {2,4,8,10,12}
Mengecek apakah elemen pada himpunan sudah terurut atau belum . .
Elemen sudah terurut
R adalah relasi gabungan dari antara setiap pasangan
c R d = {(2,2),(2,4),(2,8),(2,10),(2,12),(3,2),(3,4),(3,8),(3,10),(3,12),(4,2),(4,4),(4,8),(4,10),(4,12)}
d R c = {(2,2),(2,3),(2,4),(4,2),(4,3),(4,4),(8,2),(8,3),(8,4),(10,2),(10,3),(10,4),(12,2),(12,3),(12,4)}
c habis bagi d = {(2,2),(2,4),(2,8),(2,16),(2,12),(3,12),(4,4),(4,8),(4,12)}
d habis bagi c = {(2,2),(2,4),(4,4)}
=====
PS D:\Document Ndik\Kuliah\Semester 2\Matematika Diskrit\Praktikum\Pertemuan 2>

```

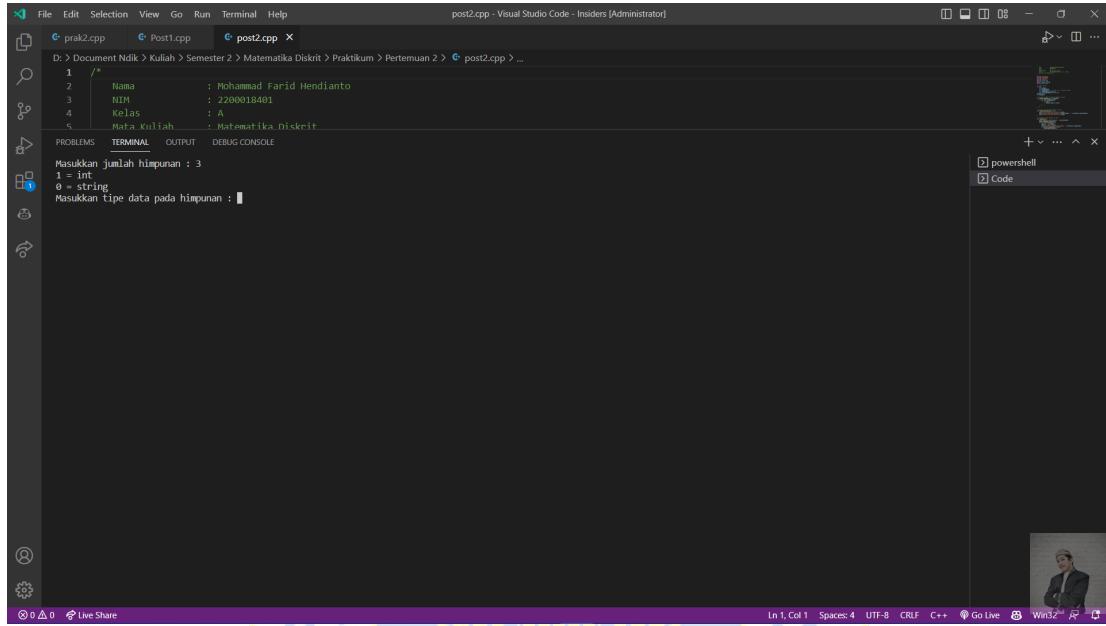
Gambar di atas merupakan tampilan jika memilih opsi default.

```

1 /*
2 2 Nama : Mohammad Farid Hendianto
3 3 NIM : 2200018401
4 4 Kelas : A
5 5 Mata Kuliah : Matematika Diskrit
Masukkan jumlah himpunan :

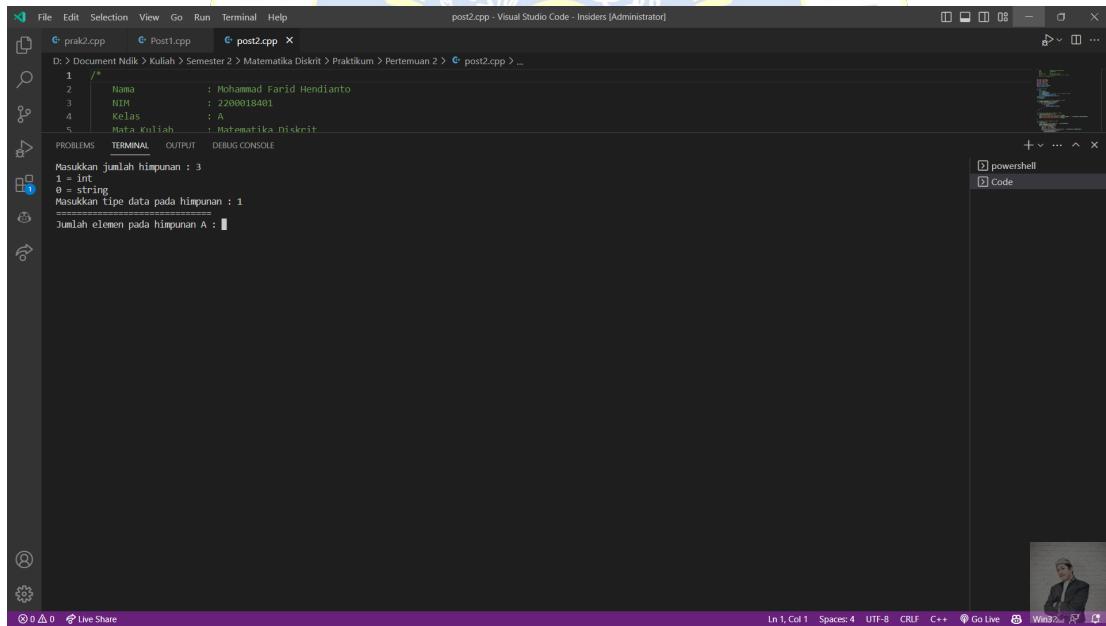
```

Jika memilih opsi manual, akan muncul pertama berapa banyak jumlah himpunan yang ingin di input.



```
File Edit Selection View Go Run Terminal Help
prak2.cpp Post1.cpp post2.cpp
D:\Document Ndik>Kuliah>Semester 2>Matematika Diskrit>Praktikum>Pertemuan 2>post2.cpp>...
1 /*
2  Nama : Mohammad Farid Hendianto
3  NIM : 2200018401
4  Kelas : A
5  Mata Kuliah : Matematika niskerit
PROBLEMS TERMINAL OUTPUT DEBUG CONSOLE
Masukkan jumlah himpunan : 3
1 = int
0 = string
Masukkan tipe data pada himpunan : 
```

Kemudian akan muncul, tipe data himpunan yang ingin dimasukkan apa, misalnya integer untuk angka, sedangkan string untuk huruf.



```
File Edit Selection View Go Run Terminal Help
prak2.cpp Post1.cpp post2.cpp
D:\Document Ndik>Kuliah>Semester 2>Matematika Diskrit>Praktikum>Pertemuan 2>post2.cpp>...
1 /*
2  Nama : Mohammad Farid Hendianto
3  NIM : 2200018401
4  Kelas : A
5  Mata Kuliah : Matematika niskerit
PROBLEMS TERMINAL OUTPUT DEBUG CONSOLE
Masukkan jumlah himpunan : 3
1 = int
0 = string
Masukkan tipe data pada himpunan : 1
=====
Jumlah elemen pada himpunan A : 
```

Kemudian akan diminta jumlah elemen pada himpunan yang dimulai dari A

```

File Edit Selection View Go Run Terminal Help
prak2.cpp Post1.cpp post2.cpp
D:\> Document Ndik > Kuliah > Semester 2 > Matematika Diskrit > Praktikum > Pertemuan 2 > post2.cpp > ...
1 /*
2  Nama : Mohammad Farid Hendianto
3  NIM : 2200018401
4  Kelas : A
5  Mata Kuliah : Matematika Diskrit
PROBLEMS TERMINAL OUTPUT DEBUG CONSOLE
Masukkan jumlah himpunan : 3
1 = int
0 = string
Masukkan tipe data pada himpunan : 1
Jumlah elemen pada himpunan A : 3
Elemen 1 pada himpunan A :

```

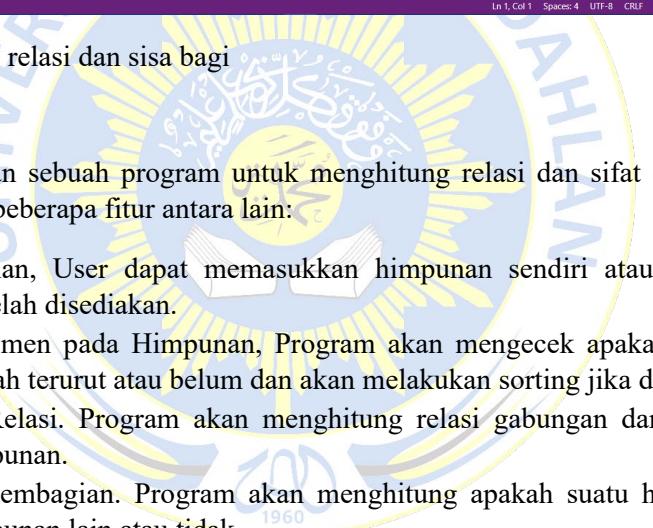
Kemudian akan mengisikan sesuai jumlah banyak elemen yang ingin diinput

```

File Edit Selection View Go Run Terminal Help
prak2.cpp Post1.cpp post2.cpp
D:\> Document Ndik > Kuliah > Semester 2 > Matematika Diskrit > Praktikum > Pertemuan 2 > post2.cpp > ...
1 /*
2  Nama : Mohammad Farid Hendianto
3  NIM : 2200018401
4  Kelas : A
5  Mata Kuliah : Matematika Diskrit
PROBLEMS TERMINAL OUTPUT DEBUG CONSOLE
Masukkan jumlah himpunan : 3
1 = int
0 = string
Masukkan tipe data pada himpunan : 1
Jumlah elemen pada himpunan A : 3
-----
Elemen 1 pada himpunan A : 3
Elemen 2 pada himpunan A : 5
Elemen 3 pada himpunan A : 6
1 = int
0 = string
Masukkan tipe data pada himpunan : 1
-----
Jumlah elemen pada himpunan B : 4
-----
Elemen 1 pada himpunan B : 9
Elemen 2 pada himpunan B : 5
Elemen 3 pada himpunan B : 3
Elemen 4 pada himpunan B : 1
1 = int
0 = string
Masukkan tipe data pada himpunan : 1
-----
Jumlah elemen pada himpunan C : 5
-----
Elemen 1 pada himpunan C : 6
Elemen 2 pada himpunan C : 0
Elemen 3 pada himpunan C : 4
Elemen 4 pada himpunan C : 3
Elemen 5 pada himpunan C : 5
Elemen pada himpunan A = {3,5,6}
Elemen pada himpunan B = {9,5,3,1}
Elemen pada himpunan C = {6,0,4,3,5}
Mengecek apakah elemen pada himpunan sudah terurut atau belum . .

```

Kemudian berulang-ulang hingga input habis, setelah itu akan mengecek apakah himpunannya sudah berurut atau belum.



```

File Edit Selection View Go Run Terminal Help
prak2.cpp Post1.cpp post2.cpp
D:\Document Ndik > Kuliah > Semester 2 > Matematika Diskrit > Praktikum > Pertemuan 2 > post2.cpp > ...
1 /*
2  * Nama : Mohammad Farid Hendianto
3  * NIM : 2200018401
4  * Kelas : A
5  * Mata Kuliah : Matematika Diskrit
6 */
7 int
8 string
9 Masukkan tipe data pada himpunan : 1
10 =====
11 Jumlah elemen pada himpunan C : 5
12 =====
13 Elemen 1 pada himpunan C : 6
14 Elemen 2 pada himpunan C : 0
15 Elemen 3 pada himpunan C : 4
16 Elemen 4 pada himpunan C : 3
17 Elemen 5 pada himpunan C : 5
18 Elemen pada himpunan A = {3,5,6}
19 Elemen pada himpunan B = {9,5,3,1}
20 Elemen pada himpunan C = {6,0,4,3,5}
21 Mengcek apakah elemen pada himpunan sudah terurut atau belum . . .
22 Elemen pada himpunan B belum berurut
23 =====
24 Elemen pada himpunan C belum berurut
25 =====
26 Elemen pada himpunan sudah diurutkan
27 Setelah elemen diurutkan . .
28 A = {0,3,4,5,6}
29 B = {1,3,5,9}
30 C = {0,3,4,5,6}
31 =====
32 R adalah relasi gabungan dari antara setiap pasangan
33 A x B = {(0,1),(0,2),(0,3),(0,4),(0,5),(0,6),(0,7),(0,8),(0,9),(1,2),(1,3),(1,4),(1,5),(1,6),(1,7),(1,8),(1,9),(2,3),(2,4),(2,5),(2,6),(2,7),(2,8),(2,9),(3,4),(3,5),(3,6),(3,7),(3,8),(3,9),(4,5),(4,6),(4,7),(4,8),(4,9),(5,6),(5,7),(5,8),(5,9),(6,7),(6,8),(6,9),(7,8),(7,9),(8,9)}
34 A x C = {(3,0),(3,1),(3,2),(3,4),(3,5),(3,6),(3,7),(3,8),(3,9),(5,0),(5,1),(5,2),(5,3),(5,4),(5,5),(5,6),(5,7),(5,8),(5,9),(6,0),(6,1),(6,2),(6,3),(6,4),(6,5),(6,6),(6,7),(6,8),(6,9),(7,0),(7,1),(7,2),(7,3),(7,4),(7,5),(7,6),(7,7),(7,8),(7,9),(8,0),(8,1),(8,2),(8,3),(8,4),(8,5),(8,6),(8,7),(8,8),(8,9),(9,0),(9,1),(9,2),(9,3),(9,4),(9,5),(9,6),(9,7),(9,8),(9,9)}
35 B x A = {(1,0),(1,1),(1,2),(1,3),(1,4),(1,5),(1,6),(1,7),(1,8),(1,9),(2,0),(2,1),(2,2),(2,3),(2,4),(2,5),(2,6),(2,7),(2,8),(2,9),(3,0),(3,1),(3,2),(3,3),(3,4),(3,5),(3,6),(3,7),(3,8),(3,9),(4,0),(4,1),(4,2),(4,3),(4,4),(4,5),(4,6),(4,7),(4,8),(4,9),(5,0),(5,1),(5,2),(5,3),(5,4),(5,5),(5,6),(5,7),(5,8),(5,9),(6,0),(6,1),(6,2),(6,3),(6,4),(6,5),(6,6),(6,7),(6,8),(6,9),(7,0),(7,1),(7,2),(7,3),(7,4),(7,5),(7,6),(7,7),(7,8),(7,9),(8,0),(8,1),(8,2),(8,3),(8,4),(8,5),(8,6),(8,7),(8,8),(8,9),(9,0),(9,1),(9,2),(9,3),(9,4),(9,5),(9,6),(9,7),(9,8),(9,9)}
36 C x R A = {(0,3),(0,5),(0,6),(0,7),(0,8),(0,9),(1,3),(1,5),(1,6),(1,7),(1,8),(1,9),(2,3),(2,5),(2,6),(2,7),(2,8),(2,9),(3,4),(3,6),(3,7),(3,8),(3,9),(4,5),(4,7),(4,8),(4,9),(5,6),(5,7),(5,8),(5,9),(6,7),(6,8),(6,9),(7,8),(7,9),(8,9)}
37 C x R B = {(0,1),(0,3),(0,5),(0,6),(0,7),(0,9),(1,3),(1,5),(1,6),(1,7),(1,8),(1,9),(2,3),(2,5),(2,6),(2,7),(2,8),(2,9),(3,4),(3,6),(3,7),(3,8),(3,9),(4,1),(4,3),(4,5),(4,7),(4,8),(4,9),(5,3),(5,5),(5,7),(5,9),(6,1),(6,3),(6,5),(6,7),(6,9)}
38 A habis membagi B = {(3,2),(3,9),(5,5),}
39 A habis membagi C = {(3,0),(3,2),(3,6),(5,0),(5,3),(6,0),(6,6)}
40 B habis membagi A = {(1,3),(1,5),(1,6),(3,3),(3,6),(5,5),}
41 B habis membagi C = {(1,0),(1,5),(1,4),(1,5),(1,6),(3,0),(3,3),(3,6),(5,0),(5,5),(9,6),}
42 C habis membagi A = {(0,1),(0,3),(0,5),(0,6),(0,7),(0,9),(3,1),(3,3),(3,5),(4,1),(4,3),(4,5),(5,1),(5,3),(5,5),(5,7),(6,1),(6,3),(6,5),(6,9)}
PS D:\Document Ndik\Kuliah\Semester 2\Matematika Diskrit\Praktikum\Pertemuan 2>

```

Terakhir akan dihitung relasi dan sisa bagi

Program ini merupakan sebuah program untuk menghitung relasi dan sifat dari himpunan. Program ini memiliki beberapa fitur antara lain:

- Input Himpunan, User dapat memasukkan himpunan sendiri atau menggunakan default yang telah disediakan.
- Mengecek Elemen pada Himpunan, Program akan mengecek apakah elemen pada himpunan sudah terurut atau belum dan akan melakukan sorting jika diperlukan.
- Menghitung Relasi. Program akan menghitung relasi gabungan dari antara setiap pasangan himpunan.
- Menghitung Pembagian. Program akan menghitung apakah suatu himpunan habis membagi himpunan lain atau tidak.



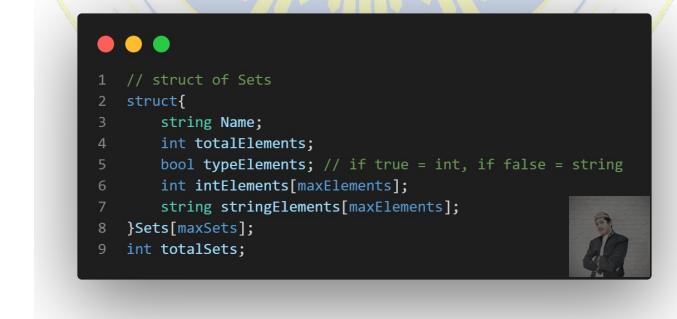
```
1 #include <iostream>
2 #include <algorithm>
3 #include <unistd.h>
4 #include <conio.h>
5 #define maxSets 100
6 #define maxElements 100
7 using namespace std;
```

Kode di atas adalah sebuah header file yang digunakan untuk memuat library dan pendefinisian konstanta-konstanta yang akan digunakan pada program utama.

Pada baris-baris awal, ada beberapa library yang dimuat seperti iostream, algorithm, serta unistd.h dan conio.h. Library iostream digunakan untuk mengakses input dan output standar pada program, sedangkan algorithm digunakan untuk melakukan operasi-operasi pada array seperti sorting atau searching. Sedangkan library unistd.h dan conio.h digunakan untuk membuat program berjalan lebih interaktif dan user-friendly dengan menambahkan beberapa fitur seperti delay dan getch.

Selanjutnya, terdapat dua konstanta yang didefinisikan yaitu maxSets dan maxElements. Konstanta maxSets digunakan untuk menentukan jumlah maksimum himpunan yang dapat dimasukkan dalam program, sedangkan maxElements digunakan untuk menentukan jumlah maksimum elemen pada setiap himpunan.

Terakhir, menggunakan namespace std agar kita dapat langsung menggunakan fungsi-fungsi dari library standard C++ tanpa harus menuliskan std:: di depannya.



```
1 // struct of Sets
2 struct{
3     string Name;
4     int totalElements;
5     bool typeElements; // if true = int, if false = string
6     int intElements[maxElements];
7     string stringElements[maxElements];
8 }Sets[maxSets];
9 int totalSets;
```

Potongan kode di atas merupakan definisi sebuah struct yang merepresentasikan himpunan dan variabel terkait. Struct ini memiliki beberapa field atau variabel yaitu:

- Name: nama dari himpunan, bertipe string
- totalElements: jumlah elemen yang ada pada himpunan, bertipe integer

- typeElements: tipe elemen yang ada pada himpunan, bertipe boolean. Jika nilainya true, maka elemen pada himpunan adalah integer, jika false maka elemen pada himpunan adalah string.
- intElements: array of integers, digunakan untuk menyimpan elemen pada himpunan jika tipe elemennya adalah integer.
- stringElements: array of strings, digunakan untuk menyimpan elemen pada himpunan jika tipe elemennya adalah string.
- Variabel Sets[maxSets] digunakan untuk menyimpan semua himpunan yang dimasukkan oleh user, dengan batasan maksimum jumlah himpunan yang dapat dimasukkan dalam program adalah sebanyak maxSets.

Terakhir, variabel totalSets digunakan untuk menyimpan jumlah total himpunan yang telah dimasukkan ke dalam program.

Dengan menggunakan definisi struct tersebut dan beberapa variabel terkait, program dapat memproses himpunan-himpunan yang dimasukkan oleh user dan melakukan operasi-operasi matematika diskrit yang berkaitan dengan himpunan.

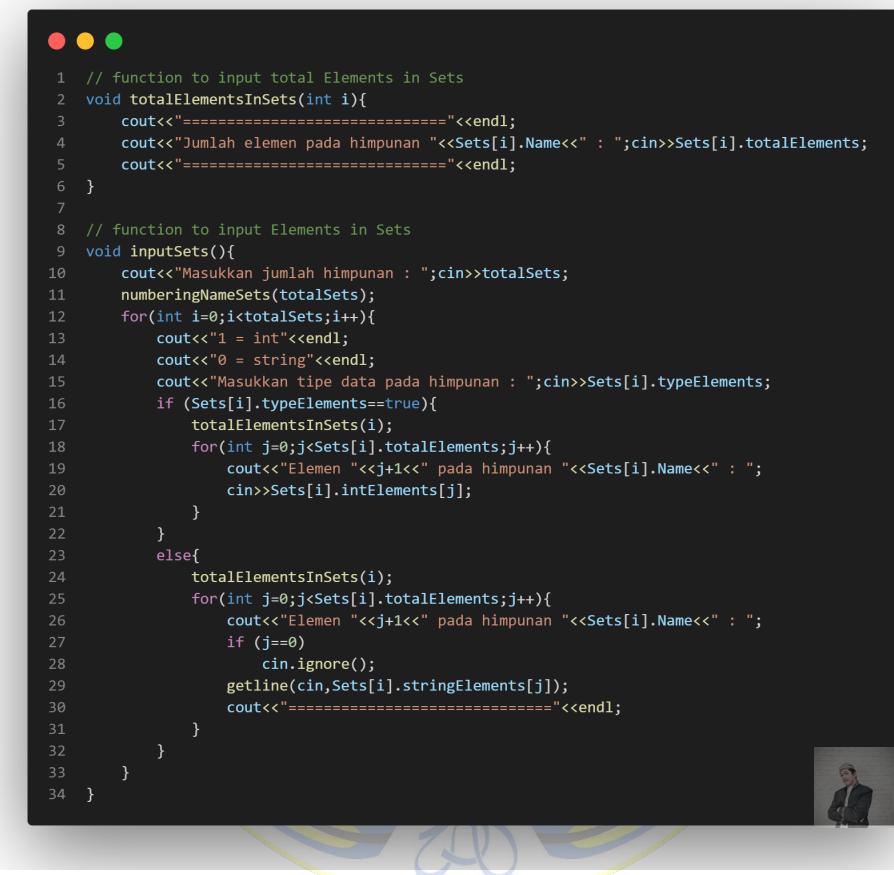


Potongan kode di atas adalah sebuah fungsi yang digunakan untuk menentukan nama dari himpunan. Fungsi tersebut diberi nama numberingNameSets dengan masukan berupa jumlah total himpunan yang telah dimasukkan ke dalam program.

Pertama-tama, dilakukan perulangan for dengan index i yang berjalan dari 0 sampai totalSets-1 untuk melihat setiap himpunan yang telah dimasukkan pada program. Kemudian, dilakukan perulangan for lagi dengan index j yang berjalan dari 0 sampai 25 untuk menginisialisasi nama dari himpunan.

Dalam perulangan kedua, dilakukan pengecekan apakah nilai index i sama dengan nilai index j. Jika benar, maka dilakukan inisialisasi nama himpunan dengan menggunakan karakter ASCII yang sesuai dengan nilai index j, yaitu antara A hingga Z. Untuk melakukan konversi dari nilai index j ke karakter ASCII, dilakukan penjumlahan dengan 65, karena huruf A memiliki nilai ASCII 65 pada tabel ASCII.

Dengan demikian, setelah fungsi numberingNameSets dipanggil dengan argumen totalSets tertentu, maka setiap himpunan yang ada pada program akan diberikan nama A, B, C, dst sesuai urutan penambahan himpunan pada program. Nama ini nantinya akan digunakan untuk mempermudah pengguna program dalam mengidentifikasi himpunan yang mereka inginkan.



```

1 // function to input total Elements in Sets
2 void totalElementsInSets(int i){
3     cout<<"======"<<endl;
4     cout<<"Jumlah elemen pada himpunan "<<Sets[i].Name<<" : ";cin>>Sets[i].totalElements;
5     cout<<"======"<<endl;
6 }
7
8 // function to input Elements in Sets
9 void inputSets(){
10    cout<<"Masukkan jumlah himpunan : ";cin>>totalSets;
11    numberingNameSets(totalSets);
12    for(int i=0;i<totalSets;i++){
13        cout<<"1 = int"<<endl;
14        cout<<"0 = string"<<endl;
15        cout<<"Masukkan tipe data pada himpunan : ";cin>>Sets[i].typeElements;
16        if (Sets[i].typeElements==true){
17            totalElementsInSets(i);
18            for(int j=0;j<Sets[i].totalElements;j++){
19                cout<<"Elemen "<<j+1<<" pada himpunan "<<Sets[i].Name<<" : ";
20                cin>>Sets[i].intElements[j];
21            }
22        }
23        else{
24            totalElementsInSets(i);
25            for(int j=0;j<Sets[i].totalElements;j++){
26                cout<<"Elemen "<<j+1<<" pada himpunan "<<Sets[i].Name<<" : ";
27                if (j==0)
28                    cin.ignore();
29                getline(cin,Sets[i].stringElements[j]);
30                cout<<"======"<<endl;
31            }
32        }
33    }
34 }

```

Potongan kode di atas adalah sebuah fungsi untuk menerima input dari pengguna berupa jumlah elemen pada setiap himpunan. Fungsi ini terdiri dari dua sub-fungsi: totalElementsInSets dan inputSets.

Fungsi totalElementsInSets akan menerima argumen i sebagai index himpunan yang sedang diproses oleh fungsi utama inputSets. Fungsi tersebut akan menampilkan pesan ke layar untuk meminta user memasukkan jumlah elemen pada himpunan dengan menggunakan cout. Kemudian user diminta untuk memasukkan nilai tersebut dengan menggunakan cin, dan nilai tersebut disimpan pada field totalElements di dalam struct Sets yang sedang diproses melalui index i.

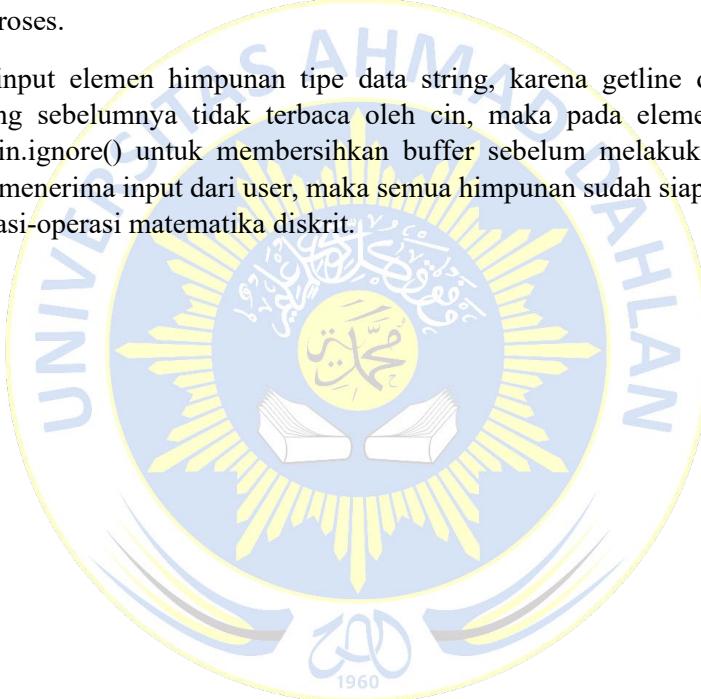
Setelah itu, fungsi inputSets akan menjalankan perulangan for sebanyak totalSets kali untuk menginput setiap himpunan secara urut. Pertama-tama, program akan menampilkan

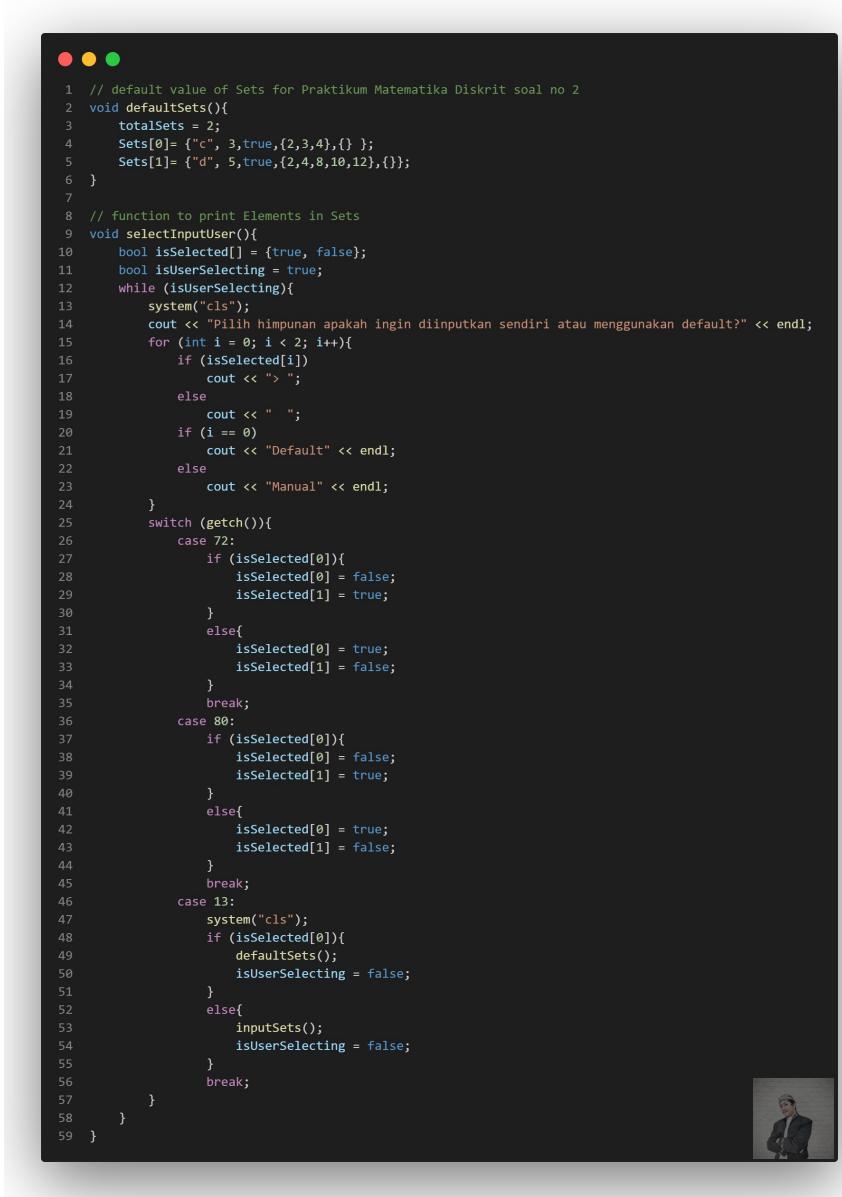
pesan ke layar untuk meminta pengguna memasukkan tipe data himpunan (int atau string) melalui cout.

Kemudian, dilakukan pengecekan apakah tipe data yang dimasukkan adalah int atau string. Jika tipe datanya adalah int, maka program akan memanggil fungsi totalElementsInSets dengan mengirimkan index himpunan saat ini sebagai argumen, kemudian menjalankan perulangan for lagi untuk menginput setiap elemen himpunan secara urut, dan menyimpan nilainya pada array intElements di dalam struct Sets yang sedang diproses.

Sedangkan jika tipe datanya adalah string, program juga akan memanggil fungsi totalElementsInSets dengan mengirimkan index himpunan saat ini sebagai argumen. Kemudian, program akan menjalankan perulangan for lagi untuk menginput setiap elemen himpunan secara urut, dan menyimpan nilainya pada array stringElements di dalam struct Sets yang sedang diproses.

Dalam input elemen himpunan tipe data string, karena getline dapat menangkap karakter '\n' yang sebelumnya tidak terbaca oleh cin, maka pada elemen pertama harus menggunakan cin.ignore() untuk membersihkan buffer sebelum melakukan input. Setelah program selesai menerima input dari user, maka semua himpunan sudah siap digunakan untuk melakukan operasi-operasi matematika diskrit.





```

1 // default value of Sets for Praktikum Matematika Diskrit soal no 2
2 void defaultSets(){
3     totalSets = 2;
4     Sets[0]= {"c", 3,true,{2,3,4},{} };
5     Sets[1]= {"d", 5,true,{2,4,8,10,12},{}} ;
6 }
7
8 // function to print Elements in Sets
9 void selectInputUser(){
10    bool isSelected[] = {true, false};
11    bool isUserSelecting = true;
12    while (isUserSelecting){
13        system("cls");
14        cout << "Pilih himpunan apakah ingin diinputkan sendiri atau menggunakan default?" << endl;
15        for (int i = 0; i < 2; i++){
16            if (isSelected[i])
17                cout << "> ";
18            else
19                cout << " ";
20            if (i == 0)
21                cout << "Default" << endl;
22            else
23                cout << "Manual" << endl;
24        }
25        switch (getch()){
26            case 72:
27                if (isSelected[0]){
28                    isSelected[0] = false;
29                    isSelected[1] = true;
30                }
31            else{
32                isSelected[0] = true;
33                isSelected[1] = false;
34            }
35            break;
36            case 80:
37                if (isSelected[0]){
38                    isSelected[0] = false;
39                    isSelected[1] = true;
40                }
41            else{
42                isSelected[0] = true;
43                isSelected[1] = false;
44            }
45            break;
46            case 13:
47                system("cls");
48                if (isSelected[0]){
49                    defaultSets();
50                    isUserSelecting = false;
51                }
52            else{
53                inputSets();
54                isUserSelecting = false;
55            }
56            break;
57        }
58    }
59 }

```

Potongan kode di atas adalah sebuah fungsi yang digunakan untuk menampilkan menu pada layar dan meminta user memilih apakah ingin menggunakan input default atau menginput sendiri. Fungsi ini diberi nama selectInputUser.

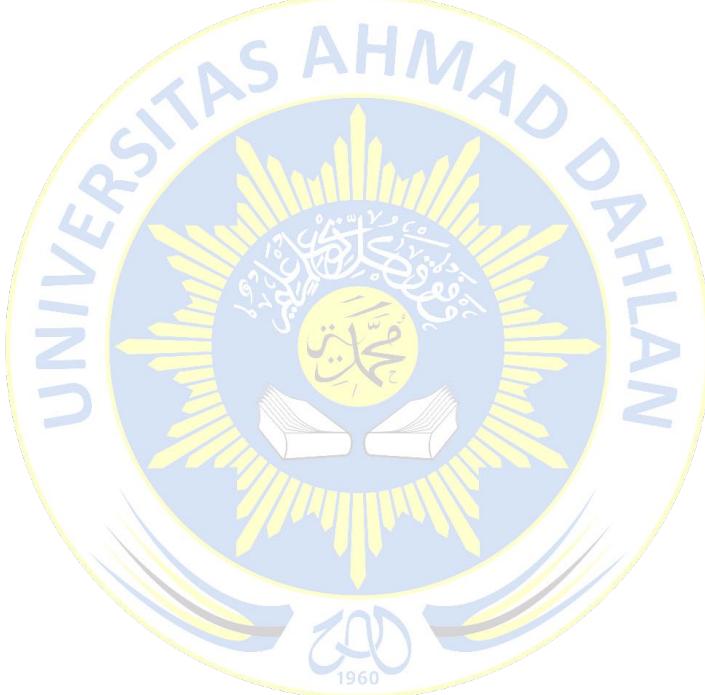
Fungsi tersebut akan menampilkan pesan ke layar dengan menggunakan cout, yaitu "Pilih himpunan apakah ingin diinputkan sendiri atau menggunakan default?" kemudian menampilkan dua pilihan dengan menggunakan perulangan for dan boolean array isSelected.

Setelah itu, program akan membaca input dari user dengan menggunakan getch() yang mengembalikan kode ASCII dari tombol keyboard yang ditekan. Jika kode ASCII yang ditekan

adalah 72 (tombol panah atas), maka program akan mengubah nilai array isSelected, jika sebelumnya bernilai false menjadi true dan sebaliknya. Sedangkan jika kode ASCII yang ditekan adalah 80 (tombol panah bawah), maka program juga akan mengubah nilai array isSelected.

Sedangkan jika kode ASCII yang ditekan adalah 13 (tombol enter), maka program akan mengecek nilai array isSelected. Jika isSelected[0] bernilai true, maka akan dipanggil fungsi defaultSets() yang akan mengisi struktur data himpunan secara otomatis. Namun jika isSelected[1] bernilai true, maka program akan meminta user memasukkan struktur data himpunan mereka sendiri melalui panggilan fungsi inputSets().

Setelah memproses pilihan user, program akan keluar dari perulangan while dan kembali ke program utama untuk melakukan operasi-operasi matematika diskrit yang diminta oleh pengguna.



```

1 // function to print Elements in Sets
2 void sortElements(int totalSets){
3     cout<<"Mengecek apakah elemen pada himpunan sudah terurut atau belum . ."<<endl;
4     sleep(2);
5     if (is_sorted(Sets[totalSets-1].intElements,Sets[totalSets-1].intElements+Sets[totalSets-1].totalElements)==true){
6         cout<<"Elemen sudah terurut"<<endl;return;
7     }
8     for(int i=0;i<totalSets;i++){
9         if (Sets[i].typeElements==true){
10            // check if the Elements in Sets is already sorted or not
11            if (is_sorted(Sets[i].intElements,Sets[i].intElements+Sets[i].totalElements)==false){
12                cout<<"Elemen pada himpunan "<<Sets[i].Name<<" belum berurut"<<endl;
13                sort(Sets[i].intElements,Sets[i].intElements+Sets[i].totalElements);
14                cout<<"======"<<endl;
15            }
16            sleep(1);
17        }
18        else{
19            // check if the Elements in Sets is already sorted or not
20            if (is_sorted(Sets[i].stringElements,Sets[i].stringElements+Sets[i].totalElements)==false){
21                cout<<"Elemen pada himpunan "<<Sets[i].Name<<" belum berurut"<<endl;
22                sort(Sets[i].stringElements,Sets[i].stringElements+Sets[i].totalElements);
23                cout<<"======"<<endl;
24            }
25            sleep(1);
26        }
27    }
28    if (is_sorted(Sets[totalSets-1].intElements,Sets[totalSets-1].intElements+Sets[totalSets-1].totalElements)==true){
29        cout<<"Elemen pada himpunan sudah diurutkan"<<endl;
30        cout<<"Setelah elemen diurutkan . ."<<endl;
31        for (int i=0;i<totalSets;i++){
32            cout<<Sets[i].Name<< "=";
33            for (int j=0;j<Sets[i].totalElements;j++){
34                // check terlebih dahulu apakah Elements in Sets adalah int atau string
35                if (Sets[i].typeElements==true){
36                    if (j==Sets[i].totalElements-1)
37                        cout<<Sets[i].intElements[j];
38                    else
39                        cout<<Sets[i].intElements[j]<< ",";
40                }
41                else{
42                    if (j==Sets[i].totalElements-1)
43                        cout<<Sets[i].stringElements[j];
44                    else
45                        cout<<Sets[i].stringElements[j]<< ",";
46                }
47            }
48            cout<<"}"<<endl;
49        }
50    }
51 }
52 // print list Elements in Sets
53 void printListElements(int totalSets){
54     for (int i=0;i<totalSets;i++){
55         if (Sets[i].typeElements==true){
56             cout<<"Elemen pada himpunan "<<Sets[i].Name<< "=";
57             for (int j=0;j<Sets[i].totalElements;j++){
58                 if (j==Sets[i].totalElements-1)
59                     cout<<Sets[i].intElements[j];
60                 else
61                     cout<<Sets[i].intElements[j]<< ",";
62             }
63             cout<<"}"<<endl;
64         }
65     }
66     else if (Sets[i].typeElements==false){
67         cout<<"Elemen pada himpunan "<<Sets[i].Name<< "=";
68         for (int j=0;j<Sets[i].totalElements;j++){
69             if (j==Sets[i].totalElements-1)
70                 cout<<Sets[i].stringElements[j];
71             else
72                 cout<<Sets[i].stringElements[j]<< ",";
73         }
74         cout<<"}"<<endl;
75     }
76 }
77 sortElements(totalSets);
78 }

```



Potongan kode di atas adalah sebuah fungsi untuk mengecek dan mengurutkan elemen pada setiap himpunan. Fungsi ini terdiri dari dua sub-fungsi: printListElements dan sortElements. Fungsi printListElements akan menerima argumen totalSets yang merepresentasikan jumlah himpunan yang telah dimasukkan pada program. Fungsi tersebut akan menjalankan perulangan for sebanyak totalSets kali untuk menampilkan semua elemen himpunan yang sudah dimasukkan oleh pengguna. Jika tipe datanya adalah int, maka program akan menampilkan elemen pada himpunan dengan tipe data int. Namun jika tipe datanya adalah string, maka program akan menampilkan elemen pada himpunan dengan tipe data string.

Setelah itu, program akan memanggil fungsi sortElements dengan mengirimkan totalSets sebagai argumen. Fungsi sortElements akan menampilkan pesan ke layar "Mengecek apakah elemen pada himpunan sudah terurut atau belum . . ." kemudian mengecek satu per satu himpunan yang ada. Jika elemen pada himpunan sudah terurut, maka program akan menampilkan pesan "Elemen sudah terurut" dan langsung keluar dari fungsi sortElements.

Namun jika elemen pada himpunan belum terurut, maka program akan menampilkan pesan "Elemen pada himpunan ... belum berurut", kemudian akan mengurutkan elemen pada himpunan tersebut menggunakan fungsi sort() bawaan C++. Setelah itu, program akan menampilkan ulang seluruh elemen himpunan yang telah diurutkan.

Setelah seluruh elemen pada himpunan diurutkan, program akan menampilkan pesan "Elemen pada himpunan sudah diurutkan" dan kemudian menampilkan ulang seluruh elemen himpunan beserta dengan urutan yang baru. Setelah itu, program keluar dari fungsi printListElements dan kembali ke program utama untuk melakukan operasi-operasi matematika diskrit yang diminta oleh pengguna.

```

1 // function to print Elements in Sets
2 void calcRelation(){
3     cout << "======" << endl;
4     cout << "R adalah relasi gabungan dari antara setiap pasangan" << endl;
5     for (int i = 0; i < totalSets; i++){
6         for (int j = 0; j < totalSets; j++){
7             if (i != j){
8                 cout << Sets[i].Name << " R " << Sets[j].Name << " = {";
9                 for (int k = 0; k < Sets[i].totalElements; k++){
10                     for (int l = 0; l < Sets[j].totalElements; l++){
11                         // check terlebih dahulu apakah Elements in Sets adalah int atau string
12                         if (Sets[i].typeElements==true){
13                             cout << "" << Sets[i].intElements[k] << "," << Sets[j].intElements[l] << ")";
14                             if (k != Sets[i].totalElements - 1 || l != Sets[j].totalElements - 1){
15                                 cout << ",";
16                             }
17                         }
18                         else{
19                             cout << "(" << Sets[i].stringElements[k] << "," << Sets[j].stringElements[l] << ")";
20                             if (k != Sets[i].totalElements - 1 || l != Sets[j].totalElements - 1){
21                                 cout << ",";
22                             }
23                         }
24                     }
25                 }
26                 cout << ")" << endl;
27             }
28         }
29     }
30     cout << "======" << endl;
31 }
32
33 void calcDivisible(){
34     for (int i = 0; i < totalSets; i++){
35         if (Sets[i].typeElements==false){
36             cout<<"Beberapa himpunan mungkin bertipe data string, sehingga tidak dapat dibagi . ."<<endl;
37             return;
38         }
39     }
40     for (int i = 0; i < totalSets; i++){
41         for (int j = 0; j < totalSets; j++){
42             if (i != j){
43                 cout << Sets[i].Name << " habis membagi " << Sets[j].Name << " = {";
44                 for (int k = 0; k < Sets[i].totalElements; k++){
45                     for (int l = 0; l < Sets[j].totalElements; l++){
46                         if (Sets[j].intElements[l] % Sets[i].intElements[k] == 0){
47                             cout << "(" << Sets[i].intElements[k] << "," << Sets[j].intElements[l] << ")";
48                             if (k != Sets[i].totalElements - 1 || l != Sets[j].totalElements - 1){
49                                 cout << ",";
50                             }
51                         }
52                     }
53                 }
54                 cout << ")" << endl;
55             }
56         }
57     }
58     cout << "======" << endl;
59 }
```



Potongan kode di atas adalah dua buah fungsi yang digunakan untuk menghitung relasi antara elemen pada setiap himpunan serta mengecek apakah suatu himpunan habis membagi himpunan lainnya atau tidak.

Fungsi pertama, yaitu calcRelation(), akan menampilkan pesan ke layar "R adalah relasi gabungan dari antara setiap pasangan" dan kemudian menjalankan dua perulangan for bersarang. Perulangan for yang pertama akan menjalankan variabel i untuk memilih himpunan pertama yang akan dibandingkan dengan himpunan kedua. Sedangkan perulangan for yang kedua akan menjalankan variabel j untuk memilih himpunan kedua yang akan dibandingkan dengan himpunan pertama.

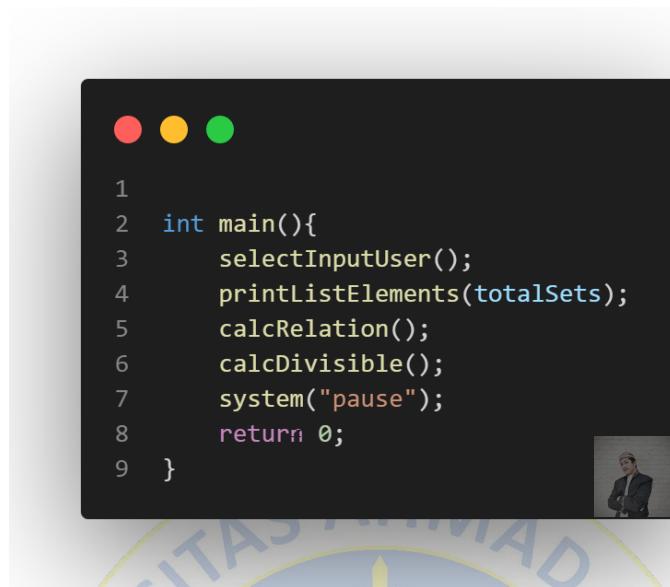
Selanjutnya, program akan mengecek apakah variabel i sama dengan variabel j atau tidak. Jika variabel i tidak sama dengan variabel j, maka program akan menampilkan pesan yang berisi nama himpunan pertama, tanda R (relasi), dan nama himpunan kedua. Setelah itu, program akan menjalankan dua perulangan for bersarang lagi untuk mengecek setiap kombinasi elemen pada himpunan pertama dan himpunan kedua.

Ketika menemukan elemen yang cocok, program akan menampilkan pasangan elemen dalam bentuk kurung siku, misalnya (2,5) dan (3,6). Jika data ketika menjelajahi set bernilai integer maka elemen himpunan ditampilkan dalam format integer (angka), namun jika datanya bertipe string, maka elemen himpunan akan ditampilkan dalam format string (teks).

Fungsi kedua, yaitu calcDivisible(), akan mengecek apakah suatu himpunan habis membagi himpunan lainnya atau tidak. Jika ada himpunan yang menggunakan tipe data string, maka program akan menampilkan pesan ke layar "Beberapa himpunan mungkin bertipe data string, sehingga tidak dapat dibagi . . ." dan keluar dari fungsi.

Namun jika seluruh himpunan memiliki tipe data integer, maka program akan menjalankan dua perulangan for bersarang untuk mengecek setiap kombinasi elemen pada himpunan pertama dan himpunan kedua. Ketika menemukan elemen yang habis membagi elemen lainnya, program akan menampilkan pasangan elemen dalam bentuk kurung siku, misalnya (2,4) dan (3,6).

Setelah menampilkan semua hasil yang ditemukan, program akan menampilkan garis pemisah "===== lalu keluar dari fungsi dan kembali ke program utama untuk melakukan operasi-operasi matematika diskrit yang diminta oleh pengguna.



```
1
2 int main(){
3     selectInputUser();
4     printListElements(totalSets);
5     calcRelation();
6     calcDivisible();
7     system("pause");
8     return 0;
9 }
```

Potongan kode di atas adalah sebuah fungsi utama (main function) yang akan dipanggil ketika program dijalankan. Fungsi ini akan menjalankan empat buah fungsi lainnya, yakni selectInputUser(), printListElements(), calcRelation(), dan calcDivisible().

Fungsi selectInputUser() akan dijalankan terlebih dahulu untuk meminta input dari pengguna mengenai jumlah himpunan yang ingin dibuat dan elemen yang ingin dimasukkan pada setiap himpunan tersebut.

Selanjutnya, fungsi printListElements() akan dijalankan untuk menampilkan seluruh elemen yang sudah dimasukkan oleh pengguna pada setiap himpunan. Kemudian, fungsi calcRelation() dan calcDivisible() akan dijalankan secara bergantian untuk menghitung relasi antara himpunan-himpunan yang telah dimasukkan oleh pengguna.

Setelah semua perhitungan selesai dilakukan, program akan menampilkan pesan "Press any key to continue . . ." dan menunggu pengguna untuk menekan tombol apa saja sebelum keluar dari program. Akhirnya, fungsi main() akan mengembalikan nilai 0 dan program akan berakhir.

Tampilkan hasil himpunan D habis membagi C!



A screenshot of Visual Studio Code showing a C++ code editor and terminal window. The code editor displays a file named 'postest2.cpp' with code related to sets and relations. The terminal window shows the output of the program, which includes printing elements of sets 'c' and 'd', calculating their intersection 'R', and checking if 'R' is a subset of 'd'. The output also includes a check if 'c' divides 'd'. The terminal window ends with the command 'PS D:\Document Ndik\Kuliah\Semester 2\Matematika Diskrit\Praktikum\Pertemuan 2> []'.

```
File Edit Selection View Go Run Terminal Help postest2.cpp - Visual Studio Code - Insiders [Administrator]
prak2.cpp X Post1.cpp

D:\> Document Ndik > Kuliah > Semester 2 > Matematika Diskrit > Praktikum > Pertemuan 2 > postest2.cpp > calcRelation()

202     if (j==Sets[1].totalElements-1)
203         cout<<Sets[1].stringElements[j];
204     else
205         cout<<Sets[1].stringElements[j]<">";
206     }
207     cout<<"}"<<endl;
208 }
209
210 sortElements(totalSets);
211 }

212 // function to print Elements in Sets
213 void calcRelation(){
214     cout << "====="
215     cout << endl;

PROBLEMS TERMINAL OUTPUT DEBUG CONSOLE
powerhell
Code

Element pada himpunan c = {2,3,4}
Element pada himpunan d = {2,4,8,10,12}
Mengecek apakah elemen pada himpunan sudah terurut atau belum . .
Elemen sudah terurut
=====
R adalah relasi gabungan dari antara setiap pasangan
c R d = {(2,2),(2,4),(2,8),(2,10),(2,12),(3,2),(3,4),(3,8),(3,10),(3,12),(4,2),(4,4),(4,8),(4,10),(4,12)}
d R c = {(2,2),(2,3),(2,4),(4,2),(4,3),(4,4),(8,2),(8,3),(8,4),(10,2),(10,3),(10,4),(12,2),(12,3),(12,4)}
=====
c habis membagi d = {(2,2),(2,4),(2,8),(2,10),(2,12),(3,12),(4,4),(4,8),(4,12)}
d habis membagi c = {(2,2),(2,4),(4,4)}
=====

PS D:\Document Ndik\Kuliah\Semester 2\Matematika Diskrit\Praktikum\Pertemuan 2> []

Ln 226, Col 89 Spaces: 4 UFT-8 CRLF C++ Go Live Win32 Prettier
```